



SOA Prinzipien

Klassifizierung *	Nicht klassifiziert / Intern / Vertraulich
Status **	In Arbeit / In Prüfung / Abgeschlossen
Projektname	SOA-Voraussetzungen - SOA-Vorgaben
Projektabkürzung	SOVO - Service-Verzeichnis
Projektnummer	7766
Projektleiter	willy.mueller@isb.admin.ch
Auftraggeber	peter.fischer@isb.admin.ch
Autor	willy.mueller@isb.admin.ch
Initiale	muw
Bearbeitende	muw
Prüfende	Wolfgang Tietz; Philipp Hoernes
Genehmigende	Projektausschuss
Verteiler	Projektmitarbeiter, Projektausschuss

* Nicht klassifiziert, Intern, Vertraulich

** In Arbeit, In Prüfung, Abgeschlossen

Änderungskontrolle, Prüfung, Genehmigung

Version	Datum	Beschreibung, Bemerkung	Name oder Rolle
1.0	2008-06-08	Zusammenstellung bestehender Dokumentation des ISCeco	Willy Müller

«Die Projektführungsmethode HERMES ist ein offener Standard der schweizerischen Bundesverwaltung. HERMES wird vom Informatikstrategieorgan Bund (ISB) herausgegeben. Inhaberin der Urheberrechte an HERMES und der Markenrechte am HERMES-Logo ist die Schweizerische Eidgenossenschaft, vertreten durch das ISB.»

Definitionen, Akronyme und Abkürzungen

Begriff / Abkürzung	Bedeutung
SOVO	Projekt SOA-Voraussetzungen
TOGAF	The Open Group Architecture Framework
SOA	Service Orientierte Architektur

Referenzen

Erkennungszeichen	Titel, Quelle

Inhaltsverzeichnis

1	Zweck des Dokuments	4
2	SOA Prinzipien	4
2.1	Business Relevanz	4
2.2	Abstraktion	4
2.3	Wiederverwendbarkeit.....	4
2.4	Lose Kopplung (durch Abstraktion der Implementierung).....	5
2.5	Formelle Beschreibung und Kontrakt.....	5
2.6	Auffindbarkeit.....	5
2.7	Zusammensetzbarkeit.....	6
2.8	Eigenständiger Lifecycle	6
2.9	Statuslosigkeit	6
2.10	Autonomie	7
2.11	Technologie-Neutralität durch Standardisierung	7

Abbildungsverzeichnis

1 Zweck des Dokuments

Dieses Dokument dient als Referenz für die grundlegenden SOA-Prinzipien. Sowohl die Referenzarchitektur als auch die Policies der SOA Governance bauen auf den Prinzipien auf und konkretisieren diese.

2 SOA Prinzipien

Unter SOA **Prinzipien** verstehen wir grundlegende Eigenschaften, die auf jede Implementierung einer SOA unabhängig von der gewählten Technologie und auch unabhängig von der implementierenden Organisation zutreffen. Die Einhaltung der Prinzipien garantiert, dass eine SOA auch den erhofften Nutzen in der Praxis generieren kann.

1. Business Relevanz von Services
2. Abstraktion der technischen Details durch Services
3. Wiederverwendbarkeit von Services
4. Lose Kopplung bei der Nutzung von Services
5. Formelle Beschreibung (bzw. Kontraktierung) der Services
6. Auffindbarkeit in einem Verzeichnis
7. Zusammensetzbarkeit (mehrer Services zu Komplexeren)
8. Eigenständiger Service-Lifecycle
9. Statuslosigkeit
10. Autonomie
11. Technologie-Unabhängigkeit durch Standardisierung

2.1 Business Relevanz

Bezeichnung	Business Relevanz
Beschreibung	Jeder (Business-) Service muss einen für die Fachseiten verständliche und für das Geschäft nutzbringende Funktionalität haben ¹ .
Nutzen	<ul style="list-style-type: none">• Der Service deckt tatsächlich benötigte Funktionalität ab, die ggf. im Kontext mehrerer Prozesse verwendet werden kann (Wiederverwendung)• Der Service ist für die Fachseite verständlich, es kann also auf der Basis von Services (und ohne Kenntnis technischer Details) über die Funktionalität von Software entschieden werden

2.2 Abstraktion

Bezeichnung	Abstraktion
Beschreibung	Ein Service abstrahiert die zugrundeliegende Implementierung; er ist ohne detailliertes Wissen über dieselbe nutzbar.
Nutzen	<ul style="list-style-type: none">• "Lose Kopplung" wird erst möglich• Trennung von technischen und fachlichen Themen• Langfristige Flexibilität

2.3 Wiederverwendbarkeit

Bezeichnung	Wiederverwendbarkeit
Beschreibung	Beim Service-Design ist darauf zu achten, dass in Zukunft andere als die ursprünglichen Service-Consumer den gleichen Service wiederverwenden können. Ein Business-Service soll also von verschiedenen Service-Consumern parallel im

¹ Die gilt direkt nur für sog. Business-Services. Technische Services zur Unterstützung der Plattform haben keine geschäftliche Funktionalität.

Kontext verschiedener Businessprozessen verwendet werden können.

- Nutzen
- Zeit (Anforderungen schneller erfüllen) und Geld (geringere Kosten durch Wiederverwendung)
 - Zukünftige Anforderungen mit jetzigem Service adressieren und dadurch Entwicklungsaufwand reduzieren
 - Redundante Logik abbauen, bzw. nicht entstehen lassen

2.4 Lose Kopplung (durch Abstraktion der Implementierung)

- Bezeichnung **Lose Kopplung (durch Abstraktion der Implementierung)**
- Beschreibung Die Services sind so zu programmieren, sind so zu entwickeln, dass Provider und Consumer nur die Service-Beschreibung bzw. die Kenntnis des Service-Interfaces brauchen, um miteinander interagieren zu können. Service-Interfaces abstrahieren und verbergen die dem Service unterliegende implementierte Logik so, dass sie für den Service-Consumer und die Service-Kommunikation irrelevant sind. Ein Redeployment oder ein Austausch der Implementierung eines Providers oder Consumers hat dann keine Auswirkungen auf die Implementierung des jeweiligen Gegenstücks, solange das Interface sich nicht ändert.
- Nutzen
- Business kann sich auch eine technologie-unabhängige Schicht beziehen
 - Schneller und flexibler Austausch von Services möglich, ohne das Consumer geändert werden müssen
 - Flexible Anpassung der Infrastruktur möglich
 - Eine SOA kann über die Konfiguration von Metadaten (Beschreibung der Services) gesteuert werden können

2.5 Formelle Beschreibung und Kontrakt

- Bezeichnung **Formelle Beschreibung und Kontrakt**
- Beschreibung Die Funktionalität eines Services wird in einer Service Beschreibung formell beschrieben. Ein Service Kontrakt beschreibt und regelt darüber hinaus die Beziehung zwischen Service Providern und jedem individuellen Service Consumer. Ein Service Kontrakt bezieht sich auf eine Service Beschreibung und legt auch die geforderten nicht-funktionalen Eigenschaften der Servicenutzung durch den Consumer verbindlich fest.
- Nutzen
- Garantierte Service Provision (Bereitstellung der Leistung) für jeden Consumer im Rahmen der vertraglichen Vereinbarungen
 - Agilität, weil ein neuer Consumer sehr schnell und flexibel einen existierenden und umfassend vertraglich beschriebenen Service benutzen kann
 - Wichtiger Beitrag zur IT-Governance

2.6 Auffindbarkeit

- Bezeichnung Auffindbarkeit
- Beschreibung Services sind so entworfen, beschrieben und abgelegt, dass die Service-Beschreibungen und Service-Kontrakte in einem Verzeichnis wieder auffindbar sind und somit eine → Wiederverwendbarkeit praktisch ermöglicht wird.
- Nutzen
- Gute Service-Beschreibung verhindert langes Suchen
 - Ermöglichen von → Wiederverwendbarkeit
 - Gutes Service-Design ermöglicht eine präzise Beschreibung des Services in einem Service Contract

- Zeitersparnis zu Designzeit
- Nutzen von → Wiederverwendbarkeit

2.7 Zusammensetzbarkeit

Bezeichnung	Zusammensetzbarkeit
Beschreibung	Business-Prozesse, die nicht manuellen Tätigkeiten entsprechen, können durch bei deren Automatisierung auf Business Service(s) zurückgreifen. Zu diesem Zweck können höhere Funktionen durch die Zusammensetzung ² einfacherer Services realisiert werden.
Nutzen	<ul style="list-style-type: none">• Services sollen so designed werden, dass sie einzelnen oder zusammengesetzt in mehreren Business Prozessen verwendet werden können und damit der → Service Wiederverwendung dienen• Kürzere Implementationszeiten, geringere Kosten durch Wiederverwendung• Bessere Wartbarkeit

2.8 Eigenständiger Lifecycle

Bezeichnung	Eigenständiger Lifecycle
Beschreibung	Die Entwicklung und Weiterentwicklung von Services erfolgt unabhängig von dem oder den implementierenden Systemen. Insbesondere muss jeder Service einer eigenständigen Versionierung unterworfen werden.
Nutzen	<ul style="list-style-type: none">• Unterstützung von Abstraktion und loser Kopplung• Entkopplung der Lebenszyklen von Service Provider und Service Consumer, Vermeidung von harten Abhängigkeiten• Management von Komplexität

2.9 Statuslosigkeit

Bezeichnung	Statuslosigkeit
Beschreibung	Services sind so zu entwickeln, dass sie „stateless“ sind, also keine Informationen über die Kommunikation mit ihren Consumern speichern.
Nutzen	<ul style="list-style-type: none">• Komplexitätsreduktion• Unterstützt → Service Wiederverwendung• Ein ‚stateless‘ Service muss nichts über den Business Prozess wissen, in den er eingebunden ist. Das unterstützt → lose Koppelung und auch → Wieder-verwendung• Das Verwalten von State-Information ist eine aufwendige und Ressourcen-intensive Aufgabe; sie erschwert die Skalierbarkeit und Verfügbarkeit des Services massiv

Dieses Prinzip kann grundsätzlich von der Forderung nach Zusammensetzbarkeit und der losen Kopplung abgeleitet werden, da im Allgemeinen eine Zusammensetzbarkeit nicht erfolgen kann, wenn ein Service explizit vom Status seiner Kommunikation mit einem Consumer abhängig ist. Trotzdem führen wir hier das Prinzip separat auf.

² Wir verwenden hier das neutrale Wort „Zusammensetzung“, das die verschiedenen technischen Optionen der Orchestrierung, der Orchestrierung und ggf. auch der Komposition (im Sinne von SCA) einschliesst.

2.10 Autonomie

Bezeichnung	Autonomie
Beschreibung	Services sind so zu entwickeln, dass sie möglichst autonom, d.h. unabhängig voneinander sind. Services sollen eine maximale Kontrolle über ihre Geschäftslogik haben. Die Grenzen der Geschäftslogik, die von einem Service kontrolliert werden, sollen möglichst genau definiert sein.
Nutzen	<ul style="list-style-type: none">• Service Autonomie unterstützt → lose Koppelung und auch → Wiederverwendung

Dieses Prinzip kann grundsätzlich von der Forderung nach Zusammensetzbarkeit und der losen Kopplung abgeleitet werden, da im Allgemeinen eine Zusammensetzbarkeit nicht oder nur sehr schwer erfolgen kann, wenn explizite Abhängigkeiten zwischen Services bestehen (etwa in Form von Service-übergreifenden Transaktionen). Trotzdem führen wir hier das Prinzip separat auf.

2.11 Technologie-Neutralität durch Standardisierung

Bezeichnung	Technologie-Neutralität durch Standardisierung
Beschreibung	Bei der Implementierung der SOA sind Standards zwingend einzuhalten, die unabhängig von der jeweils gewählten Technologie die Interoperabilität von Services garantieren.
Nutzen	<ul style="list-style-type: none">• Optimierung der TCO• Zeitersparnis in der Planungs- und Entwicklungsphase• Erhöhen der Unabhängigkeit zu proprietären Technologien• Zeitersparnis in der Planungs- und Entwicklungsphase, da nicht nach Technologien gesucht werden muss• Einsatz von weltweit gültigen und standardisierten Technologien• Zugriff auf weltweites Know How

Dieses Prinzip kann grundsätzlich von der Forderung nach der losen Kopplung abgeleitet werden, die Standardisierung letztlich eine Umsetzung der losen Kopplung darstellt. Trotzdem führen wir aufgrund der Wichtigkeit der Standards hier das Prinzip separat auf.