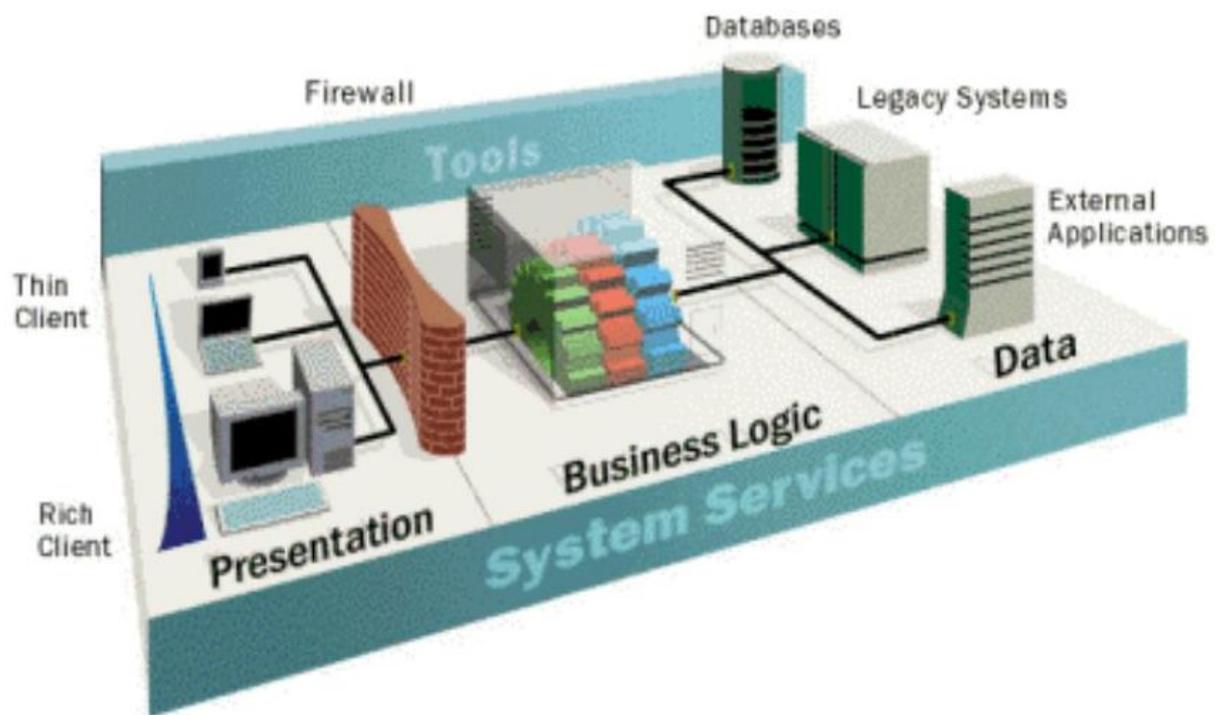




WEB Applikationen Schichten Modell



Einführung

Jedes Software-System hat eine Architektur, welche die aktuelle Organisation des Software-Systems darstellt, unabhängig davon, wie sorgfältig diese Organisation ausgewählt worden ist. Trotzdem kann man eine „gute“ Architektur intuitiv von einer „schlechten“ Architektur unterscheiden. Da dieses Spektrum zwischen einer guten und einer schlechten recht breit ist, präsentieren wir in diesem Abschnitt zuerst einige Qualitätsattribute von Software-Architekturen, bevor wir danach auf die einzelnen architekturellen Prinzipien und Bausteine näher eingehen.

Qualitätsattribute einer Software-Architektur

In der Literatur werden einige Qualitätsattribute zur Bewertung von architekturellen Entwürfen vorgeschlagen. Wir wollen hier jene von Bass, Clements und Kazman speziell herausgreifen, da diese Attribute breiten Anklang in der Beurteilung von Software-Architekturen gefunden haben. Auf der Ebene eines Software-Systems können zwei große Kategorien von Qualitätsattributen unterschieden werden, an denen ein Software-System gemessen werden kann:

- Beobachtbar durch Ausführung: Wie gut erfüllt ein Software-System seine Verhaltensanforderungen während der Ausführungszeit? Liefert es die geforderten Resultate in der vorgesehenen Zeit? Sind die Resultate korrekt oder innerhalb der vorgesehenen Genauigkeit? Funktioniert das System in Zusammenarbeit mit anderen weiteren Software-Systemen?
- Nicht beobachtbar durch Ausführung: Wie leicht ist es, das System zu integrieren, zu testen und zu modifizieren? Wie aufwendig (in Form von Zeit und Kosten) war es zu entwickeln? Wie lange war die Produkteinführungszeit (time to market)?

Information aus dem beobachtbaren Bereich liefert keinerlei Antworten zu Fragen aus dem nicht beobachtbaren Bereich. So können beispielsweise Software-Systeme, die ihre Laufzeit-Anforderungen erfüllen, unterschiedlich lange (und kostenintensiv) entwickelt worden sein. Andererseits müssen leicht modifizierbare Systeme nicht notwendigerweise ihre funktionalen Anforderungen erfüllen. Auch innerhalb dieser Kategorien können keine sicheren Schlüsse gezogen werden: Ein 70 4 Architekturelle Prinzipien, Bausteine und Muster stabiles Software-System muss nicht automatisch auch korrekte Ergebnisse liefern. Genauso können Systeme, die in wenigen Wochen entwickelt worden sind, Monate für bestimmte Modifikationen erfordern.

Qualität muss daher in allen Phasen von der Analyse, über das Design, die Implementierung bis hin zur Inbetriebnahme (deployment) berücksichtigt und mittels verschiedener Qualitätsmaße ausgedrückt und überprüft werden. Nicht jedes Qualitätskriterium hat einen Einfluss auf die Architektur des Software-Systems: Benutzerfreundlichkeit ist mehr ein Kriterium für den Anwender der Software und nicht der Architektur selbst. Die Ausprägung einer Benutzerschnittstelle und deren Realisierung mittels diverser Software-Komponenten oder Programmbibliotheken hat andererseits wiederum sehr wohl Einfluss auf die Modifizier- und Änderbarkeit eines Software-Systems.

Modifizierbarkeit ist ein an sich stark architekturelles Attribut. Ein System ist modifizierbar, wenn Änderungen nicht viele und nicht große Teile des Systems betreffen. Performanz hat demgegenüber sowohl architekturelle als auch nicht architekturelle Bedeutung: Einerseits hängt diese zum Teil von der erforderlichen Kommunikation zwischen Komponenten und der den Komponenten jeweils

zugeordneten Funktionalitäten ab (architekturell); andererseits ist die Wahl und die Kodierung des Algorithmus für die Performanz wesentlich (nicht-architekturell). Bass Clements und Kazmann fassen zusammen, was hinsichtlich Architektur und Qualitätsattributen wesentlich ist:

- Architektur ist kritisch für die Realisierung von vielen Qualitätsattributen eines Software-Systems. Diese Qualitätsattribute sollten auch auf der Ebene der Architektur evaluiert werden.
- Manche Qualitätsattribute sind nicht architekturabhängig und ein Erreichen dieser durch architekturelle Mittel ist keineswegs zielführend.

Jede Diskussion von Qualitätsattributen bedingt darüber hinaus, dass kein Qualitätsmerkmal alleine und unabhängig von anderen betrachtet und realisiert werden kann. Qualitätsattribute sind nicht orthogonal und können nur im gegenseitigen Wechselspiel unter Berücksichtigung von sinnvollen und erreichbaren Ausprägungen anderer Charakteristiken erreicht werden.

Zu den zur Laufzeit beobachtbaren Qualitätsattributen zählt man u.a. Performanz, Sicherheit, Verfügbarkeit, Funktionalität oder Benutzerfreundlichkeit. Nicht zur Laufzeit beobachtbar sind vielmehr Modifizier- und Änderbarkeit, Portabilität, Wiederverwendbarkeit, Integrierbarkeit oder Testbarkeit. Darüber hinaus gibt es noch so genannte Geschäfts-Qualitätsattribute, die ein Software-System prägen: Produkteinführungszeit, Kosten, projektierte Einsatzzeit des fertigen Systems, Marktausrichtung, Markteinführungsplan oder auch Integration mit anderen, existierenden Systemen.

4- Tier Architektur für WEB Applikationen

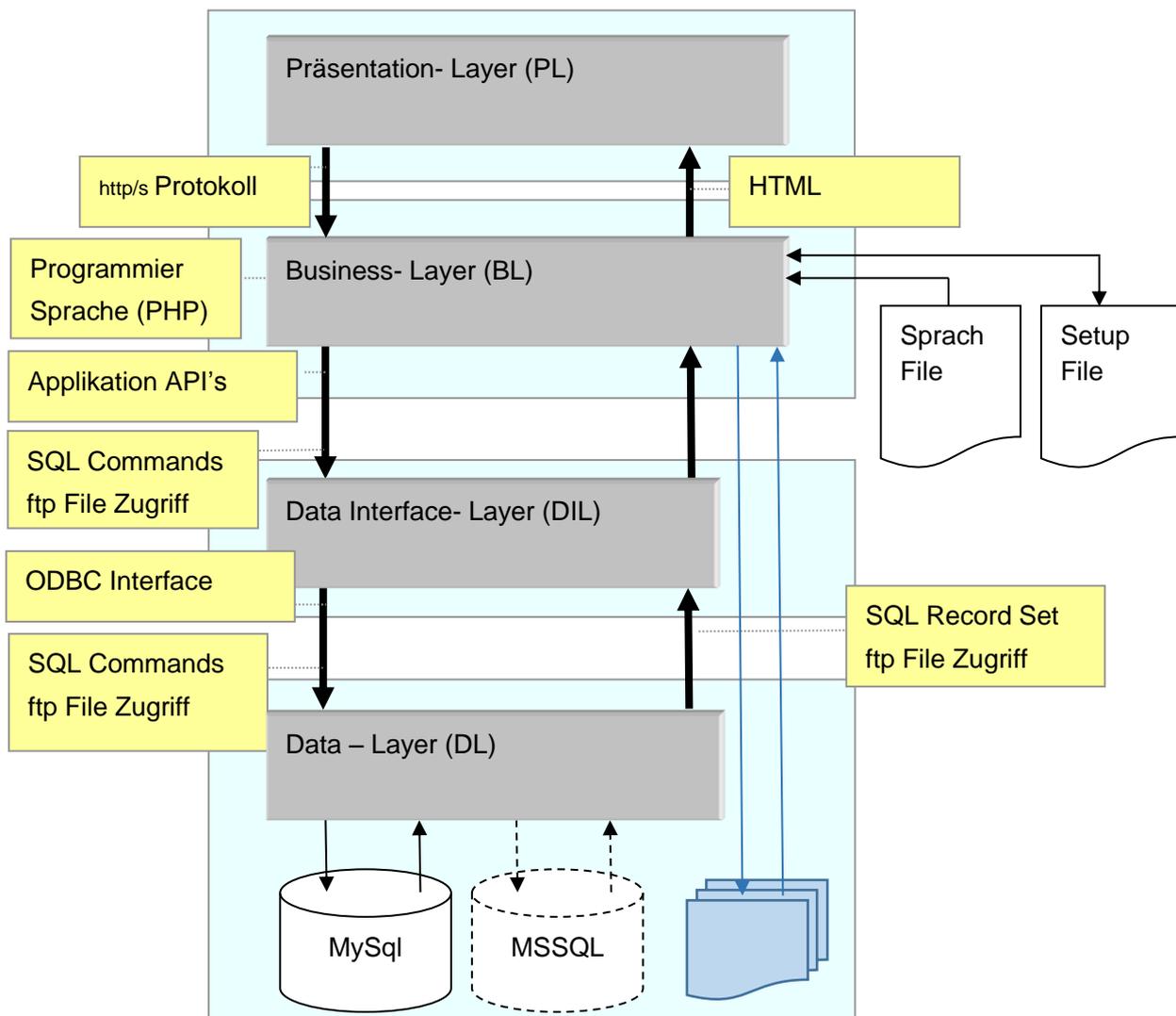


Fig1.

5- Tier Architektur für WEB Applikationen

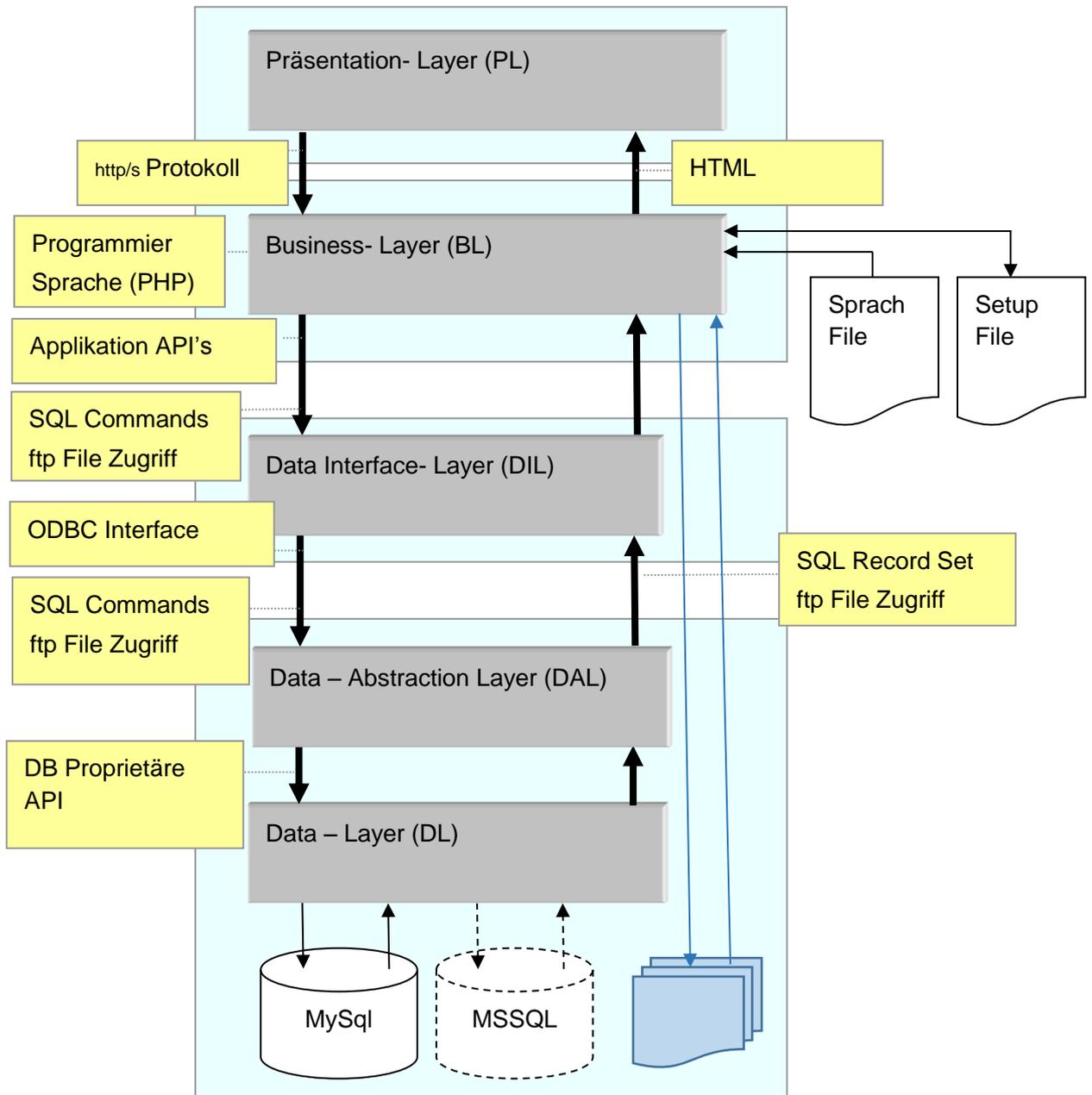


Fig2.

Applikations- Layer Beschreibung

Präsentation Layer

Dieser Layer ist für die Repräsentation der Daten, Benutzereingaben, Benutzerführung und Benutzerschnittstelle verantwortlich. In diesem Layer werden keine Applikations Funktionen implementiert.

Business Layer

Wird auch als Logikschicht, Middle Tier oder Businessschicht bezeichnet. Sie beinhaltet alle Verarbeitungsmechanismen und Applikations- Abläufe. Im Business Layer wird der Programm Code implementiert.

Data Interface Layer

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

Data Abstraktions- Layer

Ermöglicht sequenzielle Data Layer relevante Zusammenhänge und Gegebenheiten zu verwalten. Es werden keine Business relevante Funktionen abgebildet.

Data Layer

Auch als Datenhaltungsschicht bezeichnet. Sie enthält die Datenbanken und File Ablage einer Applikation. Sie ist verantwortlich für das Laden, Speichern und Ändern von Applikationsdaten.

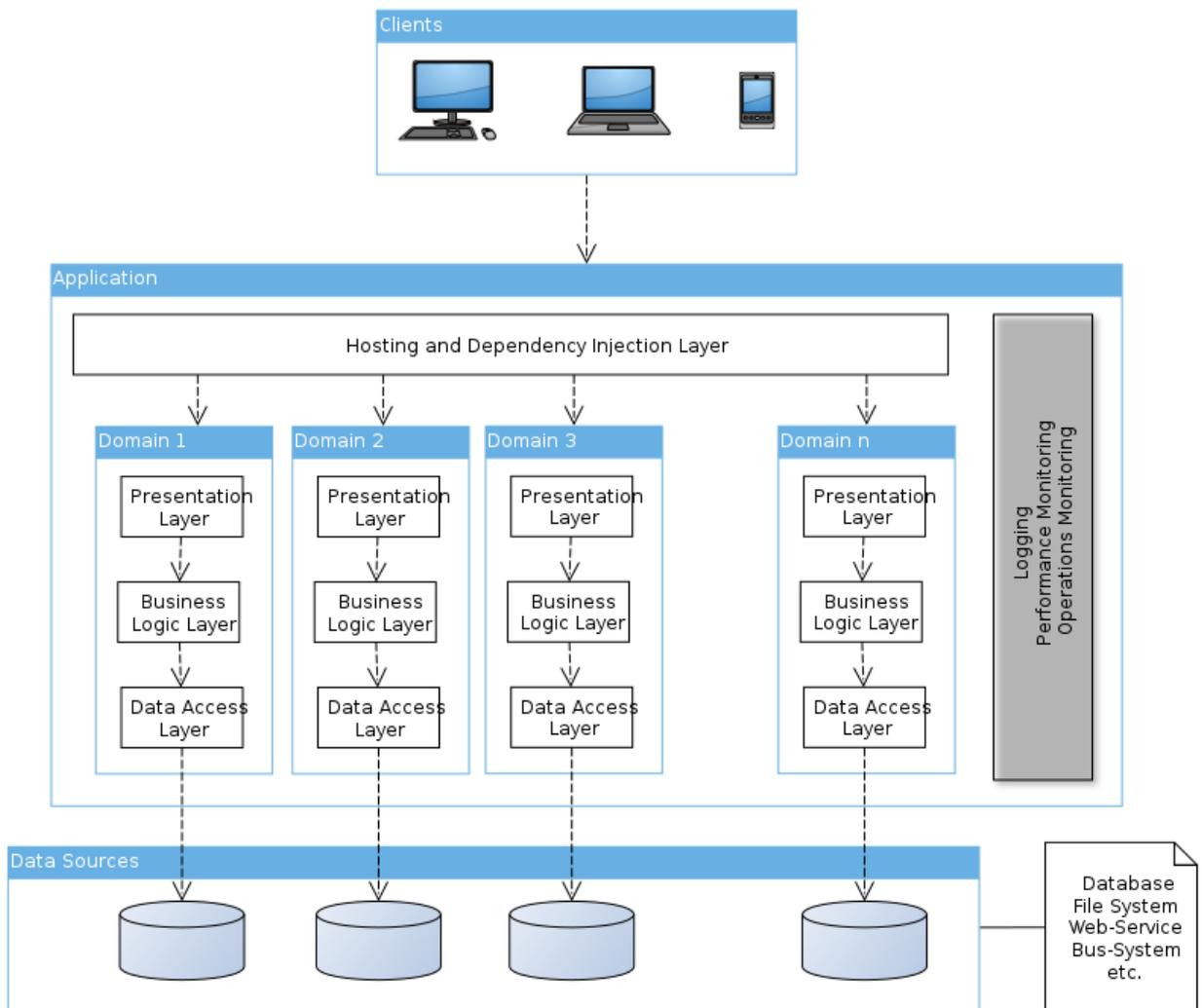


Fig 3.