Architectures for distributed information systems research paper example

Sociology, Communication

\n[toc title="Table of Contents"]\n

\n \t

\n[/toc]\n \n

The near future tends to hint at the fact that information systems will have to be heterogeneous, open and distributed. The concept of distributed computing has become the norm in almost all organizations ranging from small companies to multinational conglomerates. Numerous individuals have been employed in this sector. The idea to move away from centralized computing resulted from the availability of reliable data communication networks. The desire by various organizations to make efficient and cost effective use of computing resources can also be attributed to the rise of distribution in information systems (Arregoces and Maurizio112). Distributed computing enables organizations to right-size through the means of allowing organizations to replace large and costly central processors with ones that are smaller, flexible and relatively cheap.

The distribution and heterogeneity at the hardware level come with a few problems of its own. On the other hand, the software application level is typified by an array of systems including those that are involved in mission-critical legacy systems that have been left over from the gone days of centralized computing. This paper analyzes the n-tier web applications, how

it aids in scaling growth, its limitation and the types of computations that are performed. The paper also looks at the types of data that can be stored and where within the architecture the data is stored. The final aspect that is analyzed is that of the various types of interactions with other external systems that can be found in such architectures (Santana 101).

N-Tier Data Application

It can be described as a data application that has been separated into various multiple tiers. This means that this application has separate processing units for discrete tiers that have been distributed between a server and the client. In a more elaborate explanation, one can say that it is a client-server architecture where the data management, application processing and presentation are all separate processes. For instance, an application that utilizes middleware to service its data requests between a database and a user tends to employ multi-tier architecture. A person who develops applications that are involved in data access should have a clear separation between the various existing tiers that make up the application (Bradford, Ed, and Lou 69).

There are several forms of the n-tier application, but the most common one is the 3-tier application. The-tier application structure tends to imply and emphasize more on the client to server program model. In the event that there are more than three distribution tiers involved, all the additional tiers are usually associated with the business logic tier. The are several advantages that come with the use of the-tier, and one of them is that it can run on the appropriate processor, and it can also be updated independently

of the other levels.

Basically, an n-tier has three levels; the presentation, middle and data tier. The best way of using the n-tier data application is by separating application components into separate tiers. This leads to an increase in the scalability and maintainability of the application. It achieves this by allowing an easier adoption of new and novel technologies which can be applied to a single tier without having to redesign the entire solution (Stair, Ralph, and George 89). Disadvantages are several within the-tier structure and some include; interfaces within the-tier structure are static. This means that the desire to change them frequently is tantamount to going back to the problem that a person is trying to solve. Another great downside of the-tier application is that it does not necessarily lead to an increase in the speed of the application being used. The reason behind this being that every effort to redesign an application results in an overcode intermodule communication and object encapsulation (Stair, Ralph, and George109).

Three-tier Architecture

Presentation tier is at the apex of the application. It represents the topmost level, and it displays information that is related to service like purchasing, browsing merchandise and shopping cart contents. Its main work is to communicate with the other two tiers and transmit the results to the client tier and all the other levels in the network. In layman terms, it can be described as a layer that can be accessed by users directly. Examples are web pages and operation systems user interface. It also accesses the middle tier by use of service reference. It uses the middle layer as an access to the

data layer because it does not have a direct connection to the data tier (Horowitz, Ellis, and Sartaj 101).

The second layer is one the known as the middle tier or the logic tier. It acts as a midpoint through which the presentation and data tier use to communicate to one another. The business element of the middle layer is engaged in processes such as data validation and business rules. The other component of the middle layer is the data access and logic unit (Santana 122). It is involved in calculations and making of logical decisions and evaluations. A crucial aspect that it is involved in is the movement and processing of data between the two surrounding layers.

The third and last layer of an-tier application architecture is known as the data tier. The data level or tier is involved in the storage and retrieval from the existing database. It stores all the application data. Data persistence mechanisms are contained in this layer and aid in the exposure of the data. The concept of making changes independent of the application tier means that the data layer is crucial. By ensuring that dependencies are avoided on the mechanisms used for storage, the independent updating and changing can be achieved without the clients on the application tier noticing. From the data layer, information is passed back to the logic layer. It moves from the logic layer and given back to the user or client at the application tier.

Conclusion

In a nutshell, one realizes that the-tier has numerous advantages and disadvantages. However, one advantage stands out conspicuously. Its three-tier architecture allows for independent upgrade and replacement in regards

to the changes and new developments in technological requirements. Summed up, it can be differentiated into three distinct levels, namely; the presentation, middle or logic and data tier. The most comforting notion and capability of the-tier application are that it allows a person to balance between the processing load and the resources at hand, thus resulting in the best possible performance. All the three tiers are independent as compared to the centralized system, however on layer relies on the other for a specific role it plays in the-tier application system.

Works Cited

Arregoces, Mauricio, and Maurizio Portolani. Data Center Fundamentals. Indianapolis, Ind: Cisco, 2003. Print.

Bradford, Ed, and Lou Mauget. Linux and Windows Interoperability Guide. Upper Saddle River, N. J: Prentice Hall PTR, 2002. Print.

Horowitz, Ellis, and Sartaj Sahni. Fundamentals of data structures. Pitman, 1983.

Santana, Gustavo A. A. Data Center Virtualization Fundamentals. Indianapolis, IN: Cisco Press, 2014. Internet resource.

Stair, Ralph, and George Reynolds. Fundamentals of information systems. Cengage Learning, 2013.