William Simpson

WEB420

17June2018

Assignment 7.3 Legacy to SOA

1.1 Service-Oriented Architecture design diagram.

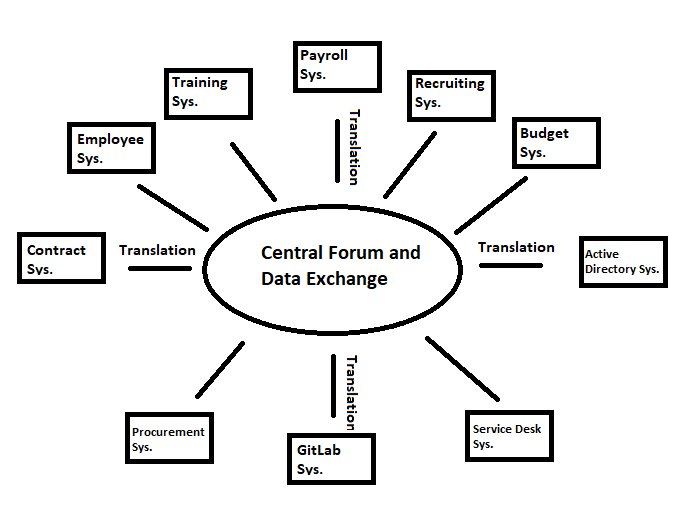


Diagram Explanation:

Figure 1.1 represents a new possible design for our Legacy architectural systems. Previously, each system was driven by its own data, and each applications data was only able to be processed by the specific system that it was created for. This is due to the fact that there is no middleware or implementation strategy that focuses on making the data accessible by all applications on the system. A legacy system such as ours provides many obstacles to any kind of scaling of systems, for each new application it needs its own language and data interface and database that won’t necessarily be compatible with all the other systems that we currently use.

The proposed Service-Oriented Architecture, is meant to create a centralized storage and data exchange for our systems and applications. It is here inside the centralized storage and data center that information will be sent from each individual application and through the use of an Enterprise Service Bus API, the data will be translated from the centralized data structure, to the designated data type that the application uses, and back again. This will be extremely beneficial because it means that applications coded in completely different languages, utilizing different data schemes and technologies will be able to send data to each other directly. For example, instead of data being sent from the training system application to its own storage of data, that information will now be centrally located and accessible, which means that the employee system could pull that data and update itself once and employee has received new or necessary training. This in turn could update the payroll system if the training results in a pay increase and would finally trickle down to the budget system since the extra deduction would need to be taken into account. Previously, each data our system would have either been unable to share the data or would have had to have been manually translated into a language that could be utilized by both.

Setting up an efficient and structurally sound Service-Oriented Architecture is quite the upfront investment, there is a lot of software and planning that will be needed for the transition. Much like an SQL database, it takes longer to get it ready, but once it is up and running it will be exceptionally powerful.

The first step of the migration we will have to determine which applications need to be up and running right away, how best to update these systems, and an order for which every application following will be migrated. Taking a top priority will be the financial systems, specifically the payroll and budget systems. These are key elements in keeping operations running smoothly. Unpaid employees aren’t the best workers. After the payroll and employee systems are transitioned and their data is able to exchange with the central database, our GitLab, training, active directory and service desk functions will be the next most important. Finally, we will bring the procurement, contract systems online. Following these steps should insure that everything is running as smoothly as possible for the transition.