Cortex Brainware
Consulting & Training GmbH

Service Oriented Architectures

Dr.-Ing. Hartmut Kocher

Cortex Brainware
Consulting & Training GmbH
Kirchplatz 5
D - 82049 Pullach im Isartal

Tel. 089 / 744 850 0
http://www.cortex-brainware.de
mailto:hwk@cortex-brainware.de

Cortex Brainware
Consulting & Training GmbH

Übersicht

Geschäftliche Anforderungen

Von Prozeduren zu Diensten

SOA

Windows Communication Foundation (WCF)

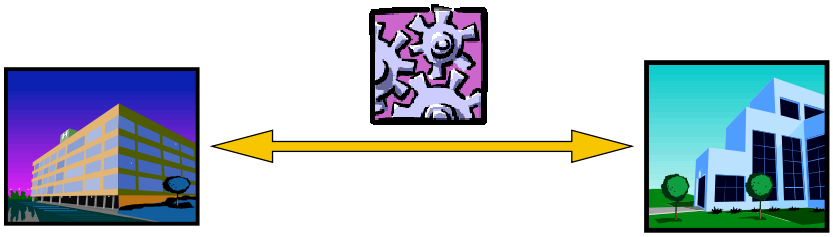
Zusammenfassung

Geschäftliche Anforderungen

Cortex Brainware
Consulting & Training GmbH

Um maximale Wertschöpfung zu erzielen, müssen Anwendungen integriert werden.

- Zunehmende Komplexität
- Geschäftsprozesse müssen eng verzahnt sein, um manuelle Eingriffe zu minimieren.

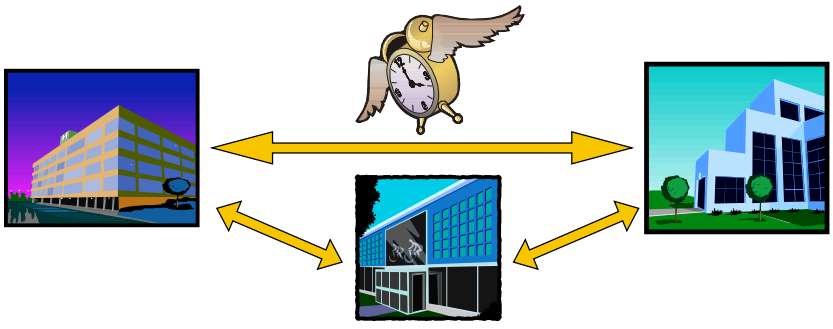


Flexible Geschäftbeziehungen

Cortex Brainware
Consulting & Training GmbH

Flexible Geschäftsbeziehungen erfordern flexible Geschäftsprozesse.

- Integration verschiedener Technologien notwendig.
- Geschäftsprozesse müssen leicht an neue Anforderungen anpassbar sein.
- Geschwindigkeit entscheidet oft über Geschäftserfolg.
  - Begrenztes Zeitfenster für neue Geschäftsideen



Anforderungen

Cortex Brainware
Consulting & Training GmbH

Moderne Anwendungen sind verteilt.

- Es gibt praktisch keine isolierten Anwendungen
- Integration mit anderen Anwendungen ist notwendig.

Probleme:

- Unterschiedliche Plattformen und Technologien erschweren die Zusammenarbeit.
- Einheitliche Ansätze in der Vergangenheit sind gescheitert.

Technologien für verteilte Systeme

Cortex Brainware
Consulting & Training GmbH

RPC (Remote Procedure Calls)

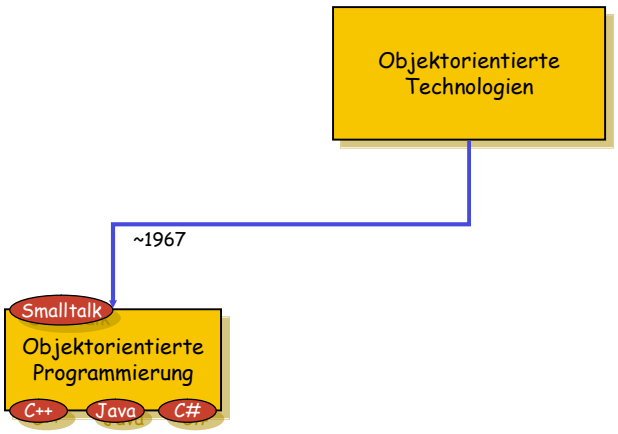
- Entfernte Aufrufe über Prozess- und Systemgrenzen hinweg.

Objektorientierung

- Verkapselung und Integration von Funktionen und Daten
- Verteilte Objekte

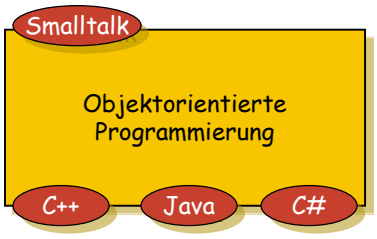
Objektorientierte Technologien

Cortex Brainware
Consulting & Training GmbH



Objektorientierte Programmierung

Cortex Brainware
Consulting & Training GmbH



Objekte arbeiten innerhalb eines Programmes zusammen:

- Eine Programmiersprache
- Ein Adressraum

Objektorientierung hat sich seit vielen Jahren bewährt.

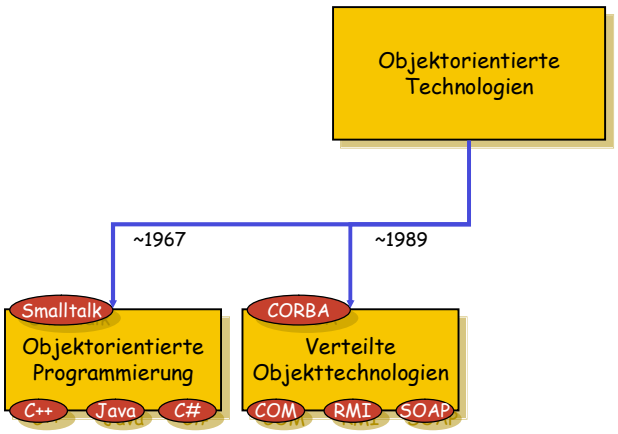
- Grundlagen sind definiert und verstanden.
- Alle neuen Technologien basieren auf objektorientierten Prinzipien.

Grundlagen von Objekten:

- Ganzheitliche Betrachtung von Verhalten und Daten
- Jedes Objekt besitzt eine Identität.
- Eine Klasse beschreibt eine Menge von Objekten mit ähnlichen Eigenschaften.

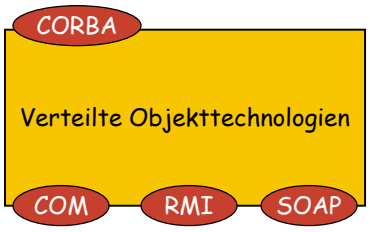
Objektorientierte Technologien

Cortex Brainware
Consulting & Training GmbH



Verteilte Objekttechnologien

Cortex Brainware
Consulting & Training GmbH



Objekte arbeiten innerhalb eines Systems zusammen:

- Eine oder mehrere Programmiersprachen
- Mehrere Adressräume

Erweiterung des objektorientierten Paradigmas um verteilte Objekte:

- Objekte können in verschiedenen Prozessen zusammen arbeiten.
- Grundlegende Dienste unterstützen die Zusammenarbeit.
  - Namensdienste
  - Persistenz
  - Transaktionen

Höherwertige Dienste stehen nicht zur Verfügung.

- Diese müssen selbst definiert werden.

Verteilte Systeme

Cortex Brainware
Consulting & Training GmbH

Es gibt verschiedene Gründe, ein System auf mehrere Teilsysteme zu verteilen:

- Geographische Verteilung
- Lastverteilung
- Redundanz, Fehlertoleranz, Verfügbarkeit
- usw.

Die Gründe, warum ein System verteilt wird, sind nicht OO-spezifisch.

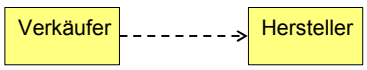
- Die Implementierung kann jedoch Vorteile der OO-Technik nutzen.

Interprozesskommunikation

Cortex Brainware
Consulting & Training GmbH

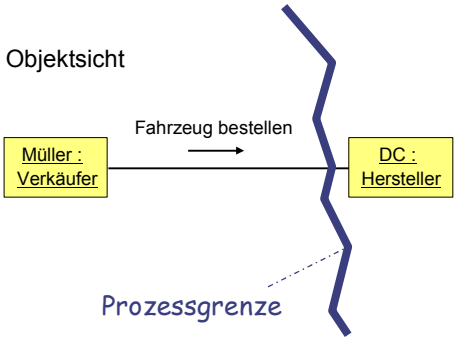
Was passiert, wenn Objekte zweier Klassen in verschiedenen Prozessen miteinander kommunizieren wollen?

Klassensicht



Verkäufer bestellt ein Auto beim Hersteller.

Objektsicht



Objekte befinden sich in verschiedenen Prozessen.

Objektorientierte Kommunikationsmechanismen

Cortex Brainware
Consulting & Training GmbH

Prinzip: Broker-Pattern

- Aufruf von Methoden entfernter Objekte
- Datenübertragung in Form von Parametern oder Rückgabewerten
- Client instantiiert lokalen Stellvertreter (Proxy) für entferntes Objekt und alle Methodenaufrufe laufen über diesen Stellvertreter.
- Proxy ist dann für den Aufruf der eigentlichen Funktion auf dem Server zuständig.
- Aufrufmechanismen werden von einer bestimmten Kommunikations--Middleware geregelt.

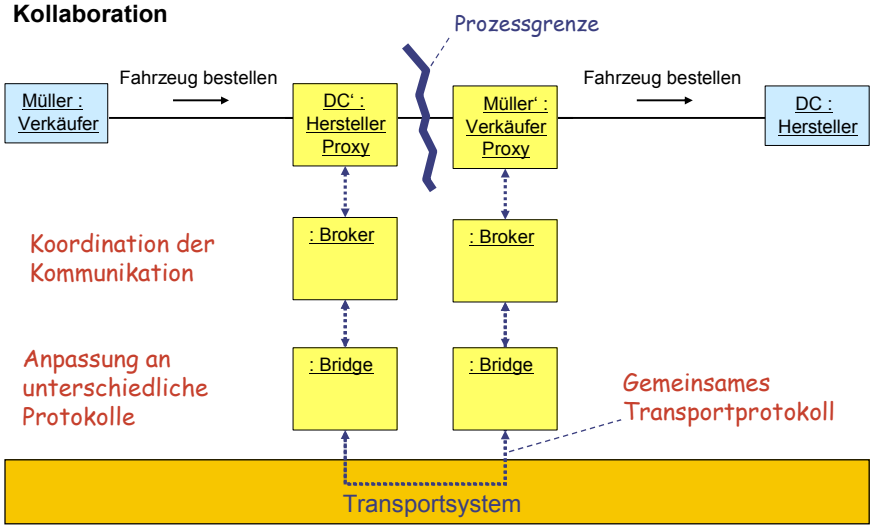
Beispiele:

- CORBA, RMI, DCOM

Broker-Pattern

Cortex Brainware
Consulting & Training GmbH

Kollaboration



Broker-Pattern

Cortex Brainware
Consulting & Training GmbH

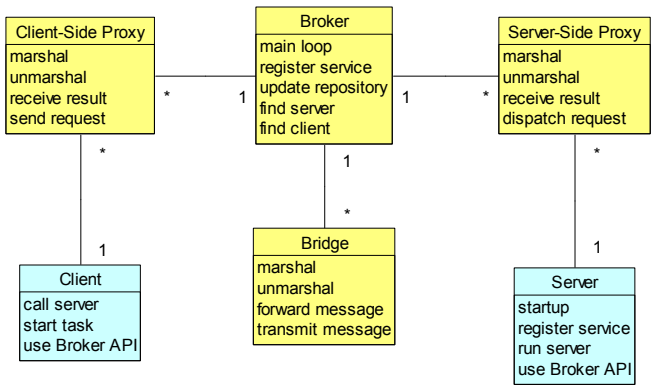
Zweck

- Strukturierung verteilter Systeme mit entkoppelten und interagierenden Komponenten
- Broker koordiniert Kommunikation zwischen Komponenten.
  - Erlaubt ortstransparente Kommunikation zwischen Komponenten.
- Austausch, Hinzufügen und Entfernen von Komponenten zur Laufzeit

Broker-Pattern

Cortex Brainware
Consulting & Training GmbH

Struktur



Broker-Pattern

Cortex Brainware
Consulting & Training GmbH

Konsequenzen

- Broker und Proxies sorgen für Ortstransparenz und verbergen Kommunikationsmechanismen.
- Betriebssytemdetails werden verborgen.
  - Prozess- und Threadingmodelle sind verkapselt.
- Interoperabilität zwischen Brokern wird mittels Bridges sichergestellt.
- Weniger effizient als direkte lokale Kommunikation
  - Performanz kann optimiert werden, wenn Broker nur die Verbindung aufbauen und der Nachrichtenaustausch direkt zwischen Proxies erfolgt.
- Höhere Fehleranfälligkeit, da zusätzliche Hardware- und Softwarekomponenten beteiligt sind.
- Test und Debugging sind komplexer.

CORBA

Cortex Brainware
Consulting & Training GmbH

Allgemein:

- Standard der OMG (Object Management Group)
- Weit verbreitet und offen
- System- und sprachübergreifend

Funktionsweise:

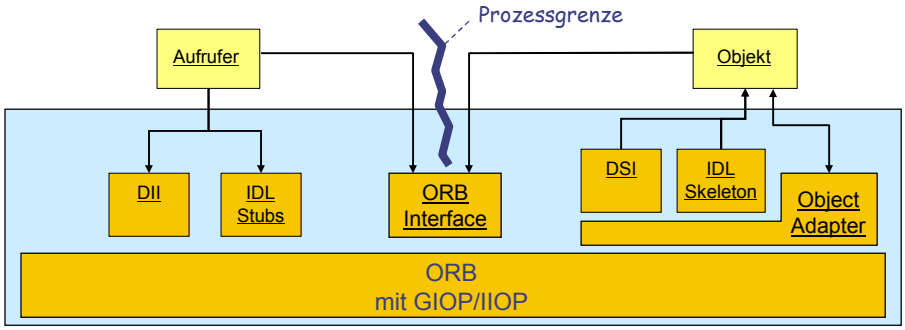
- Kommunikation wird über so genannten „Object Request Broker“ (ORB) realisiert.
- Interface wird in einer neutralen Meta-Sprache definiert, der Interface Definition Language (IDL).
- Interface-Definition in IDL wird für gewünschte Zielsprache kompiliert.
  - Es gibt „Language Bindings“ für alle gängigen Programmiersprachen

Object Request Broker - Architektur

Cortex Brainware
Consulting & Training GmbH

Schnittstellen werden mit der „Interface Definition Language“ (IDL) beschrieben.

- Daraus werden Stubs und Skeletons (Proxies) erzeugt.
- DII und DSI erlauben den dynamischen Aufruf von Schnittstellen.
- Object Adapter ist die Anpassung von Objekten an den ORB.



Probleme verteilter Objektsysteme

Cortex Brainware
Consulting & Training GmbH

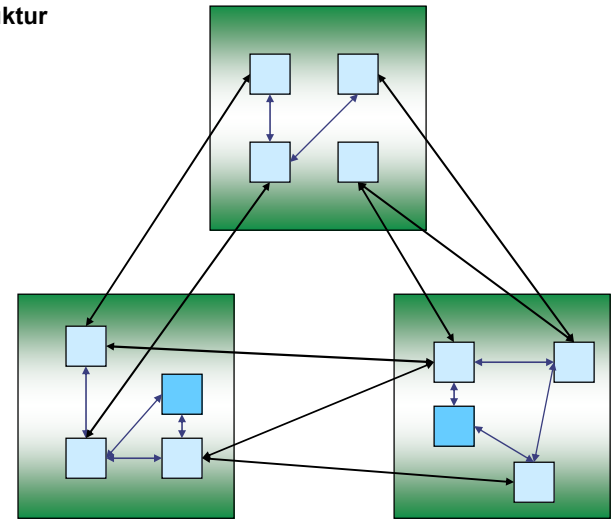
In der Praxis sind verteilte Objektsysteme schwierig in der Handhabung:

- Objekte können aus Gründen der Performanz nicht beliebig verteilt werden.
- Systeme sind über gemeinsame Objekte eng gekoppelt.
- Verteilte Business-Objekte sind schwer wartbar.
  - Interne Objekte werden nach außen sichtbar.
- Zusammenarbeit verschiedener Systeme ist nicht gegeben.
- Viele Dienste (Authentifizierung, Transaktionen, Lastverteilung usw.) müssen immer wieder neu implementiert werden.

Einzelne Objekte

Cortex Brainware
Consulting & Training GmbH

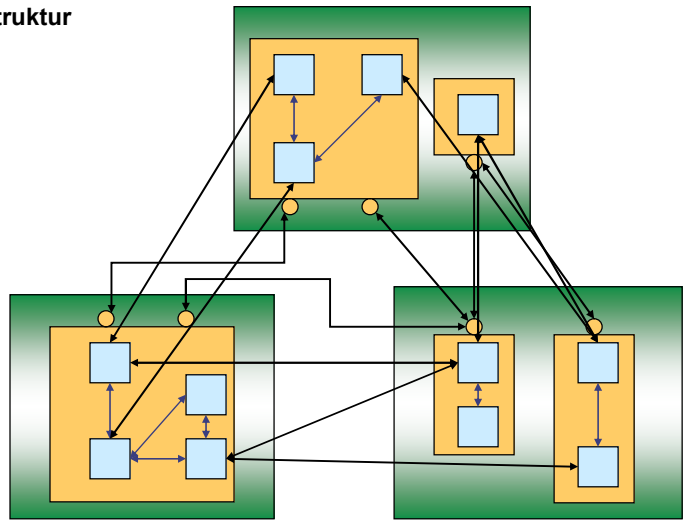
Struktur



Komponenten

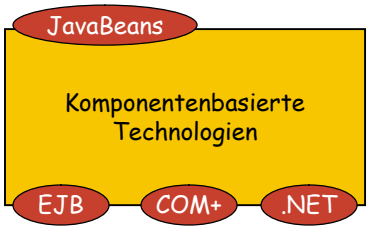
Cortex Brainware
Consulting & Training GmbH

Struktur



Komponentenbasierte Technologien

Cortex Brainware
Consulting & Training GmbH



Komponenten arbeiten zusammen:

- Eine oder mehrere Programmiersprachen
- Einheitlicher Komponentenstandard
- Mehrere Adressräume

Anwendungen werden aus vorgefertigten Komponenten zusammengebaut.

- Konfigurieren statt Programmieren
- Einheitliche Komponentenschnittstellen
- Komponentenstandards

Komponenten sind austauschbare Teile des Systems, die Dienste zur Verfügung stellen und andere benutzen.

- Komponentenbegriff ist stark überladen!

Komponenten

Cortex Brainware
Consulting & Training GmbH

Komponenten sind keine Klassen.

- Komponenten werden als „Bausteine“ konzipiert.
- Komponenten haben einen höheren Abstraktionsgrad als Klassen.
- Komponenten erfüllen einen Komponentenstandard.

Komponentenstandards definieren, wie Komponenten mit ihrer Umwelt zusammenarbeiten:

- Welche Dienste die Umgebung erwartet bzw. zur Verfügung stellt.
  - Z.B. Transaktionen, Authentifizierung usw.
- Welche Schnittstellen eine Komponente unterstützen muss, damit sie mit anderen Komponenten zusammenarbeiten kann.
- Dies erhöht die Produktivität, da alle Komponenten ähnlich verwaltet werden.

Komponenten und Container

Cortex Brainware
Consulting & Training GmbH

Container enthalten Komponenten.

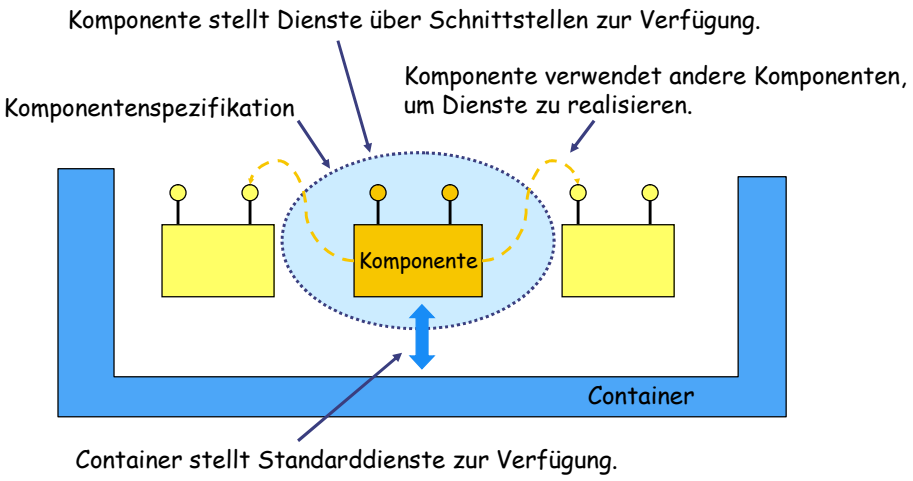
- Der Container stellt Standarddienste für Komponenten zur Verfügung.
- Komponente kann sich auf Anwenderlogik konzentrieren.
  - Höhere Produktivität und geringere Fehlerquote
  - Optimierungen durch Container möglich
- Klare Aufgabentrennung zwischen Komponentenentwickler und Containerhersteller
  - Jeder konzentriert sich auf sein Spezialgebiet.

Verschiedene Container für verschiedene Anwendungsgebiete:

- Container für graphische Elemente
- Web-Container
- Application-Server

Komponenten im Überblick

Cortex Brainware
Consulting & Training GmbH



Stärken und Schwächen moderner verteilter Systeme

Cortex Brainware
Consulting & Training GmbH

Schwächen

- Kommunikation wird versteckt (Transparenter Ansatz / Broker Pattern)
  - Kommunikation kann aber nicht vernachlässigt werden, da sie sich auf das Verhalten des Systems auswirkt.
- Einheitliche Implementierung erwartet
  - Bei CORBA/COM usw. werden einheitliche Architekturen erwartet
  - Komponenten gehen von einem gemeinsamen Komponentenstandard aus.

Stärken

- Zunehmende Trennung von Schnittstellen und Implementierung
- Grob granulare Kommunikationsbeziehungen durch Komponentenschnittstellen
  - Lose Kopplung
  - Minimierung von Latenzzeiten

Herausforderungen moderner Softwareentwicklung

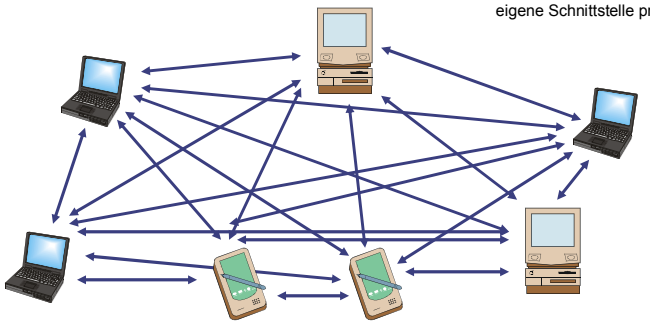
Cortex Brainware
Consulting & Training GmbH

Verflechtungen zwischen Systemen werden enger:

- Integration zwischen Anwendungen ist wichtig.
- Insbesondere bei E-Business-Anwendungen arbeiten viele verschiedene Systeme zusammen.

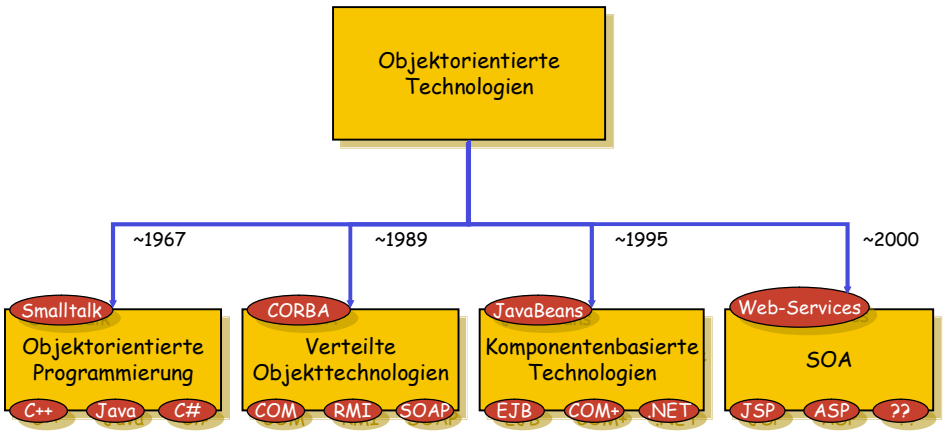
Abhängigkeiten zwischen Systemen nehmen zu.

- Standards reduzieren die Anzahl der Schnittstellen.
  - Gemeinsame Schnittstellen statt jeweils eigene Schnittstelle pro Partner



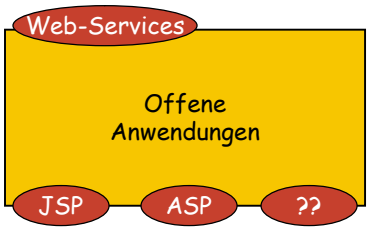
Objektorientierte Technologien

Cortex Brainware
Consulting & Training GmbH



Offene Anwendungen

Cortex Brainware
Consulting & Training GmbH



Anwendungen arbeiten zusammen:

- Beliebige Programmiersprachen
- Verschiedene Komponentenstandards
- Verschiedene Zugriffsmöglichkeiten
- Standardisierung auf Business-Ebene notwendig

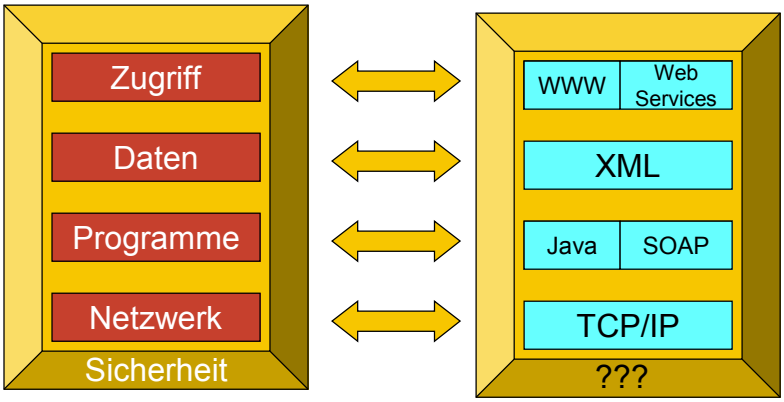
Offene Anwendungen erlauben die Integration über Systemgrenzen hinweg.

- Gemeinsame Standards ermöglichen die Zusammenarbeit unterschiedlichster Systeme.
- Standards verteilen die Verantwortung auf mehrere Partner:
  - Jeder ist selbst verantwortlich, den jeweiligen Standard einzuhalten.
- Durch Standardprotokolle können auch Anwendungen über das Internet verbunden werden (Firewalls).
- Gemeinsame Anwendungsschnittstellen erforderlich, z.B. ebXML.

Offene Standards

Cortex Brainware
Consulting & Training GmbH

Standards für offene verteilte Systeme



Web-Services

Cortex Brainware
Consulting & Training GmbH

Web-Services sind Komponenten, die ihre Dienste über das Internet/Intranet zur Verfügung stellen.

- Als Transportprotokoll wird SOAP verwendet.
- Offen für beliebige Anwendungsarchitekturen
- Damit mehrere Anwendungen sinnvoll zusammenarbeiten können, sind weitere Dienste erforderlich:
  - WSDL (Web Service Description Language) zur Beschreibung des Dienstes,
  - UDDI (Universal Description, Discorvery, and Integration) zur Verwaltung von Web-Services,
  - Dienste auf Anwendungsebene wie ebXML.

Service-orientierte Architekturen (SOA)

Cortex Brainware
Consulting & Training GmbH

Anwendungen stellen Funktionalität als modulare Services bereit.

- Systeme integrieren verschiedene Services.
- Dienste werden neutral beschrieben.
  - Unabhängig von Implementierung.
  - Integration über System- und Architekturgrenzen hinweg
  - z.B. mit XML Schema
- Services sind *grob granulare* Dienste, die von anderen konsumiert werden können.
  - Austausch von Dokumenten statt Remote Procedure Calls

Implementierung kann mit Standards erfolgen.

- Prinzipiell sind verschiedene Lösungen möglich.
- Standards erleichtern Zusammenarbeit
  - Web-Services

Grundlagen von Services

Cortex Brainware
Consulting & Training GmbH

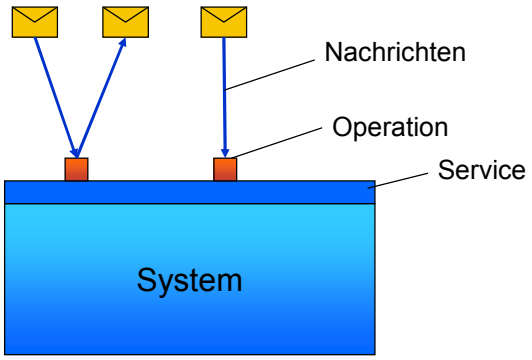
Es gibt 4 Grundregeln:

- Grenzen sind explizit
  - Alles was ausgetauscht wird, muss spezifiziert werden.
  - Saubere Trennung von Schnittstellen und Implementierung
  - Expliziter Austausch von Nachrichten
- Beschreibung von Diensten mit Metadaten
  - Interoperable Schemata
- Policies beschreiben Laufzeitverhalten
  - Kommunikationsverhalten und Parameter festlegen
- Services sind autonom
  - Dienste sollten möglichst unabhängig voneinander sein.
  - Versionierung
  - In der Praxis nicht immer durchsetzbar

Service Übersicht

Cortex Brainware
Consulting & Training GmbH

Ein System bietet Dienste nach außen an.

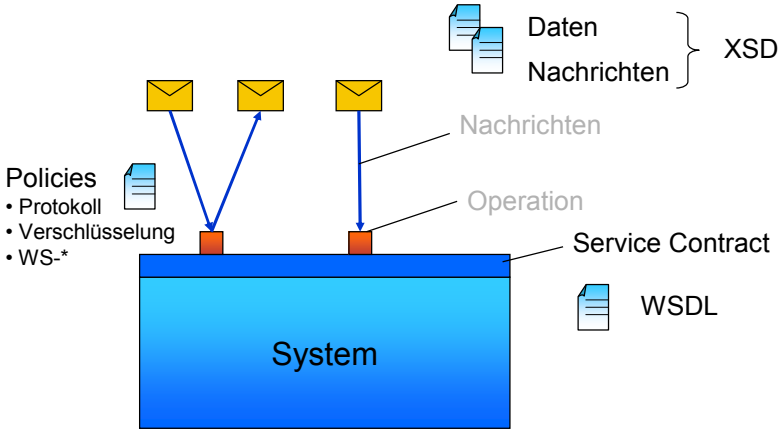


Service Übersicht

Cortex Brainware
Consulting & Training GmbH

Ein System bietet Dienste nach außen an.

- Alles wird explizit festgelegt.



Standards

Cortex Brainware
Consulting & Training GmbH

XML Infoset

- Basis für Standards
- Hat nichts mit <…> zu tun!
- XML 1.0 ist nur eine Darstellung von XML Infoset

Gremien

- W3C
- Oasis
- OMG

Beschreibung von Diensten

Cortex Brainware
Consulting & Training GmbH

Service Contract

- Daten und Nachrichten werden mit XML-Schema beschrieben
- Operationen und das Binding an Endpunkte wird mit der WSDL beschrieben.
- Ein Dienst kann über mehrere Endpunkte mit unterschiedlichen Protokollen angeboten werden.

Policies

- Diese beschreiben die Randbedingungen der Kommunikation
  - Protokoll
  - Verschlüsselung
  - Dienstgüte
  - WS-* Standards (WS-Security, WS-Reliable Messageing, …)

Orchestrierung von Services

Cortex Brainware
Consulting & Training GmbH

Komplexe Anwendung entstehen durch das Zusammenspiel von Diensten.

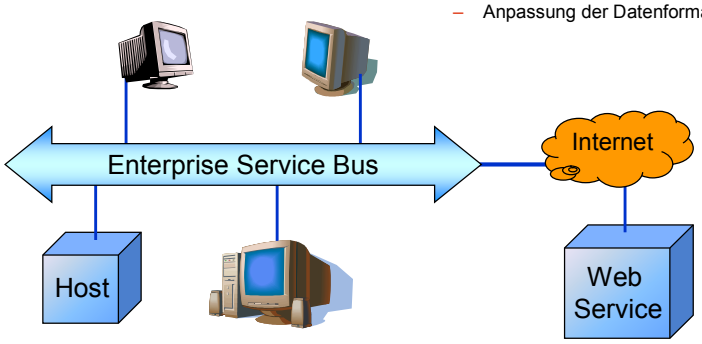
- Durch die Unabhängigkeit der Dienste von ihrer Implementierung können Dienste, die auf verschiedenen Art realisiert sind, zusammen arbeiten.
- Durch Verkettung von Diensten können Workflows abgewickelt werden.
- Workflow Engines bekommen größere Bedeutung.
- BPEL erlaubt standardisierte Workflow-Beschreibung
  - Business Process Execution Language

Enterprise Service Bus (ESB)

Cortex Brainware
Consulting & Training GmbH

Ein Enterprise Service Bus erlaubt die Integration von Anwendungen.

- Koordinierung von Abläufen
  - Steuerung von Geschäftsprozessen
- Lose Kopplung zwischen Anwendungen
  - Keine direkte Kommunikation notwendig
  - Asynchrone Kommunikation erlaubt Entkopplung
- Anpassung an unterschiedliche Dienste
  - Anpassung der Datenformate



Beispiel Windows

Cortex Brainware
Consulting & Training GmbH

Konkurrierende Technologien mit jeweils anderem Programmiermodell

- COM/COM+/DCOM
- .NET Remoting
- MSMQ Messaging
- Web Services

Windows Communication Foundation (Indigo)

- Einheitliches Programmiermodell für Services
- Gleicher Service kann über mehrere Endpunkte angeboten werden
- Policies können konfiguriert werden

Windows Workflow Foundation

- Basis für zukünftige Workflow-Produkte
- Orchestrierung von Diensten

Zusammenfassung

Cortex Brainware
Consulting & Training GmbH