

Microsoft S+S

ein Entwurfsmuster für modulare Dienste im Vergleich mit herkömmlicher Software-as-a-Service

Marc Bischof `bischohc@studi.informatik.uni-stuttgart.de`

Institut für Architektur von Anwendungssystemen
Universität Stuttgart
Universitätsstraße 38, 70565 Stuttgart

Zusammenfassung In einem Artikel heißt es „Microsoft hat bisher offenbar keinen Ansatz gefunden, wie man die Dominanz aus dem PC-Sektor ins Web-Zeitalter herüberretten könnte. Dabei wäre Microsoft eigentlich ideal positioniert“ [1]. Mit S+S bietet Microsoft jedoch ein Entwurfsmuster für modulare Dienste, die zum einen Kundenorientierung auch für große Unternehmen erlaubt und zum anderen das herkömmliche Software-as-a-Service Modell verallgemeinert.

Dieser Artikel klärt zunächst, was unter herkömmlicher Software-as-a-Service und Software-plus-Service zu verstehen ist. Der anschließende Vergleich geht auf die Erfüllung von Anforderungen an Dienste und auf die Variationen von SaaS ein. Außerdem verallgemeinert er Microsofts Modell um mit dem Software-Service Kompositionsmodell und dem S+S Liefermodell die zukünftige SOA-Welt zu beschreiben.

1 Einführung

Durch günstige Speichermöglichkeiten und flächendeckende Breitbandnetze ist die internetgebundene Softwarelieferung die logische Weiterentwicklung des klassischen Client-basierten Modells. Der Markt dafür ist allerdings dominiert von Angeboten für Endkunden und kleine Unternehmen, die von dem Software-as-a-Service (SaaS)-Liefermodell profitieren da sie keine hohen Integrationsansprüche stellen. Große Unternehmen verlangen jedoch sowohl nach lokal installierter Software als auch nach gehosteten Diensten [2] mit ihren eigenen Qualitäts- und Integrationsansprüchen. Je wichtiger dabei eine Anwendung für ein Unternehmen ist, desto weitreichender sind die Folgen die zu berücksichtigen sind. Anwendungen als Dienst zu nutzen ist daher für Unternehmen ein großes Integrationsvorhaben mit vielen wichtigen Punkten, wie vertraglichen Aspekten, Sicherheit und Datenbesitz [3].

SaaS befreit Unternehmen hierbei von einem Großteil der Investitionen in Wartung, Pflege und IT-Infrastruktur. Software-plus-Services (S+S) greift zusätzlich das Bedürfnis nach kundenspezifischen Lösungen in einer Software-Service-Welt auf und folgt damit dem wiederauflebenden Interesse an vernetzten Desktopanwendungen, die Aktivitäten auf Webseiten mit lokalen Medien verbindet [4]. Microsoft glaubt dabei, dass „one-size-fits-all“-Ansätze für Unternehmen der

Zukunft ungeeignet sind. Da sie entweder komplett Client-Server basierte Software oder komplette SaaS-Dienste nutzen. Zukünftige Unternehmen werden danach individuelle Endgeräte, Server und SaaS unterstützte Dienste einer Client-Server-Service Konfiguration nutzen um eine vollständige „Service Plattform“ zu bieten [5].

Der Aufbau dieses Artikels ist wie folgt. Zunächst folgt als Motivation die Vision einer Software-Service-Welt und es wird geklärt was unter herkömmlicher SaaS zu verstehen ist. Anschließend wird Microsofts Softwareliefermodell S+S vorgestellt und mit dem konventionellen SaaS-Ansatz verglichen.

1.1 Vision einer Software-Service-Welt

Laut Microsoft werden in Zukunft Software-Services, die sowohl herkömmliche SaaS als auch S+S umfassen, die IT-Welt bestimmen. Im strengeren Sinn stellen dabei Software-Services das Software-Kompositionsmodell dar und SaaS eine spezielle Ausprägung des S+S Liefermodells. Neben reinen SaaS-Diensten bieten dabei Anbieter außerdem Schnittstellen an, sodass Kunden die Software anpassen und eigene Softwaremodule schreiben können. In letzter Instanz können diese dann sogar mit anderen Kunden geteilt werden.

Die Vision von S+S ist es Anwendungsarchitekturen zu erstellen, die Ressourcen von Clients und anderen Geräten wirksam einsetzen, um dem Endbenutzer reiche und intuitive Fähigkeiten zu bieten [6]. S+S ist dabei ein zusätzliches Modell, das zwischen Paketsoftware und herkömmlicher SaaS liegt. Es geht über erstere hinaus, lagert aber dennoch die Dienste nicht vollständig aus. Es gibt Kunden mehr Flexibilität und Wahlmöglichkeiten in Bezug auf den Softwareeinsatz – einschließlich der lokalen Installation, Auslieferung über das Internet oder Mischlösungen die das Beste aus beiden Welten liefern [2].

Grundlegend bei diesem Konzept ist es, die verschiedenen Ausprägungen von Diensten in Bezug auf Software zu unterscheiden. Zum einen in das Liefermodell, zum anderen in die Komposition und weiterhin die Erscheinung. S+S verbindet dabei als Instanz vom Software-Service-Kompositionsmodell die Lieferung mit der Erscheinung. Die Nutzung der Dienste erfolgt neben dem PC durch verschiedene andere Endgeräte mit diversen (Browser)-Clients, Servern und anderen Diensten [7]. S+S ist also die plattformunabhängige Wiederverwendung von Diensten auch im Backend oder anders die Verallgemeinerung von SaaS auf modulare Dienste.

2 Konventionelle SaaS

SaaS ist ein Software-Liefermodell mit der Philosophie, Software als Dienstleistung basierend auf Internettechnologien bereitzustellen, betreuen und zu betreiben. Der Kunde braucht sich dabei nicht um den Betrieb von Anwendung und Infrastruktur kümmern. Er zahlt nicht für den Besitz der Software, sondern für deren Benutzung, entweder auf Nutzenbasis oder als Abonnement, weshalb SaaS auch als Softwaremiete, Software on Demand oder Softwareleasing bezeichnet wird. Dadurch haben vor allem kleine Unternehmen und private Anwender die

Möglichkeit kommerziell lizenzierte Software zu nutzen ohne die damit verbundene Infrastruktur, Know-how und Einstiegskosten zu tragen.

Im SaaS-Modell werden Anwendungen von einem zentralen Anbieter über das Internet genutzt oder gemietet [8]. Bei diesem Konzept profitieren beide Seiten, sowohl Anbieter durch regelmäßige Einkünfte und Kundenbindung, als auch Kunden durch Einsparung von Lizenzgebühren und Investitionen in Wartung, Pflege und Infrastruktur. Dieses Vorgehen unterscheidet sich von der Auslagerung kompletter Geschäftsprozesse dahingehend, dass nicht der gesamte Prozess ausgelagert wird, sondern lediglich die Software. Im Unterschied zu „Hosting Services“ liegt bei SaaS der Fokus nicht auf dem Betreiber der Software und Infrastruktur sondern auf dem Nutzer [8]. Kernkonzept ist es daher Zugang und Management zentralisierter Anwendungen mit einem „one-to-many“ Angebot zu bieten.

Anbieter profitieren dabei von drastischen Kostensenkungen bei der Code-Auslieferung. Da mit einer gehosteten Anwendung Änderungen schneller und mit weniger Fehlersuche zum Kunden gelangen, werden herunterladbare Änderungen unnötig. Desweiteren ergibt sich aus dem Liefermodell eine erweiterte Kundenbasis und regelmäßige Einnahmen [8]. Außerdem sinken die Konfigurationskosten für Anbieter, Kunden und Nutzer der Software. SaaS-Anbieter ziehen auch einen großen Vorteil daraus, wenn ihre Kunden Systeme nutzen, die sie selbst betreiben. Hauptvorteile für Kunden sind vor allem, dass SaaS-Anbieter bessere Möglichkeiten und Know-how bezüglich Sicherheit, Backups und Infrastruktur haben. Der Preis den Kunden dafür zahlen müssen liegt neben dem Geld auch in einem Vertrauensvorschuss in Bezug auf Sicherheit und betrieblichem Risiko gegenüber der SaaS-Anbieter.

Bevor im nächsten Abschnitt auf Microsofts S+S-Ansatz eingegangen wird, folgt eine Zusammenfassung von konventioneller SaaS.

- Anstatt Software innerhalb der eigenen Infrastruktur zu betreiben, bietet SaaS die Möglichkeit Software als Dienst von SaaS-Anbietern zu beziehen, die Eigentümer und Betreiber der Software sind.
- Der Anbieter betreibt aus Kundensicht eine einzige große Instanz der Software, deren Nutzungsrecht von allen Benutzern gemietet wird.
- SaaS erlaubt nutzerspezifische Anpassung der Benutzeroberfläche, Unternehmensregeln, -prozesse und Datenmodellschichten (multi tenancy).
- Änderungen können so sämtlichen Nutzern sofort verfügbar gemacht werden.
- Anstatt einer Lizenzgebühr wird bei SaaS für das monatliche Abonnement oder für den wirklichen Nutzen gezahlt.
- Die Teilnehmer nutzen SaaS-Anwendungen genauso wie die üblichen modernen Unternehmensanwendungen meist über eine Browserschnittstelle.

3 Microsoft S+S

Software-plus-Services oder besser Software-and-Services kommt als Trend, nicht allein bei Microsoft, eine große Bedeutung zu. Kernidee von S+S ist es die lokalen

Bestandteile zu minimieren und den Rest mittels Diensten umzusetzen, indem eine Kombination von Internetdiensten mit lokaler Client- und Server-Software oder Diensten angeboten wird.

In einer Welt in der sich oft zwischen Fat oder Thin Client – mobilem Gerät oder Desktop – lokaler Unternehmensinterner oder Webanwendungen entschieden werden muss, schlägt Microsoft eine Mischung aus beidem vor [4]. S+S ist hierbei nicht auf lokale Geräte beschränkt die lediglich das SaaS-Modell nutzen, sondern kann vielmehr als erweitertes SOA (serviceorientierte Architektur)-Modell gesehen werden, bei dem Unternehmen einen Teil ihrer Dienste auch lokal in der eigenen Infrastruktur und einen anderen Teil im Internet ausführen können. Abbildung 1 gibt hierzu einen Überblick zwischen Client-Server-Architekturen und der Client-Server-Service-Konfiguration. Der Unterschied zwischen Server und Service kann dabei in der Service-Delivery-Plattform (SDP) gesehen werden. Vereinfacht ist diese ein „Enterprise-Service-Bus (ESB) for the cloud“ [9].

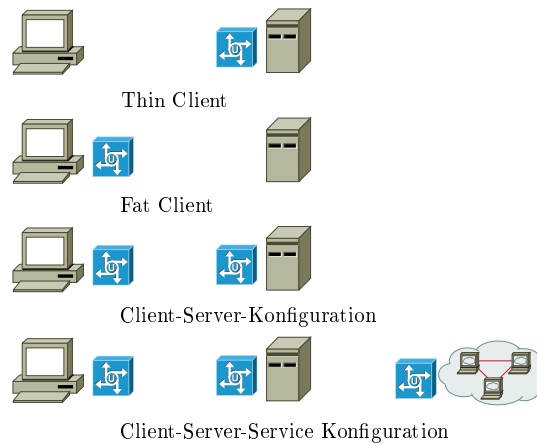


Abbildung 1. Überblick Client-Server-Service Konfigurationen

Die grundlegende Botschaft bei S+S ist, dass Mischarchitekturen gut sind. Das heißt, dass es sinnvoller ist die Welt als Ganzes zu betrachten und die Vorteile aller Teile zu vereinen, anstatt eine Standardlösung für alle anzubieten. Ein wichtiges Hilfsmittel dabei ist die Gewohnheitsneigung vieler Menschen zu nutzen, indem Dienste bei gewohntem, gleichbleibendem Frontend im Backend genutzt werden. Dadurch entsteht zunehmend die Anforderung Dienste wiederverwendbar in verschiedenen Systemen zu machen, sodass es egal ist woher der Dienst kommt – lokal, unternehmensintern oder global.

Wichtiger Aspekt dabei ist die Mobilität der Anwendungen, also eine Ausdehnung ihrer Funktionen auf die Endgeräte [4]. Das heißt, leistungsbezogene Anwendungen die Interaktivität und Offline Arbeiten ermöglichen und somit die bestmögliche Leistung eines Endgerätes bieten [4]. Mit S+S geht Microsoft

den Weg der Mitte, bei dem einige Lösungen als Dienst mit einer Clientseitigen Komponente gesehen werden, andere dagegen als Client-Software, mit Dienstkomponenten [4].

Als Entwurfsmuster ist S+S für eine Reihe von Lösungen geeignet. Zu denken ist hierbei z. B. an eine PC-basierte Clientanwendung zum Erstellen, Editieren und Verwalten von Daten. Eine internetbasierte Clientanwendung zum Suchen und zur gemeinsamen Benutzung und eine mobile Anwendung zum entfernten globalen Zugriff [4]. S+S erreicht durch eine integrierte Kombination von lokal installierter Software und gehosteten Diensten im Unternehmensnetz und im Internet einen größeren Kundennutzen vor allem für große Unternehmen [2].

Für ein besseres Verständnis geben die nächsten Abschnitte einen Überblick über die Ausgangslage und Zielsetzung sowie über Ausprägungen und Anwendungsformen von S+S.

3.1 Ausgangslage und Zielsetzung von S+S

Um S+S ein Gesicht zu geben beschreibt G. Carraro, Leiter SaaS-Architektur bei Microsoft, vier Welten, deren jeweilige Vorteile, Möglichkeiten und Architekturmuster S+S in sich vereint. Dies sind die Desktop-Welt, die Unternehmens-Welt, die Online-Welt und die Geräte-Welt [10].

Die Desktop-Welt zeichnet sich aus durch ihre reichen Möglichkeiten, die Nutzung lokaler Ressourcen und die Erweiterbarkeit der Systeme. Sie hat aber zum Nachteil, dass die Möglichkeiten lokal beschränkt sind.

Die Unternehmens-Welt hat hohe Qualitätsstandards und setzt auf Vertrags- und Service-Level-Agreement (SLA) -basierte Konzepte. Nachteilig ist hierbei, dass Unternehmen neue Konzepte oft nur langsam übernehmen und ihre Mobilität unter anderem durch Altlasten gebremst wird.

Die Online-Welt zeichnet sich durch ihre große Reichweite, ihre Gemeinschaft und ihre rasanten Neuerungen, sowohl technisch als auch geschäftlich aus. Nachteilig dabei ist, dass das Vertragswesen eher rudimentär veranlagt ist und Transaktionen nach dem „Ich gebe mein bestes“-Prinzip ablaufen.

Die Geräte-Welt ist geprägt von Mobilität, globalem (Daten)-Zugriff und dem Spaß-Faktor. Gerätespezifische Eigenschaften können hierbei die Möglichkeiten jedoch limitieren.

Das S+S-Konzept setzt auf einen Zusammenschluss dieser vier Welten zu einer gemeinsamen Welt und erreicht dadurch, dass spezifische Geräteeigenschaften besser genutzt werden können. Es vereint damit die Vorteile des Internet mit denen von Desktops und anderen Geräten, die Dienste oft als Hub nutzen [4]. Abbildung 2 verdeutlicht diesen Zusammenhang noch einmal und zeigt außerdem, dass S+S seinen Hauptansatz besonders im Bereich der Unternehmen und Endgeräte sieht, da diese vom herkömmlichen SaaS-Modell bisher nur ungenügend unterstützt werden.

Um den Nutzen von S+S zu verdeutlichen, wird im nächsten Abschnitt auf deren Ausprägungen und Anwendungsformen eingegangen.

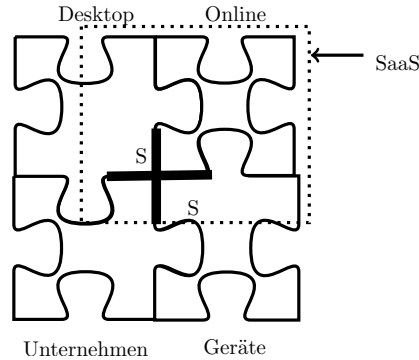


Abbildung 2. Ausgangslage für S+S – 4 Welten

3.2 Ausprägungen und Anwendungsformen von S+S

Neben der herkömmlichen Paketsoftware (lokale Software) sind unter dem Mantel von S+S viele Szenarien denkbar, aus denen sich aber drei grundlegende abheben. Diese lassen sich in Frontend-Dienste, Backend-Dienste, und Internet-Dienste einteilen. Zusätzlich ist es bei S+S möglich Dienste aus dem Internet als Unternehmensdienste in die Unternehmensinfrastruktur zu integrieren. Dieser Abschnitt erläutert diese fünf Szenarien und gibt in Abbildung 3 einen Überblick dazu.

Unter Paketsoftware werden Anwendungen innerhalb der eigenen Infrastruktur verstanden, auf die exklusiv von (internen) Benutzern zugegriffen wird. Sie werden im eigenen Haus installiert und betrieben und sind eher interessant für den Endnutzer. Ein Beispiel für diese Kategorie ist z. B. MS Exchange.

Unter Frontend-Diensten wird verstanden, dass lokale Software einen Internetdienst vervollständigt. Dies könnte zum Beispiel ein Outlook-basiertes Interfaces zu CRM-Live (Customer Relationship Management) sein. Dieses Modell kombiniert interaktives und bekanntes Benutzerverhalten mit den Vorteilen von SaaS (one-to-many SaaS)¹ [10]. Vorteile beim Entwurf eines Frontends für SaaS Anwendungen sind vor allem Anwendungspersonalisierung, Geschwindigkeit und Skalierbarkeit von Multi-Tier-Architekturen. Und auf Seite der Dienstsicht, funktionale Webdienste und Metadaten Webdienste für unterschiedliche Konsumanforderungen (Multi-Head-Anwendungen) [11].

Unter Backend-Diensten wird verstanden, dass ein Internetdienst eine lokale Software erweitert. Dies könnte zum Beispiel ein Anti-Spam / Anti-Phishing Dienst sein, der den lokalen Mailserver unterstützt. Dieses Modell erlaubt es verschiedene Dienste zu nutzen um die existierenden Systeme zu unterstützen (many-to-one SaaS)². Dies ist besonders interessant, wenn die Systeme Daten verwalten, die innerhalb des Unternehmens bleiben sollten, wie es z. B. in Ent-

¹ im Sinne von: Ein Dienst und viele (verschiedene) Frontends

² im Sinne von: Viele Dienste und ein System

wicklungsabteilungen der Fall ist. Vorteile beim Entwurf eines Backends als Dienst sind Möglichkeiten zum Entwurf erweiterbarer Web- und Smart-Clients mit Personalisierung, Offlineoperationen und Integration mit lokalen Ressourcen [11].

Unter Unternehmensdiensten wird schließlich die Nutzung externer Dienste innerhalb der Unternehmensinfrastruktur (many-to-many SaaS)³ verstanden. Dadurch lassen sich vertrauliche Daten vom Unternehmen selbst kontrollieren [2]. Mischlösungen sind dann zum Teil beim Kunden und zum anderen Teil beim Anbieter. Zum Beispiel ist eine Kombination aus traditionellem Ladengeschäft und Onlineshop, mit der Shop-Anwendung beim Anbieter und der lokalen Anwendungen vor Ort beim Kunden, denkbar [2].

Die ursprünglichen SaaS-Dienste können in diesem Zusammenhang als Internetdienste betrachtet werden, das heißt Dienstnutzung außerhalb der Unternehmensinfrastruktur. Sie sind analog zu kompletten Anwendungen, aber über das Internet mit dem SaaS Modell geliefert. Anwendungen die via SaaS geliefert werden, sind bisher vor allem interessant für Endkunden und kleine Unternehmen.

Um einen Eindruck vom Zusammenspiel der Dienste bei S+S zu bekommen folgt als Beispiel ein S+S Unternehmenssuchsystem.

3.3 Beispiel – Unternehmenssuchsystem

Als Beispiel für eine S+S Anwendung sei ein Unternehmenssuchsystem genommen, das in Abbildung 4 dargestellt ist. Um eine einheitliche Benutzeroberfläche zu bieten, wird es aus drei Diensten bestehen. Dies sind ein Internetsuchsystem eines SaaS Anbieters, ein Unternehmensinternes Suchsystem auf einem Unternehmensserver und ein Desktopsuchsystem⁴ für beliebige Endgeräte und Medien [5].

Als Internetsuchsystem nutzt das Unternehmen in diesem Beispiel den Web-Search-Dienst von Google. Yahoo stellt z. B. eine portable Variante BusinessSearch seines Suchdienstes zur Verfügung, welches das Unternehmen innerhalb seiner Infrastruktur als Unternehmensdienst nutzt um z. B. seine Entwicklungsdaten zu schützen. Desweiteren besitzt das Unternehmen Desktoprechner auf denen Frontend-Dienste zur Desktopsuche laufen und mobile Geräte, die eine solche Funktion nicht bieten. Als Unternehmenssuchsystem wird aus diesen Komponenten ein aufgesetzter Dienst nach dem Lego-Prinzip erstellt, der zentral als Dienst von den Clients genutzt wird. Somit lassen sich bei einheitlicher Benutzeroberfläche verschiedene Suchdienste kombinieren. Ein Beispiel aus dem Web für eine solche Kombination ist die Personensuchmaschine yasni.de.

4 Vergleich von SaaS und S+S

In diesem Abschnitt folgt der Vergleich der beiden vorgestellten Konzepte. Dabei wird insbesondere auf die Erfüllung von Anforderungen an Dienste und die

³ im Sinne von: Viele Dienste und viele Systeme

⁴ Hier ist lediglich die Funktion der (vernetzten) Desktopsuche ohne eigene Oberfläche gemeint.

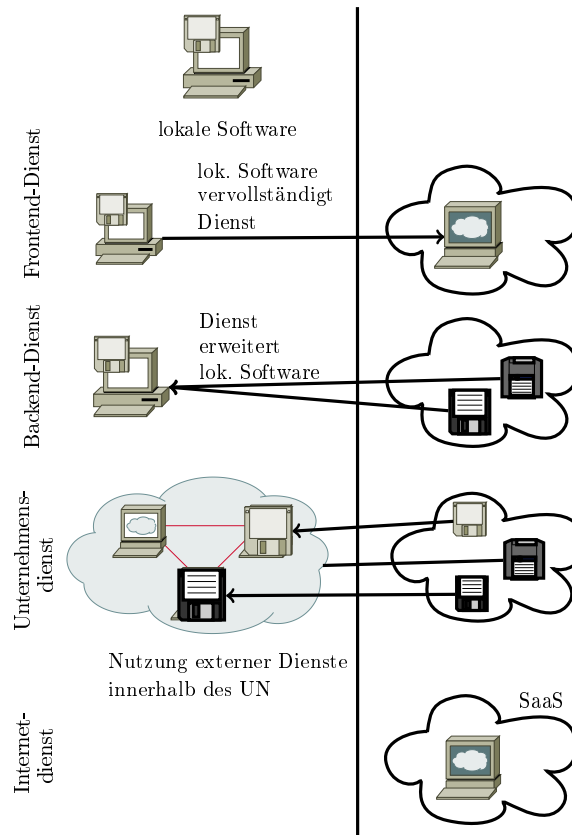


Abbildung 3. Überblick S+S Ausprägungen

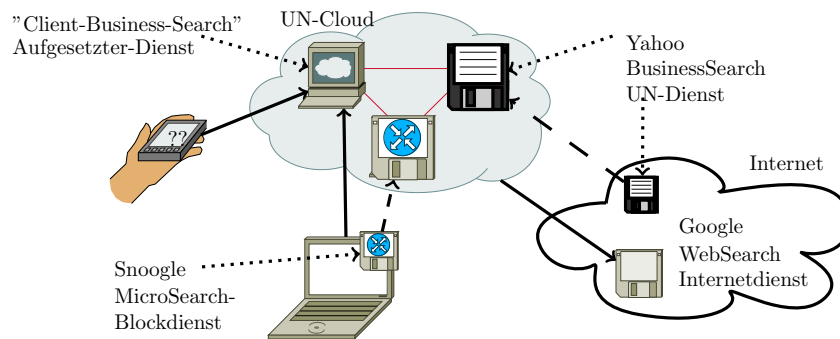


Abbildung 4. Beispiel einer S+S Konfiguration

Einordnung von S+S in die herkömmliche SaaS-Welt eingegangen. Anschließend folgt eine Verallgemeinerung von SaaS und S+S auf eine Software-Service-Welt sowie ein Vergleich mit den Variationen von SaaS.

Als Oberbegriff für SaaS und S+S schlägt Sangwell „Software-Services“ vor und verdeutlicht somit, dass Software traditionell als Paketsoftware (erstellt oder gekauft), als ein fertiger Dienst geliefert über das Internet (SaaS) oder als eine Mischlösungen von Paketsoftware und Internetdiensten (S+S) gesehen werden kann. SaaS bezieht sich also auf die Verwendung von Anwendungen, die aufgesetzte Dienste nutzen oder solche, die mit Blockdiensten erstellt wurden [3]⁵.

Im Gegensatz zu SaaS lagert der S+S Ansatz nicht die gesamte Software aus, sondern bietet verschiedene Wege, die entsprechende Funktionalität der Software zu nutzen. Das hat den Vorteil, dass der Benutzer zum einen nicht länger dienstgebunden ist und er bei gleichbleibender Oberfläche Anbieter und/oder Dienst wechseln, austauschen oder erweitern kann. Das heißt für die Architektur eine Zerlegung der Software in ihre funktionalen Bestandteile und ihre Benutzeroberfläche. Zum anderen verlagert das S+S Modell die Entscheidungsgewalt über Daten und Ausgestaltung der Softwaresysteme vom SaaS-Anbieter zurück zum Unternehmen. Denn aus rechtlichen und datenschutzrechtlichen Gründen ist es manchen Daten unmöglich das Unternehmen zu verlassen, sodass eine Verschiebung von Diensten in die Unternehmen notwendig wird. Dadurch können auch die Kosten von Support und Training weiter reduziert werden.

Die Nutzung wohldefinierter externer Dienste ist für große Unternehmen weniger interessant als die Nutzung eines fertigen (internen) Dienstes [3]. Der Großteil der Anwendungen die als Dienst über das Internet angeboten werden (SaaS) richten sich aber an Endkunden und kleine Unternehmen [3]. In großen Unternehmen sind die Anforderung aber komplett verschieden von denen der Endkunden. So treffen einige Annahmen des „Supporting the Long Tail“ nicht im Unternehmenskontext zu. Zum Beispiel müssen sich Endkunden nicht mit der Einhaltung von Standards oder Unternehmensanwendungsintegration beschäftigen [3].

Für das Verständnis von Unternehmenszielen ist wichtig das Auslagerung auf Kosteneffizienz abzielt. Das heißt eine Übertragung der Kosten und des Risikos einer Anwendung an einen Dritten gegen Bezahlung. Der Einsatz eines Softwaredienstes meint also ein gemischtes Projekt aus Beschaffung und Integration. Somit dient er der Bedürfnissbefriedigung wie z. B. der besseren Kundenverwaltung. S+S vereint dabei die Vorteile der beiden [3]. Auf die sich daraus ergebenden Anforderungen an Softwaredienste geht der folgende Abschnitt ein.

4.1 Erfüllung von Anforderungen an Dienste

Der Einsatz von Softwarediensten erfordert neue Anforderungen an das Identitäts- und Zugriffsmanagement, die Daten und die Operationen [3]. Fertige Dienste erfordern Erweiterungen der Prozesse, da S+S Anwendungen z. B. einen Backend-Dienst innerhalb der Firewall betreiben könnten. Wenn ein fertiger Dienst verschie-

⁵ die Begriffe werden in Abschnitt 4.2 erläutert.

dene Authorisationstufen hat, dann gibt es entweder die Möglichkeit jemanden innerhalb der Firma zu beauftragen diese auf die interne Mitarbeiterstruktur abzubilden – versteckte Kosten! – oder es wird das interne Rollenmodell auf das der externen Anwendung erweitert. Wenn der externe Dienst verschiedene Benutzeridentitäten verlangt, müssen die internen Prozesse dahingehend erweitert werden. Die Regeln der Dienstbenutzeraccounts müssen mit den Internen abgestimmt werden [3]. Aus Sicht der Unternehmensanwendungsintegration sollten Block- und aufgesetzte Dienste einfach zu integrieren sein, da sie dafür entwickelt wurden als Erweiterungen lokaler Anwendungen benutzt zu werden. Es ist aber dennoch eine Analyse von Anforderungen an Datenintegration erforderlich. Firewall-Regeln müssen diese Integration erlauben und eine Filterung der Anwendungsdaten bereitstellen. Außerdem sind Authentifizierungsmechanismen notwendig [3].

Für fertige Dienste ist es notwendig zu wissen, welche internen Anwendungen den Softwaredienst in welcher Form nutzen werden. Block- und aufgesetzte Dienste werden vermutlich eine lokale Backend Infrastruktur nutzen, die Hauptaugenmerk für Datenintegration und Sicherheit bietet. Der Austausch verschlüsselter Daten braucht eine PrivateKey-Infrastruktur die mit einigem Aufwand verbunden ist. Mit lokaler Backend Infrastruktur in S+S ist eine Verschlüsselung und Signierung einfacher da weniger Endknoten und somit eindeutige Schnittstellen existieren. Der Benutzer möchte gern vertraute Anwendungen weiterhin benutzen (u. a. Kostenaspekt für Schulungen). Daher haben Anbieter bereits begonnen ihre Anwendungen in große Anwendungen wie MS Office zu integrieren [3].

Der Betrieb der Software liegt bei SaaS im Verantwortungsbereich des Anbieters, das heißt aber HelpDesk und Schulungen sind notwendig, deren Qualität und Nutzen schwer zu kontrollieren ist [3]. Bei S+S hingegen liegt zwar die Dienstkomponente im Verantwortungsbereich des Anbieters, aber HelpDesk und Schulung können unternehmensintern geregelt sein.

Je wichtiger also die Anwendung für ein Unternehmen ist, desto mehr Folgen müssen berücksichtigt werden. Eine Reihe von Unternehmensanwendungen als fertigen Dienst in Anspruch zu nehmen ist daher ein großes Integrationsprojekt [3]. Dabei ist vor allem die Verwendung der Dienste wichtig, also welchen Client der Dienst nutzt. Falls er den Browser nutzt, müssen Fragen wie Versionsupdates, PlugIns, Sicherheitseinstellungen geklärt sein. Falls nicht der Browser genutzt wird, ist die interne IT verantwortlich für die Bereitstellung der Clients. Bei Block- und aufgesetzten Diensten wird entweder eine Verteilung der Backend Infrastruktur im Unternehmens-DataCenter, bei den Clients oder beides benötigt [3].

Für die Auswahl der Anbieter sind SLAs außerordentlich wichtig. Es ist darauf zu achten, was passiert, wenn der Anbieter das Geschäft aufgibt oder Daten einbehält um den Wechsel zur Konkurrenz zu verhindern. Es bleibt dabei immer eine Frage des Vertrauens, dass der Anbieter das Richtige tut, wenn das Schlimmste passiert [3]. Bei S+S kann zum einen bei Geschäftsaufgabe des Anbieters weiterhin produktiv gearbeitet werden, da lediglich auf die Dienstkomponente verzichtet werden muss. Des weiteren kann beim Vorhandensein weiterer Anbieter problemlos der Dienst gewechselt oder ein eigener Dienst entwickelt werden –

Funktionalität und Oberfläche bleiben dabei nahezu gleich. Der Datenaspekt entfällt bei S+S teilweise, da die Daten im Unternehmen verbleiben können.

Bevor der nächste Abschnitt S+S in die SaaS-Welt einordnet, gibt Tabelle 1 noch einmal einen Überblick über die Erfüllung von Anforderungen an Dienste.

	SaaS	S+S
Identitäts- & Zugriffsmanagement	interne Struktur auf Dienststruktur abbilden oder Dienststruktur auf interne Struktur abbilden	
Daten	bei Anbieter	bei Unternehmen
Sicherheit	PrivateKey Infrastruktur	lokale Backend-Infrastruktur (weniger Knoten, eindeutige Schnittstellen)
GUI	proprietär	gewohnte GUIs durch Integration
Support & Schulung	extern	intern
Abhängigkeiten	extrem Funktion & Oberfläche ändern sich bei wechselnden Diensten	marginal gleiche Oberfläche (evt. mit funktionellen Erweiterungen) bei wechselnden Diensten
Betriebsstart	sofort	verzögert

Tabelle 1. Erfüllung von Anforderungen an Dienste

4.2 Einordnung von S+S in die SaaS-Welt

S+S Dienste lassen sich in systemnahe Blockdienste und erweiterte aufgesetzte Dienste unterteilen. Blockdienste bieten hierbei Ressourcen mit systemnaher Funktionalität, die von Entwicklern für größere Anwendungen genutzt werden können. Sie sind also Internetdienste von denen Anwendungen erstellt werden können und somit vor allem interessant für Entwickler [3]. Ein Beispiel hierfür sind AmazonS3 und WindowsLiveAPIs.

Aufgesetzte Dienste bieten Ressourcen mit erweiterter Funktionalität in Verbindung mit Blockdiensten. Anwendungen nutzen diese aufgesetzten Dienste um ihre eigene Funktionalität zu erweitern – sie sind also Dienste mit eigenständiger Funktionalität und somit interessant für Entwickler und Administratoren [3]. Ein Beispiel hierfür ist WindowsLiveUpdate.

Abbildung 5 verdeutlicht noch einmal das S+S Dienste im Prinzip zwischen Paketsoftware und herkömmlicher SaaS einzuordnen ist, wobei die beiden S+S Dienste als ein gemeinsamer Block zu sehen sind. Die Abbildung bezieht sich dabei zum einen auf die Verwendung von Clientressourcen, die von links nach rechts abnehmen. Und zum anderen auf die Verwendung von Diensten, die von links nach rechts zunehmen.

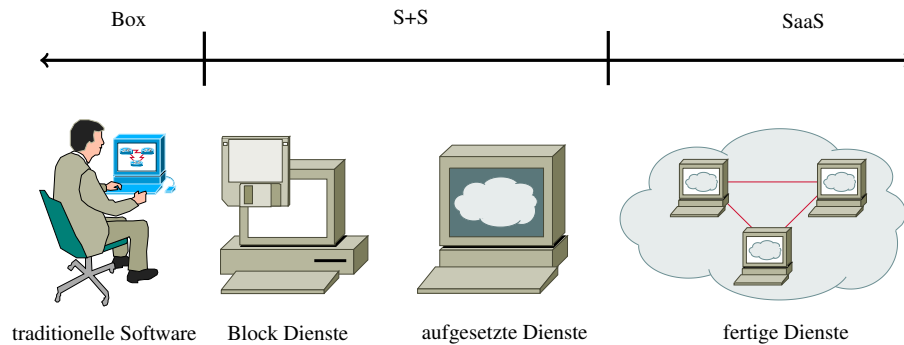


Abbildung 5. Einordnung von S+S in die herkömmliche SaaS-Welt

4.3 Verallgemeinerung von SaaS und S+S auf eine Software-Service-Welt

Dieser Abschnitt löst sich von der bisherigen Betrachtungsweise und verallgemeinert die vorgestellten Konzepte zu einer Welt von Software-Services. Software-Services sind dabei als Kompositionsmodell der SOA zu sehen, die S+S als Liefermodell nutzen. Für das Liefermodell S+S bietet sich folgende Unterteilung unter dem Oberbegriff Software-Services an, die in Abbildung 6 dargestellt ist.

- Software and no Service – die reine, ursprüngliche Paketsoftware
- Software and a Service – Mischform. Kombination von Software und Diensten nach dem Lego-Prinzip.
- Software as a Service – reine Dienstform. Die Software ist der eigenständige Dienst.

Kerngedanke bei Software-Services ist dabei Modularität der Software. Also portable Softwaredienste die je nach Anforderung auf den Endgeräten, unternehmensintern, im Internet läuft, oder Mischformen dieser Möglichkeiten bietet.

In der S+S Welt sind Lösungen weder reine SaaS Anwendungen noch reine Paketlösungen. Vielmehr sind sie eine Kombination lokaler Software und externer Dienste, die miteinander interagieren. Das heißt, lokale Software verbessert dabei Internetdienste und Internetdienste verbessern lokale Software [11].

S+S steht für eine Kombination von Internetdiensten und Paketlösungen, von der sich Microsoft verspricht, dass besonders große Unternehmen Paketlösungen und Clientseitige Software Lösungen nutzen werden und gute Erfahrungen mit dem wirksamen Einsatz von gehosteter und offline Software machen [12].

Indem SaaS-Anbieter ihre bestehenden Anwendungen portierbar machen oder neue Anwendung auf Portabilität ausrichten, können sie auch weiterhin am Markt bestehen und dazu außerdem ihren Kundenstamm erweitern. Das bedeutet, dass Verbraucher weder auf ihre Investitionen in Desktops, Server und lokale Umgebungen verzichten, noch werden sie die Vorteile gehosteter Dienste ignorieren, da sie kosteneffektiv und wartungsarm sind [12].

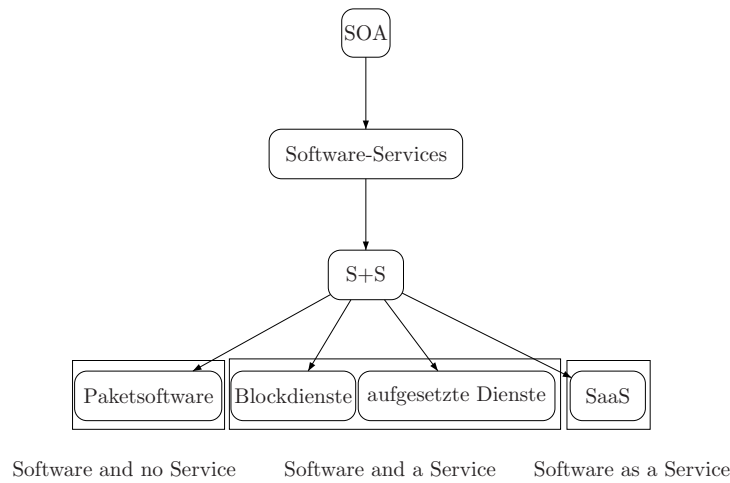


Abbildung 6. Verallgemeinerung von SaaS und S+S auf Software-Services

Nachdem nun eine neue Software-Service-Welt beschrieben wurde, ordnet der nächste Abschnitt ein paar grundlegende Begriffe aus der SaaS-Welt in die neue Sichtweise ein.

4.4 S+S im Vergleich mit Variationen von SaaS

Durch die Software-Service-Welt lässt sich der Begriffsdschungel rund um die Variationen von SaaS relativ einfach ordnen, da viele der entstandenen Varianten als Instanz des Software-Service-Modells gesehen werden können.

S+S stellt das Prinzip der Wiederverwendung von Funktionalität im Backend dar, die sich vielfältig zu neuer Software kombinieren oder in bestehende Software integrieren lassen. Das heißt jeder Kunde hat je nach seinen Bedürfnissen eine Mischung aus unternehmensinternen Funktionalitäten, lokaler Software und Internetdiensten. Dies ist die Welt der modularen Dienste.

Im Unterschied zu Application-Service-Providern (ASP) wird Software nicht für jeden Kunden separat entwickelt und modifiziert, sondern als Dienst je nach den Bedürfnissen in die Unternehmen integriert. Bei Verzicht auf die Integration in bestehende Unternehmenssoftware wird von Platform-as-a-Service (PaaS) gesprochen. Dieser Begriff wird von salesforce.com geprägt und bezeichnet ein Modell bei dem maßgeschneiderte Unternehmensanwendungen online entwickelt werden und diese ohne Clientsoftware auf den Endgeräten laufen kann.

Sind nicht in erster Linie große Unternehmen angesprochen, so wird von SaaS gesprochen. SaaS nutzt als Instanz vom S+S Liefermodell das Prinzip der Wiederverwendung im Frontend. Das heißt jeder Kunde hat bis auf bestimmte Personalisierungen die selbe Software und die gleiche Funktionalität verteilt über das Internet. Als Mashup-as-a-Service (MaaS) wird überwiegend der Zusam-

menschluss von SOA und dem Web2.0 verstanden. Dabei verschieben sich im Prinzip die Block- und aufgesetzten Dienste komplett ins Internet.

5 Zusammenfassung

Dieser Artikel gibt einen vergleichenden Überblick über Microsofts S+S – Software-plus/and-Services – und dem herkömmlichen Software-as-a-Service Ansatz. Dabei wird zunächst die herkömmliche SaaS als internetgebundene Mietsoftware herausgestellt. Diese nutzt im Prinzip das Internet als Verteilungsplattform für die Software um so sowohl Anbietern als auch Nutzern viele Vorteile und neue Möglichkeiten zu bieten.

Anschließend wird S+S mit seinen Bestandteilen und Ausprägungen vorgestellt. Das S+S Liefermodell umfasst dabei im wesentlichen die zwei Varianten lokale Installation der Software und lokal oder global gehostete Dienste, von denen Mischformen möglich sind [2]. Die Hauptanwendungen sieht Microsoft dabei vor allem in der unternehmensinternen Anwendung und der Vernetzung von Diensten und anderen Endgeräten.

Im anschließenden Vergleich von SaaS und S+S erfolgt eine Analyse der Anforderungen an Softwaredienste und eine Einordnung von S+S in die herkömmliche SaaS Welt. Dabei werden unabhängig von der Microsoft Sichtweise die wesentlichen Vorteile des S+S-Ansatzes auf die Welt der SOA übertragen. Hierbei stellt sich S+S als übergeordnetes Liefermodell für Software heraus, deren Ausprägungen sich nach der Art der Dienstmutzung unterscheiden. Die gegebenen Grenzen sind dabei natürlich fließend und werden mit zunehmender Verbreitung verschwimmen.

Durch diese Verallgemeinerung ergibt sich eine gute Möglichkeit den SaaS Begriffsdschungel etwas zu ordnen, da sich viele der Begriffe als spezielle Ausprägungen dieses Konzeptes erweisen.

Abschließend lässt sich sagen, dass Microsoft mit S+S nicht nur einen Weg sucht seine Dominanz aus dem PC-Sektor ins Web-Zeitalter herüberzuretten und einen weiteren Begriff in den SaaS-Dschungel wirft, sondern eher etwas Ordnung schafft und dabei einen großen Schritt in der Verbreitung SOA geht. In Zukunft wird es spannend sein, den weiteren Weg und die Verbreitung dieses Ansatzes zu verfolgen.

Literatur

1. Göldi, A.: Platform-as-a-Service: Das nächste IT-Schlachtfeld. (05 2008) <http://netzwertig.com/2008/05/14/platform-as-a-service-das-naechste-it-schlachtfeld/>.
2. o. V.: Software + Services von Microsoft: Partnerchancen. Microsoft Deutschland GmbH (August 2007)
3. Sangwell, K.: Implications of Software + Services Consumption for Enterprise IT. Number 12. (Oktober 2007) <http://msdn.microsoft.com/en-us/architecture/bb906058.aspx>.

4. Ozzie, R., Guthrie, S.: MIX '07 – Transcript of Keynote Remarks. (April 2007) <http://www.microsoft.com/presspass/exec/ozzie/04-30-07MIX.msp>.
5. Platt, M.: Software + Services: An Architectural Perspective. (August 2006) <http://msdn.microsoft.com/en-us/library/aa905319.aspx>.
6. Wilson, C., Josephson, A.: Microsoft Office as a Platform for Software + Services. Number 12. (Oktober 2007) <http://msdn.microsoft.com/en-us/architecture/bb906062.aspx>.
7. o. V.: Software-plus-Services Overview. <http://www.microsoft.com/softwareplusservices/softwarePlusServicesFullStory.aspx>.
8. o. V.: Software as a Service: Strategic Backgrounder. Software & Information Industry Association (Februar 2001) <http://www.sii.net/estore/ssb-01.pdf>.
9. Eliaz, T.: A walkthrough with SOA and S+S towards Agility. (10 2007) <http://seminaarit.codezone.fi/video/20071004/5/>.
10. Carraro, G.: A Planet Ruled by Software Architectures. Number 12. (Oktober 2007) <http://msdn.microsoft.com/en-us/architecture/bb906059.aspx>.
11. Carraro, G.: Gianpaolo's blog. (Mai 2008) <http://blogs.msdn.com/gianpaolo>.
12. Karnakota, S.: Moving Beyond SaaS Towards Software + Services - Part 1. (Dezember 2007) <http://softwareplusservice.spaces.live.com/blog/cns!CBEE7FF8FF7598FC!124.entry>.
13. o. V.: Software as a Service: Alphabet Soup. (November 2007) <http://www.longjumpblog.com/blog/2007/11/08/software-as-a-service-alphabet-soup/>.
14. Hall, M.: SaaS vs S+S. (April 2008) http://blogs.computerworld.com/saas_vs_s.s.
15. Chong, F., Carraro, G.: Architecture Strategies for Catching the Tail. (April 2006) [http://msdn.microsoft.com/de-de/architecture/aa479069\(en-us\).aspx](http://msdn.microsoft.com/de-de/architecture/aa479069(en-us).aspx).
16. Natis, Y.: Introducing SaaS-Enabled Application Platforms: Features, Roles and Futures. Gartner (August 2007) <http://mediaproducts.gartner.com/reprints/oracle/150447.html>.

Onlinequellen zuletzt am 23.05.2008 geprüft.