3-Tier Client/server Architecture

The presentation server is any input device that we can use to control an SAP system. But we're not limited just to using a SAP GUI. We can a web browser, mobile device or any other form of input you can think of.

The presentation layer communicates with the application server.

And the application server is the brains of an SAP system. This is where all the central processing gets done. You can see here, we're showing the application sever isn't just one system in itself. Your application server can be made up of multiple instances of the processing system.

Now the application server in turn communicates with the database layer of the 3-tier architecture.

The database is kept on a separate server, a separate system in itself, mainly for performance reasons, but also for security as well. It provides a separation and that's why we got these three different layers in this whole SAP system.

The presentation server communicates with the application server. The application server does all the processing, it makes calls to the database. Data is passed back to the application server. more processing is done before the results are then sent to the presentation server.

Landscape Architecture

Now let's quickly discuss a typical landscape architecture. Now, I say typical, but you'll find that when you work with SAP, there is no typical in inverted commas landscape architecture that most companies use.

What you do find that's very common, is you'll find the development system, you'll find the testing system and then you'll find a production system. Now why do we have these three systems?

All the development work and initial unit testing that we do in our SAP work gets done on a development system. This ensures we do not affect any other system that is being used by the company. Once our developments, we think, are good enough to be tested by maybe an external source or someone else within your company whose role it is to carry out testing. We move our developments using what's called a transport system to the next system.

Which, in this case, is the testing system. On the testing system normally, no developments is done at all. It's just used for testing what developments were carried out in the developments system. If everything works out and everything passes in the test system.

Within user transport system again to move our developments or our program changes into the production environment. When code goes into a production environment, that's when it's turned on and that's when it's used within the business itself. Now our landscape architecture is not separated just for development purposes. Your company can have other reasons.

They can be the quantity of data that anormal production system holds. It can be too great to actually be used in the development environment because normally your development system and your testing system are not as large as a production system. You only want a subset of data to test on. There is also the security element that you need to look at.

Often, companies do not want developers to see live production data for data security issues. one thing that is common is that each system that you do have in your landscape architecture will have it's own application server and it's own database server. This then ensures we have platform independence.