Commentary on Bernstein’s atricle   
“Middleware: a model for distributed system services”

Ing. Alberto Gabriel Ramos Salvio

Centro de Investigación Científica y de Educación Superior de Ensenada

Although the article Middleware: a model for distributed system services was written in nineteen ninety-six, I feel it is still very relevant today as Middleware is an abstraction layer that allows for better cleaner code.

Supporting many different systems in one’s software package is a really difficult thing to do since there are many protocols and proprietary call that differ on each system. Having an abstraction layer really makes developing software much easier. We can see examples of this in the past. Much like the basic input output system on personal computers allowed operating system designers to support a great range of hardware, Middleware allows software developers to focus on writing the core code without worrying of supporting different server types. This was achieved through creating a standardized set of functions that could be called independently of which system the application was running on. The middle ware then translates those function calls to the correct system class for the system that it is running on.

Another interesting point that the article made which rings very true is that costumers don’t buy servers, they buy applications so if your server is not compatible with a popular application this would mean that customers are more likely to buy from the competition. Thus middleware is usually developed by vendors to make their servers more compatible. This to me seems like a symbiotic relationship between vendors and server application developers.

References

Bernstein, P. A. (1996). Middleware: a model for distributed system services. Communications of the ACM, 39(2), 86-98.