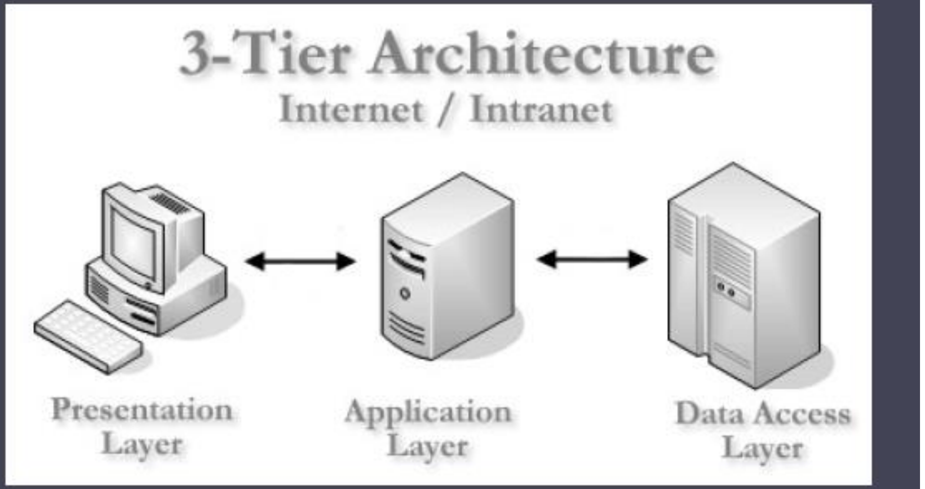


3- tier Architecture

Topics Covered

- What is 3-tier Architecture?
- Evolution of 3-tier Architecture Development.
- What does 3-tier Architecture Offer You?
    - Advantages & Disadvantages
- An Example of how to implement 3-tier Architecture in .Net Platform Using C#

What is 3-tier Architecture?

- A three-way interaction in a client/Server environment
    - The User Interface is stored in the Client.
    - The Business Application Logic is Stored in one or more Servers.
    - The Data is Stored in a Database Server.

Presentation tier

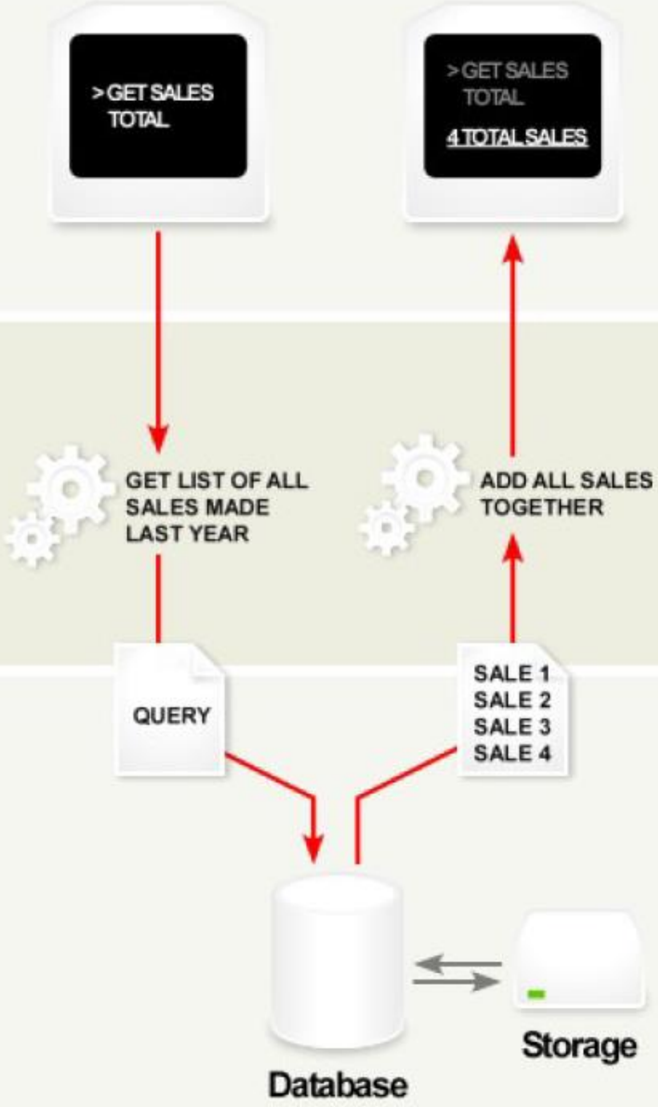
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

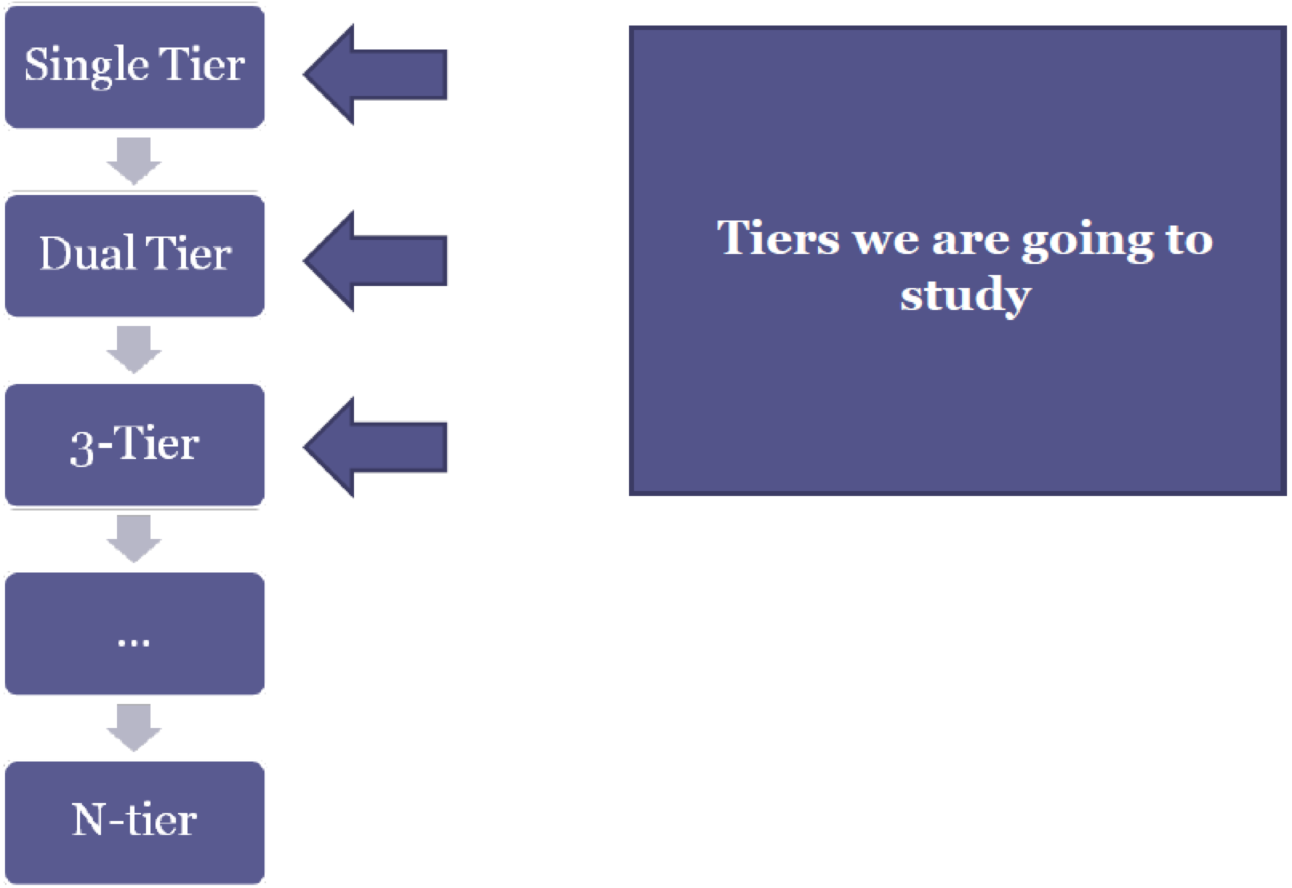
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.



Evolution to the 3-tier Architecture



Single Tier

Dual Tier

3-Tier

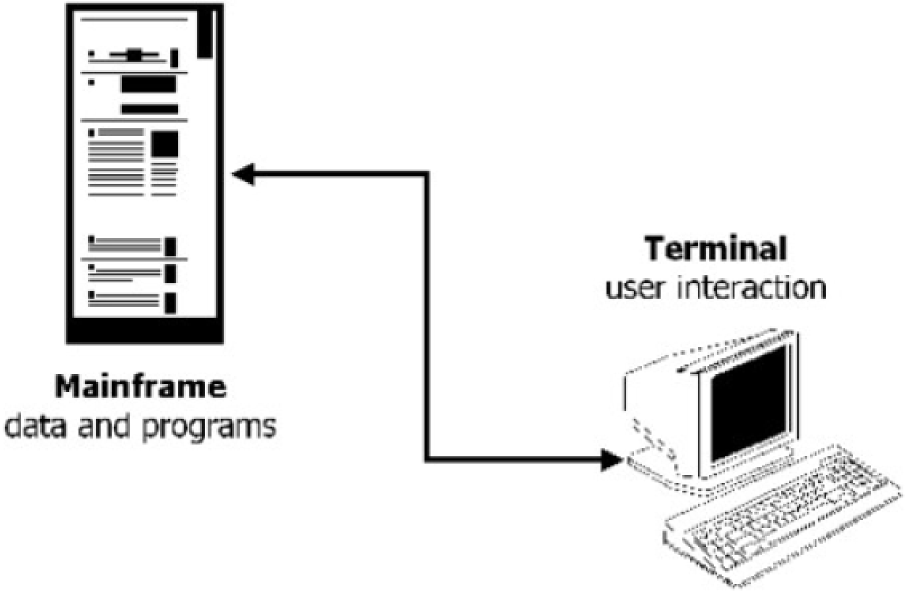
...

N-tier

Tiers we are going to study

Single Tier Architecture

- Time of Huge "Mainframe"
- All Processing in Single Computer
- All Resources Attached to the same Computer
- Access Via Dumb Terminals

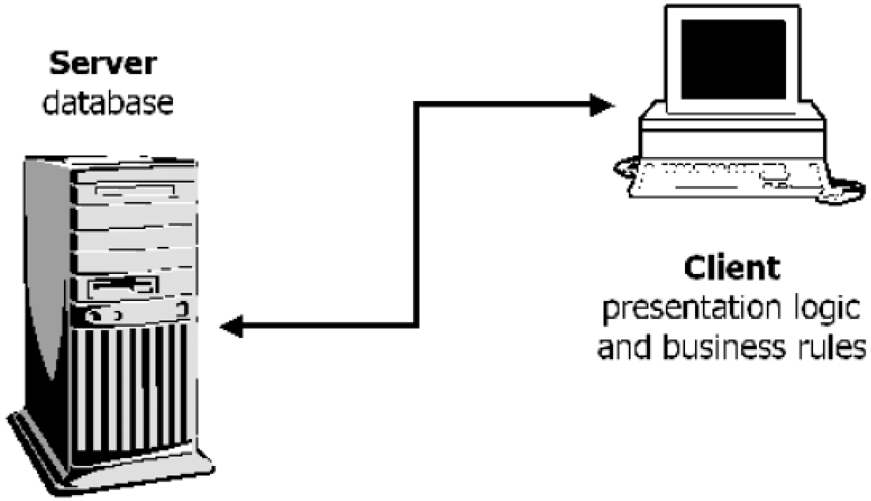


Single Tier – Advantages & Disadvantages

- Advantages
  - Simple
  - Efficient
  - Uncomplicated
- Disadvantages
  - Very Expensive

Dual Tier Architecture

- The Personal Computer
- Necessity of Providing Personal Software
- The Client Server Model was Born!!
- Logical System Components – Most of which are on the Client

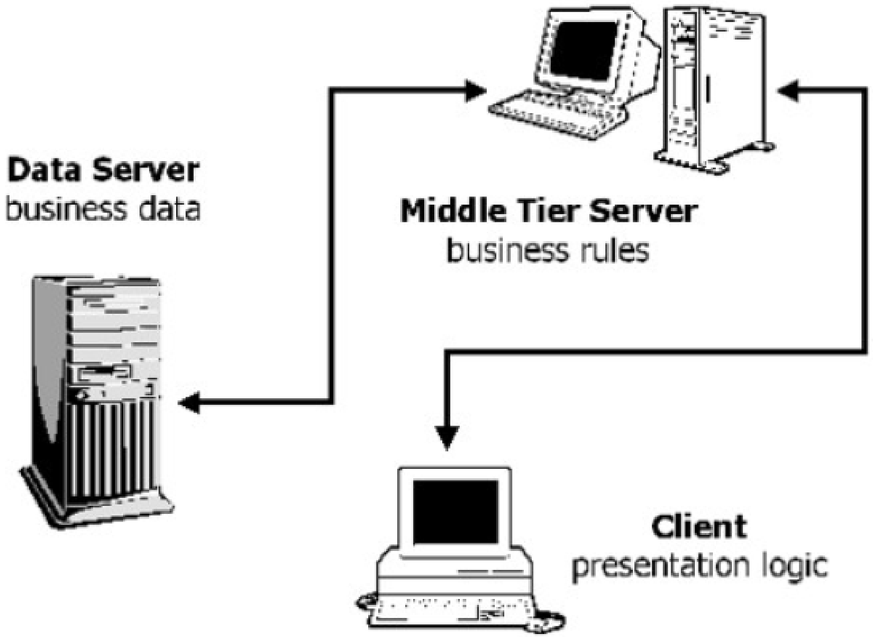


Dual Tier – Advantages & Disadvantages

- Advantages
  - Less Expensive than Mainframe
- Disadvantages
  - The Connections to the Database Server are very Expensive
  - One can only connects a limited number of users to a server before Database Server spends more time managing connections than processing requests
  - Cost-ineffective. Many users only use their connections 2-3% of the time.

3-Tier Architecture

- These Applications runs on the Traditional Client/Server Model But from a Application server.
- Client only Displays the GUI and data, but has no part in producing results
- Database Server Serves to few Connections



3-Tier Advantages

- Scalability
  - The Application Servers can be deployed on many machines
  - The Database no longer requires a connection from every client.
- Reusability
  - If a standard object is employed, the specific language of implementation of middle tier can be made transparent.
- Data Integrity
  - The middle tier can ensure that only valid data is allowed to be updated in the database.

3-Tier Advantages

- Improved Security
  - Since the client doesn’t have direct access to the database, Data layer is more secure.
  - Business Logic is generally more secure since it is placed on a secured central server.
- Reduced Distribution
  - Changes to business logic only need to be updated on application servers and need not to distributed on clients
- Improved Availability
  - Mission Critical Applications can make use of redundant application servers and redundant application servers, so it can recover from network of server failures.

3-Tier Disadvantages

- Increased Complexity / Effort
  - In General 3-tier Architecture is more complex to build compared to 2-tier Architecture.
  - Point of Communication are doubled.

Conclusions

- Complex business rules easy to implement in application server layer.
- Business Logic off-loaded from database server and client, which improve performance
- Changes to business logic automatically enforce by server.
- All tiers can be platform independent.
- Superior Performance for medium to High Volume Environments