e-Business-Engineering

Philipp Dopichaj
dopichaj@student.uni-kl.de

Universität Kaiserslautern

Inhalt

- Besonderheiten des E-Business
- Komponenten
  - Grundlagen
  - Vorstellung verschiedener Modelle
- Web Services
  - Allgemeines
  - Vorstellung einiger Basistechnologien

Besonderheiten des e-Business

- Zugriff auf interne Firmendaten
- Skalierbarkeit
- ähnliche Aufgabenstellungen
- Software-Erstellung durch Laien
- Kooperation mit anderen Firmen → SCM
- Sicherheit

Komponenten

Komponenten

- Software-Einheiten
- Kontextunabhängig
  - konzeptuell
  - technisch
- Black Box

Komponentenmodelle

- Richtlinien
- Properties
- Introspection
- Customization
- Persistence
- Packaging

Komponenten vs. Objektorientierung

Komponenten	Objektorientierung
Einheiten der Wiederverwendung	
Komponenten und Schnittstellen	Objekte/Klassen
Reuse by. . .	
. . . composition	. . . inheritance

Komponenten im e-Business

- hoher Abstraktionsgrad
- Plug and Play
- Betonung der Wiederverwendung
- Auslagerung von Geschäftsprozessen
- Dynamik
- Skalierbarkeit
- Herstellerunabhängigkeit

Komponententechnologien

Bewertungskriterien

Komponentenmodell

Programmiersprachenunabhängigkeit

Plattformunabhängigkeit

e-Business-Plattform

Laufzeitumgebung

Technologien – OMG

CORBA vor 3.0

– Komponentenmodell	+ Plattformunabhängigkeit
+ Sprachunabhängigkeit	– e-Business-Plattform
– Laufzeitumgebung	

CORBA ab 3.0

+ Komponentenmodell	+ Plattformunabhängigkeit
+ Sprachunabhängigkeit	– e-Business-Plattform
– Laufzeitumgebung	

Technologien – Microsoft

DCOM

+ Komponentenmodell	– Plattformunabhängigkeit
+ Sprachunabhängigkeit	– e-Business-Plattform
– Laufzeitumgebung	

.NET

+ Komponentenmodell	? Plattformunabhängigkeit
+ Sprachunabhängigkeit	+ e-Business-Plattform
+ Laufzeitumgebung	

Technologien – Sun

Java/RMI

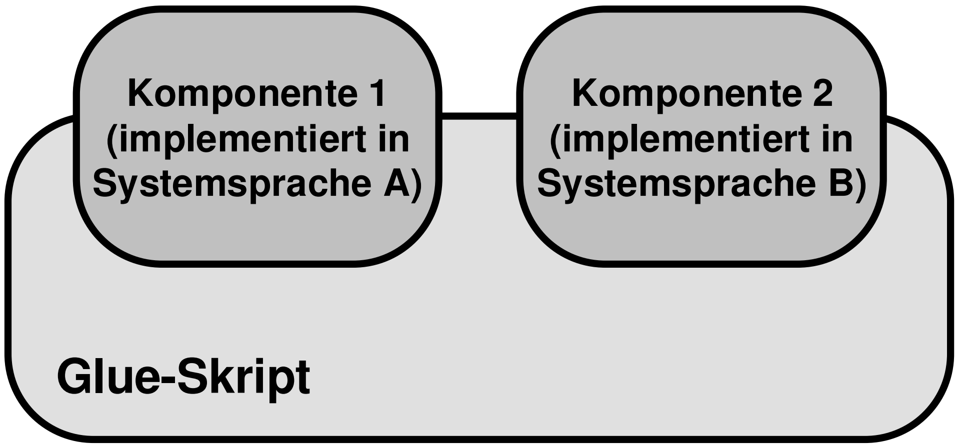
– Komponentenmodell	+ Plattformunabhängigkeit
– Sprachunabhängigkeit	– e-Business-Plattform
+ Laufzeitumgebung	

J2EE

+ Komponentenmodell	+ Plattformunabhängigkeit
– Sprachunabhängigkeit	+ e-Business-Plattform
+ Laufzeitumgebung	

Skriptsprachen

- interpretiert
- keine Typprüfung zur Übersetzungszeit
- hoher Abstraktionsgrad
  ⇒ leicht erlernbar



Probleme von Komponenten

Problem	führt zu
keine zentralen Marktplätze	zu wenig Wiederverwendung
Interoperabilität zwischen Technologien schlecht	Vorabentscheidung für Technologie nötig
fehlende Standards für Schnittstellen	keine Austausch- barkeit

Web Services

Laut IBM:

- new breed of Web application
- self-contained, self-describing, modular
- can be published, located and invoked across the Web

Im engeren Sinn:

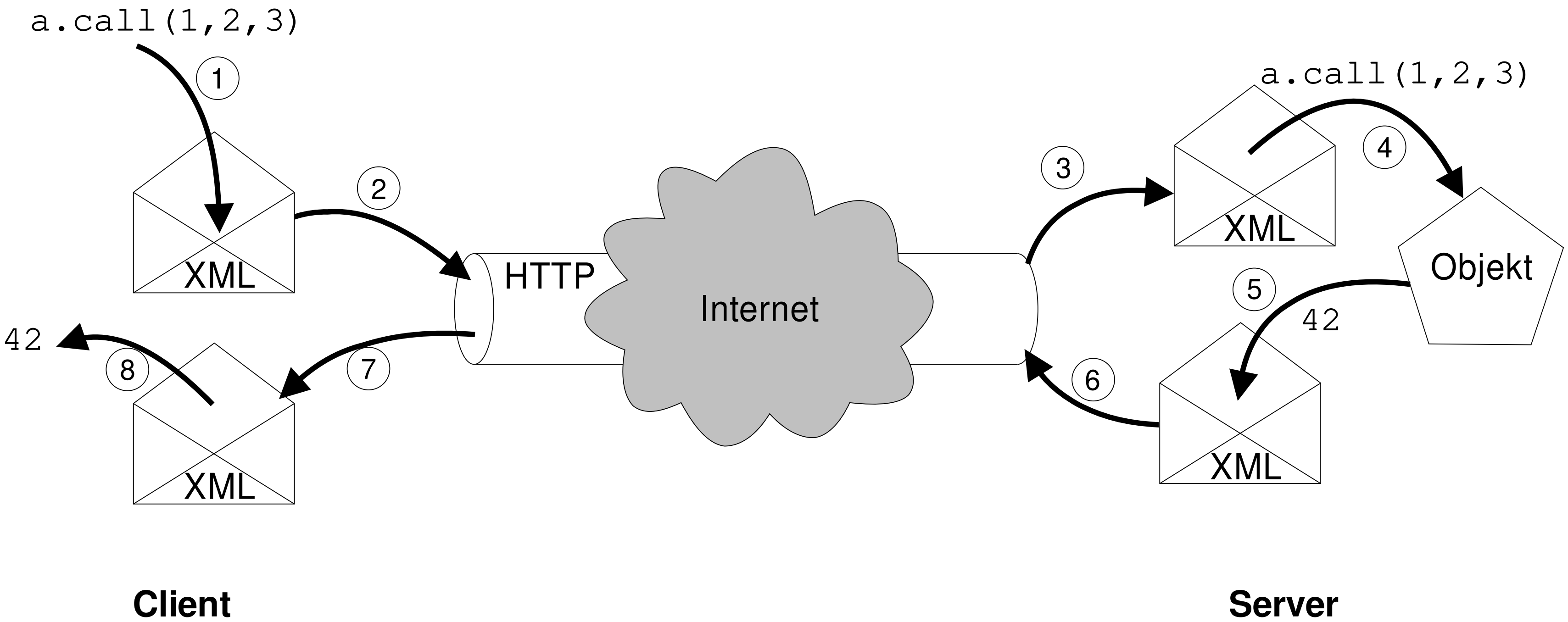
- Satz von Spezifikationen
- Zentral: objektorientiertes RPC (SOAP)

Web Services

- auch als Komponententechnologie zu sehen
- Ergänzung zu den anderen Technologien
- Verwendung verbreiteter Protokolle und Technologien
  - HTTP
  - XML
- Bibliotheken dafür sind praktisch überall verfügbar

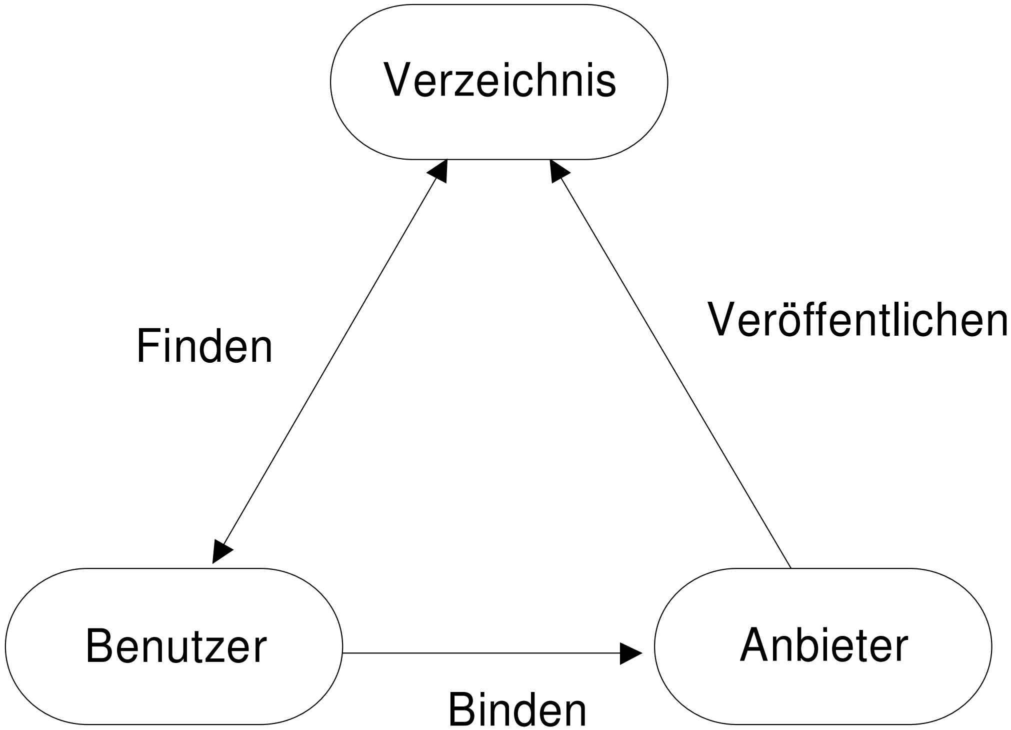
SOAP

- RPC objektorientiert
- basiert auf XML und HTTP



UDDI

Verzeichnisdienst für Web Services



Weitere WS-Technologien

Transaktionen	
XLANG	optimistisch → Kompensation
TAML	unter anderem verteiltes 2PC

Sicherheit	
SSL	verschlüsselte Übertragung
Passport.NET	Authentifizierung
XKMS	Zertifizierung, Signaturen

SOAP-Server

```
#!/usr/local/ActivePerl-5.6/bin/perl -w
use strict;
use SOAP::Transport::HTTP;

SOAP::Transport::HTTP::Daemon->new(LocalPort => 9001)
  ->dispatch_to('Test')->handle;

package Test;
sub dividiere {
  my ($class, $dividend, $divisor) = @_;

  die 'Man kann nicht durch 0 teilen!' if $divisor == 0;
  return $dividend / $divisor;
}
```

SOAP-Client

```
#!/usr/local/ActivePerl-5.6/bin/perl -w
use strict;
use SOAP::Lite;

my $server = SOAP::Lite->uri('http://localhost:9001/Test')
  ->proxy('http://localhost:9001/');

my $result = $server->dividiere(10, 5);
if (!$result->fault) {
  print 'Ergebnis: ', $result->result;
} else {
  print 'Fehler: ', $result->faultstring;
}
```

Zusammenfassung

Im E-Business bietet sich die Verwendung von Komponenten an

- leider ist der Markt noch zersplittert (viele Technologien)
- es ist unklar, wer gewinnt
  (beste Chancen für J2EE und .NET)
- Web Services als Lösungsansatz: Aufsetzen auf „niedrigeren“ Technologien
- Schnittstellenstandards fehlen