

SOFTWARE-PLUS-SERVICES ALS WEITERENTWICKLUNG EINER SERVICE-ORIENTIERTEN ARCHITEKTUR

Unternehmen wissen heute zu Recht die Vorteile einer service-orientierten Architektur zu nutzen. Die Kapselung verteilter IT-Funktionalität in Softwareservices und deren lose Kopplung zur Bereitstellung von Anwendungsfunktionalität bietet Unternehmen Agilität und Flexibilität. Microsofts Software-plus-Services Ansatz führt die Idee der SOA weiter, indem er die Möglichkeit eröffnet, IT-Systeme flexibel aus lokal vor-Ort betriebenen Diensten und Diensten, die über das Internet bereitgestellt werden, zu erstellen und für jeden eingesetzten Service das bestgeeignete Deploymentmodell wählen zu können.

Service-Orientierung der IT

Die service-orientierte Architektur (SOA) ist als Architekturstil heute häufig fester Bestandteil der IT-Strategie in Unternehmen. Durch die lose Kopplung verteilter IT-Funktionalität, die als eine Menge von Diensten (Services) implementiert ist, soll Agilität, Adaptierbarkeit und Wiederverwendbarkeit der IT-Systeme erreicht werden, die letztlich zu einer Agilität des Unternehmens selbst führen soll. Neue Anwendungsfunktionalitäten sollen durch Aggregation oder Komposition von Softwareservices implementiert werden können. In der Realität wird SOA leider häufig auf die IT-Systeme innerhalb eines Unternehmens beschränkt gesehen, wenngleich das Konzept an sich auch über Unternehmensgrenzen hinweg anwendbar ist. Weitere Einschränkung ist, dass SOA zwar etwas über das Angebot von IT-Funktionalitäten als Services, aber nichts über den Konsum und damit letztlich darüber, wie Anwender auf die Services zugreifen können, aussagt.

Unter anderem in diesen zwei Aspekten erweitert Microsofts „Software-plus-Services“-Strategie das SOA-Konzept. „Software-plus-Services“ geht davon aus, dass Unternehmen in Zukunft für ihre IT-Anforderungen eine Kombination aus Internetservices und lokal betriebenen Client- und Server-Anwendungen einsetzen,

auf die Anwender über die verschiedensten Endgeräte mit bestmöglicher User Experience zugreifen. Für die Entscheidung, welche Funktionalitäten durch lokal betriebene Software und welche durch internetbasierte Services erbracht werden, wird größtmögliche Flexibilität angestrebt. D.h. Unternehmen sollen von Fall zu Fall entscheiden können, welche Teile ihres IT-Portfolios lokal und welche in der „Cloud“ betrieben werden. Diese Entscheidung soll sowohl für ganze Anwendung, als auch für einzelne Services bis hin zu Anwendungskomponenten möglich sein. Dabei kann sich die Entscheidung auch über die Zeit hinweg ändern. So ist es vorstellbar, dass ein selbst vor Ort betriebener IT-Dienst nach gewisser Zeit beispielsweise aus Skalierungsgründen zu einem IT-Dienstleister ausgelagert wird.

Wahlfreiheit beim Deployment

Grundsätzlich stehen Unternehmen für den Betrieb von Softwarekomponenten vier Deploymentoptionen zur Auswahl: Zunächst ist und bleibt der klassische Softwarebetrieb vor Ort im Unternehmen möglich. Da die Anwender im Unternehmen auf die IT-Systeme zugreifen, ist dies die einzige Deploymentoption für Client Software, die auf dem Gerät des Anwenders ausgeführt wird. Beim Softwarebetrieb vor Ort hat das



Holger Sirtl

(E-Mail: holger.sirtl@microsoft.com)

ist Architect Evangelist, S+S Lead und seit 2006 als Architekturberater bei Microsoft in München tätig. Schwerpunktthema seiner Arbeit ist Microsofts "Software + Services"-Strategie. In diesem Zusammenhang berät er Unternehmen im Aufbau .NET-basierter Anwendungsarchitekturen, die vor Ort betriebene Software mit internetbasierten Anwendungsdiensten verknüpfen. Vor seinem Einstieg bei Microsoft arbeitete Holger Sirtl sechs Jahre lang als Technologieberater für eine international führende Unternehmensberatung sowie zwei Jahre lang als Senior-IT-Projektmanager für einen großen deutschen Energieversorger.

Unternehmen volle Kontrolle über mögliches Customizing und die Integration der Software in die restliche IT-Landschaft. Individuelle Anforderungen können im Unternehmen selbst implementiert werden. Die zweite Möglichkeit des Deployments ist der Softwarebetrieb auf der Infrastruktur eines Service Providers, der die Software dann als IT-Service dem Unternehmen bereitstellt. Dieses profitiert von höheren Skaleneffekten des Service Providers. Dieser kann Infrastrukturkosten auf mehrere Unternehmen verteilen. Das nachfragende Unternehmen erkaufte sich diesen Vorteil allerdings mit gewissen Einschränkungen beim Customizing. Dieses muss mit dem Service Provider abgestimmt werden und darf nicht mit Softwarediensten anderer Kunden kollidieren. Auch die Integration wird durch die nun zu überbrückende und typischerweise durch diverse IT-Sicherheitssysteme (Firewalls etc.) abgesicherte Unternehmensgrenze erschwert. Je enger die Zusammenarbeit mit dem Service Provider ist, desto leichter lassen sich diese Einschränkungen abmildern. Bei der dritten Deploymentoption, dem Betrieb auf einer sogenannten „Cloud Plattform“, ist eine solche enge Zusammenarbeit mit dem Anbieter nicht mehr möglich. Das Angebot derartiger Plattformanbieter wie Amazon (mit der EC2 Plattform), Salesforce (Force.com) oder Microsoft ist stark stan-

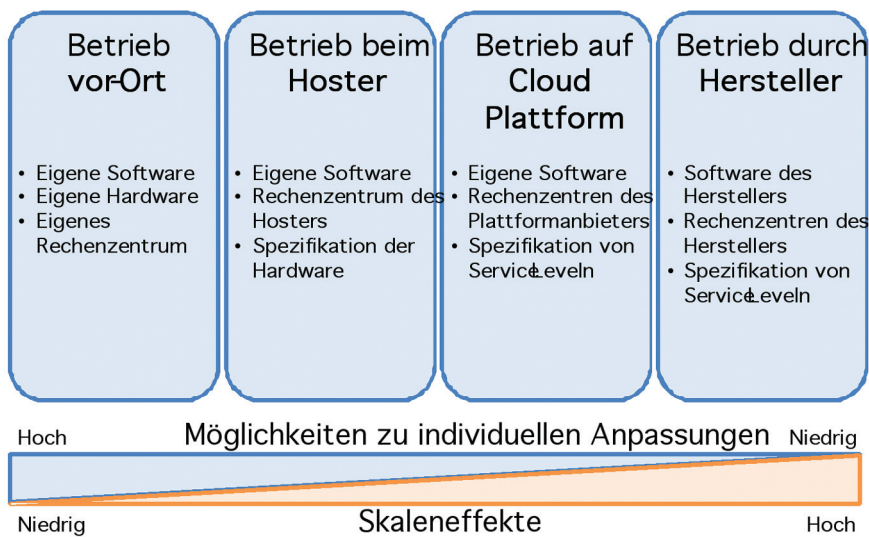


Abbildung 1: Alternativen für den Softwarebetrieb

dardisiert. Die zu betreibenden Anwendungen müssen ggf. an die Plattform angepasst werden. Diese Einschränkung rechtfertigt sich allerdings durch die hohe Skalierbarkeit der Plattformen. Während bei den ersten drei Deploymentoptionen die Anwendung selbst vom Unternehmen gestellt wird, bezüglich der Funktionalität also Vorgaben gemacht werden können, greift das Unternehmen bei der vierten Option auf eine durch einen Hersteller bereitgestellte Anwendung zu. Beispiele für diese Variante sind die CRM-Lösung von Salesforce und SharePoint Online von Microsoft. Ein Customizing der Anwendung ist in diesem Fall zwar möglich, die Gesamtfunktionalität ist aber vom Angebot des Herstellers abhängig.

Voraussetzung für diese Wahlfreiheit

Um Unternehmen nun möglichst große Freiheit bei der Wahl des Deploymentmodells zu bieten, sind zwei Faktoren entscheidend. Zunächst ist dies eine starke Service-Orientierung der IT. Eine Ausgestaltung der IT-Landschaft nach SOA-Prinzipien erlaubt es, IT-Funktionalität, die in einem Service gekapselt ist, zu mobilisieren. Einzige Voraussetzung ist, dass die Schnittstelle konstant bleibt und der Service weiterhin zugreifbar bleibt. Für den Aufrufer ist es deshalb unwichtig, wo der Service konkret betrieben und die Funktionalität erbracht wird.

Wenngleich das innere Verhalten eines Services für einen Aufrufer keine Rolle spielt, so ist die Funktionsweise des Services doch für den Betreiber des Services von Bedeutung. Der Betreiber muss eine Plattform und Laufzeitumgebung bereitstellen, auf der der Service ausgeführt werden kann und die Schnittstellen unterstützt, die von potenziellen Aufrufern angesprochen werden. Die Mobilität eines Service ist demzufolge dann gegeben, wenn Plattform und Laufzeitumgebung in den unterschiedlichen Deploymentoptionen verfügbar sind und sich möglichst wenig voneinander unterscheiden.

Die Services Plattform von Microsoft

Während die Umgebung bei Service Providern noch sehr vergleichbar mit der vor Ort bestehenden ist, werden im Falle von Cloud Plattformen bzw. der Bereitstellung von Anwendungsdiensten von Herstellern Plattformen erforderlich, die die hohen Anforderungen an Skalierbarkeit, Stabilität und Funktionalität erfüllen. Die Funktionalitäten, die eine solche Services Plattform ausmachen, umfassen Unterstützung für Multimandanten-Szenarien, Scale-out (im Unterschied zu Scale-up), Virtualisierung und Virtualisierungsmanagement, verschiedene Abrechnungsszenarien, Überwachung und Management, Verknüpfung von Services und schließlich auch Entwickler-

Werkzeuge, die diese Funktionalitäten unterstützen.

Microsoft bietet eine derartige Services Plattform an und stellt darauf verschiedene Dienste für Anwender (sogenannte Finished Services) und Entwickler (Building Block Services) zur Verfügung. Für alle Serverprodukte von Microsoft werden in Zukunft entsprechende Services bereitgestellt. Im Falle von Exchange, SharePoint und Dynamics CRM wird es für den Endanwender keinen Unterschied machen, welches Betriebsmodell gewählt wird. Anpassungen und Zugriffe erfolgen in jedem Modell gleich. Daneben gibt es noch sogenannte „native“, d.h. eigens für den Betrieb in der Cloud entwickelter Services. Office Live Meeting, Exchange Hosted Services und Virtual Earth sind alles Beispiele für Services, die nur über die Cloud zugreifbar sind.

Migration zwischen Deploymentmodellen

Diese Unterscheidung in Plattform Services, die direkt vor Ort betreibbaren Serverprodukten entsprechen, und solchen, die speziell für den Einsatz in der „Cloud“ geschrieben wurden, ist entscheidend für die Aufwände, die notwendig sind, wenn von einem Deploymentmodell zu einem anderen gewechselt werden soll. Bei erstgenannten ist die Migration vergleichsweise einfach: ein derartiger, auf ein bestimmtes Serverprodukt aufsetzender Service kann ohne größere Änderungen in die neue Umgebung migriert werden. Dabei muss lediglich sichergestellt werden, dass Service-Aufrufe entsprechend umgeleitet werden. Dies kann im Falle einer SOA beispielsweise durch Umkonfiguration des Service Bus geschehen. Der Service selbst kann in der neuen Umgebung ohne größere Änderungen laufen. Eine solche Migration kann z.B. bei einem Wechsel eines lokal betriebenen SharePoint Servers zur Online-Variante „SharePoint Online“ erfolgen.

Bei Plattform Services, die speziell für den Betrieb in der „Cloud“ entwickelt wurden, liegt die Situation anders: der Cloud Service bietet ggf. nicht den gleichen Funktionsumfang wie das entsprechende Serverprodukt. In diesem Fall muss entschieden werden, ob und wie die gewünschte Funktionalität in allen Deploymentszenarien bereitgestellt werden kann. In einer SOA wirken sich die Details des Funktionsumfangs des Plattform Service



idealerweise nicht auf die Schnittstelle, die Aufrufer verwenden, aus. In diesem Fall müssen keine weiteren Änderungen in den Aufrufern vorgenommen werden. Anpassungen erfolgen dann nur innerhalb des Services. Eine Migration erfolgt somit in den drei Schritten:

1. Prüfung und Bewertung des Funktionsumfangs des Plattformservices
2. Sicherstellung der Konstanz der Service-Schnittstelle
3. Anpassung der internen Funktionsweise des Service

Ein Beispiel für diese Problematik ist der Themenbereich der Datenspeicherung.

Beispiel: Datenspeicherung vor Ort vs. Datenspeicherung in einer Cloud Datenbank

Für die Datenspeicherung vor Ort stehen auf Microsofts Plattform eine Reihe von Möglichkeiten zur Verfügung: Neben der Speicherung in einfachen Dateien oder XML-Dateien gibt es mit SQL Server auch ein leistungsfähiges relationales Datenbanksystem. Im .NET Framework stehen für diese Datenspeicher zahlreiche Zugriffsmechanismen zur Verfügung.

Die SQL Server Data Services (SSDS) sind Microsofts Cloud Database Service und gehören den sogenannten Building Block Services der Microsoft Services Plattform an. Die SSDS unterstützen sowohl SOAP als auch REST Schnittstellen. Dadurch wird es möglich, die SSDS aus allen Entwicklungsumgebungen (inkl. diverser Programmiersprachen) heraus zu nutzen, die diese Standards unterstützen.

Abbildung 2 listet die Datenspeicher mit den jeweiligen zur Verfügung stehenden Zugriffsmechanismen auf.

Deployment-option Speicher	Datenspeicherung vor Ort			Microsofts Cloud Plattform SQL Server Data Services
	Dateien	XML	RDBMS (SQL Server)	
Struktur	•Unstrukturiert	•XML	•Tabellen	•Hierarchie
Zugriffstechnik	•File Stream	•XML Stream •Linq 2 XML	•ADO.NET •Linq 2 DataSet •Linq 2 SQL •Entity Fwk. •Linq 2 Entity	•SOAP •REST

Abbildung 2: Alternativen für die Datenspeicherung

SSDS gehören zu den oben beschriebenen „nativen“ Services. SSDS sind also keine direkte Entsprechung zu einem vor Ort betriebenen SQL Server, sondern eine eigene auf die speziellen Anforderungen des Betriebs in der „Cloud“ optimierte Implementierung für Datenspeicherung.

Für eine Migration der Datenhaltung von einem lokalen Betrieb hin zur „Cloud“ sind deshalb deutliche Anpassungen der betreffenden Anwendung erforderlich. Ein direkter Übertrag der Daten und Zugriffsmechanismen ist nicht möglich.

Während Daten im SQL Server in Tabellen gespeichert werden und die von einem relationalen Datenbankmanagementsystem (RDBMS) erwarteten Konzepte wie Views und Stored Procedures zur Verfügung stehen, sind die Daten in SSDS hierarchisch in Authorities, Container und Entities strukturiert. Views und Stored Procedures stehen nicht zur Verfügung. Werden diese unbedingt benötigt, scheidet eine Migration der Datenhaltung hin zu SSDS aus. Als einzige Option besteht neben dem vor-Ort-Betrieb dann noch der Betrieb bei einem Service Provider, der dann wiederum einen SQL Server bereitstellt, dessen Funktionsumfang dem eines lokalen SQL Servers entspricht.

Bei einer Migration der Daten von SQL Server nach SSDS muss das relationale

Datenmodell auf ein hierarchisches Modell abgebildet werden. Tabellen können dabei – ggf. denormalisiert – auf Entities abgebildet werden. Tabellenbeziehungen werden in Programmlogik nachgebildet.

Fazit

Software-plus-Services geht davon aus, dass das IT-Portfolio von Unternehmen in Zukunft aus lokal betriebenen Software-services und internetbasierten Services zusammengesetzt ist. In dieser Hinsicht erweitert Software-plus-Services das Architekturprinzip einer SOA, das in der Praxis bisher oft auf eine unternehmensinterne Sicht begrenzt war. Eine Service-Orientierung schafft die Voraussetzung, in Services gekapselte IT-Funktionalitäten zu Service Providern, Cloud Plattformen oder Software-Herstellern zu migrieren. Microsoft bietet eine durchgängige Plattform, die einen Betrieb in den unterschiedlichen Deploymentoptionen unterstützt und damit Unternehmen größtmöglichen Entscheidungsspielraum beim Softwarebetrieb bietet. ■

Weiterführende Informationen, die direkt an diesen Artikel anknüpfen, finden Sie unter <http://www.microsoft.de/architecture/os1108>