



# **Multi-User-Applikationen objektorientiert realisieren**

Softwarearchitektur

25.11.24

# Handlungsnotwendige Kenntnisse

- 2.2 Kennt wichtige Architekturvarianten und -konzepte (Client/Server, Multi-Tier, Middleware, Framework, Klassenbibliothek).

# Ziele

- Du weißt, was man unter Softwarearchitektur versteht.
- Du kennst wichtige Architekturvarianten und -patterns.
- Du kennst die Vor- und Nachteile von verschiedenen Architekturen.
- Du weißt, worauf du beim Entwurf der Softwarearchitektur bei Multiuser-Applikationen achten musst.

# Softwarearchitektur

## Was ist das?

- Die Aufteilung von Zuständigkeiten von Software in verschiedene
  - Komponenten
  - Module
  - Schichten
- Exakt definierte Schnittstellen, z.B.
  - APIs als Schnittstelle zum Server
  - Sprachen (z.B. SQL als Schnittstelle zur Datenbank)

# Architekturbegriffe

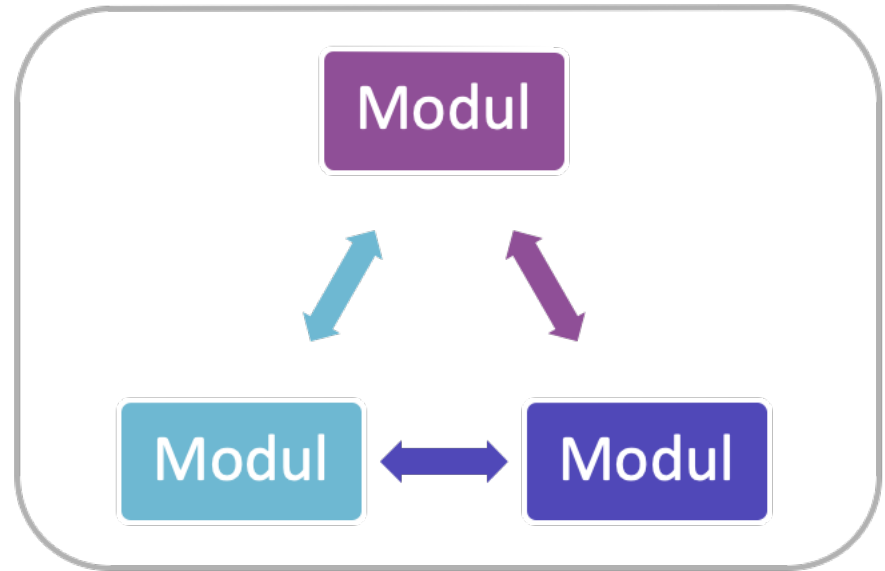
## Modul

- Definiert eine funktionale Einheit, z.B.
  - Userverwaltung
  - Zeiterfassung
  - Notifikationen

# Architekturbegriffe

## System / Applikation

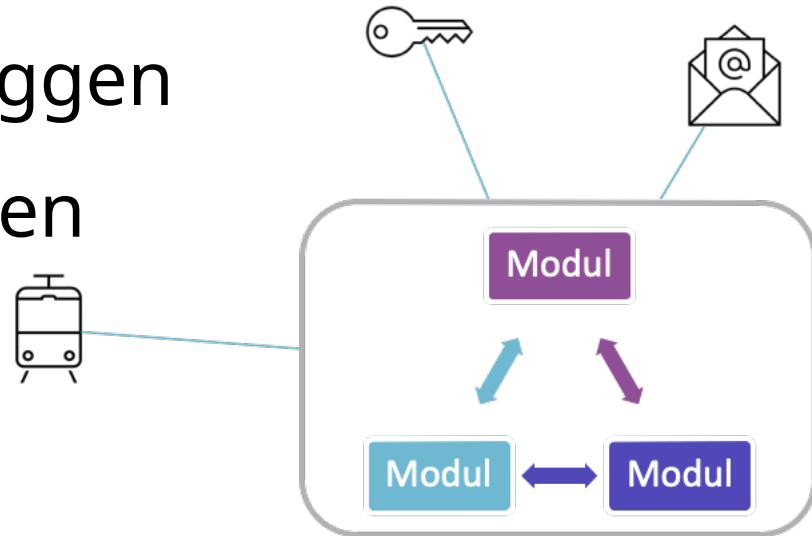
Mehrere kombinierte Module, welche zusammen einen Zweck erfüllen sollen.



# Architekturbegriffe

## Systemgrenzen

- Schnittstelle zu Umsystemen, z.B.
  - mit Mailserver eine E-Mail versenden
  - mit Google / Facebook einloggen
  - Fahrplan bei der SBB abfragen



# Architekturbegriffe

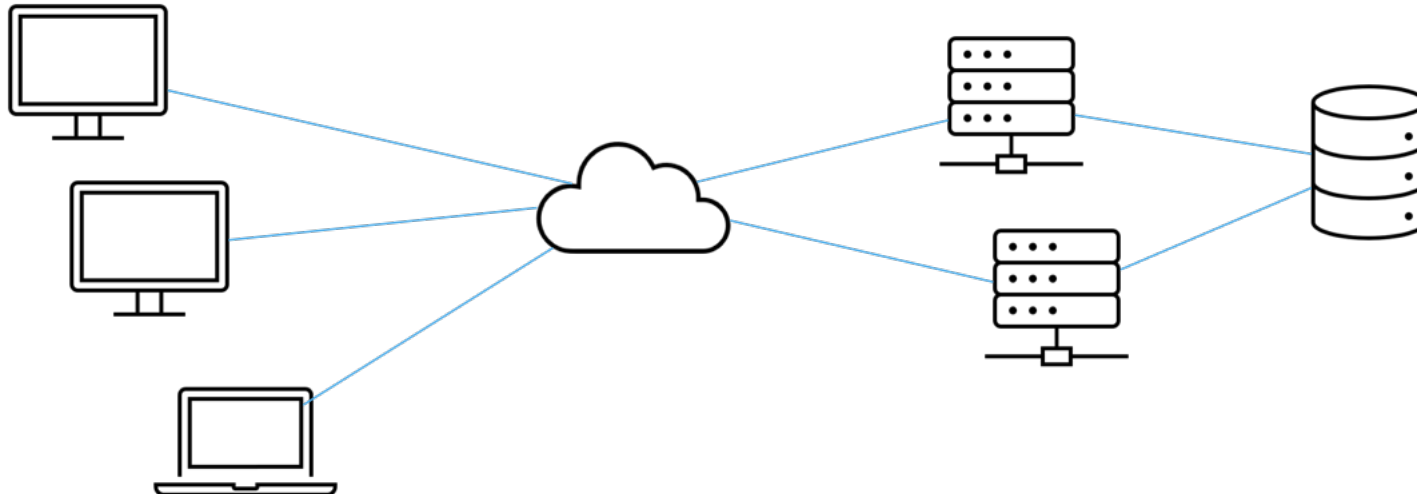
## Schichten

- Physikalische Schichten
- Logische Schichten



# Physikalische Schichten

- Mehrere, unterschiedliche physikalische Rechner...
- ...können an unterschiedlichen Standorten sein



# Logische Schichten

- Aufteilungen innerhalb von Software
- Zugriffe zwischen Schichten sind strikt geregelt
  - Unidirektional
  - Kein Überspringen



# Softwarearchitektur

## Wozu?

- Vorteile
  - Überschaubarkeit
  - Wiederverwendbarkeit
  - Skalierbarkeit
  - Wartbarkeit
  - Parallelisierbare Entwicklung
- Nachteile
  - Höhere Komplexität
  - Performance

# Architekturarten

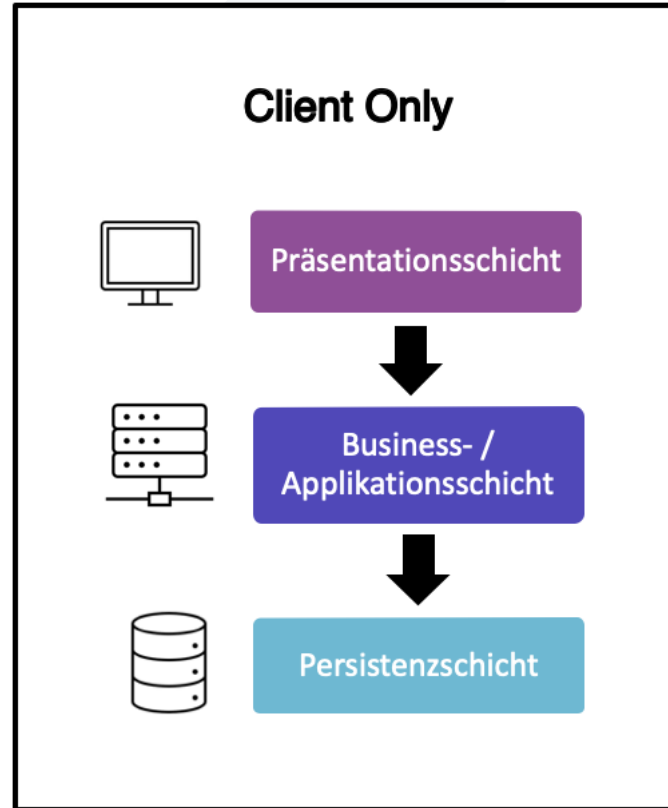
## 1-Tier, 2-Tier, 3-Tier, n-Tier

- Aus dem englischen tier:
  - Ebene
  - Reihe
  - Stufe
- 1-tier: Einstufig
- 2-tier: Zweistufig
- 3-tier: Dreistufig
- n-tier: n-stufig



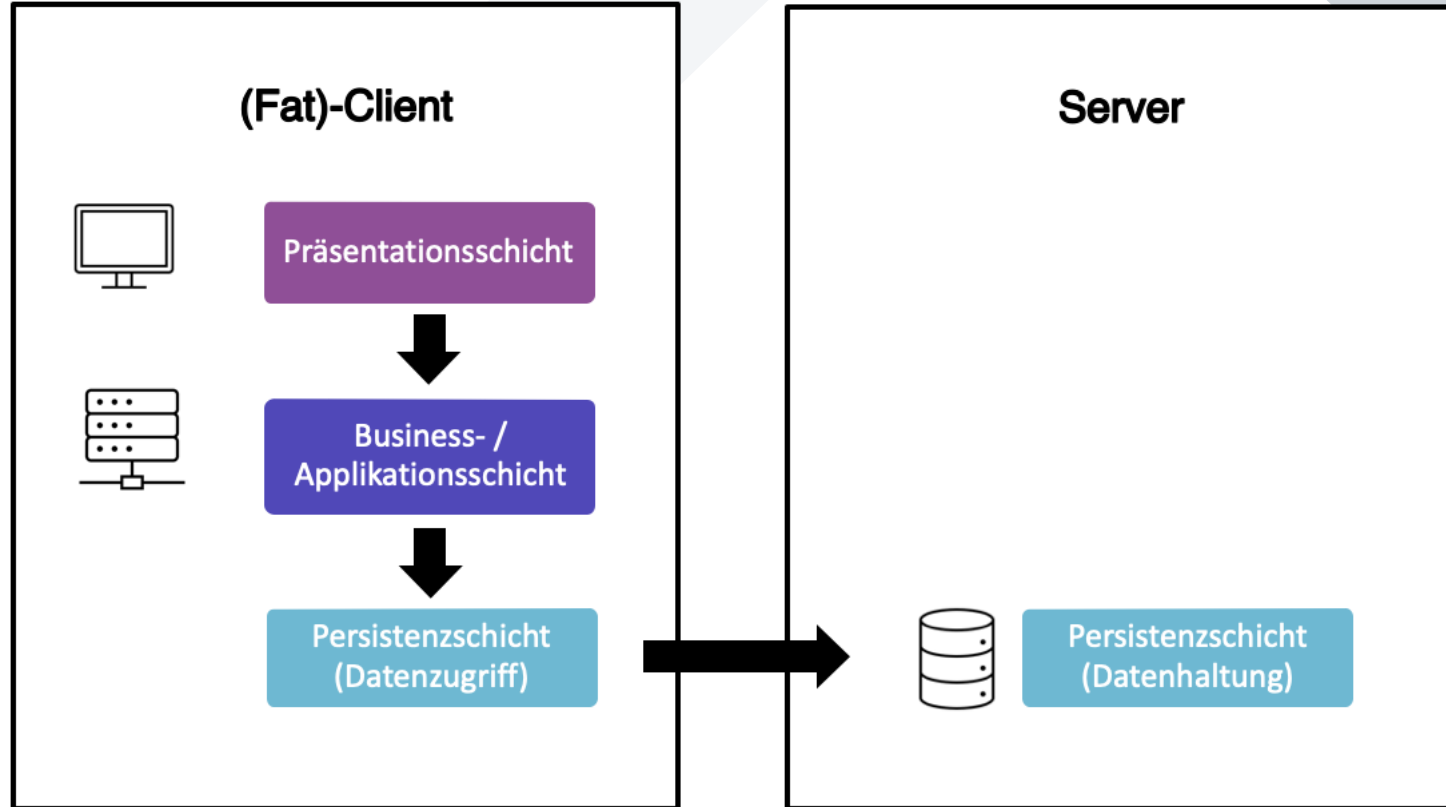
# Beispielarchitekturen

## 1-Tier



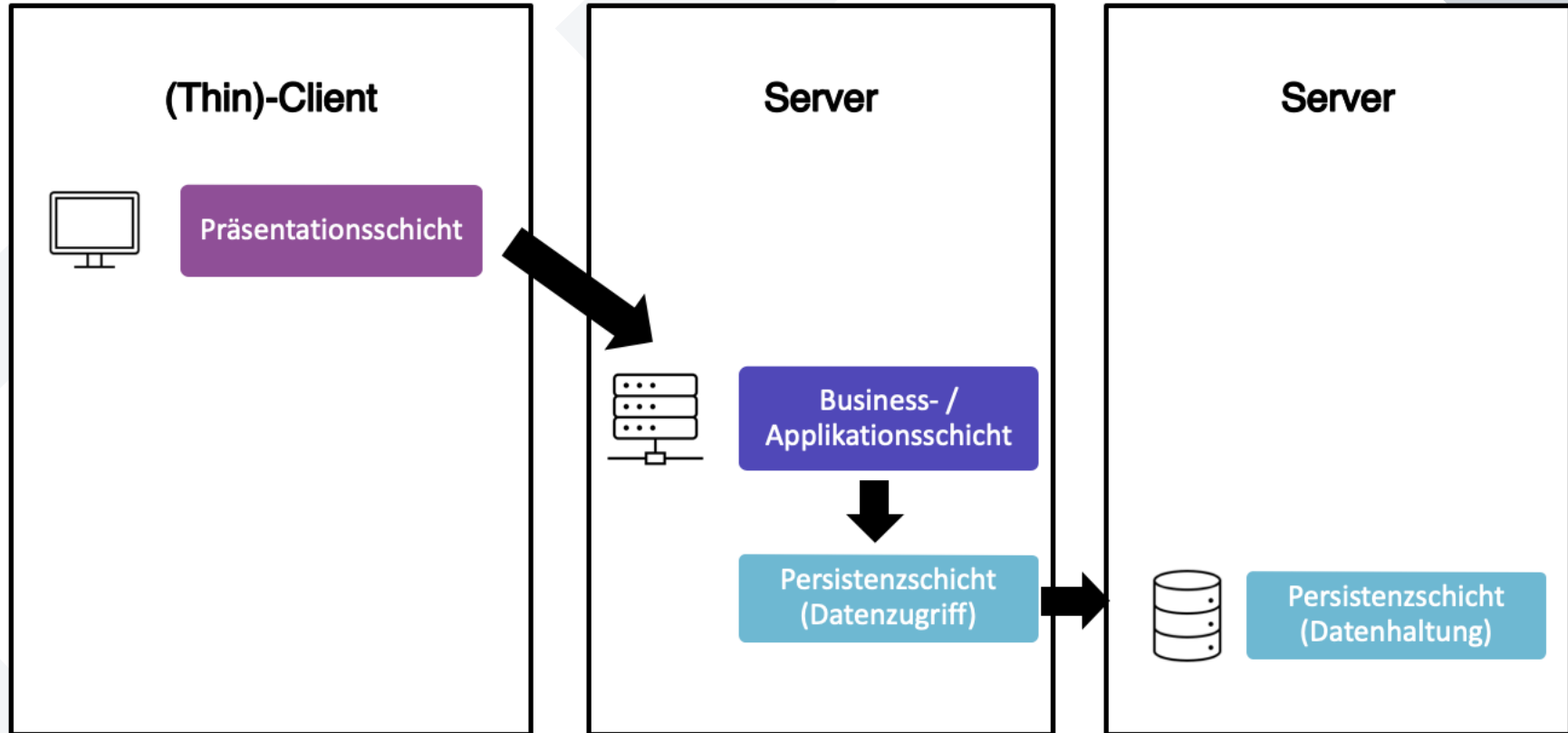
# Beispielarchitekturen

## 2-Tier



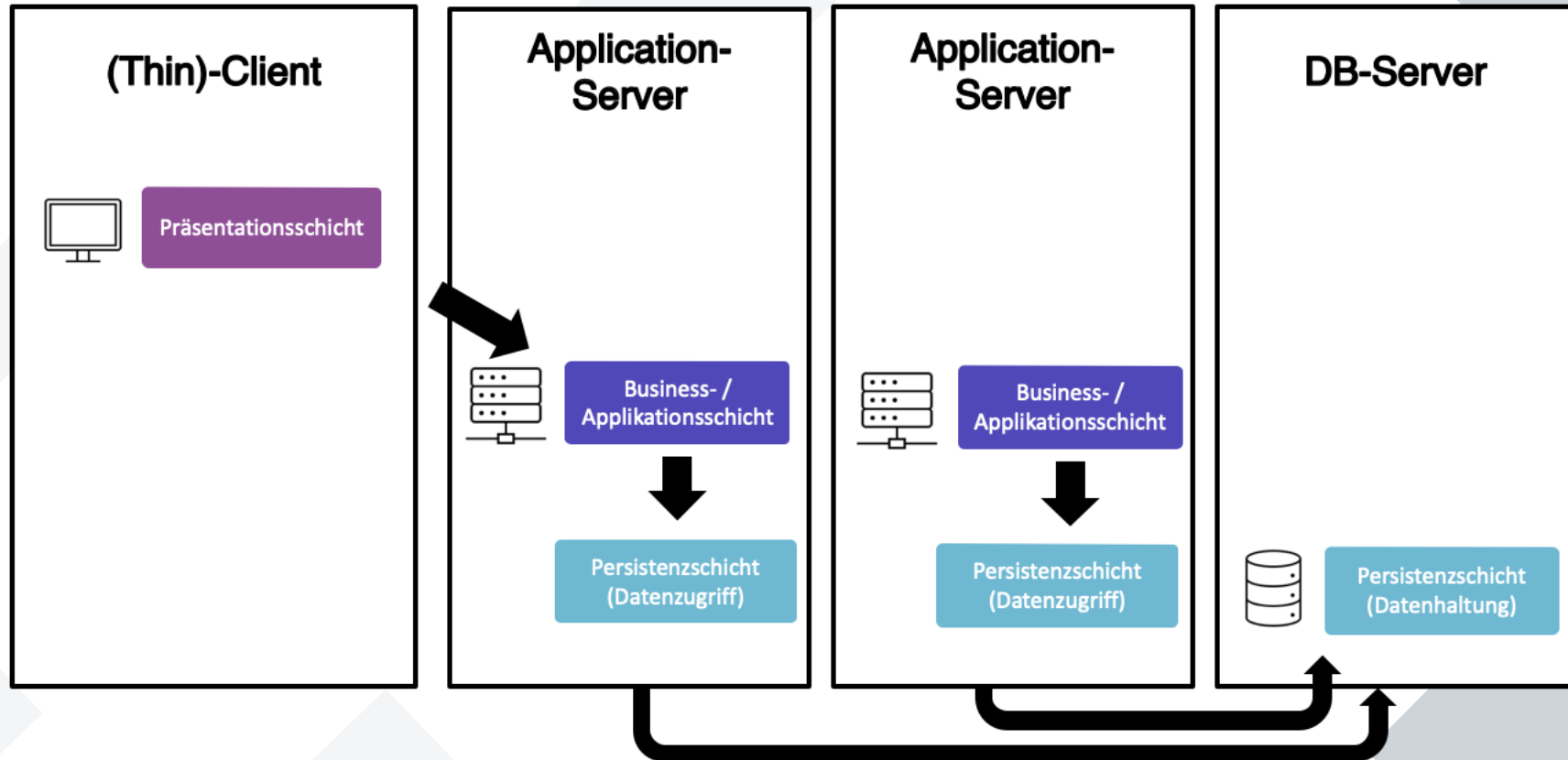
# Beispielarchitekturen

## 3-Tier



# Beispielarchitekturen

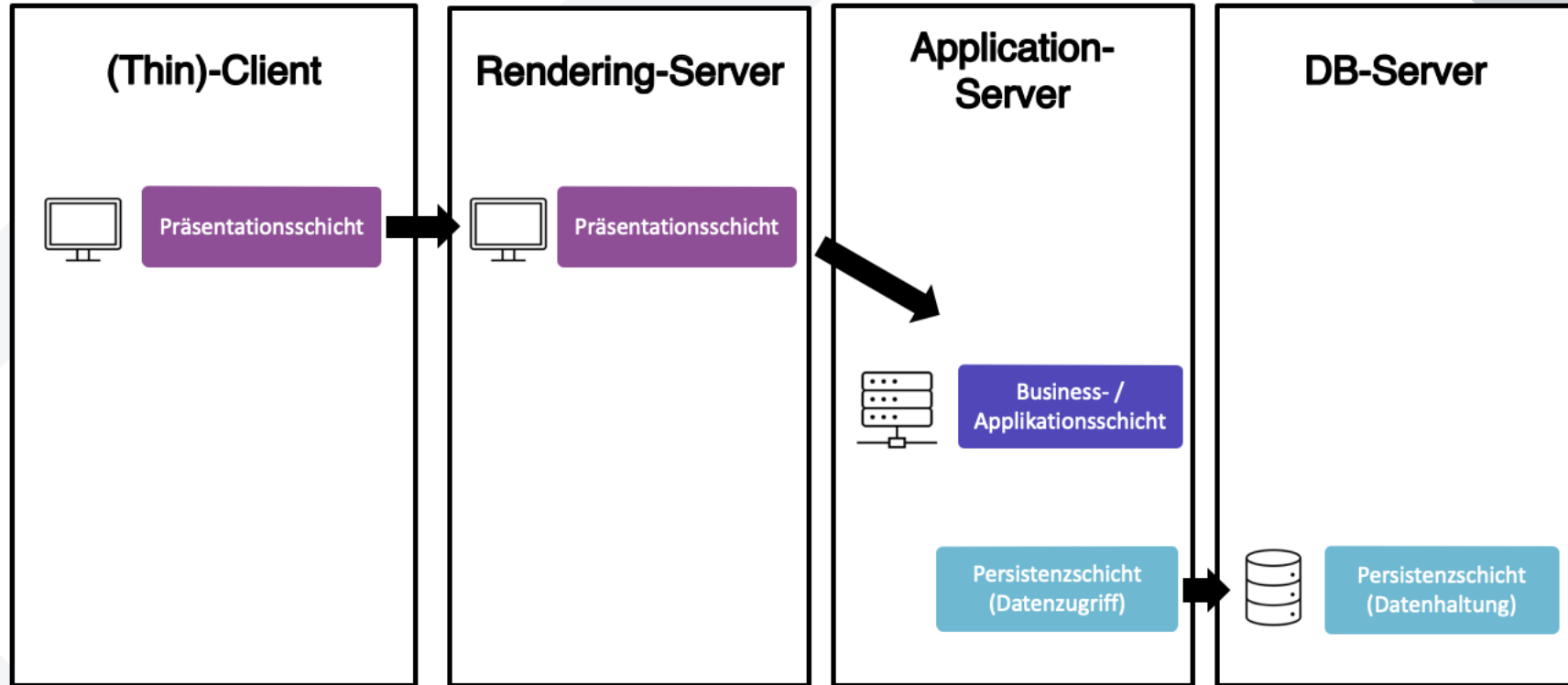
## n-Tier





# Beispielarchitekturen

## n-Tier



# Beispielarchitekturen

## n-Tier

- Beliebig viele weitere Schichten möglich
  - Caching
  - Messaging-Layers
  - Security



**Fragen?**