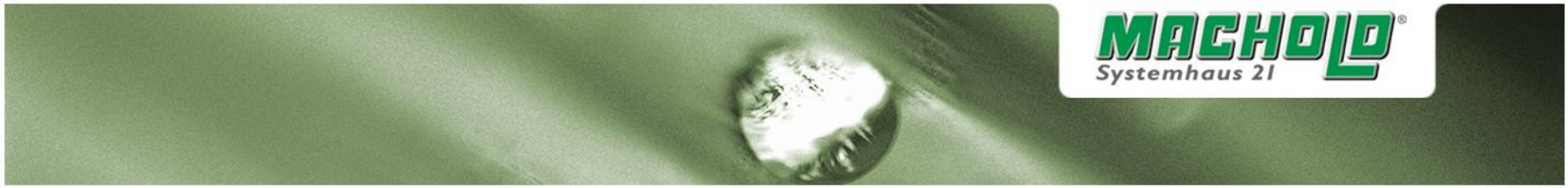
Einfache Mainframe-Integration in .NET/Java/SOA – es geht wirklich

Praxisbeispiele und LIVE-Vorführung mit Ivory

Axel Rittershaus, Machold Systemhaus 21, Stuttgart

21. VM/VSE IT-Leiter Kolloquium 2006



Vielen Dank für die Einladung an

Herrn Michael Sieger, becom

und die Kollegen der IBM

Vorgeschichte

- Als Dienstleister mit über 60 Projektleitern und Entwicklern, kamen unsere Kunden auf uns zu, um ihnen bei der Lösung der Anbindung von Großrechner-Systemen an moderne Technologien zu helfen.
- Die Anforderungen:
  - Standard-Software
  - Einfache und schnelle Entwicklungsmöglichkeit zur Anbindung
  - Möglichst keine Veränderung des bestehenden Codes
  - Möglichst keine Middleware
  - Vorhandene Sicherheit muss beibehalten werden

Vorgeschichte

- Nach einer umfassenden Marktrecherche war von einer handvoll Lösungen noch eine übrig geblieben:

- Hersteller: GT Software, Atlanta, GA, USA – seit über 20 Jahren Hersteller von Mainframe-Lösungen mit über 2000 Installationen weltweit

Ausgangslage

- Der seit den neunziger Jahren propagierte schnelle Tod der Mainframes hat sich nicht bewahrheitet
- Viele geschäftskritische Systeme werden weiterhin unter VSE, z/OS, etc. betrieben und Millionen von COBOL-Programmen arbeiten zuverlässig
- Die Anforderungen an IT-Systeme verändern sich rasch und „moderne“ Technologien (heute Java/.NET, morgen ??) scheinen dafür die Lösung zu sein
- Die transparenten Kosten für Mainframes scheinen erheblich höher als die intransparenten Kosten für Unix, Linux, Windows
- Das Mainframe-Know how nimmt ab

Zukunftsoptionen

Option 1 – Alles ersetzen, „big bang“ und Abschalten des Mainframes

- Ersetzen sämtlicher bisher Mainframe-basierter Lösungen durch
  - Standardsoftware (u.a. SAP als Glücklichmacher)
  - Neu implementierte Java/.NET/xyz-basierte Lösung
- „Alles wird neu, schön, bunt, besser zu bedienen, …“
- Höhere Flexibilität bei entsprechender Implementierung
- Lange Projektlaufzeit
- Hohes Risiko bzgl. Abbildung der Funktionalität, Performance, „überraschender“ Abhängigkeiten, …
- Bestehende Anwendung über x Jahre entwickelt und (nahezu) fehlerfrei, neue Anwendung braucht noch etwas „Reifezeit“

Zukunftsoptionen

Option 2 – Alles bleibt wie es ist

- Mainframe-Anwendung als 3270 bleibt erhalten
- Anwendung ist sicher, erprobt, stabil, „berechenbar“
- Flexibilität zur Anpassung an neue Anforderungen nur eingeschränkt gegeben

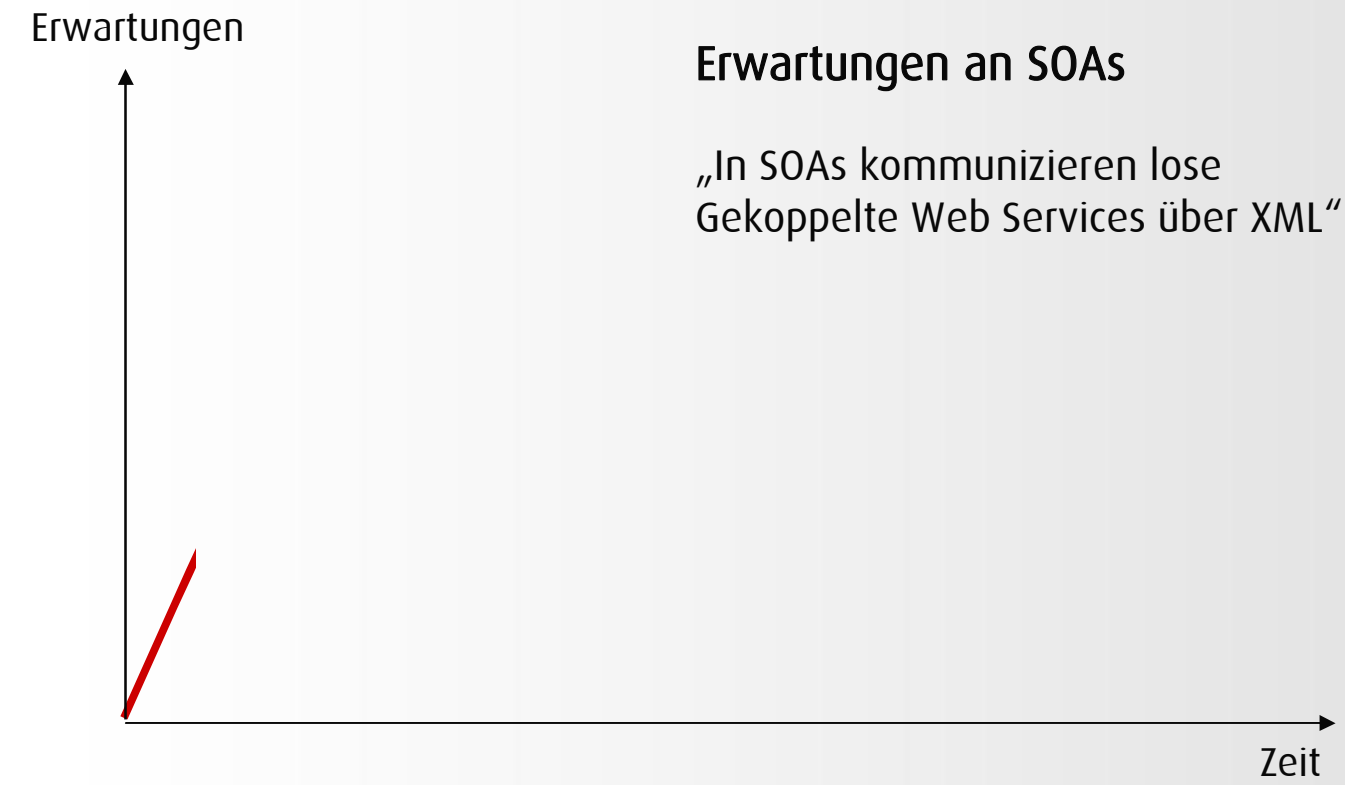
Zukunftsoptionen

Option 3 – Koexistenz und Kombination der besten Ansätze

- Bestehende Geschäftsvorfälle auf Großrechner belassen
- Bestehende Geschäftslogik auch anderen Technologien zur Verfügung stellen
- Neue Benutzeroberflächen in neuen Technologien implementieren, Logik auf Host und/oder neuer Application Server-Plattform realisieren
- Risiko minimieren
- Flexibilität dort aufbauen, wo sie benötigt wird

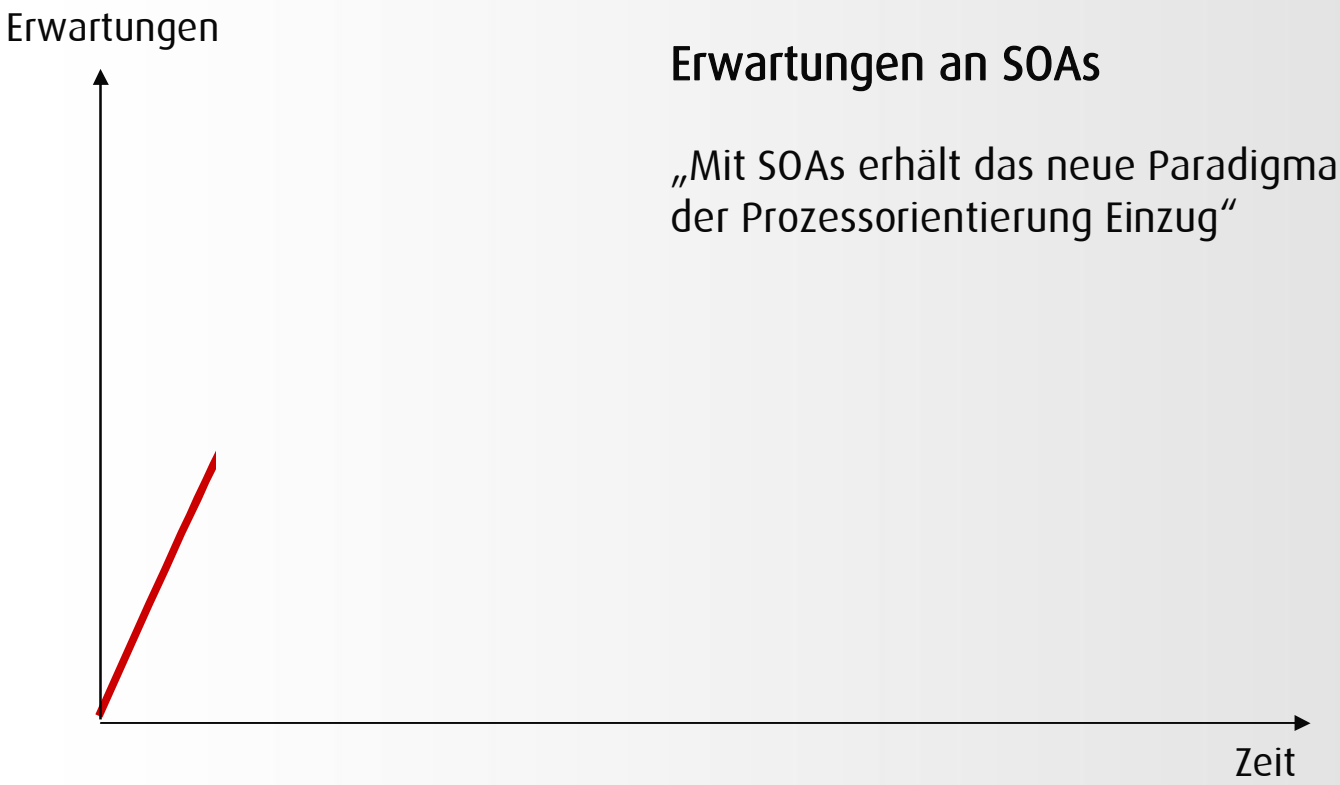
Die Lösung von heute für die Modernisierung

SOA – Service Oriented Architecture



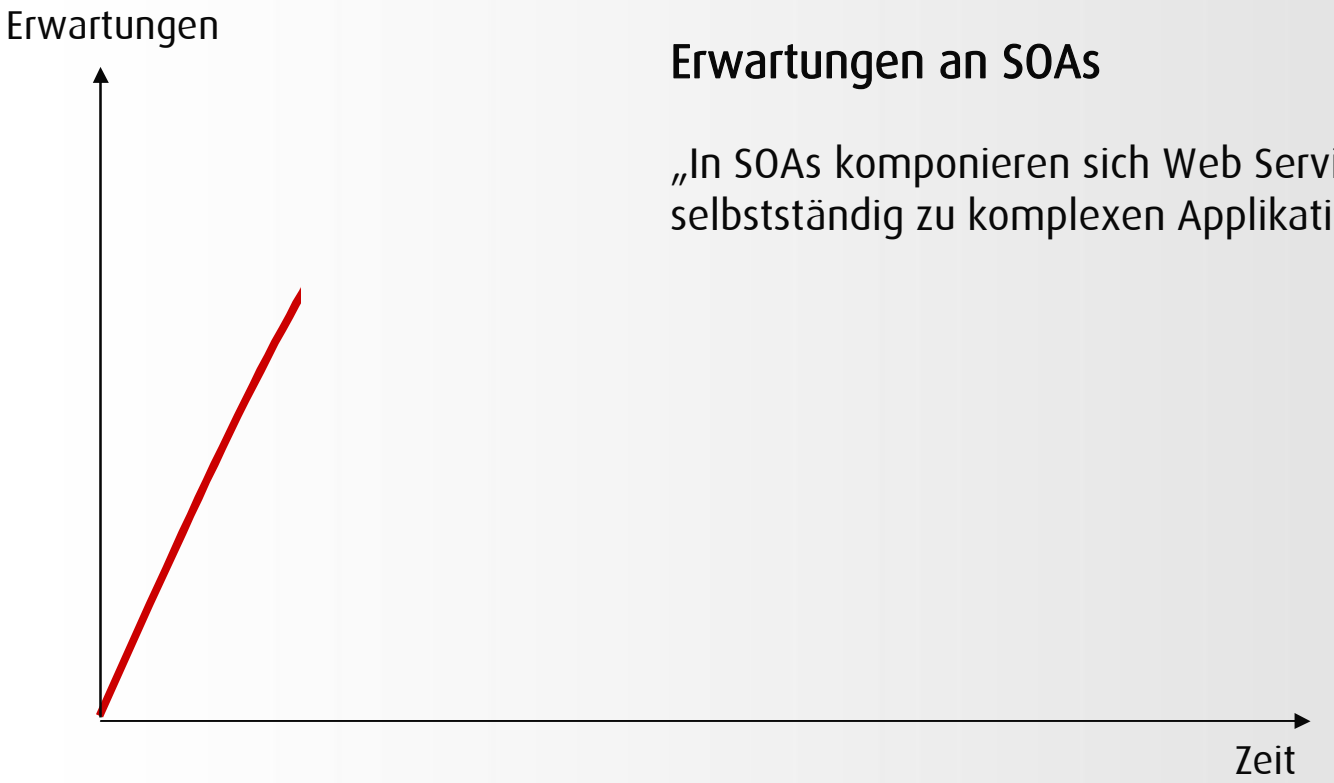
Erwartungen an SOAs

„In SOAs kommunizieren lose
Gekoppelte Web Services über XML“



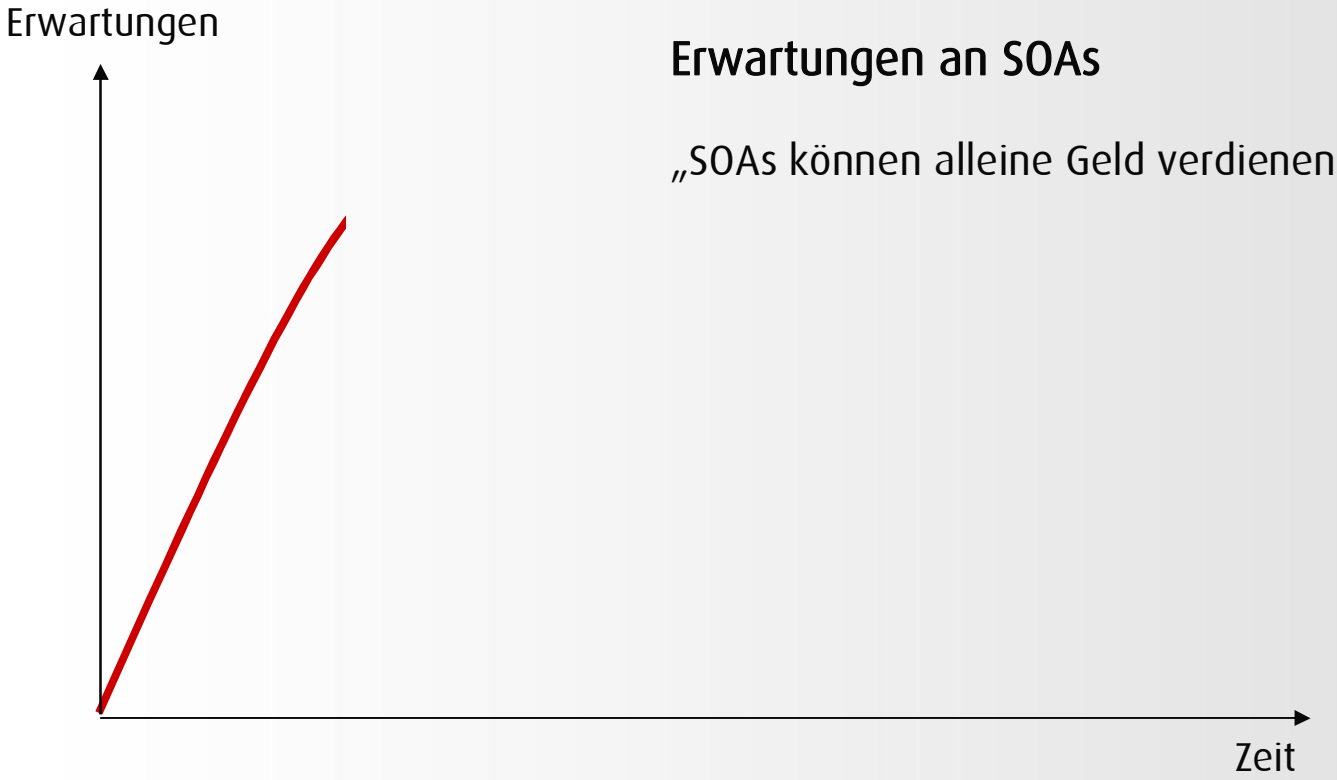
Erwartungen an SOAs

„Mit SOAs erhält das neue Paradigma der Prozessorientierung Einzug“



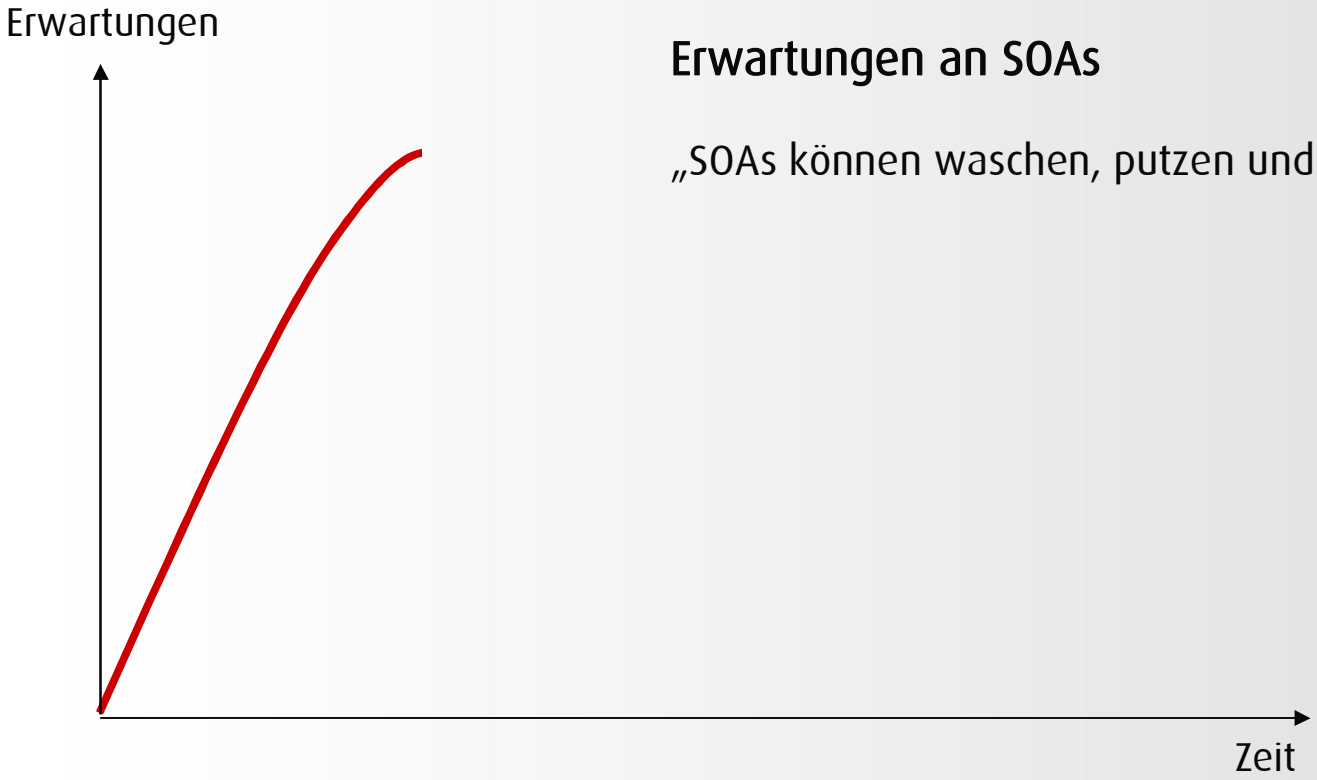
Erwartungen an SOAs

„In SOAs komponieren sich Web Services selbstständig zu komplexen Applikationen“



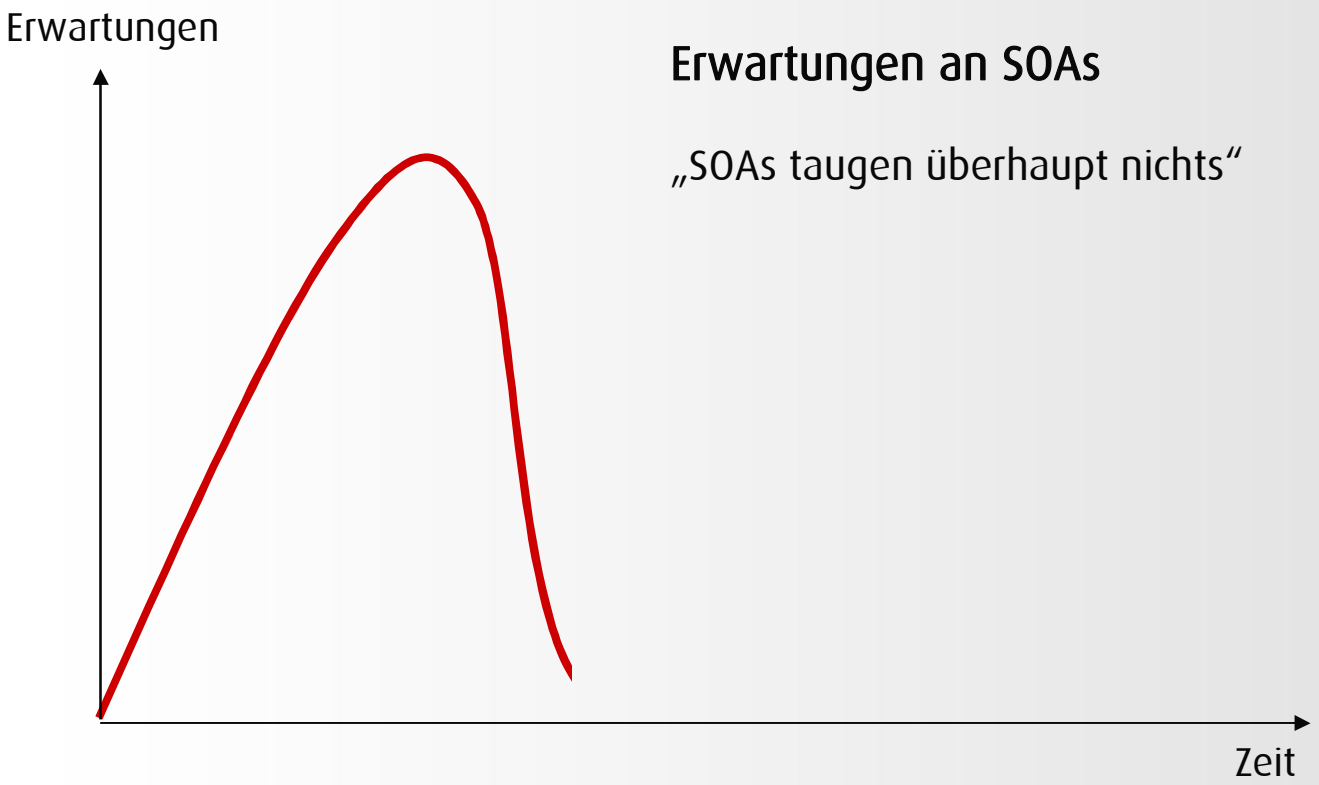
Erwartungen an SOAs

„SOAs können alleine Geld verdienen“



Erwartungen an SOAs

„SOAs können waschen, putzen und bügeln“

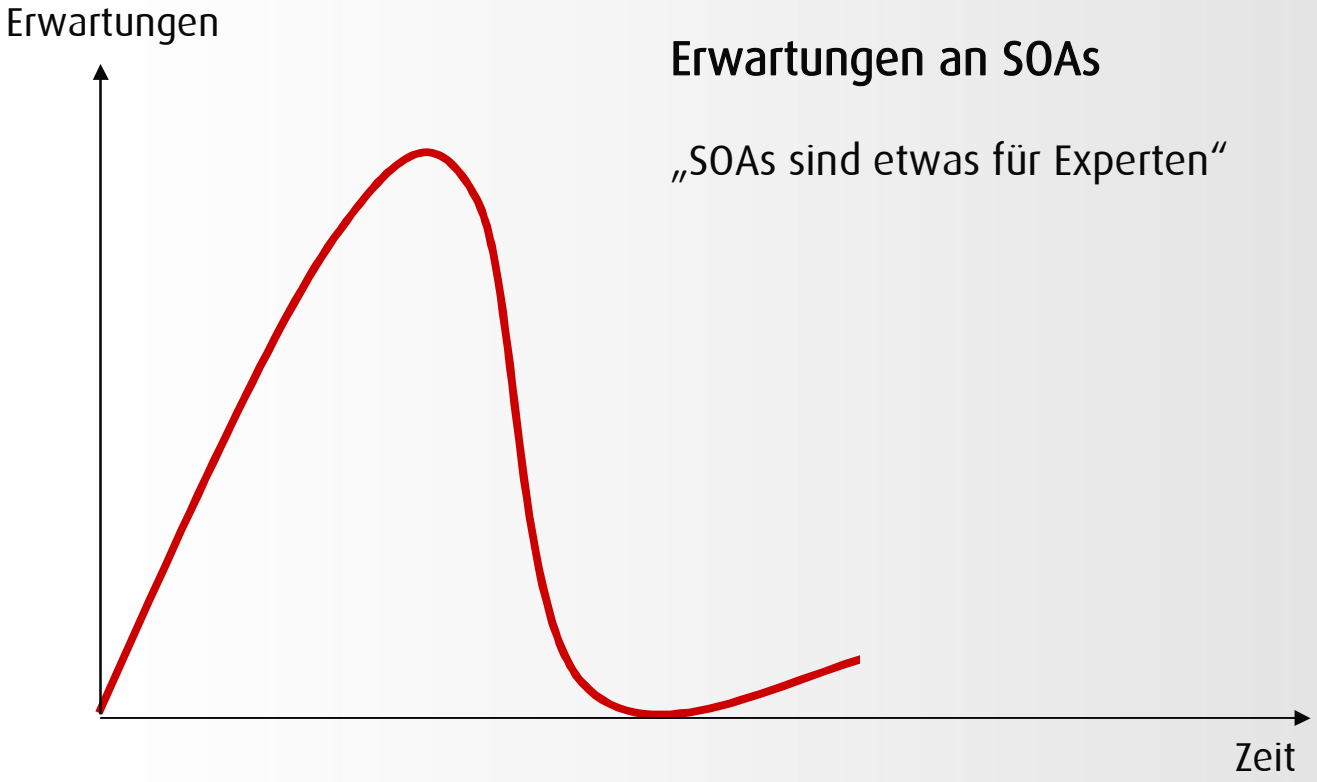


Erwartungen an SOAs

„SOAs taugen überhaupt nichts“

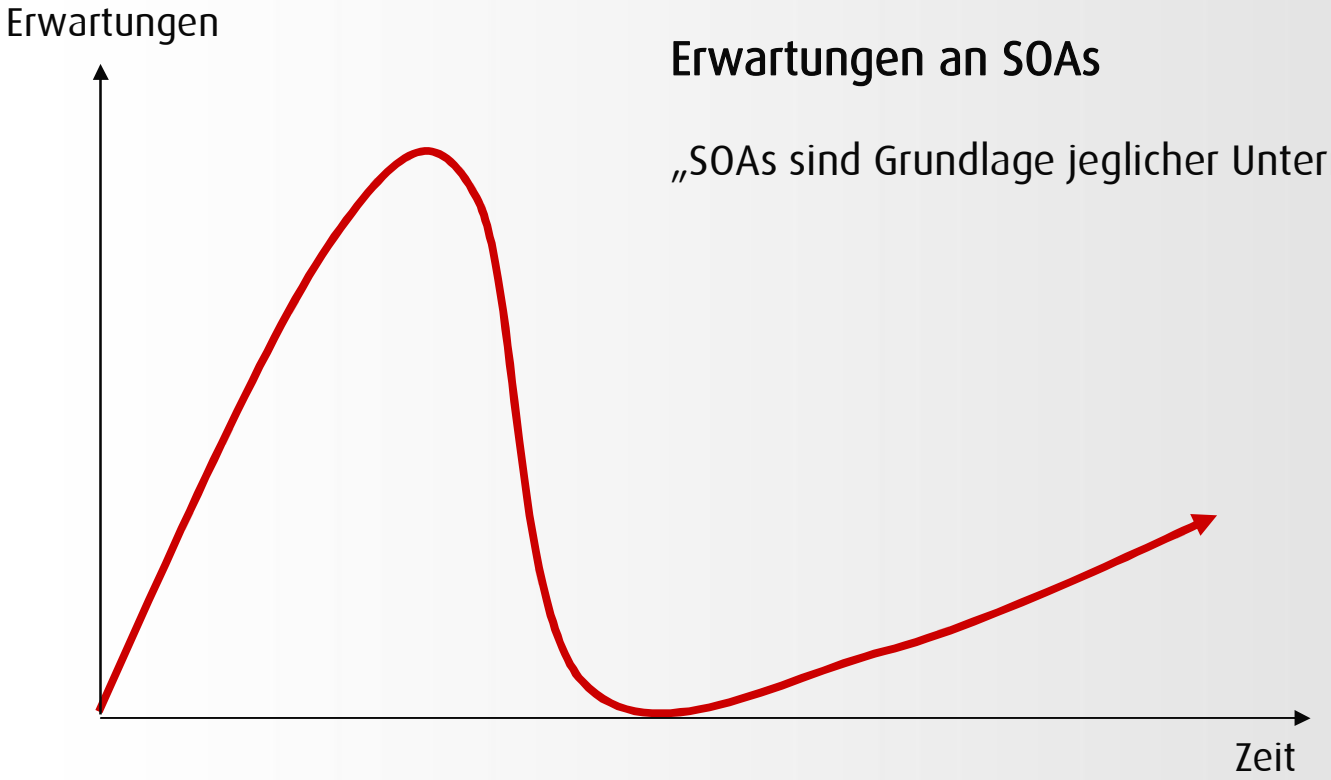
Erwartungen an SOAs

„SOAs sind etwas für Experten“



Erwartungen an SOAs

„SOAs sind Grundlage jeglicher Unternehmenssoftware“



SOA – die Idee

- Lose Koppelung
- Konzentration auf Services/Geschäftsvorfälle und -teilprozesse anstatt auf komplexe Module
- Steigerung der Flexibilität
- Wiederverwendbarkeit
- Komplette Entkoppelung der Logik von der Darstellung
- Nutzung eines Services in verschiedenen „Welten“ (z.B. .NET und Java)
- Plattformunabhängigkeit
- Basiert auf allgemein anerkannten Standards (http, XML, SOAP,…)

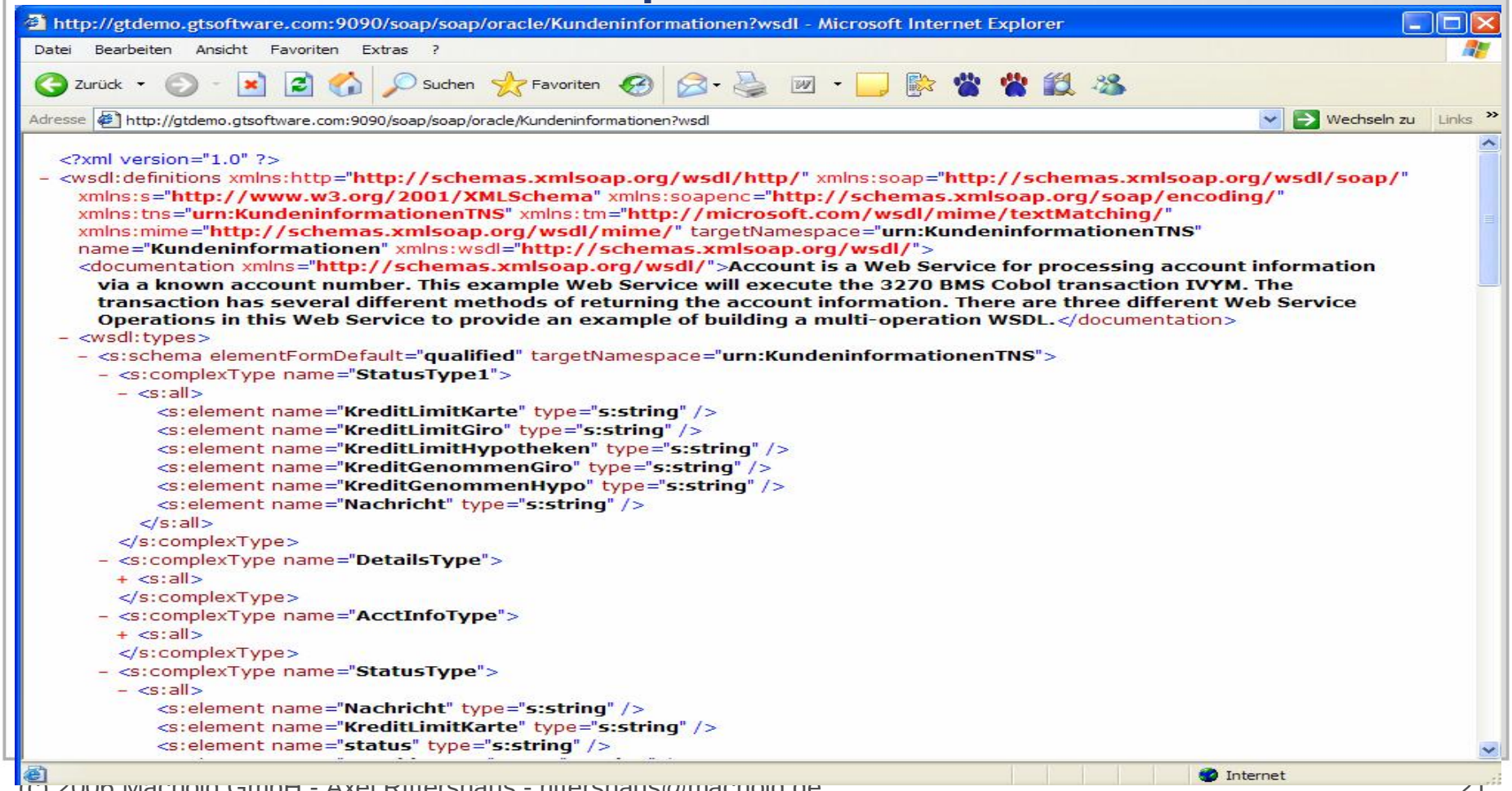
Das Ziel

- Sicherung bestehender Investitionen
- Steigerung der Reaktionsfähigkeit der IT-Systeme auf Geschäftsanforderungen
- Nutzung des vorhandenen Wissens
- Nutzung vorhandener Geschäftslogik und schnelle sowie kostengünstige Kombination mit neuen Funktionen
- Benutzerfreundliche Interaktion
- Vereinfachte Integration über Plattformen hinweg
- Orientierung am Kerngeschäft und Beschäftigung damit – anstatt mit Technologiediskussionen

Ivory und Web Services zur Host-Integration

- Web Services = Standardisierte Schnittstelle
  - Aufruf über HTTP/HTTPS
  - Datenaustausch über SOAP = XML
  - Jede Art von Service-Aufruf = Schnittstelle möglich (Lesen, Schreiben, Löschen, Services mit/ohne Rückgabewerte)
  - Sehr einfache Integration in Java-Anwendungen möglich
  - Sehr einfache Integration in andere Technologien (.NET, SAP, …) möglich
  - Einmal definierter Web Service kann von beliebigen Anwendungen genutzt werden, die diesen Service benötigen (ohne Zusatzaufwand)

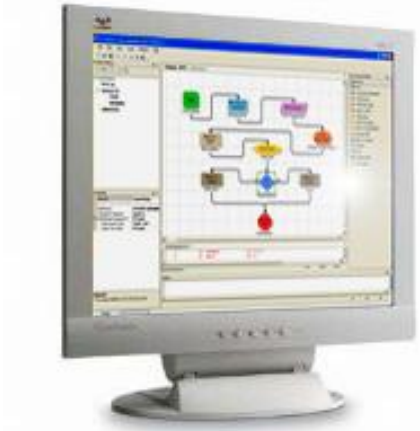
Web Service – WSDL Beispiel



Ivory Grundidee

Keine zusätzliche Middleware

Ivory™ Studio

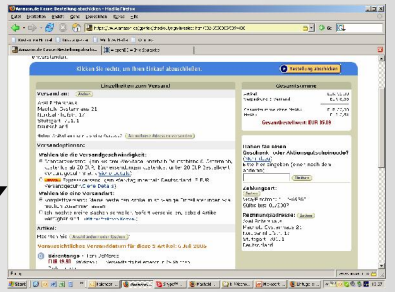


Entwicklungskomponente
auf XP, W2000, …

Ivory™ Server

Server-Komponente für IBM-Host
MVS, z/OS, VSE, OS/390

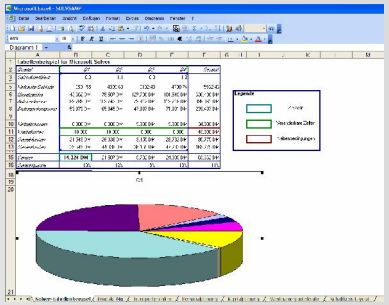
SOAP



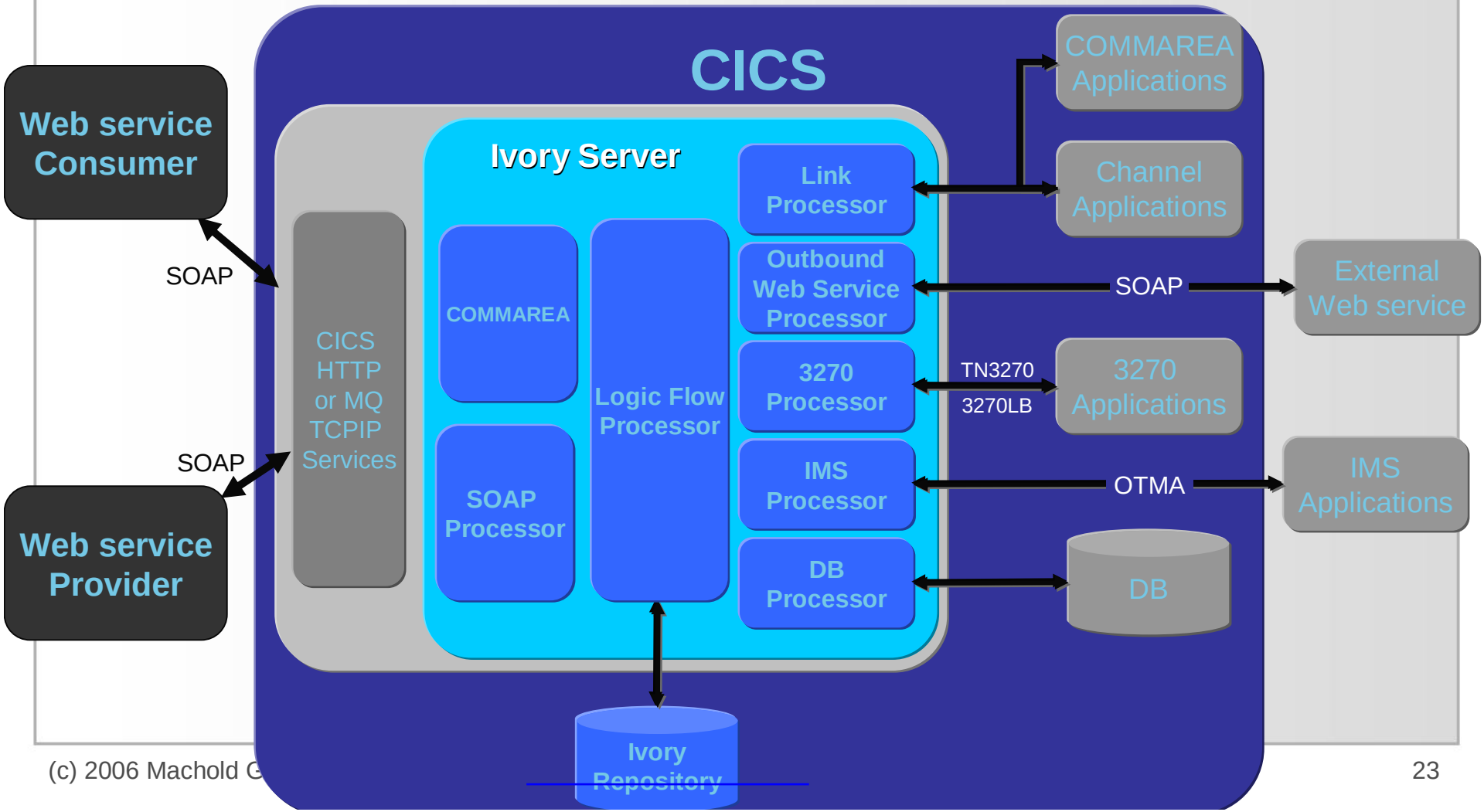
SOAP

SAP

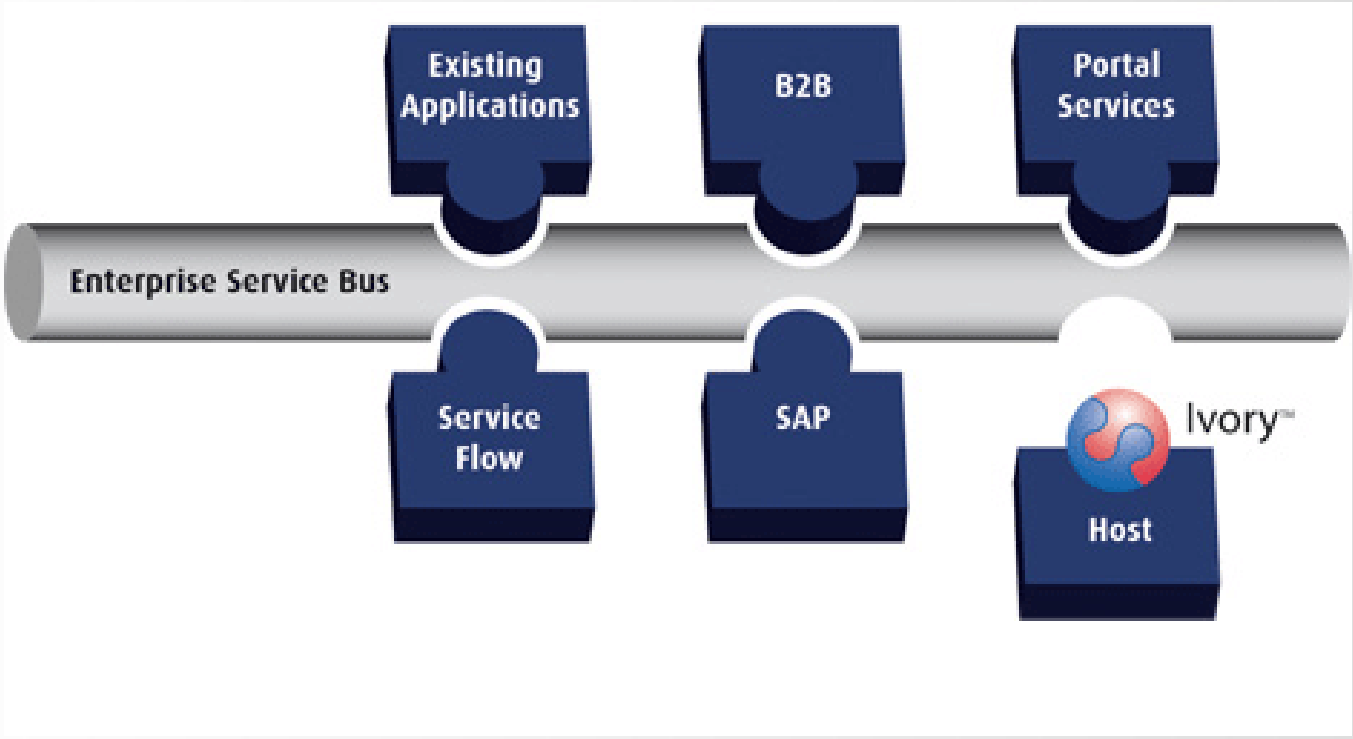
SOAP



CICS Architektur



SOA Architektur mit Ivory



Beispiel eines Kundenprojektes (1)

- Aufgabe:
  Sehr schnelle (< 6 Monaten) Realisierung eines neues Front-ends für über 10.000 Mitarbeiter als Ersatz einer bisherigen 3270-Oberfläche
- Situation:
  Sehr wenig Zeit für Implementierung, alte Funktion steht zur Verfügung, die aber angepasst werden muss.
  Weiteres Projekt in Pipeline mit vergleichbaren Anforderungen an Funktionalität

Beispiel eines Kundenprojektes (1)

- Vorgehensweise:
  Überzeugt davon, dass mit klassischer Entwicklung der Zeitplan nicht zu halten ist, wurde ein Proof of Concept mit Systemtechnik (Alter > 50 Jahre), Anwendungsentwicklung (Alter zw. 30 und 50 Jahren) und Java-Entwicklern (Alter ca. 30 Jahren) zur Evaluierung eines SOA-Ansatzes (Ivory) durchgeführt
- Ergebnis:
  Begeisterung sämtlicher Beteiligten, einen neuen Weg einzuschlagen. Kombination erfahrener und junger Entwickler, um der Kritikalität des Systems gerecht zu werden.
  Erfahrene Entwickler beginnen mit SOA, junge Entwickler nutzen Leistungsfähigkeit des Mainframes

Beispiel eines Kundenprojektes (2)

- Ausgangssituation:
  Von 3 noch vorhandenen OS/2-PCs (Hardware) fällt eine Maschine aus. Es gibt keinen Ersatz mehr! Anwendung ist existentiell für das Unternehmen (wenn auch nur von <5 Mitarbeitern genutzt).

- Aufgabe:
  Ersatz der OS/2-Anwendung und Weiterverwendung der als CICS-Programme vorhandenen Businesslogik.
  Berücksichtigung weiterer Ausbaupläne einzelner dieser Funktionen u.a. für Vertriebssysteme, SAP, etc.

Beispiel eines Kundenprojektes (2)

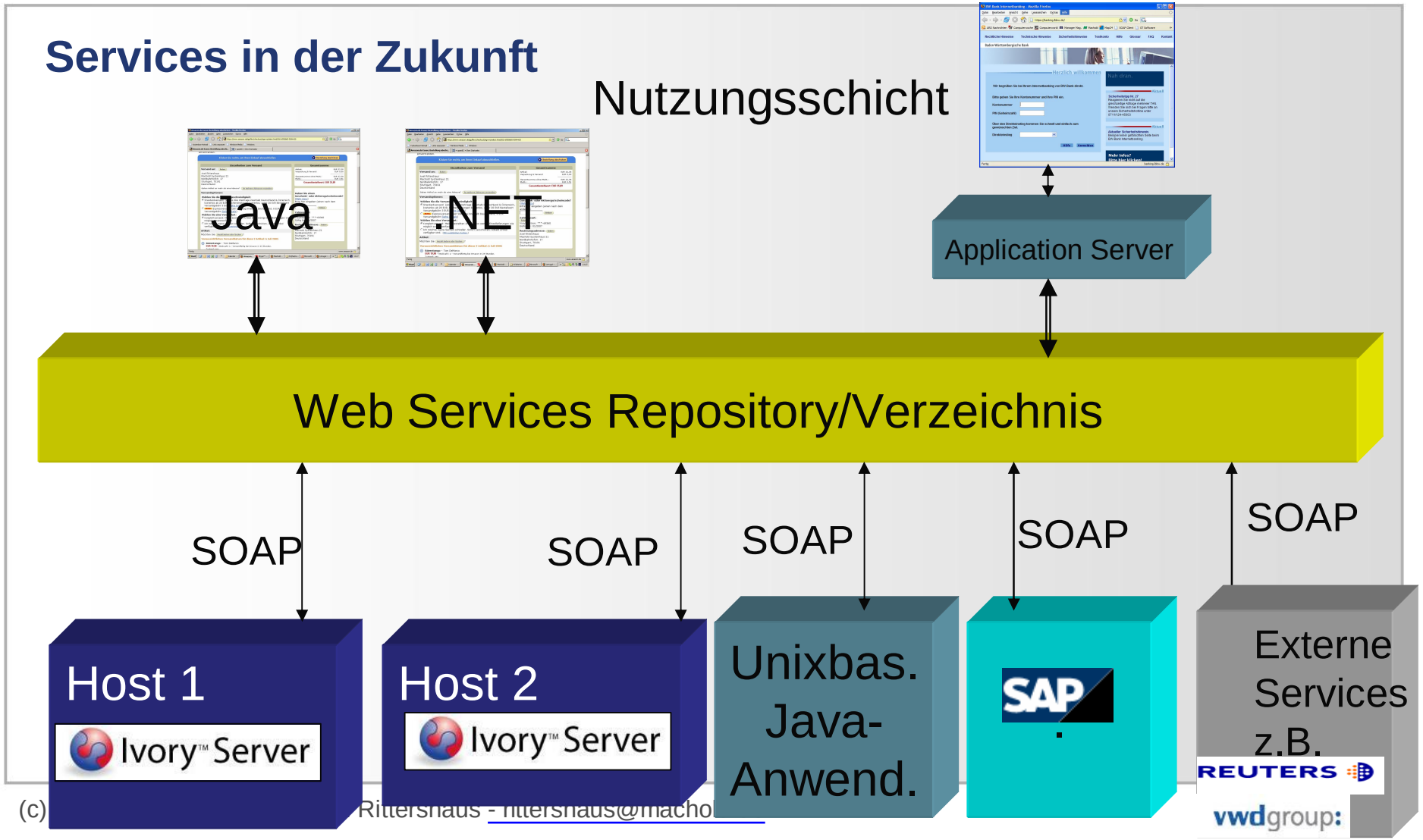
- Lösung:
  Implementierung von ca. 40 Web Services mit Ivory
  Implementierung einer neuen Java-basierten Oberfläche mit ca. 40 – 60 Bildschirmseiten
  Verbesserung des Benutzerkomforts u.a. durch direktes Verlinken von Informationen, Anzeigen von Zusatzdaten, Kombination zuvor getrennter Anzeigeseiten, etc.
- Realisierungszeit: ca. 6 Monate mit 1 Großrechner- und 2-3 Java-Entwicklern
  Enorme Begeisterung bei Fachbereichen in der Nutzung der Anwendung

Ivory

- Liefert bewährte Mainframe Funktionalitäten an beliebige .NET, J2EE und Web Service-nutzende Anwendungen
- Graphische IDE (Ivory Studio) und Mainframe basierte Serverkomponente
- Kombination von mehreren Host-Transaktionen zu einem Web Service möglich
- Import wizard für Copybooks, Includes, BMS maps, …
- Bringt die Mainframe-Vorteile Performance, Skalierbarkeit, Stabilität und Sicherheit und die vorhandenen Anwendungen in die SOA ein

- KEINE zusätzliche Middleware (HW/SW) notwendig
- KEINE Veränderung an bestehenden Mainframe Anwendungen
- KEINE Programmierung bei Ivory-Nutzung
- KEIN Zusatzwissen (XML, SOAP, WSDL, ..) notwendig

Services in der Zukunft



LIVE DEMO

Entwicklung Web Service für COBOL-Programm

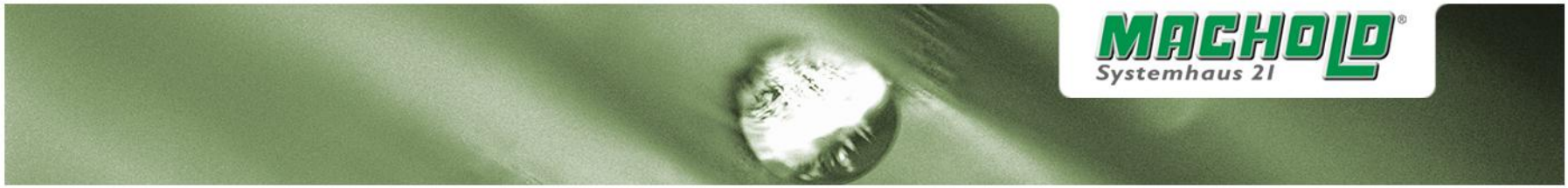
Nutzung des Test-Mainframes von GT Software in Atlanta, GA, USA

Vorteile auf einen Blick

- Hohe Produktivität bei Entwicklung von Web Services, keine Programmierung
- Nutzung des Host, keine Middleware notwendig
- Schutz und Absicherung bestehender Investitionen in Mainframe-Anwendungen
- Bestehende Mainframe-Anwendungen werden nicht/wenig verändert
- Zugriff auf Transaktionen, (Unter-)Programme und 3270 möglich
- Kombination mehrerer Host-Transaktionen/Programme zu einem WebService
- Volle CICS und IMS Unterstützung
- Kein Know how Aufbau notwendig, Mainframe-Entwickler werden gestärkt

- Schnell
- Effizient
- Kostengünstig
- Sicher

Ivory fördert und stabilisiert den weiteren Einsatz von Mainframes und vereinfacht SOA



Kontakt

EDV-Beratung Machold GmbH
Systemhaus 21
Nordbahnhofstr. 17
70191 Stuttgart
Germany
www.machold.de
www.vse-soa.de

Axel Rittershaus
Tel.: +49 (711) 2 57 72-11
Fax: +49 (711) 2 57 72-22
Mobil: +49 (160) 93 860 118

E-mail: axel.rittershaus@machold.de