Layered Architecture Style by Roast Duck

Simon – What is layered architecture style

Nova – Advantages

Brendan – Disadvantages (Negative behaviours)

Andrew – Why and how it is useful over time

What is layered architecture style?

Layered architecture style is the most common architecture style. Modules or components with similar functionalities are organized into horizontal layers, therefore, each layer performs a specific role within the application.

The layered architecture style does not define how many layers are in the application. Architects, designers, and developers could have as many layers as they want while they are developing an application.

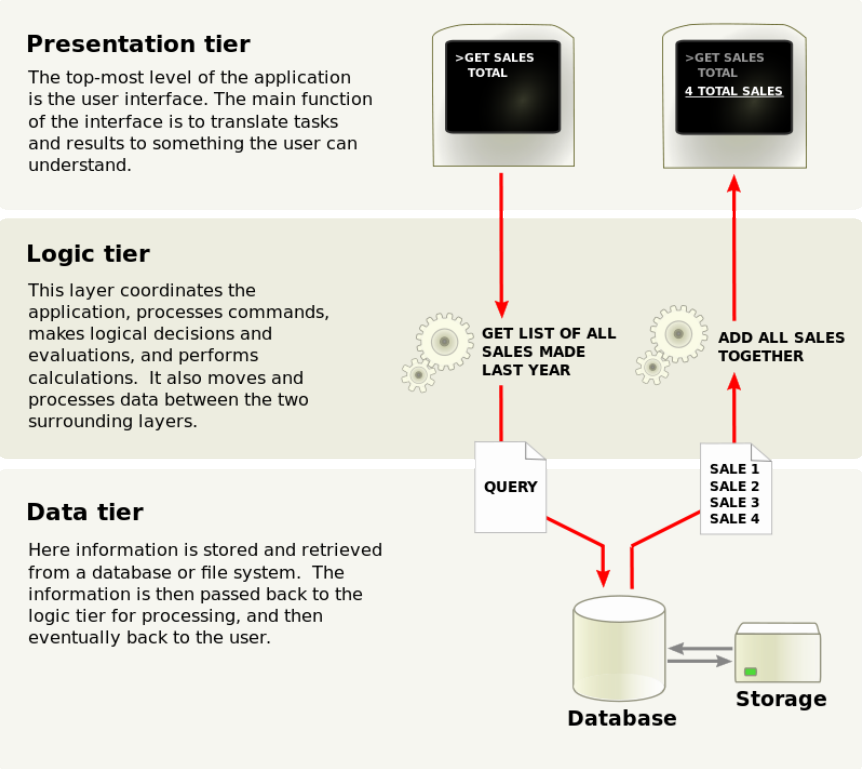
Layered architecture style abstracts the view of the system as whole while providing enough detail to understand the roles and responsibilities of individual layers and the relationship between them.

Isolation: each layer is independent of the other layers, thereby having little or no knowledge of the inner workings of other layers in the architecture(topological constraint) but expose interface (API) to be used by above layers.

Most applicable to specific kinds of problems?

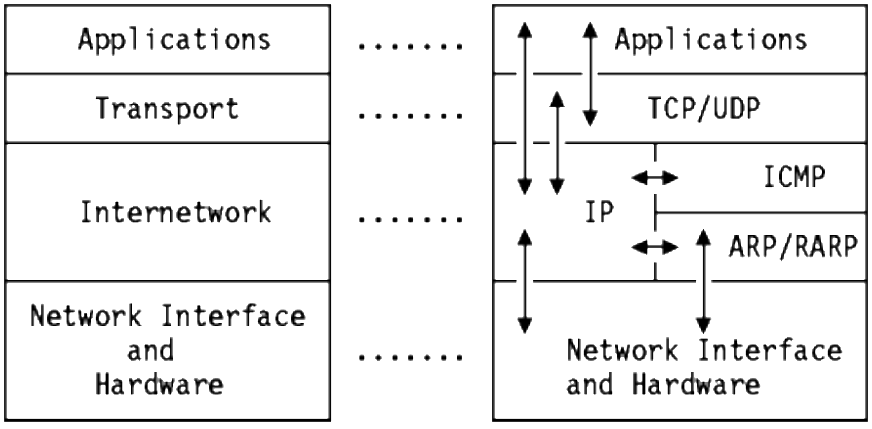
For example, here is a simple CRUD application using 3 tier architecture. If a user wants to upload a new file to the server, user will first need to interact with the presentation layer, which is the top-most layer in this diagram, the main function of this layer is to translate tasks and results to something the user can understand. And then user request will be handled by the middle layer which is called logic layer. It is used to coordinates the application layer and makes logical decisions by client's request, in this case upload a file into the database. Then it goes to data layer, which has control to the actual database and file system, to perform the database writing.

After data layer finished the upload process, normally it will return a value to logic tier to inform the operation is success or not. And in the end, the logic tier will return back to presentation layer and let the user know whether the request has been satisfied. Till now, the operation is finished.



(reference form: https://en.wikipedia.org/wiki/File:Overview_of_a_three-tier_application.png)

Another good example is internet protocol. As you can see, network protocol suite is generally being divided into five or seven layers. The data in application layer keep encapsulate into lower layer data packet and frames to be transmitted without error.



(reference form: http://mykeepit.blogspot.ca/2012/09/draw-and-explain-architecture-of-tcpip.html)

Advantage

Ease of development:
Developers don’t need full knowledge to the project. Different people have different skills. They can develop applications by separating skill sets by layers. (e.g. if you are good at front end, you can develop presentation layer. If you have skills at backend or database, you can develop business layer or database layer.)

Testability:
A developer can mock a presentation component or screen to isolate testing within a business component, as well as mock the business layer to test certain screen functionality.

Maintainability:
The layers of isolation concept mean that changes made in one layer of the architecture generally don't impact or affect components in other layers: the change is isolated to the components within that layer

Abstraction. Layered architecture abstracts the view of the system as whole while providing enough detail to understand the roles and responsibilities of individual layers and the relationship between them.

Encapsulation. No assumptions need to be made about data types, methods and properties, or implementation during design, as these features are not exposed at layer boundaries.

Clearly defined functional layers. The separation between functionality in each layer is clear. Upper layers such as the presentation layer send commands to lower layers, such as the business and data layers, and may react to events in these layers, allowing data to flow both up and down between the layers.

High cohesion. Well-defined responsibility boundaries for each layer, and ensuring that each layer contains functionality directly related to the tasks of that layer, will help to maximize cohesion within the layer.(business layer has the all logical application.

Low coupling. Communication between layers is based on abstraction and events to provide low coupling between layers.

Reusable. Lower layers have no dependencies on higher layers, potentially allowing them to be reusable in other scenarios.

Disadvantage

Performance Degraded

While it is true some layered architectures can perform well, the pattern does not lend itself to high-performance applications due to the inefficiencies of having to go through multiple layers of the architecture to fulfill a business request.

Not Easy to deploy

Depending on how you implement this pattern, deployment can become an issue, particularly for larger applications. One small change to a component can require a redeployment of the entire application (or a large portion of the application), resulting in deployments that need to be planned, scheduled, and executed during off-hours or on weekends. (Because the whole running applications have to stop to rebuild from 0) As such, this pattern does not easily lend itself toward a continuous delivery pipeline, further reducing the overall rating for deployment.

Low Scalability

Because of the trend toward tightly coupled and monolithic implementations of this pattern, applications build using this architecture pattern are generally difficult to scale. You can scale a layered architecture by splitting the layers into separate physical deployments or replicating the entire application into multiple nodes, but overall the granularity is too broad, making it expensive to scale.

Support/inhibit specific NFPs?

This architectural style is efficient, as it allows layers to work and this style is also scalable (you can add the new NFP layered in the architecture any where you need). It also inhibits dependability, if the new layered is broken, the application may be crash.