**Differences Between Two-Tier and Three-Tier Database Architectures**

***Two-Tier Architecture***

The two-tier architecture, also known as client-server architecture, consists of two main components:

1. *Client Tier*: The user interface runs on the client machine. This layer contains the application software that interacts directly with the database.
2. *Server Tier*: The database server manages the database and processes queries from the client.

In this architecture, the client is responsible for presenting the data and executing business logic, while the server handles the database management. This direct communication between client and server can result in quicker data access but may become inefficient with a large number of clients (Elmasri & Navathe, 2016).

***Three-Tier Architecture***

The three-tier architecture introduces an additional layer between the client and the server, typically comprising:

1. *Client Tier*: The presentation layer where the user interface resides.
2. *Middle Tier*: The application server where business logic and data processing are executed. This layer acts as an intermediary, processing client requests and communicating with the database server.
3. *Data Tier*: The database server that stores and retrieves data.

This separation allows each tier to be developed, managed, and scaled independently. The middle tier improves performance and scalability by handling multiple client requests efficiently and offloading business logic from the client (Kroenke & Auer, 2013).

***Major Differences***

1. **Scalability**:
   * *Two-Tier*: Limited scalability as the client directly communicates with the database server, leading to bottlenecks as the number of clients increases (Elmasri & Navathe, 2016).
   * *Three-Tier*: Better scalability since the middle tier can handle numerous clients and distribute the load more effectively (Kroenke & Auer, 2013).
2. **Maintainability**:
   * *Two-Tier*: Less maintainable due to tightly coupled client and server components. Changes in business logic require updates to all clients (Elmasri & Navathe, 2016).
   * *Three-Tier*: Enhanced maintainability because the middle tier can be updated independently, minimizing the impact on clients and servers (Kroenke & Auer, 2013).
3. **Performance**:
   * *Two-Tier*: May offer faster performance for smaller systems as client requests are processed directly by the server (Elmasri & Navathe, 2016).
   * *Three-Tier*: Potentially better performance for larger systems due to load balancing and efficient resource management by the middle tier (Kroenke & Auer, 2013).
4. **Security**:
   * *Two-Tier*: Less secure as clients have direct access to the database, increasing vulnerability (Elmasri & Navathe, 2016).
   * *Three-Tier*: Improved security with the middle tier acting as a buffer, protecting the database from direct client access (Kroenke & Auer, 2013).

**References**

Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of database systems* (7th ed.). The Handa Library. <http://library.uc.edu.kh/userfiles/pdf/7.Fundamentals%20of%20Database%20Systems.pdf>

Kroenke, D. M., & Auer, D. (2013). *Database concepts* (6th ed.). Tolino Labs. <https://api.pageplace.de/preview/DT0400.9781292076249_A24580004/preview-9781292076249_A24580004.pdf>