

# SERVICE MESH

## KONSEQUENZEN BEI MICROSERVICE-ARCHITEKTUREN.

Michael Hofmann im Interview mit Ulrich Parthier über die Vorteile, den Nutzen und die Kosten für IT-Abteilungen in Unternehmen.

**Ulrich Parthier:** Der Begriff der Microservice-Architektur ist in den letzten Mo-

Bild 2: Echtzeitgraph von Microservice-Abhängigkeiten bei Amazon.

Quelle: <https://twitter.com/Werner/status/741673514567143424>, Werner Vogels, CTO Amazon

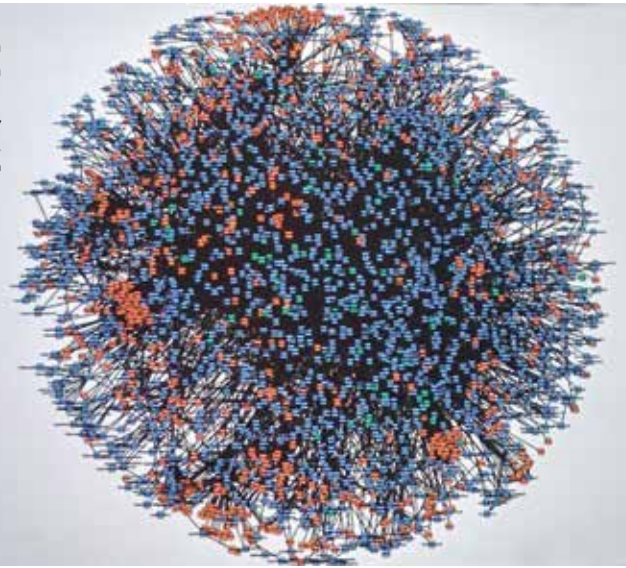


Bild 1: Die Death Star-Architektur bei Netflix.  
Quelle: Adrian Cockcroft (Netflix) / Martin Fowler

erste Definition bezeichnet einen Service Mesh als das Zusammenspiel von vielen kleineren Anwendungen, welche in der Regel als Microservices implementiert werden. Durch die Interaktion der Microservices kann dabei ein Kommunikations-Geflecht entstehen, das nur schwer ohne Werkzeuge zu betreuen und zu verwalten ist. Hier kommt jetzt die zweite Definition ins Spiel. In dieser erweiterten Form wird auch das Werkzeug zur Verwaltung eines Services Meshs mit in die Definition integriert.

naten verstärkt in den Fokus von IT-Abteilungen geraten. Wie kann diesen Begriff am besten beschreiben?

**Michael Hofmann:** Der aktuelle Architektur-Trend geht weg von monolithischen Systemen hin zu kleineren Anwendungen (Microservices), die in sich fachlich abgeschlossen sind und über definierte Schnittstellen mit anderen Anwendungen kommunizieren. Diese fachliche Abgeschlossenheit geht sogar so weit, dass diese Anwendungen ihre eigene Persistierung besitzen und direkte Zugriffe anderer Services auf diese Datenbanken verboten sind. Somit werden Anwendungslandschaften geschaffen, die

unabhängig voneinander entwickelt und released werden können, was insgesamt zu mehr Flexibilität führt.

Änderungen an den kleineren Anwendungen können schneller erfolgen, wodurch rascher auf neue Anforderungen reagiert werden kann.

**Ulrich Parthier:** Neuerdings taucht im Zusammenhang mit Microservice-Architekturen der Begriff „Service Mesh“ auf. Was verbirgt sich dahinter?

**Michael Hofmann:** Derzeit existieren zwei Definitionen des Begriffs Service Mesh. Die

**Ulrich Parthier:** Gibt es Tools, die dabei helfen, die Übersicht zu behalten?

**Michael Hofmann:** Aktuell führend sind derzeit zwei Werkzeuge, die zur Verwaltung des Service Mesh entwickelt werden: Istio und Linkerd.

**Ulrich Parthier:** Können Sie ein Beispiel für eine Microservice-Anwendung und Service Meshs nennen?

**Michael Hofmann:** Die bekanntesten Beispiele hierfür sind Anwendungen bei Netflix (Bild 1) oder Amazon (Bild 2). In beiden Fäl-



”

FRÜHER ODER SPÄTER WERDEN SIE SICH MIT DEM THEMA SERVICE MESH IN IHREM MICROSERVICE-PROJEKT AUSEINANDER SETZEN MÜSSEN.

Michael Hofmann, Software Architekt | [www.hofmann-itconsulting.de](http://www.hofmann-itconsulting.de)

len interagieren mehr als hundert Services miteinander und bilden somit eine komplexe Fachlichkeit ab. Abhängigkeitsgraphen, welche zur Laufzeit erstellt wurden, zeigen eindrucksvoll wie komplex diese Kommunikations-Beziehungen sein können.

**Ulrich Parthier:** Was wird gerne bei der Betrachtung der Services vernachlässigt oder gar vergessen?

**Michael Hofmann:** Durch die Möglichkeit einzelne Services voneinander unabhängig zu releasen existieren in Produktionsumgebungen neue Versionen von Services oft neben älteren Versionen des selben Services. Der Grund für diese Koexistenz liegt darin, dass aufrufende Services noch abhängig sind von der Vorgänger-Version des aufgerufenen Services. Parallel dazu befinden sich aber schon Services in Produktion, die von den neuen Funktionalitäten des Partner-Services profitieren. Damit entsteht ein Mix an alten und neuen Versionen von Services, die gleichzeitig verwendet werden. Multipliziert man die Anzahl an unterschiedlichen Versionen mit der Anzahl an Services kommt man schnell in einen zweistelligen oder sogar dreistelligen Bereich an Services die verwaltet werden müssen.

**Ulrich Parthier:** Was kann beim Beherrschen eines Service Mesh helfen?

**Michael Hofmann:** Notwendige Funktionalitäten um einen Service Mesh in den Griff zu bekommen sind beispielsweise Traffic-Management, Security bei der Service-Kommunikation, Policies wie zum Beispiel Rate Limiting und Sammlung von Telemetrie-Daten zur Überwachung des Service Mesh. Dabei verbirgt sich hinter dem Begriff

Traffic-Management der umfangreichste Teil der Funktionalität. Neben einer parametrisierten Festlegung von Request-Routings (Service A ruft bei gewissen Vorbedingungen Service B in unterschiedlichen Versionen auf) sollten auch Möglichkeiten zum Testen von neuen Services existieren.

Ein Service Mesh Werkzeug sollte so etwas wie Blue-Green-Deployment oder Canary-Releasing bieten, um den Übergang zwischen den Versionen eines Services möglichst reibungslos zu gestalten. Da es bei verteilten Systemen naturgemäß auch zu Fehlern in der Kommunikation kommen kann, bieten Service Mesh Tools Funktionalitäten zur Erhöhung der Resilienz an. So kann man beispielsweise bei Istio neben Request-Timeouts auch verschiedene Circuit-Breaker definieren. Das ganze erfolgt ohne Eingriff in den Source-Code des Services, was wiederum einen großen Vorteil darstellt. Oft werden die notwendigen Resilienz-Patterns erst im produktiven Umfeld erkannt. Sollen diese Patterns nun im Source-Code umgesetzt werden, so kann dies nicht ohne ein erneutes Deployment erfolgen.

**Ulrich Parthier:** Wodurch unterscheiden sich Service-Mesh-Werkzeuge wie etwa Istio oder Linkerd. Gibt es ein herausragendes Tool oder so etwas wie eine Stärken/Schwächen-Matrix?

**Michael Hofmann:** Beide Tools Istio und Linkerd unterscheiden sich kaum in deren Funktionalität. Von daher ist es eher eine Frage der persönlichen Präferenzen für welches Werkzeug man sich entscheidet. Einen detaillierten Vergleich bietet folgender Beitrag: <https://abhishek-tiwari.com/a-sidecar-for-your-service-mesh/>

**Ulrich Parthier:** Wo es Vorteile gibt, lauern meist auch Risiken. Hier auch?

**Michael Hofmann:** Ein Service Mesh besitzt an sich schon eine gewisse Komplexität und auch das Tool zur Verwaltung des Service Mesh muss erstmal beherrscht werden. Von daher geht die Empfehlung ganz klar in die Richtung sich möglichst frühzeitig mit dem Service Mesh Tool zu beschäftigen. Auch wenn aktuell nur wenige Services zu verwalten sind sollte man dafür ein Tool einsetzen, da davon auszugehen ist, dass die Anzahl der Services sehr schnell steigen wird und sei es nur über die unterschiedlichen Versionen eines Services.

In der Regel findet die Service zu Service Kommunikation abgesichert statt. Eine der Mindestvoraussetzungen ist TLS aber auch mutual TLS kann notwendig sein. Der zugehörige Austausch von SSL-Zertifikaten kann dabei schnell eine umfangreichere Aufgabe werden. Hierbei kann das Service Mesh Werkzeug helfen indem es diese Aufgabe selbständig übernimmt. Aus Sicherheitsgründen ist eine automatisierte Verteilung der Zertifikate gegenüber einer manuellen Administration vorzuziehen. Wer schon mal abgelaufene Zertifikate von Hand in Produktion tauschen musste will so etwas nicht nochmal erleben.

**Ulrich Parthier:** Herr Hofmann, wir danken für das Gespräch.

”  
THANK  
YOU