

# A Comparative Study of Selected Software Architecture Style using Quality Factor

<sup>1</sup>Shweta Adlak, <sup>2</sup>Shashi Patel

<sup>1</sup>Guest Lecture, <sup>2</sup>Assistant Professor

Department of Computer Science and Application

<sup>1</sup>Makhanlal Chaturvedi Rashtriya Patrakarita Vishwavidyalaya, Bhopal, India

<sup>2</sup>Sam Girls Collage Bhopal, India

**Abstract :** This paper covers software architecture and comparison among six different architectures. Each architecture has some advantages, disadvantages and preferred Application area. We compare this architecture on factor like reliability, performance, security, testing etc.

**IndexTerms** – Software, Quality Factor, Architecture.

## I. INTRODUCTION

The software architecture of a program or computing system is the structure of the system, which comprise software component, the externally visible properties of those components and the relationship among them. The architecture is created based on a set of requirements that it has to fulfill. These requirements are collected from the stakeholders of the system, e.g., users and developers. The functional requirements describe what the system should do, e.g., the functions that the system should provide to the users.

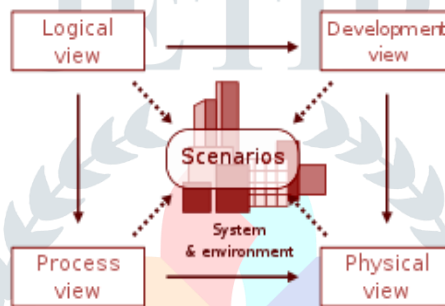


Figure 1 software architecture

### View of software architecture

**Logical view** –it should be possible to relate the system requirements to entities in this goal view.

**Process view**-this view is useful for making judgments about nonfunctional system

**Development view**-this view is useful for software managers and programmers

**Physical view**-this view useful for system engineers planning a system deployment.

## II. SOFTWARE ARCHITECTURE STYLE

Architectural styles and patterns define the way how to organize the components of the system so that one can build a complete system and achieve the requirements of the customer. They are dataflow system, call and return system, independent components and network based system.

Architecture styles are reusable packages of design decision and constraints that are applied to architecture to induce chosen desirable qualities [2]. It can be divided in to many types, but they are explanations for selective types are given below.

(I) **Blackboard architecture**

(II) **Component based architecture**

(III) **Client server architecture**

(IV) **Pipe and filter architecture**

(V) **Layered architecture**

(VI) **Object oriented architecture**

**Blackboard architecture**-A blackboard system is an artificial intelligence approach based on the blackboard architecture. It is used by several knowledge to solve the problem. There is a central data store, the blackboard, and agents writing and reading data. The agents may be implicitly invoked when data changes, or explicitly by some sort of external action such as a user command. The blackboard system is useful by dividing the core functionalities of a system into three main

Components:

- (i) **Knowledge sources**- the software specialist modules, which are called knowledge sources.
- (ii) **Blackboard**—it shared repository of problems, partial solution, and suggestion and contributed information.
- (iii) **Control component**-the control component which control the flow of problem solving activity in the system.

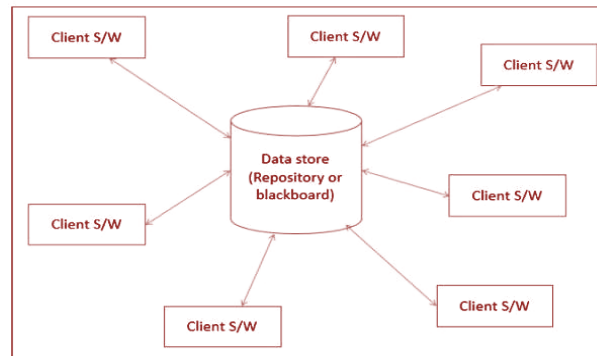


Figure 2 blackboard architecture

#### Advantage

- (i) It is software reuse.
- (ii) It is concurrency.
- (iii) Supports experimentation for hypotheses.

#### Disadvantage

- (i) Tight dependency b/w the blackboard and knowledge source.
- (ii) Difficult to make a decision.
- (iii) Synchronization of multiple agents is an issue.

**Component based architecture**- component based architecture is an architecture that focuses on decomposing software design in to functional or logical components with their own methods, event and properties.

Component-based architectural style is based on the concept of separation of concern. According to the principle of separation of concern, a system is divided into several partitions such that each partition deals with separate concern. [6]

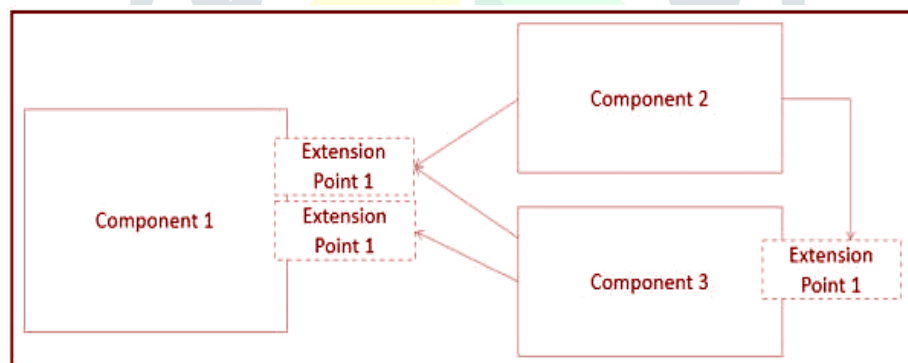


Figure 3 component based architecture

#### Advantage-

- (i) Easy to deployment.
- (ii) Reduced cost.
- (iii) Reusable.

#### Disadvantage-

- (i) Message handling.
- (ii) Testing can be difficult.
- (iii) Complexity.

**Client server architecture**-client server architecture is a computing model in which the server hosts, delivers and manage most of the resources and service to be consumed by the client. This type of architecture has one or more client computers connected to a central server over a network or internet connection. This system share computing resources. [7]

In this figure show three components client, server and medium of communication.

**Client**-client process is request of the server.

**Server-** server process accept the request of the client, perform the processing, gather the required information, generate the solution for the clients demand and send the generate response to the client process.

**Medium of communication-**it is defining the medium by which client and server communicate which each other. In this figure arrow represent medium of communication which bidirectional.

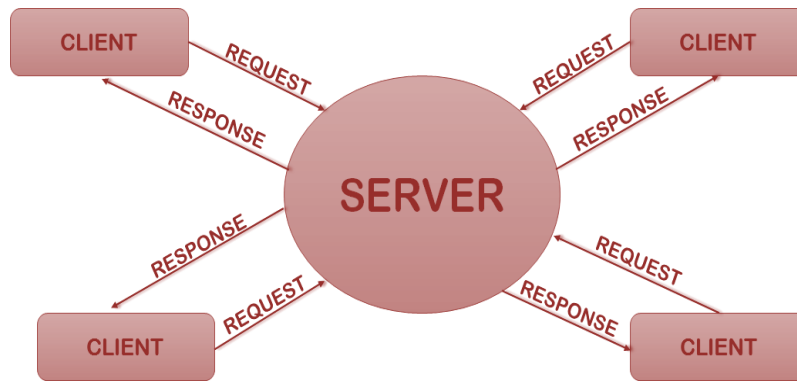


Figure 4 client server architecture

#### Advantage

- (i) Higher security.
- (ii) Centralized data access.
- (iii) Simple to maintenance.

#### Disadvantage

- (i) Server is quite expensive.
- (ii) Management of application.

**Pipe and filter architecture-**the name pipe and filter come from the original UNIX system where it was possible to link process using pipes. These passed a text stream from one process to another. A very simple and powerful architecture that is also very robust. it consist of any number of components (filter) that transform or filter data, before passing connectors (pipes) to other components. The filters are all working at the same time.

In this figure show four components pump, pipe, filter, and sink.

**Pump-**pump or producer is the data source .it can be a static text file.

**Pipe-**pipe is the connector that passes data from one filter to the next. it is directional stream of data.

**Filter-**filter is a component that transforms the data it receives the pipes. Filter can have any number of input pipes and any number of output pipes.

**Sink-**sink or consumer is the data target. it can be another file, a database, or a computer screen.

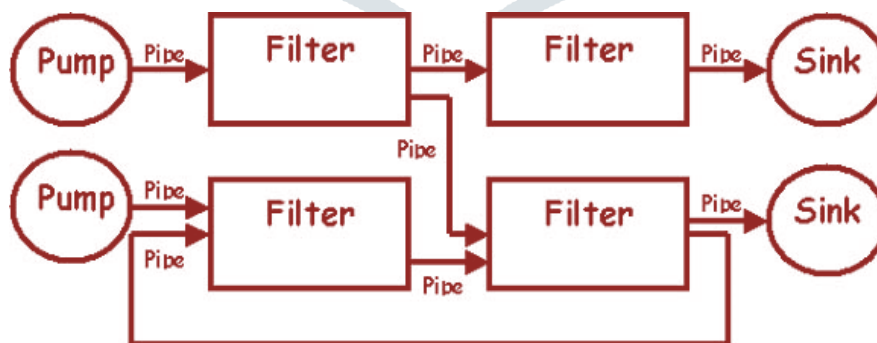


Figure 5 Pipe and filter architecture

#### Advantage

- (i) It is easy to maintain and update
- (ii) The style is conceptually simple, making the design task straightforward [8].
- (iii) Filter can execute parallel
- (iv) Filter can be implementing and tested independently

**Disadvantage**

- (i) Poor performance
- (ii) Not appropriate for interaction

**Layered architecture-** Layered architecture groups the responsibilities of the software into several loosely coupled layers. A system is layered on the basis of separation of concern communications between the different layers must only occur in the interfaces between two adjacent layers.

The best example of layered architecture is presented in OSI (Open system interconnect) and TCP/IP model. In OSI model services of the system is decomposed into seven distinct layers.

Physical layer-

Data link layer-

Network layer-

Transport layer-

Session layer-

Presentation layer-

Application layer-

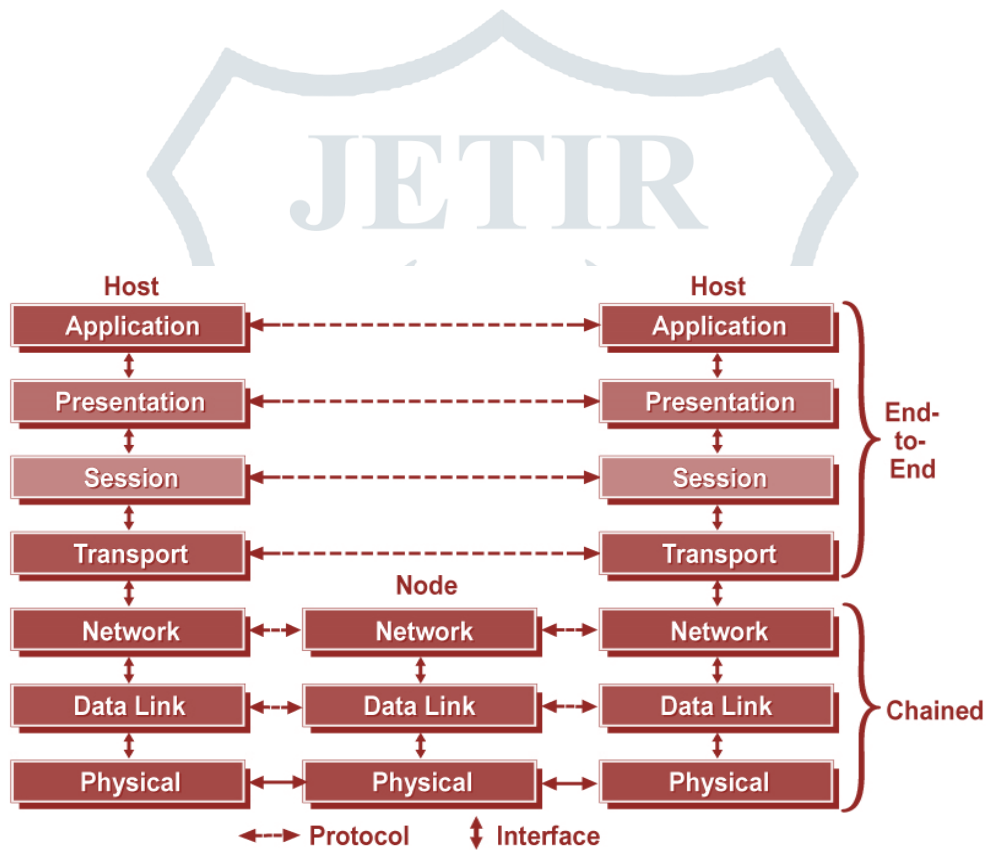


Figure 6 layered architecture

**Advantage**

- (i) Simplicity.
- (ii) Consistency.
- (iii) Increase flexibility.

**Disadvantage**

- (i) Lack of inbuilt scalability.
- (ii) Hidden use cases.
- (iii) No dependency inversion.

**Object oriented architecture-** Object Oriented Architecture is an important concept for developing the software. The popular approach of object-oriented design is to view a software system as a collection of entities known as objects. Object oriented is based on modeling real-world objects. [9]It can be discussed whether object-orientation is an architectural style or belongs to lower levels of design [10]

The four major concept in object oriented architecture.

**Class-** A class represents description of objects that share same attributes and actions.

**Encapsulation-** It is the process of binding the elements of an abstraction. It binds the data in a single unit.

**Inheritance-** it is a technique of deriving a new class from existing one.

**Polymorphism-** it means having multiple forms.



Figure 7 Object oriented concepts

#### Advantage

- (i) it more understandable
- (ii) It is easy to maintain
- (iii) Improves the quality of the system due to program reuse.
- (iv) It is reduces the development time and cost.

#### Disadvantage

- (i) Object-Oriented architecture has difficulty to determine all the necessary classes and objects required for a system.
- (ii) This methodology does not lead to successful reuse on a large scale without an explicit reuse procedure.

#### Quality factors-

**Reusability-** Reusability is a chance of using a component or system in other components/systems with small or no change.

**Performance-** Performance shows the response of the system to performing certain actions for a certain period of time.

#### Security-

**Portability-** The capability of the software product to be transferred from one environment to another. The environment may include organizational, hardware or software environment, meaning that the system is not bounded to a specific platform/location to offer its service. [4]

**Reliability-** Reliability is an attribute of the system responsible for the ability to continue to operate under predefined conditions.

**Testability-** Testability shows how well the system allows performing tests, according to predefined criteria. [3]

Table 1: Compare selected architecture using factor

Factor	Blackboard architecture	Component based architecture	Client server architecture	Pipe and filter architecture	Layered architecture	Object oriented architecture
Reliability	No	Yes	No	No	Yes	No
Performance	No	yes	Yes/no	No	Yes/no	Yes
Portability	No	Yes	No	No	Yes	No
Reusability	Yes	Yes	Yes	Yes	Yes	Yes
Security	No	yes	Yes	No	Yes	No
Scalability	Yes	Yes	Yes/no	No	No	Yes
Testability	No	Yes/no	No	Yes	Yes	No

Table 2: Area of application in selected architecture

Model	Area of application
Blackboard architecture	Computer vision, case based reasoning, symbolic learning
Component based architecture	Pluggable application
Client server architecture	File transfer, mail transfer, web based application
Pipe and filter architecture	Unix programs, compilers
Layered architecture	OSI model, mobile phone, telecommunication domain
Object oriented architecture	Real time system, object oriented database, AI and Expert system

### III. CONCLUSION

In this study we discussed on various software Architecture. Each architecture has its unique feature and specialty. All above discussed architectural styles are compared on the basis of quality attributes i.e. Reliability, Performance, Portability, Scalability, Security, Testability, Reusability, and presented in the table 1. They are also compared on the basis of application areas. Since Blackboard architecture is mainly used in the field of Artificial Intelligence, it results in high complexity because artificial intelligence applications are difficult and devious to build.

### REFERENCES

- [1]. Dobrica, Liliana, and EilaNiemela. "A survey on software architecture analysis methods." *Software Engineering, IEEE Transactions on* 28.7 (2002): 638-653.
- [2]. UCI Software Architecture Research – UCI Software Architecture Research: Architectural Styles. *Isr.uci.edu*. Retrieved on 2013-07-21.
- [3]. <https://hackernoon.com/quality-attributes-in-software-architecture-3844ea482732>
- [4]. Pijpers, V., J. Gordijn, 2008. Considering Software Quality Requirements as Networked Business Quality Requirements. Technical Report, HEC University Lausanne
- [5]. Hayes-Roth, B. (1985). "A blackboard architecture for control". *Artificial Intelligence*. 26(3): 251–321. doi:10.1016/0004-3702(85)90063-3.
- [6]. Heineman, George T., and William T. Council. "Component-based software engineering: putting the pieces together." (2001).
- [7]. <https://www.techopedia.com/definition/438/clientserver-architecture>
- [8]. David Garlan and Mary Shaw, *An Introduction to Software Architecture*, CMU-CS, Pittsburgh, PA, 1994, pp. 6-8.
- [9]. <https://www.tutorialride.com/software-architecture-and-design/object-oriented-architecture.htm>
- [10]. Land, R., 2002. A Brief Survey of Software Architecture. Mälardalen Real-Time Research Center (MRTC) Report Department of Computer Engineering, Mälardalen University, Vasteras, Sweden.

