



MODERNE SERVICEARCHITEKTUR IM AUSLANDSSCHULWESEN

Auch wenn der Einsatz von Microservices für die Behörden-IT mit erheblichen Einschränkungen verbunden ist, muss man diesen Ansatz nicht gleich komplett ad acta legen. Anhand eines aktuellen Projekts im Bundesverwaltungsamt (BVA) zeigen wir eine moderne Softwarearchitektur, die Aspekte aus serviceorientierter Architektur (SOA) und Microservices verknüpft.

| von **ANDREAS RAQUET** und **RALF LEONHARD**

Mit Microservices hat sich in den letzten Jahren ein neuer Architekturstil etabliert. Die Kombination von Serviceorientierung, DevOps und Cloud-Computing ermöglicht ein Entwicklungsmodell, das radikal auf effiziente, ja sogar „experimentelle“ Entwicklung getrimmt ist.¹

Der Ansatz hat sich vor allem bei Internetunternehmen bewährt. Deren Webanwendungen müssen sehr schnell entwickelt und an sich in schneller Folge ändernde Anforderungen umgestellt werden. An einer einzelnen Anwendung arbeiten oft mehr als 100 Entwickler, und die Anwendungen müssen auf mehrere Millionen Nutzer skalieren. Neue Funktionsmerkmale müssen binnen kürzester Zeit in Betrieb genommen werden – oft als unmittelbare Reaktion auf einen Schachzug der Konkurrenz.

In der Behörden-IT sind solche Ansätze schwer umzusetzen. Zum einen sind sie kaum mit der Praxis der für den Betrieb verantwortlichen IT-Dienstleistungszentren (IT-DLZen) vereinbar. Zum anderen besteht auch gar nicht der Bedarf für dieses extreme Maß an Skalierbarkeit und Flexibilität um jeden Preis.

Dieser Artikel zeigt anhand eines aktuellen Großprojekts, wie moderne Software-Architekturen dennoch zentrale Stärken der Microservices auch im Behördenumfeld nutzbringend umsetzen können.

PRAXISBEISPIEL AUSLANDSSCHULWESEN

Im Projekt DAISY (Deutsches Auslandsschularbeits-Informationssystem) entsteht über einen Zeitraum von mehreren Jahren eine neue Anwendungslandschaft für die Zentralstelle für das Auslandsschulwesen (ZfA), eine Fachabteilung des Bundesverwaltungsamts (BVA). Rund 100 Mitarbeiter und Mitarbeiterinnen der ZfA und ungefähr 3.000 externe Benutzer, wie beispielsweise Schulleiter/-innen, bearbeiten alle Angelegenheiten des Auslandsschulwesens, von der Förderung von Schulen und Lehrkräften, über die Personalgewinnung und -vermittlung bis hin zur Abwicklung des Deutschen Sprachdiploms.

¹ Siehe Raquet, Andreas, Sind Microservices die bessere SOA?, .public 01-2017

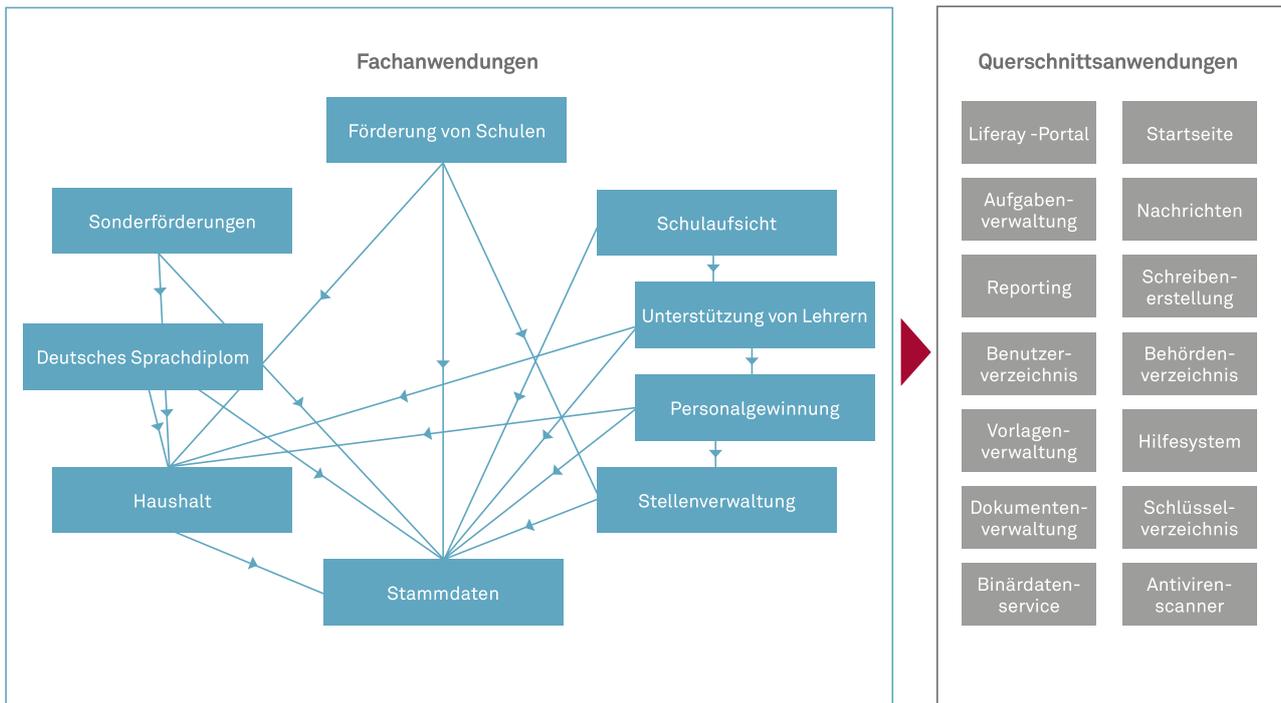


Abbildung 1: DAISY-Anwendungslandschaft

Das System wird im Vollausbau aus rund zehn Fachanwendungen bestehen, die direkt die genannten Geschäftsprozesse unterstützen. Ungefähr 20 Querschnittsanwendungen übernehmen übergreifende Dienste wie Benutzer- und Behördenverzeichnis, Aufgaben- oder Dokumentenverwaltung. Derzeit befinden sich eine Fachanwendung und zehn Querschnittsanwendungen im Produktivbetrieb. Alle Fachanwendungen verfügen über eine einheitliche, nach dem Styleguide des BVA gestaltete Weboberfläche. Der Zugang zum System erfolgt über ein zentrales Portal, basierend auf dem Liferay-Portalserver.² Über eine Reihe von Portlets kann man aktuelle Nachrichten, Berichte und Aufgaben auf einen Blick einsehen. Auch die Oberfläche des Portalservers ist nach dem Styleguide des BVA gestaltet, sodass sich die gesamte Anwendungslandschaft dem Nutzer wie eine große Anwendung präsentiert.

EINE KLASSISCHE SOA ...

Ausgeschrieben wurde DAISY im Jahr 2014 als Anwendungslandschaft nach den „Grundregeln einer SOA“. Tatsächlich besteht die Lösung aus zahlreichen Anwendungen. Jede davon setzt eine klar umrissene fachliche Domäne um und bietet

ihre Funktionalität über eine über das Portal anzusteuern Weboberfläche sowie andere Anwendungen über Services an. Die Anbindung an externe Behörden erfolgt perspektivisch mittels SOAP-Webservices, die über einen Service Gateway Zugang zur Anwendungslandschaft erhalten.

Die Architektur des Gesamtsystems basiert auf der Referenzarchitektur der IsyFact des BVA – einer im Jahr 2006 entwickelten klassischen SOA-basierenden Referenzarchitektur (siehe Infobox „IsyFact“). Die Referenzarchitektur kommt in mehreren Anwendungslandschaften des BVA zum Einsatz, allen voran im Registerportal, der zentralen Plattform des BVA für den Betrieb von Registern im Bereich der öffentlichen Sicherheit.

Alle bisher aufgeführten Charakteristika sind typisch für SOA-Anwendungslandschaften. Aber hier enden die Analogien auch schon. Das sieht man bereits am Umfang. Auch wenn DAISY mit mehr als 150 Personenjahren Gesamtvolumen ein Großprojekt ist, handelt es sich doch nur um eine einzelne Fachdomäne. Damit ist die Anwendungslandschaft weit von einer behördenweiten Anwendungslandschaft auf Basis einer zentralen Infrastruktur entfernt, wie die klassische SOA dies vorsieht.

² Siehe <https://www.liferay.com/de>

Aspekt	SOA	DAISY	Microservices
Geltungsbereich	Gesamte Behörden-IT	Einzelanwendung	Einzelanwendung
Steuerung	Architekturboard	Architekturboard	Nicht vorgesehen ⁴
Infrastruktur	JEE Server, ESB, Process Server, Governance Suite	Webapplication Server ⁵	Webapplication Server ⁵
Datenhaltung	Relationales DBMS	Relationales DBMS	Unterschiedlich, oft NoSQL Datenbanken
Netztopologie	Layered SOA via Bus	Punkt-zu-Punkt	Punkt-zu-Punkt
Kommunikationsprotokoll	SOAP, WS-*	http	http, REST
Frameworks	JEE, Spring	JEE, Spring	Spring IO, Netflix Stack
Anwendungsschnitt	Keine Vorgabe (oft Altsysteme)	Domain Driven Design ⁷	Domain Driven Design
Anzahl Services	Sehr unterschiedlich	Ca. 25	Mehrere Hundert
Umfang eines Services	Eher groß	Eher groß ⁹	Klein ⁹
Betriebsmodell	Nach ITIL oder vergleichbar	Nach ITIL oder vergleichbar	Durch Entwicklungsteam (DevOps) definiert
Betriebsplattform	Klassisches Rechenzentrum	Klassisches Rechenzentrum	Cloud-Plattformen, z. B. AWS

Tabelle 1: DAISY zwischen SOA und Microservices

Auch das Schichtenmodell der klassischen SOA findet man hier nicht. Dieses sieht eine Anordnung der Services in drei Schichten – Prozessdienste, zusammengesetzte Dienste, Basisdienste – mit klar definierten Aufrufbeziehungen vor. In der Praxis hat sich die Schichtung jedoch nicht bewährt. Bereits in der IsyFact-Referenzarchitektur wurde die Anzahl der Schichten auf zwei beschränkt: auf Geschäftsanwendungen und Register. In DAISY gibt es keine Register. Die Fachanwendungen werden rein nach fachlichen Aspekten geschnitten und beinhalten sowohl Prozesse als auch Daten.

Auch die für klassische SOA typischen, schwergewichtigen Technologien gibt es in der IsyFact nicht, weder einen Enterprise Service Bus (ESB) noch einen Process Server oder eine Governance Suite. Selbst auf „echte“ Applikationsserver wird verzichtet. Sämtliche Anwendungen laufen auf Tomcat-Webapplication-Servern, was selbst bei den durchaus komplexen Anwendungslogiken und den mittelschweren Lastanforderungen völlig ausreichend ist.

Schließlich spielt auch der in der SOA zentrale Ansatz der hausweiten Wiederverwendung dedizierter Dienste kaum eine Rolle. Es ist nicht Ziel des Projekts, die Dienste des Auslandsschulwesens außerhalb des geplanten Einsatzfeldes zu nutzen.³ Ein Be-

darf dafür ist nicht absehbar. Gleichmaßen nutzt auch DAISY nur vereinzelt die anderweitig im BVA bereitgestellten Dienste.

... ODER DOCH MICROSERVICES?

Auch wenn DAISY in einigen Bereichen von der klassischen SOA abweicht: Um eine Microservices-Architektur handelt es sich deswegen noch nicht. Bei näherem Hinsehen finden sich allerdings mehr Parallelen, als vielleicht erwartet. So stellt DAISY das Nachfolgesystem für das bisherige IT-System ISAS (InformationsSystem AuslandsSchulwesen) dar, im Wesentlichen ein Monolith, der jetzt – ganz nach der Lehre der Microservices – in Einzelanwendungen aufgespalten wird.

Die Fachanwendungen sind dabei nach fachlichen Prozessen und damit implizit auch nach den dafür zuständigen Fachabteilungen zugeschnitten. Das entspricht genau dem fachlichen Schnitt, wie er im Domain-Driven-Design gefordert und von den Microservices aufgegriffen wird. Alle Anwendungen verfügen über eigenständige Datenbestände. Dabei werden auch Redundanzen zwischen den Anwendungen akzeptiert. Damit stimmen sowohl der Geltungsbereich als auch das grundsätzliche Vorgehen schon einmal gut mit Microservices-Ideen überein.

³ In DAISY entwickelte Querschnittsanwendungen werden durchaus auch in anderen Umgebungen genutzt, allerdings nicht im Sinne einer SOA. Stattdessen werden dort eigene Instanzen betrieben.

⁴ Viele Anwender etablieren dennoch ein zentrales Gremium, das den Ausbau der Servicelandschaft koordiniert und die Entwicklungsteams berät. Dieses agiert jedoch weniger formal als ein Architektur-Board.

Auch technologisch folgt DAISY ähnlichen Ansätzen wie die Microservices: Anstelle schwergewichtiger, zentraler Infrastruktur werden die Anwendungen dezentral entwickelt und Punkt-zu-Punkt integriert. Als Betriebsinfrastruktur kommen lediglich leichtgewichtige Tomcat-Application-Server zum Einsatz, wie sie in eingebetteter Form auch bei Microservices Verwendung finden. Ansonsten kommunizieren die Anwendungen über den bewährten Spring-httpInvoker-Mechanismus. Der ist zwar nicht gerade „bleeding edge technology“, leistet aber zuverlässig und ohne großen Overhead seine Dienste – und das sogar noch leichtgewichtiger als die bei Microservices verbreiteten REST-Services. Der Preis dafür: Die Technologie steht nur in Java zur Verfügung.

EIN PRAGMATISCHER MITTELWEG

Microservices-Verfechter werden anmerken, dass DAISY damit immer noch weit von einer Microservices-Architektur entfernt ist. So besteht DAISY nur aus rund 25 Anwendungen und nicht aus Hunderten von Services. Die Anwendungen sind damit auch deutlich größer und somit eben nicht „micro“. Die Infrastruktur ist zwar leichtgewichtig, aber es gibt noch klassische Webapplication-Server. Und überhaupt kommen die vielen modernen Frameworks, die sich mittlerweile mit dem Microservices-Banner schmücken, gar nicht zum Einsatz.

Allerdings war es auch gar nicht das Ziel der Architektur von DAISY, als Microservices-Architektur vermarktet werden zu können. Vielmehr greift die Architektur von DAISY pragmatisch Erkenntnisse und Erfahrungen aus vielen Jahren der Implementierung serviceorientierter Architektur auf und setzt diese konsequent um, um den ursprünglichen Zielen der SOA – Effizienz und Flexibilität – gerecht zu werden:

- Serviceorientierung im Kleinen, auf Ebene einer Fachdomäne.
- Weitgehender Verzicht auf schwergewichtige zentrale Infrastruktur.
- Fachlicher Anwendungsschnitt entlang von Fachaufgaben nach Domain-Driven-Design.

Eine weiterführende Umsetzung der Ansätze von Microservices würde auch eine Umsetzung der tragenden Trends DevOps und Cloud-Computing erfordern. Erst das würde eher „explorative“ Entwicklungsansätze ermöglichen, wie man sie beispielsweise bei Netflix findet. Und erst in diesem Szenario entfalten die typischen Technologien, wie sie heute beispielsweise in der Spring-IO-Plattform gebündelt sind, ihren Nutzen. Derzeit orientieren sich die IT-DLZ allerdings eher an ITIL und entfernen sich damit sogar von diesen Trends, wie bereits in der .public-Ausgabe 01-2017 nachzulesen ist. Andererseits benötigen Fachverfahren

5 Eingebettet in die Applikation

6 Nicht eingebettet

7 Implizit durch Schnitt entlang der Fachprozesse und der ausführenden Fachabteilungen

INFOBOX ISYFACT REFERENZARCHITEKTUR

Bereits 2006 hatte das BVA im Rahmen der Migration von host-basierten Altsystemen auf eine moderne serviceorientierte Architektur damit begonnen, für die Domäne „Register“ eine Umgebung aufzubauen, die Referenzarchitektur, Bausteine, Werkzeuge, Methodik und betriebliche Plattform zusammenfasst. Diese wird seither unter dem Namen Register Factory (RF) sehr erfolgreich in einer Vielzahl von Verfahren im Registerportal des Bundesverwaltungsamtes eingesetzt und weiterentwickelt. Seit 2008 stellt das BVA diesen Standard auf Anfrage auch anderen Institutionen zur Verfügung, die darauf aufbauend eigenständig ihre spezifischen Anwendungen entwickeln und betreiben können.

In den folgenden Jahren wurde die Plattform unter dem Namen IsyFact verallgemeinert, sodass nun auch andere Anwendungssysteme als Register damit umgesetzt werden können. Seit 2015 steht die IsyFact auch nichtbehördlichen Anwendungen als Open Source zur Verfügung.

Die zugrunde liegende Referenzarchitektur adressiert den Aufbau komplexer SAGA-konformer, serviceorientierter Anwendungslandschaften, die langfristig betrieben und weiterentwickelt werden können. Dazu werden Blaupausen bereitgestellt, die eine durchgängige Konzeption der Anwendungslandschaft, der Anwendungssysteme bis hin zu den Anwendungskomponenten ermöglichen. Hierbei wurde auf bewährte Architekturmuster und -prinzipien zurückgegriffen, wie die Definition der Referenzarchitektur in unterschiedlichen Sichten (fachlich, software-technisch, technische Infrastruktur), Unterteilung der Anwendungslandschaft in fachliche Domänen und kategorisierte Systemtypen, starke Komponentenorientierung, Mehrschichtarchitektur und vieles mehr.

Die IsyFact sieht den Aufbau einer serviceorientierten Anwendungslandschaft vor. Dienste werden vertikal in die verschiedenen fachlichen Domänen und Querschnitte unterteilt. Vertikal wird unterschieden zwischen Registern, die Basisdienste zur Verwaltung der Daten bereitstellen, und Geschäftsanwendungen, die die Basisdienste nutzen und selbst komplexe fachliche Dienste (sowohl Prozessdienste als auch zusammengesetzte Dienste) bereitstellen.

Weitere Information unter: <http://www.isyfact.de>

8 Vereinzelt werden Größenbeschränkungen auf bis zu 200 Codezeilen gefordert
9 Größenordnung: ca. 100.000 Codezeilen

zur IT-Unterstützung der Prozesse des Auslandsschulwesens, deren Änderungsfrequenz eher in Jahren als in Tagen gemessen wird, nicht unbedingt explorative Entwicklungsansätze.

Das ursprüngliche Ziel der Microservices erreicht DAISY mit seinem pragmatischen Ansatz durchaus: An die Stelle eines riesigen Anwendungssystems tritt eine Anwendungslandschaft, deren einzelne Bausteine noch überschaubar und beherrschbar sind und die damit über die nächsten Jahre geordnet weiterentwickelt werden können.

FAZIT

Klassische SOA und Microservices bewegen sich aufeinander zu. Während die klassische SOA einige zentrale Versprechungen niemals einlösen konnte, sind Microservices in ihrer Reinform – zumindest im Behördenumfeld – kaum umsetzbar. Wie aber in diesem Artikel beschrieben, lassen sich Software-Architekturen irgendwo in der Mitte zwischen SOA und Microservices so gestalten, dass zum einen die Fehler der klassischen SOA vermieden, zum anderen aber auch die Umsetzungsprobleme der Microservices umgangen werden. Diese Verbindung liefert also nicht etwa „das Schlechte aus beiden Welten“, sondern einen pragmatischen Architekturansatz, den das BVA als Hausstandard IsyFact etabliert hat und weiterentwickelt. Dieser kommt dem Ziel der Effizienz und Flexibilität näher als beide Extreme. Dabei ist der Ansatz sowohl mit den Forderungen der Behörden-IT nach Serviceorientierung als auch mit den aktuellen Betriebsmodellen der IT-DLZ gut vereinbar.

Der in DAISY verfolgte Ansatz liegt auf diese Weise nahe bei einem weiteren aktuellen Architekturstil, den „Self-Contained Systems“, die sich in jüngerer Zeit als einfachere, weniger radikale Alternative zu Microservices etablieren. Allerdings bauen diese Self-Contained Systems, ähnlich wie Microservices, auf Continuous Delivery und DevOps und dürften damit ähnlich schwer in die Behörden-IT zu übertragen sein.

DAISY hat mit seinem Architekturansatz gezeigt, dass man moderne und klassische Architekturansätze zu einem pragmatischen Gesamtkonzept verbinden kann. Dadurch lassen sich die Stärken der Microservices und der klassischen SOA für die Herausforderungen nutzen, die bei der Modernisierung von Altverfahren in den nächsten Jahren auf die Bundesverwaltung zukommen. ●

ANSPRECHPARTNER – RALF LEONHARD

IT-Architekt
Bundesverwaltungsamt



ANSPRECHPARTNER – ANDREAS RAQUET

Principal IT-Consultant
Public Sector Solutions Consulting

