Bethany Beery

CIT275 – Instructor Jeff Straw

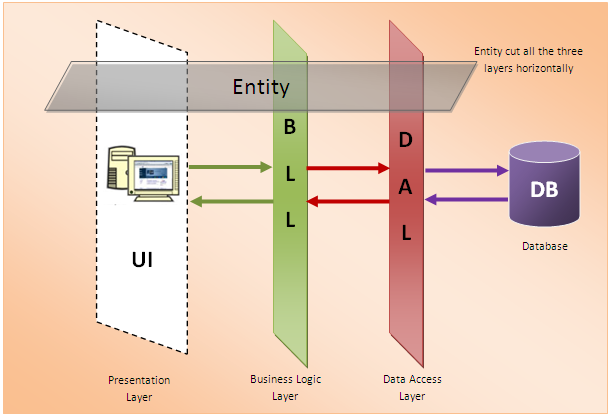
Northwestern Michigan College

MULTI-TIER ARCHITECTURE

Multi-tier architecture refers to the organizational structure of the application. Each layer or tier is logically separated by concern. For example, most commonly employed is a three-tier architecture which divides the application into a presentation layer, a business service layer, and a data access layer.

The presentation layer is the graphical user interface; the part of the application the user interacts with. This layer allows for data manipulation and data entry into a database. In the case of WPF and Window Store Applications, this would also include the XAML code. Next, the business service layer processes all commands. It enforces all business rules, performs calculations, and evaluates data. This layer acts like a mediator between the presentation and data layer. Finally, the data layer interacts directly with a database. This can be a SQL Server database, MySQL, Access, or data stored in a permanent location. It will store or retrieve data and then send it back to the business layer for processing and final delivery to the presentation layer.

The benefit of separating out concerns is the ability for each layer to be independent of each other. This allows for flexibility, manageability, scalability, and code reusability. Also, multiple developers can work on the same project without affecting each other’s work. Each layer can be dramatically changed with little to zero impact on the other layers. For example, a change of operating system in the presentation tier would only affect the user interface code. Following is a graphical representation of a basic three tier architecture found on stackoverflow.com.



Full credit goes to [Moiz](http://stackoverflow.com/users/1613827/moiz)’s response post on Stackoverflow:

http://stackoverflow.com/questions/13786549/how-to-implement-3-tiers-architecture-in-c-sharp