

Application Performance Management

Skalierung & Load Balancing

Michael Faes

Modul-Übersicht 2. Teil

Technische Grundlagen für performante Web-Apps

- Skalierung & Load Balancing
- Clustering
- Virtualisierung/Containerisierung
- Caching
- Autoscaling

Performance im Web

- Mess-Setup
- Arten von Performance-Tests

Praktisch

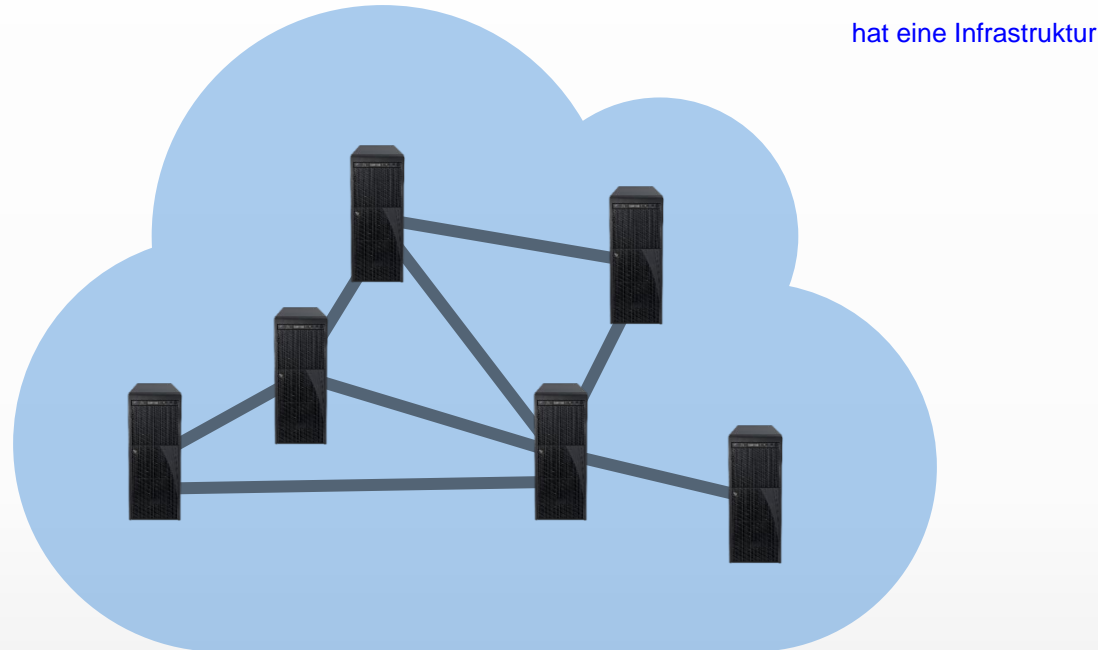


Aufbau einer «Cloud»-
Infrastruktur

Entwickeln einer skalierbaren,
hochverfügbaren Web-App

Umgang mit Last-Generator,
Messungen durchführen

Was ist eigentlich eine «Cloud»?



“I don’t like the term ‘cloud’; it’s *nebulous*.”

— Richard Stallman

“There is no cloud; it’s just someone else’s computer.”

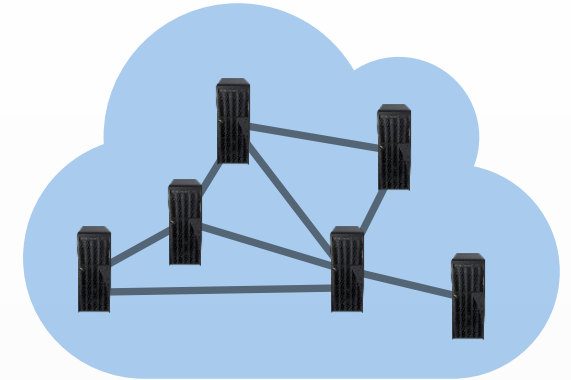
— unbekannt

Cloud Computing ist ein

Modell für «Selbstbedienung» von geteilten Rechen-Ressourcen über ein Netzwerk.

Beschaffung und Abgabe von Ressourcen findet schnell und ohne Provider-Interaktion statt.

(ungefähre Definition von NIST)



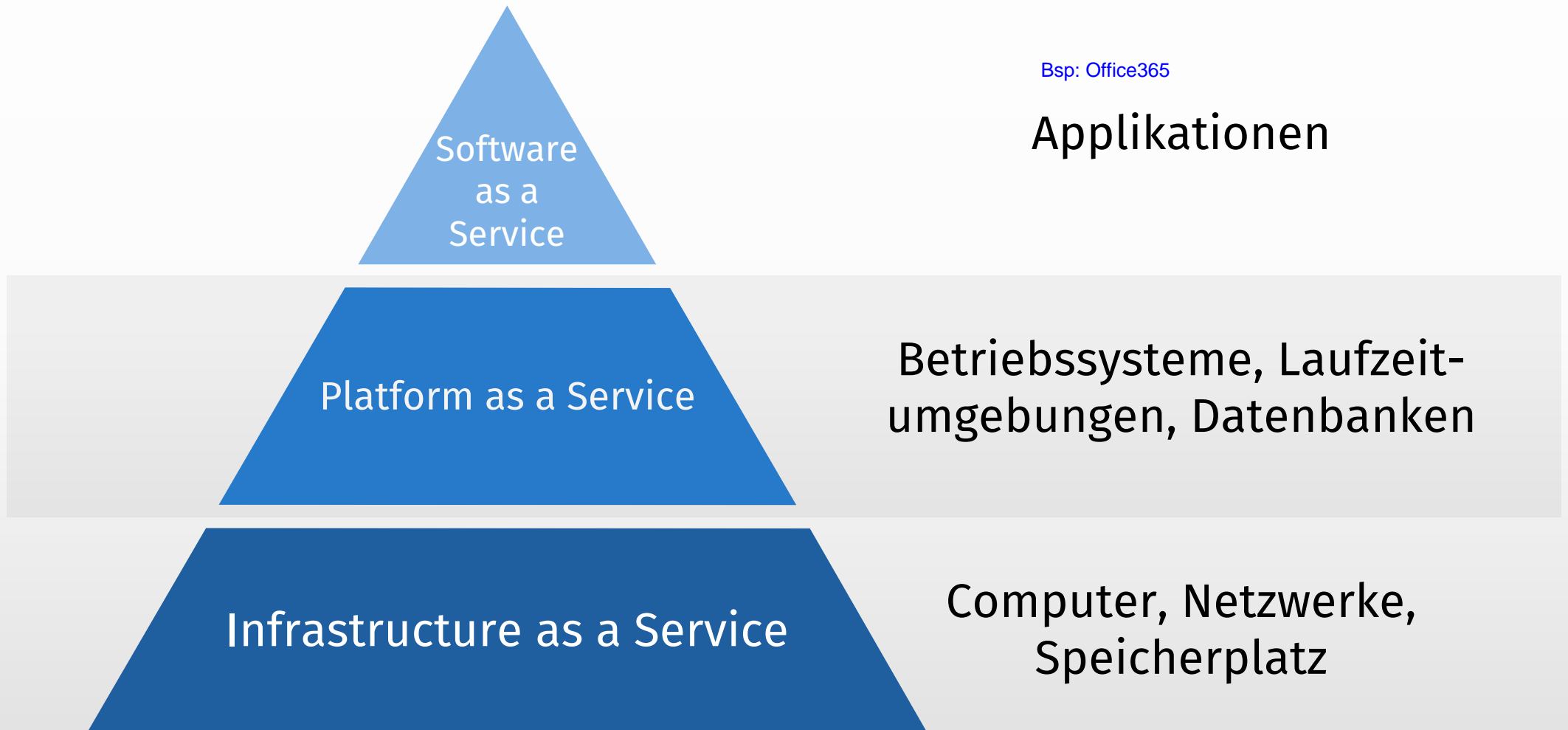
5 Hauptmerkmale:

- Selbstzuweisung von Ressourcen
- Gemeinsame Ressourcen-Nutzung
- Zugriff über Internet (Standard-Protokolle)
- Schnelles Skalieren von Ressourcen
- Messen und Überwachen von Nutzung



werden auf diese Aspekte fokussieren

Service-Modelle



Übersicht Woche 9

1. Einführung Modul-Teil 2
2. Web-Apps & Skalierung
3. Load-Balancing
 - Methoden
 - Monitoring
 - Persistenz
4. Übung

Web-Apps & Skalierung

11 I've been hearing about the Millenium Bug (AKA the Y2K bug) and I'm not sure what to change about my program to prepare for it. I know it has to do with the dates, but there might also be something to do with leap years?

Here's what I know so far:

- On January 1st, 2000, programs that use double digits for years that assume 1900 will change from 1999 to 1900 instead of 2000. That shouldn't happen in my program.
- I should update any dates in my database that assume 1900 to explicitly specify the full year.
- February is a leap year?? So something should change?

Are these all the issues related to Y2K? I've been asking around on IRC, but everyone's giving me different answers.

y2k

[share](#) [improve this question](#)

[edited yesterday](#)

asked 2 days ago

 [Jon Chan](#) ♦

552 ● 1 ● 5 ● 23

4 [letmeaskjeeveshatforyou.com](#) - Adam Lear 2 days ago

2 My company hired a consultant, but I don't think it'll matter when the apocalypse comes. I'm putting my savings into gold and canned goods. - Jon Ericson 2 days ago

1 possible duplicate of [Stackoverflow.com/questions/18200744/... - m0sa 6 yesterday](#)

Possible duplicate of [What Does Y2K Complaint Mean? - Vinko Vrsalovic yesterday](#)

[add a comment](#)

5 Answers [active](#) [oldest](#) [votes](#)

▲ If you store your dates in Unix time as a signed 32-bit binary integer, that'll get you until 03:14:07 UTC on Tuesday, 19 January 2038 which should, like, be totally good enough.

[share](#) [improve this answer](#)

answered 2 days ago



[Meg Risdal](#)
106 • 1

That sounds perfect. I'll give that a shot! – [Jon Chan](#) yesterday

And if you need more, just store 2 signed 32-bit binary registers side by side in your SQL 7 table. That way you can use one as an overflow register and be good practically forever! – [Haneey](#) yesterday

[add a comment](#)

asked 2 days ago
viewed 1 times
active yesterday

BLOG

 [The Next CEO of Stack Overflow](#)

The image shows the Pyroflex Corporation logo, which consists of a blue stylized 'P' icon and the text 'Pyroflex Corporation' in a bold, sans-serif font. Below the company name is the text 'Computer Software'. Underneath the logo, there are three job listings. The first listing is for a 'Cryptocurrency Software Engineer' in 'Crem, UT' with a salary of '\$50k - \$100k' and a 'Remote' tag. The second listing is for a 'Linux Systems Administrator' in 'Crem, UT' with a salary of '\$50k - \$85k' and a 'Paid relocation' tag. The third listing is for a 'Director of Engineering' in 'Crem, UT' with a salary of '\$80k - \$110k' and a 'Paid relocation' tag. Each listing has two tags: 'scala' and 'linux'. At the bottom of the image is a light blue button with the text 'View all 3 job openings!'.

Linked

5 What Does Y2K Compliant Mean?

Related

0 Preferred method for handling Y2K date conversions?

2 Win32: How to prevent DateTimePicker from accepting 2-digit years?

1 How Do I Convert a Y2K Date to SQL DATE In SSIS?

8 How to parse string dates with 2-digit year?

 [question feed](#)

Name:	<div>FIHHHHIARRSSSSSTTTTTT</div> <div>@meg - 2019/04/01</div>
Comment:	<div>I think they got haxxxed</div> <div>@josh - 2019/04/01</div>

Motivation

Das Web hat sich verändert

- Tausende → 100 Millionen Users
- Schnellere Endgeräte
- **Statisch → Dynamisch**

facebook

 stackoverflow

twitter

 reddit


Instagram

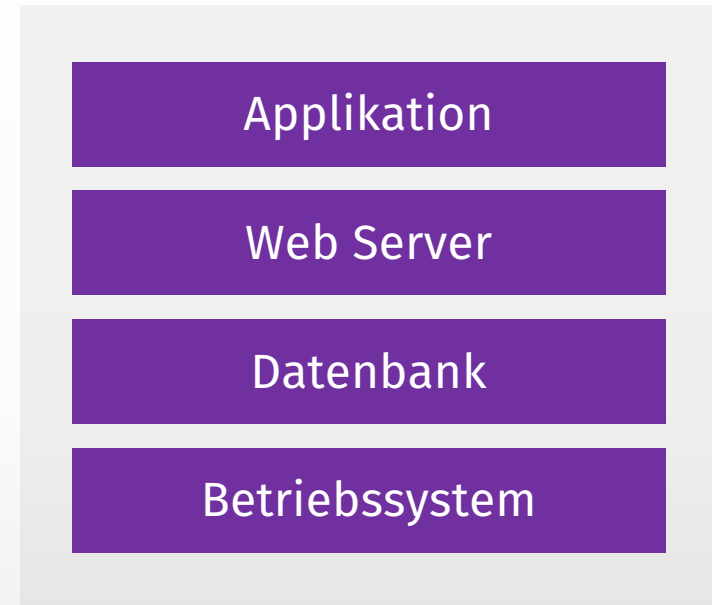
Google

Rückblick: Performance-Grundprobleme



Probleme der Last

z.B. zu viel Benutzer (bei Webapp)

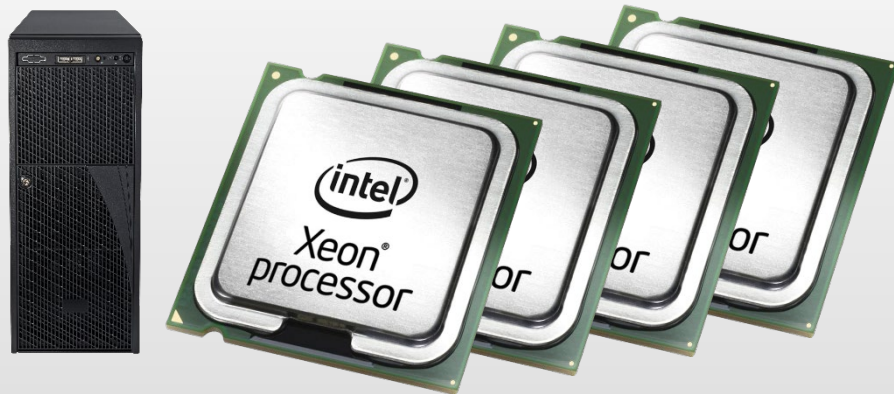


Probleme der Architektur/Implement.

Selbst ohne Probleme mit **Architektur, Implementation, Konfiguration**, usw., kann Performance leiden, wenn **Last** zu gross ist!

Skalierung

Scale-up



Ersetzen und Erweitern von Prozessoren

Limitation auf Hardware

Scale-out



Fügen mehrere Maschinen hinzu.

Braucht entsprechende Architektur

Scale-up vs. Scale-out

Scale-up

Teuer

Wenig flexibel

Einfach, App muss (evtl.)
nicht angepasst werden

Irgendwann ist
Grenze erreicht...

Scale-out

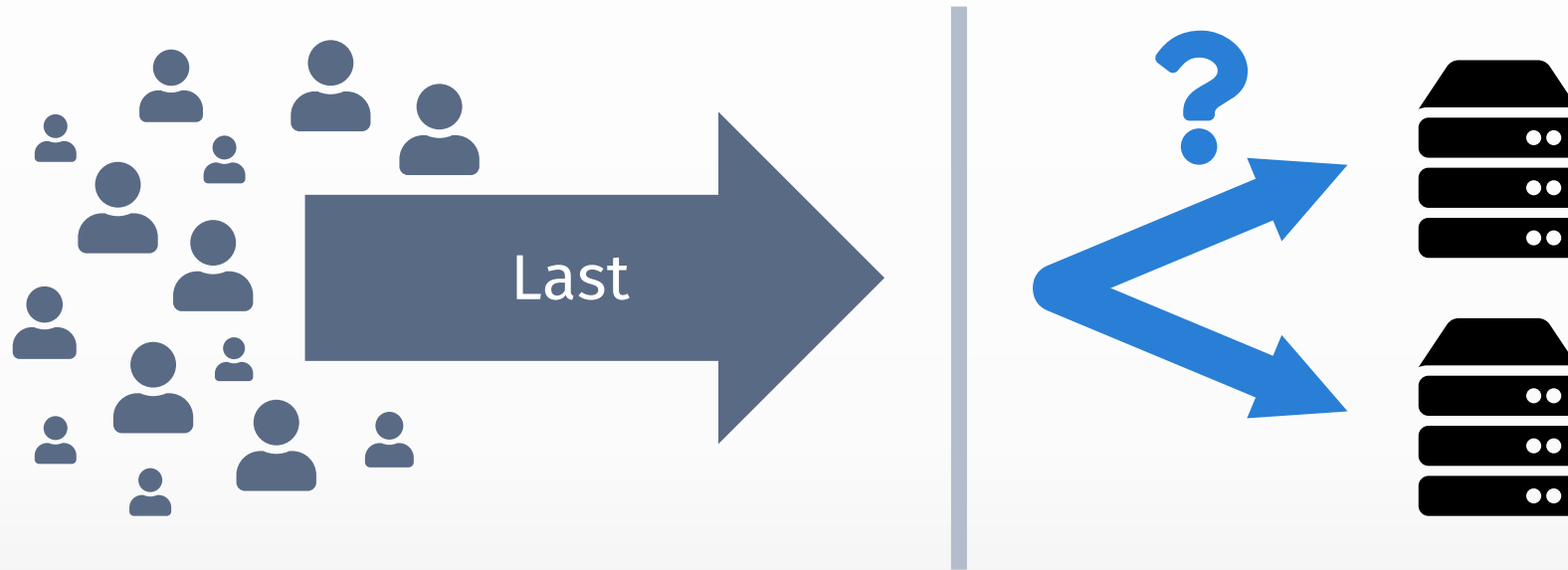
Relativ günstig, flexibel

App muss an verteiltes
Setting angepasst werden!

Prinzipiell immer machbar
(aber nicht jede App
ist parallelisierbar)

→ *Längerfristig einziger Weg*

Verteilen der Last



Last muss auf mehrere Server verteilt werden! Aber wie?

Einfache Methode: «Vorverteilen» der Benutzer

- DNS! Unterschiedliche Server-Adressen verteilen
- Nachteile: Wenig Kontrolle, keine Lösung für hohe Verfügbarkeit



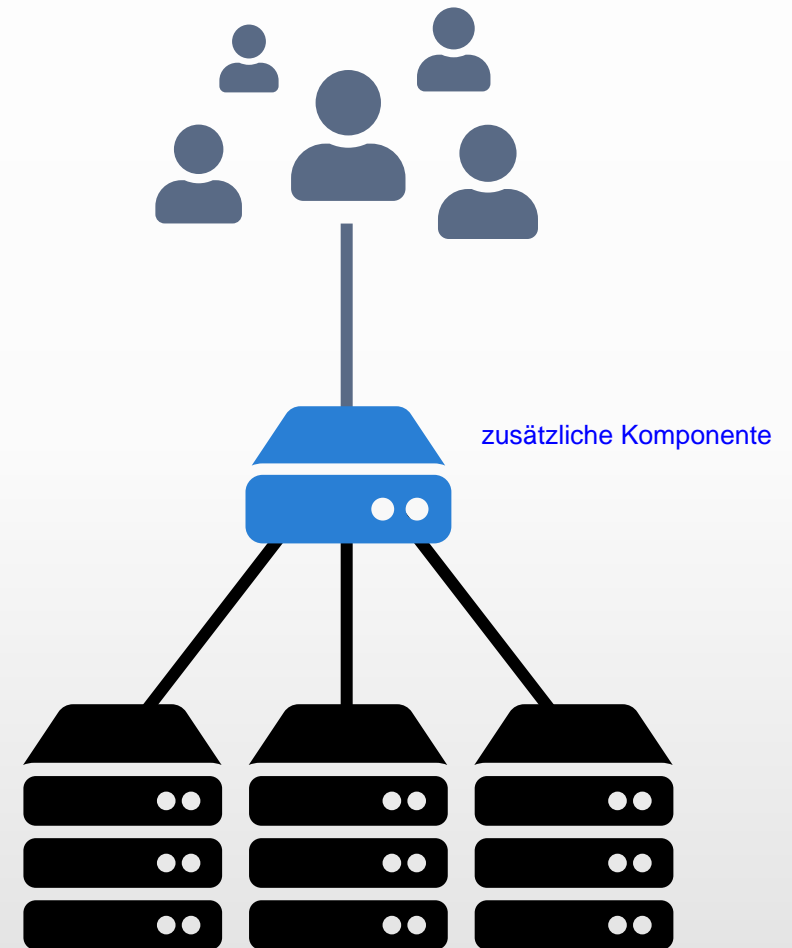
Load Balancing

Load Balancing: Server-Requests werden dynamisch auf mehrere Server verteilt

Neue Komponente: *Load Balancer*

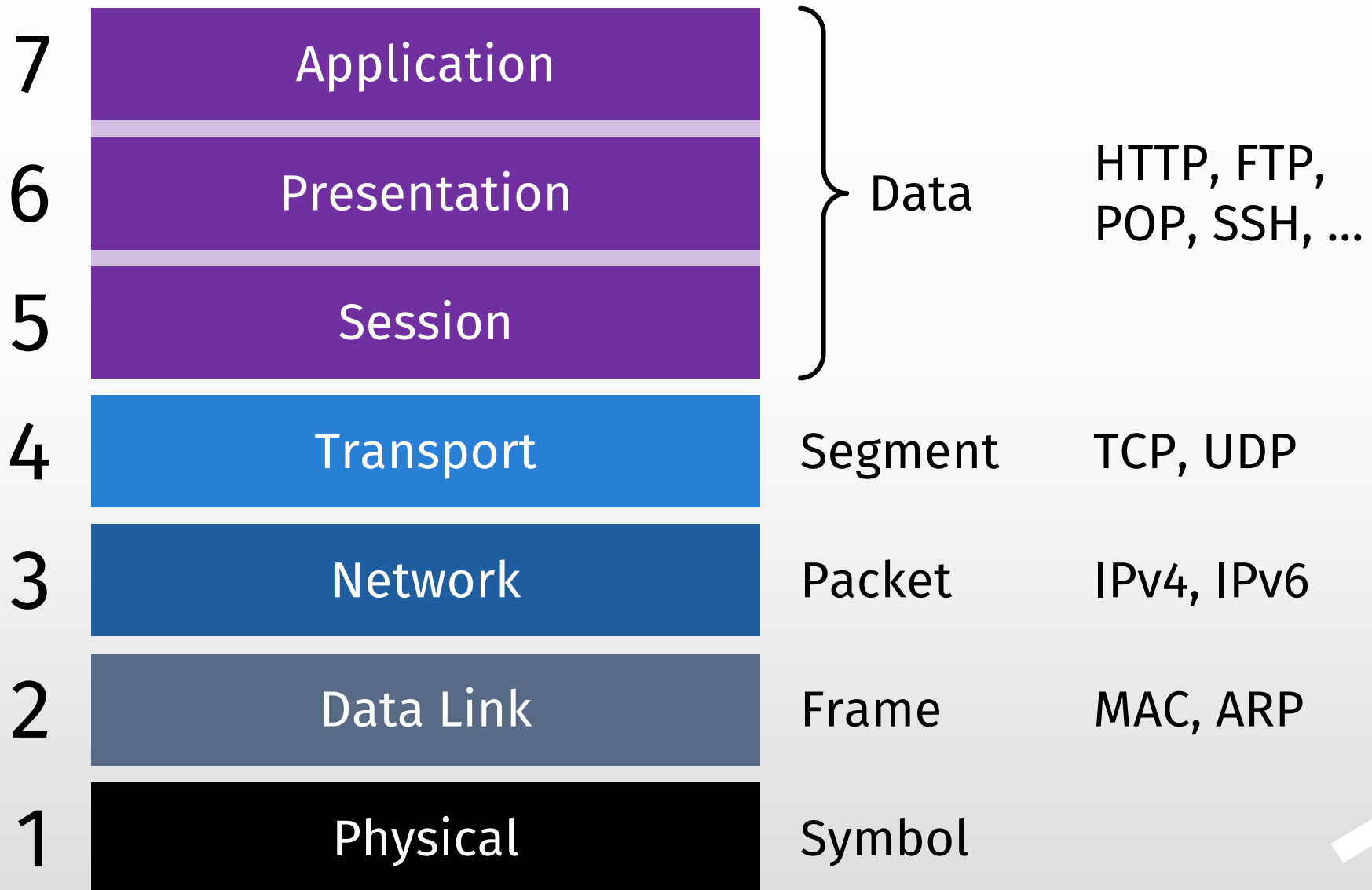
Separate Maschine

- Kann auch virtuell sein
- Einstiegspunkt des Dienstes
- Hardware- oder Software-basiert



Load-Balancing-Methoden

Refresher: OSI Modell



LB-Methode: *Direct Routing* (L2/3)

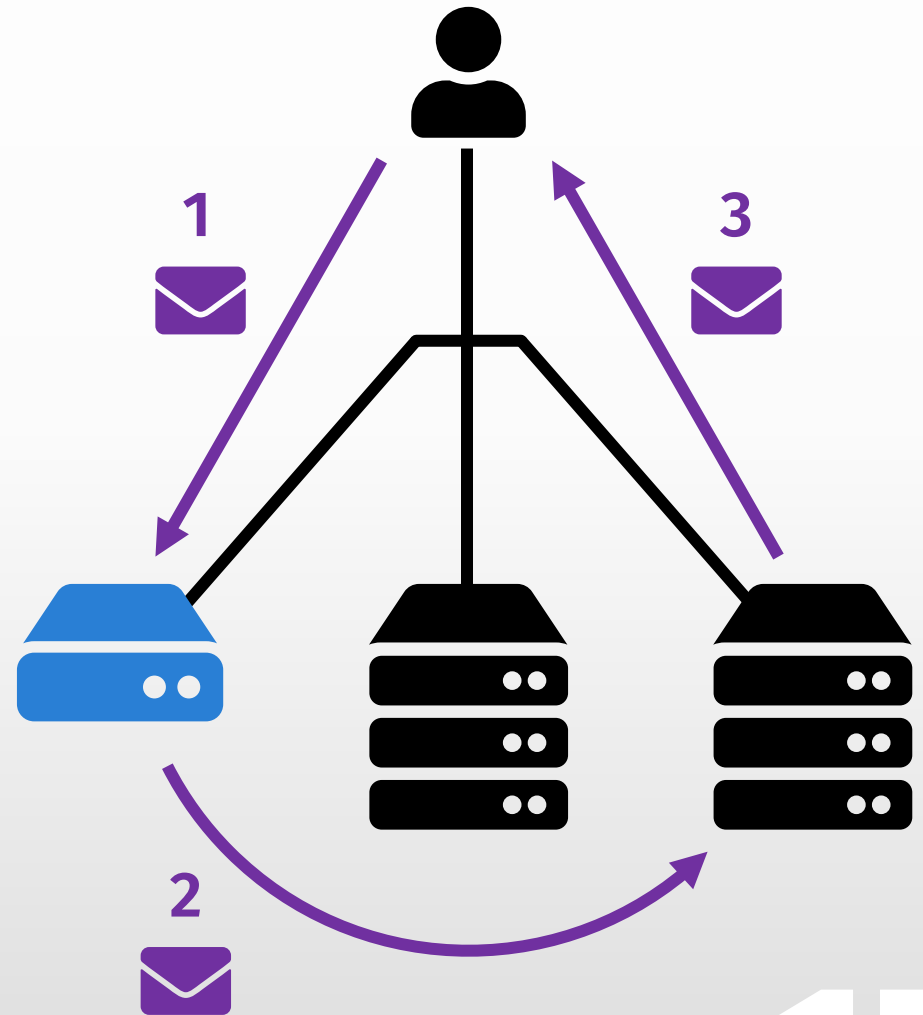
LB und Server **im selben Subnetz**, alle haben **gleiche IP konfiguriert**

- **Nur LB antwortet auf ARP-Requests**, d.h. alle Packets landen bei ihm

LB ändert Frame, schickt Packet an **Server weiter**

Server antwortet direkt an Client!

Vorteil: Oberhalb von IP-Level (Layer 3+) muss nichts an Request geändert werden!



LB-Methode: *NAT* (Layer 3/4)

LB «trennt» Client von Server, welche in privatem Subnet sind

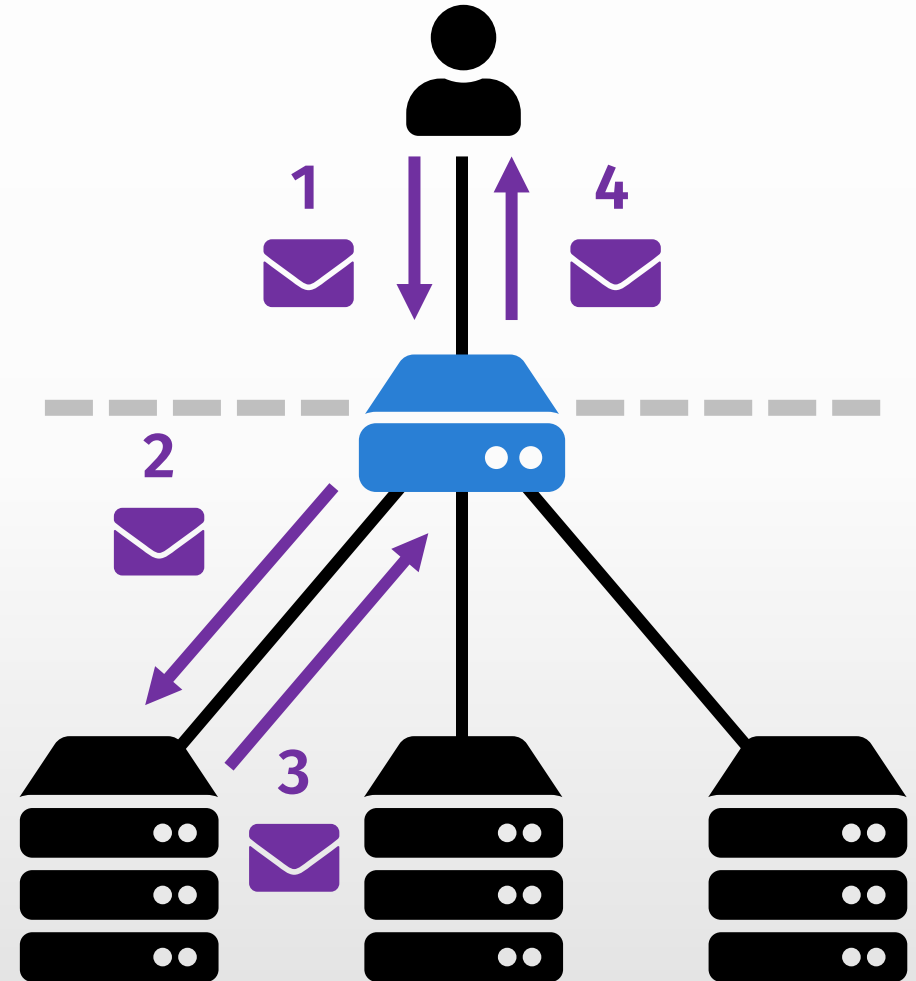
Wenn LB Request erhält, übersetzt er einfach Ziel-IP-Adresse (über Router)

Vorteile:

- Einfachere Konfiguration
- Möglichkeit für Traffic Inspection

Nachteil:

- App muss damit umgehen können, dass Client andere IP-Adresse sieht



LB-Methode: *Reverse Proxy* (L7)

Trennung wie bei NAT, aber Requests werden auf Application Layer decodiert

Beispiel HTTP: Client baut HTTP-Verbindung mit LB auf, LB mit Server

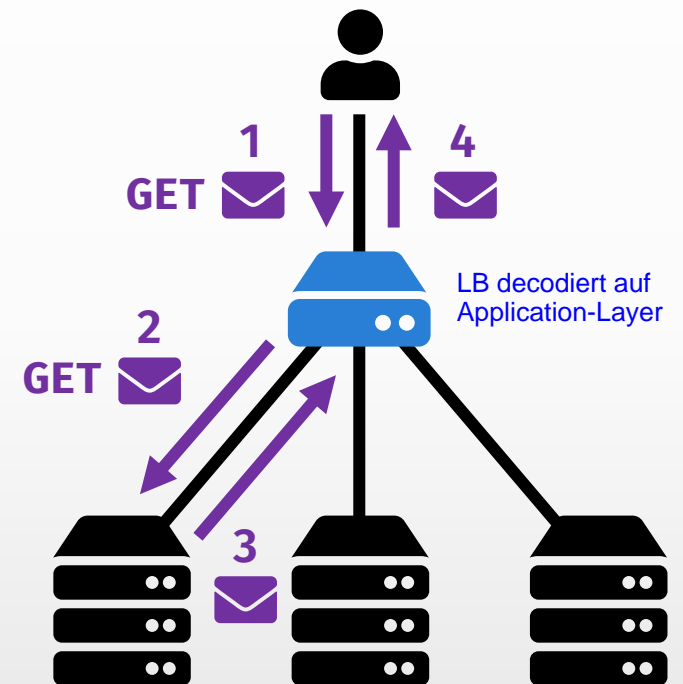
Vorteile:

Session-aware (mit Cookies)

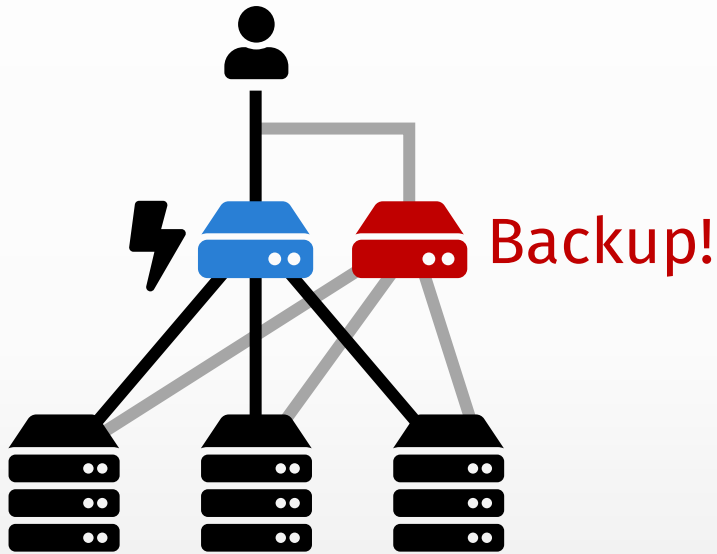
- Balancing-Entscheid basierend auf App-Informationen (z. B. Cookies) möglich
- LB kann TLS und Caching übernehmen

Nachteil:

- Ressourcen-intensiver

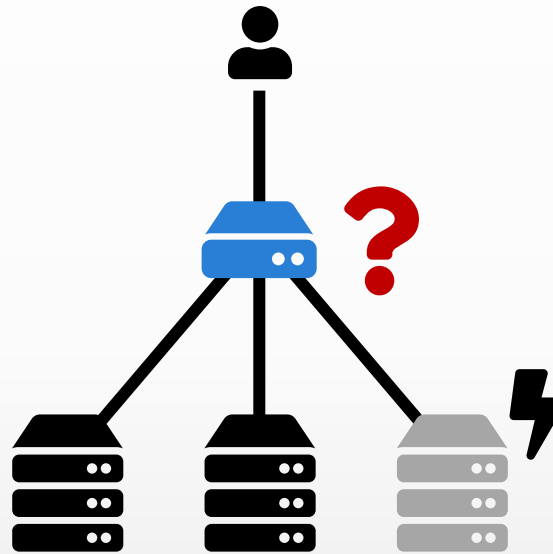


Load Balancing: Herausforderungen

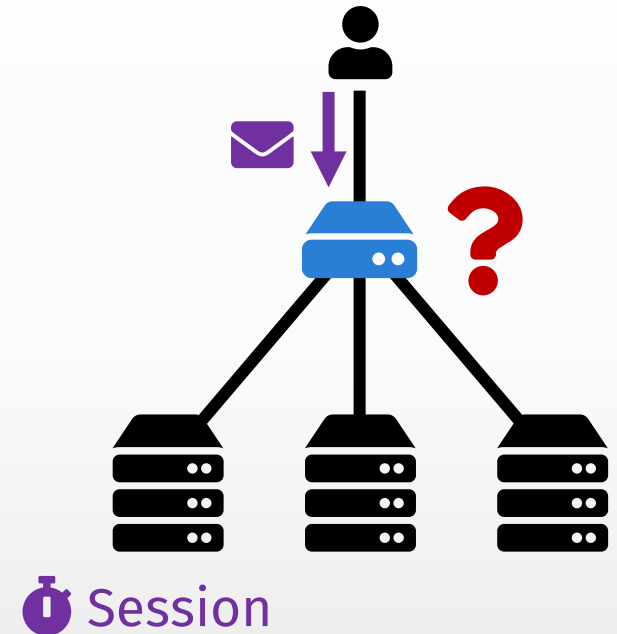


LB-Verfügbarkeit

nächste Woche



Monitoring
(Health Checks)



Persistenz

selber Benutzer landet auf selben Server

Monitoring

Damit Load Balancer weiss, welche Server verfügbar sind, kann er regelmässige *Health Checks* durchführen

Welche Art von Health Checks?

Pings

TCP-Verbindungsaufbau

HTTP-Requests

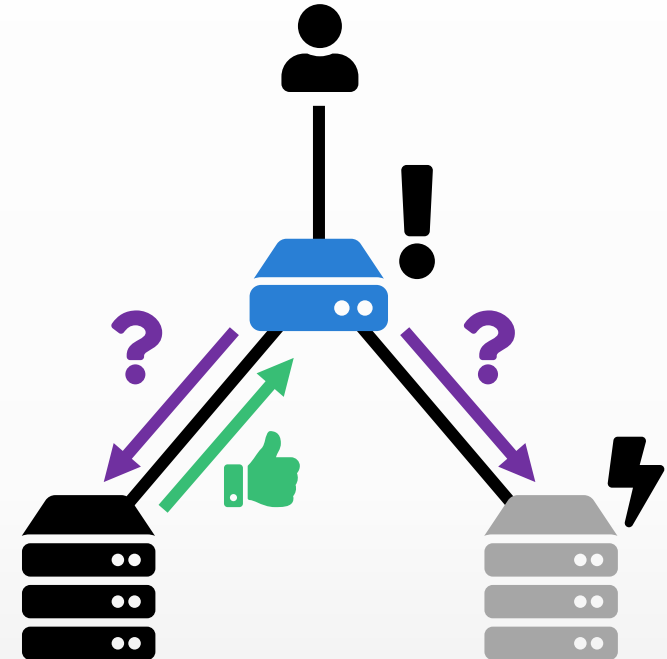
- Welche?

Wieder ein
Trade-Off:

**umfassendere,
häufigere Checks**
aktuellere, genauere Info



**einfachere,
seltener Checks**
kleinere Belastung



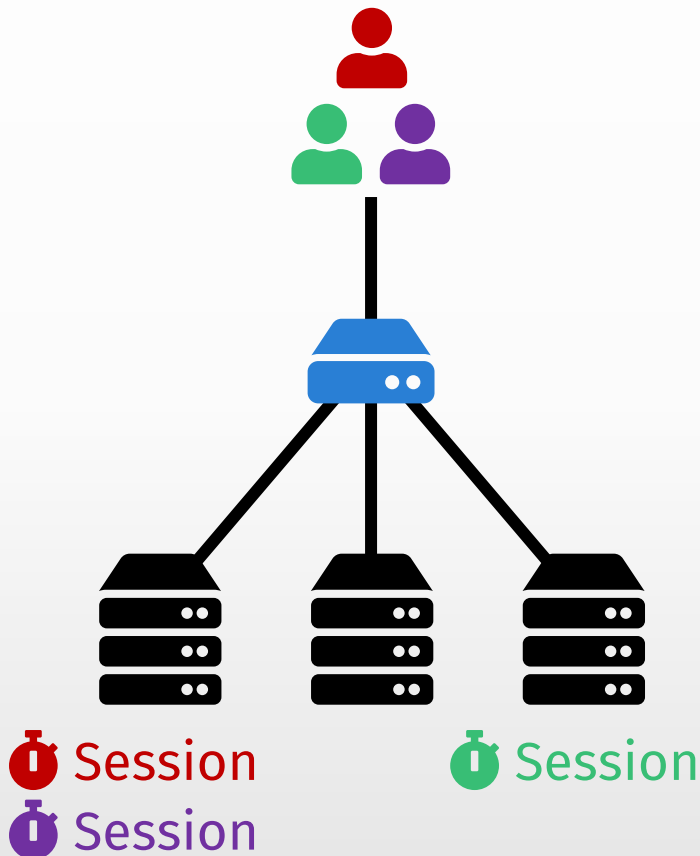
Persistenz

LB sollte sicher stellen, dass Requests von selbem Client immer auf gleichem Server landen!

(Zu) Einfach: Zu Beginn HTTP-3xx-Umleitung zu einem Server

- Geht nicht für getrennte Netze

Besser: LB muss Client-Server-Verhältnis «lernen»



Persistenz: Methoden

IP-Layer

- LB erstellt Tabelle mit Zuordnung **Client-IP → Server**
- Einfach, aber keine Lösung für Clients mit wechselnder IP!

Cookie Learning (HTTP-Layer)

- LB inspiziert HTTP-Request (Session-Cookie), erstellt Tabelle mit Zuordnung **Session-ID → Server**
- Probleme: 1) Endlicher Speicher, 2) Ausfall von Master-LB

Cookie Insertion (HTTP-Layer)

- LB fügt *eigenes Cookie* in HTTP-Messages ein
- Löst beide Probleme, aber ist rechenintensiver

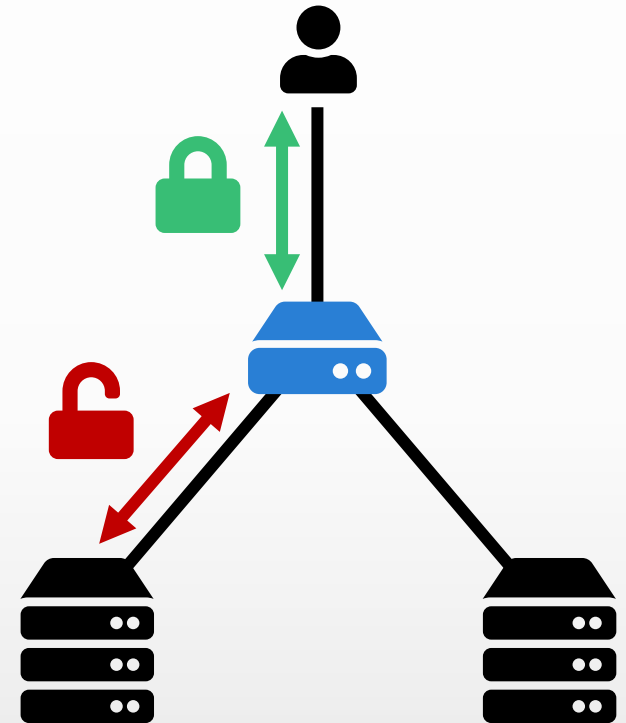
Persistenz und HTTPS/TLS

HTTP-Layer-Persistenz erfordert
Einsicht/Änderung von HTTP-Messages.

Was, wenn Traffic verschlüsselt ist?

Mögliche Lösung: TLS kann auf Load
Balancer gemacht werden

- LB ist Reverse Proxy, der zwischen HTTPS und HTTP «konvertiert»
- Entlastet Server, aber belastet einzigen Load Balancer...



Persistenz und HTTPS/TLS

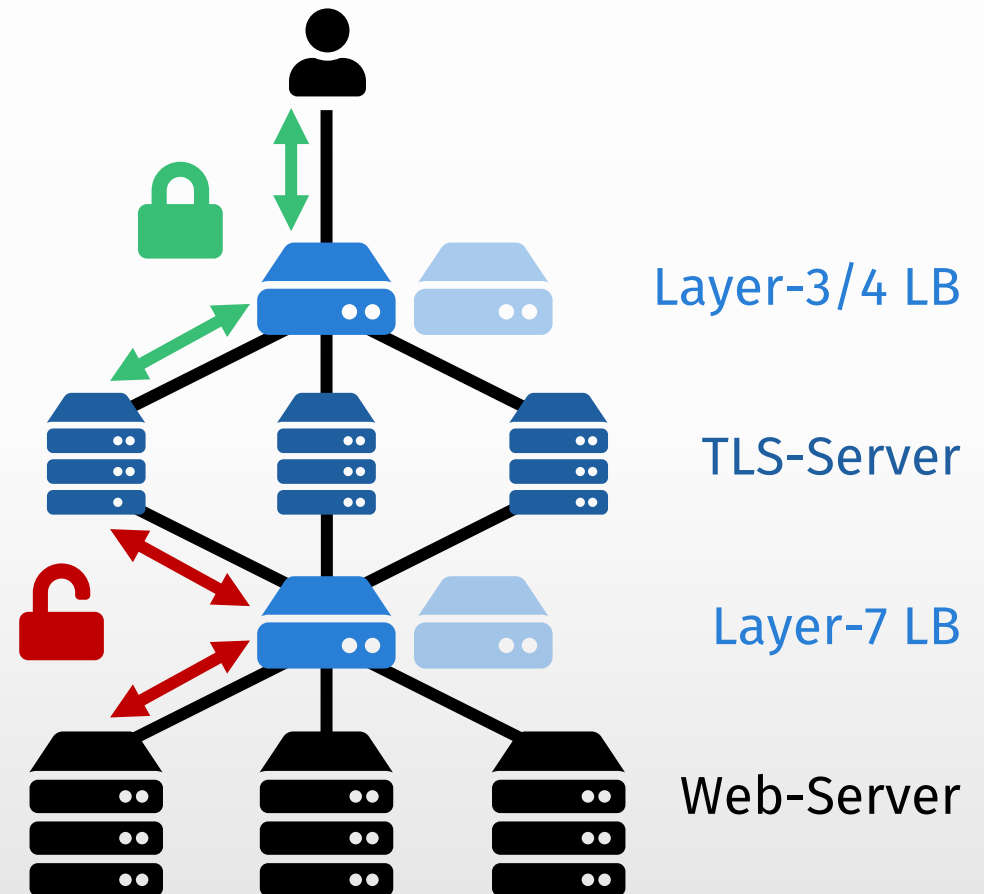
TLS auf Load Balancer: LB kann Bottleneck werden!

Mehrere Load Balancers?

- Nicht empfehlenswert: Farm von LBs verwalten ist schwierig
- Jeder LB muss Health Checks durchführen!

Lösung: *TLS-Farm!*

- Billig, entkoppelt, flexibel
- Aber auch komplex...



Fragen?

