Web-Engineering

Web-Applikationen / Architekturvarianten

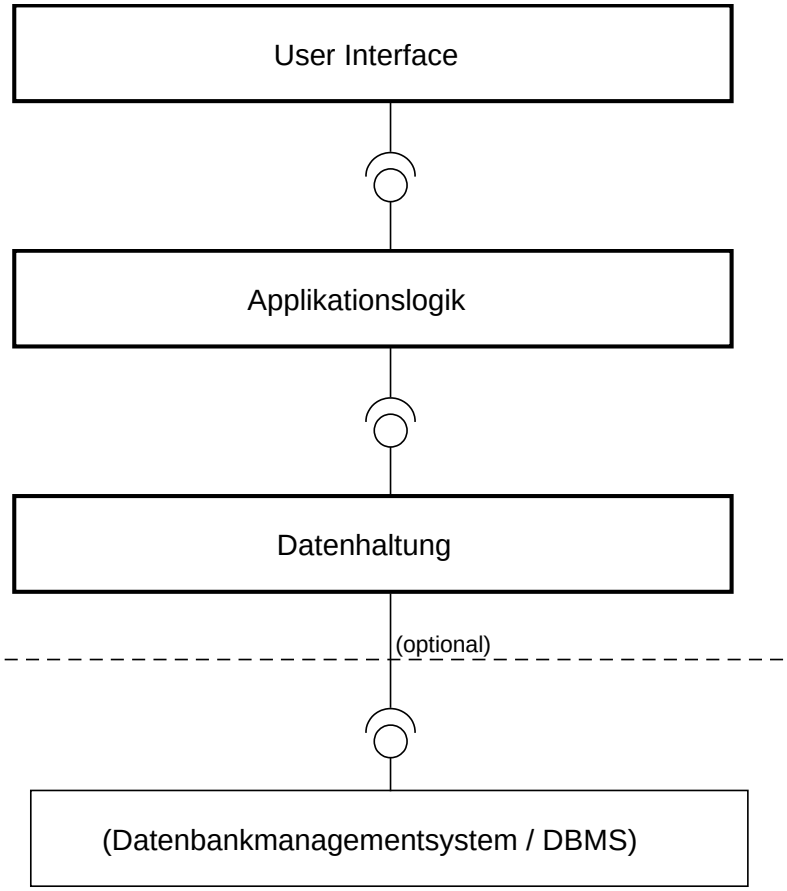
Hochschule Niederrhein
University of Applied Sciences
Elektrotechnik und Informatik
Faculty of Engineering and Computer Science

Web-Applikationen (1)

- User Interface
  - Präsentation
  - Interaktionen
- Applikationslogik
  - Fachliche Zusammenhänge
- Datenhaltung
  - Implementierung Datenmodell (Elemente, Beziehungen)
  - Persistenz (dauerhafte Speicherung)

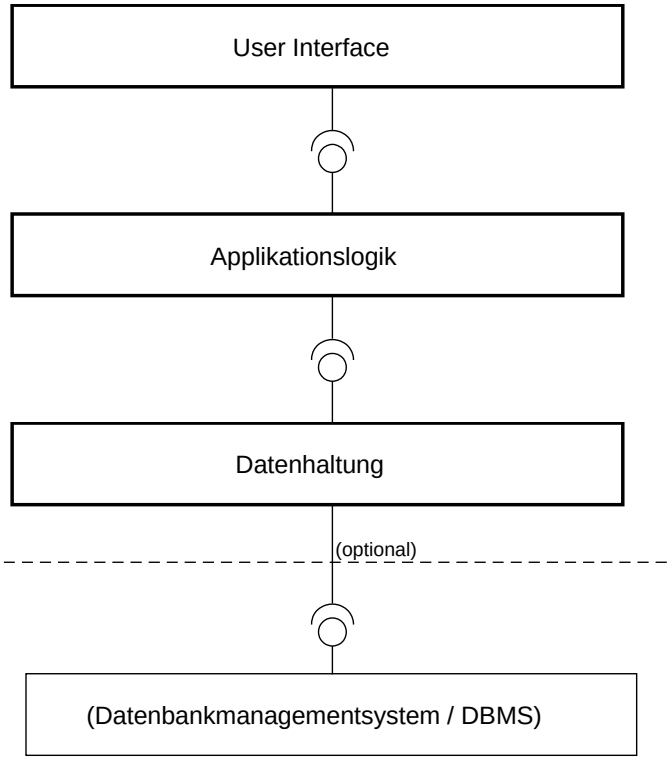
Web-Applikationen (2)

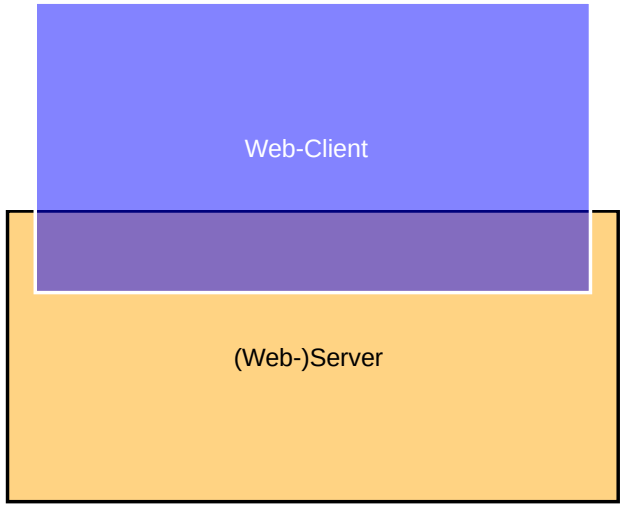
Schichtenmodell



Web-Applikationen (3)

Zuordnungsproblem
Schichtenmodell
<-->
Client-/Server-Aufteilung





Web-Applikationen: Verantwortlichkeiten

Aufteilung Verantwortlichkeiten → Varianten!

- Server:
  - Datenhaltung✓
  - Applikationslogik (in welchem Umfang?)
  - Auslieferung User Interface✓ (Applikationslogik ?)
- Client:
  - User Interface✓ (Applikationslogik ?)

Web-Applikationen: Aufteilungsvarianten (1)

- Variante 1 – 3: Schwerpunkt serverseitige Verarbeitung

	Client	Server
1	HTML5: Links, Formulare	Statische Ressourcen, direkte Aufbereitung
2	HTML5: Links, Formulare	Template Engine, Daten aufbereiten
3	HTML5: Links, Formulare Javascript (UI-Ereignisse)	Template Engine, Daten aufbereiten

Web-Applikationen: Aufteilungsvarianten (2)

- Variante 4: Verarbeitung aufgeteilt

	Client	Server
4	Single-Page, Hintergrund-Transfer HTML5, javascript (UI-Ereignisse, AJAX)	Template Engine, Daten aufbereiten

Web-Applikationen: Aufteilungsvarianten (3)

- Varianten 5 – 7: Schwerpunkt clientseitige Verarbeitung, Server ist (nur noch) Datenlieferant

	Client	Server
5	Single-Page, Hintergrund-Transfer (Daten!), Markup direkt erzeugen HTML5, javascript (UI-Ereignisse, AJAX)	Schnittstelle Daten verwalten (Erzeugen, Lesen, Ändern, Löschen)

Web-Applikationen: Aufteilungsvarianten (4)

- Varianten 5 – 7: Schwerpunkt clientseitige Verarbeitung, Server ist (nur noch) Datenlieferant

	Client	Server
6	Single-Page, Hintergrund-Transfer (Daten!), Template-Engine HTML5, javascript (UI-Ereignisse, AJAX)	Schnittstelle Daten verwalten (Erzeugen, Lesen, Ändern, Löschen)

Web-Applikationen: Aufteilungsvarianten (5)

- Varianten 5 – 7: Schwerpunkt clientseitige Verarbeitung, Server ist (nur noch) Datenlieferant

	Client	Server
7	Single-Page, Hintergrund-Transfer (Daten/Javascript), Template-Engine, HTML5, javascript (UI-Ereignisse, AJAX)	Schnittstelle Daten verwalten (Erzeugen, Lesen, Ändern, Löschen) Script(s) bereitstellen

Web-Applikationen: Strukturen bilden (1)

Serverseitig:

- Requests bearbeiten (→ Aufgaben verteilen)
- Daten verwalten (→ DB API)
  - Auch: „Geschäftsprozesse“ = Regeln beachten
- Responses erzeugen (→ Darstellung transformieren)

Clientseitig:

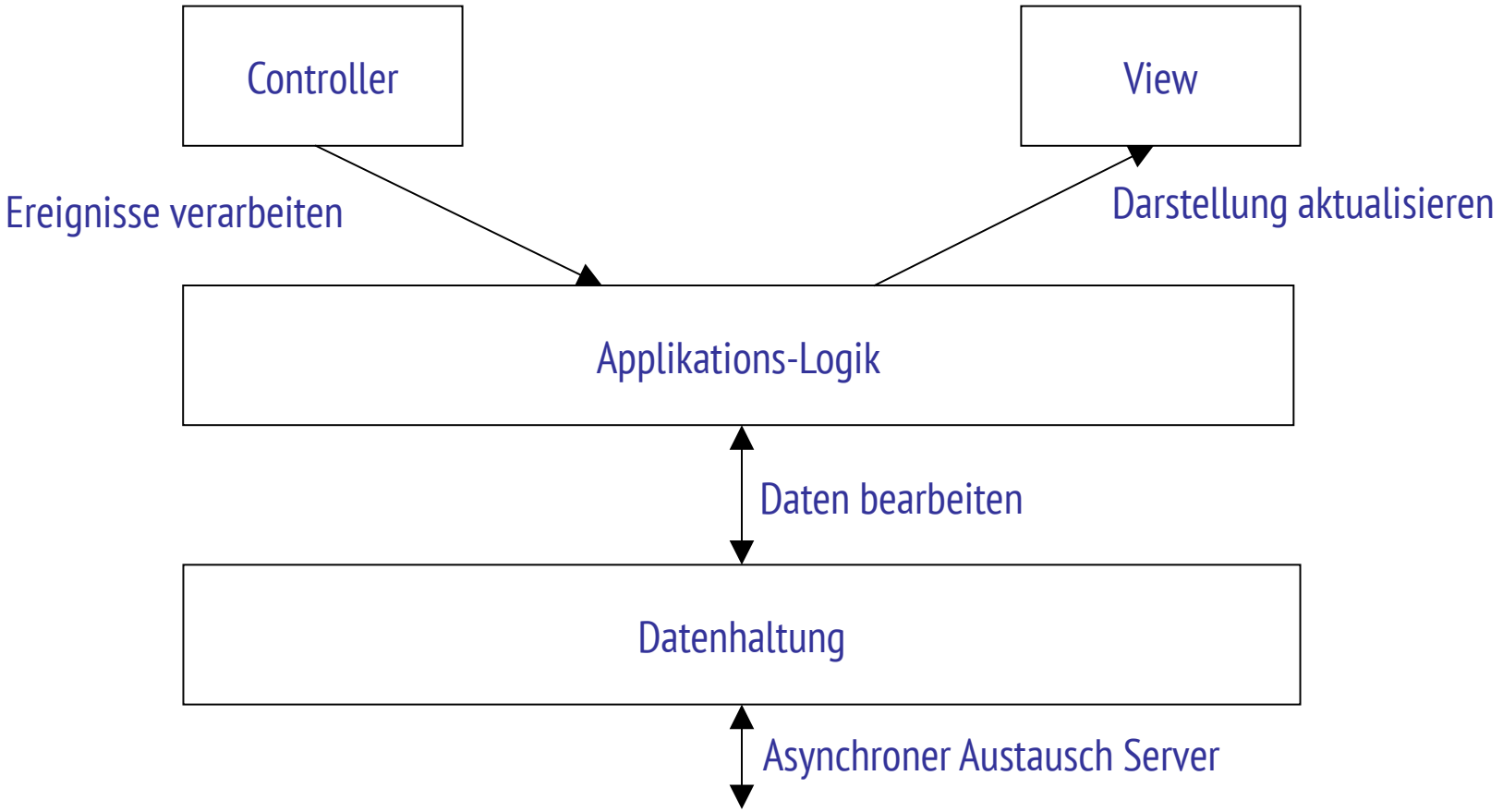
- Asynchrone Abläufe
  - UI-Ereignisse
  - Netzwerk-Ereignisse (HTTP-Response)

Web-Applikationen: Strukturen bilden (2)

Clientseitig:

- Eventservice: Ereignisse normieren
- Ideen MVC-Ansatz nutzen:
  - Controller:
    - (externe) Ereignisse entgegennehmen
    - verteilen
  - Model:
    - Daten und Regeln verwalten
  - View:
    - Sicht erzeugen

Web-Applikationen: Strukturen bilden (3)



Web-Applikationen: Strukturen bilden (4)

Realisierungsmöglichkeit:

- Eventservice verwenden
  - Basis: Entwurfsmuster „Publish-Subscribe“
  - In javascript: nur 1 Thread
  - „timeout“-Funktion: Ausführung nach Abschluss des aktuellen Ausführungskontextes

Web-Applikationen: Strukturen bilden (5)

Clientseitig:

- Einfache javascript-Funktionen reichen i.d.R. nicht mehr aus
- Erweiterbarkeit
  - Prototype-Konzept
  - Implementierung von „Klassen“ und Vererbung
- Verwendung von JS-Bibliotheken / Frameworks