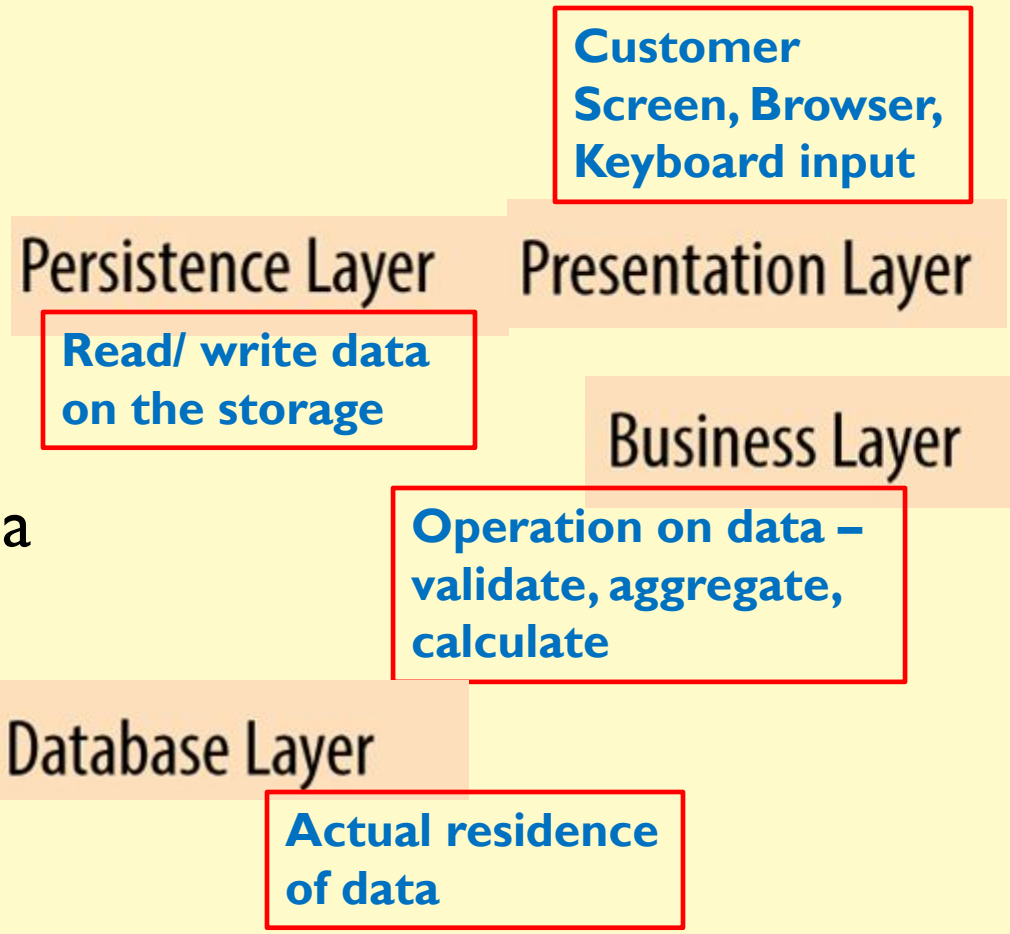
CSE 470 – Layered Architecture

BRAC University

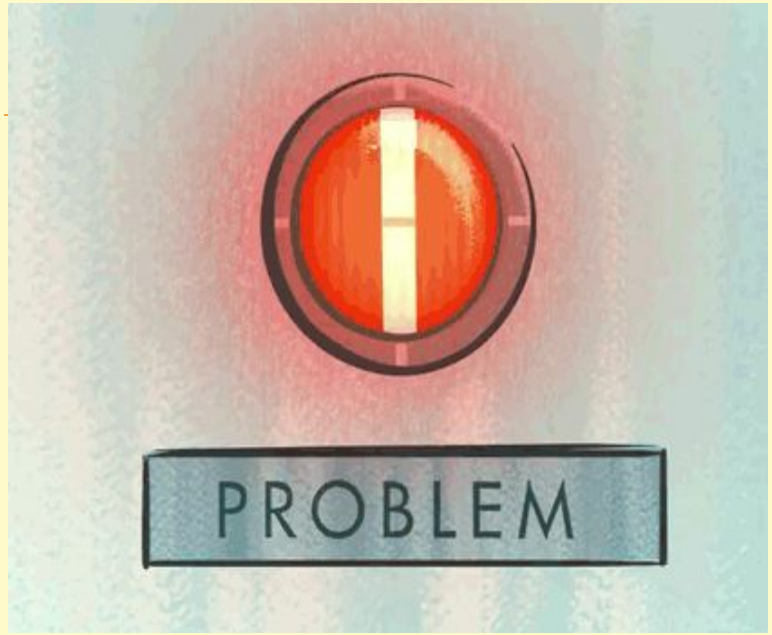
BRAC UNIVERSITY
Inspiring Excellence

Lets recall Monolithic Software

- The end product come at end of the process model
- There is no separation of concern (code) of different software components.
- All code may be written in a single file with html, sql queries, logic checking etc.



BRAC UNIVERSITY
Inspiring Excellence

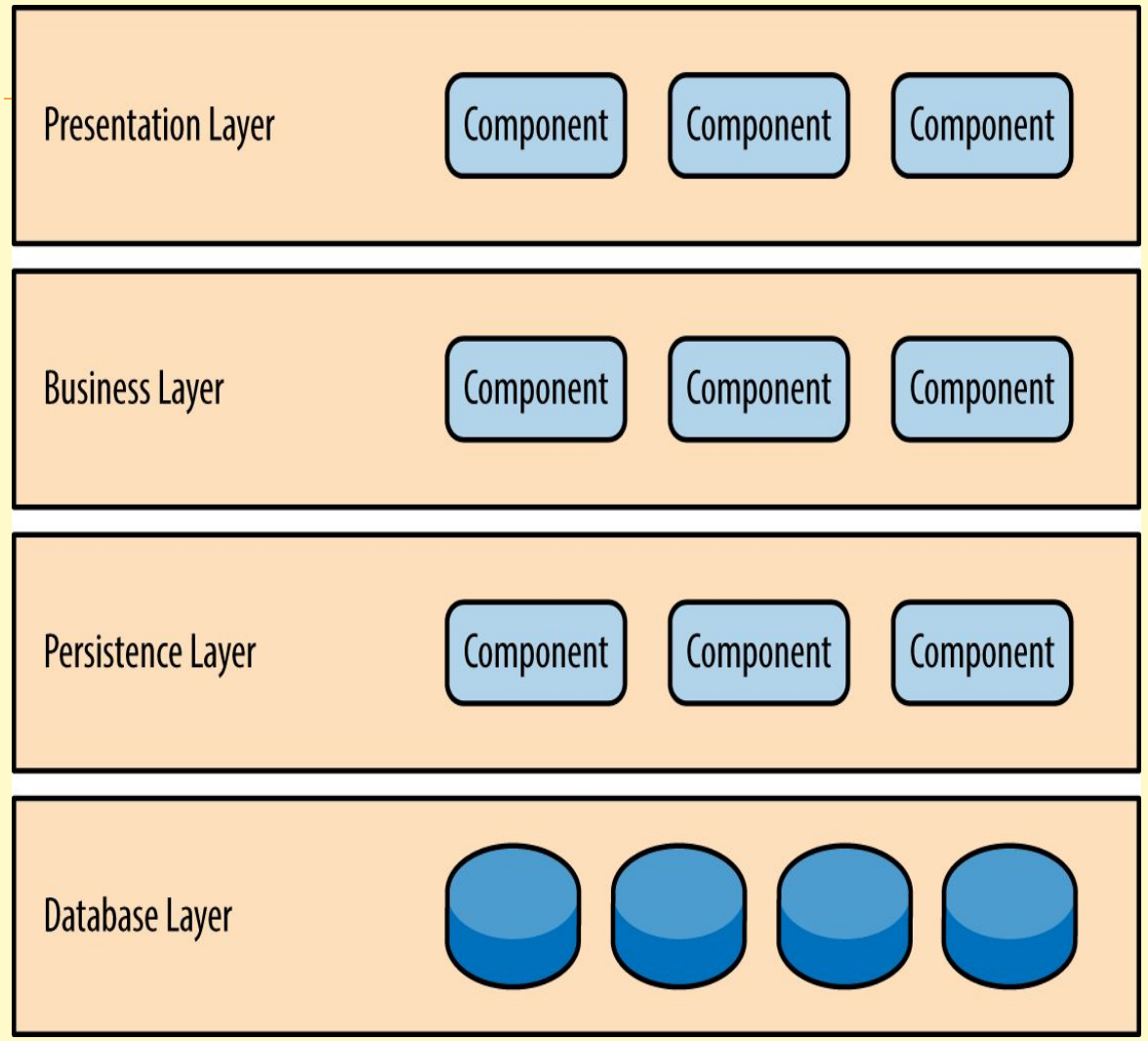


- No separation between components
- Changing a component affects other components. For example – What if I want to change the UI from JavaScript to Angular ?

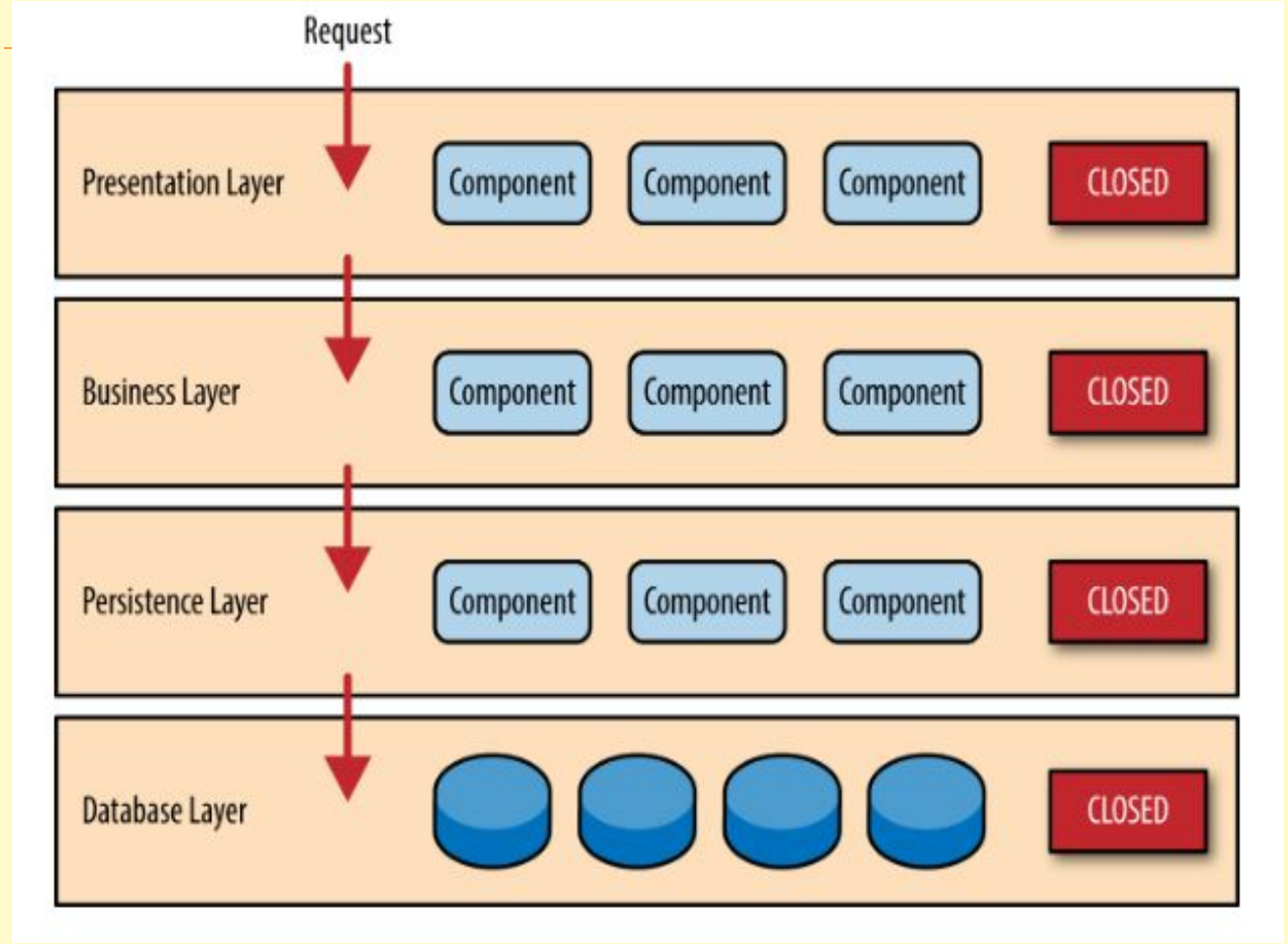
Layered Architecture

- Organises the system into a set of layers (or abstract machines) each of which provide a set of services.
- Supports the incremental development of sub-systems in different layers. When a layer interface changes, only the adjacent layer is affected.

- Organizes the system into layers with related functionality associated with each layer.
- A layer provides services to the layer above it so the lowest-level layers represent core services that are likely to be used throughout the system.

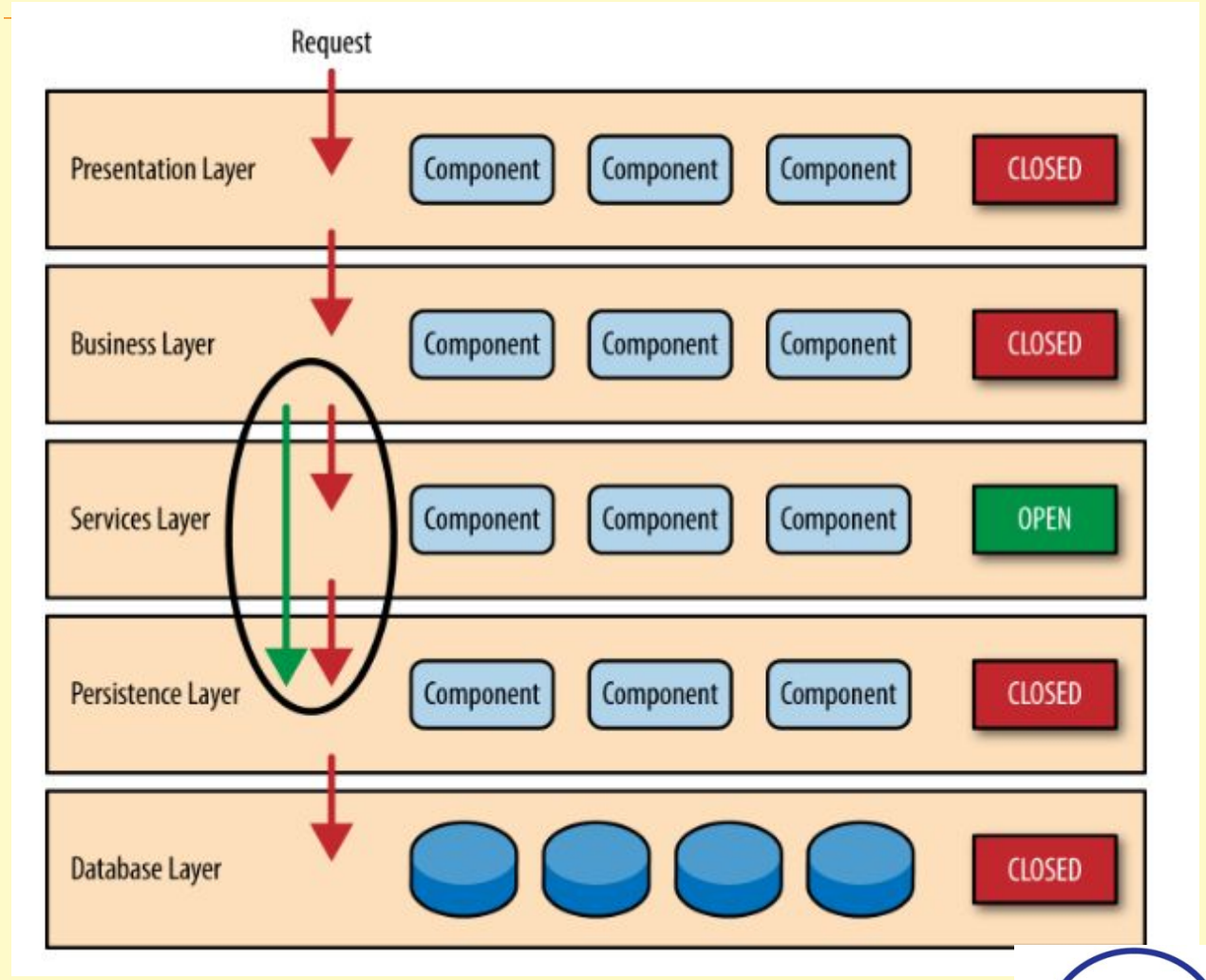


- Layers can be open or closed
- A closed layer can only be accessed by the layer above.
- A change in one layer does not affect others. It provides isolation.
- **However,** what if we want to add a new layer where shared utilities will be provided to be used by the **Business Layer.** But, we need to use it sometimes.

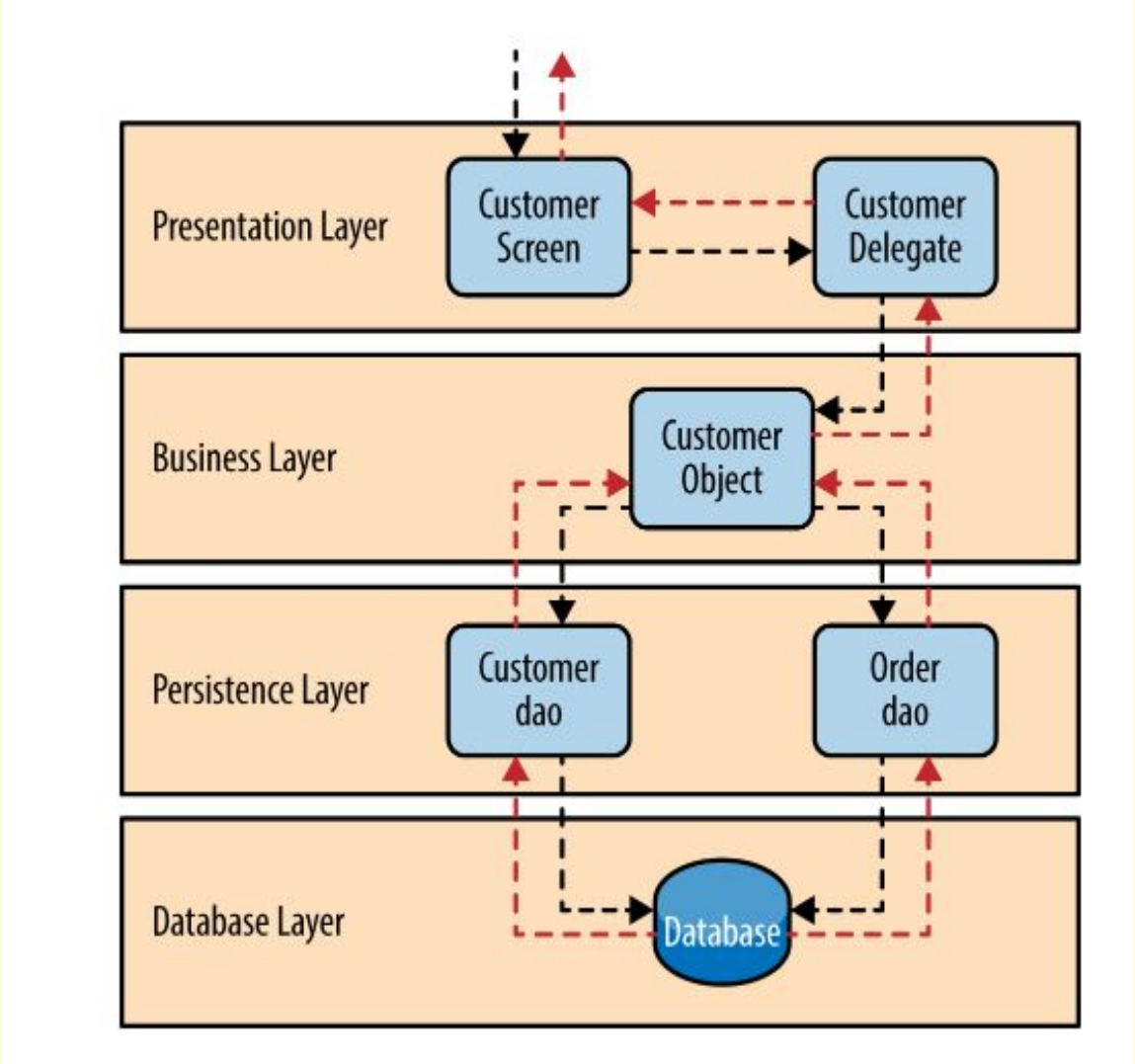


BRAC UNIVERSITY
Inspiring Excellence

- Here comes the concept of open layers.
- An open can be bypassed by upper layers.
- Too many open layers may affect the actual essence of layered architecture.



Example !!



When Used:

1. Used when building new facilities on top of existing systems
2. When the development is spread across several teams with each team responsible for a layer of functionality
3. When there is a requirement for multi-level security (for instance, any secured transaction based system)

Advantages:

1. Allows easy replacement or addition of entire layers so long as **the interface is maintained**.
2. Testing is easy as components are isolated

Disadvantages:

1. In practice, providing a clean separation between layers is often difficult and a high-level layer may have to interact directly with lower-level layers rather than through the layer immediately below it.
2. Performance can be a problem because of multiple levels of interpretation of a service request as it is processed at each layer.
3. Parallel processing is not possible (Sequential structure)
4. Example: OSI model, J2EE/Java 2 Enterprise Edition programming model