**Case Study 9**

**Task 1:**

Security Concern is the function of threat and vulnerability. Product teams need to incorporate security concerns into their product plans, not as an ancillary service, but as a core feature of their product. Why? For one, digital insecurity has become normalized. It is pervasive, multi-headed hydra. Each week brings new of new attacks and new vulnerabilities. In fact, that week's headlines ranged from flaws in fingerprint scanning to Rammstein-blaring ransomware. There is little doubt that such attacks and vulnerabilities will follow Moore's law and continue to grow exponentially.

Product teams do not need to become experts in ALL areas of cyber security — just the ones that touch your product. Here are three simple heuristics to consider:

1. Each new platform is a new attack vector.
2. Over time, attacks get more sophisticated.
3. Frequently, human behavior is the weakest link in the security chain

**Task 2:**

The following application architecture pattern should use to develop new system or to maintain current existing system:

- Layered (N-Tier) architecture

- Model-view-controller architecture

- Microkernel architecture

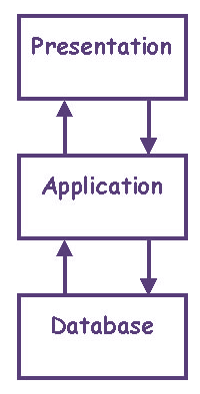
- Microservices architecture

- Event-driven architecture

- Client-server architecture

- Peer-to-peer architecture

**N-Tier architecture**

**The n-tier architecture (multi-tier architecture)** is an industry-proven software architecture model. It is suitable to support enterprise level client-server applications by providing solutions to scalability, security, fault tolerance, reusability, and maintainability. It helps developers to create flexible and reusable applications.

N-Tier Architecture Diagram

These three layers can be further subdivided into different sub-layers depending on the requirements.

There are different types of N-Tier Architectures, like **3-tier Architecture, 2-Tier Architecture and 1- Tier Architecture.**

**3-Tier Architecture**: has three different layers

* Presentation Layer: is user interface layer.
* Business Layer: coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations.
* Database Layer: here information is stored and retrieve from a database or file system.

**2-Tier Architecture:** It is like Client-Server architecture, where communication takes place between client and server. In this type of software architecture, the presentation layer or user interface layer runs on the client side while dataset layer gets executed and stored on server side. There is no Business logic layer or immediate layer in between client and server.

**Single Tier or 1-Tier Architecture:** It is the simplest one as it is equivalent to running the application on the personal computer. All of the required components for an application to run are on a single application or server. Presentation layer, Business logic layer, and data layer are all located on a single machine.

**Task 3:**

Product team should use Tier Architecture. Because it provides many benefits for production and development environments by modularizing the user interface, business logic, and data storage layers. A tier application program is one that is distributed among three or more separate computers in a distributed network. This architecture model helps software developer to create reusable applications/systems with maximum flexibility. And also separate the user applications and physical database, support DBMS characteristics, program data independence, support multiple views of data.

**Task 4:**

Advantages of N-Tier Architecture

* Scalability
* Flexibility
* Reusability
* Improved Security
* Data integrity
* Easy to manage

Disadvantages of N-Tier Architecture

* Increase in effort
* Increase in complexity