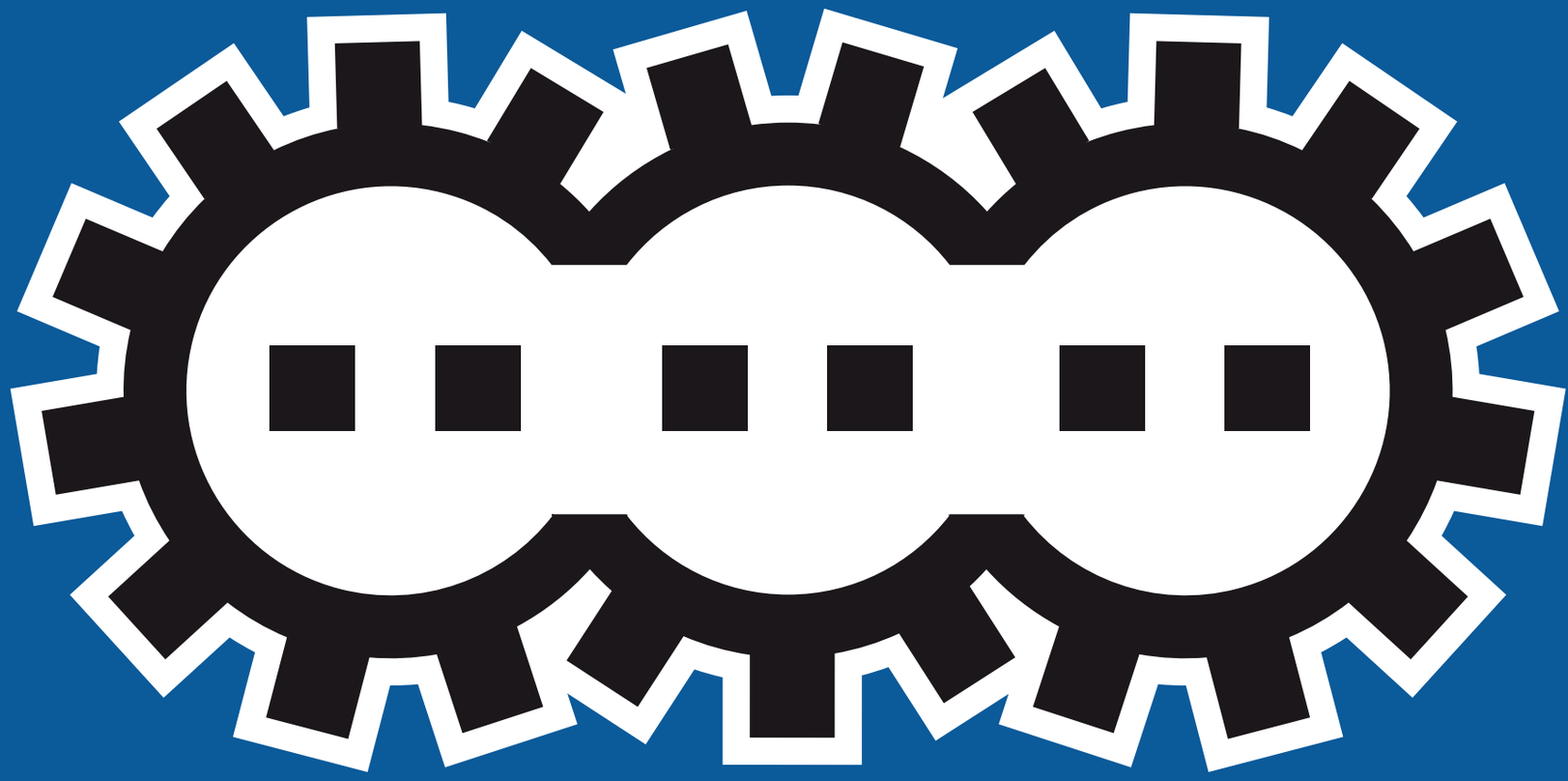
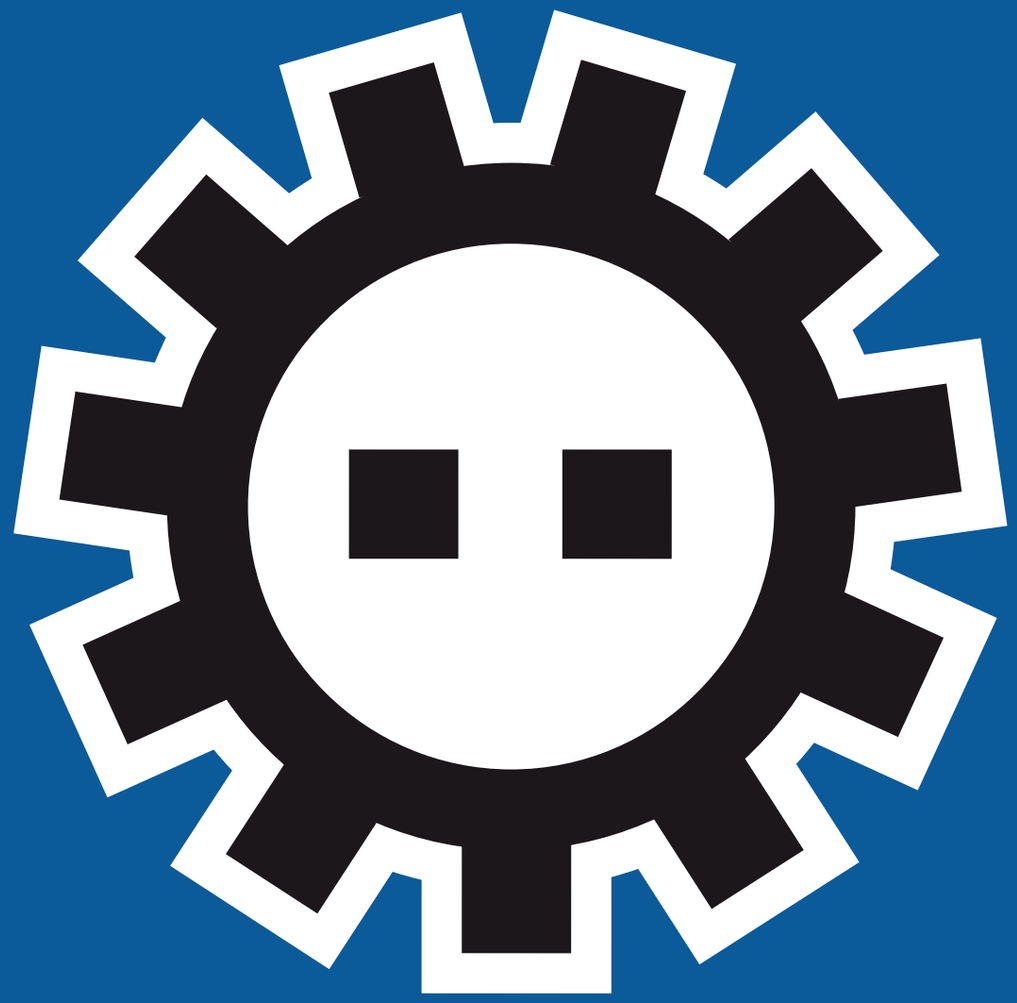


Software Engineering 2

08 - Web Application Server

Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen



Motivation

Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Wie sind HTML-Seiten aus dem Internet erreichbar?

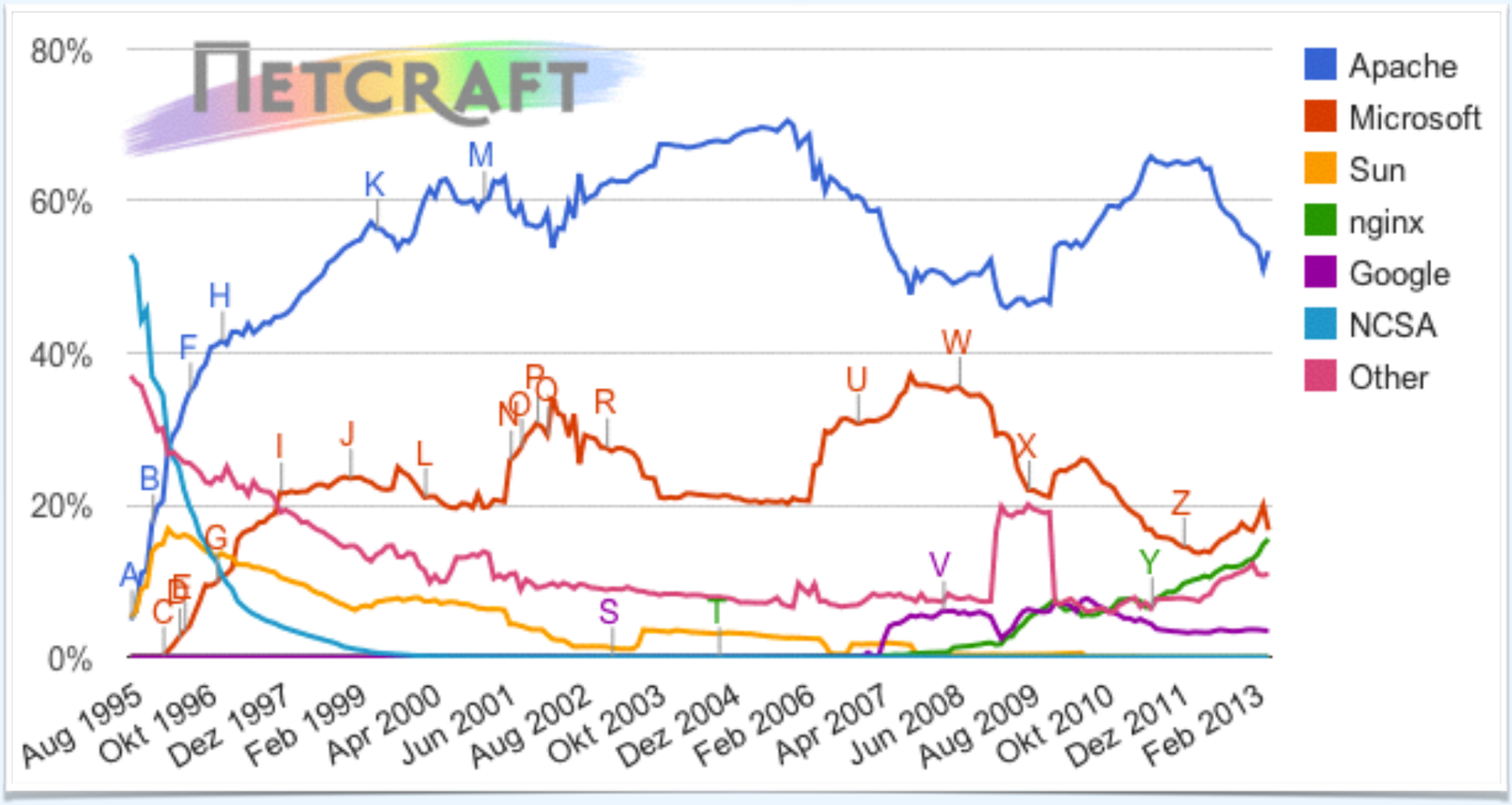
Web Server

Web-Server

- Virtuelle Verzeichnisse
  - Abbildung von URL auf Dateien im Filesystem
  - Ausliefern einer Datei
- Zugriffskontrolle
  - Passwörter für geschützte Ressourcen
  - Zugang in Abhängigkeit von IP-Adressen / Hostnamen
- Umleitungen / Redirects
- Protokollierung
- Virtuelle Hosts

HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Web Server



Hersteller	Anzahl Server	Marktanteil
Apache	331,112,893	51.01%
Microsoft	129,516,421	19.95%
nginx	96,115,847	14.81%
Google	22,707,568	3.5%

Quelle: Netcraft Mai 2013 http://www.netcraft.com/survey

Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Apache Web-Server

- mit Abstand beliebtester Web Server: ca. 60% Marktanteil
- Grundlage ist NCSA httpd 1.3 von 1995
- frei verfügbar (open source)
- für die wichtigsten Plattformen verfügbar:
  - Unix: AIX, Linux, Solaris, HPUX,...
  - Win32, Win64, MacOS X, OS/2
- Größe der Installation: ca. 15 MB
- Quelle : http://www.apache.org
  - Dokumentation, Quellen, Binaries, etc.

HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Apache Web-Server

- Ziele: Sicherheit , Effizienz, Skalierbarkeit, Erweiterbarkeit
- Design:
  - Core: wenig Features, implementiert API
  - Module
    - » nutzen API für Implementierung von Funktionalität
    - » als Schnittstelle dienen sogenannte Handler
    - » Basis-Funktionalität auch als Modul implementiert
- Auslegung auf hohe Performance
  - Prozessstruktur: Parent- und Child-Server

HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Web-Server Prozessse

- Parent-Server
  - Verwaltung der Child-Server zur Laufzeit
  - je nach Last werden zusätzliche Child-Server erzeugt
  - läuft mit vollen Rechten
- Child-Server
  - verarbeitet die Anfragen:
    - Initialisierung
    - Warten auf Anfrage
    - Bearbeitung der Anfrage
    - Versenden der Antwort
  - läuft mit eingeschränkten Rechten

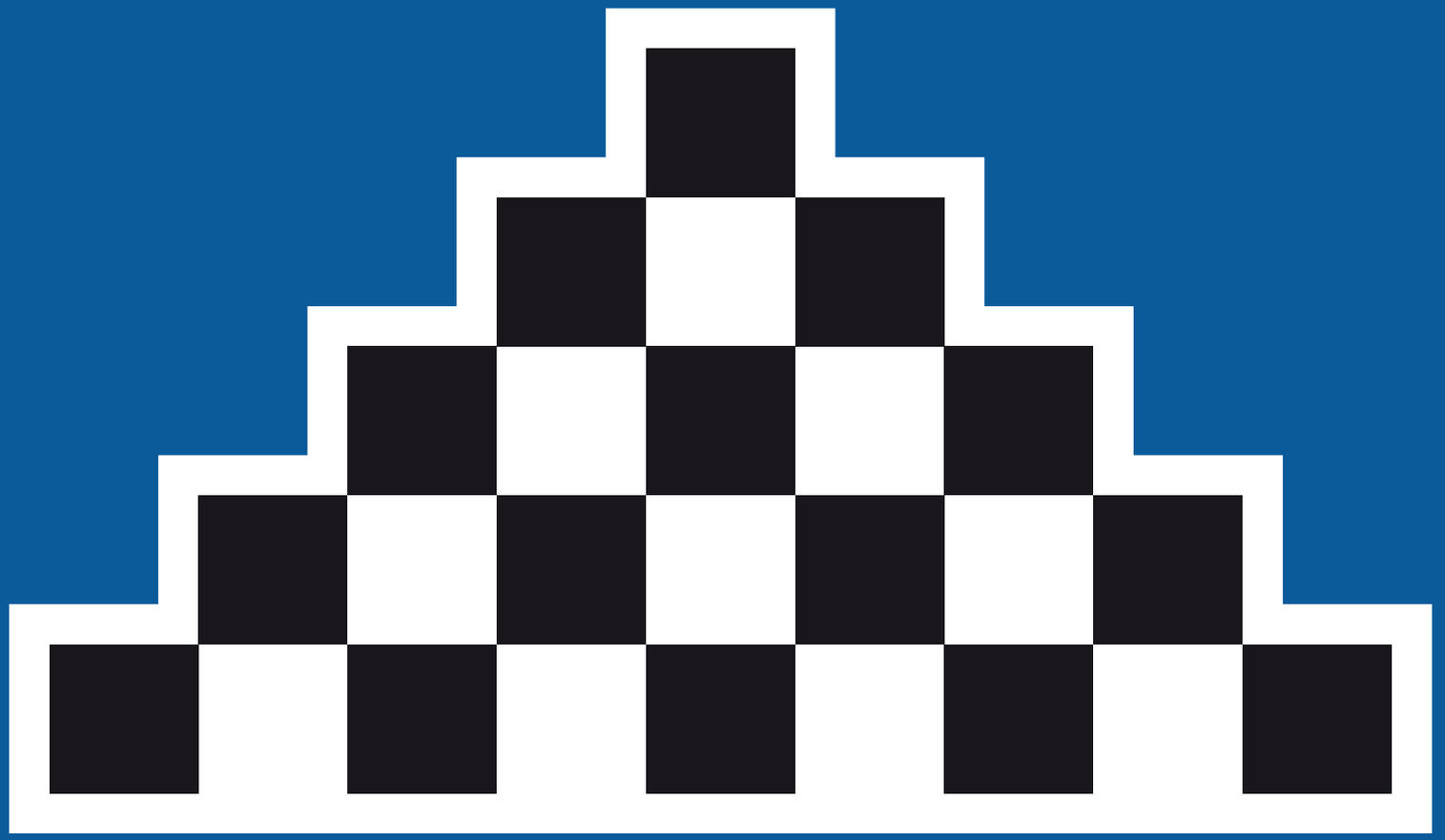
HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Web-Server Access-Log

- Jede Zeile entspricht einem Zugriff
- Durch Leerzeichen getrennte Informationen:
  - IP-Adresse des anfragenden Clients
  - Zwei leere Felder (gekennzeichnet durch -)
  - Der Zeitpunkt der Anfrage
  - Die erste Zeile der Anfrage
  - Der Statuscode der Antwort
  - Die Länge der übermittelten Antwort
- dahinter können, je nach Konfiguration, weitere Informationen auftauchen z.B: Referer, User-Agent

```
193.25.22.51 - - [17/Apr/2014:09:05:27 +0200] "GET /de/aktuell.html HTTP/1.1" 200 5138
193.25.22.51 - - [17/Apr/2014:09:05:27 +0200] "GET /resources/images/small.png HTTP/1.1" 200 1276
193.25.22.51 - - [17/Apr/2014:09:05:27 +0200] "GET /resources/images/logo.png HTTP/1.1" 200 4995
193.25.22.51 - - [17/Apr/2014:09:05:27 +0200] "GET /files/image-63340443.png HTTP/1.1" 200 170766
```


HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen



Grundlagen

Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

von Web-Servern zu Anwendungsservern

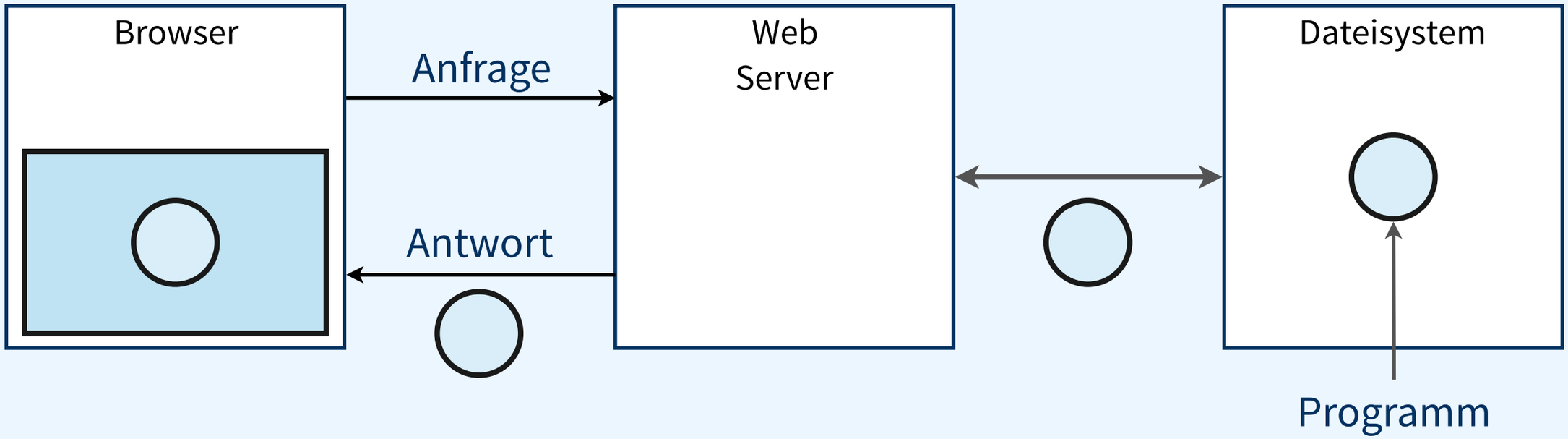
Dynamische Web-Seiten

einfacher Ansatz

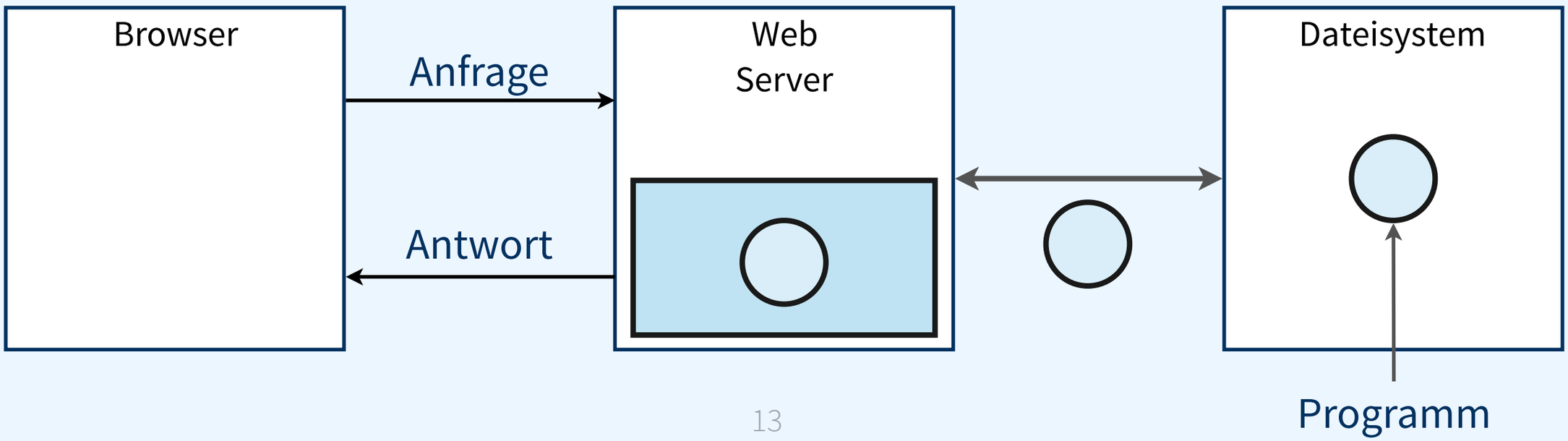
- HTML-Dateien auf den Web-Server legen
- Sämtliche Dynamik mit Javascript programmieren
- Schwerwiegende Nachteile:
  - nur lesender Zugriff (Wikis, Blogs, etc. wären nicht möglich)
  - Sicherheitsrisiken (Client kann nicht vertraut werden)
- Alternative Client-Technologien helfen dabei nicht
  - z.B.: Flash, Silverlight, Java-Applets
  - Annahmen sind schwierig zu treffen
- Server-seitige Technologien erweitern unsere Möglichkeiten

HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Verarbeitung der Programme im Client (z.B. JavaScript):



Verarbeitung der Programme auf dem Server:



HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Statisch vs. Dynamisch

- Statische Server (Web-Server)
  - Anfrage nach Datei
  - lokalisieren und ausliefern
- Dynamische Server (Application-Server)
  - (Teile der) Antwort vor Auslieferung generiert
  - Datenbankoperationen
  - spezielle Aufbereitung (z.B. Bilder anpassen)
  - Benutzerverwaltung / Sicherheit

HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

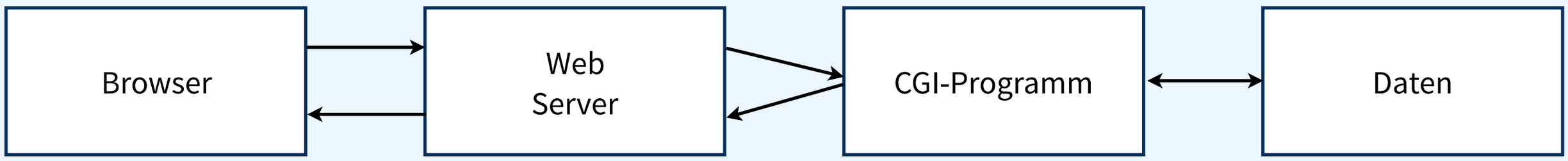
server-seitige Technologien

- CGI (Common Gateway Interface)
- Java Servlets
- Server Side Scripting
  - PHP (PHP: Hypertext Preprocessor)
  - Python
  - ASP (Microsoft Active Server Pages)
  - JSP (Java Server Pages)

HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Common Gateway Interface

- Protokoll zwischen Web-Server und Anwendungsprogramm
  - Web-Server ruft Programm auf dem Server auf
  - über Umgebungsvariablen werden Daten ausgetauscht:
    - Anfrage-Daten
    - Informationen zum Web-Server
- freier Standard, kostenlos, einfach
- jedes Programm verwendbar
  - Perl, Shell-Scripte, C, Pascal, Java, COBOL…



HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

CGI Vorteile

- Trennung in verschiedene Prozesse
  - HTTP-Server enthält allgemeine Web-Funktionen
  - anwendungsspezifisches in separater Applikation
- (Prozess-)Unabhängigkeit vom HTTP-Server
  - Applikation hängt nicht vom Server ab
  - Absturz der Anwendung beeinträchtigt nicht den Server
  - von nahezu allen Servern unterstützt
- einfache Verwendung
  - offener Standard
  - Programmierbibliotheken in vielen Sprachen

HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

CGI Nachteile

- schlechte Performance
  - Neustart der Programme bei jedem Aufruf
  - Performance hängt sehr stark von der Plattform ab
- Ausgabe sind immer komplette HTML-Seiten
  - oft ist aber HTML statisches Gerüst mit dynamischen Teilen
- Nur Erstellung von HTML-Ausgabe, kein Eingriff in andere Phasen (Zugriffsrechte, Logging)
- Sicherheitsprobleme, Browser führt Anwendungen auf Server aus

HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Web-Anwendungsarchitektur

Begriffe

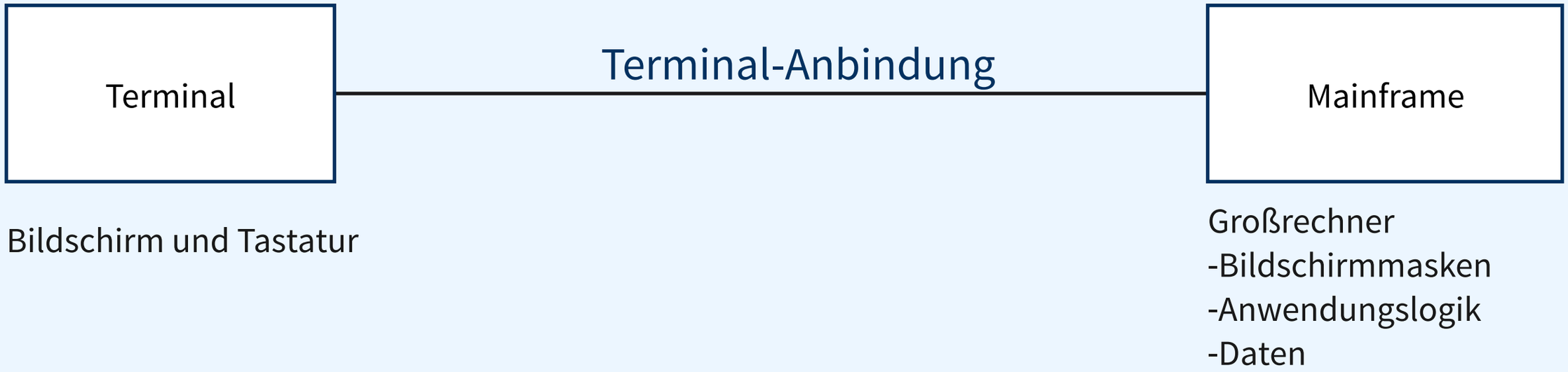
Architekturen von Web-Anwendungen

- Anwendungen sind Software-Systeme
- Unterstützen Menschen bei Tätigkeiten, z.B.:
  - Bearbeiten eines Schadens bei Versicherung
- Automatisiert Abläufe, z.B.:
  - Ausführung einer Überweisung
  - Steuert die Bremsen in einem Auto (ABS)
  - Führt ein Waschprogramm durch

HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Historische Entwicklung

Mainframes



HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

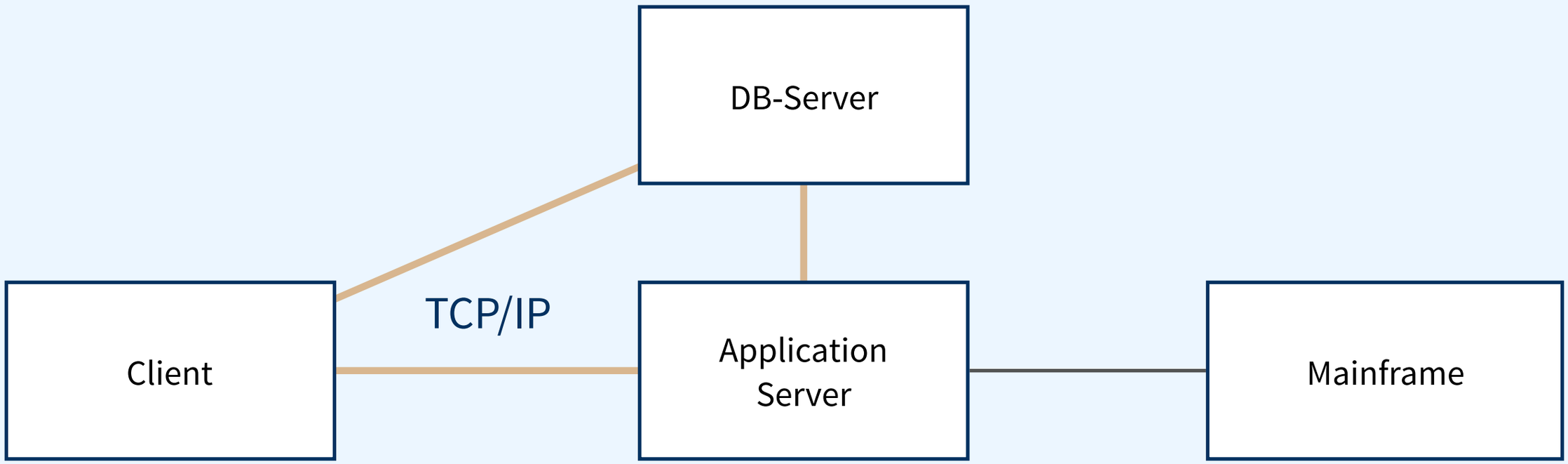
Mainframes

- Hardware und Betriebssysteme:
  - IBM, Siemens, …
- Datenmanagement
  - Dateisysteme
  - hierarchische Datenbanken
- Kommunikation:
  - proprietär
- Entwicklung:
  - Cobol, Fortran

HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Historische Entwicklung

Client/Server



Dezentralisierung!

Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Client/Server

- Hardware und Betriebssysteme:
  - Unix, Windows, Linux
- Datenmanagement
  - Dateisysteme
  - relationale Datenbanken
- Kommunikation:
  - TCP/IP
- Entwicklung:
  - C, C++

HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Begriffe

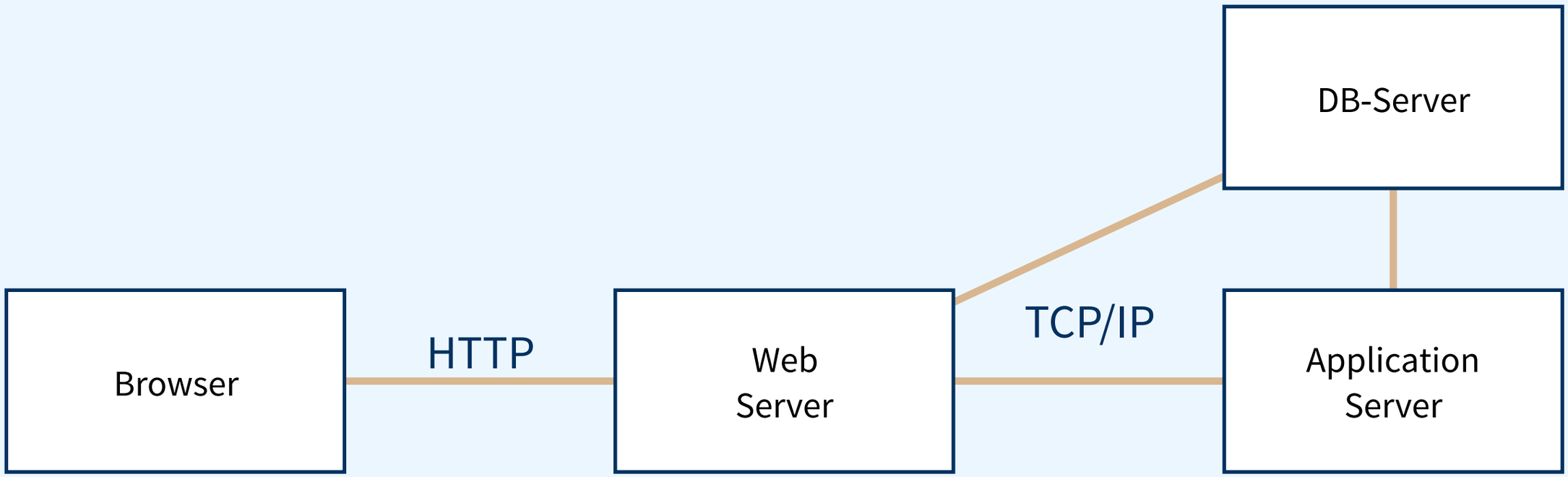
Architekturen von *Web-Anwendungen*

- Web-Anwendungen basieren auf Internet-Technologien
- Nutzung der Techniken des World Wide Web:
  - HTTP (Netzwerkprotokoll)
  - HTML (Dokumentenformat)
  - URL (Adressierung)
- Anwendungsbereiche:
  - Internetauftritte / E-Business
  - Intranetanwendungen

HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Historische Entwicklung

Web-Architektur



Arbeitsplatzrechner
-grafische Oberfläche
-Anwendungslogik

Stellt Zugang aus Internet/
Intranet bereit

Zentrale Rechner
-Anwendungslogik
-Datenhaltung

Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Web-Architektur

- Hardware und Betriebssysteme:
  - Unix, Windows, Linux
- Datenmanagement
  - relationale Datenbanken
  - Content-Repositories (CMS)
- Kommunikation:
  - HTTP, TCP/IP
- Entwicklung:
  - Java, C#, PHP,…,

HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Architektur

- Jedes System besitzt eine Architektur
  - Implizit
    - » Entsteht während der Entwicklung
    - » Akzeptabel bei ganz kleinen Anwendungen
    - » Vergleich traditionelle Architektur
      Für Hundehütte braucht man keinen Architekturentwurf
  - Explizit
    - » Separater bewusster Schritt im Entwicklungsprozess
    - » Größere Systeme benötigen explizite Architektur
    - » Vergleich: Für jedes größere Gebäude wird eine Architektur entworfen
- Explizite Architekturen bewirken viele Vorteile

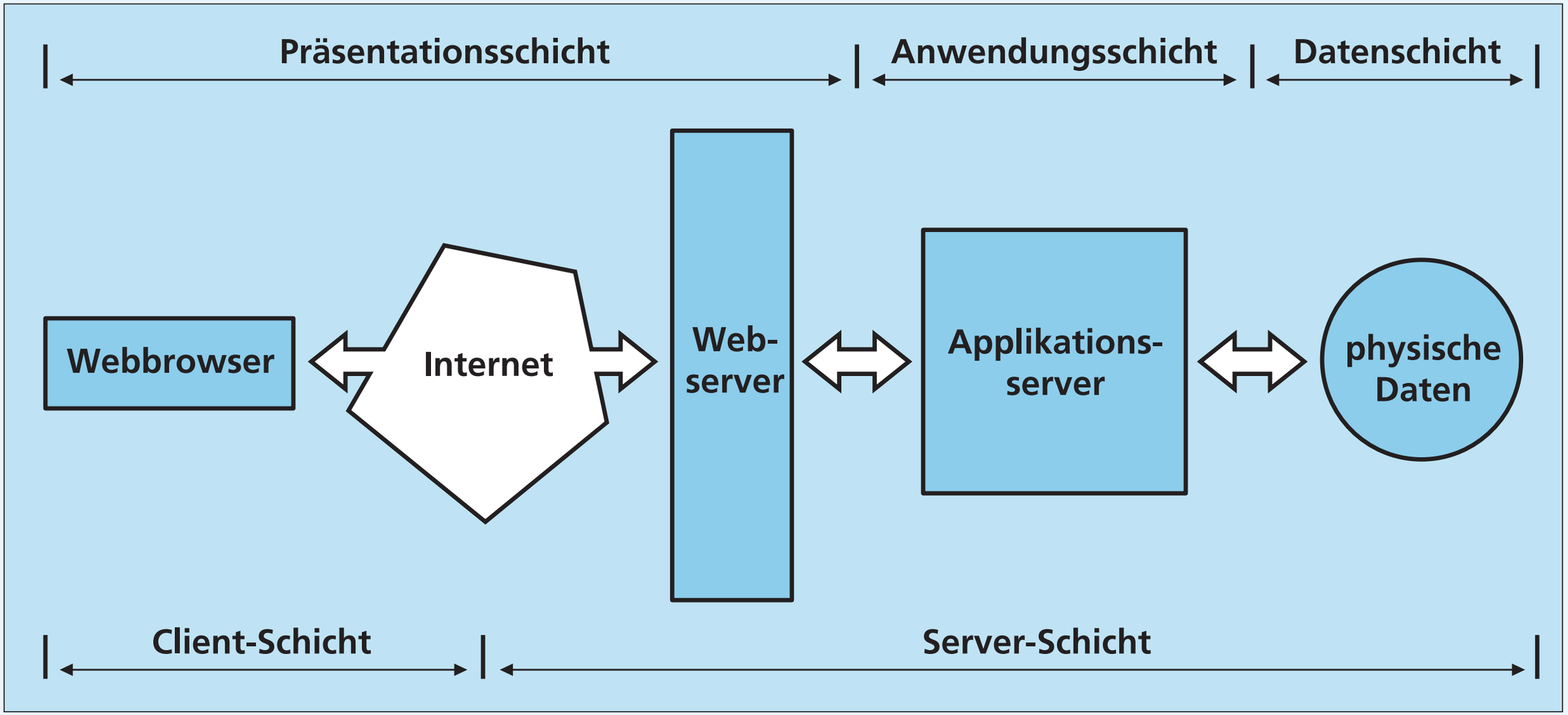
HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Architekturvorteile

- Reduktion der Komplexität der Anwendung
- Verteilung von Aufgaben im Entwicklerteam
- Zuordnung zu Systemkomponenten
- Einsatz von Standards
- Einsatz von Produkten
- Wiederverwendung von Architekturkomponenten
- Produktlinienansatz
  z.B.: Alle Web-Anwendungen eines Unternehmens verwenden die gleichen GUI-Elemente

HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Web Architektur



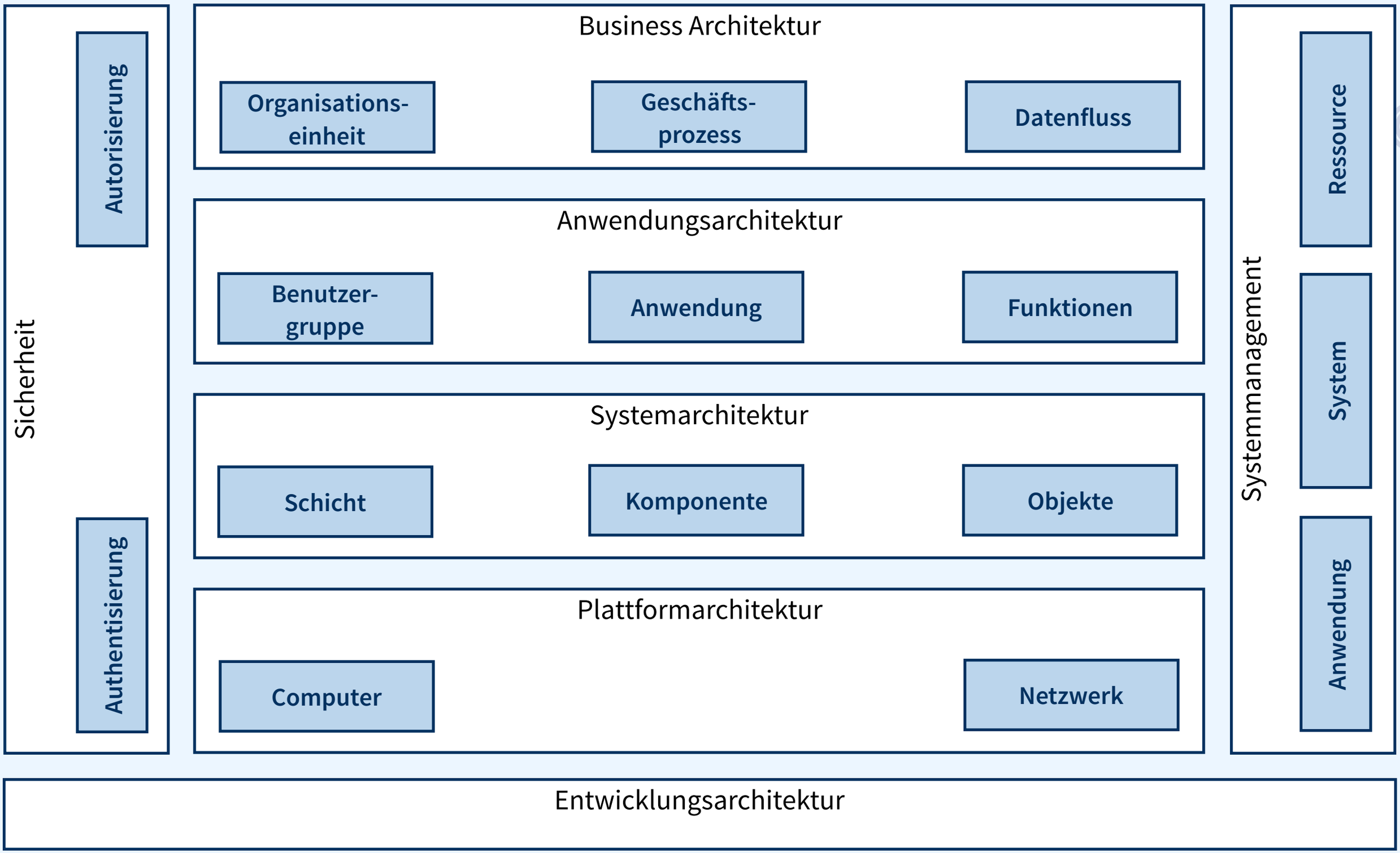
HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Vergleich

	Benutzung	Management	Entwicklung
Mainframe	-Einfach, "dumme" Terminals -Primitive Oberfläche -Verbindung zwischen Terminal und Rechner paketorientiert	-Zentral -Einfache Kontrolle und Verwaltung -Einfache Installation und Updates	-Schwache Toolunterstützung -Einfache Architektur
Client/Server	-Komfortable grafische Oberfläche -Benutzung komplexer -Direkte Verbindung zwischen Client und Server (RPC)	-Fette Clients, Sicherheit? -Verteilung der Anw. auf Client und Server -Probleme mit großer Anzahl Clients bei Installation und Update	-Sehr gute Tools -Architektur schon komplexer
Web	-Browser basiert -Gute Kombination der Stärken von C/S und Mainframe	-Thin Clients -Re-Zentralisierung -Management auf wenige Instanzen beschränkt	-Sehr gute Tools -Komplexe Architektur (Viele logische Komponenten) -Netzwerk

Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Architekturebenen



HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Demo!

HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Anwendungsserver für zentrale Dienste

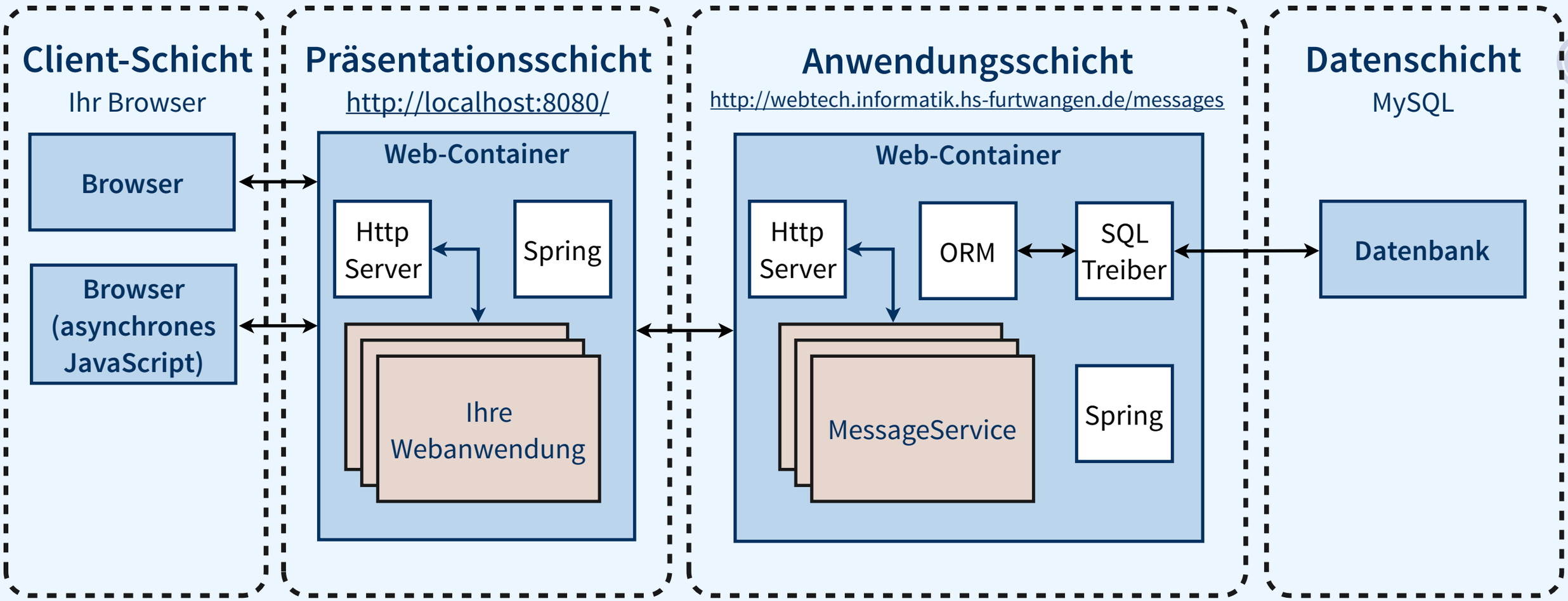
Application Server

Definition

- Integrieren verschiedene Server-Komponenten
  - Web-Server
  - Geschäftslogik
  - Integrationskomponente: Mail, Datenbank, Dateisystem
- Vorteile:
  - Sicherheit: Zugangsschutz an zentraler Stelle
  - Verwaltung: Updates an zentraler Stelle
  - Datenintegrität: Backups an zentraler Stelle
  - Performance: Load-balancing mit mehreren Application-Servern

HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Web Architektur



Application-Server

HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Web-Container

- Application-Server sind groß und komplex
  - Funktionalität sprengt den Rahmen dieser Vorlesung
- Wir beschränken uns auf den sogenannten Web-Container
  - Komponente im Application-Server für die Generierung und Auslieferung von HTML-Seiten
  - nicht zu verwechseln mit Web-Servern!
- In jedem Application-Server existiert diese Komponente, häufig wird einer der folgenden open-source Produkte verwendet:
  - Apache Tomcat
  - Eclipse Jetty

HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Web-Container

- Java definiert Standards, nach denen Web-Anwendungen entwickelt werden sollten:
  - Servlet API
  - JavaServer Pages
    - werden in den nächsten Vorlesungen vorgestellt
- Web-Anwendungen können auf allen Application-Servern/Web-Containern ausgeführt werden
- Unterschiede reduzieren sich auf Konfiguration und Wartung
- Wir werden exemplarisch Apache Tomcat verwenden

HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Anbieter

Web Container

Apache Tomcat

Eclipse Jetty

Application Server

Apache Geronimo

JBoss AS

Oracle GlassFish

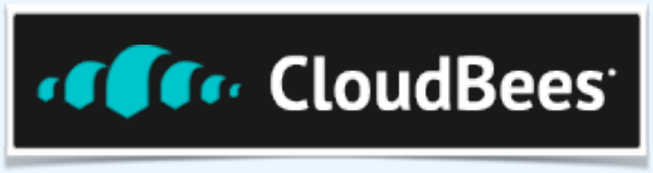
SAP Netweaver

IBM WebSphere

Oracle WebLogic

Cloud Services

(kostenloser Einstieg)



HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Bsp: Google App Engine

- Platform-as-a-Service (PaaS) für Java Web-Anwendungen
- kostenlos solange folgende Einschränkungen erfüllt sind:
  - weniger als 6,5 Stunden CPU Zeit pro Tag
  - weniger als 10 Anwendungen je Nutzer
  - maximal 100 Mails am Tag verschicken
  - maximal 1GB Datenfluss am Tag

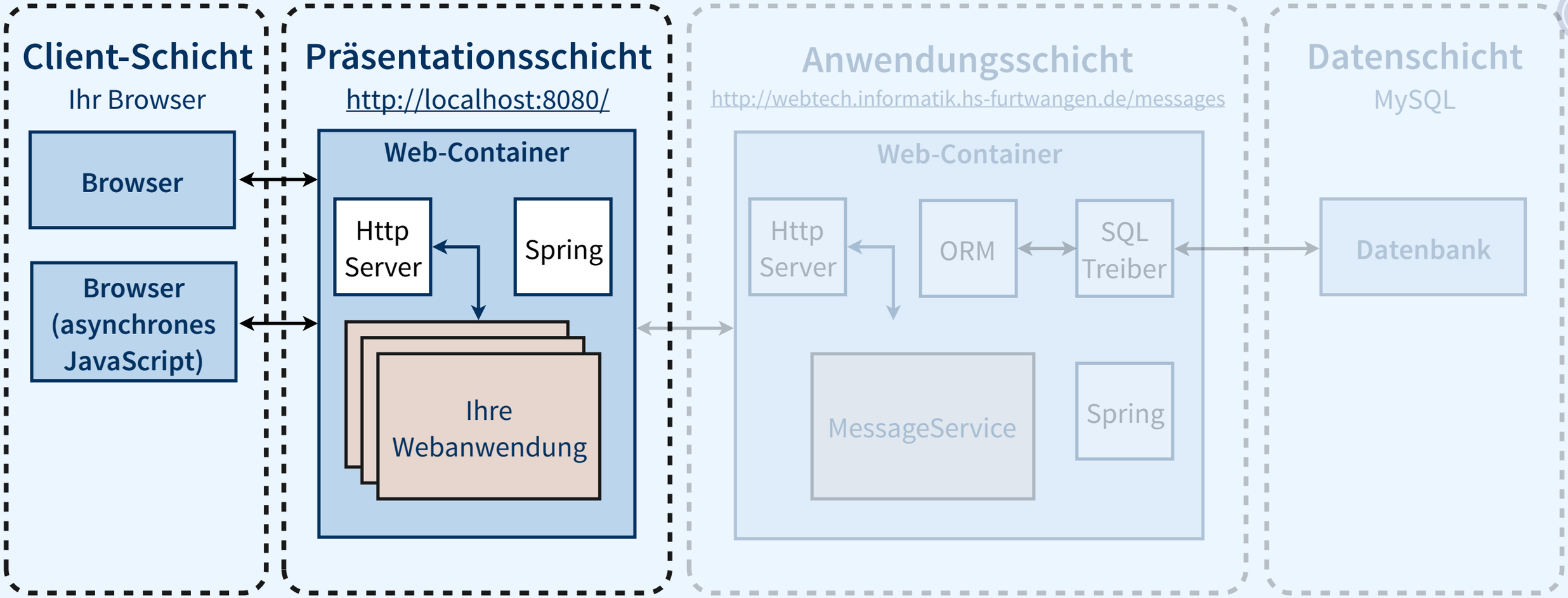
HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Web-Anwendungen

- Java Web-Anwendungen sind keine Java-Anwendungen
  - besitzen keine `main()`-Methode
  - können nicht einzeln gestartet werden
- Java Web-Anwendungen sind Pakete, die in einem Web-Container installiert werden (deployment)
- Die Struktur der Pakete ist standardisiert, das gleiche Paket kann in allen kompatiblen Web-Containern verwendet werden

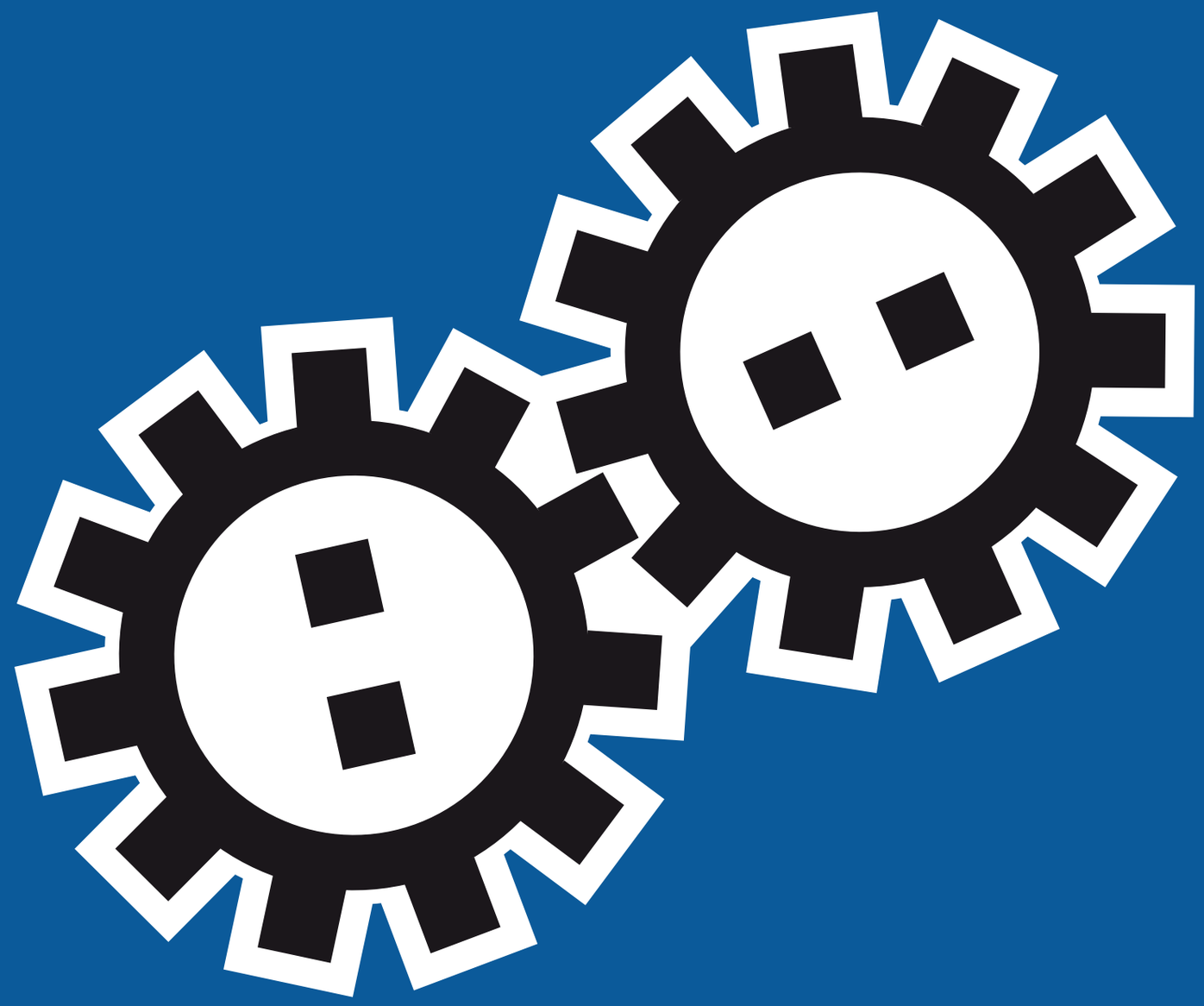
HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Web-Container



Wir betrachten nur den Web-Container!

Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen



Techniken

Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

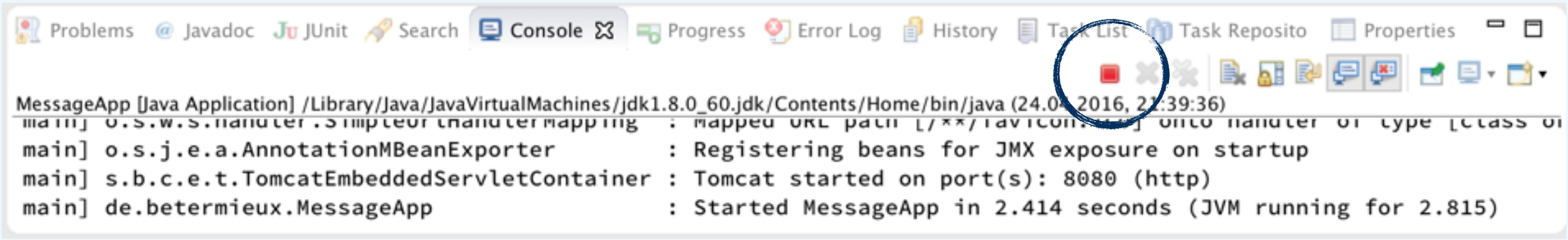
Java Web-Anwendungen

Web-Container aktivieren

- In der `pom.xml` die Abhängigkeit `spring-boot-starter-web` hinzufügen (Kommentarzeichen entfernen)

```
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
```

- Ab jetzt startet bei jedem Ausführen des Programms ein Tomcat-Server:
  - wartet auf HTTP-Anfragen auf Port 8080
  - beendet sich nicht automatisch, sondern muss manuell beendet werden (rotes Quadrat) →



HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

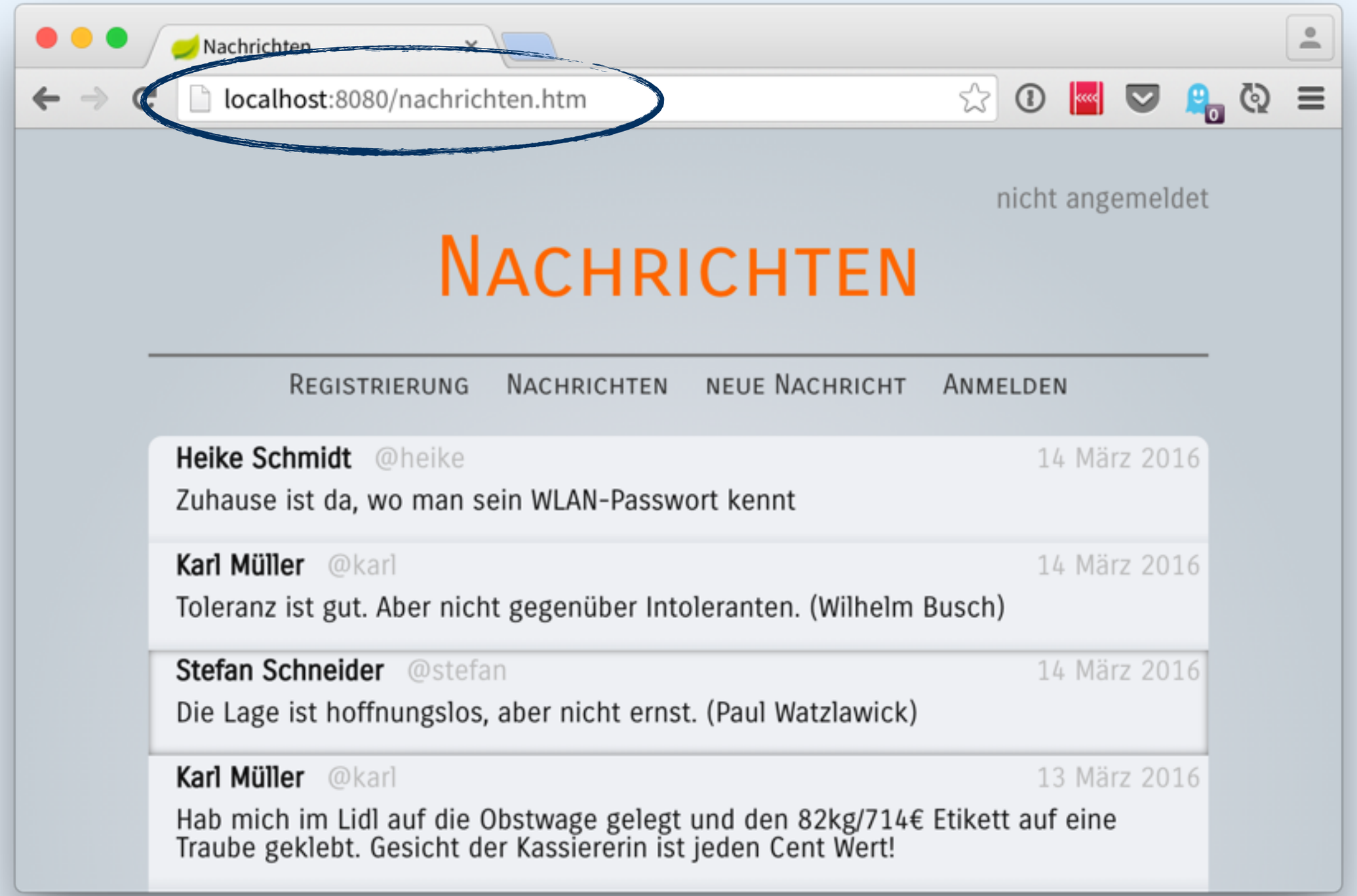
Statische Webseiten

```
projekt/
+-src
| +-main
| | +-java
| | | +-klasse.java
| | +-resources
| |   +-application.properties
| |   +-static
| |     +-nachrichten.htm
| |     +-registrierung.htm
| |     +-javascript/
| |     | +-geolocation.js
| |     +-css/
| |       +-gestaltung.css
| +-test
|   +-java
|     +-testklasse.java
+-target
| +-classes
| +-site
| +-...
+-pom.xml
```

- Der `static` Ordner innerhalb von `resources` enthält alle statischen Web-Dateien
  - HTML-Dateien
  - CSS-Dateien
  - JavaScript-Dateien
- Da Spring HTML-Seiten bereits dynamisch generiert, müssen wir die statischen Seiten testweise nach .htm umbenennen
- Statische Webseiten werden später nicht mehr benötigt

HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

http://localhost:8080/nachrichten.htm



HFU
Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Dynamische Webseiten - Servlets

- Servlets sind spezielle Java-Klassen, die
  - auf HTTP-Anfragen Antworten erzeugen
  - nicht alleine (nur in Web-Containern) ausgeführt werden können
  - Plattform- und Produktunabhängig erstellt werden
- Typische Anwendungsziele der Servlets sind:
  - Bearbeitung bzw. Speicherung von Daten, die mit Hilfe von HTML-Formularen eingegeben wurden
  - dynamische Content-Erstellung, z.B. mit Hilfe von Datenbankabfragen

HFU Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Servlet erstellen

HelloServlet.java

```java
@WebServlet("/hello.html")
public class HelloServlet extends HttpServlet {

  public void doGet(HttpServletRequest req, HttpServletResponse res)
         throws ServletException, IOException {
   res.setContentType("text/html");
   PrintWriter out = res.getWriter();
   out.append("<!DOCTYPE html><html><body><h1>Hello Browser!</h1></body></html>");
 }
}
```

Definiert die URL, um dieses Servlet aufzurufen

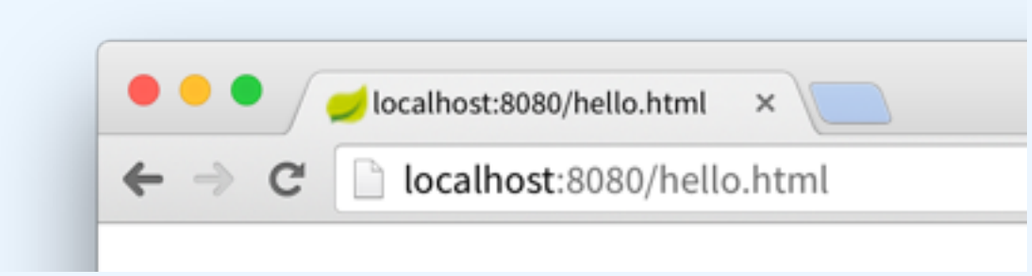
MessageApp.java

```java
@SpringBootApplication
@ServletComponentScan
public class MessageApp {

  public static void main(String[] args) {
    ApplicationContext ctx = SpringApplication.run(MessageApp.class, args);

    MessagePrinter messagePrinter = ctx.getBean(MessagePrinter.class);
    messagePrinter.outputMessages();
  }
}
```

Damit Spring nach den Servlets sucht

HFU Informatik | Hochschule Furtwangen
Prof. Dr. Stefan Betermieux | Fakultät

Verarbeitungsschritte



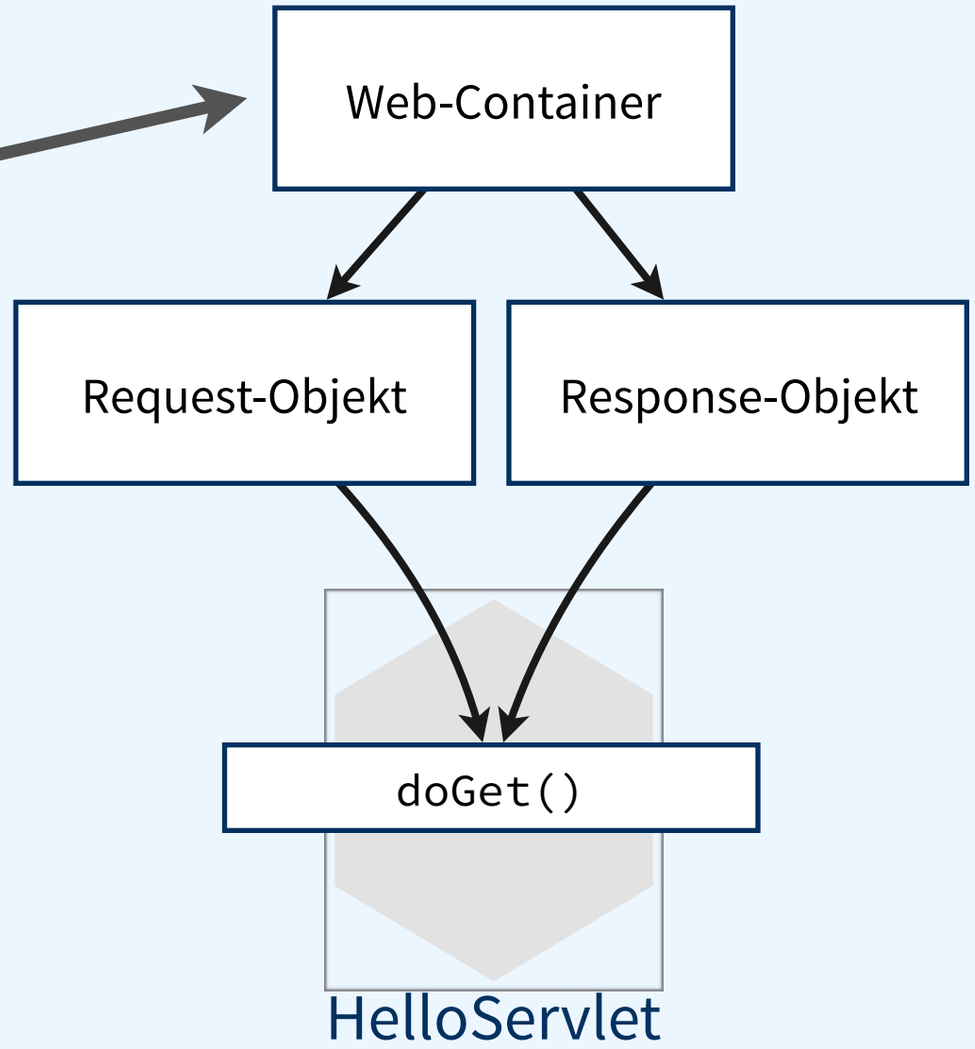

```
hello.html → HelloServlet
HTTP GET Anfrage → doGet()
```

Http-Request

Anfrage-Zeile
`GET hello.html HTTP/1.1`

Header-Zeilen
`Host: localhost:8080`
`User-Agent: Mozilla/5.0`

Body
`<leer>`



Prof. Dr. Stefan Betermieux | Fakultät Informatik | Hochschule Furtwangen

Danke