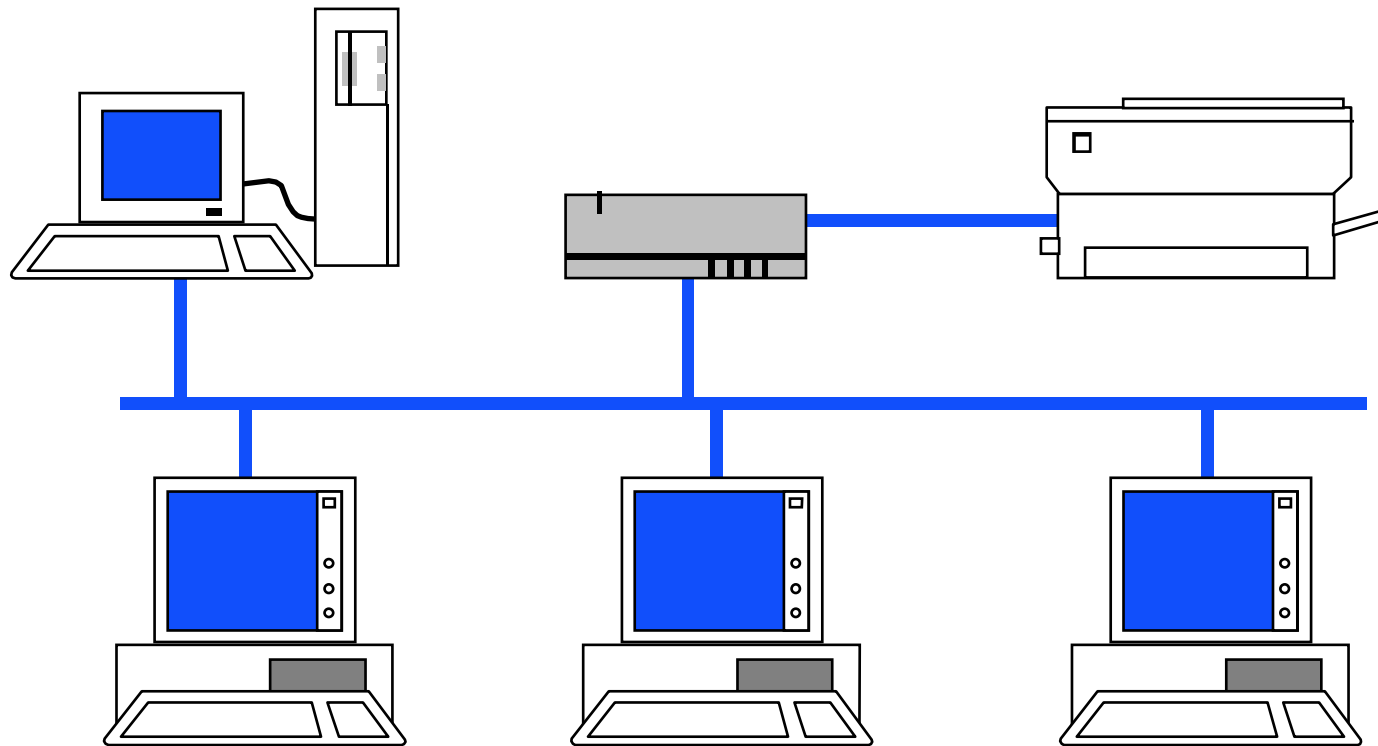


NVS1 5. Jahrgang

Mag. Dr. Klaus Coufal



Übersicht

- Kompetenzbereiche
 - Architektur und Entwicklung verteilter Systeme

Architektur und Entwicklung verteilter Systeme

- Algorithmen in verteilten Systemen
samt Synchronisation und
Interprozesskommunikation
- Serviceorientierte Architektur (SOA)
- Enterprise Application Architecture
- Cloud Computing
- Middleware
- Multi Tier Systeme.

I Webapplikationen

- Architektur von Webapplikationen
- Verteilung der Aufgaben bei Webapplikationen
- Speicherung von Daten bei Webapplikationen
- Entwicklungswerkzeuge für Webapplikationen

Architektur von Webapplikationen

- Webserver
- Applikationsserver
- Authentifikationsserver
- Client

Webserver - Aufgaben

- Dokumente (HTML-Dateien, CSS-Dateien, Bilder, ...) an Clients ausliefern
- Das verwendete Protokoll ist dabei HTTP (Port 80) bzw. HTTPS (Port 443)
- Heute zunehmend dynamisch erstellte Dokumente (SSI, PHP, JSP, ASP, ...)

Webserver - Zusatzaufgaben

- Cookieverwaltung
- Zugriffsbeschränkungen
- Weiterleitung – Rewrite
- Fehlerbehandlung (Fehlerseiten)
- Protokollierung
- (Caching)

Webserver – Produkte

- Apache HTTP Server
 - Apache Software Foundation
- IIS (Internet Information Server)
 - Microsoft
- ...
 - CERN httpd
 - lighttpd

Applikationsserver – Aufgaben

- Server auf dem Anwendungsprogramme ausgeführt werden
- Laufzeitumgebung für den Serverteil einer Client-Server-Anwendung
- Daten vom Client verarbeiten und die Ergebnisse wieder dem Client zur Verfügung stellen

Applikationsserver – Produkte

- Adobe ColdFusion
- IBM Websphere
- Oracle (BEA) Weblogic
- Apache Tomcat
- Zope
- ...

Authentifikationsserver 1

- Webinterface für Benutzer (i.A. https)
- Sicherheit nach außen (Problematik mehrerer Server im Inneren und damit unterschiedliche Zertifikate)
- Funktionsweise eines „Reverse Proxy“
- Oft Anbindung an Verzeichnisdienst, selten eigene Benutzerdatenbank

Authentifikationsserver 2

- Sun ONE (Open Network Environment)
-> Sun Java Enterprise System
- Novell iChain -> Novell Access
Manager -> NetIQ Access Manager
- Microsoft Forefront

Client

- Browser
 - Mozilla Firefox
 - Google Chrome
 - Microsoft Internet Explorer
 - Apple Safari
 - ...
- Apps

II Client-Server-Architektur

- Definition
- Zutaten
- Serverbeispiele
- Schichten
- 2-Schicht-Architektur
- 3-Schicht-Architektur

Definition

- Client-Server-Architektur beschreibt ein Modell Aufgaben in einem Netzwerk zu verteilen.
- Dabei übernehmen Server die zentralen Aufgaben und stellen Sie Clients zur Verfügung

Zutaten

- Server (Dienstleister)
- Client (Nutzer)
- Protokoll (erlaubt dem Client einen Request an den Serverdienst zu stellen und eine entsprechende Response auszuwerten)

Serverbeispiele

- Fileserver (NCP, SMB, NFS, ...)
- Printserver (LP, NCP, ...)
- CD-ROM-Server (Medienserver)
- Datenbankserver
- Mailserver (smtp, pop, imap)
- Webserver (http)
- Applikationsserver

Schichten

- Um die Komplexität von Anwendungen zu reduzieren bzw. die Modularität zu erhöhen, wird auch bei Softwarearchitekturen ein Schichtenmodell eingesetzt, allerdings mit deutlich weniger Schichten, wie im ISO-Referenzmodell

2-Schicht-Architektur

- „Two-Tier-Architecture“
- Klassische Client-Server-Architektur
- Clientschicht (oft auf Fat-Client, d.h. Client übernimmt auch Logikaufgaben)
- Serverschicht (immer auf Fat-Server)

3-Schicht-Architektur

- „Three-Tier-Architecture“
- Aktuelle Realisierungen beruhen meist auf diesem Modell
- Präsentationsschicht (Client Tier)
- Logikschicht (Middle Tier)
- Datenhaltungsschicht (Database Server Tier)

Client Tier

- „Front End“
- Benutzerschnittstelle (GUI)
- Benutzereingaben
- Kommunikation mit der Middle Tier
- i.A. keine direkte Kommunikation mit dem Back End
- Oft im Webbrowser realisiert

Middle Tier

- „Application-Server-Tier“, „Enterprise Tier“, „Businesslogikschicht“
- Umsetzung der Businesslogik
- Kommunikation mit den anderen beiden Schichten (getrennt von einander)
- Realisierung mit einem Anwendungsserver

Database Server Tier

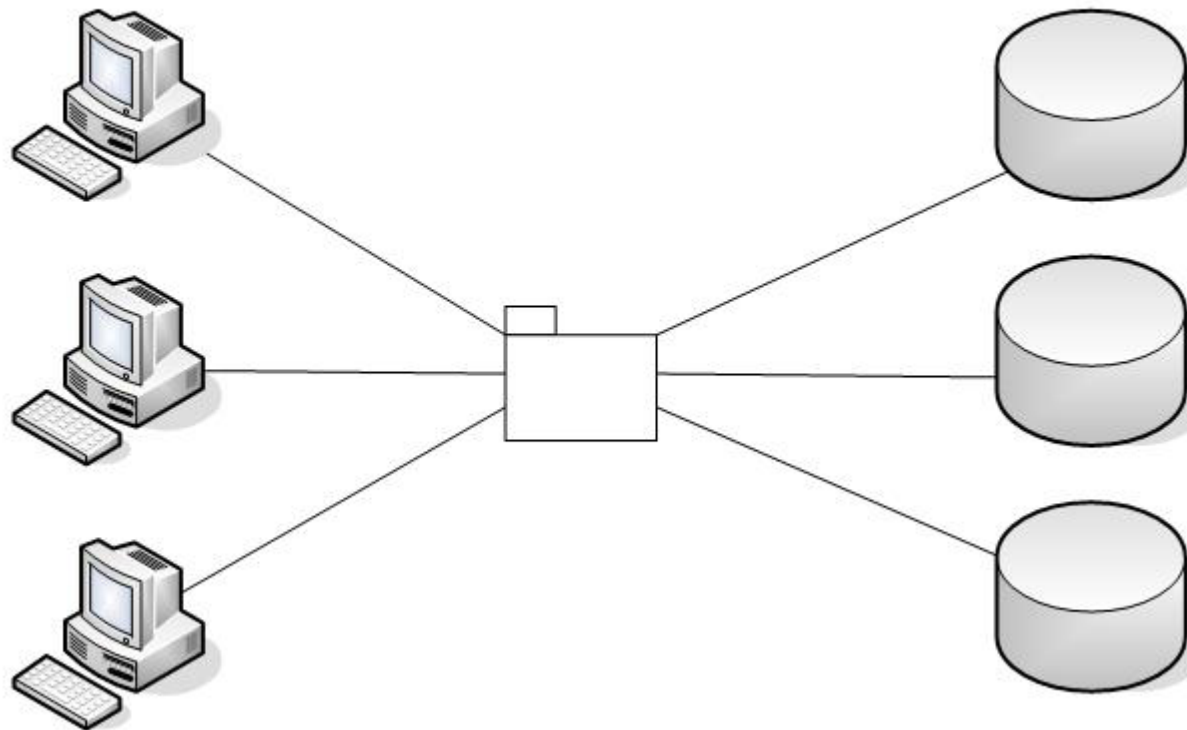
- „Back End“
- Datenhaltung i.A. auf einem Datenbankserver
- Kommunikation mit der Middle Tier

3-Schicht-Architektur-Beispiel

Anwendungsschicht

Domänenschicht

Datenschicht



Quelle: <http://de.wikipedia.org/wiki/3-Tier-Architektur#Drei-Schichten-Architektur>

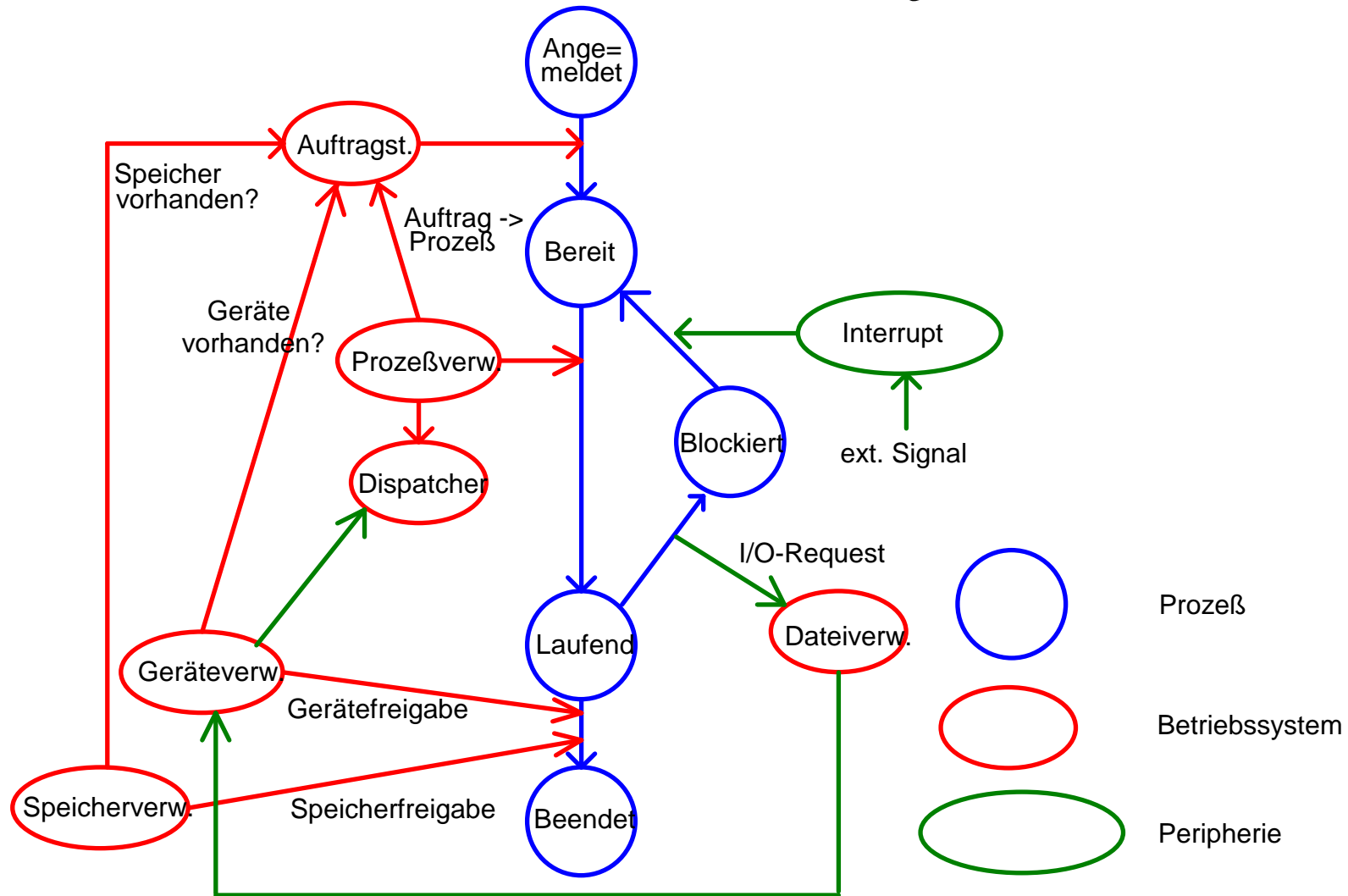
Mehrschichtarchitekturen

- Hier wird das 3-Schichten-Modell erweitert, in dem einzelne Aufgaben aus einer der drei Schichten herausgelöst werden und in einer eigenen Schicht implementiert werden
- z.B.: Präsentation, Steuerung, Datenzugriff

III 3 Algorithmen & Synchronisation

- Prozeßkommunikation
- Semaphoren
- Pipes
- RPC

Prozeßlebenszyklus



Prozess-Synchronisation

Prozesse sind oft voneinander abhängig

z.B.:

- Schreibprozess muß auf Daten warten
- Zwei Prozesse wollen gleich HW nutzen

=> 2 Grundklassen der Synchronisation

- Wechselseitiger Ausschluss
- Prozesskooperation

Das Semaphorkonzept

- Binärsemaphor
 - Ampel,
 - Flag,
 - ...
- allgemeine Semaphore
 - Zähler

Pipe-Konzept

Manche Betriebssysteme bzw. Prozessoren stellen für die einfachere Synchronisation sogenannte Pipes ("Röhren" für einen Datenstrom) zur Verfügung. Dabei werden die Ergebnisse eines Befehls als Eingabe für den nächsten Befehl verwendet (FIFO).

z.B.: `dir | sort | more >filelist.txt`

Pipe-Symbole

- | Anonyme Pipe
- < Named Pipe für die Eingabe
- > Named Pipe für die Ausgabe
- >> Named Pipe für die Ausgabe (append)
- [unit]> Angabe einer Ein- oder Ausgabeeinheit (z.B. 2>&1)

RPC

- Remote Procedure Call
- RFC 707, 1057, 5531
- Viele inkompatible Implementierungen
 - XNS Xerox Network System
 - Sun ONC RPC (Open Network Computing)
 - DCE RPC (Distributed Computing Env.)
 - MSRPC → DCOM → .NET Remoting

RPC – Ablauf

- Client sendet eine Anfrage an den Server
- Server verarbeitet diese Anfrage
- Server schickt eine Antwort
- Client arbeitet mit der Antwort weiter

RPC: SOAP

- Simple Object Access Protocol →
Eigenständiges Acronym
- Weiterentwicklung von XML-RPC
- Spezifikationen:
<http://www.w3.org/TR/soap/>

SOAP Request Beispiel

```
<?xml version="1.0"?>
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-
  envelope">
  <s:Body>
    <m:TitleInDatabase xmlns:m="http://www.spg.at/db/soap">
      NVS1 und Coufal
    </m:TitleInDatabase>
  </s:Body>
</s:Envelope>
```

SOAP Reply Beispiel

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header>
    <m:RequestID xmlns:m="http://www.spg.de/db/soap">FC</m:RequestID>
  </s:Header>
  <s:Body>
    <m:DbResponse xmlns:m="http://www.spg.at/db/soap">
      <m:title value=„NVS1 und Coufal">
        <m:Choice value="1">NVS1_H4.PDF</m:Choice>
        <m:Choice value="2">NVS1_B34.PDF</m:Choice>
      </m:title>
    </m:DbResponse>
  </s:Body>
</s:Envelope>
```