Case Study(9)

Task(1): Product teams need to incorporate security concerns into their product plans, not as an ancillary service, but as a core feature of their product.

Each week brings news of new attacks and new vulnerabilities.

In fact, this week's headlines ranged from flaws in fingerprint scanning software to Rammstein blaring ransomware.

Product teams do not need to become experts in ALL areas of cybersecurity. Security is a product team's responsibility.

One of the most important things a product team can do to improve the security of its product is simple in concept but difficult in practice: developing a deep and real understanding of and empathy for customers .

I'll cover risk profiles, threat models, friction, usability testing, and frequent security challenges for product owners.

Task(2): **Client-server pattern**

This pattern consists of two parties; a server and multiple clients.

The server component will provide services to multiple client components.

Clients request services from the server and the server provides relevant services to those clients. Furthermore, the server continues to listen to client requests.

This pattern is used to create a number of different types of software products. One example of a company that uses this pattern is Blizzard's Battle.net service which hosts online games for World of Warcraft.

Some examples of software products that are commonly made by client-server architecture pattern include:

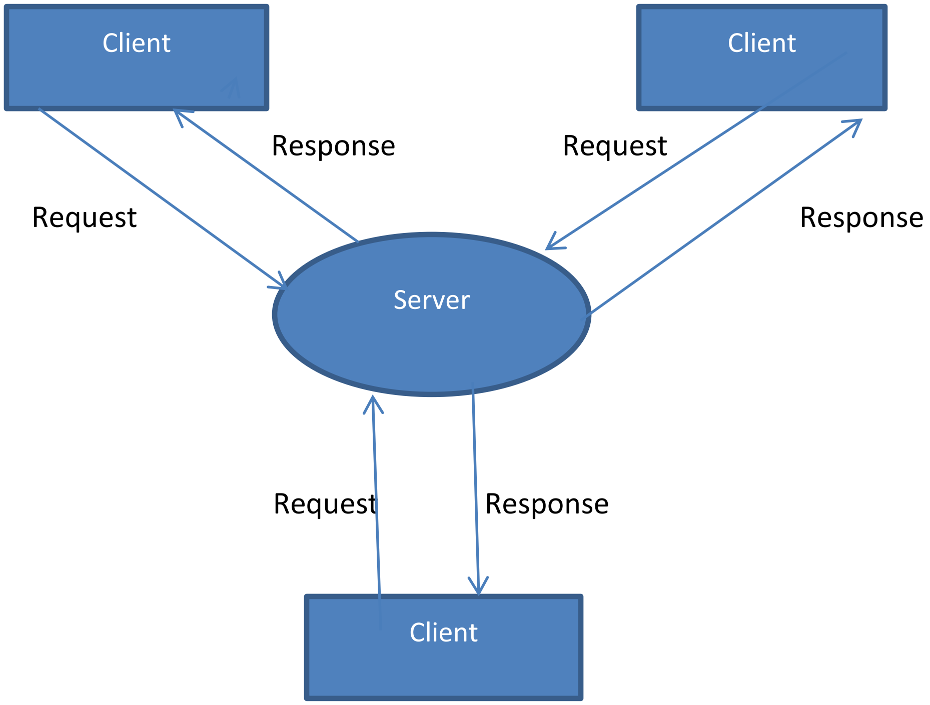
.Email server providers

.File-sharing services

.Online banking

.Online multiplayer gaming

Client-server software design pattern allows developers create a software product centered on a single server providing the same resource to multiple users at the same time.



Client-server software architecture pattern

Task(3) : Typically, business-process benefits impact enterprise architecture maturity and governance.

These controls act as a supporting framework for decision making.

Enterprise architecture improves organizational impacts through productivity, agility, product and service timeliness, revenue growth and cost reduction.

The enterprise architecture benefits include more efficient business operation with lower costs , more shared capabilities, lower management costs, more flexible workforce, more organization, less duplication and redundancies and improved business productivity .

Task(4): **Advantages of layer architecture**

Layered architecture increases flexibility, maintainability, and scalability. In a layered architecture we separate the user interface from the business logic and business logic from the access logic.

Separation of concerns among these logical layers and components is easily achieved with the help of layered architecture.

Layered architecture enables teams to work on different parts of the application parallel with minimal dependencies on the teams.

Layered architecture enables develop loosely coupled systems.

Layered architecture also makes it possible to configure different levels of security to different components deployed on different boxes.

Layered architecture enables you to secure portions of the application behind the firewall and make other components accessible from the internet.

Layered architecture helps you to test the components independently of each other.

Disadvantages of layered architecture

There might be a negative impact on the performance as we have the extra overhead of passing through layers instead of calling a component directly.

Development of user-intensive applications can sometime take longer if the layering prevents the use of user interface components that directly interact with the database.

The use of layers helps to control and encapsulate the complexity of large applications, but adds complexity to simple applications.

Changes to lower level interface tend to percolate to higher levels, especially if the relaxed layered approach is used.