

ARCHITEKTUREN

Kompetenznachweis

Team

Adriana Arteaga, Nuria Anaya
Andreas Nydegger, Kevin Schleuniger

Anaya Nuria
TBZ-Modul

Inhalt

Schichtenarchitektur	2
Vorteile	2
Nachteile	2
MVC	2
3-Tier Architektur	3
Presentation-Layer	3
Business-Layer	3
Data-Layer	3
Beispiel	3
4-Tier Architektur	4
Presentation-Layer	4
Business-Layer	4
Data Interface-Layer	4
Data-Layer	4
Beispiel	4
5-Tier Architektur	5
Presentation-Layer	5
Business-Layer	5
Data Interface-Layer	5
Data –Abstraction Layer	5
Data –Layer	5
Beispiel	5
Quellen	6

Schichtenarchitektur

Die Schichtenarchitektur ist ein häufig angewandtes Strukturierungsprinzip für die Architektur von Softwaresystemen.

Vorteile

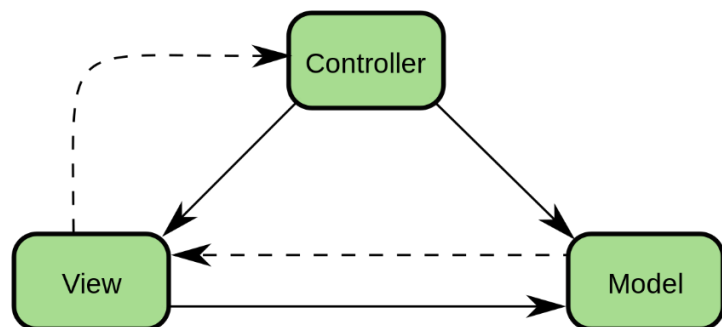
- Komplexität der Abhängigkeiten innerhalb des Systems wird reduziert
- Geringe Kopplung
- Einfacheres Verständnis der Applikation
- Einzelne Schichten gut austauschbar

Nachteile

- Ausführungsgeschwindigkeit durch Weiterleitung und Transformation reduziert

MVC

Auf den ersten Blick scheinen die drei Ebenen dem MVC-Konzept (Model View Controller) ähnlich zu sein. Topologisch sind sie jedoch unterschiedlich. Eine grundlegende Regel in einer dreischichtigen Architektur ist, dass die Client-Schicht niemals direkt mit der Datenschicht kommuniziert. In einem dreistufigen Modell muss die gesamte Kommunikation über die mittlere Ebene erfolgen. Konzeptionell ist die dreistufige Architektur linear. Die MVC-Architektur [model-view-controller] ist jedoch dreieckig: Die Ansicht sendet Aktualisierungen an den Controller, der Controller aktualisiert das Modell und die Ansicht wird direkt vom Modell aktualisiert.



3-Tier Architektur

Die 3-Tier-Architektur ist ein Modell, bei dem man das Programm in drei Teile teilt. In eine Grafik-Schicht, Logik-Schicht und Daten-Schicht. Diese baut von unten nach oben aufeinander auf und sollen in einer Richtung Abhängigkeiten vermeiden. Somit ist die höhere Schicht immer von der unteren Schicht abhängig aber nie anders herum. Somit ist die Grafik-Schicht von der Logik-Schicht abhängig und indirekt von der Datenschicht, die Logik-Schicht ist von der Datenschicht abhängig und die Datenschicht ist von niemanden abhängig.

Presentation-Layer

Die Grafik-Schicht ist für die grafische Benutzeroberfläche zuständig und dort werden alle Steuerelemente und grafische Komponenten platziert. Zudem werden auch die Event-Handler (Button-Click usw.) definiert, jedoch selber keine Berechnungen vorgenommen. Diese rufen lediglich die Funktionen aus dem Logik-Layer auf.

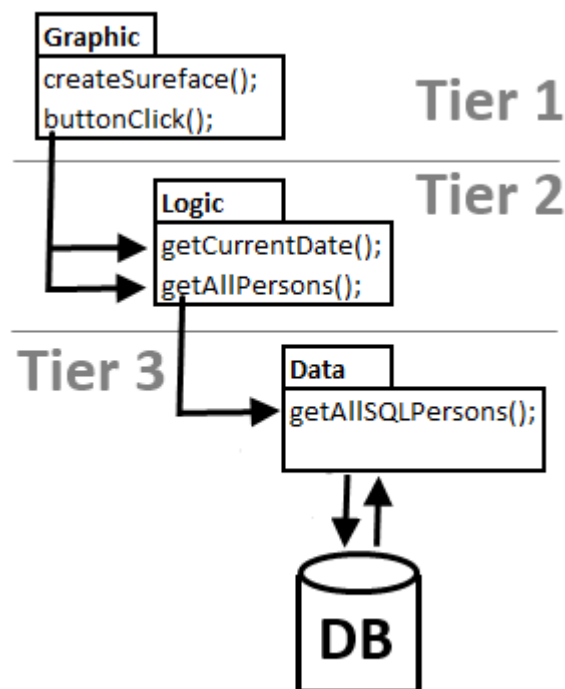
Business-Layer

In der Logik-Schicht werden sämtliche Methoden zu Berechnungen und umgesetzte Algorithmen implementiert. Jedoch werden keine Datenquellen direkt angesteuert. Braucht man Daten aus einer Datenbank, oder von Webservices usw. wird die Datenschicht angesteuert.

Data-Layer

In der Datenbankschicht werden alle Methoden verankert die Daten direkt aus einer Datenquelle lesen. Diese könnten SQL Datenbank-Abfragen beinhalten, oder Methoden welche APIs von Webservices ansteuern oder welche ganz simple Daten aus Dateien (Excel, Textdatei) auslesen.

Beispiel



4-Tier Architektur

Presentation-Layer

Der Präsentations-Layer ist für die grafische Benutzeroberfläche zuständig und dort werden alle Steuerelemente und grafische Komponenten platziert. Zudem werden auch die Event-Handler (Button-Click usw.) definiert, jedoch selber keine Berechnungen vorgenommen. Diese rufen lediglich die Funktionen aus dem Logik-Layer auf.

Business-Layer

In der Logik-Schicht werden sämtliche Methoden zu Berechnungen und umgesetzte Algorithmen und Code implementiert. Jedoch werden keine Datenquellen direkt angesteuert. Braucht man Daten aus einer Datenbank, oder von Webservices usw. wird die Datenschicht angesteuert.

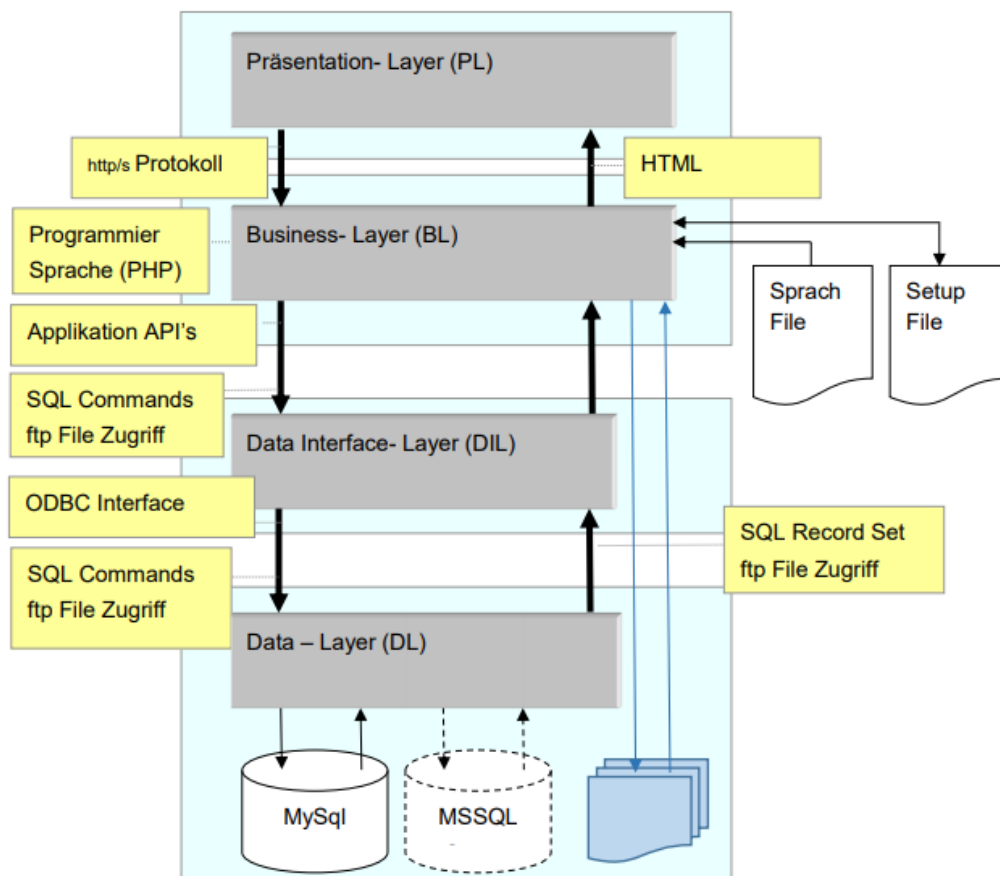
Data Interface-Layer

Der Data Interface-Layer entwickelt die Datenbank Zugriffe (API's) der unterschiedlichen Datenbank Systemen durch einen Standard Data Interface.

Data-Layer

Der Data-Layer wird auch Datenhaltungsschicht genannt. Diese Schicht enthält alle Datenbanken, Files oder sonstige Ablagen. Sie ist verantwortlich für das Lade, Speichern und Ändern der Applikationsdaten.

Beispiel



5-Tier Architektur

Presentation-Layer

Der Präsentations-Layer ist für die grafische Benutzeroberfläche zuständig und dort werden alle Steuerelemente und grafische Komponenten platziert. Zudem werden auch die Event-Handler (Button-Click usw.) definiert, jedoch selber keine Berechnungen vorgenommen. Diese rufen lediglich die Funktionen aus dem Logik-Layer auf.

man Daten aus einer Datenbank, oder von Webservices usw. wird die Datenschicht angesteuert.

Business-Layer

In der Logik-Schicht werden sämtliche Methoden zu Berechnungen und umgesetzte Algorithmen und Code implementiert. Jedoch werden keine Datenquellen direkt angesteuert. Braucht man Daten aus einer Datenbank, oder von Webservices usw. wird die Datenschicht angesteuert.

Data Interface-Layer

Der Data Interface-Layer entwickelt die Datenbank Zugriffe (API's) der unterschiedlichen Datenbank Systemen durch einen Standard Data Interface.

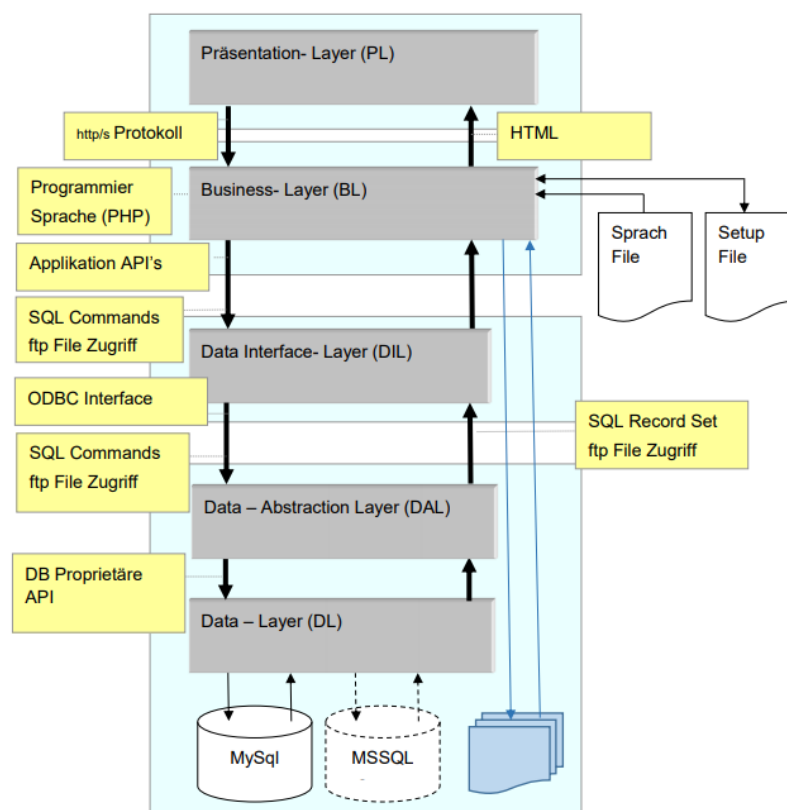
Data –Abstraction Layer

Ermöglicht relevante Zusammenhänge und Gegebenheiten zu verwalten. Es wird jedoch keine Business relevante Funktionen abgebildet.

Data –Layer

Die Datenhaltungsschicht enthält alle Datenbanken, Files oder sonstige Ablagen. Sie ist verantwortlich für das Lade, Speichern und Ändern der Applikationsdaten.

Beispiel



Quellen

https://www.researchgate.net/publication/280235378_A_Proposed_Collaborative_Architecture_for_Data_Visualization_System