

# *manage it*

[[ IT - S t r a t e g i e n u n d L ö s u n g e n ]]

## **Die vierte Generation**

Darwin und die Portale

## **Identity Management**

Kontrolliert zugreifen

## **IT-Infrastruktur**

Fort Knox und der Mainframe

## **UMTS und IP-basierte Anwendungen**

Das mobile Büro



Geschäfts- und IT-Strategie vereinen

# Die Verschmelzung

Nutzen Sie unser Angebot für  
Sonderdrucke oder E-Publishing-Dateien  
von Artikeln dieser Ausgabe

Tel.: +49 8092 87543

Service Oriented Architecture

# Der perfekte Unterbau

iel einer Service Oriented Architecture (SOA) ist eine hochflexible Applikations-Infrastruktur, die an wechselnde Anforderungen anpassbar ist. Die Technologie wiederverwendbarer Services ist aber nur der Anfang. Der Mehrwert entsteht erst durch die effektive Steuerung der darauf aufbauenden Geschäftsprozesse.

Das grundlegende Modell einer Service- oder auch Komponenten-orientierten Architektur scheint auf den ersten Blick so neu nicht. Bisherige, weitgehend proprietäre Lösungsansätze konnten jedoch das Versprechen einer flexiblen Architektur nicht einlösen. Konzeptionell sind in einer Komponenten-orientierten Architektur die wesentlichen Funktionen einer Applikation als Softwareeinheiten organisiert, die über Schnittstellen für andere Systeme zugänglich gemacht werden.

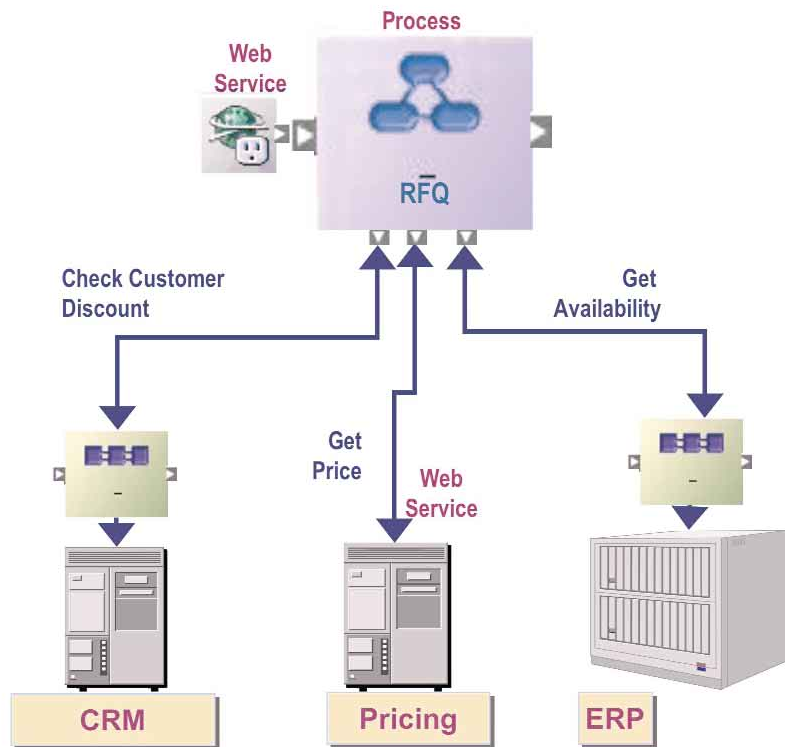
Als typische Vertreter solcher frühen SOA-Ansätze gelten Microsofts Komponentenmodell COM/DCOM oder auch der objektorientierte Kommunikationsstandard CORBA. Beide verbuchten jedoch kaum mehr als Teilerfolge. Während CORBA zu komplex war, die Objekte zu stark miteinander verzahnte und viele Hersteller ihre »eigenen« CORBA-Varianten erstellten, konnte COM/DCOM letztendlich die Hürde der Plattformabhängigkeit nicht nehmen.

Im Vergleich zu DCOM und CORBA versprechen Web-Services den Aufbruch zu neuen Horizonten. Auf Grund allgemein akzeptierter Internetstandards ermöglichen sie auch die Integration von Applikationen, die früher nur schwer einzubinden waren. Das Prinzip der Web-Services ist einfach: Ein Service-User (Client) fordert von einem Service-Provider (Server)

Dienste an, die dieser durch die Abarbeitung bestimmter Funktionen bereitstellt. Wichtig dabei ist, über welche Eigenschaften Web-Service-User und Web-Service-Provider verfügen müssen, damit beide sich verstehen.

Für die Kommunikation zwischen beiden führt das Modell der Web-Services Funktionen ein, damit der Server seinen Dienst über die XML-basierte Beschreibungssprache WSDL (Web Service Description Language) zur Verfügung stellt. WSDL kümmert sich – unabhängig vom darunter liegenden Kommunikationsprotokoll – um die Definition einer strukturierten Form der Kommunikation. Die WSDL-Spezifikation geht von einer abstrakten Interaktion zwischen Service-Provider und Service-User aus, bei der Provider und User Nachrichten übermitteln. Jede Message verfügt über strukturierte Inhalte. Einzelne Übertragungen oder auch einzelne Nachrichten haben die Form von Operationen – konkretisiert als Funktionsaufruf mit Rückgabewert.

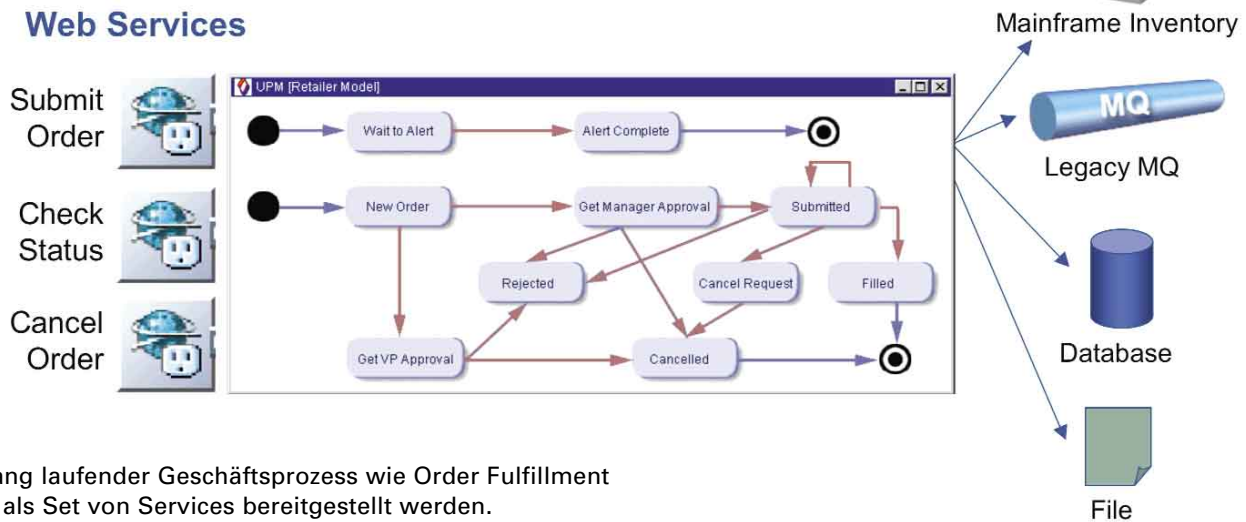
**Eine serviceorientierte Infrastruktur ist mehr als Web-Services.** Web-Services liefern damit eine geeignete Technologie, um serviceorientierte Ar-



Der Geschäftsprozess RFQ (Request for Quotation; die Aufforderung, ein Angebot abzugeben) greift im Rahmen einer SOA-Architektur auf vorhandene ERP- und CRM-Applikationen zurück und verpackt sie zu einem neuen Service.

Quelle: Vitria

## Order Fulfillment Prozess (lang laufend)



Ein lang laufender Geschäftsprozess wie Order Fulfillment kann als Set von Services bereitgestellt werden.

chitekturen zu realisieren. Die Gleichung »SOA = Web-Services« ist jedoch unzulässig. Denn serviceorientierte Architekturen lassen sich auch mit anderen Technologien wie CORBA oder Enterprise Java Beans implementieren.

Eine SOA ist weniger eine Technologie, sondern vielmehr ein konzeptionelles Modell zur effizienten Realisierung komplexer IT-Infrastrukturen. Dieser umfassendere Ansatz lässt sich an Hand einiger Kernmerkmale beschreiben:

- ⌈ Von einem Service (einer Softwareeinheit) existiert jeweils nur eine Instanz.
- ⌈ Services sind untereinander lose gekoppelt. Damit sie mit anderen Komponenten effizient zusammenarbeiten können, müssen sie »eigenständig« sein.
- ⌈ Der Nachrichtenaustausch zwischen Services sollte so einfach wie möglich und deskriptiv sein sowie XML-Dokumente und XML-Schemata nutzen; diese sind deutlich flexibler als herkömmliche Remote Procedure Calls.
- ⌈ Wo immer möglich, sollten Services zusätzlich zum synchronen auch asynchrones Messaging beherrschen, um so langlaufende Geschäftsprozesse zu unterstützen.

In mancherlei Hinsicht erinnert eine SOA an den industriellen Fertigungsprozess. Insbesondere im Automobilbau setzen Unternehmen bei der Produktplanung und -entwicklung auf die System- und Modulbildung. Als Module gelten komplette und funktionsfähige Komponenten, die einbaufähig geliefert werden. Ein Bremssystem, die Klimaanlage oder der Fahrzeugantrieb entstehen, indem Funktionen aus meh-

renen Baugruppen, Komponenten und Modulen zusammenwirken. Derartige Standardkomponenten – die ja auch typisch sind für weite Teile einer Service-orientierten Architektur – lassen sich auftragsneutral vorfertigen. Die auftragsabhängigen Bausteine (beim Automobil beispielsweise die Ausstattung, die Farbe und das Zubehör) müssen abhängig von bestimmten Ereignissen individuell geordert werden.

### Web-Services: unabhängig von Plattformen und Programmiermodellen

Einer der großen Vorteile von Web-Services besteht darin, dass sie unabhängig von Programmiersprachen sind. In dieser Sichtweise ähneln sich die Zielsetzungen von Web-Services und CORBA (Common Object Request Broker Architecture). Beide befassen sich mit verteilten Applikationen. Während CORBA auf die Interface Definition Language (IDL) setzt, bauen die Web-Services auf das XML-basierende SOAP (Simple Object Access Protocol) auf. Allerdings müssen die im Rahmen von CORBA implementierten Dienste bereits während der Entwicklung einer Applikation hart kodiert werden, die SOAP-Aufrufe dagegen erfolgen erst zur Laufzeit. Dies macht SOAP weit flexibler. Solange ein Web-Service die SOAP-Message (das XML-Dokument) erfassen kann, spielt es für ihn keine Rolle, auf welcher Plattform der Service entwickelt wurde und nun abläuft. Auch wenn CORBA in komplexen Anwendungsszenarien noch immer leistungsfähiger als Web-Services ist, hat es sich wegen seiner Schwerfälligkeit nie als breit unterstützter Standard durchgesetzt. Während heute nahezu jeder Anbieter von Business-Software die wichtigsten Web-Services-Protokolle unterstützt, bietet kaum einer standardisierte CORBA-Schnittstellen, um seine mit anderen Applikationen zu verbinden.

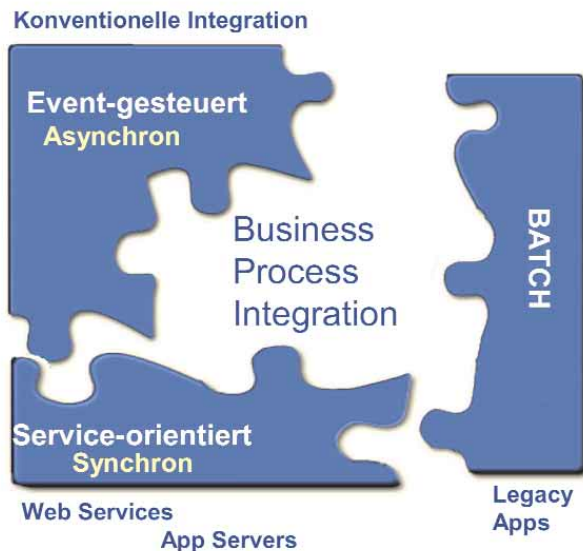
Solch eine ereignisgesteuerte Konstellation lässt sich recht gut mit einer asynchronen, lose gekoppelten SOA abbilden und letztlich auch steuern. Voraussetzung dafür sind Methoden und Verfahren, um die Nachricht (eine Message) vom Auftreten eines bestimmten Events zu veröffentlichen beziehungsweise empfangen zu können –

mit dem Fachterminus: Publish and Subscribe.

Eine derartige Nachrichteninfrastruktur wie sie die Integrationsplattform Vitria:BusinessWare mit ihrer Communicator-Komponente bietet, sorgt für eine Übertragung von Daten in einem konsistenten Format über Kommunikationskanäle (Channels).

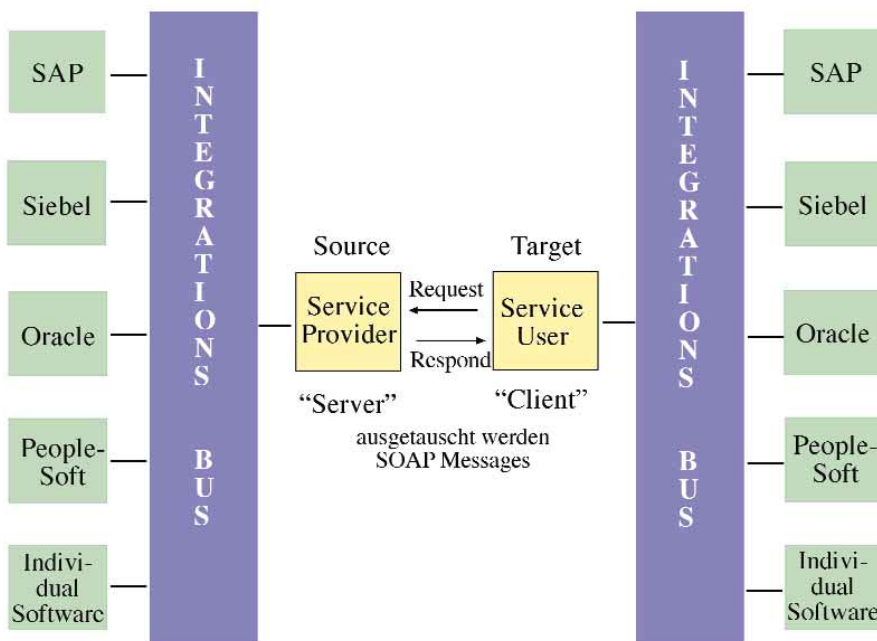
Als Backbone für die Nachrichtenübertragung kommt es auf eine hohe Servicequalität durch Service Level Agreements (SLAs), garantierte Leistung, vollständige Transaktionsverarbeitung, Persistenz der Daten und geordnete Datenwiederherstellung an. Durch eine zielgerichtete Erweiterung werden so aus Integrationsplattformen, die ihre Praxistauglichkeit bereits im EAI-Umfeld unter Beweise gestellt haben, moderne serviceorientierte Architekturen.

Ein ereignisgesteuertes Modell innerhalb einer SOA sollte langlaufende Transaktionen unterstützen, bei denen im zeitlichen Ablauf von Geschäftsprozessen wie einer Auftragsbearbeitung eine Vielzahl von Events zu analysieren und miteinander zu verknüpfen sind. Restümee: Eine serviceorientierte Architektur baut in vielerlei Hinsicht auf den EAI-Fundamenten auf und erweitert sie um moderne Technologien wie Web-Services. Die synchrone, häufig serviceorientierte Einbindung von Web-Services wird dabei um eine asynchrone Ereignis-gesteuerte Integration ergänzt.



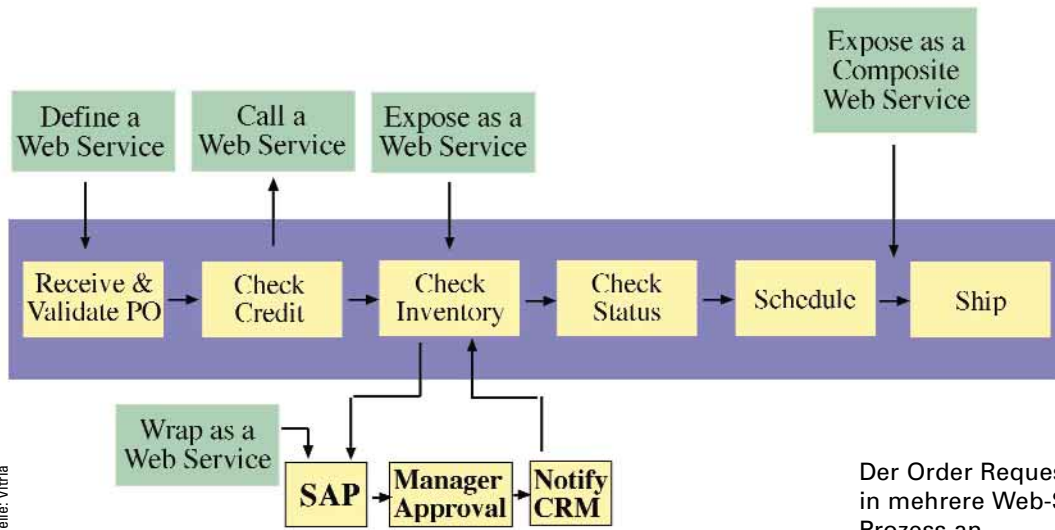
Quelle: Vitria

Business Process Management ermöglicht die Einbindung verschiedener Architekturmodelle: asynchron, synchron und Batch-orientierte.



Quelle: Vitria

Das konzeptionelle Modell von Web-Services, bei dem ein Client die Services einer anderen Applikation aufruft.



Der Order Request eines Client stößt einen in mehrere Web-Services untergliederten Prozess an.

Die dazu notwendigen Techniken und Methoden stellt ein Integrationsserver wie Vitria:BusinessWare bereit. Gerade die Kombination aus Service- und Message-basierter Integration liefert den Mehrwert. Web-Services bieten sich hier als hervorragende Lösung für Fragen rund um Low-Level-Connectivity-Probleme an. In dieser Perspektive stellt eine serviceorientierte Architektur ein Umdenken in der Entwicklung von Anwendungen dar. Die Schaffung von Softwarekomponenten, die als Dienste über ein standardisiertes Interface universell eingesetzt werden können, bietet eine Reihe von Vorteilen.

Eine der zentralen Punkte ist die Flexibilität beim Aufbau einer Integrationslösung durch Konzentration auf den Geschäftsprozess. Durch die Kapselung einer bestehenden Systemlandschaft als Service ist zugleich ein höherer Grad an Wiederverwendung gewährleistet. Zugleich ergibt sich damit die Möglichkeit, Änderungen im Organisationsablauf von Prozessen schneller umsetzen zu können. Um weitgehend standardisierte Integrationsaufgaben zu implementieren, bietet sich der Einsatz vorkonfigurierter Lösungen an. Beispiele dafür sind Straight Through Processing für die Wertpapierabwicklung, Integrated Order Management

für Telekommunikationsanbieter oder Lösungen für die Abwicklung von Schadensmeldungen in der Versicherungsbranche. Abgerundet wird eine umfassende SOA, die sich nicht nur auf den Einsatz Web-Services beschränkt, durch die Nutzung einer Geschäftsprozessautomatisierung und -steuerung, die unabhängig von den ausführenden Systemen ist. Durch diese Entkopplung wird eine höhere Flexibilität bei der Umsetzung von Integrationslösungen erreicht.

*Thomas Egeling*

Thomas Egeling ist Systems Engineer bei Vitria Technology in der Geschäftsstelle München.

3 Ausgaben

# Einblick Durchblick Ausblick

f ü r d r e i z e h n f ü n f z i g !



[ ] **Ja**, ich bestelle drei Ausgaben » *manage it* « zum Preis von Euro 4,50 pro Ausgabe. Dieses Probeabonnement verlängert sich nicht automatisch.

Schicken Sie diesen Coupon an:

**ap Verlag GmbH  
Postfach 1380  
85554 Ebersberg**

oder faxen Sie die Seite einfach an die Nummer

**+49 8092 87544**

Titel: \_\_\_\_\_

Vorname: \_\_\_\_\_

Nachname: \_\_\_\_\_

Position: \_\_\_\_\_

Firma: \_\_\_\_\_

Straße: \_\_\_\_\_

PLZ: \_\_\_\_\_ Ort: \_\_\_\_\_

E-Mail: \_\_\_\_\_

Telefon: \_\_\_\_\_

Fax: \_\_\_\_\_