Architecture Patterns / 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 problems. You can think of architecture styles and patterns as sets of principles that shape an application.

Some of the most common architecture styles are:

Client/Server: 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.

Component-Based Architecture: Decomposes application design into reusable functional or logical components that expose well-defined communication interfaces.

Domain Driven Design: An object-oriented architectural style focused on modeling a business domain and defining business objects based on entities within the business domain.

Layered Architecture Partitions the concerns of the application into stacked groups (layers).

Message Bus: 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.

N-Tier / 3-Tier: 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.

Object-Oriented 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.

Service-Oriented Architecture: Refers to applications that expose and consume functionality as a service using contracts and messages.

Distributed Architecture: Refers to several Applications that execute in different computers and that they communicate through a network.

Event-driven architecture: Refers to applications promoting the production, detection, consumption of, and reaction to events.

Batch/Sequential: refers to execution of a series of programs or "jobs" on a computer without manual intervention and usually in a sequential and predefined order.

Repository: refers to a communication pattern for software compnents based on share a common repository and to establish change notifications.

Blackboard: a common knowledge base, known as the "blackboard", is iteratively updated by a diverse group of components in order to find cooperatively a solution of a problem.

Rule Based System: refers to a system build in terms of data and rules, and execution instructions are derived from such data and rules

Model-View-Controller: is a computer software architectural pattern that separates the representation and management of information from the user's interaction with it.

The architecture of a software system is almost never limited to a single architectural style, but is often a combination of architectural styles that make up the complete system.

En grups de tres persones, preparar un maxim de 8 transparencies explicant un patró arquitectònic de la llista anterior o algun altre que trobeu interessant (prèvia proposta al professor).

En la presentació del patró cal incloure:

- Context o escenaris en que es pot aplicar
- Problema que intenta resoldre
- Una descripció genèrica i resumida de la solució proposada.
- Avantatges i Inconvenients
- Exemple concret d’aplicació
- Bibliografia utilitzada.