

WIS ARCHITECTURE REPORT

Group C1.04.05
13/02/2023

Author(s):

Carlos Bermejo Soria, carberson@alum.us.es

Daniel Gallardo Martos, dangalmar@alum.us.es

Pedro González Marcos, pedgonmar2@alum.us.es

Fernando José Mateos Gómez, fermatgom@alum.us.es

Carlos Zarzuela Reina, carzarrei@alum.us.es

Table of contents

Table of contents	2
1. Executive summary	3
2. Revision table	3
3. Introduction	3
4. Contents	3
4.1. What is a WIS	3
4.2. Client server 2-tier application	3
4.3. Client server 3-tier application	4
4.4. Client server n-tier application	4
5. Conclusions	4
6. Bibliography	4

1. Executive summary

This report is provided to the lecturer in order to show the knowledge acquired about WIS previous to this subject.

2. Revision table

Version	Date	Description
1.0	13/02/2023	Initial draft of the document.
1.1	14/02/2023	Added WIS architecture tiers

3. Introduction

In the following report we will be discussing the acquired knowledge of the group regarding web information systems architecture.

4. Contents

4.1. What is a WIS

Firstly, we must talk about the concept of a web information system. A web information system is a type of information system that uses the internet to deliver information or services to users or other applications.

In this type of applications, it is commonly used the server-client architecture pattern. The users act as clients requesting information to the server normally using http protocol and the server processes client information in order to give them a response. We have to take note that not only real people can act as clients, other types of systems can also be clients too, for instance Netflix microservice based network.

4.2. Client server 2-tier application

The simplest scenario we could possibly think of is a 2-tier client server system. The structure is simple. The browser running in the computer acts as the client and the web server process multiple client requests. The requests sent by the user are handled by a web server and then redirected to an application to process its information (commonly called the application server) and finally the application server will interact with a DBMS to retrieve the requested information.

If the web server, the application server, and the DBMS is located in the same machine it follows a 2-tier client server.

This type of architecture has many disadvantages:

- As all these components are in the same place, users will be interacting with an overloaded server and will not perform well leading to an undesired quality of service to clients.
- Another downside to this architecture is that as all its components are stacked together if one of them fails the whole system will be completely down. This is an undesired problem that modern business cannot afford to have.

4.3. Client server 3-tier application

In this scenario, the structure is the same except that the DBMS is located in a different machine. This is another commonly used pattern adopted in the industry.

We could split the 2-tier system into a 3-tier application by delegating the responsibility of storing data and serving raw data from a different machine with a DBMS inside.

In this model we have the data storage in another physical layer enabling us to add service reliability to the users. We could deploy our application in many different servers that interact with our database, if the web or application server fails in a machine the service will not be as compromised as in the 2-tier application.

4.4. Client server n-tier application

This type of architecture is commonly used in large electronic commerce enterprises that are totally dependent on 24 hour working services.

Mention that microservices architecture are built using different applications encapsulating one feature of all the business logic. The applications are made following REST architecture. In the industry these RESTFUL web services are normally managed and deployed using an orchestration system like Kubernetes.

5. Conclusions

We have talked about the different types of architecture that are commonly used in a WIS, we have also discussed the advantages and disadvantages of the different architecture tiers.

6. Bibliography

Previous information was extracted by coursing these subjects:

- Introduction to SQL and DBMS: INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE Y LOS SISTEMAS DE INFORMACIÓN 1 Second year Software engineering

- Software architecture patterns: INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE Y LOS SISTEMAS DE INFORMACIÓN 2 Second year Software engineering
- Software architecture patterns, introduction to microservices (only theory lecture), design of a API RESTFUL and deployment of an API in App Engine: ARQUITECTURA E INTEGRACIÓN DE SISTEMAS SOFTWARE 2 Second year Software engineering
- Packaging software in containers (Podman) SISTEMAS OPERATIVOS Second year Software engineering
- Software architecture patterns: DISEÑO Y PRUEBAS 1 Third year Software engineering

External sources:

- Tiers in a WIS:
["https://teaching.shu.ac.uk/aces/rh1/de/web_based_systems_architectures_1_tutorial.htm"](https://teaching.shu.ac.uk/aces/rh1/de/web_based_systems_architectures_1_tutorial.htm)