embarc
Software Consulting GmbH

Netflix - xilfteN

Die umgekehrte Architekturbewertung eines Internet-Giganten

STEFAN TOTH, EMBARC

JUG Kaiserslautern, 27. Mai 2015

Stefan Toth

stefan.toth@embarc.de

@st_toth

xing.to/sto

www.embarc.de
www.swamuster.de

embarc
Software Consulting GmbH

Agenda

1 **Einstieg**

2 Was ist Netflix?

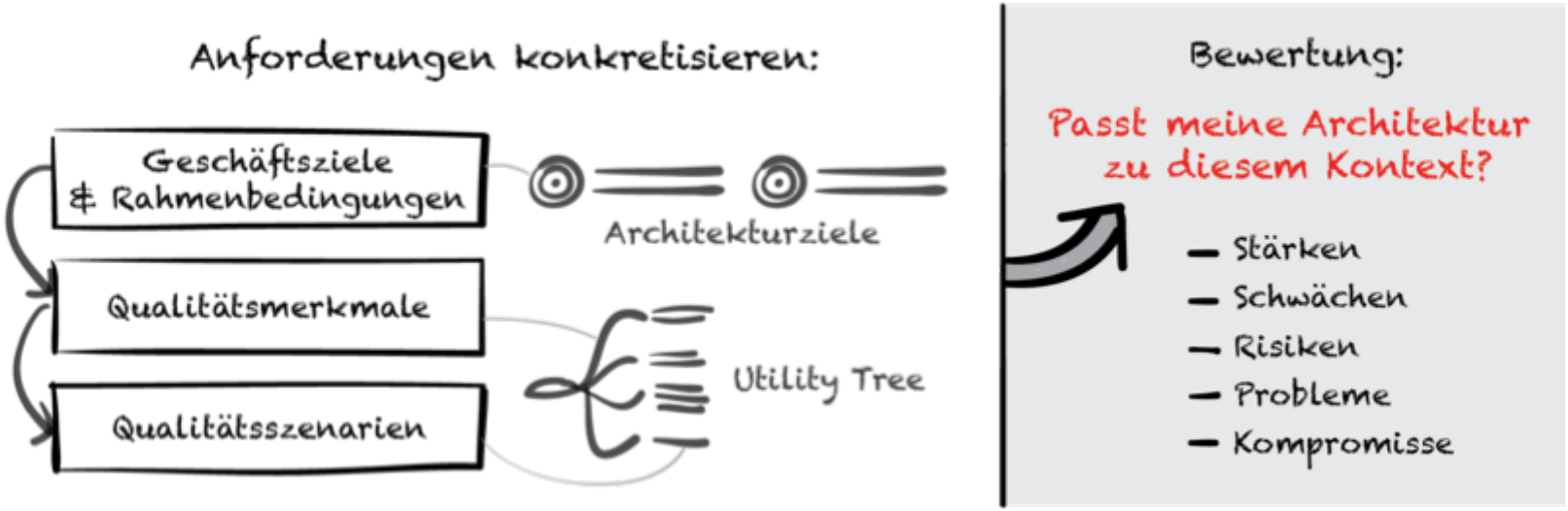
3 Lösung im Detail

4 Ergebnisse zusammengefasst

5 Weitere Informationen

1

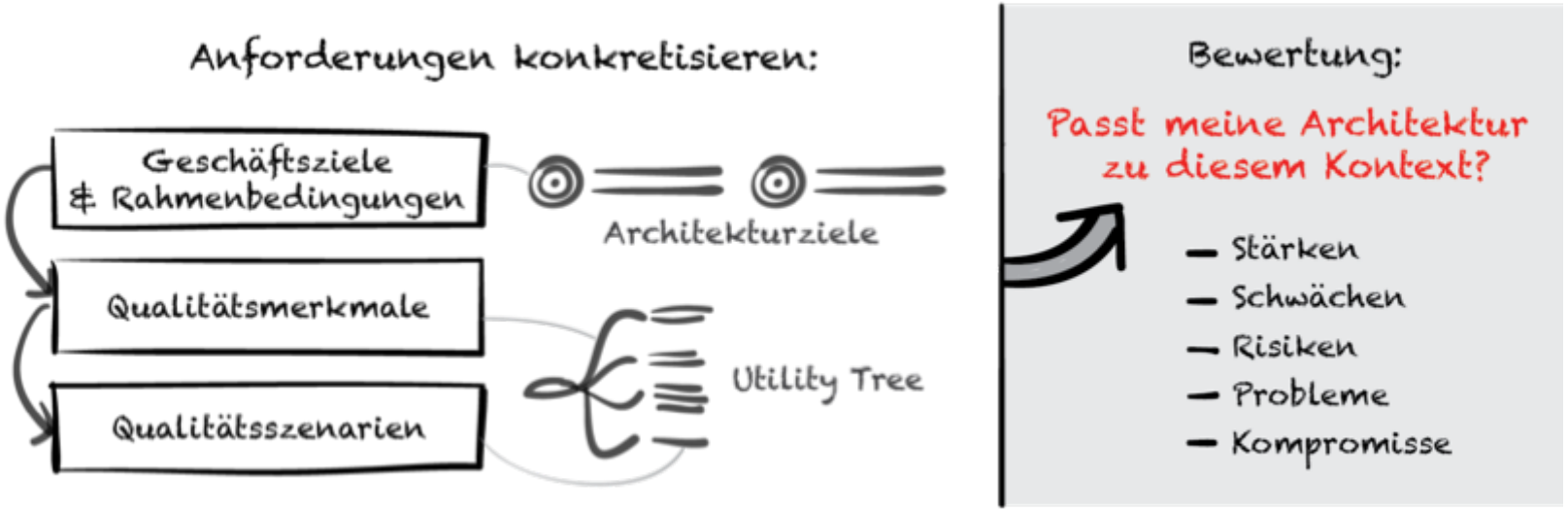
Architekturbewertung

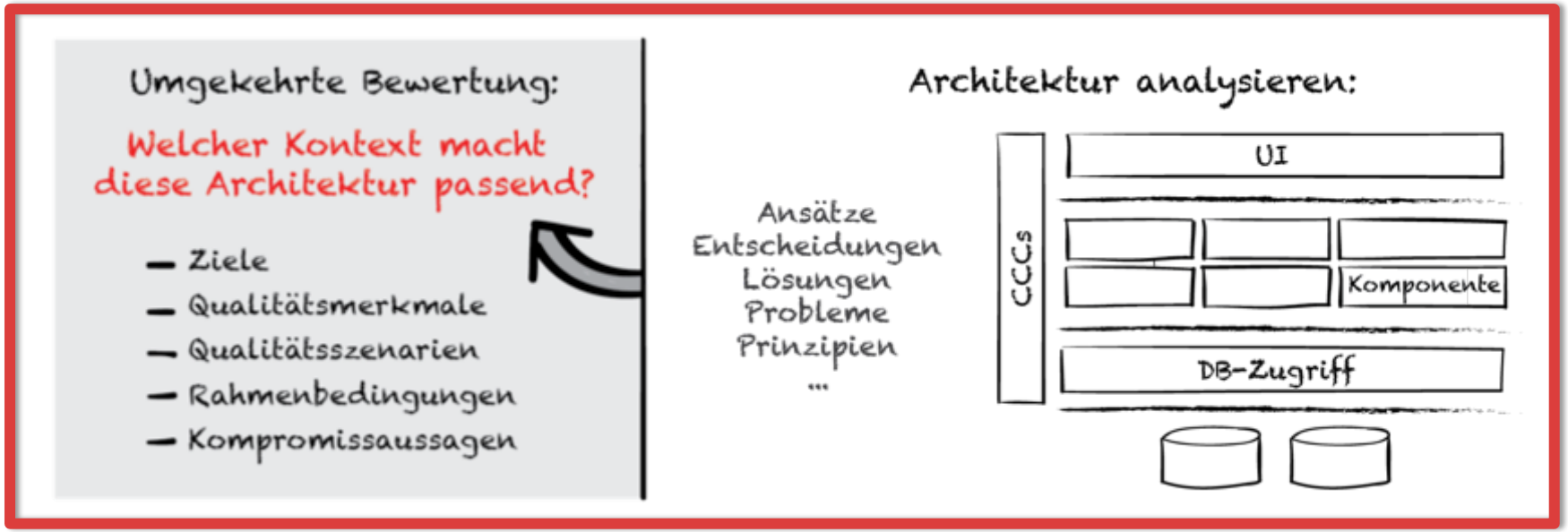


Methoden:

- **ATAM** – Architecture Tradeoff Analysis Method
- **CBAM** – Cost-Benefit Analysis Method
- **SACAM** – Software Architecture Comparison Analysis Method
- **LAAAM** – Lightweight Architecture Alternative Assessment Method
- …

Umgekehrte Bewertung?





Warum umgekehrt bewerten?

- Weil die Netflix Architekturkonzepte „trendig“ sind
  - Cloud
  - Microservices
  - Reactive Extensions
  - NoSQL Datenbanken
  - ...
- Weil sich viele fragen: Passt das auch zu uns?
- Weil ich „kommt drauf an“ zu allgemein finde ...

Agenda

1 Einstieg

2 **Was ist Netflix?**

3 Lösung im Detail

4 Ergebnisse zusammengefasst

5 Weitere Informationen

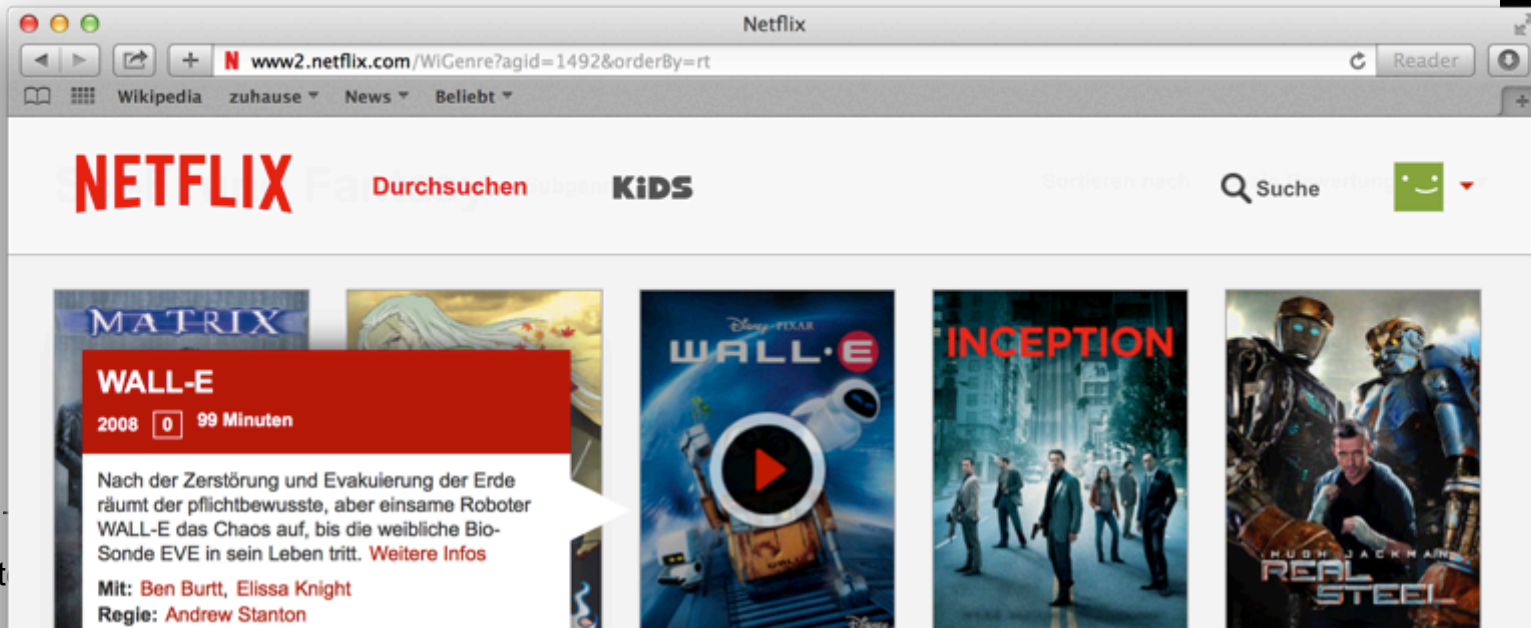
2


NETFLIX

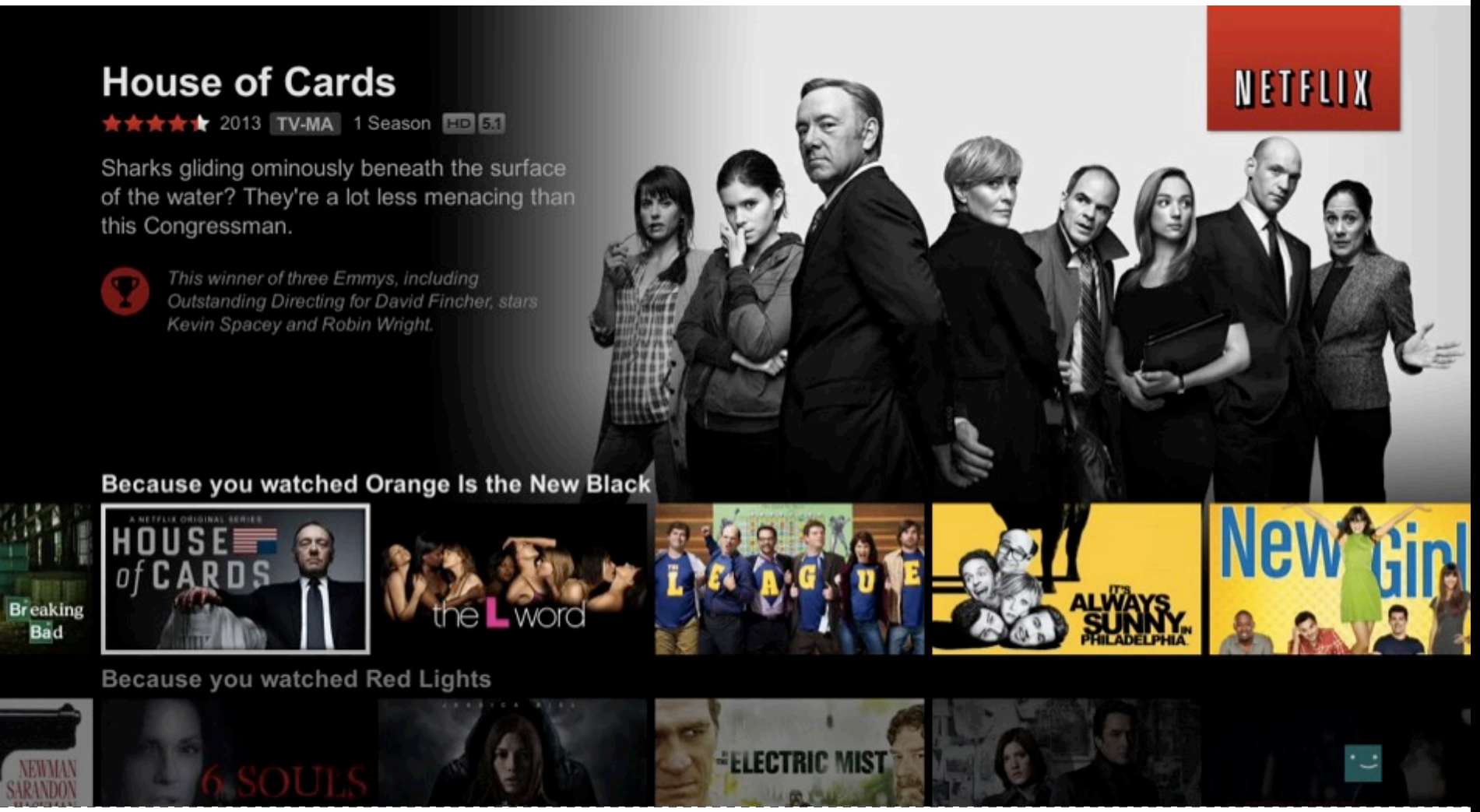
Was ist Netflix?

“Netflix is the world’s leading Internet television network with over 57 million members in nearly 50 countries enjoying more than two billion hours of TV shows and movies per month, including original series. For one low monthly price, Netflix members can watch as much as they want, anytime, anywhere, on nearly any Internet-connected screen. Members can play, pause and resume watching, all without commercials or commitments.”

➔ http://ir.netflix.com



‚Netflix is the king of online streaming, using more global bandwidth than cat videos and piracy combined.‘

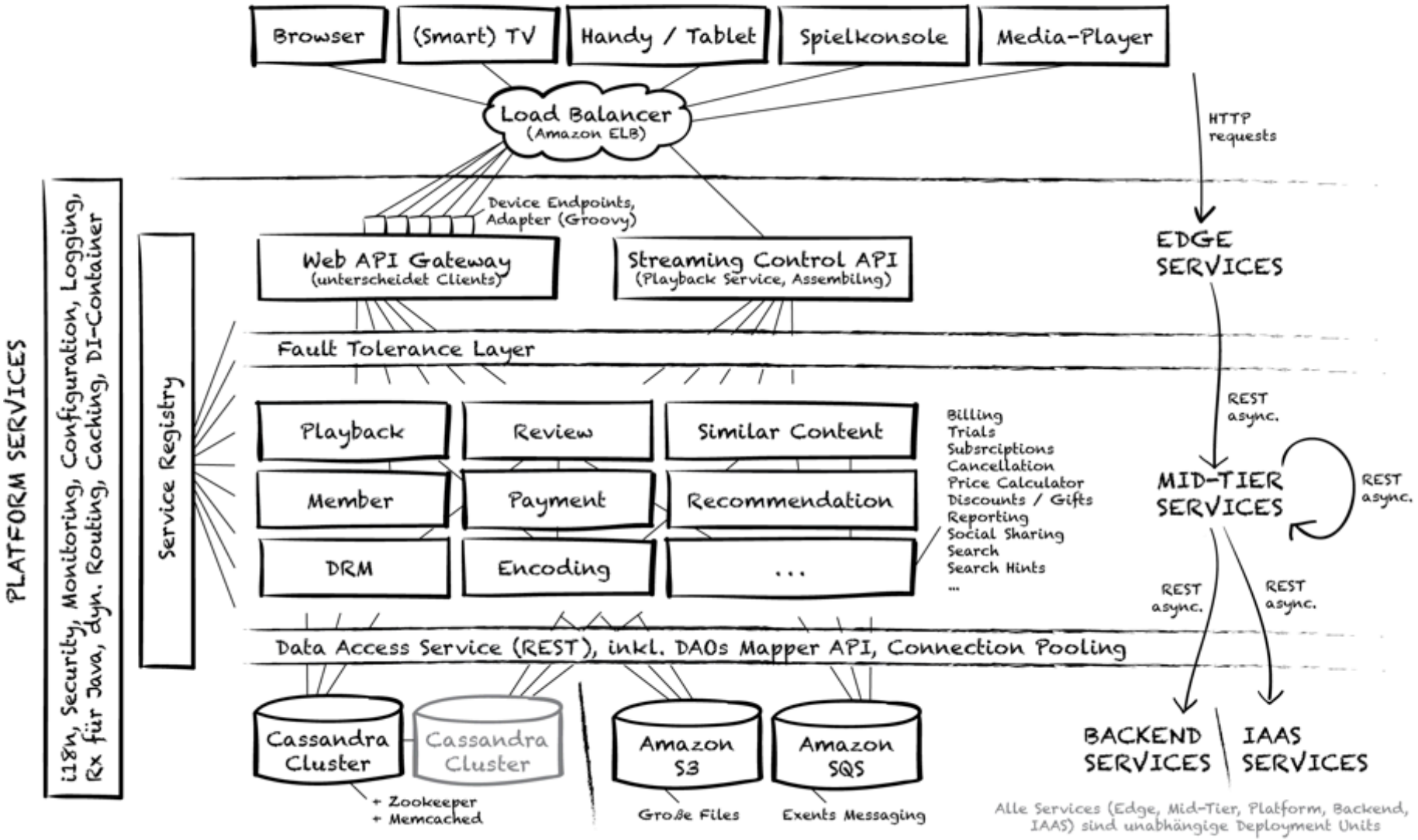


Netflix – Wie groß ist ‘groß’?

- **600+** Services und Applikationen
- **Milliarden** Requests am Tag
- **> 2 Milliarden** Stunden Filme und Serien
- **10.000e** Ec2 Instanzen in mehreren AWS Regionen/Zonen
- Cassandra NoSQL DB in einem Regionen-übergreifenden, globalen Ring mit **Terabytes** an Daten
- Zu Spitzenzeiten **1/3** der Internet-Bandbreite in den USA (Downstream)

Netflix Architekturüberblick I

Netflix Architekturüberblick II



Agenda

1 Einstieg

2 Was ist Netflix?

3 **Lösung im Detail**

4 Ergebnisse zusammengefasst

5 Weitere Informationen

3

Lösung im Detail (1)

Microservices

Was sind Microservices?

> *“In short, the microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API.”*

(James Lewis, Martin Fowler)

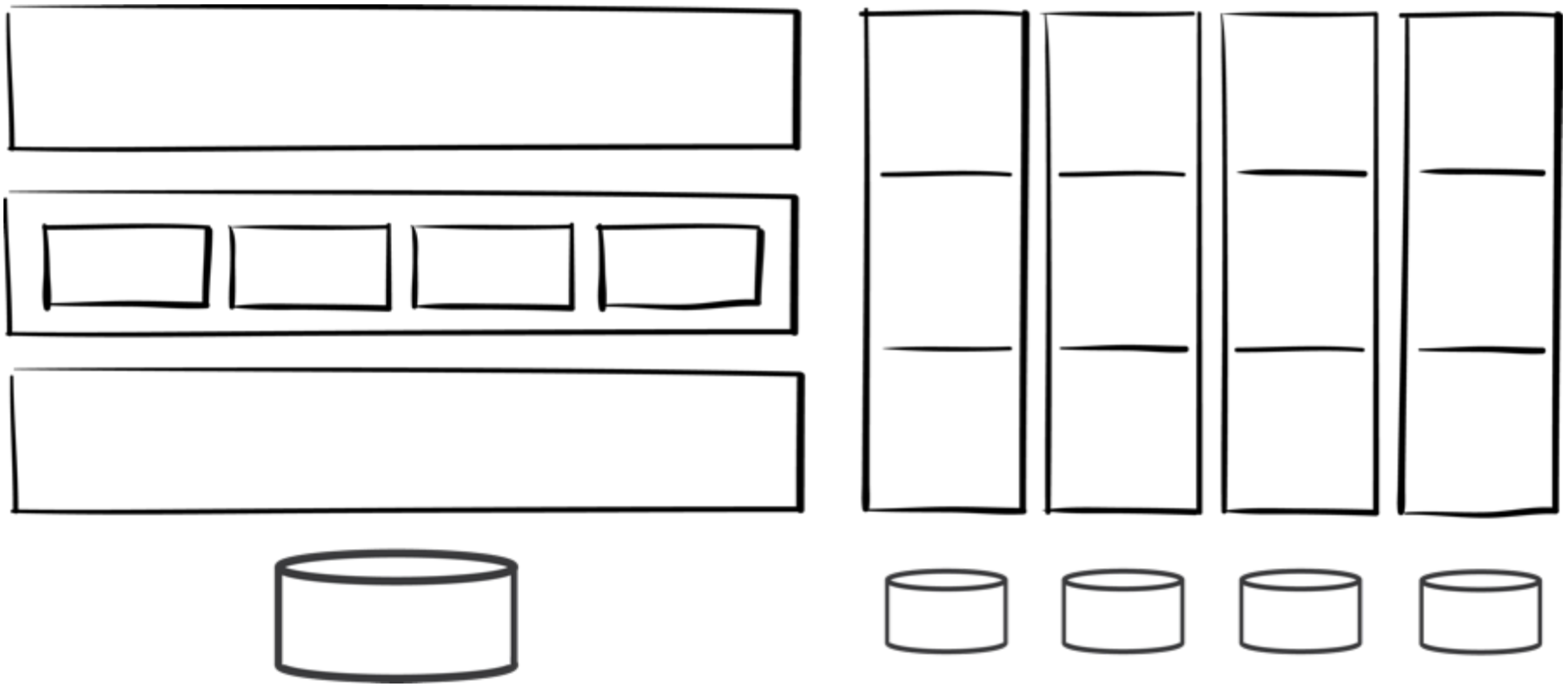
Charakteristische Eigenschaften

- Zerlegung in relativ kleine (fachliche) Services
- Services sehr lose gekoppelt
- Services einzeln installierbar und upgradebar
- Dezentrale Datenhaltung
- Hoher Freiheitsgrad bei Technologieauswahl

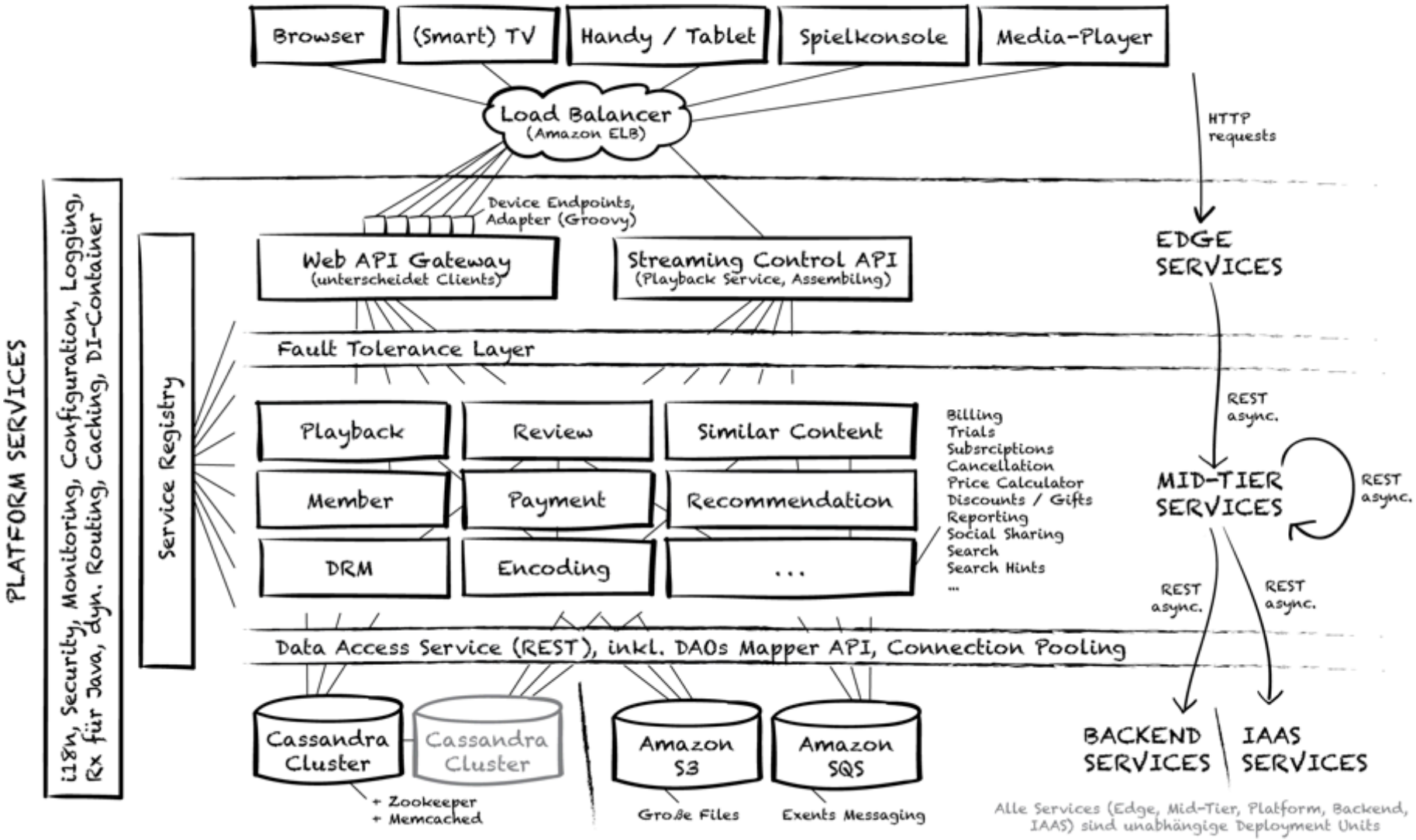
Im Prinzip…

Schichten

Vertikalen



Microservices eingeordnet …

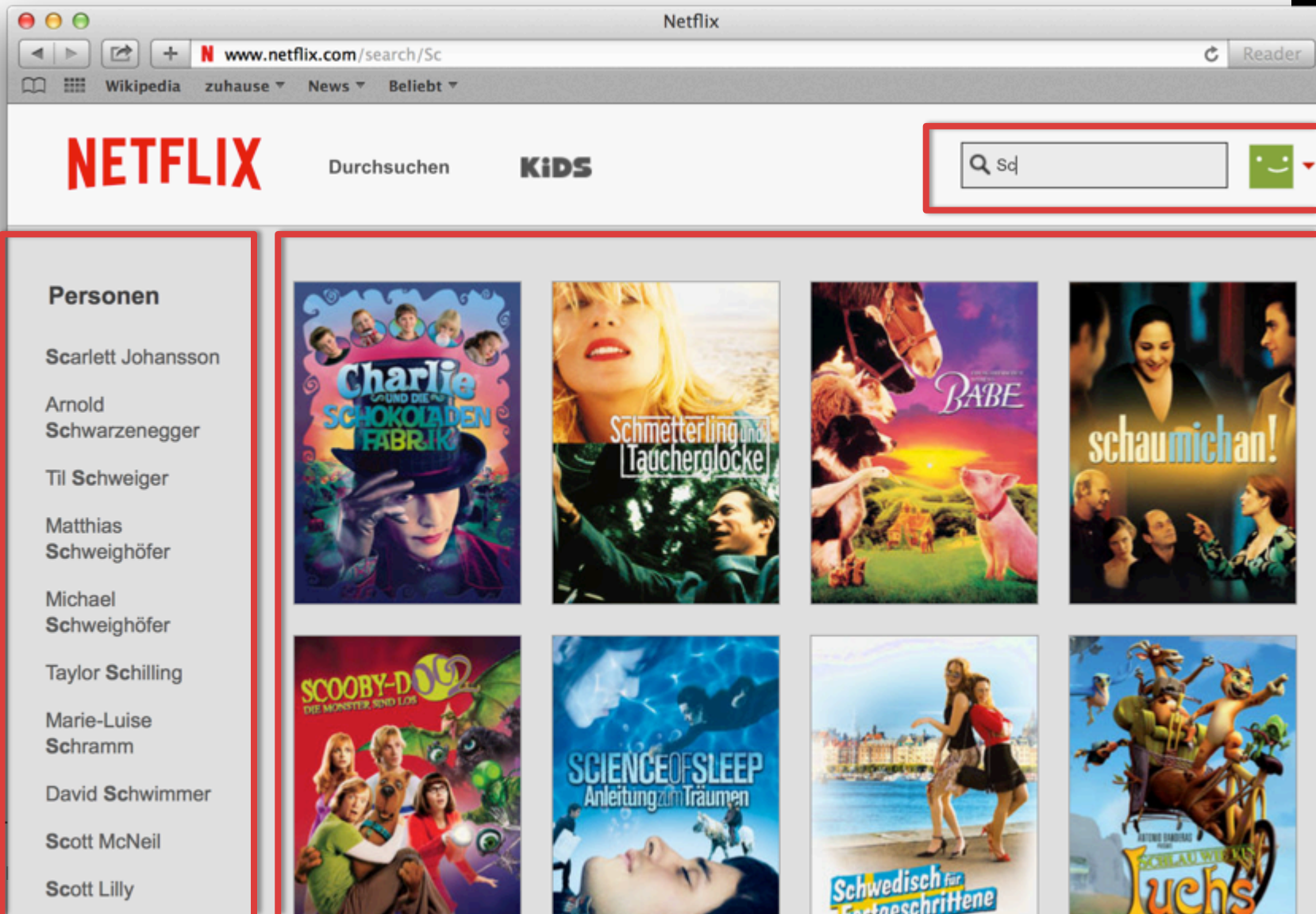


Microservices bei Netflix

Funktionalität

- Netflix-System besteht aus 600+ Services
- Beispiele für Services („Applications“)
  - Registrierung (Sign-up)
  - Suche (Search)
  - Empfehlungen (Recommendation)
  - Bewertungen (Ratings)
  - Leihhistorie (Rental History)
  - ...

Beispiel für eine Service-Trennung



Microservices bei Netflix (org.)

Teams vollumfänglich für ihre Services **verantwortlich**

- Entwicklung
- Release und Deployment
- Betrieb

Keine klassische **Management-Steuerung**

Keine Beeinflussung durch andere Teams oder eine zentrale Instanz

- Wenige technologische Vorgaben
- Release nach Belieben

"Freedom & Responsibility"

Verwendete Technologien

Plattformen

- Apache HTTP Server
- Apache Tomcat
- Bottle (Python)
- ...

Programmiersprachen

- Java
- Scala
- JavaScript
- Dart
- C++
- Groovy
- Python
- Clojure
- Ruby
- ...

Persistenz

- Cassandra
- RDBMS (MySQL)
- in-memory caches
- Amazon S3
- CDN
- ...

Konsequenzen für Netflix

- ➜ Neue Technologien sind leicht ausprobiert
- ➜ Technologie-Stack kann nach und nach modernisiert werden (keine langfristige Binding)
- ➜ Fehler in einem Service haben wenig Einfluss auf andere Services (Resilience-Grundlage)
- ➜ Gute Time-to-Market für neue Funktionalität

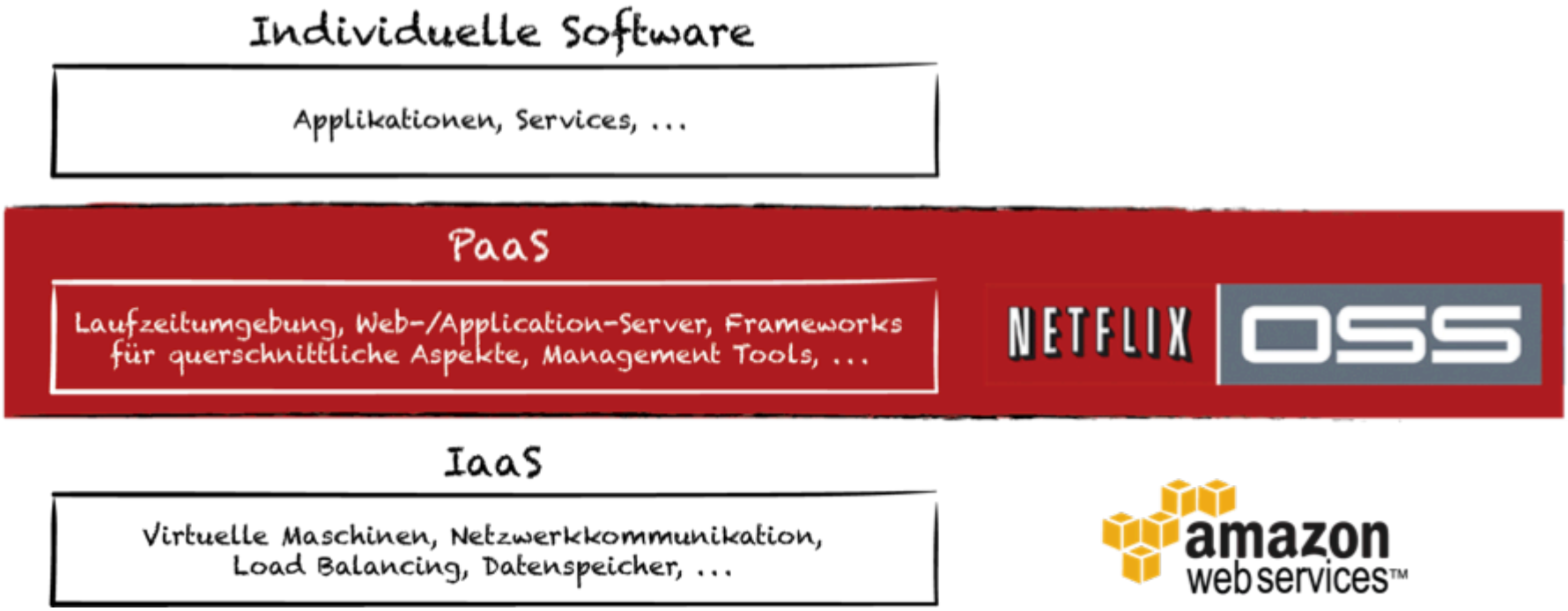
Herausforderungen

- Die Gesamtmenge der verwendeten Technologien ist sehr heterogen
- Abstimmung / Koordination schwierig
- Jedes Service-Team muss Know-How für eigentlich querschnittliche Themen haben

Lösung im Detail (2)

Netflix OSS Komponenten

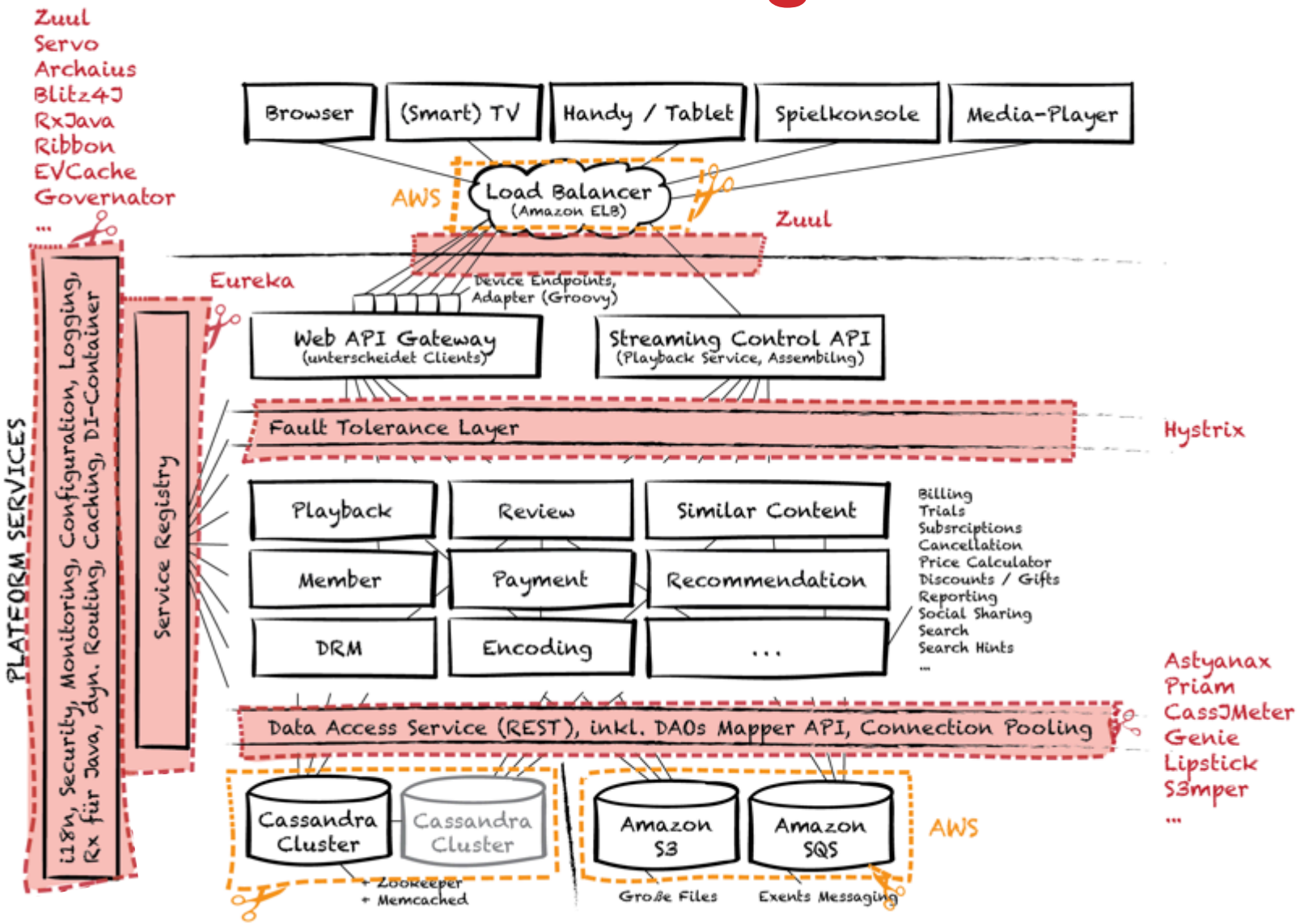
Netflix Cloud Stack



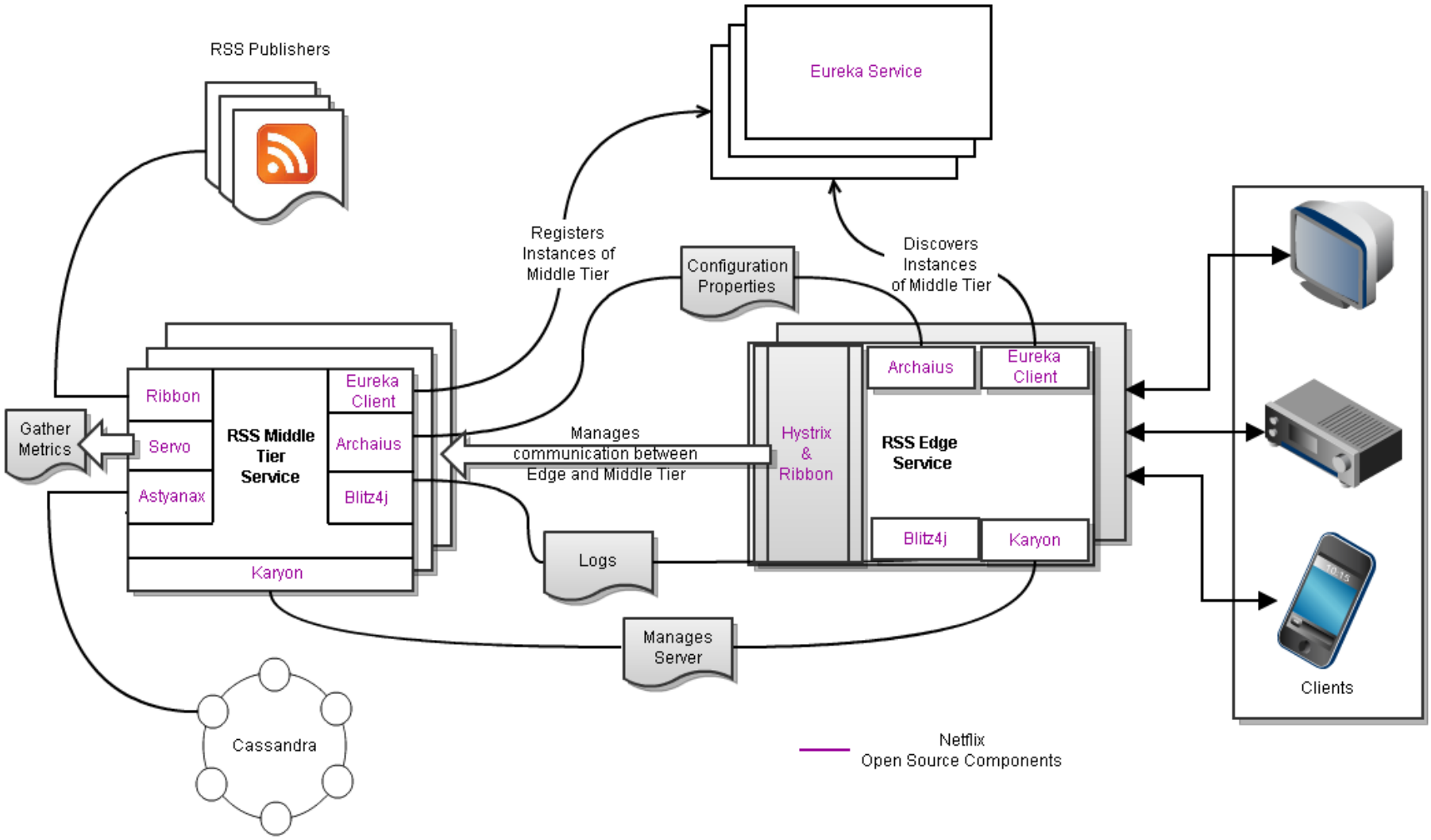
The Netflix Open Source Platform Components fill gaps in Amazon Web Services. The goal is to make cloud infrastructure more robust, flexible and glitch free.

Netflix Open Source Services

Netflix OSS eingeordnet …



Beispielanwendung (2 Services)



Konsequenzen für Netflix

- ➜ Anforderungen an Entwickler sinken
- ➜ Schnelle Time-to-Market für neue Features
- ➜ Architektur-Prinzipien werden unterstützt (Serviceaufbau, Kommunikation, Verwendung von 3rd Party etc.)
- ➜ „Open-sourcing“ von Projekten fördert:
  - Ein Mindestmaß an Dokumentation (Wiki, Techblog)
  - Sauberes Design
  - Bild als interessanter Arbeitgeber

Herausforderungen

- Netflix-Spezifika in Entwicklung recht prominent
- Neues Projekt auf dieser Basis hat Overhead

Lösung im Detail (3)

Netflix Deployment

Deployment bei Netflix

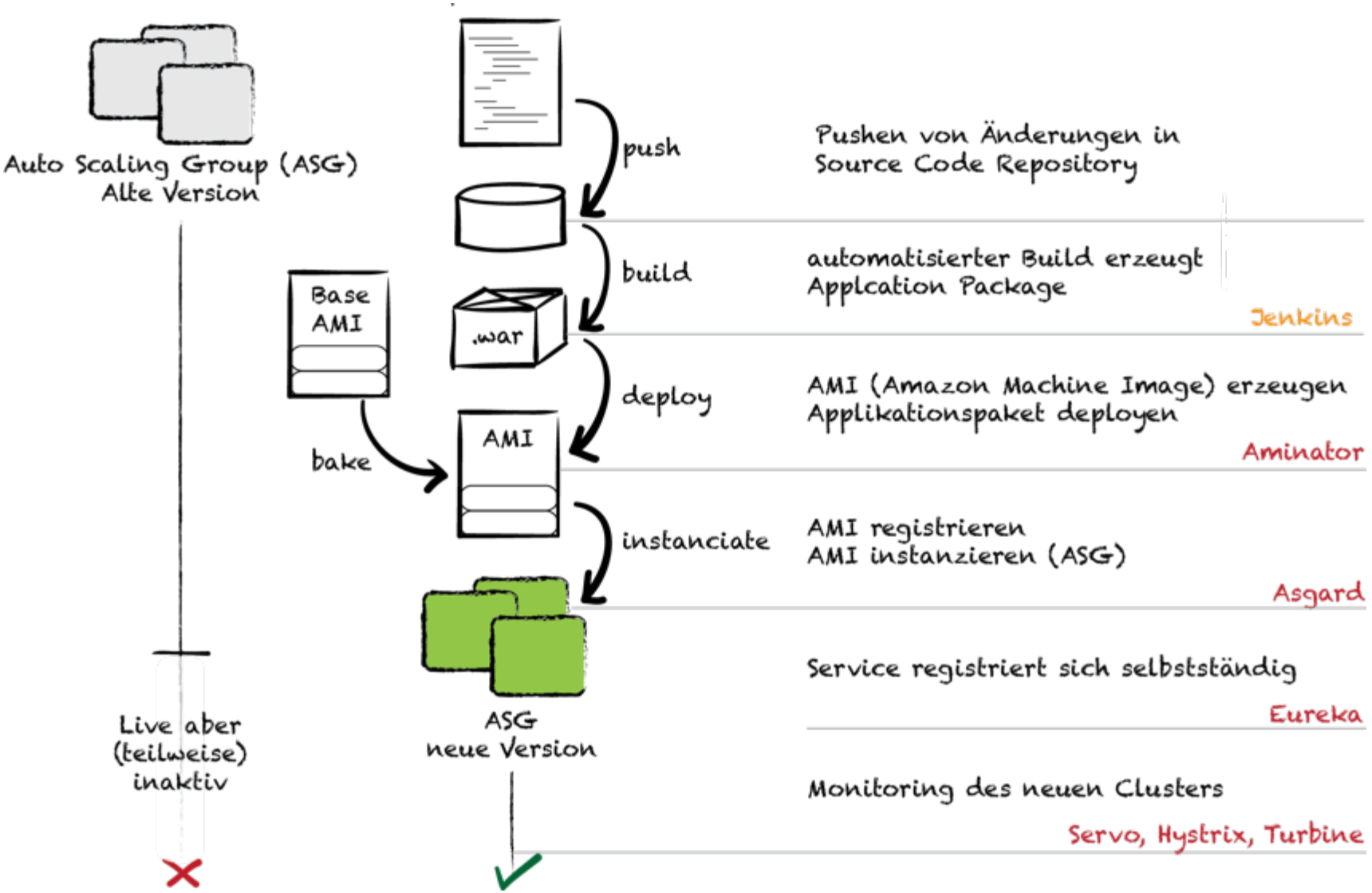
Antwort auf **Koordinationsproblem** bei Deployments?

Antwort auf **Komplexität** bei Teamabhängigkeiten?

Unterstützte Anarchie

- **ca. 100** Deployments pro Tag
- Teams arbeiten selbstverantwortlich und unabhängig
- **Keine** Qualitätssicherungsabteilung
- **Keine** Release Engineers
- **Keine** Gesamtkoordination von Releases / Deployments

Komplette Automatisierung ...



Konsequenzen für Netflix

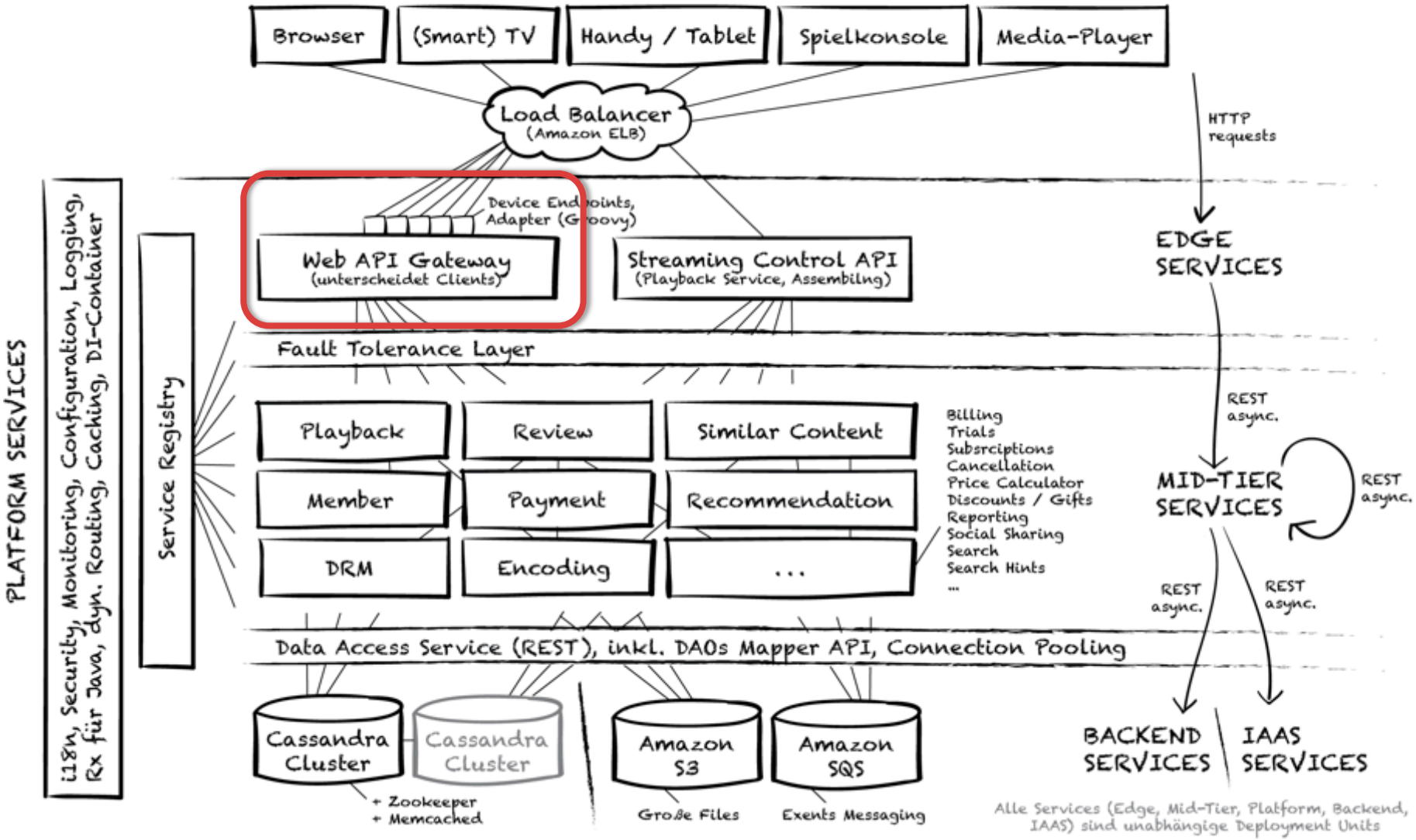
- Schnelles Rollback (bzw. Fallback)
- Billige Tests in Produktionsumgebung statt teurer Testumgebung (die schwer realistisch zu gestalten ist)
- Toolkette abstrahiert und entkoppelt Entwickler von zentralen Einstellungen und Konfigurationen

Herausforderungen

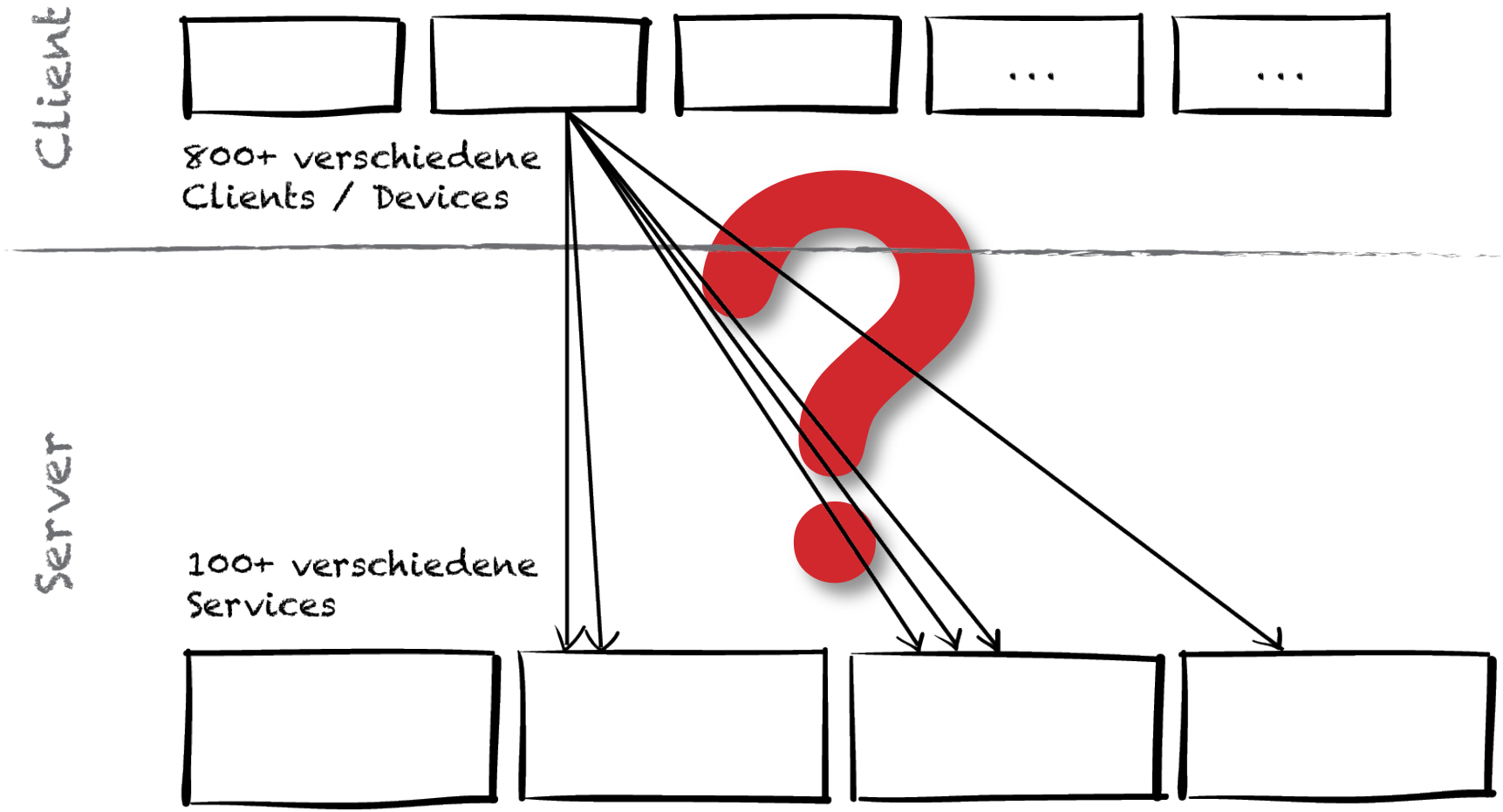
- Mehrfache Hardware erforderlich
- Koordinationsprobleme auf First-come-first-serve heruntergebrochen
- Hohe Grad an Werkzeugunterstützung und Automatisierung notwendig
- Hohe Anforderungen an Logging und Monitoring

Lösung im Detail (4)

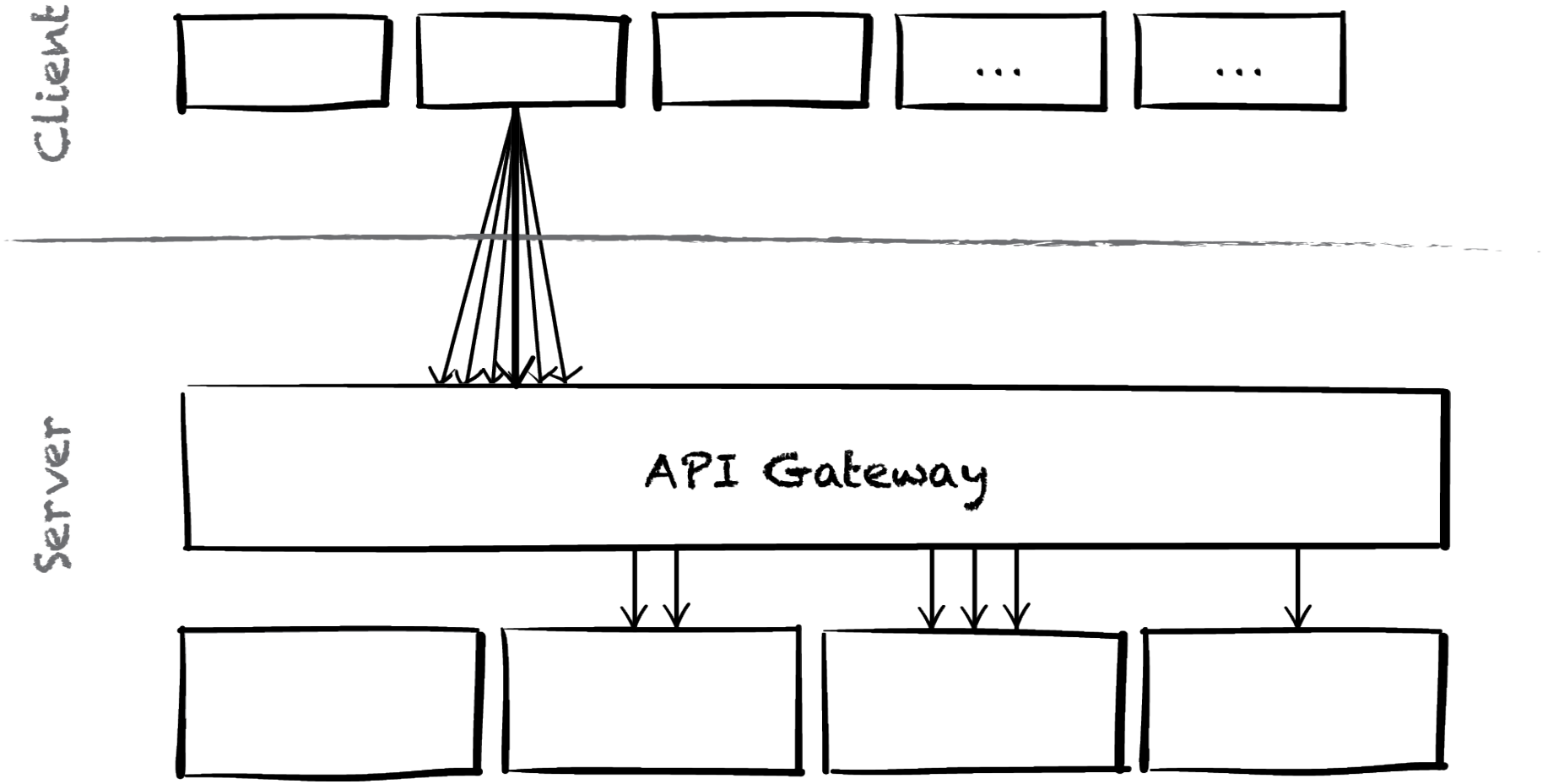
Netflix API eingeordnet …



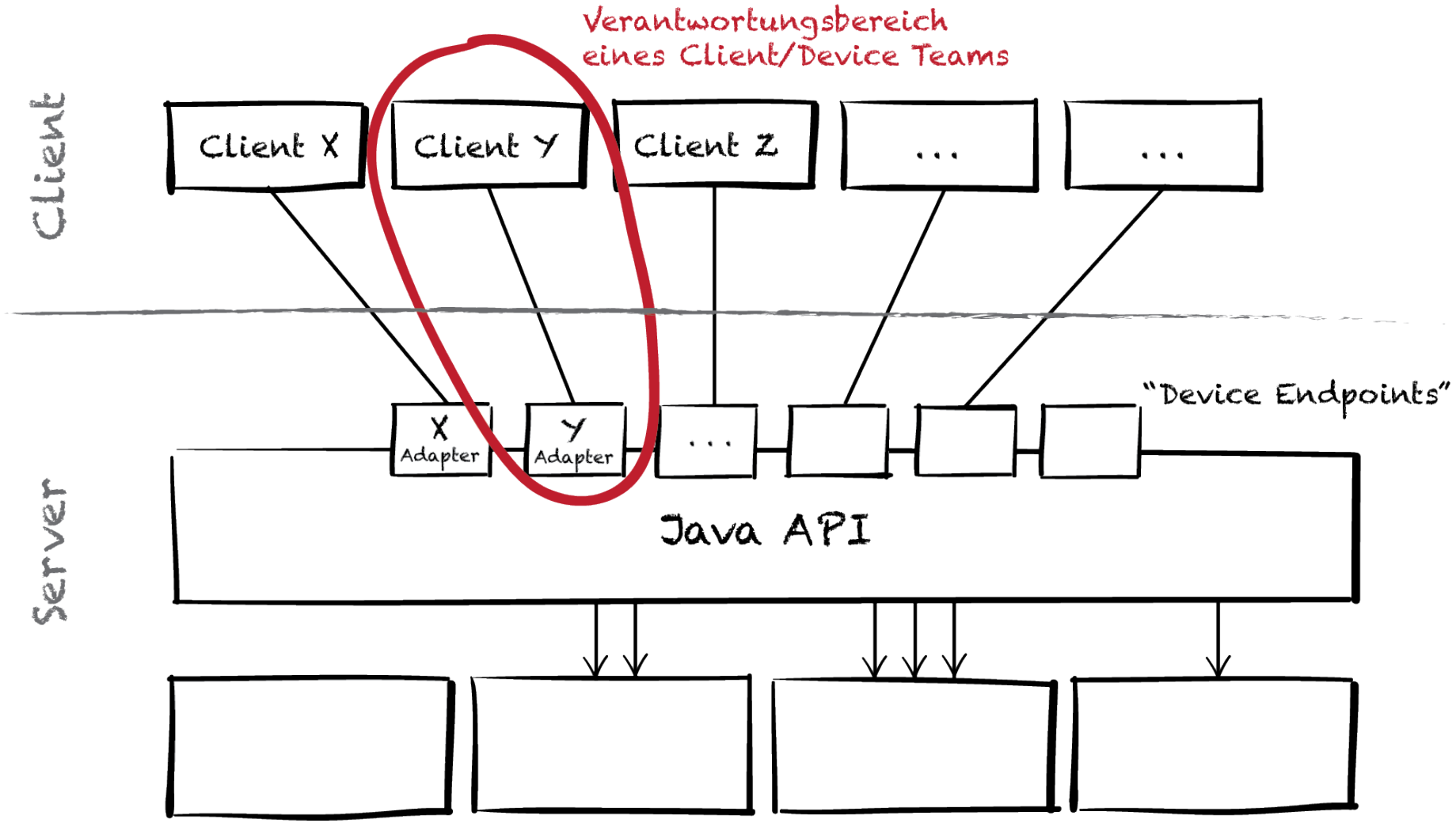
Wie nutzen / kombinieren Clients die Services?



API Gateway („Remote Facade“)



Netflix API


Verantwortungsbereich
eines Client/Device Teams
Client
Client X
Client Y
Client Z
...
...
"Device Endpoints"
X
Adapter
Y
Adapter
...
Server
Java API

Konsequenzen für Netflix

- Exzellente UI Experience / Performance pro Device
- Kurze Abstimmungswege zwischen Client- und Server-seitigen Entwicklern
- Framework unterstützt schnelle Entwicklung von Client-spezifischen APIs

Herausforderungen

- Die UI-Teams müssen auch Server-code schreiben und verantworten
- Mit vielen Geräten entstehen immer mehr Client-spezifische APIs

Agenda

1 Einstieg

2 Was ist Netflix?

3 Lösung im Detail

4 **Ergebnisse zusammengefasst**

5 Weitere Informationen

4

Qualitätsanforderungen

Zentrale Randbedingungen

Welche Rahmenbedingungen bräuchte man aus unserer Sicht für die Netflix-Lösung?

1. (Weiter-)Entwicklung eines **langlebigen Produktes**
2. Größe des Produktes rechtfertigt **mehrere Teams**
3. Teams können **selbstbestimmt** arbeiten
4. Deployment in **Cloud** machbar
5. Scheitern im Deployment möglich
6. Einsatz von **Open Source**-Lösungen OK

Das hier!

Zusammenfassende Kompromissaussagen

ist wichtiger als das hier...

Technologieentscheidungen auf Teamebene und **Experimente** zur besseren Erreichung qualitativer Ziele...

...sind einer **homogenen Systemlandschaft** mit hoher Integrität vorzuziehen.

Innovationskraft und **Wachstum** sind sehr wichtige Aspekte der Softwareentwicklung.

Kontrolle, **Planungsfähigkeit** und Status-aussagen gegenüber dem Management sind diesen Aspekten klar untergeordnet.

Die **schnelle** Erstellung und **Auslieferung** von Funktionalität...

... ist wichtiger als gut getestete Lösungen und „**Fehlerfreiheit**“ in Produktion.

Um **herausragende Qualität** und entsprechende Marktvorteile zu erzielen...

...sind redundante Entwicklungstätigkeiten und **geringere Wiederverwendungs**-möglichkeiten jederzeit in Kauf zu nehmen.

Höhere initiale **Aufwände für Framework-** und Infrastrukturkomponenten sind in Ordnung...

...um die langfristige Brauchbarkeit und **zeitgemäße Ausprägung** der Lösung zu garantieren

Agenda

1 Einstieg

2 Was ist Netflix?

3 Lösung im Detail

4 Ergebnisse zusammengefasst

5 **Weitere Informationen**

5

Netflix Tech Blog

➔ http://techblog.netflix.com

Software Engineering Radio #216

➔ http://www.se-radio.net/2014/12/episode-216-adrian-cockcroft-on-the-modern-cloud-based-platform/

se-radio.net

Software Engineering Radio
The Podcast for Professional Software Developers

Home | About SE-Radio | Be A Host | Praise | Team | Contact | Legal

Episode 216: Adrian Cockcroft on the Modern Cloud-based Platform

Filed in Episodes by SE-Radio on December 10, 2014 • 7 Comments

Adrian Cockcroft discusses the challenges in creating a dynamic, flexible, cloud-based platform with SE Radio host Stefan Tilkov. After briefly discussing the definition of "cloud computing," Adrian explains the history behind Netflix's move to the cloud (which he led). After highlighting some of the differences that have developers and architects must face, Adrian talks about lessons for other kinds of companies and industries. Microservices and microservices architecture are a hot topic

Connect

IEEE computer society

IEEE Software offers pioneering ideas, expert analyses, and thoughtful insights for software professionals who need to keep up with rapid

Artikel im Javamagazin 5.15

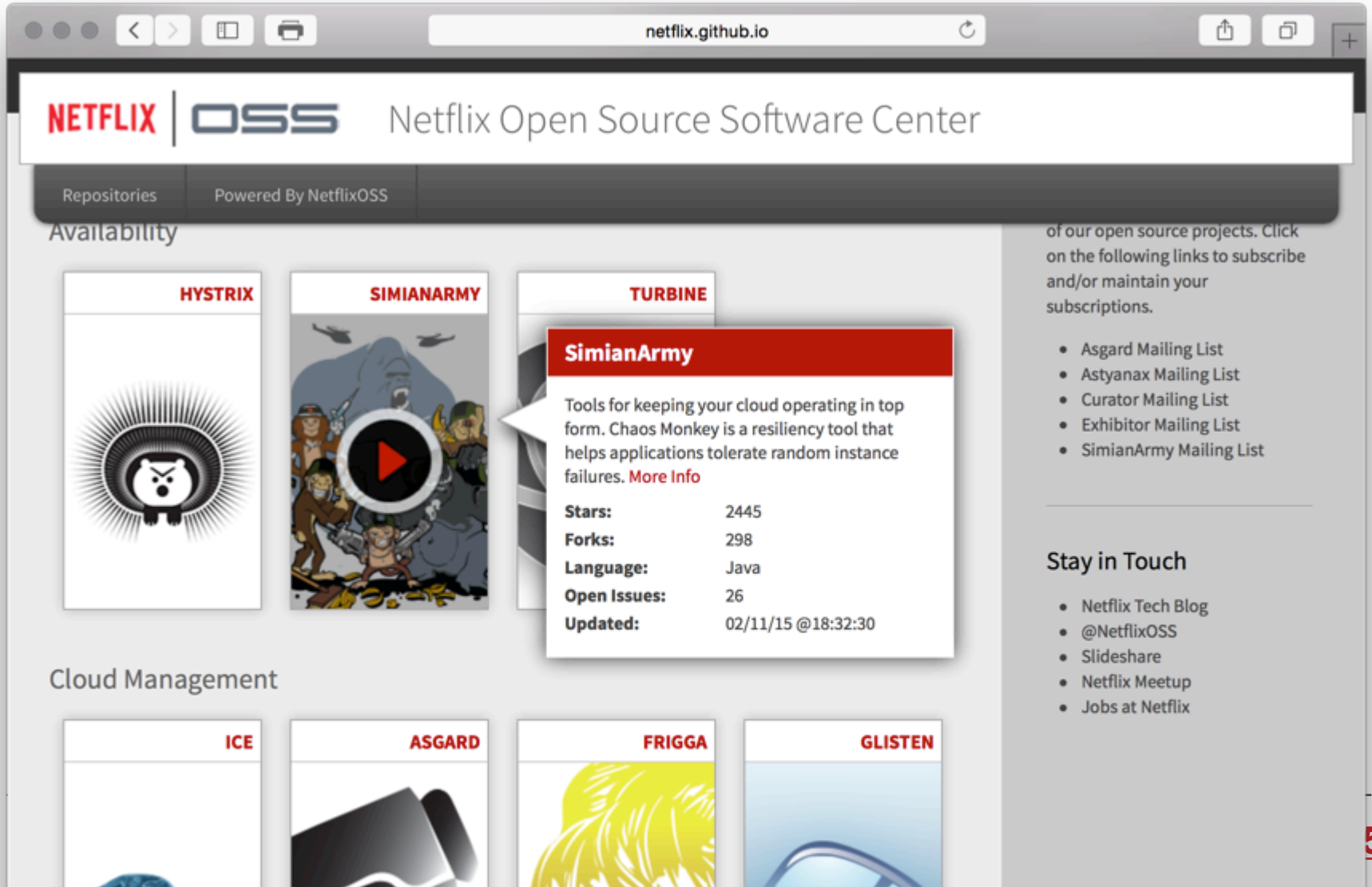
➔ House of Guts –
Keine Angst vor Chaos

Netflix: Resilience
konsequent
zu Ende gedacht

Stefan Toth
Javamagazin 5.15

Netflix Open Source Software

➔ http://netflix.github.io



Vielen Dank.

Ich freue mich auf Eure Fragen!

 stefan.toth@embarc.de

 @st_toth

 xing.to/sto

DOWNLOAD FOLIEN:
http://www.embarc.de/blog/

embarc
Software Consulting GmbH