

23 Serviceorientierte Architekturen

Rainer Berbner, Karsten Meister, Tanja Schmedes, Stefan Schulte

Serviceorientierte Architekturen (SOA) haben sowohl bei der Integration von Legacy-Systemen als auch bei der Einbindung von Geschäftspartnern zunehmend an Einfluss gewonnen. Eine SOA basiert auf Services als Grundbausteinen. Unter einem Service wird in diesem Zusammenhang eine in sich abgeschlossene Software-Komponente verstanden, die eine wohldefinierte Funktionalität anbietet [Pap03, SJB04]. Das SOA-Paradigma ermöglicht die Überwindung interner Heterogenitäten bzgl. verschiedener Plattformen und Betriebssysteme im Unternehmen sowie die Kopplung mit den unterschiedlichsten IT-Systemen der Geschäftspartner. Die Komposition von Services stellt eine Möglichkeit dar, flexible, modular aufgebaute Anwendungssysteme und Geschäftsprozesse zu entwickeln, und führt zu einer prozessorientierten Ausrichtung der Anwendungslandschaft. Durch die Komposition von Services zu Workflows können flexible Produkte und Dienstleistungen nach dem Baukastenprinzip erstellt werden. Aufgrund der losen Kopplung der beteiligten Services ist es möglich, auch Services externer Anbieter zu verwenden. Dies ermöglicht die Realisierung unternehmensübergreifender Workflows basierend auf Services.

In diesem Kapitel wird anhand zweier Anwendungsszenarien der konkrete Einsatz einer SOA und die Umsetzung der in Kapitel 6 vorgestellten SOA-Konzepte in der Praxis vorgestellt. Hierzu wird zunächst in Abschnitt 23.1 anhand eines Anwendungsszenarios aus dem Bereich Energiemanagement gezeigt, welche Anforderungen an SOA in betrieblichen Umgebungen tatsächlich gestellt werden und welche Vorteile die Entwicklung einer SOA mit sich bringen. Somit wird im ersten Teil dieses Kapitels eine domänenspezifische Beispielimplementierung vorgestellt, während der zweite Teil dieses Kapitels auf das komplexe Anwendungsgebiet des Servicemanagements fokussiert. In Abschnitt 23.2 wird dazu ein lebenszyklusbasierter Ansatz für das Servicemanagement beschrieben, der das umfassende Management der Services und der aus Services zusammengesetzten Geschäftsprozesse ermöglicht. Die zur Umsetzung dieses Ansatzes benötigten Architekturkomponenten werden eingeführt und deren Funktionalität beschrieben.

23.1 DEMS – ein SOA-Anwendungsszenario Energiemanagement

Die Energieversorgung wird mit immer mehr gesetzlichen, organisatorischen und gesellschaftspolitischen Anforderungen sowie technologischen Innovationen konfrontiert. Um diesen Rahmenbedingungen angemessen zu begegnen, müssen Energieversorgungsunternehmen (EVU) ihre Geschäftslösungen neu konzipieren und ihre bestehende IT-Landschaft umgestalten [USLA05]. So erfordert beispielsweise die Durchführung eines stabilen, sicheren und diskriminierungsfreien Energiemanagements nicht nur die elektrische Integration neuer Erzeuger und Verbraucher in die bestehenden Energienetze, sondern auch die technische Integration dieser Erzeuger und Verbraucher über ihre Schnittstellen in die bestehende IT-Landschaft. Die strukturellen Veränderungen in der Energieversorgung erfordern zudem neue Geschäftsprozesse und Funktionalitäten für das Energiemanagement der Zukunft.

Unter dem Schlagwort DEMS werden in einem Forschungsprojekt verschiedener Forschungseinrichtungen wie dem OFFIS, dem Systemhaus BTC und dem Energieversorgungsunternehmen EWE u. a. diese neuen Prozesse, Funktionalitäten und Systeme untersucht und entwickelt. DEMS steht hierbei für »dezentrales Energiemanagement-System« und zeigt insbesondere den Fokus des Forschungsprojekts hinsichtlich der Integration einer Vielzahl kleiner, verteilter Erzeuger und Verbraucher auf. Zudem verdeutlicht das »dezentral« auch den Bedarf an einem verteilten Energiemanagement sowie hierfür benötigter, unterstützender Systeme. Das Architekturkonzept des DEMS muss hierbei insbesondere die folgenden Rahmenbedingungen berücksichtigen:

- Integration vorhandener Systeme: Im Forschungsprojekt DEMS müssen vorhandene Altsysteme in eine heterogene IT-Landschaft des EVU integriert werden. Zudem muss das Forschungsprojekt eine Vielzahl kleiner, verteilter Systeme in Form von Erzeuger- und Verbraucherschnittstellen einbinden.
- Anbindung von Forschungspartnern: Im Forschungsprojekt sind Fachexperten aus verschiedenen Domänen an der Entwicklung innovativer Funktionalitäten für das dezentrale Energiemanagement beteiligt. Diese Fachexperten verwenden dabei bekannte und in ihrer Domäne bewährte Software und Programmiersprachen. Aufgrund des Forschungscharakters ist zudem die Qualität und Vollständigkeit der entwickelten Funktionalitäten offen.
- EVU-spezifische Aspekte: Das Forschungsprojekt muss EVU-spezifische Aspekte bei der Umsetzung beachten, um eine spätere Vermarktbarkeit des Systems zu gewährleisten. Diese betreffen beispielsweise die Trennung der bisher integrierten Daten der Sparten Netz und Vertrieb oder die Berücksichtigung autonomer Akteure.

Gefordert ist somit eine flexible, erweiterbare, effektive, programmiersprachen- und plattformunabhängige Systemarchitektur. In den folgenden Abschnitten wird am Beispiel des Forschungsprojekts DEMS vorgestellt, wie diesen Anforderungen durch die Entwicklung einer serviceorientierten Architektur (SOA) für das dezentrale Energiemanagement, der »DEMS-SOA«, begegnet wird. Durch das unterliegende SOA-Konzept werden die geforderten und erwarteten Eigenschaften der Systemarchitektur erfüllt (siehe auch Abschnitt 6.3.2).

23.1.1 Zielarchitektur DEMS-SOA

Die Ausgangsarchitektur des Forschungsprojektes besteht aus einer wenig strukturierten Sammlung von Alt- und Forschungssystemen. Diese müssten, um ein DEMS zu bilden, über eine Vielzahl zu erstellender Adapter miteinander verbunden werden, siehe Abbildung 23.1.

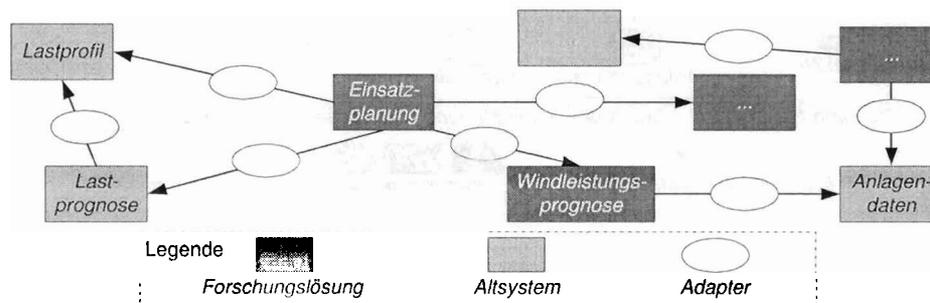


Abbildung 23.1: Ausgangsarchitektur DEMS

Diese Ausgangsarchitektur widerspricht offensichtlich den gestellten Anforderungen wie Flexibilität oder Erweiterbarkeit. Die in Abbildung 23.2 dargestellte Zielarchitektur als konkrete Instanz der in Abbildung 6.4, (S.133) vorgestellten Referenzarchitektur basiert daher auf SOA-Konzepten wie

- ❑ einem gemeinsamen Information Model auf Basis der Norm IEC 61970 [Com05], s. auch Abschnitt 23.1.2
- ❑ einer strukturierten Kapselung fachlicher Geschäftskomponenten als Dienste (Services) als Ergebnis einer Serviceidentifikation und -entwicklung (s. auch Abschnitt 23.1.3)
- ❑ einer ausgelagerten Ablaufsteuerung [Sie07] durch einen Enterprise Service Bus, (s. auch Abschnitt 23.1.4) sowie
- ❑ einer zentralen Verwaltung aller Services innerhalb eines Service-Repositories, (s. auch Abschnitt 23.1.5).

Zusammenfassend werden somit die fachlichen Geschäftskomponenten der

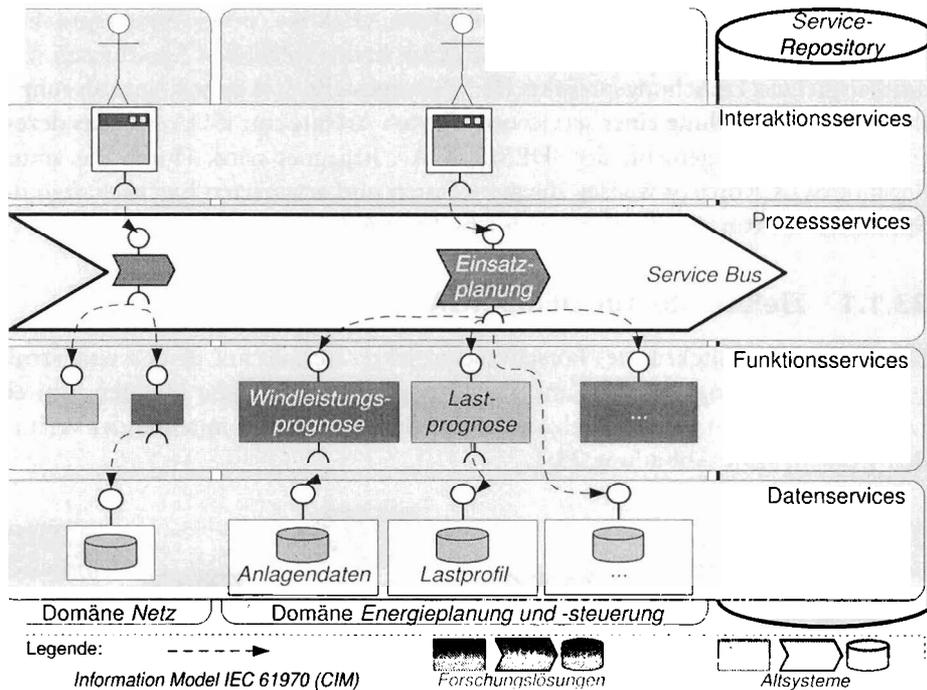


Abbildung 23.2: Zielarchitektur DEMS-SOA

Energiewirtschaft, die innerhalb des Forschungsprojektes entwickelt oder durch die Anbindung von Altsystemen zur Verfügung gestellt werden, nach außen gekapselt und über ein standardisiertes Format als DEMS-Services angeboten. Diese Services können dann von verschiedenen Akteuren des Energiemarktes über das Service-Repository für ihre jeweiligen Bedürfnisse genutzt werden.

23.1.2 Information Model CIM

Die DEMS-SOA strebt nicht nur die Integration vorhandener Altsysteme und zu entwickelnder DEMS-Lösungen der Forschungspartner an, sondern wird vielmehr im Sinne einer späteren Marktreife entwickelt. Hierfür wird insbesondere ein gemeinsames Information Model benötigt, das sowohl die semantische als auch die syntaktische Struktur der auszutauschenden Nachrichten und deren Nachrichtenobjekte festlegt und die Interoperabilität in der Energiebranche sichert [UG07]. Somit ist für die DEMS-SOA festzulegen, was die Services unter welcher inhaltlichen Bedeutung wie austauschen.

In der Energiewirtschaft existiert hierfür bereits durch die IEC-Norm 61970 das sogenannte Common Information Model CIM [Com02]. Dieses Information

Model umfasst sowohl Datenobjekte aus dem technischen (wie Erzeuger, Stationen oder Schalter) als auch Datenobjekte aus dem kaufmännischen Bereich (wie Kunde oder Dokument) und modelliert diese als UML-Diagramme. Listing 23.1 stellt beispielhaft einen Ausschnitt aus einer auf CIM-basierenden Nachricht dar.

```
<?xml version="1.0" encoding="UTF-8"?>
<MesswertListe xmlns="MesswertListe" xmlns:mdimsg="mdimsg"
  >
  <MessageHeader>
    ...
    <mdimsg:Noun>MesswertListe </mdimsg:Noun>
    <mdimsg:Source>SCADA</mdimsg:Source>
  </MessageHeader>
  <MessagePayload>
    <Measurement>
      <aliasName>WG01_SW1</aliasName>
      <description>Stellung des Schalters WG 01 SW 1</
        description >
    </Measurement>
    <MeasurementType>
      ...
    </MeasurementType>
    <MeasurementValue>
      ...
      <timeStamp>20070709:1545</timeStamp>
      <value>closed </value >
    </MeasurementValue>
  </MessagePayload>
</MesswertListe >
```

Listing 23.1: Beispiel der Darstellung eines Messwerts in CIM

Durch die Verwendung des Information Model CIM kann die Interoperabilität der DEMS-SOA gewährleistet werden, da alle Services einem standardisierten, akzeptierten und verbreiteten Datenmodell unterliegen.

23.1.3 Serviceidentifikation und -entwicklung

Die Identifikation benötigter Services für ein DEMS kann sowohl »top-down«, ausgehend von den Soll-Geschäftsprozessen, als auch »bottom-up«, ausgehend von den vorhandenen Altsystemen, erfolgen (s. auch Abschnitt 6.4.2). Im Forschungsprojekt wurden dazu zunächst zukünftige Soll-Geschäftsprozesse für ein dezentrales Energiemanagement identifiziert und modelliert. Auf Basis dieser

Prozesse erfolgt eine Gruppierung und Strukturierung der Prozessentitäten zu fachlichen Services. So werden beispielsweise Prozessentitäten zur Prognose von Windenergieeinspeisungen zu einem Service »Windleistungsprognose« zusammengefasst.

Neben diesem »Top-down«-Vorgehen wurden zudem Services »bottom-up« durch Analyse der vorhandenen Altsysteme erfasst. Auch hier erfolgt wieder eine Gruppierung und Strukturierung anhand fachlicher Kriterien. Zur Serviceidentifikation siehe auch [HHV06].

Neben dieser fachlichen Strukturierung werden die DEMS-Services anhand der folgenden vier Kategorien nach [HHV06] eingeteilt (siehe Abschnitt 6.2):

- Datenservices ermöglichen den Zugriff auf benötigte Grunddaten für ein DEMS. Datenservices können von Funktions- und Prozessservices genutzt werden. Datenservices basieren i. d. R. auf den Altsystemen des EVU.
- Funktionsservices sind atomare Dienste zur Bereitstellung benötigter Funktionalitäten für ein DEMS. Funktionsservices nutzen Datenservices oder können selbst in Prozessservices genutzt werden. Funktionsservices werden sowohl durch die Forschungspartner entwickelt als auch durch die Altsysteme angeboten.
- Prozessservices bilden die orchestrierten Geschäftsprozesse eines DEMS ab. Prozessservices nutzen für die Erfüllung höherwertiger Geschäftsprozesse Daten- und Funktionsservices und können selbst wieder in anderen Prozessservices genutzt werden. Die Prozessservices für ein DEMS werden im Forschungsprojekt in Absprache mit den Fachansprechpartnern im EVU als Soll-Geschäftsprozesse identifiziert und neu entwickelt.
- Interaktionsservices dienen als Schnittstelle für den Anwender und greifen auf Daten-, Funktions- und Prozessservices zu. Interaktionsservices können sowohl in existierende Anwendungen und Oberflächen eines EVU integriert als auch komplett unabhängig entwickelt werden.

Neben den Servicekategorien ist insbesondere die kommunikationstopologische (kurz topologische) Position auf einer Hierarchieebene (Netz-, Teilnetz-, Anlagen- oder Geräteebene) relevant, auf der ein DEMS-Service angeboten wird. Denn jeder DEMS-Service kann auf einer dieser vier topologischen Hierarchieebenen angeboten werden, wobei die Positionierung insbesondere von den konkreten, zu erfüllenden nichtfunktionalen Anforderungen, den sogenannten Quality of Services (QoS, s. auch Abschnitt 23.2.1) abhängt, die ein Service zu erfüllen hat. Soll zum Beispiel ein Service »Anlagendaten« zur Bereitstellung von Erzeugerdaten lokal autonom durchgeführt werden können, muss er dezentral auf der sogenannten Anlagenebene implementiert werden. Wird hingegen gute Änderbarkeit das entscheidende Kriterium, müsste derselbe Service zentral auf der sogenannten Netzebene implementiert werden. Bei der Entwicklung einer SOA für das dezentrale Energiemanagement dient somit die hierarchische Struktur des

Energienetzes und die darauf aufsetzende Kommunikationstopologie als Basis des Architekturkonzepts [Sch07b]. Abbildung 23.3 stellt einen Ausschnitt des Metamodells der DEMS-SOA unter expliziter Betrachtung der beiden Aspekte »topologische Position« (links) und »Servicekategorie« (rechts) dar.

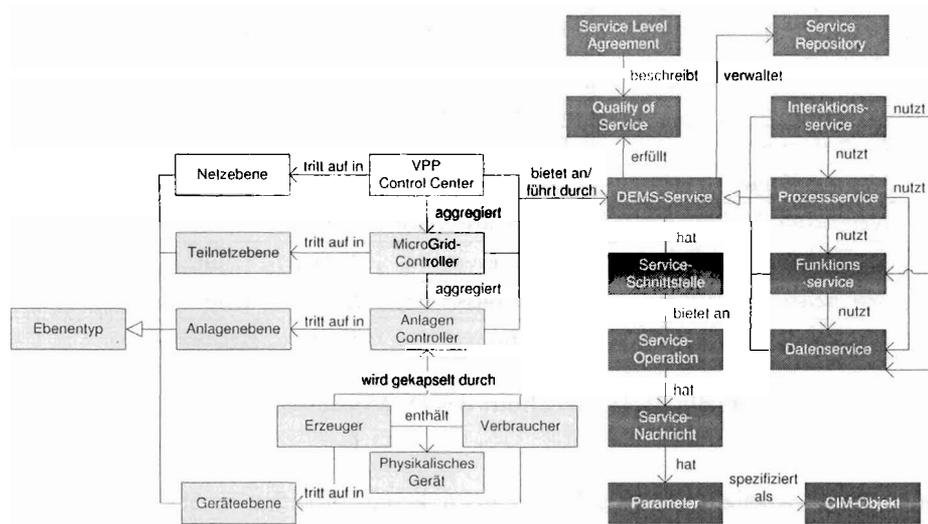


Abbildung 23.3: Ausschnitt des Metamodells der DEMS-SOA

Innerhalb des hier vorgestellten Forschungsprojektes werden verschiedene DEMS-Services identifiziert. Tabelle 23.1 stellt einen Ausschnitt dieser Services in Form eines DEMS-Servicekatalogs dar (siehe auch [LMW07]). Hierbei werden, wie auch bereits im Metamodell in Abbildung 23.3 dargestellt, zu jedem Service die beiden Aspekte »Servicekategorie« sowie »topologische Position« aufgeführt. Bezüglich der topologischen Position werden in der Tabelle die möglichen Hierarchieebenen angegeben, auf denen der Service angeboten werden kann. Die konkrete Allokation der Services auf die Hierarchieebenen erfolgt abhängig von den verfolgten QoS. Auf eine ausführliche Erläuterung der einzelnen Services muss an dieser Stelle aus Platzgründen verzichtet werden.

Die Umsetzung der Servicedefinitionen in konkrete lauffähige Services erfolgt ähnlich der Identifikation mit unterschiedlichen Vorgehensweisen und Werkzeugen. Die identifizierten Prozessservices werden zunächst grafisch mithilfe des Biz-Talk-Orchestration-Designers in einer an BPEL angelehnten Modellierungssprache modelliert (siehe auch Abschnitt 23.1.4). Auf diese Weise wurden in mehreren Orchestrierungen prototypisch die für das Forschungsprojekt relevanten Energiemanagementprozesse ausschließlich über Aufruf von Services implementiert. Dabei ist wichtig, dass die entwickelten Prozessservices nur prozessbezogene und keinerlei funktionale Logik enthalten. Da die Prozessservices projektbegleitend

<i>Service</i>	Servicekategorie	topologische Position
<i>Anlagendaten</i>	Datenservice	Anlagen-, Teilnetz-, Netzebene
<i>Einsatzplanung</i>	Prozessservice	Teilnetz-, Netzebene
<i>Erzeugerprognose</i>	Prozessservice	Anlagen-, Teilnetz-, Netzebene
<i>Fahrplanmanagement</i>	Prozessservice	Anlagen-, Teilnetz-, Netzebene
<i>Knotenlastprognose</i>	Funktionservice	Anlagen-, Teilnetz-, Netzebene
<i>Lastprofil</i>	Datenservice	Anlagen-, Teilnetz-, Netzebene
<i>Lastprognose</i>	Funktionservice	Anlagen-, Teilnetz-, Netzebene
<i>Netztopologie</i>	Datenservice	Teilnetz-, Netzebene
<i>Netzzustandsidentifikation</i>	Funktionservice	Teilnetz-, Netzebene
<i>Verbraucherdaten</i>	Datenservice	Anlagen-, Teilnetz-, Netzebene
<i>Windleistungsprognose</i>	Funktionservice	Anlagen-, Teilnetz-, Netzebene
<i>Wetterprognose</i>	Funktionservice	Teilnetz-, Netzebene

Tabelle 23.1: Ausschnitt des DEMS-Servicekatalogs

entwickelt wurden, standen häufig noch keine aufrufbaren Services der Forschungspartner zur Verfügung. In diesem Fall wurden zunächst »Platzhalter«-Services verwendet, die in ihrer Aufruf- und Parameterstruktur den späteren produktiven Services entsprechen.

Bei der Entwicklung der Funktions- und Datenservices werden Altsysteme des EVU und Komponenten der Forschungspartner durch sogenannte »Service-wrapper« für die DEMS-SOA nutzbar gemacht. Findet man auf den Seiten der Forschungspartner lediglich bedingt servicegeeignete oder monolithische Programme und Systeme, beispielsweise auf Basis von Skriptsprachen vor, besteht die Notwendigkeit der Entwicklung von Serviceadaptern. So wurde z. B. im Rahmen des Projektes ein allgemein verwendbarer MatLab-Adapter entwickelt, der die asynchrone Kommunikation mit MatLab-Programmen ermöglicht und so die Funktionalität von MatLab-Modulen als Funktionservices für DEMS zur Verfügung stellt.

23.1.4 ESB als Laufzeitumgebung

Eine zentrale Aufgabe einer SOA-Architektur für das dezentrale Energiemanagement ist die Integration der bei einem EVU bereits vorhandenen mit neu entwickelten und über Services gekapselten DEMS-Komponenten, die im Rahmen des Forschungsprojektes entstehen (siehe Abschnitt 23.1.3). Hierzu sowie zum Austausch der Daten zwischen den verschiedenen IT-Systemen und der prozessorientierten Nutzung der verschiedenen Services wird im Projekt DEMS das Konzept eines Enterprise Service Bus (ESB, siehe [KBS04]) verfolgt (siehe

Abbildung 23.4). Als zentrale Message-orientierte Middleware (MOM) für den ESB wird ein Microsoft BizTalk Server 2006 eingesetzt.

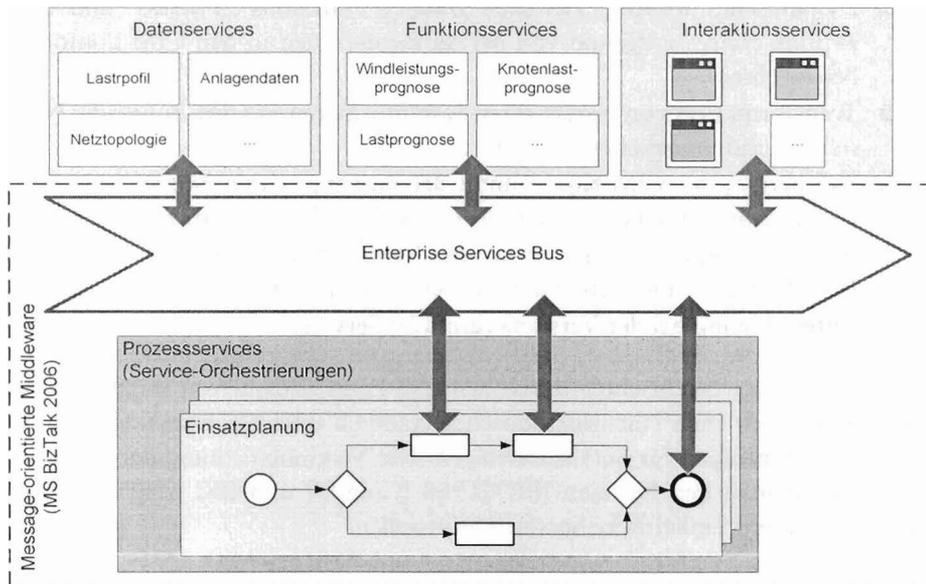


Abbildung 23.4: DEMS ESB

Die Funktionalität aller Komponenten des dezentralen Energiemanagements wird vollständig über Daten-, Funktions-, Prozess- und Interaktionsservices gekapselt (siehe auch Abschnitt 23.1.3). Der Datenaustausch zwischen den Komponenten der verschiedenen Architekturschichten des DEMS findet in Form von Serviceaufrufen statt, die nach dem »Hub-and-Spoke«-Prinzip über die zentrale Middleware (im Projekt als »DEMS-Server« bezeichnet) gesendet werden. Der DEMS-Server übernimmt die Rolle eines zentralen Brokers (»Hub«), der für die Zustellung der Nachrichten an die über Services integrierten Komponenten (»Spoke«) zuständig ist [Erl05].

Zudem dient er als Ablaufumgebung für die übergeordneten Energiemanagementprozesse (Prozessservices). Die Modellierung der übergeordneten Prozesse erfolgt mithilfe des BizTalk-Orchestration-Designers, der zum einen die grafische Modellierung von servicenutzenden Prozessen gestattet und zum anderen in der Lage ist, BPEL-konforme Prozessbeschreibungen zu importieren. Obwohl im Forschungsprojekt ein zentraler DEMS-Server zum Einsatz kommt, wird beim Design der Prozesse darauf geachtet, dass diese in einer vollständig dezentralen Systemarchitektur oder auch in einer föderierten Umgebung, d. h. in einem Verbund mehrerer Zentralsysteme lauffähig sind.

Neben der Bereitstellung einer MOM übernimmt der DEMS-Server weitere Schlüsselfunktionen wie:

- ❑ Dynamisches Routing zwischen Daten-, Funktions-, Prozess- und Interaktionservices aufgrund von Metainformationen in den CIM-konformen Nachrichten
- ❑ Transformation von proprietären Datenformaten in CIM-konforme Nachrichten und umgekehrt [USL⁺05]
- ❑ Validierung der CIM-Konformität der ausgetauschten Nachrichten durch Überprüfung anhand der hinterlegten XML-Schemainformationen
- ❑ Bereitstellung einer Registry für das Veröffentlichen von Services über in BizTalk eingebaute Deployment-Mechanismen sowie
- ❑ Unterstützung bei der Versionierung von Services.

Das Routing der Nachrichten durch den ESB kann anhand von Business Rules, durch den Inhalt einer Nachricht, durch Metadaten einer Nachricht oder anhand von UDDI-Einträgen für Services erfolgen. Die Vereinheitlichung der Nachrichteninhalte gemäß den Normen IEC 61968 [Com07] und IEC 61970 [Com05] ermöglicht dabei einheitliche Serviceschnittstellen.

23.1.5 Service-Repository und SOA-Management

Die im Forschungsprojekt DEMS entwickelten Services werden im Projektverlauf in marktreife Produktivservices überführt, um sowohl intern bei der EWE AG als auch bei anderen Akteuren, wie beispielsweise Stadtwerken oder anderen EVUs, für den Produktivbetrieb genutzt zu werden. Um den Anwendern die gezielte Suche nach, den Zugriff auf und die Nutzung dieser DEMS-Services zu ermöglichen, wird im Forschungsprojekt ein zentrales Service-Repository zur Verwaltung der DEMS-Services aufgebaut. Dieses Service-Repository stellt die Möglichkeiten bereit, die DEMS-Services zu erkennen und die zur Nutzung der DEMS-Services benötigten Informationen gezielt zu erfragen [KBS04]. Relevante Informationen, die in dem Service-Repository für DEMS bereitgestellt werden, sind beispielsweise (in Anlehnung an [KBS04] und [MLM⁺06]) die

- ❑ Beschreibung der Serviceschnittstellen, beispielsweise in Form einer WSDL,
- ❑ Beschreibung der Servicefunktionalität,
- ❑ Beschreibung der benötigten Zugriffsrechte wie Authentifizierung oder Verschlüsselung und
- ❑ Beschreibung der nicht funktionalen Eigenschaften des Service in Form von Qualität of Services (QoS).

Diese Informationen werden durch die Verwendung des gemeinsamen Informationsmodel CIM (siehe auch Abschnitt 23.1.2) erweitert.

Im weiteren Projektverlauf ist neben der Verwaltung der DEMS-Services in einem Service-Repository insbesondere das SOA-Management zur Gewährleistung einer unternehmensweiten bzw. -übergreifenden DEMS-Lösung von Relevanz. Das SOA-Management muss beispielsweise den Lebenszyklus der Services unterstützen und dokumentieren und die Versionierung der DEMS-Services erlauben [Lie07]. Dies ist insbesondere im vorgestellten Forschungsprojekt zu beachten, da Services schrittweise erweitert werden – von »Platzhalter«-Services (siehe Abschnitt 23.1.3) bis hin zu marktreifen Services. Zudem erfordert die dynamische Domäne der Energiewirtschaft aufgrund technischer oder gesetzlicher Neuerungen eine ständige Weiter- und Neuentwicklung der Dienste. Dies betrifft im verstärkten Maße die Funktions- und Prozessservices.

23.2 SOA-Anwendungsszenario und Servicemanagement aus einer technischen Perspektive

Wie am Szenario aus dem Bereich Energiemanagement (Abschnitt 23.1.5) deutlich geworden ist, stellt das Servicemanagement einen zentralen Erfolgsfaktor für den erfolgreichen Betrieb einer SOA dar. Daher wird in diesem Kapitel ein lebenszyklusbasierter Ansatz zum Servicemanagement vorgestellt. Hierfür werden Architekturkomponenten zur Umsetzung dieses Ansatzes eingeführt. Zuvor wird Dienstgüte als technischer Ansatzpunkt für das Servicemanagement vorgestellt.

23.2.1 Dienstgüte als Grundlage für das Servicemanagement

Dienstgüte im Rahmen von Netzwerken und Diensten, die Netzwerke benutzen, kennzeichnet das definierte, kontrollierbare Verhalten eines Systems bezüglich quantitativ messbarer Parameter [Sch01a]. Dienstgüte im Rahmen einer SOA beschreibt die nicht funktionalen Eigenschaften eines Service. Unternehmen werden nur dann bereit sein, Services externer Anbieter in unternehmenskritischen Workflows zu verwenden, wenn deren Dienstgüteeigenschaften bekannt sind bzw. sogar garantiert werden [Ber07, BL06]. Neben den Dienstgüteeigenschaften sind die Kosten einer Servicenutzung ein wichtiges Kriterium bei der Auswahl zwischen Services gleicher Funktionalität. Daher werden die Kosten auch als weitere eigenständige nicht funktionale Eigenschaft eines Service betrachtet [GKG03]. Eine Auswahl der am häufigsten in der Literatur genannten Dienstgütekriterien ist in Abbildung 23.5 grafisch dargestellt. Eine ausführliche Diskussion dieser Kriterien kann [Ber07] entnommen werden.

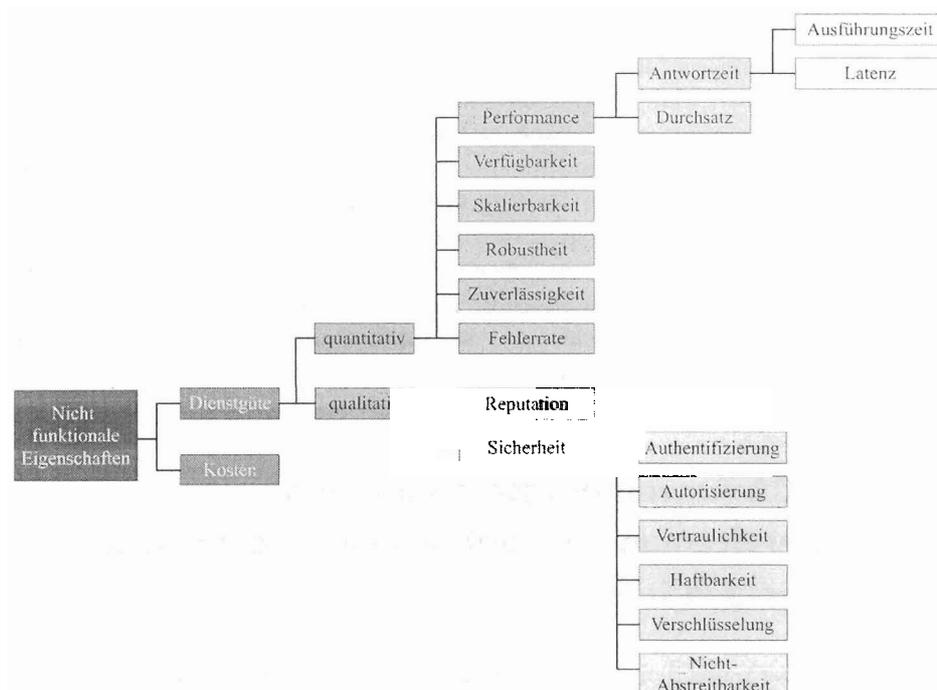


Abbildung 23.5: Klassifizierung von Dienstgütekriterien

23.2.2 SOA-Management-Referenzarchitektur

Die SOA-Management-Referenzarchitektur basiert auf den Ergebnissen mehrerer am Fachgebiet KOM der TU Darmstadt durchgeführter Forschungsprojekte. Sie wurde in Teilen veröffentlicht und in wissenschaftlichen Beiträgen diskutiert [BGR⁺07, BSR⁺07]. Die Ergebnisse dieser Diskussionen sind in die im Folgenden beschriebene Referenzarchitektur integriert worden.

Die SOA-Management-Referenzarchitektur besteht aus den in Abbildung 23.6 dargestellten Einzelkomponenten. Komponenten werden nach ihrem Verwendungszweck, d. h., ob sie primär bei der Erstellung einer SOA oder der Ausführung von Services genutzt werden, entweder der Komponentengruppe »Construction« (Erstellung) oder »Execution« (Ausführung) zugeordnet. Die folgenden Unterabschnitte stellen die jeweiligen Einzelkomponenten vor.

Discovery-Komponente Ein initialer Schritt bei der Erstellung eines Workflows auf Basis der Komposition von einzelnen Services ist das Auffinden adäquater Services, die einen Teil des Geschäftsprozesses abbilden. Aus der Menge der passenden Services wird dann im nächsten Schritt (innerhalb der Komponente

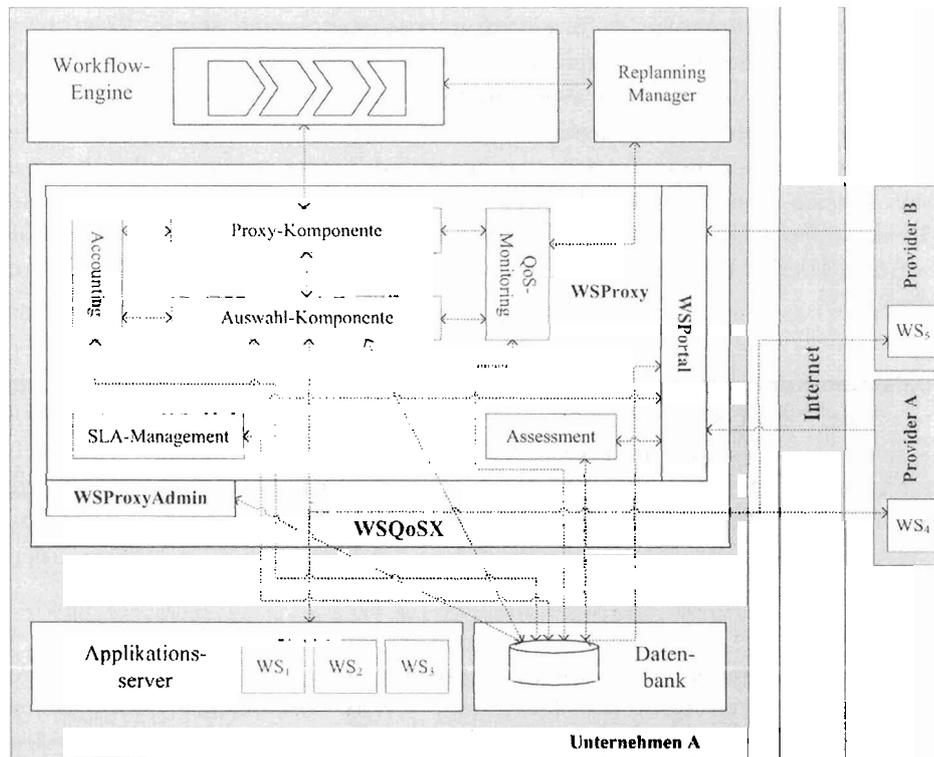


Abbildung 23.6: SOA-Management-Referenzarchitektur

»Selection«) der am besten geeignete Service ausgewählt und zur späteren Integration in den Workflow vorgesehen. Nachdem der Geschäftsprozess in einzelne Prozesskomponenten zerlegt ist, führt die Discovery-Komponente eine Suche nach adäquaten Services durch. Um einen hochflexiblen Workflow zu generieren, sollten einzelne Prozesskomponenten der untersten Hierarchiestufe möglichst genau durch einzelne Services abgebildet werden. Aus der Geschäftsperspektive hingegen kann es sinnvoller sein, Services auf einer höheren Hierarchiestufe zu nutzen, um einen größeren Teil der gewünschten Funktionalität durch einen Service bereitzustellen. Ist die Suche nicht erfolgreich, da ein oder mehrere Services nicht verfügbar oder im Service-Repository nicht vorhanden sind, so wird der Erstellungsprozess durch die Discovery-Komponente abgebrochen. Weiterhin muss die Komponente den Verantwortlichen darüber informieren, dass kein passender Service zur Verfügung steht, damit entweder der Geschäftsprozess angepasst oder der fehlende Service implementiert werden kann. Um eine Suche durchführen zu können, ist eine genaue Spezifikation der Suchkriterien notwendig. Dies erfolgt in Form einer Referenzspezifikation, anhand derer nach Services

in unternehmensinternen (z. B. einem unternehmensweiten Service-Repository) und -externen Servicekatalogen (z. B. einem Marktplatz für Services) gesucht werden kann. Ein Vergleich zwischen Referenzspezifikation und (gefundenen) Servicespezifikationen aus den genannten Quellen kann zu Problemen führen, da Serviceeigenschaften von Serviceanbieter und -nutzer unterschiedlich benannt sein können. Um dieser Problematik entgegenzuwirken, sollten sowohl bei der Servicesuche als auch bei der -beschreibung standardisierte Begriffe eingeführt werden. Hierbei können Ontologien oder Fachnormsprachen zum Einsatz kommen.

Selection-Komponente Nachdem in der Discovery-Komponente eine Menge passender Services gefunden wurde, werden diese Services in der Selection-Komponente bewertet und verglichen. Anhand von funktionalen und nicht funktionalen Dienstgüteparametern des Service wird der am besten passende Service in den Workflow integriert. Bewertungs-, Vergleichs- und Entscheidungsalgorithmen entstammen dabei dem Gebiet der Entscheidungstheorie und dem Operations Research.

Rating-Komponente Die Rating-Komponente errechnet anhand eines gegebenen Service Level Agreement (SLA) und bereits vorhandener Informationen aus vorherigen Ausführungen des betreffenden Service bzw. anderer Services des gleichen Servicegebers eine Rangliste der infrage kommenden Services für ein Einsatzszenario. In einem SLA werden sowohl automatisch messbare als auch nichtmessbare Dienstgüteparameter definiert. Beide Typen von Dienstgüteparametern werden in der Rating-Komponente evaluiert, wobei nichtmessbare Kriterien, wie z. B. Reputation des Serviceanbieters oder Sicherheitsaspekte, vorher von einem menschlichen Experten bewertet werden müssen. Eine solche Beurteilung kann z. B. auf Basis einer Evaluationsmatrix durchgeführt werden. Nach der Beurteilung durch den menschlichen Experten und der Berechnung der messbaren Werte wird dem betrachteten Service eine Punktzahl zugeordnet, die genutzt wird, um eine Rangliste aller potenziellen Services zu erstellen. Diese Rangliste kann als Basis für eine spätere dynamische Auswahl von Services genutzt werden. Die durch die Rating-Komponente erstellte Rangliste kann durch weitere vom menschlichen Experten definierte Bedingungen ergänzt werden.

Assembly-Komponente Nachdem alle zur Umsetzung des Workflows benötigten Services ausgewählt und bewertet wurden, muss die Integration der Services in den Workflow vorbereitet werden. Für die Prozesskontrolle kommt ein Workflow-Management-System (WfMS) zum Einsatz. Zur Laufzeit ist eine Abstimmung der Interaktion zwischen Services und WfMS notwendig, die durch die Assembly-Komponente erfolgt. Hierzu können bei Bedarf auch Wrapper zur Verfügung gestellt werden.

Execution-Komponente Zur Laufzeit liegt die Hauptaufgabe des Management-systems in der Kontrolle der Geschäftsprozessausführung. Dies beinhaltet den Aufruf der beteiligten Services zum richtigen Zeitpunkt und in der korrekten Reihenfolge sowie die Bereitstellung des benötigten Inputs (falls notwendig durch einen Wrapper). Diese Aufgabe kann einem WfMS zugeteilt werden, das als Execution-Komponente agiert und über das Management von Software bzw. Services hinaus auch in der Lage ist, auf andere Ressourcen der Organisation, wie beispielsweise auf Mitarbeiter oder Wissen, zuzugreifen.

Accounting-/Billing-Komponente Accounting bezeichnet die Zuordnung von Aktivitäten des Workflows (also der Nutzung einzelner Services) zu einer verantwortlichen Kostenstelle (dem Verursacher). Diese Zuordnung erfolgt im Normalfall durch den Servicegeber, um als Grundlage für die Rechnungsstellung zu dienen. Hauptaufgabe der Accounting- bzw. Billing-Komponente ist daher die Aufstellung einer Übersicht aller Service-Requests und -Responses, die von Services gesendet bzw. empfangen wurden. Die Komponente sollte verschiedene Abrechnungsmodelle, wie z. B. Pay-per-Use oder Service-Flatrates, ermöglichen.

Monitoring-Komponente Die im SLA definierten Dienstgüteparameter werden zur Laufzeit von einer Monitoring-Komponente überwacht. Diese Komponente analysiert in der Accounting-Komponente gesammelte Daten und vergleicht diese mit den im SLA garantierten Werten. Kommt es zu Abweichungen zwischen SLA-Vorgaben und gemessenen Werten, werden sowohl der betreffende Servicegeber als auch der Servicenehmer benachrichtigt. Weiterhin können weniger performante Services automatisch durch einen anderen Service, der die gleiche Funktionalität anbietet, ersetzt werden, indem die Selection-Komponente benachrichtigt wird und die Planung des Workflows bzw. die Auswahl geeigneter Services erneut angestoßen wird.

SLA-Management-Komponente Wie bereits beschrieben, handelt es sich bei SLAs um Verträge zwischen Servicenehmer und -geber, in denen Dienstgüteparameter bzw. die Anforderungen des Servicenehmers und die Zusicherungen des Servicegebers definiert sind. Weiterhin kann ein SLA Informationen über den Preis und sonstige Vertragsdaten beinhalten. SLAs sind das Ergebnis von Verhandlungen zwischen Kunden und Servicegeber. In der hier betrachteten Referenzarchitektur werden SLAs von der SLA-Management-Komponente verwaltet. Diese Komponente ist in der Lage, ein gegebenes SLA einzulesen und die relevanten Informationen über garantierte Werte an die Monitoring-Komponente weiterzugeben.

Repository System Repositories dienen der Dokumentation von (Software-)Artefakten und beinhalten Datenstrukturen, Softwarekomponenten, Services und

andere Sprachartefakte. Ihre Dokumentation wird in Form einer vorgegebenen Beschreibungssprache vorgenommen. Die Datenstruktur zur Dokumentation der Artefakte wird als Metaschema oder Dokumentationsstruktur des Repositories bezeichnet. Innerhalb des Repositories der SOA-Referenzarchitektur werden die Services selbst verwaltet sowie Baupläne von Workflows auf Basis von Services für die spätere Wiederverwendung. Auch wird der Verwendungsnachweis einzelner Services innerhalb des Repositories vorgehalten. Dies dient der Nachvollziehbarkeit (im Sinne des Auditings) sowie der Wartbarkeit des gesamten Systems.

23.2.3 WSQoSX: Prototypische Implementierung eines SOA-Management-Systems

In diesem Abschnitt wird mit WSQoSX (Webservices Quality of Service Architectural Extension) [BGR⁺05, Ber07] eine erweiterte Webservice-Architektur mit umfassender Dienstgüteunterstützung vorgestellt. WSQoSX ermöglicht die dynamische, dienstgütebasierte Komposition von Webservices zu Workflows. Ein Schwerpunkt liegt auf der Auswahl, Einbindung und Ausführung der beteiligten Webservices. Externe Anbieter müssen das Laufzeitverhalten ihrer Webservices in Form von SLAs garantieren. WSQoSX steuert und überwacht die Dienstgüte der entsprechenden Webservices auf Grundlage eines Vorgehensmodells, das aus neun Phasen besteht.

WSQoSX besteht aus den drei Kernkomponenten WSPProxy, WSPProxyAdmin und WSPortal, deren Funktionalität in diesem Abschnitt detailliert beschrieben wird. Neben diesen Kernkomponenten werden noch ein Datenbankserver sowie eine Workflow-Engine benötigt. Beide sind selbst kein Bestandteil von WSQoSX im engeren Sinne. Die Workflow-Engine nimmt aus der Perspektive von WSQoSX die Rolle eines Client ein, der Webservices aufruft. Allerdings werden im Rahmen von WSQoSX diese Aufrufe auf den WSPProxy umgeleitet. Durch diesen zusätzlichen Indirektionsschritt beim Aufrufen von Webservices wird ein dynamisches Routing von Webservice-Aufrufen möglich.

WSQoSX stellt über die Komponente WSPortal eine Anbieterschnittstelle in Form eines Webportals zur Verfügung, über das interne Fachabteilungen und externe Serviceanbieter ihre Webservices zur Nutzung bereitstellen können. Über das Portal kann ein Unternehmen, das WSQoSX verwendet, Ausschreibungen für benötigte Services potenziellen Nutzern bekannt machen. Das Portal kann mittels eines Webbrowsers plattformunabhängig verwendet werden.

Über das Administrations-Frontend WSPProxyAdmin wird festgelegt, wie der WSPProxy die Webservice-Aufrufe seitens der Workflow-Engine verarbeiten und weiterleiten soll. WSPProxy kann einen Webservice-Aufruf entweder statisch an einen bestimmten Webservice weiterleiten oder dynamisch zur Zeit des Aufrufs einen geeigneten Webservice ermitteln. WSQoSX erlaubt ein dienstgütebasiertes Management von Webservices, beginnend mit deren Anmeldung und Registrierung am Portal bis hin zur endgültigen Deaktivierung (vgl. Abbildung 23.7).

Im Folgenden wird ein im Rahmen dieser Arbeit entwickeltes, aus neun Phasen bestehendes Vorgehensmodell zum Dienstgütemanagement von Webservices vorgestellt und es wird beschrieben, wie diese Phasen durch WSQoSX unterstützt werden.

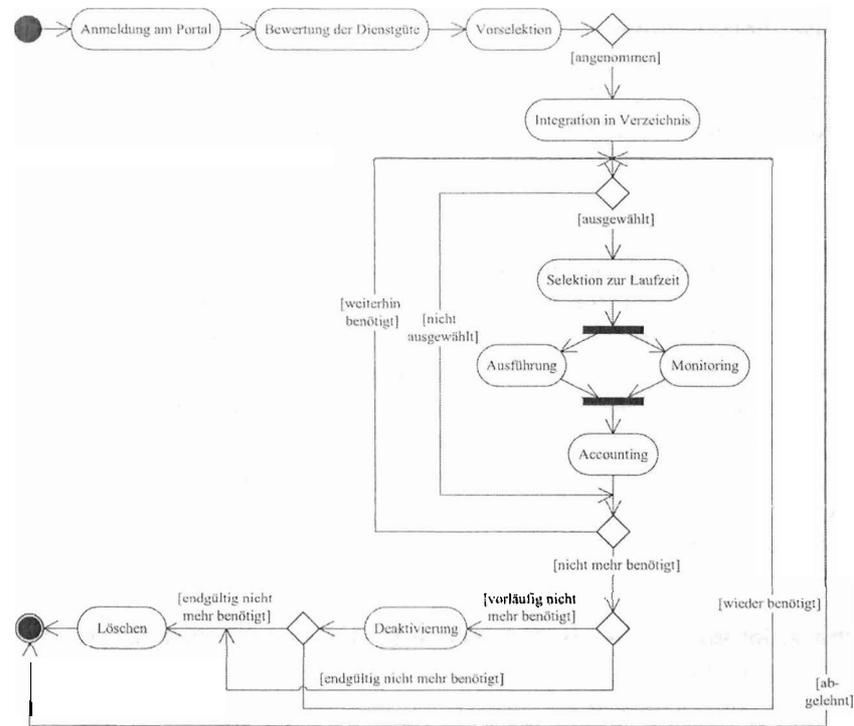


Abbildung 23.7: Vorgehensmodell zur Dienstgüteunterstützung

1. Phase: Anmeldung am WSQoSX-Portal des Nutzers Unternehmen, die ihre Webservice-Workflows durch WSQoSX steuern, können an ihrem von WSQoSX bereitgestellten Portal Ausschreibungen zu Webservices veröffentlichen, die sie an einen externen Anbieter vergeben wollen. Möchte ein externer Webservice-Anbieter als Reaktion auf eine solche Ausschreibung einen Webservice zur Verfügung stellen, so muss er sich zuvor, falls noch nicht geschehen, am WSQoSX-Portal des ausschreibenden Unternehmens registrieren.

2. Phase: Bewertung der Dienstgüte Nachdem der entsprechende Webservice mit zugehörigem SLA am Portal angemeldet wurde, erfolgt dessen Bewertung anhand der Dienstgütekriterien, die in den SLAs garantiert sind. Die Gewichtung dieser Dienstgütekriterien untereinander wird aus der Präferenzstruktur des

Nutzers abgeleitet und geht ebenfalls mit in die Bewertung ein. Eine ausführliche Diskussion, wie die Dienstgüte der Webservices im Rahmen von WSQoSX bewertet wird, erfolgt in [Ber07].

3. Phase: Vorselektion In dieser Phase wird auf Grundlage der bisher vorgenommenen Bewertung entschieden, ob der Webservice in das interne Verzeichnis übernommen wird. Hierzu können Ausschlussregeln definiert werden, die sich entweder auf die Gesamtbewertung des Webservice oder auf einzelne Teilkriterien beziehen. Eine Ausschlussregel bezüglich der Gesamtbewertung kann beispielsweise lauten: »Webservices, deren Gesamtbewertung kleiner als acht Punkte ist, werden nicht in das interne Verzeichnis übernommen.« Bezogen auf einzelne Teilkriterien können Ausschlussregeln definiert werden, die z. B. besagen, dass Webservices, deren externe Referenzen mit weniger als vier Punkten bewertet wurden, abgelehnt werden. Es ist auch möglich, Ausschlussregeln miteinander zu verknüpfen. Neben den absoluten Regeln können auch relative definiert werden, die beispielsweise Webservices ausschließen, deren Gesamtbewertung nicht unter den 10 bestplatzierten liegt. Unabhängig von den Bewertungen können in der Phase Vorselektion auch solche Webservices ausgeschlossen werden, mit deren Anbieter der Nutzer, z. B. aus strategischen oder regulatorischen Gründen, nicht zusammenarbeiten möchte bzw. darf.

4. Phase: Integration eines Webservice in das interne Verzeichnis Webservices, die die Phase Vorselektion erfolgreich durchlaufen haben, werden in das interne Verzeichnis übernommen und stehen zur Integration in die jeweiligen Geschäftsprozesse zur Verfügung. Die Anbieter der nicht berücksichtigten Webservices werden über die Ablehnung (und ggf. über die Gründe hierfür) benachrichtigt.

5. Phase: Selektion eines Webservice zur Laufzeit Ausschlaggebend, ob und wie häufig ein Webservice aus dem internen Verzeichnis tatsächlich ausgewählt wird, ist die Bewertung seiner Dienstgüteeigenschaften aus Phase 2. Grundsätzlich wird derjenige Webservice ausgewählt, der die höchste Gesamtbewertung in seiner Kategorie erhalten hat. Im Zusammenhang mit Geschäftsprozessen, an die besonders hohe Anforderungen gestellt werden, können aber auch noch für die infrage kommenden Webservices zusätzliche Nebenbedingungen definiert werden, die die Ausschlussregeln aus der Phase Vorselektion verschärfen. So ist es Unternehmen möglich, flexibel auf äußere Einflüsse zu reagieren. Sollte beispielsweise kurzfristig mit einem starken Anstieg von Kundenanfragen für einen bestimmten Webservice zu rechnen sein, wird durch eine zusätzliche Nebenbedingung festgelegt, dass der zum Einsatz kommende externe Webservice einen besonders hohen Durchsatz garantiert.

6. Phase: Ausführung Derjenige Webservice, der in der vorigen Phase als am besten geeignet bezüglich der vorgegebenen Kriterien ermittelt wurde, wird nun dynamisch zur Laufzeit in den Workflow integriert und ausgeführt.

7. Phase: Monitoring und Accounting Für extern bezogene Webservices erhält das Unternehmen eine Abrechnung, die auf unterschiedlichen Preismodellen (z. B. nutzungsabhängige und -unabhängige Tarife) basieren kann [GKG03]. Zur Erstellung dieser Abrechnung führt der externe Webservice-Anbieter ein Accounting durch. Unter Accounting versteht man die verursachungsgerechte Zurechnung von Aktivitäten eines Informationssystems zu einer Ressource [Com01]. Aber auch für den Nutzer der Webservices ist ein Accounting von Bedeutung, da so eine verursachungsgerechte Kostenzuordnung zu den internen Organisationseinheiten, die den Service tatsächlich in Anspruch genommen haben, möglich wird. Außerdem werden die durch das Accounting gewonnenen Daten für die Berechnung der Dienstgütekriterien (z. B. Verfügbarkeit) verwendet.

8. Phase: Deaktivierung eines Webservice Wird die von einem Webservice bereitgestellte Funktionalität vorläufig nicht mehr benötigt, so kann der Nutzer diesen deaktivieren. Sollte der Webservice zu einem späteren Zeitpunkt wieder benötigt werden, hat der Nutzer die Möglichkeit, diesen wieder zu aktivieren.

9. Phase: Endgültige Löschung eines Webservice Wird ein Webservice seitens des Nutzers endgültig nicht mehr benötigt, kann der Nutzer diesen aus dem Verzeichnis löschen. Es kann auch dann notwendig werden, einen Webservice zu löschen, wenn der Anbieter den entsprechenden Webservice aus seinem Angebot nimmt.

23.3 Fazit

Das Paradigma der serviceorientierten Architektur gewinnt im Zusammenhang mit der Überwindung von Komplexität in heterogenen Anwendungslandschaften und der architekturellen Unterstützung unternehmensübergreifender Workflows zunehmend an Bedeutung. Energieversorgungsunternehmen (EVU) nutzen SOA beispielsweise zur Entwicklung flexibler, innovativer und effektiver Systeme, wie am Beispiel des Forschungsprojekts DEMS dargestellt.

Fachliche Geschäftsfunktionen der Energiewirtschaft werden nach außen gekapselt und über standardisierte Formate als Dienste angeboten. Durch diese Kapselung wird sowohl die im Forschungsprojekt benötigte Plattform-, Programmiersprachen- und Standortunabhängigkeit als auch die Integration vorhandener Altanwendungen erreicht. Die Prozesse für das zukünftige Energiemanagement sind ein entscheidender Faktor für EVU, da sie Energieplanung, -handel oder auch -steuerung unterstützen. Das Konzept der SOA stellt diese Prozesse in den Fokus

und ermöglicht die flexible Gestaltung der Prozesse durch Verwendung funktionaler und datenbezogener Dienste. Die Nutzung eines logischen ESB erlaubt eine lose Kopplung der Dienste, insbesondere dadurch, dass durch die Nutzung von Metainformationen für das Routing sowie normierter Prozessschnittstellen und Nachrichtenformate auf Basis der Normen IEC 61968 und IEC 61970 die Kommunikation innerhalb des Systems sowie mit externen Services erheblich vereinfacht ist.

Entscheidend für den weiteren Erfolg des SOA-Paradigmas und der Webservice-Technologie ist eine umfassende Unterstützung des Dienstgütemanagements der entsprechenden Services und der aus Services zusammengesetzten Workflows. Im Rahmen dieses Abschnitts wurde daher die Dienstgütearchitektur WSQoSX vorgestellt, die das Dienstgütemanagement für serviceorientierte Workflows unterstützt. WSQoSX übernimmt das Dienstgütemanagement basierend auf einem neun Phasen umfassenden Vorgehensmodell.

Auf diese Art und Weise wird durch WSQoSX sichergestellt, dass nur Webservices ausgewählt werden, deren Dienstgüteeigenschaften den Anforderungen des Nutzers entsprechen. Die Dienstgüteeigenschaften der entsprechenden Webservices müssen durch Service Level Agreements (SLAs) zugesichert werden. Die Einhaltung der SLAs wird ebenfalls durch WSQoSX überwacht.