



Diplomanden- und Doktorandenseminar
des Instituts für Informatik

Architecture Styles

M.Sc. Seraj Fayyad, Hochschule Harz, Wernigerode

Software Architecture:



Software Architecture is defined to be the rules, heuristics and patterns governing:

- Partitioning the problem and the system to be built into discrete pieces.
- Techniques used to create interfaces between these pieces.
- Techniques used to manage overall structure and flow.
- Techniques used to interface the system to its environment.
- Appropriate use of development and delivery approaches, techniques and tools.

Architecture Benefits:

- System qualities addressing.
- Consensus driving.
- Planning process supporting.
- Architectural integrity driving.
- Helping in complexity management.
- basis for reuse Providing.
- maintenance costs reducing.
- Impact analysis supporting.

Architectural Styles:

An architectural style, sometimes called an architectural pattern, is a set of principles—a coarse grained pattern that provides an abstract framework for a family of systems. An architectural style improves partitioning and promotes design reuse by providing solutions to frequently recurring problem.



Architectural Style Categories:

Category	Architecture styles
<i>Communication</i>	Service-Oriented Architecture (SOA), Message Bus
<i>Deployment</i>	Client/Server, N-Tier, 3-Tier
<i>Domain</i>	Domain Driven Design
<i>Structure</i>	Component-Based, Object-Oriented, Layered Architecture

Architectural Styles Types:

Architecture style	Description
<i>Client/Server</i>	Segregates the system into two applications, where the client makes requests to the server. In many cases, the server is a database with application logic represented as stored procedures.
<i>Component-Based Architecture</i>	Decomposes application design into reusable functional or logical components that expose well-defined communication interfaces.
<i>Domain Driven Design</i>	An object-oriented architectural style focused on modeling a business domain and defining business objects based on entities within the business domain.
<i>Layered Architecture</i>	Partitions the concerns of the application into stacked groups (layers).
<i>Message Bus</i>	An architecture style that prescribes use of a software system that can receive and send messages using one or more communication channels, so that applications can interact without needing to know specific details about each other.
<i>N-Tier / 3-Tier</i>	Segregates functionality into separate segments in much the same way as the layered style, but with each segment being a tier located on a physically separate computer.
<i>Object-Oriented</i>	A design paradigm based on division of responsibilities for an application or system into individual reusable and self-sufficient objects, each containing the data and the behavior relevant to the object.
<i>Service-Oriented Architecture (SOA)</i>	Refers to applications that expose and consume functionality as a service using contracts and messages.

Dienstag, den 12.04.2011

14:30 Uhr in Raum 203, Ifl, Hörsaalgebäude,
Albrecht-von-Groddeck-Straße 7