

Cloud Native Initiative

Erfahrungen und aktuelle Schwerpunkte
aus Sicht der
Banken-Software-Entwicklung



Raiffeisen Software GmbH

Unsere Skills gehen über die Grenzen von Österreich hinaus.



„Mit mehr als 50 Jahren Erfahrung im Finanzdienstleistungsbereich sind wir zentraler Bestandteil der Raiffeisen Bankengruppe.“

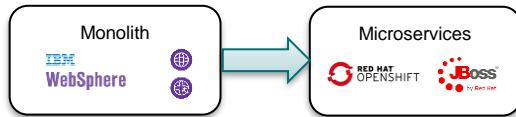
Dr. Hermann Sikora
Vorsitzender der Geschäftsführung

5 Standorte	> 850 Mitarbeiter	> 70 Agile Teams
> 50 J Erfahrung	122 Mio Umsatz 2020	> 850 Applikationen

Vom Monolith zu Cloud Native Microservices

Erfahrungen und aktuelle Schwerpunkte

2016 Flexibilität und Kosteneffizienz als Treiber für die Einführung von OpenShift und Microservices.



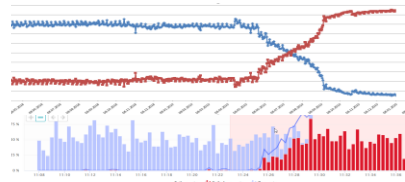
2021 Cloud Nativ Initiative mit Schwerpunkten auf Cloud Native Microservices und Einführung Service Mesh.



Primäre Zielsetzungen:

- Reduzierung des Ressourcenverbrauchs
- Unterstützung flexibles Auto-Scaling bei Lastspitzen
- Korrekter fachlicher und technischer Service-Schnitt
- Standardisierung Service-Discovery und Entwicklungs-Frameworks

2019 Komplexität und sprunghaft steigende Nutzerzahlen stellten uns vor große Herausforderungen.



Quelle: [TMDb](https://www.tmdb.org/)

Umfangreiche Service-Landschaft erfordert sensibles Migrationsvorgehen.



Produktionsumgebung Online Banking

2020 Cloud Native Lab mit einer kritischen Architekturprüfung auf Basis repräsentativer Cloud Native PoCs.

Infrastruktur

Toolstack



Polyglotte Microservices

Motivation und Zielsetzungen



Polyglotte Software-Lösungen sind durch **unterschiedliche Technologien** und **Programmiersprachen** gekennzeichnet.

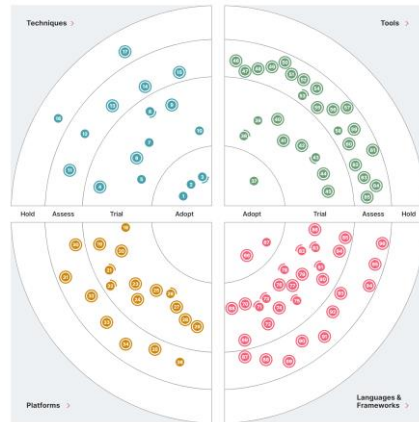
Container-Plattformen und **Microservices** unterstützen polyglotte Software-Lösungen.



Neal Ford 2006

„It's all about **choosing** the **right tool** for the job and **leveraging it correctly**“

Quelle: http://nealford.com/memeagora/2006/12/05/Polyglot_Programming.html



Quelle:
Thoughtworks Technology Radar,
<https://www.thoughtworks.com/radar>

Kontroverielle Diskussion bzgl. der sinnvollen/notwendigen Technologieviefalt.

"When developers can create resources like virtual machines and containers for no cost (either monetary or time), a governance model that values a single solution becomes **innappropriate** governance. A better approach appears in many microservices environments. One common characteristic of microservices architectures is the embrace of polyglot environments, where each service team can choose a suitable technology stack to implement their service rather than try to homogenize on a corporate standard."

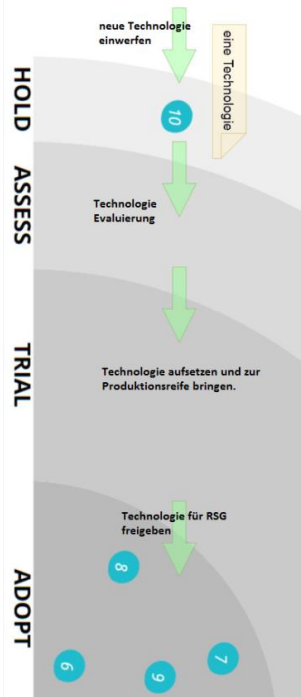
Quelle: Building Evolutionary Architectures, Neal Ford, Rebecca, Parsons, Patrick Kua, Auszug aus Forbes Magazine, Nov 30, 2011

"Unlike Zalando, other established tech companies (Google, Spotify, Tencent, [Foursquare](#), and other [CNCF End User companies](#)) use a much stricter technology selection process, limit programming language choices, and invest into changing the way applications are built to leverage centralized control planes, which increases development velocity. They limit the tech stack choices due to the amount of investment into infrastructure support and the high cost of removing technologies that did not prove to be useful."

Quelle: <https://engineering.zalando.com/posts/2020/07/technology-choices-at-zalando-tech-radar-update.html>

Polyglotte Microservices

Technologie-Management und Cloud Native Kriterien



RSG verfolgt aktives **Technologie-Management** zur **prozesshaften Einführung** neuer bzw. zur **Ablöse** bestehender Technologien.

Regulatorische Rahmenbedingungen fordern stetige **Bewertung** und **Zertifizierung** der eingesetzten Technologien.

Reduzierung der Technologievietfalt als klare Zielsetzung.

253 Technologien im Fokus

239 Technologien im Status ADOPT

24 Entwicklungssprachen

5 Generationen des RSG Java-Frameworks

1 Entwicklungs-Stack für die PaaS

RSG **Technologie-Management** ist für eine neue **Cloud Native** Microservices **Technologien nicht ausreichend**. Es braucht **weitere Kriterien** in Abstimmung mit den Betreibern.



Pipelines und Images

- Einbindung in Self-Service-Portal und Build-Pipelines
- Auswahl des Base-Images
- Festlegung der Wartungsverantwortung



Querschnittliche Aspekte

- Sicherstellung der Security-Anforderungen
- Logging im JSON-Format für zentrales Log-Management
- Bereitstellung von REST-APIs
- Integration in die zentrale Monitoring-Infrastruktur



Betriebliche Aspekte

- Abfrage des Betriebsstatus (Readiness, Liveness)
- Abschätzung des Ressourcenverbrauchs (CPU, Mem)
- Graceful Shutdown in definiertem Zeitfenster

Polyglotte Microservices

Cloud Native Java Runtimes

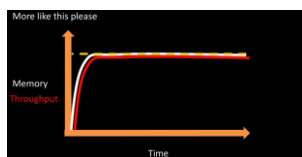
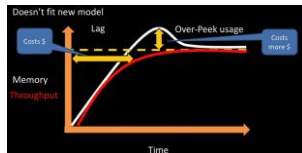


Nicht optimaler Service-Schnitt („**Microliths**“) und **träges Startup-Verhalten** verhindern dynamische Skalierung.

Das von der RSG definierte **EAP-Base-Image** ist **schwer zu warten** und unterstützt **kaum Modularisierung**.

Neue Trends im Java-Ökosystem wurden im Cloud Native Lab **verifiziert**.

Quelle: [TMDB](https://www.imdb.com/title/tt0115105/)



Quelle:
Devoxx Belgien 2019 (Steve Poole, IBM)
<https://devoxx.be/talk/?id=52055>

Cloud Native Java Runtimes und Umsetzung von **Cloud Native Microservices** als wesentlicher **Erfolgsfaktor**.

Entscheidung für **duale Strategie** zur Sicherstellung einer **sanften Migration** der bestehenden Applikationen.



Open Liberty mit Java 8

- Jakarta EE und Microprofile
- vollständiger Umfang des RSG Java-Frameworks
- bessere Modularisierung des Base-Images
- Sanfte Migration (Build-Pipeline, server.xml)
- spürbare Reduzierung des Ressourcen-Verbrauchs



Quarkus mit Java 11

- Microprofile und Best of Breed Libraries
- Ausschnitte des RSG Java-Frameworks
- Aktuelles Redhat Java S2I als Basis
- Modularisierung vollständig durch Entwickler gesteuert
- primär für Neuentwicklung und „Quarkus-Enthusiasten“
- drastische Reduzierung des Ressourcen-Verbrauchs

Polyglotte Microservices

Quarkus – Spersonic Subatomic Java






A Kubernetes Native Java stack tailored for OpenJDK HotSpot and GraalVM, crafted from the best of breed Java libraries and standards.

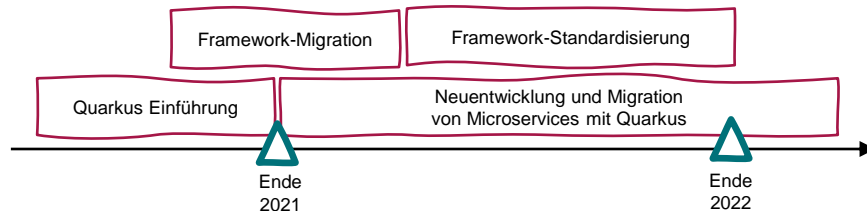
Quelle: <https://quarkus.io/>

- **Microprofile Standards** und Integrationen via Extensions
- Cloud First Runtime - **Startupzeit, Ressourcenbedarf**
- **Entwicklerproduktivität** und Dokumentation
- Aktive **Community** und **Enterprise Support**

Rahmenbedingungen für **Technologie-** und **PaaS-Einführung** konnten **rasch erfüllt** werden. Die **Migration** des **RSG Java-Framework** zur Sicherstellung querschnittlicher Aspekte ist **schwieriger als erwartet**.

-  Pipelines und Images
-  Betriebliche Aspekte

-  **Querschnittliche Aspekte**
 - Viele Aspekte werden durch RSG Java-Framework abgedeckt
 - Migration von Java 8 auf Java 11
 - Unterschiede in der Jakarta EE Implementierung (OLP, Quarkus)
 - Koexistenz zwischen Standards und Eigenentwicklungen



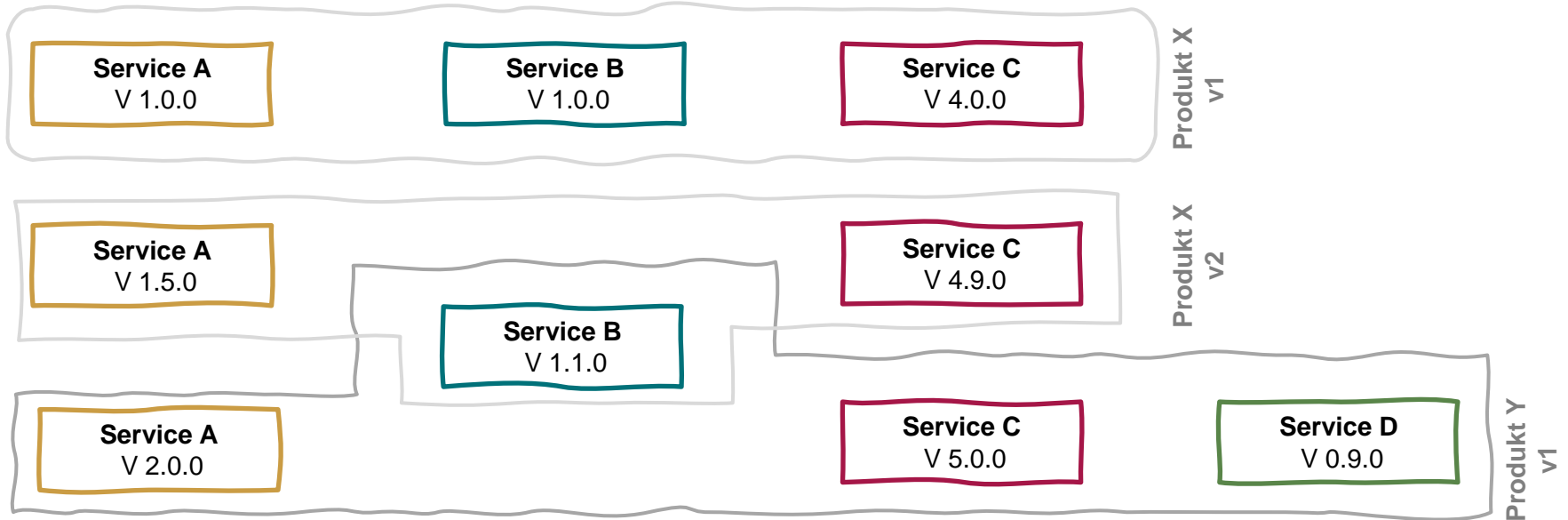
Unsere Entwickler*innen lieben Quarkus ...



... und mich 😎

Service Mesh

Anforderungen an Service Discovery and Routing



Service Mesh

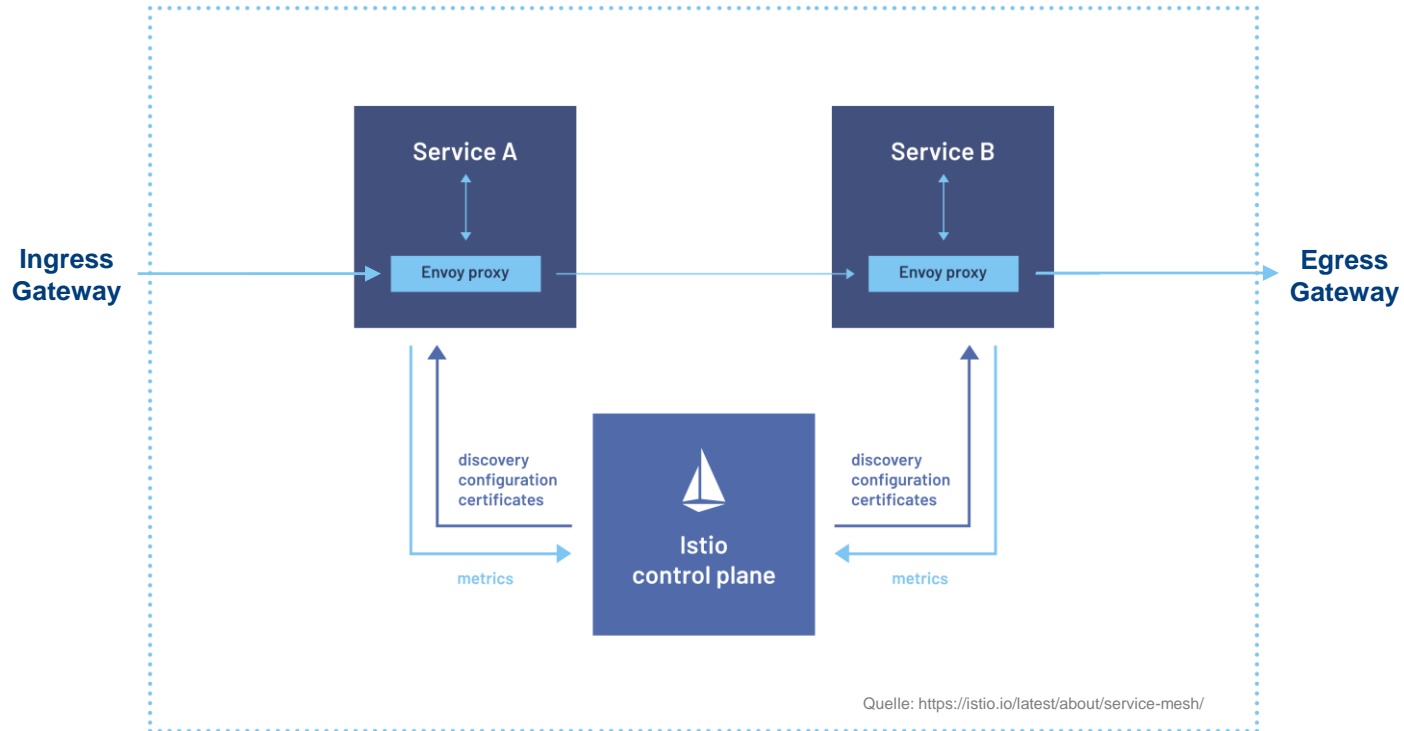
Was ist ein Service Mesh?

Infrastruktur-Framework

- Steuerung der Kommunikation (Canary Deployment, Header-based Routing)
- Überwachung der Kommunikation
- Sichere Kommunikation
- Resilienz (Circuit Breaker, Retry Mechanismus, Timeouts,...)

Service Mesh

Istio Architekturbild



Service Mesh

Herausforderungen

- Abbildung der Komplexität der Eigenlösung in einer Service Mesh
- Konfigurationen über die Releaseprozesse bereitstellen
- Housekeeping
- Verantwortlichkeiten neu definieren
- Sanfte Migration durch Side By Side Ansatz
 - keine zusätzlichen Namespaces
 - keine parallel installierten Versionen
 - Wiederverwendung der Infrastruktur (Hazelcast, AMQ,..)

Service Mesh

Nächste Schritte

Aktuelle Tasks:

- Ausgewählte Applikationen in der Testzone migrieren
- Workshop mit Dynatrace
- Anpassungen unserer Release Pipeline
- Verifizierung mit Experten (Redhat / IBM)

Zukünftige Tasks:

- Umstellung in Produktion (Sanfte Migration)
- Weitere Aspekte (Fehlertoleranz, Securityprüfungen,...) in die Service Mesh verlagern

Kontakt

Cornelia Neumüller

IT-Architekt
Raiffeisen Software GmbH

Goethestraße 80
4020 Linz
+43 7326929 - 21354
+43 676 814121354

cornelia.neumueller@r-software.at
www.r-software.at

Thomas Kriechbaum

IT-Architekt
Raiffeisen Software GmbH

Goethestraße 80
4020 Linz
+43 7326929 - 21372
+43 676 81411372

thomas.kriechbaum@r-software.at
www.r-software.at

