

Lernzettel

Architektur von Informationssystemen: Schichten, Komponenten und Schnittstellen

Universität: Technische Universität Berlin
Kurs/Modul: Informationssysteme und Datenanalyse
Erstellungsdatum: September 19, 2025



Zielorientierte Lerninhalte, kostenlos!
Entdecke zugeschnittene Materialien für deine Kurse:

<https://study.AllWeCanLearn.com>

Informationssysteme und Datenanalyse

Lernzettel: Architektur von Informationssystemen: Schichten, Komponenten und Schnittstellen

(1) **Ziel und Überblick.** Die Architektur von Informationssystemen beschreibt die Struktur aus Schichten, Komponenten und Schnittstellen, über die Daten durch das System fließen. Sie dient der Klarheit von Verantwortlichkeiten, der Skalierbarkeit und der Wiederverwendbarkeit von Bausteinen.

(2) **Schichtenmodell – Grundidee.** Ein typisches Schichtenmodell trennt Funktionen in sinnvolle Ebenen, z.B.:

- **Präsentationsschicht** (UI): Interaktion mit Nutzenden, Darstellung von Ergebnissen.
- **Logikschicht** (Application Layer): Geschäftslogik, Orchestrierung von Prozessen.
- **Datenhaltungsschicht** (Daten Layer): Persistenz, Zugriff auf Datenbanken.

(3) **Erweiterte Schichten und Integrationsschicht.** In realen Systemen kommt oft eine *Integrationsschicht* bzw. Middleware hinzu, die den Austausch zwischen Schichten steuert und lose Kopplung ermöglicht. Typische Aufgaben der Integrationsschicht:

- Vermittlung von API-Aufrufen, Authentifizierung, Protokoll-Umsetzung.
- Orchestrierung von Prozessen über Services hinweg.
- Transformation und Validierung von Datenformaten.

(4) **Typische Komponenten eines Informationssystems.** Wichtige Bausteine sind unter anderem:

- *Datenhaltungssysteme*: relationale DBMS, NoSQL-Datenbanken.
- *Transaktionssysteme*: OLTP-Funktionen, Konsistenz, ACID.
- *Analytische Systeme*: Data Warehouse, OLAP, Reporting-Komponenten.
- *Datenstrom-Management*: DSMS, Streaming-Analytics, Echtzeit-Verarbeitung.
- *Datenintegration*: ETL/ELT-Pipelines, Data-Governance, Qualitätssicherung.
- *BI- und Data-Science-Komponenten*: Klassifikation, Clustering, Visualisierung.

(5) **Schnittstellen – Arten und Zwecke.** Schnittstellen verbinden Bausteine innerhalb des Systems sowie das System mit der Außenwelt.

- *Interne Schnittstellen*: API-Views, Dienste, Events.
- *Externe Schnittstellen*: Frontend, Mobile Apps, Partner-Systeme.
- *Schnittstellentypen*: REST, GraphQL, gRPC, SOAP.

- *Messaging und Events*: Kafka, RabbitMQ, JMS, Event-Driven Architecture.
- *Datenformate und Protokolle*: JSON, Avro, Protobuf, XML.

(6) Architekturmuster – Beispiele. Gängige Muster, die sich in Informationssystemen finden:

- *Schichtenarchitektur*: klare Trennung von UI, Logik und Persistence.
- *Client-Server*: Clients greifen auf Server-Funktionen zu.
- *Microservices*: lose gekoppelte, spezialisierte Dienste, kommunizieren über APIs.
- *Event-driven Architecture*: Aktionen werden durch Events ausgelöst und verarbeitet.
- *Service-Oriented Architecture (SOA)*: funktionsbezogene Dienste über definierte Schnittstellen.

(7) Data-Warehouse vs. Transaktionssystem – Perspektiven. Wichtige Unterschiede in kurzen Stichpunkten:

- *Transaktionssysteme (OLTP)*: Transaktionen, konsistente Zustände, schnelle Schreibzugriffe.
- *Data-Warehouse (OLAP)*: analytische Abfragen, historical Data, Schwerpunkt auf Lesbarkeit und Reporting.
- *Datenfluss*: OLTP-Systemen werden Daten extrahiert, transformiert und in OLAP-Systeme geladen (ETL/ELT).
- *Zweck*: operatives Geschäft vs. ad-hoc Analysen und Entscheidungsunterstützung.

(8) Typischer Datenfluss in einem IS.

- Nutzer/Clients → Frontend (UI) → API/Service → Geschäftslogik → Datenhaltung
- Datenpersistenz in DBMS oder Data Warehouse, ggf. Streaming-Quellen (DSMS)
- Analytische Auswertung via BI/ML-Komponenten, Rückführung der Ergebnisse

(9) Kernbegriffe (Glossar).

- Schichtenarchitektur, lose Kopplung, Schnittstelle, API, Middleware
- OLTP, OLAP, ETL/ELT, Data Warehouse, Data Lake
- DSMS, Streaming-Analytics, Event-Driven Architecture, Microservices

(10) Zusammenfassung. Architektur von Informationssystemen deckt die Struktur aus Schichten, Komponenten und Schnittstellen ab. Ein klares Schichtenmodell unterstützt Skalierbarkeit und Wartbarkeit, während Komponenten wie DBMS, Transaktionssysteme, Data Warehouse und DSMS die verschiedenen Anforderungen an operatives Processing und Analytics abdecken. Schnittstellen ermöglichen lose Kopplung und flexible Integration.