

Event-Driven-Design Serviceorientierter Architektur für das schweizerische Personenmeldewesen

Susanne Patig ^{*}), Willy Müller [°])

^{*}) Universität Bern
Institut für Wirtschaftsinformatik
Engelhaldestrasse 8
CH-3012 Bern
susanne.patig@iwi.unibe.ch

[°]) Eidgenössisches Finanzdepartement EFD
Informatikstrategieorgan Bund ISB
Architekturen, Standards, Technologien AST
Friedheimweg 14
CH-3003 Bern
willy.mueller@isb.admin.ch

Abstract: Ein wesentliches Motiv für die Einführung von serviceorientierter Architektur (SOA) ist die Überwindung fachlicher, technischer und organisatorischer Heterogenität. Im föderalistischen Kontext der Schweiz geht diese Heterogenität einher mit der weitgehenden Autonomie der Organisationseinheiten, die Verwaltungsprozesse bearbeiten. Folglich unterscheiden sich die Verwaltungsprozesse in der Art, Anzahl, Reihenfolge und dem Automatisierungsgrad der Aktivitäten, in den verwendeten Datenstrukturen und den direkt oder indirekt beteiligten Akteuren. Diese Ausgangslage führt dazu, dass traditionelle Ansätze des Designs von Software-Services in SOA nicht anwendbar sind. Der Beitrag zeigt am Beispiel des schweizerischen Personenmeldewesens eine alternative Methode für das Software-Service-Design, die auf Ereignissen beruht, und stellt den Event-Bus Schweiz vor, die implementierte Infrastruktur der resultierenden SOA. Die im Personenmeldewesen und im Bereich Geoinformationssysteme gemachten Erfahrungen mit SOA werden abschließend zu einem generellen Vorgehensmodell für das SOA-Service-Design verallgemeinert.

1 Motivation

Die E-Government-Strategie der Schweiz, die der Bundesrat am 24. Januar 2007 verabschiedet hat, nennt als Ziele [GS07, S. 4] die elektronische Abwicklung des Geschäftsverkehrs (1) zwischen Wirtschaft und Behörden, (2) zwischen und innerhalb von Behörden sowie (3) zwischen Behörden und Bevölkerung – vor allem in den häufigsten Geschäftsfällen. Durch die elektronische Abwicklung werden Medienbrüche vermieden, was Kosten spart und Durchlaufzeiten für Entscheidungen verkürzt. Für die Bürgerinnen und Bürger soll die Verwaltung insgesamt transparenter und (aufgrund des Abbaus zeit-

licher und räumlicher Einschränkungen) flexibler werden; im Idealfall entfallen unnötige Behördengänge ganz.

Die Umsetzung der E-Government-Strategie ist vor dem föderalistischen Hintergrund der Schweiz eine Herausforderung, weil sie es erzwingt, alle potentiellen Anbieter von E-Government-Leistungen miteinander zu vernetzen, d.h. den Bund, 26 Kantone und 2765 Gemeinden. Erschwert wird die Vernetzung durch die weitgehende Autonomie der Anbieter (alle haben eigene Gesetzgebungs- und Regelungshoheit) und die daraus resultierende Heterogenität (s. Abschnitt 2).

Ein Bereich, in dem sämtliche Ebenen der schweizerischen Verwaltung zusammenarbeiten müssen und dessen Kontakt zu den Bürgern direkt und häufig ist, ist das *Personenmeldewesen*. Damit sind alle Prozesse ‚zu einer Person‘ gemeint, z. B. das Zivilstandswesen, Ausweise, Zu-/Wegzüge; Details erläutert Abschnitt 2. Von einer elektronischen Umsetzung dieser Prozesse im Sinne der E-Government-Strategie profitieren die Bürgerinnen und Bürger direkt; sie hat deshalb hohe Priorität.

Im Jahr 2007 wurde auf Bundesebene entschieden, die E-Government-Strategie überall und damit auch im Personenmeldewesen durch serviceorientierte Architektur (SOA) zu realisieren [S007]. Ausschlaggebend dafür waren vor allem die positiven Erfahrungen, die *Swisstopo*, das Geoinformationszentrum des Bundes, mit der Einführung von SOA gemacht hatte: Im Jahr 2002 begann man dort, Karten- und Luftbilddienste¹ (WMS: Web Mapping Service) sowie kartenbezogene Suchdienste (z.B. alle Gemeinden in einem bestimmten Umkreis) als Web-Services anzubieten. Die Dienste werden sehr stark genutzt, da sie es den Kunden ersparen, mehr als ein Terabyte Datenspeicher für die Kartendaten vorzuhalten; außerdem sind die Karten immer aktuell (kein Update erforderlich) und bereits in Geoinformationssysteme (GIS) eingebunden (kein Entwicklungs-Know-how nötig). Kostenschätzungen gehen davon aus, dass Anwendungen auf Basis der *Swisstopo*-Kartendienste einen Entwicklungsaufwand von 20 Personentagen sowie Betriebskosten von CHF 15.000-30.000 pro Jahr verursachen – im Vergleich zu 200-400 Personentagen und CHF 200.000-500.000 für die Eigenentwicklung einer GIS-basierten Lösung.

Insgesamt und übertragen auf den Bereich Personenmeldewesen erhoffte man durch den Einsatz von SOA folgende *Vorteile*: (1) Machbarkeit der E-Government-Strategie auf den unteren Ebenen der Verwaltung (z.B. Gemeinden), die nicht über eigene Ressourcen für die Entwicklung und den Betrieb von Software verfügen; (2) schnelle, flexible Reaktion auf spezifische Anforderungen der einzelnen Verwaltungseinheiten und sonstigen Nutzer sowie (3) Überwindung der technologischen, funktionalen und organisatorischen Heterogenität, die sich aus der weitgehenden lokalen Autonomie innerhalb der föderalistischen Struktur der Schweiz ergibt. Vor allem die Überwindung von Heterogenität war im Bereich Personenmeldewesen ein zentrales Argument für SOA; Abschnitt 2 schildert die entsprechenden Besonderheiten. Diese Besonderheiten trugen allerdings dazu bei, dass traditionelle Vorgehensweisen des Software-Service-Design (s. Abschnitt 3) nicht anwendbar waren. Abschnitt 4 beschreibt das Vorgehen für das Software-Service-Design im Personenmeldewesen und die resultierende Lösung, den Event-Bus Schweiz. Abschnitt 5 verallgemeinert das Vorgehen und gibt Hinweise zu seiner Anwendbarkeit in anderen (Verwaltungs-) Kontexten.

¹ Siehe z.B. *ecoGIS* unter <http://www.ecogis.admin.ch/>

2 Ausgangslage im schweizerischen Personenmeldewesen

Das schweizerische Personenmeldewesen umfasst 103 Prozesse in den Bereichen Zivilstandswesen (Eheschein, Scheidungsantrag etc.), Ausweise (Identitätskarte, Reiseausweis, Lebensbescheinigung etc.), Zu-/Wegzüge, Aufenthaltsbewilligungen für verschiedene Ausländerkategorien, Einbürgerungen sowie Informationsdienstleistungen und Statistiken (z.B. Volkszählungen). Meist ist an diesen Prozessen mehr als eine Verwaltungsebene beteiligt. Tabelle 1 zeigt dies anhand eines Ausschnitts aus dem Prozessinventar; einen Beispielprozess (Zuzug/Wegzug) erläutert Abschnitt 4.1.

Tabelle 1: Prozessinventar Personenmeldewesen (Auszug)²

L. Nr.	Leistung (L)	ID	Benennung		Ebene					Hauptakteur	Primärverantwortung
			Objekt	Aktion	Schweiz	Kanton	Gemeinde	Private Untern.	Intern		
05	Aufenthaltsbewilligung										
05	Aufenthaltsbewilligung	01124	Ausweis B EG/ EFTA (Aufenthaltsbewilligung)	beantragen	x	x	x	x			IMES
05	Aufenthaltsbewilligung	01126	Ausweis CI EG/EFTA (Aufenthaltsbewilligung mit Erwerbstätigkeit)	beantragen	x	x	x	x			IMES
05	Aufenthaltsbewilligung	01128	Ausweis G EG/EFTA (Grenzgäumberwilligung)	beantragen	x	x	x	x			IMES
05	Aufenthaltsbewilligung	01130	Ausweis B (Aufenthaltsbewilligung)	beantragen	x	x	x	x			IMES
10	Ausweise (inkl. Bescheinigung)										
10	Ausweise (inkl. Bescheinigung)	01137	Identitätsausweis (für vorläufig Aufgenommene und Asylsuchende)	beantragen		x	x	x			BFF
10	Ausweise (inkl. Bescheinigung)	01141	Identitätskarte	bestellen			x	x			
10	Ausweise (inkl. Bescheinigung)	01144	An-/Abmeldebescheinigung	bestellen			x	x			
10	Ausweise (inkl. Bescheinigung)	01151	Heimatschein	bestellen			x	x			ZSA
10	Ausweise (inkl. Bescheinigung)		Mutation von Ausländerausweisen	durchführen		x	x	x			
15	Einbürgerung/Staatsbürgerschaft										
15	Einbürgerung/Staatsbürgerschaft	01158	Einbürgerung von Ausländern auf Bundesebene	beantragen	x	x	x	x			
15	Einbürgerung/Staatsbürgerschaft	01161	Einbürgerung von Ausländern im Kanton	beantragen		x	x	x			
15	Einbürgerung/Staatsbürgerschaft	01162	Einbürgerung von Schweizern in einem anderen Kanton	beantragen		x	x	x			
20	Wegzug/Zuzug										
20	Wegzug/Zuzug	01167	Aufrechterhaltung der Niederlassungsbewilligung von AusländerInnen	beantragen		x	x	x			
20	Wegzug/Zuzug	01168	Adressänderung innerhalb einer Gemeinde	melden				x			
20	Wegzug/Zuzug	01172	Wohnsitz	anmelden				x	x		
20	Wegzug/Zuzug	01173	Wohnsitz	abmelden				x	x		
20	Wegzug/Zuzug	01174	Niederlassungsausweis/Aufenthaltsbewilligung	bestellen				x	x		
25	Zivilstandswesen										
25	Zivilstandswesen	01181	Eheschein	bestellen				x	x		ZSA
25	Zivilstandswesen	01186	Scheidungsantrag	einreichen				x	x		ZSA
25	Zivilstandswesen		Informationen bzgl. Heirat für ausländische Staatsangehörige	einsehen		x	x	x			
99	Informationsdienstleistungen/Statistiken										
99	Informationsdienstleistungen/Statistiken		Volkszählung	durchführen	x	x	x	x			
99	Informationsdienstleistungen/Statistiken		Pressecommunique	bereitstellen	x	x	x			x	
99	Informationsdienstleistungen/Statistiken		Amtsblatt	einsehen		x	x	x	x		
99	Informationsdienstleistungen/Statistiken		Informationen bzgl. Geburten	einsehen		x	x	x	x		

Zusätzlich zur Anzahl und dem Variantenreichtum der Prozesse ist auch die Regelungshöhe verteilt: Das Einwohnerwesen wird beispielsweise auf kantonaler Ebene geregelt, das Zivilstandswesen auf der Ebene des Bundes. Dies führt horizontal (auf derselben Verwaltungsebene unterhalb des Bundes) zu *funktionaler Heterogenität*. Der Grad der IT-Unterstützung dieser Funktionalität ist ebenfalls nicht einheitlich: Was in einer Verwaltungseinheit (Gemeinde, Kanton) automatisiert durchgeführt wird, kann in einer anderen Verwaltungseinheit eine manuelle, papiergebundene Aktion sein. Die genaue Zahl der Anwendungen, die im schweizerischen Personenmeldewesen genutzt werden, lässt sich nicht ermitteln.

Im Gegensatz zur Mehrheit der Gemeinden verfügen der Bund und die meisten Kantone

² Leere Felder in der letzten Spalte zeigen eine fallabhängig verteilte Primärverantwortung an.

über eigene IT-Abteilungen, in denen die Entscheidungen über die Unternehmens- und Softwarearchitektur getroffen werden; 80 % der Softwareentwicklung und (vor allem bei Gemeinden) oft auch der Softwarebetrieb werden an externe IT-Dienstleister (z.B. BEDAG, VRSG, ABRAXAS) übergeben. Daraus folgt zum einen *technologische Heterogenität* (Programmiersprachen, Datenbanksysteme, Entwicklungsplattformen etc.). In Verbindung mit der verteilten Regelungshoheit entsteht aber auch *organisatorische Heterogenität*, die zu vier verschiedenen Rollen in einer SOA führt, welche in der Regel von unterschiedlichen Organisationen wahrgenommen werden [eS007, 6]: *Service-Eigner* ist die Fachstelle, welche die fachliche Verantwortung für einen Service übernimmt, z.B. die Gemeinde, die Todesfallmeldungen erzeugt. Der Leistungserbringer, der den Service (für die Fachstelle) technisch bereitstellt, heißt *Service-Betreiber*, z.B. der Hosting-Dienstleister für eine Gemeindeanwendung. Als *Service-Nutzer* kommen Verwaltungseinheiten, Bürger oder die Privatwirtschaft in Frage, die manuell oder über Anwendungen auf den angebotenen Service zugreifen, z.B. die AHV³-Stelle, die an Todesfallmeldungen interessiert ist. Eventuell bedient sich auch der Service-Nutzer eines Hosting-Dienstleisters (*Anwendungsbetreiber*) zur Bereitstellung der Anwendung, die den Service nutzt; ein Beispiel wäre der IT-Leistungserbringer der AHV-Stelle.

3 Stand der Forschung im Software-Service-Design

Auf der strategischen Ebene sind *Software-Services* elektronisch angebotene Leistungen, die von Anwendungen, anderen Software-Services oder Menschen genutzt werden können. Operativ beruht *serviceorientierte Architektur (SOA)* auf *Software-Services* als unabhängigen Programmen, die jeweils eine bestimmte, logisch eng zusammenhängende (*kohäsive*) Funktionalität bündeln, die Implementierung der Funktionalität verbergen (*Abstraktion*) und über ein Interface, das in einer standardisierten Sprache beschrieben wird (*standardisierter Vertrag*), nach außen anbieten. Das Service-Interface ist unabhängig von der Service-Implementierung (*lose Kopplung*). Zwischen Service-Aufrufen soll keine Information im Software-Service zwischengespeichert werden (*Zustandslosigkeit*), um insgesamt zu ermöglichen, dass die Software-Services in Verwendungsszenarien wiederverwendet werden können, die beim Service-Design noch nicht bekannt waren. Die kursiv gesetzten Begriffe bilden die *Grundprinzipien von SOA* [Er08a], [PH06]. Grob gibt es zwei Gruppen von Ansätzen für das Design von Software-Services in SOA⁴: *Prinzipiengetriebene Ansätze* (z.B. [Er08b]) geben eine Reihe von Pattern an, die dabei helfen, die SOA-Prinzipien beim SOA-Design zu realisieren. Ein systematisches Vorgehen von den Anforderungen zu SOA (in Spezifikation oder Implementierung) fehlt jedoch. Solch ein Vorgehen findet sich in den *hierarchischen Ansätzen*, die entweder top-down oder bottom-up angelegt sind. *Top-down* beginnt das Design von Software-Services bei Geschäftsanforderungen⁵ (z.B. Zielen [KSR04], [Ar+08], funktionalen Bereichen [Ar+08] und Aufgaben, [LEG08] oder Geschäftsprozessen [Ar+08] [KKB07]) und fügt immer detailliertere technische Informationen zu – bis letztlich zur Software-Service-Spezifikation. Ausgangspunkt für das *Bottom-up-Design* von Software-Services

³ AHV: Alters- und Hinterlassenenversicherung; <http://www.ahv-iv.info/>

⁴ Interessierte Leser/-innen finden mehr Informationen und weitere Quellen in [Pa09].

⁵ Primär sind diese Anforderungen funktional, sie können jedoch auch die Softwarequalität und Randbedingungen betreffen.

sind existierende Anwendungen, deren implementierte Funktionalität (oft mittels Techniken des Reverse Engineering; z.B. [LEG08]) extrahiert und dann zu Software-Services abstrahiert wird.

Da die schweizerische Verwaltung bemüht ist, ein Prozessdenken zu verankern (die Fachgruppe ‚Prozessinventar‘ hat alle E-Government-relevanten Prozesse erfasst und in eine Klassifikation eingeordnet [MEE04]), schien es naheliegend, das SOA-Design für das Personenmeldewesen top-town durchzuführen, ausgehend von den Geschäftsprozessen. Die dabei gemachten Erfahrungen schildert Abschnitt 4.1.

4 Umsetzung von SOA im schweizerischen Personenmeldewesen

4.1 Vorgehen beim Software-Service Design

Das Prozessinventar (s. Tabelle 1) listet die E-Government-relevanten Prozesse auf, ordnet diese Prozesse in eine Klassifikation ein und gibt die Prozess-Akteure an. Modelliert sind die E-Government-relevanten-Prozesse in der Regel nicht, obwohl sich die schweizerische Verwaltung zur Prozessmodellierung bekennt und dafür die Business Process Modeling Notation (BPMN [OMG09]) als Standard vorsieht. Als Ausgangspunkt für das Software-Service-Design im Personenmeldewesen wurden die Kantone und Gemeinden aufgefordert, Modelle der entsprechenden Prozesse zu erstellen. Die dabei aufgetretenen Schwierigkeiten werden exemplarisch anhand der Prozesse Zuzug/Wegzug in den Abbildungen 1 und 2 illustriert. Dargestellt ist der ‚gutartige‘ Fall, bei dem die Prüfung des Wegzugs ergibt, dass tatsächlich der Hauptwohnsitz im Sinne des ‚Lebensschwerpunkts‘ verlegt wurde.

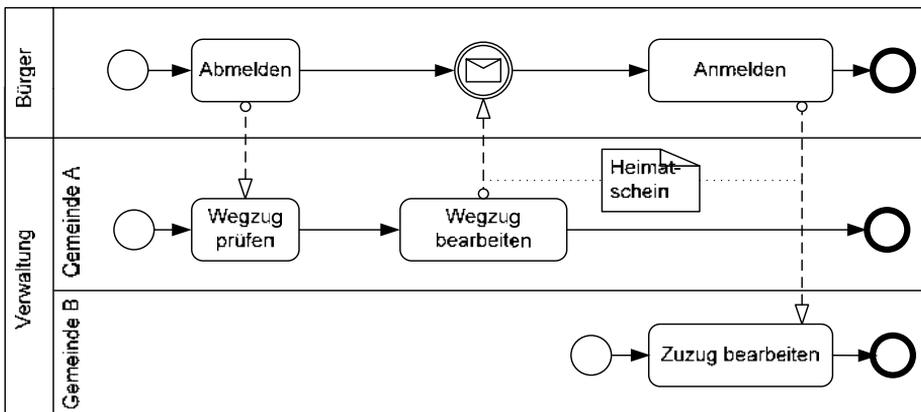


Abbildung 1: Wegzug/Zuzug mit Heimatschein (notiert mit BPMN [OMG09])

Im *Standardfall* (s. Abbildung 1) meldet sich ein wegziehender Bürger bei seiner Wohn-gemeinde A ab (Ereignis ‚Wegzug‘), erhält den Heimatschein ausgehändigt, mit dem er sich in der Zuzugsgemeinde B anmeldet (Ereignis ‚Zuzug‘). Dieser Standardfall ist z.B. in den Gemeinden der Kantone Bern und Aargau zu finden. Der Prozess in anderen Gemeinden (z.B. der Gemeinden im Kanton Baselland, Stadt Zürich) weicht davon ab (s.

Abbildung 2): Hier kann sich der Bürger direkt bei der Zugzugsgemeinde B anmelden (Ereignis ‚Zuzug‘). Die Zugzugsgemeinde B übergibt die Prüfung der Wegzugserlaubnis an die vorherige Wohngemeinde A. Das Ereignis ‚Wegzug‘ wird also nicht durch den Bürger ausgelöst, sondern durch die Wegzugsgemeinde. Erst nach der Prüfung des Wegzugs wird der Zuzug bearbeitet (umgesetzt oder storniert). Dabei ist der Wegzug z.T. erst dann vollzogen, wenn die Gemeinde B den Heimatschein von der Gemeinde A erhält (Zürich); die Gemeinden in Baselland haben hingegen den Heimatschein abgeschafft.

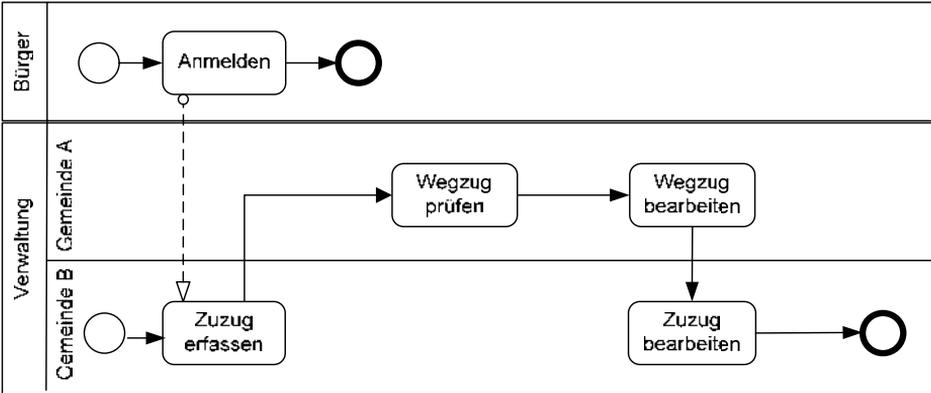


Abbildung 2: Vereinfachter Zuzug (notiert mit BPMN [OMG09])

Das Beispiel zeigt, dass die laut Klassifikation gleichen Prozesse in Verwaltungseinheiten derselben Ebene beträchtlich variieren – als Folge interkantonaler und interkommunaler Unterschiede (verschiedene gesetzliche Vorgaben und organisatorische Umsetzungen). Im Detail betreffen die Unterschiede die Art, Anzahl und Reihenfolge der Aktivitäten eines Prozesses, den Automatisierungsgrad der Aktivitäten, die (nicht) verwendeten Datenstrukturen (Heimatschein) sowie die Akteure, die direkt oder indirekt am Prozess beteiligt sind. Indirekte Beteiligung meint die Folgen eines Ereignisses: Beispielsweise muss ein Zuzug in einigen Gemeinden an die jeweilige Kirche gemeldet werden (z.B. Bern, Zürich), in anderen Kantonen (z.B. Genf) darf die Religion nicht erfasst und folglich auch nicht weitergeleitet werden. Außerdem haben einzelne Gemeinden eigene Wasserwerke (z.B. Belp), die über einen Zuzug informiert werden müssen, die meisten Gemeinden haben jedoch keine.

Natürlich lassen sich lokale Besonderheiten durch eine Verallgemeinerung ausschalten, was zu einem generischen Verwaltungsprozess mit den Aktivitäten ‚Eröffnen des Verfahrens‘, ‚Anliegen prüfen‘ und ‚Antrag entscheiden‘ führt. Nur sind diese Aktivitäten zu allgemein, um daraus IT-Unterstützung ableiten zu können.

Im Rahmen des Software-Service-Designs stellten sich die Ereignisse (z. B. ‚Zuzug‘, ‚Wegzug‘) schnell als einziger stabiler Sachverhalt heraus, der gleichzeitig die Autonomie des jeweiligen Behördenhandelns garantierte. Die kontroversen Diskussionen zu den Prozessmodellen wurden deshalb abgebrochen und durch eine Standardisierung der Ereignismeldungen ersetzt [eCH-0020]. Mit Ausnahme der zeitlich ausgelösten Meldungen des Bundesamts für Statistik werden *Ereignismeldungen* versandt, wenn fachliche

Ereignisse zur Registrierung oder Änderung von Daten im Einwohner- bzw. Zivilstandsregister führen, wie z. B. Geburt, Tod, Eheschließung, Scheidung, Einbürgerung, Zu- oder Wegzug,

Für das Personenmeldewesen wurden insgesamt 39 Ereignistypen identifiziert [eCH-0020]. Pro Ereignis sind die zu übermittelnden, standardisierten Daten spezifiziert, ohne dass im Einzelfall bekannt sein muss, welche Verarbeitungen der Ereignismeldung vorausgehen, wer die Ereignismeldungen empfängt und wie er sie weiterverarbeitet. Die *Fachservices*, die durch Anwendungen bereitgestellt werden, um eine fachliche Leistung erbringen, entsprechen dem Senden der Ereignismeldungen, z.B. ‚Senden Zuzugs-meldung‘. Abschnitt 3.2 beschreibt die technische Lösung, die diese SOA ermöglicht.

4.2 Realisierte SOA-Lösung

Der *Event-Bus Schweiz* (EBS) stellt die technische und organisatorische Infrastruktur zur Verfügung, die den Austausch von Ereignismeldungen zwischen Behörden und sonstigen Kunden erlaubt. Damit bildet der EBS das Rückgrat der SOA zur Unterstützung von organisationsübergreifenden Prozessen – soweit aus fachlicher Sicht keine Verarbeitung in Echtzeit mit Request – Reply – Interaktionen nötig ist. Der EBS ist virtuell und entsteht durch das Zusammenwirken der EBS-Services der Teilbusse, s. Abbildung 3. An einen Teilbus sind in der Regel mehrere Endsysteme über jeweilige Adapter angeschlossen.

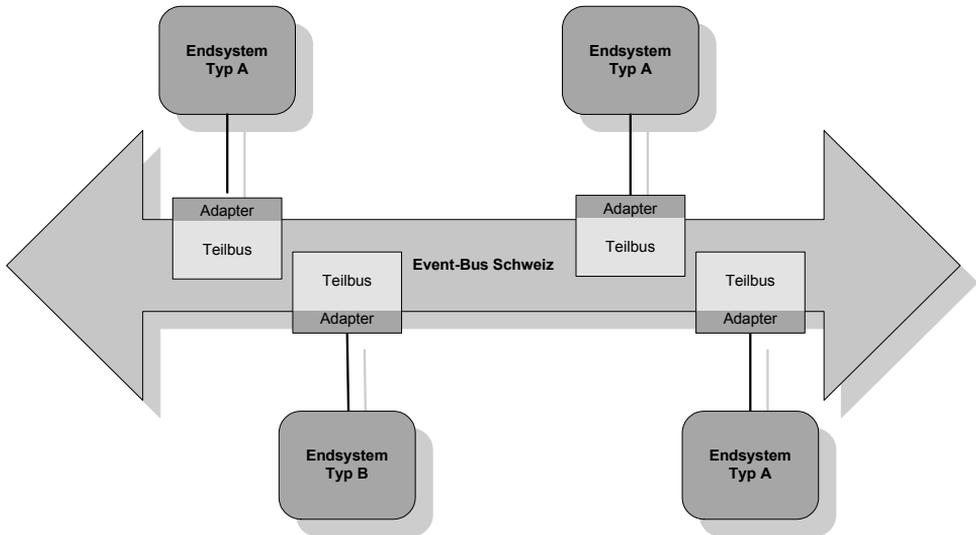


Abbildung 3: Grundstruktur des Event-Bus Schweiz (EBS)

Endsysteme (mehrerheitlich von Verwaltungsbehörden) treten im EBS als Anbieter (oder Nutzer) von *Fachservices* auf, indem sie Ereignisse verschicken (oder empfangen) - und zwar entweder manuell (Eingabe in Masken, Ausdrucken von Papier) oder technisch integriert. *Adapter* stellen durch Transformationen und Konvertierungen die Verbindung zwischen Nicht-EBS-Umgebung und EBS her. Es wird unterschieden zwischen Daten-

modell-Adaptoren (Transformation von Datenstrukturen und -typen zwischen dem Datenmodell des Endsystems und dem kanonischen Datenmodell des EBS), Datenformat-Adaptoren (Transformation zwischen XML und CSV etc.) und Transportprotokolladaptoren (Konvertierung zwischen HTTP, JMS, IIOP etc.). Das Hintereinanderschalten mehrerer Adapter ist möglich.

Die *Teilbusse* werden von einem Servicebetreiber (öffentliche Verwaltung oder Privatwirtschaft) bereitgestellt und bieten eine Vielzahl (technologischer) EBS-Services an. Die wichtigsten sind:

- *Directory-Service (Z)*: Bereitstellung aller logischen Adressen, an die der Teilbus/ EBS adressieren kann; Operationen zum Finden, Ändern, Löschen und Hinzufügen dieser Adressen
- *Event Catalogue (Z)*: Auflistung der unterstützten Ereignistypen (inkl. der Berechtigungen für Sender, Empfänger und Abonnenten der Ereignisse)
- *Abonnement Service (Z)*: Führen des Abonnentenregisters für die Ereignisse und Auslieferung von Ereignissen an interessierte Endsysteme und Teilbusse
- *Routing-Service*: Ermittlung von Zieladressen auf der Basis von Ereignistyp sowie fachlichen und rechtlichen Anforderungen (s.u.); Weiterleitung des Ereignisses an Adressaten
- *Security Services*: Dienste für Authentisierung, Autorisierung und Zugriffsschutz, Zeitstempel und Verschlüsselung

Weitere EBS-Services existieren z. B. für das nochmalige Senden, die Simulation und das Logging sowie für die Korrektur und den Widerruf von Ereignissen und die Mehrfachmeldungsbehandlung. Grundsätzlich können alle EBS-Services dezentral, d.h. in jedem Teilbus, angeboten werden. Für Software-Services, die eine gemeinsame Datenbasis benötigen (in der obigen Aufzählung durch ‚Z‘ gekennzeichnet), ist dann eine Föderierung der Daten vorzusehen.

Für das *Routing* von Ereignismeldungen gibt es drei verschiedene Möglichkeiten: Existiert genau einen Empfänger und ist dieser bekannt, dann wird er durch den Absender *direkt adressiert* (z.B. Volkszählungsinformationen an das Bundesamt für Statistik). Gesetzliche Vorgaben zu den erlaubten Empfängern einer Ereignismeldung (z. B. Datenschutz, wie bei den Zivilstandsereignissen) sind als *ereignistypabhängige Routing-Regeln* implementiert, welche den/die Empfänger ermitteln. Schließlich werden Ereignismeldungen, die hinsichtlich des Datenschutzes nicht kritisch sind, über *Publish/Subscribe-Mechanismen* verteilt (z.B. die Informationen des Bundesamtes für Statistik über neue oder geänderte Einträge im Gemeindeverzeichnis).

Mit Ausnahme von zwei Ereignistypen (Geschlechtsumwandlung, Adoption) sind alle 39 erkannten implementiert. Zivilstandsereignisse (u.a. Geburt, Tod, Eheschließung, Trennung, Scheidung) haben ca. 1500 Anbieter und mehr als 2000 Nutzer. Die Nutzeranzahl von Einwohnerkontrollereignissen wie Zuzug, Wegzug und Umzug (ca. 1500 Anbieter) ist nicht bekannt. Die jeweiligen Fachservices haben genau eine Operation, nämlich ‚Sende Ereignismeldung‘. Das ‚Senden‘ geschieht dabei teilweise implizit durch

das Schreiben der Ereignismeldung in eine Datei, da nicht alle Anwendungen technisch in der Lage sind, WSDL [W3C01] zu realisieren.

Die EBS-Services sind mit der WSDL spezifiziert. Gegenwärtig hat ein EBS-Service (d.h., ein WSDL-Interface) genau eine Operation.

5 Generalisiertes Vorgehensmodell für den Service-Entwurf in SOA

Die Umsetzung der E-Government-Strategie verlangt die Einführung von SOA auf allen Ebenen und in allen Organisationseinheiten der öffentlichen Verwaltung. Deshalb ist es sinnvoll, die bei Swisstopo und im Personenmeldewesen gemachten Erfahrungen zu generalisieren und ein allgemeines Vorgehensmodell für den Entwurf von SOA in öffentlichen Verwaltungen abzuleiten (s. Abbildung 4). *Anwendbar* ist das Vorgehensmodell, wenn (1) viele (Verwaltungs-) Prozesse in unterschiedlichen Varianten vorliegen, (2) die Software-Services viele und autonome Stakeholder haben, (3) die SOA-Rollen auf der fachlichen Seite (Geschäftsprozesssicht) Service-Eigner und –Nutzer unterscheiden sowie auf der technischen Seite (IT-Sicht) Service-Betreiber und Anwendungsbetreiber und (4) eine große, im Detail unbekannte Anzahl potenzieller SOA-Nutzer existiert. Die SOA-Nutzer profitieren von den generalisierten Software-Services, die das Vorgehensmodell (s. Abbildung 4) für das SOA-Design hervorbringt.

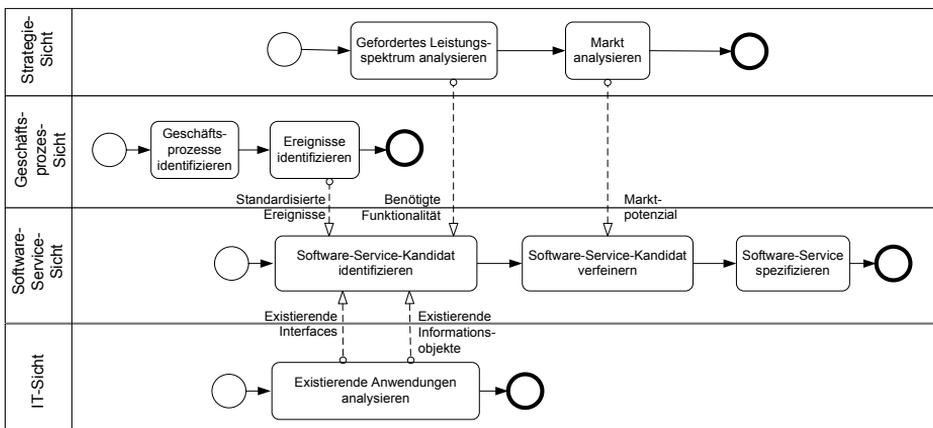


Abbildung 4: Generalisiertes Vorgehensmodell für das SOA-Design (in BPMN [OMG09])

Insgesamt ist ein *hybrides Vorgehen* (d.h. top-down und bottom-up) sinnvoll, vor allem dann, wenn Anwendungen existieren, deren Funktionalität unter SOA weiter genutzt werden soll. Zur Beherrschung der Komplexität von Anwendungskontext und Systemlandschaft (d.h. der Anzahl und der Heterogenität der beteiligten Objekte) empfiehlt sich ein dreistufiges Vorgehen, in dem die initial definierten Software-Services in einem zweiten Schritt geprüft und verfeinert werden, bevor die Spezifikation beginnt.

Die *Top-down-Identifikation* der Software-Services kann ausgehen von den Ereignissen, die vor, während oder nach einem Prozess auftreten, oder an den Leistungen (Capabilities) ansetzen, die eine Verwaltungseinheit ihren Kunden zur Verfügung stellen soll. Vor dem föderalistischen Hintergrund der Schweiz war eine direkte Ableitung von Software-

Services aus Prozessen nicht möglich, da die Prozesse entweder zu heterogen oder zu generisch waren (s. Abschnitt 4.1). Vorhandene Prozessmodelle sollten jedoch als Wissensquelle genutzt werden, um Ereignisse abzuleiten. *Ereignisse* sind vorteilhaft, da sie einen schnellen Konsens zwischen den SOA-Stakeholdern ermöglichen, stabil bleiben (technisch und fachlich) und gleichzeitig den Organisationseinheiten genügend Autonomie für die Umsetzung lassen. Zudem sind ereignisgetriebene Architekturen ein bekanntes technologisches Mittel für die Integration in heterogenen Systemwelten [NG05], [Ta+09]. Begleitende Analysen der geforderten *Verwaltungsleistungen* (Capabilities) sichern die Vollständigkeit der SOA (auch im Sinne der E-Government-Strategie).

Die *Bottom-up-Identifikation* der Software-Services sollte *existierende Schnittstellen* von Anwendungen beibehalten, um deren Kunden auch nach Umstellung auf SOA bedienen zu können. Außerdem sind existierende *Informationsobjekte* eine Quelle von Software-Services: Zum einen ist die Informationsbereitstellung eine zentrale Aufgabe der Verwaltung (eigene Unterkategorie im Prozessinventar). Zum anderen weisen CRUD-Operationen (Create, Retrieve, Update, Delete) für Informationsobjekte auf nötige Fachservices hin. Konkret wurde im Rahmen des Software-Service-Designs geprüft, ob alle Ereignisse berücksichtigt wurden, die derartige Operationen auf Daten im Einwohner- oder Zivilstandsregister zur Folge hatten.

Die anschließende *Verfeinerung* bewirkt, dass nicht jeder identifizierte Software-Service implementiert wird. Typische Verfeinerungskriterien sind: (1) die Existenz von *Kunden* für den Software-Service – um die Kosten für die Bereitstellung des Software-Service zu decken und (2) die *Atomarität* der Ereignismeldung. Letzteres führt dazu, dass kombiniert auftretende Ereignisse (z.B. ‚Heirat‘) in atomare zerlegt werden (‚Eheschließung‘, ‚Namensänderung‘ und ‚Einbürgerung‘). Die kleineren Ereignismeldungen fördern die Wiederverwendung, erleichtern das Routing und kommen dem Datenschutz entgegen.

Die *Software-Service-Spezifikation* definiert das Interface des Software-Service, d.h. die darin enthaltenen Operationen, deren Signaturen und Konversationen. Basieren kann die Spezifikation auf REST ([Fi00]; bei Swisstopo) oder auf WSDL ([W3C01]; im Personenmeldewesen). REST ist anwendbar, wenn die Funktionalität des SoftwareService darin besteht, Informationen bereitzustellen – damit also keine transaktionalen Anforderungen existieren – und wenn REST als Standard von den service-konsumierenden Anwendungen genutzt wird⁶. WSDL hingegen ist zu bevorzugen, wenn erhöhte Anforderungen hinsichtlich transaktionaler Verarbeitung, Sicherheit und Datenschutz vorliegen und wenn die Service-Nutzer technologisch heterogen sind (bessere Verfügbarkeit von Werkzeugen) [PZL08].

6 Abschließende Beurteilung

Dem Wesen nach bündelt SOA Funktionalität in Software-Services (s. Abschnitt 2). Objekt des Software-Designs ist damit primär die *Operation* und sekundär (implementierungsabhängig; z.B. nicht für REST) das *Interface* als Zusammenfassung von Operationen. Hingegen beruht das Design der SOA im schweizerischen Personenmeldewesen auf den Prinzipien *ereignisgetriebener Architektur (EDA)* [NG05], [Ta+09]: Zentrales Design-Objekt ist in diesem Fall das *Ereignis* als etwas, das in der fachlichen Welt ‚pas-

⁶ Im Falle von Swisstopo ist der WMS eine direkte Implementierung des WMS-Standards der OGC [OGC04], der von allen GIS genutzt wird.

siert'. Ereignisse werden durch Produzenten erzeugt und durch Konsumenten (reagierend) verarbeitet. Eine entsprechende Infrastruktur (hier der EBS) ermöglicht die Kommunikation zwischen Ereignis-Produzenten und –Konsumenten. Bei der EDA-Umsetzung durch SOA garantieren Ereignisse (aufgrund ihrer für Experten überschaubaren Anzahl) eine vollständige Menge atomarer und wiederverwendbarer Fachservices.

Die Besonderheit der ereignisorientierten Architektur besteht darin, dass der Produzent nur das Ereignis kennt, das er hervorbringt, aber weder die Konsumenten noch die Art der Ereignisverarbeitung (die ‚Operation‘). Analog gilt dies für die Ereignis-Konsumenten. Für das schweizerische Personenmeldewesen ermöglichte der EDA-Architekturstil *Autonomie* (die beteiligten Organisationseinheiten können Ereignisse unabhängig voneinander, über mehrere Stufen oder parallel verarbeiten), die *Überbrückung der Heterogenität* (im Sinne von Abschnitt 2) durch fachliche, stabile sowie allgemeingültige Ereignisse und *Flexibilität*. So lassen sich Prozesse allein durch eine Neukonfiguration der Empfänger von Ereignismeldungen (alternative Routing-Regeln) ändern.

Für die angeschlossenen Organisationseinheiten senkt die Infrastruktur der ereignisgetriebenen Architektur, der Event-Bus Schweiz, die Kosten von Anwendungsentwicklung und –betrieb: Die Anwendungen bleiben einfach, denn sie benötigen keine komplexe Logik zur Gewährleistung der sicheren und vertrauenswürdigen Kommunikation mit den Partnern. Ereignis-Produzenten und -konsumenten müssen nur die Schnittstelle zum lokalen Betreiber pflegen und ein einziges Service Level Agreement aushandeln.

Der EBS selbst ist *skalierbar*, d. h., er kann Schritt für Schritt ausgebaut werden (durch das Einbinden neuer Teilbusse, EBS-Services oder Endsysteme), ohne dass Ereignisproduzenten und -Konsumenten davon wissen müssen. Als *Nachteil* bildet der EBS eine zusätzlich Infrastruktur, deren Bereitstellung Kosten verursacht. ‚Naive‘ Interaktion zwischen Web-Services (ohne erhöhte transaktionale oder Sicherheitsanforderungen) könnte auf eine derartige Infrastruktur verzichten.

Die ereignisgetriebene Architektur steht für asynchrone Interaktion. Sie sollte nicht eingesetzt werden, wo fachliche Anforderungen synchrone Interaktion erfordern, also die direkte Antwort auf eine Anfrage. Beispielsweise wird der EBS umgangen, wenn AHV-Nummern abgefragt werden müssen - dies wurde als Kommunikation im Request-Response-Stil zwischen Web-Services auf der Basis von WSDL realisiert.

Speziell zur Unterstützung der organisationsübergreifenden Prozessabwicklung in einem stark föderalistisch geprägten Umfeld ist eine ereignisorientierte Architektur eine ideale Ergänzung zu klassischen SOA-Ansätzen.

Literaturverzeichnis

- [Ar+08] Arsanjani, A. et al: A method for developing service-oriented solutions. In: IBM Systems Journal 47 (2008) 2, p. 377 396
- [eCH-0020] eCH-0020: Datenstandard Meldegründe, Version 2.0, genehmigt 2009-04-22. http://www.ech.ch/index.php?option=com_docman&task=cat_view&gid=128&lang=de
- [Er08a] Erl, T.: SOA Principles of Service Design. Prentice Hall, Upper Saddle River et al., 2008
- [Er08b] Erl, T.: SOA Design Patterns. Prentice Hall, Upper Saddle River et al., 2008
- [Fi00] Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. Dissertation. University of California, Irvine, 2000 <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

- [GS07] E-Government-Strategie Schweiz, vom Bundesrat verabschiedet am 24. Januar 2007 <http://www.egovernment.ch/de/grundlagen/index.php>
- [KSR04] Kaabi, R.S.; Souveyet, C.; Rolland, C.: Eliciting service composition in a goal driven manner. In Aiello, M. et al. (Eds.): Proc. of the 2nd Int. Conf. on Service Oriented Computing ICSOC 2004, New York, ACM Press, 2004, pp. 308-305.
- [KKB07] Klose, K.; Knackstedt, R.; Beverungen, D.: Identification of Services - A Stakeholder-based Approach to SOA development and its application in the area of production planning. In Österle, H. et al. (Eds.): Proc. of the 15th European Conf. on Information Systems ECIS 2007. St. Gallen 2008, pp. 1802-1814
- [LEG08] Lämmer, A., Eggert, S., Gronau, N.: A Procedure Model for SOA-Based Integration of Enterprise Systems. In: International Journal of Enterprise Information Systems 4 (2008) 2, pp. 1-12
- [NG05] Niblett, P.; Graham, S.: Events and service-oriented architecture: The OASIS Web Services Notification specifications. In: IBM Systems Journal 44 (2005) 4, pp. 869-886
- [MEE04] Müller, W.; Erdmann, L.; Eyholzer, K.: eCH: Inventar der E-Government-relevanten Prozesse. In: eGov-Präsenz 4 (2004) 1, S. 10-12
- [OGC04] Open GIS Consortium Inc. (OGC): OGC Web Map Service Interface. Reference No. OGC 03-109r1, 2004-01-20 <http://www.opengeospatial.org/standards/wms>
- [OMG09] Object Management Group: Business Process Modeling Notation (BPMN), Version 1.2. OMG Document No.: formal/2009-01-03 <http://www.omg.org/spec/BPMN/1.2/PDF/>
- [Pa09] Patig, S.: Cases of Software-Services Design in Practice. In: Shishkov, B. et al. (Eds.): Proc. of the 4th Int. Conference on Software and Data Technologies ICSOFT 2009. Setubal, INSTICC, 2009, pp. 376-383
- [PH06] Papazoglou, M.P.; van den Heuvel, W.-J.: Service-oriented design and development methodology. In: International Journal of Web Engineering and Technology, 2 (2006) 4, pp. 412-442.
- [PZL08] Pautasso, C.; Zimmermann, O.; Leymann, F.: RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision. In: Proc. of the 17th Int. Conf. on the World Wide Web (WWW 2008), April 21-25, 2008, Beijing, China, pp. 805-814.
- [S007] S007 - Teilstrategie SOA Bund 2008-2012, Version 1.0 vom 24. September 2007 <http://www.isb.admin.ch/themen/strategien/00745/00748/index.html?lang=de>
- [Ta+09] Taylor, H. et al.: Even-driven Architecture – How SOA Enables the Real-Time Enterprise. Addison-Wesley, Upper Saddle River et al., 2009
- [W3C01] W3C. Web Services Description Language (WSDL) 1.1. WC Note 15 March 2001 <http://www.w3.org/TR/wsdl>