

Web-Programmierung (WPR)

Vorlesung XII.

Vergleich Server-Plattformen

<mailto:wpr@gruner.org>

16 Vergleich Server-Plattformen

- Technologien
 - Perl/CGI
 - Einsatzgebiete: Kleine Websites, semiprofessioneller Bereich
 - Pro's:
 - Plattform/Serverneutralität
 - Gute Providerunterstützung
 - Con's
 - Prozessorzeugung = Bottle neck

16 Vergleich Server-Plattformen

- Technologien
 - PHP
 - Einsatzgebiet: Kleine bis mittelgroße Websites
 - Pro's:
 - Schnelle Erlernbarkeit / einfache Handhabung
 - Eignung zur Erstellung von Prototypen
 - Gute Datenbankunterstützung
 - Hohe Plattformneutralität
 - Session Tracking ab PHP4
 - Con's:
 - Unzureichende Trennung Präsentation u. Code
 - Eigene Modulentwicklung = aufwendig

16 Vergleich Server-Plattformen

- Technologien
 - JavaServlets und JSP
 - Einsatzgebiet: Mittelgroße bis sehr komplexe Websites
 - Pro's:
 - Skalierbarkeit (abhängig vom Servlet Container)
 - Managed Components
 - Einfaches Deployment
 - Saubere Trennung von Präsentation u. Logik möglich
 - Vorteile von Java (Plattformunabhängigkeit, Bibliotheken, open source,....)
 - Anbindung bestehender Applikation
 - Con's
 - Längere Einarbeitungszeit
 - Nicht geeignet für Prototypen

16 Vergleich Server-Plattformen

- Technologien
 - Active Server Pages
 - Einsatzgebiete: mittelgroße bis große WebSites
 - Pro's:
 - Gute Performance, gute Skalierbarkeit
 - Auswahl an Programmiersprachen
 - Nutzung beliebiger COM-Objekte
 - Con's:
 - Unzureichende Trennung von Präsentation u. Code
 - TAG-Definition nicht möglich
 - Interpreter, keine Erzeugung von ausführbarem Code
 - Integration von eigenen Komponenten (=>Instabilität)

Anmerkung: => Entwicklung .NET-Plattform

16 Vergleich Server-Plattformen

Kriterium \ Plattform	Perl/CGI	PHP	Servlets/JSP	ASP
Wartbarkeit	+	++	+++++	++++
Performance	+	+++	+++++	+++++
Skalierbarkeit	+	+++	+++++	++++
Verfügbarkeit	+++++	++++	+++++	+++
Plattformneutralität	+++++	+++	++++	
Abstraktionslevel	+	+++	+++++	++++
Erweiterbarkeit	+++	++	+++++	++++
Dokumentation	++++	++++	+++	+++++
Support	+++++	+++++	+++	+++
Tool-Unterstützung	+	+	++++	+++++

!! Möglichkeiten einer Plattform müssen durch geeignete Design- und Architektur-Maßnahmen erst genutzt werden.

16 Vergleich Server-Plattformen

- Kriterien: Wartbarkeit

????????????????

16 Vergleich Server-Plattformen

- Kriterien: Wartbarkeit
 - Wartbarkeit ist abhängig von der Organisationsform
 - Änderungen einer Komponente
=> keine Änderungen an anderen Komponenten
- Zielerreichung durch:
Modularisierung auf 3 Ebenen:
 - Architektur
 - Design
 - Präsentation

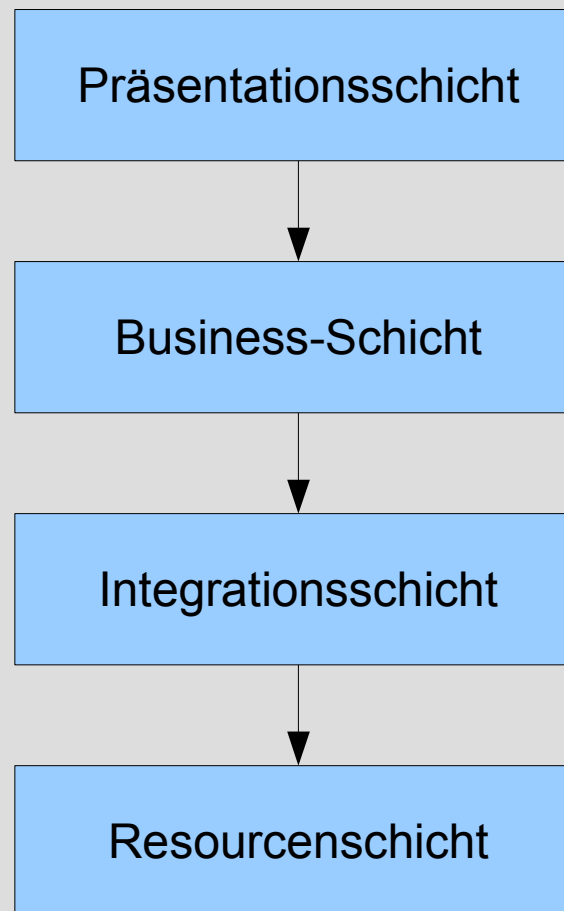
16 Vergleich Server-Plattformen

- Kriterien: Wartbarkeit -> Architektur

????????????????

16 Vergleich Server-Plattformen

- Kriterien: Wartbarkeit -> Architektur



16 Vergleich Server-Plattformen

- Kriterien: Wartbarkeit -> Design

????????????????????

16 Vergleich Server-Plattformen

- Kriterien: Wartbarkeit -> Design
- Risiken kapseln
 - ◊ Technische Risiken: z.B.: Datenbank
 - ◊ Fachliche Risiken: Zukünftige Features

=> lose Ankopplung dieser Komponenten

16 Vergleich Server-Plattformen

- Kriterien: Wartbarkeit -> Präsentation

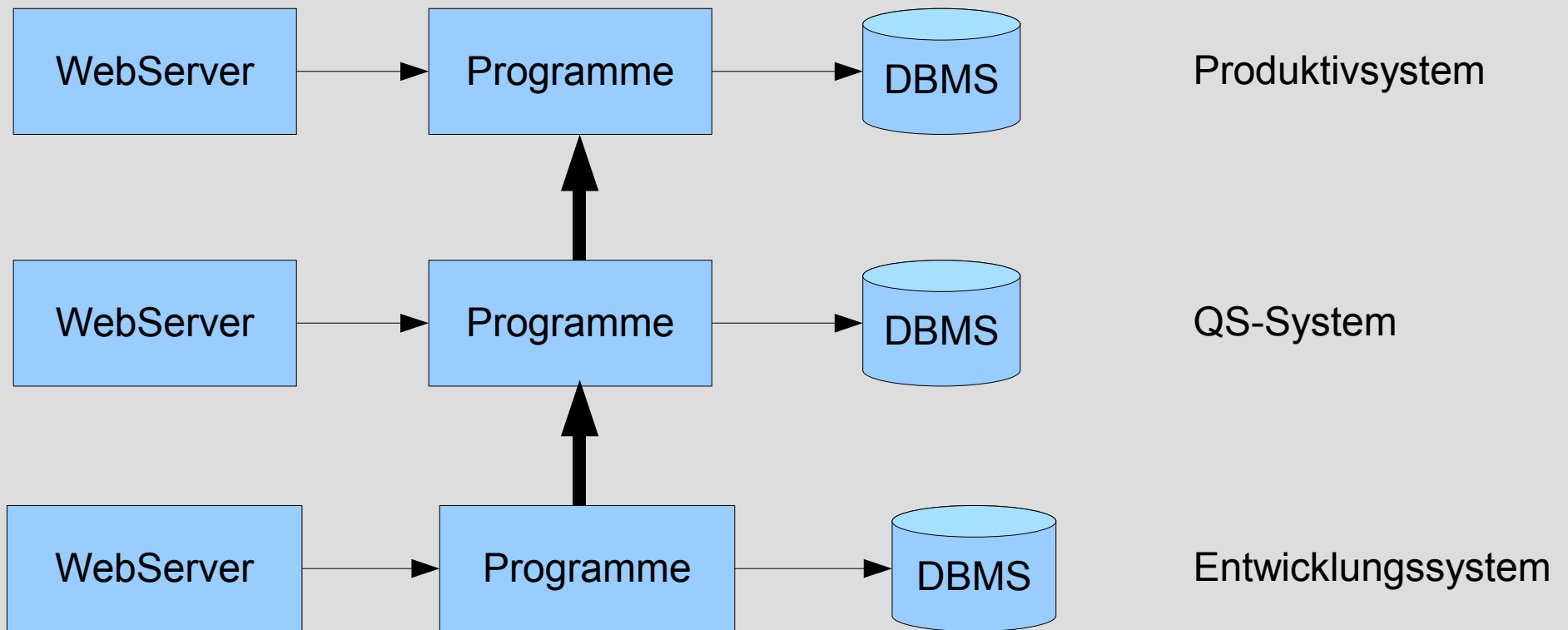
????????????????????

16 Vergleich Server-Plattformen

- Kriterien: Wartbarkeit -> Präsentation
- WICHTIG:
Strenge Trennung Inhalt und Layout

16 Vergleich Server-Plattformen

- Kriterien: Wartbarkeit -> Staging



16 Vergleich Server-Plattformen

- Kriterien: Performance

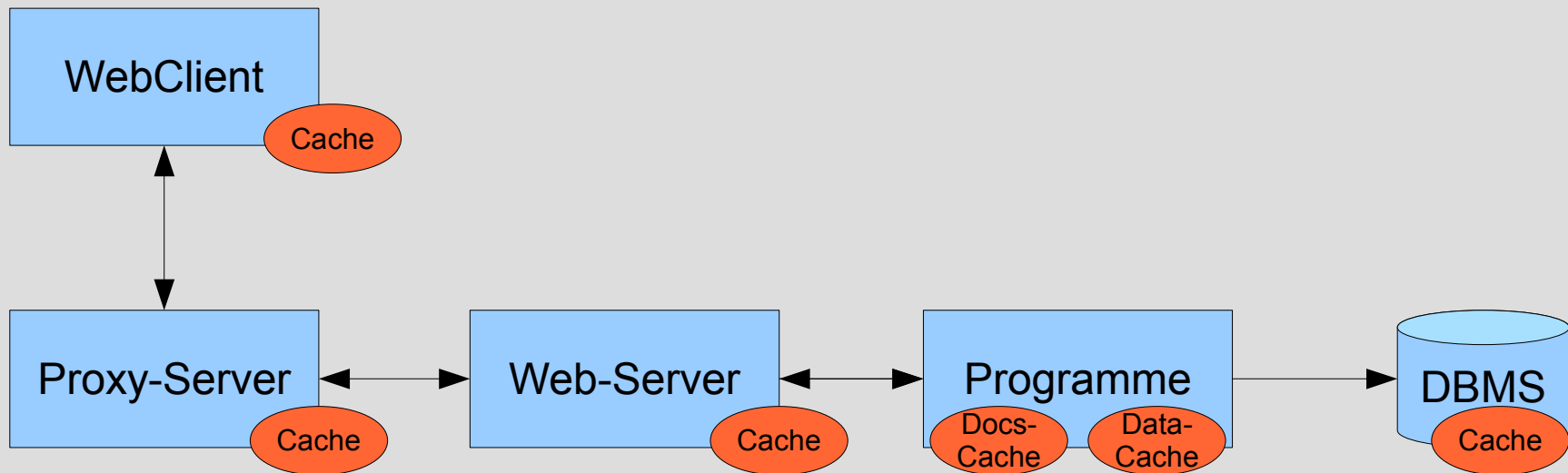
????????????????

16 Vergleich Server-Plattformen

- Kriterien: Performance
Web-Performance
„Antwortzeit auf den Request eines einzelnen
WebClient“
- Kriterien
 - ◆ Bandbreite des Netzwerkanschlusses
 - ◆ Programmcode
 - ◆ Design
 - ◆ Architektur !

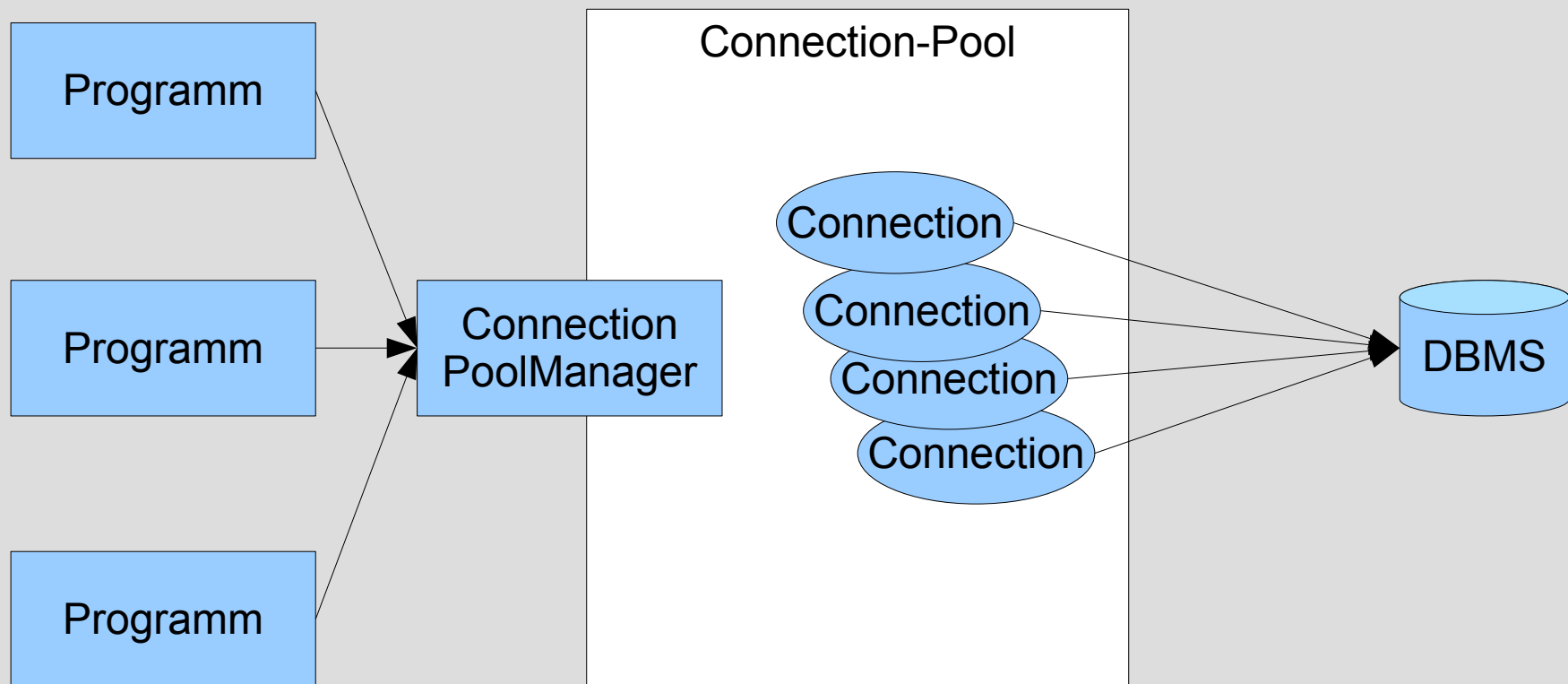
16 Vergleich Server-Plattformen

- Kriterien: Performance-> Architektur
 - Caching



16 Vergleich Server-Plattformen

- Kriterien: Performance-> Architektur
 - Pooling



16 Vergleich Server-Plattformen

- Kriterien: Skalierbarkeit

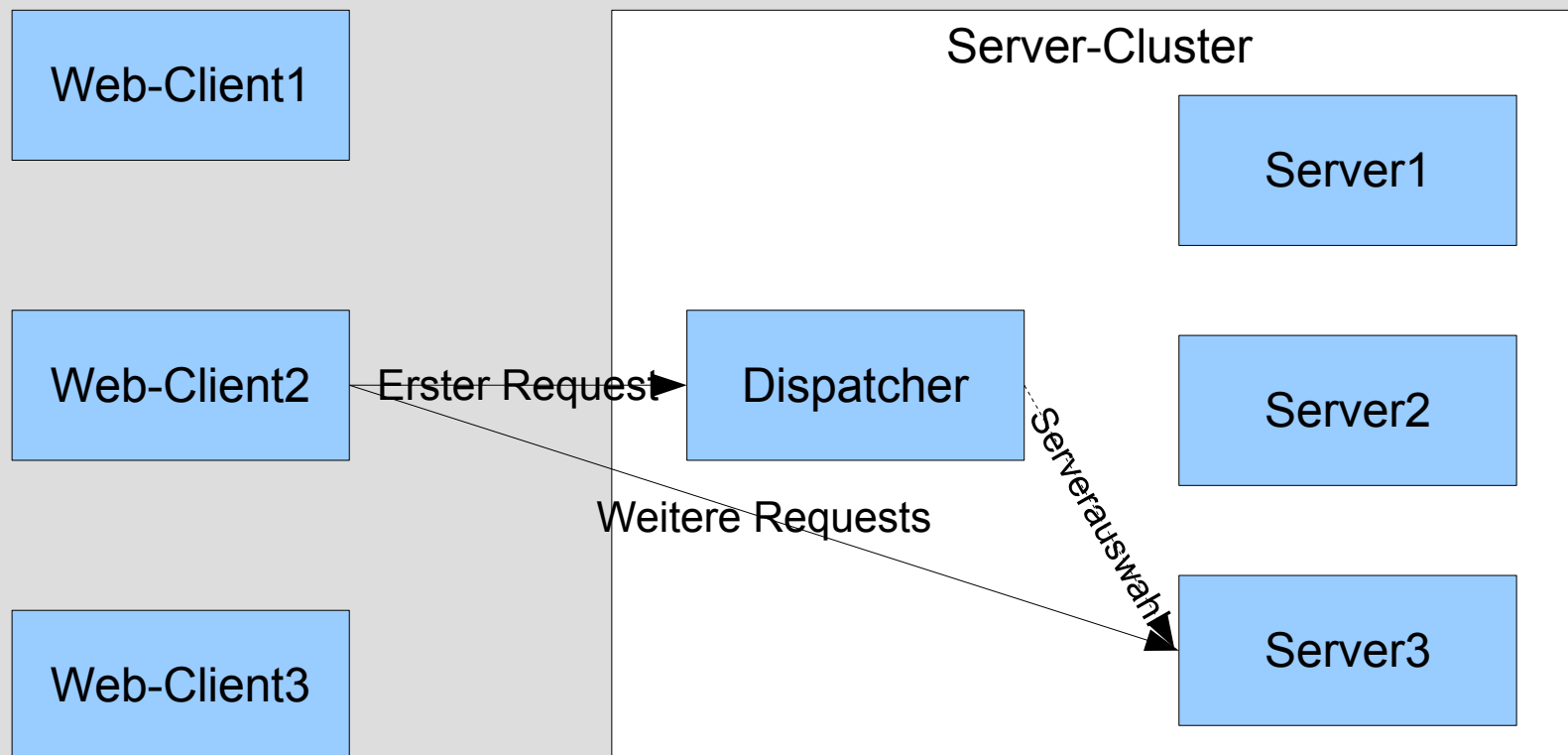
????????????????????

16 Vergleich Server-Plattformen

- Kriterien: Skalierbarkeit
„Beibehaltung definierbarer Antwortzeiten bei einer zunehmenden Anzahl gleichzeitiger Benutzer“
 - Skalierbarkeit
=> eine Frage der Systemarchitektur
- Lösungsansätze:
- Ressourcen-Pooling
 - Server-Cluster

16 Vergleich Server-Plattformen

- Kriterien: Skalierbarkeit-> Architektur
 - Clustering



16 Vergleich Server-Plattformen

- Kriterien: Verfügbarkeit

????????????????

16 Vergleich Server-Plattformen

- Kriterien: Verfügbarkeit

Verfügbarkeit des Systems

Anforderungen häufig: 99,95 %

Heutige Systeme laufen: 24 x 7 h

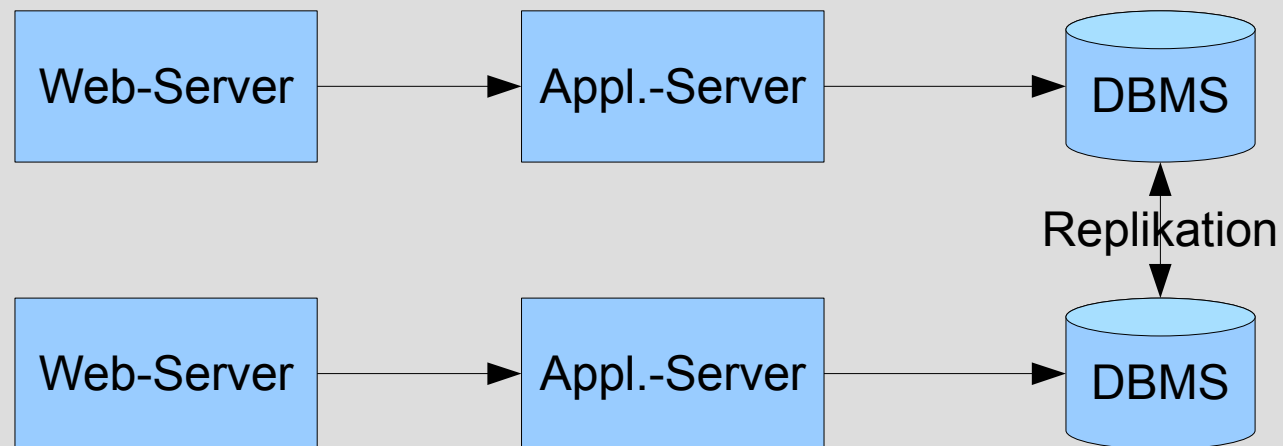
(168 h , 10080 s)

=> erlaubt: 504s Ausfall in der Woche

=> ca. 8,4 h im Jahr

16 Vergleich Server-Plattformen

- Kriterien: Verfügbarkeit
 - ♦ Lösung: Redundanz



- ♦ Ausgereifte Lösung: Transparent für Client

16 Vergleich Server-Plattformen

- Kriterien: Verfügbarkeit
 - ◆ Ausgereifte Lösung: Transparent für Client
 - ◆ Session-Failover
 - Übernahme der Session Daten
 - Hauptspeicher
 - Dateisystem
 - Datenbank
 - ◆ WICHTIG: Alle Ressourcen müssen gedoppelt sein
„Single Point of Failure“

16 Vergleich Server-Plattformen

- Kriterien: Offenheit

????????????????????

16 Vergleich Server-Plattformen

- Kriterien: Offenheit

Um für zukünftige Entwicklungen vorbereitet zu sein

- Einsatz von Standards
- Proprietären Techniken vermeiden

=> Interoperabilität

=> Portabilität

=> Schutz von Investitionen

16 Vergleich Server-Plattformen

- Kriterien: Intergrationsfähigkeit

????????????????????

16 Vergleich Server-Plattformen

- Kriterien: Integrationsfähigkeit
 - ♦ Anbindung von bestehenden Diensten/Applikationen an eine Web-Applikation
 - Besondere Schwierigkeit bei Systemdiensten
 - ♦ Lösungsansatz:
 - Einsatz von Standards

Siehe auf Kriterium Offenheit

17 Schichtenarchitekturen

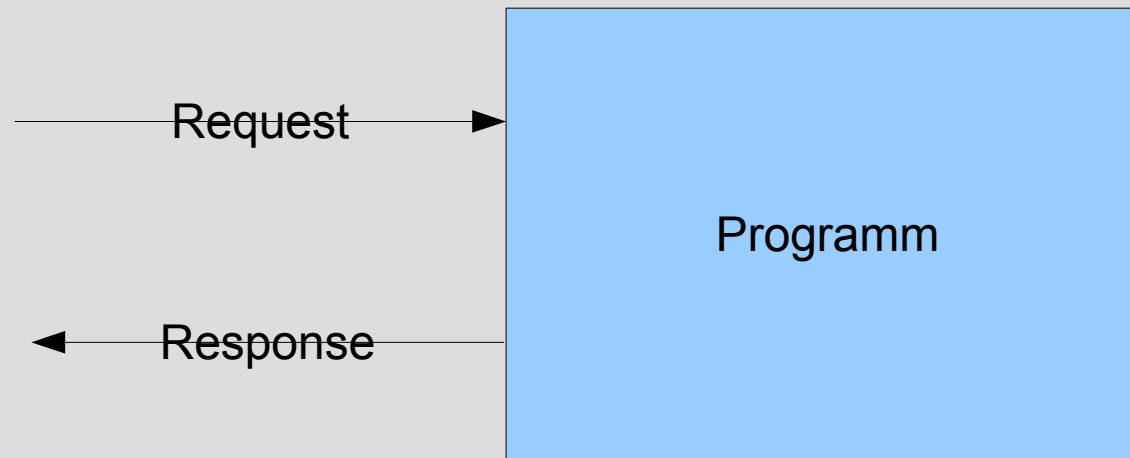
- Client/Server

- ◆ 2 schichtige Archit.
 - Hohe Prozessorlast
 - Hohe Netzlast
- ◆ 3-schichtige Archit.
 - Präsentation
 - Geschäftslogik
 - Daten

Kriterium
Wartbarkeit
Performance
Skalierbarkeit
Verfügbarkeit
Plattformneutralität
Abstraktionslevel
Erweiterbarkeit
Dokumentation
Support
Tool-Unterstützung

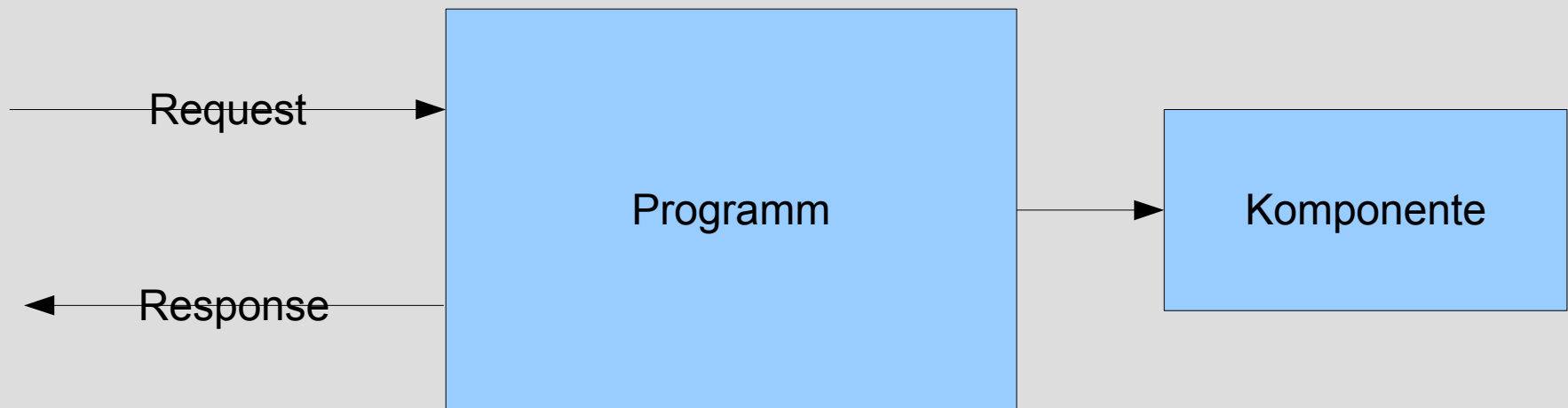
17 Schichtenarchitekturen

- Monolithischer Ansatz



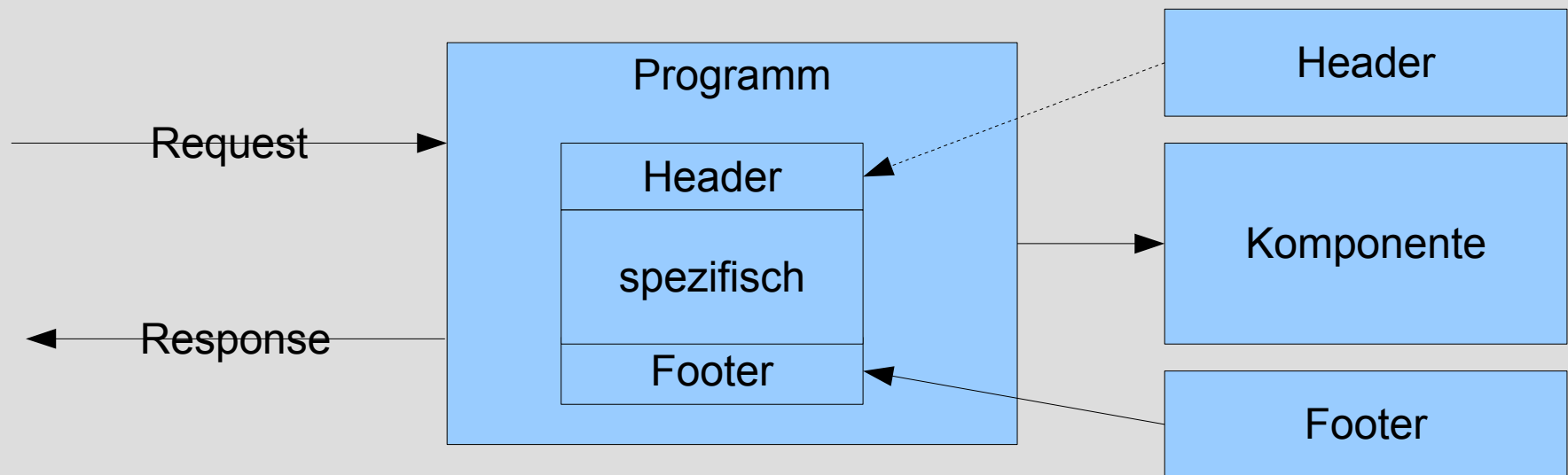
17 Schichtenarchitekturen

- Modularisierung der Logik



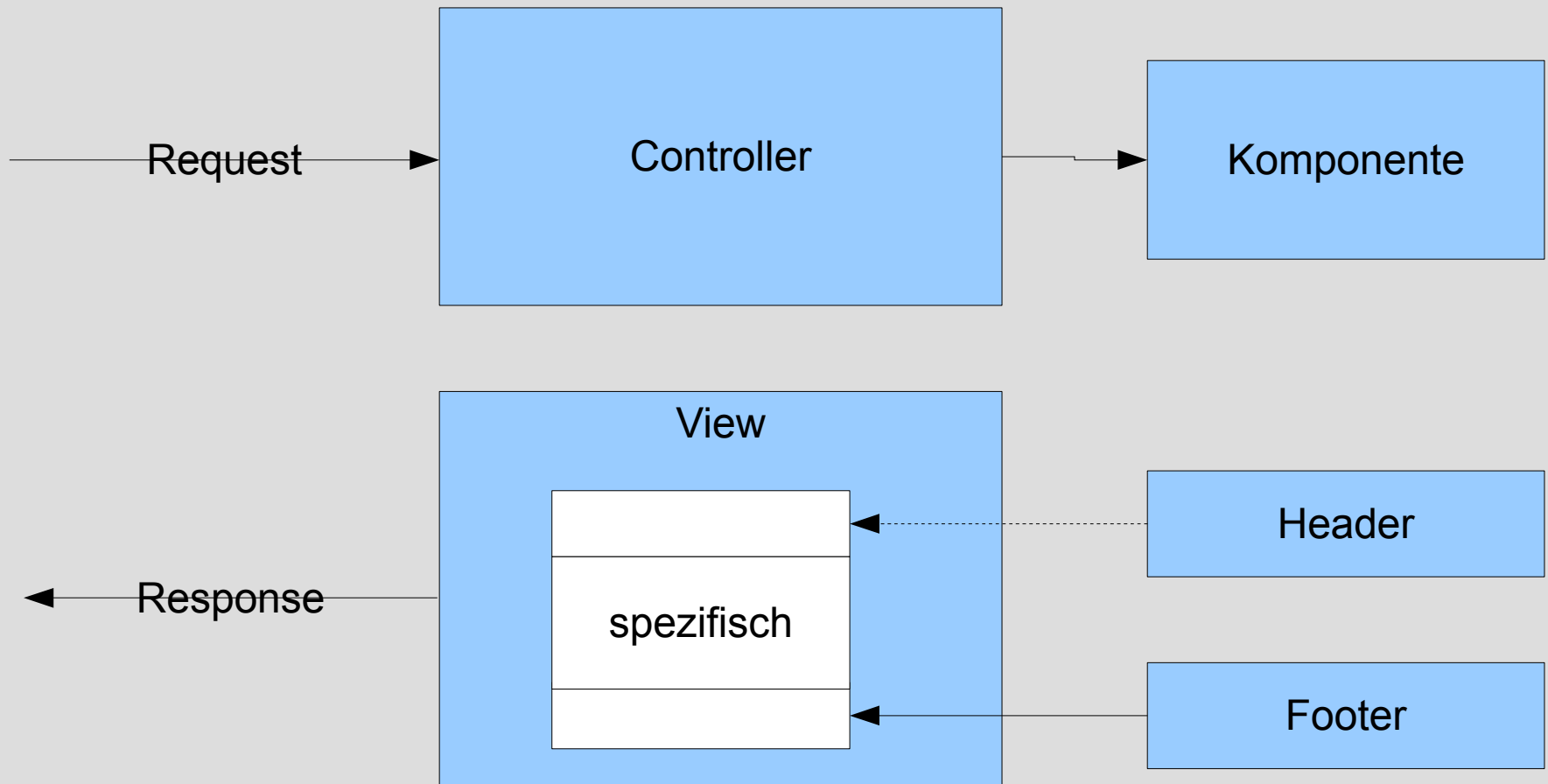
17 Schichtenarchitekturen

- Modularisierung der Präsentation



17 Schichtenarchitekturen

- Model-View-Controller



17 Schichtenarchitekturen

- Präsentationsschicht
 - ♦ Verantwortlichkeiten
 - HTTP-Management
 - Steuerung der Applikation
 - Session Management
 - Zugriff auf Business-Schicht
 - Auswahl und Aufbereitung der Präsentation
 - ♦ Standardaufgaben
 - Logging
 - Zugriffskontrolle
 - Validierung von Benutzereingaben

Ende der heutigen Vorlesung

