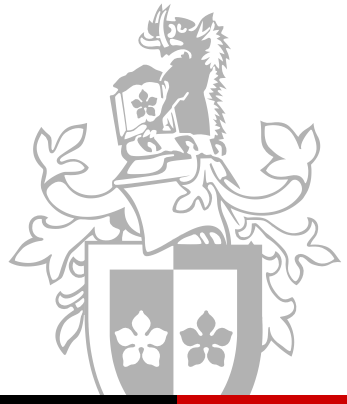
COS30041 Creating Secure and Scalable Software

Lecture 04a Business Logics 1



SWINBURNE UNIVERSITY OF TECHNOLOGY

Learning Objectives

- After studying the lecture material, you will be able to
  - ☐ Understand and describe ideas of having a business logic layer
  - ☐ Discuss issues of having a business logic layer

Pre-requisites

- Some concepts in Object Oriented Programming
- Some design concerns in OOP [coupling, cohesion, single-minded components]

Outline

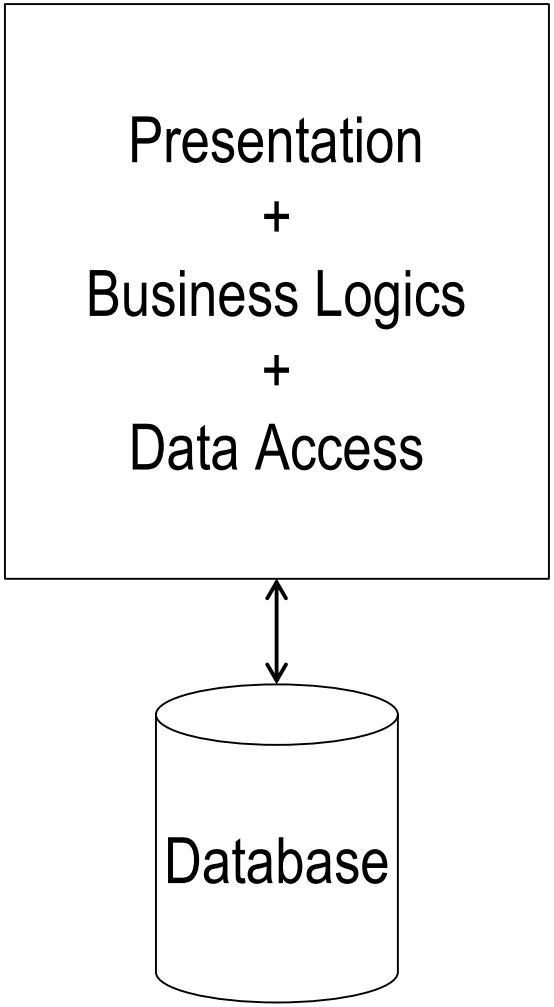
- Overview of architectural layers / tiers
- Business Logic Layer
- Stateless business object

Some Background (cont'd)

- Thick client – ALL in ONE client
  - Presentation, Business Logics, Data Access, Data
- Problem
  - Easily to mix them up
  - Difficult to uphold good design principles (coupling, cohesion, …)
  - Difficult to maintain
- One solution – put things into different layers
  - Supports MVC

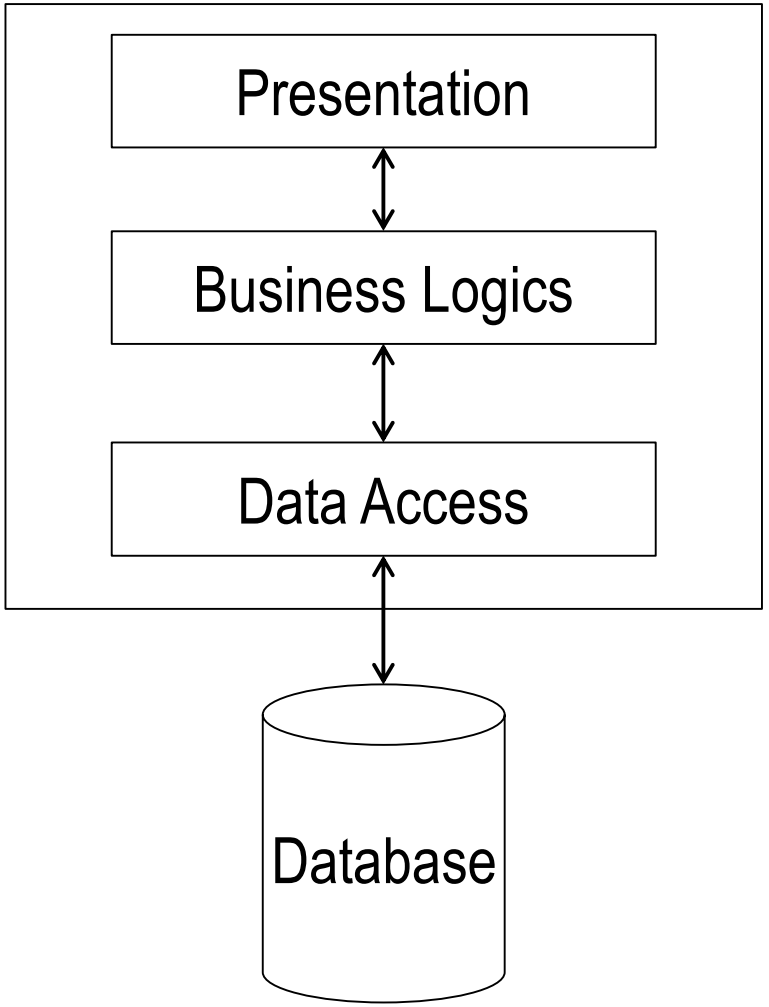
Some Background (cont'd)

Thick Client



Thick Client

(one machine, different layers)

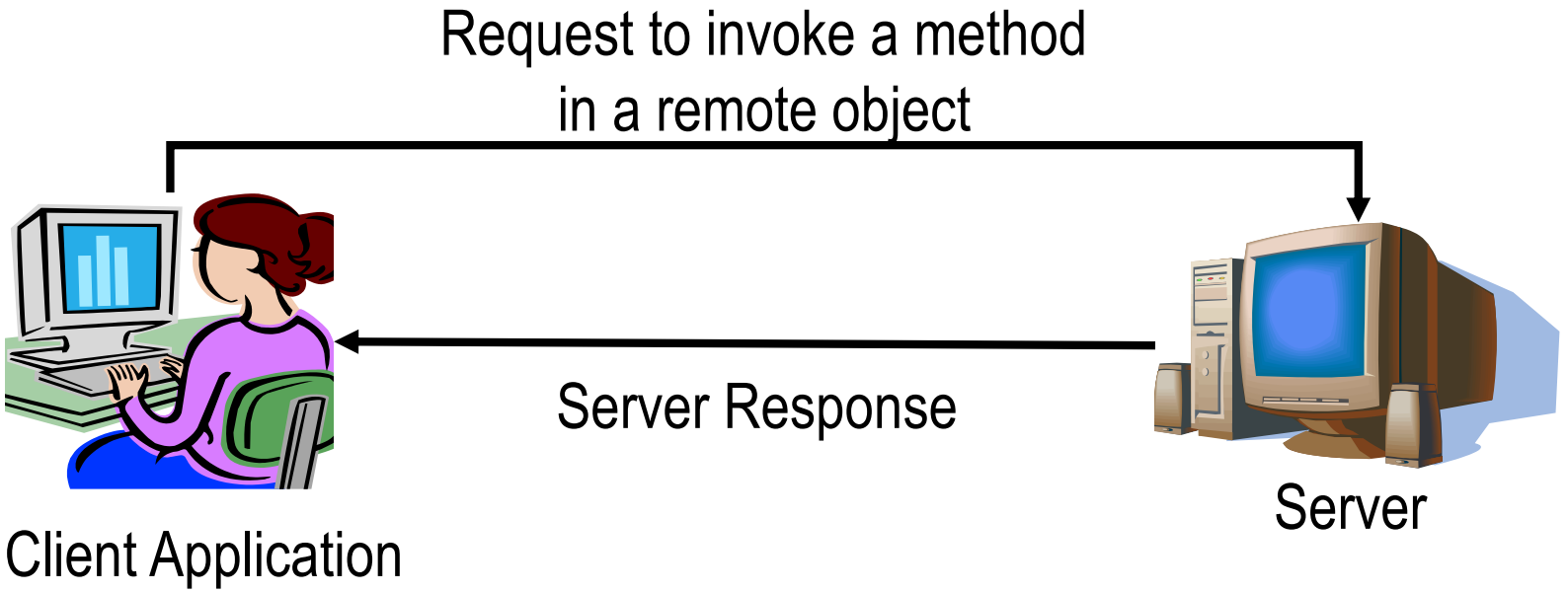


Some Background (cont'd)

- Distributed Computing
  - Allow remote access using remote procedural call (RPC)
  - Allow separation of concerns
- Thin client – Only presentation in client, rest on server**s**
  - Business logic – Business Logic Layer
  - Data Access – Data Access Layer
  - Data Persistence – Database Layer
  - Supports MVC naturally

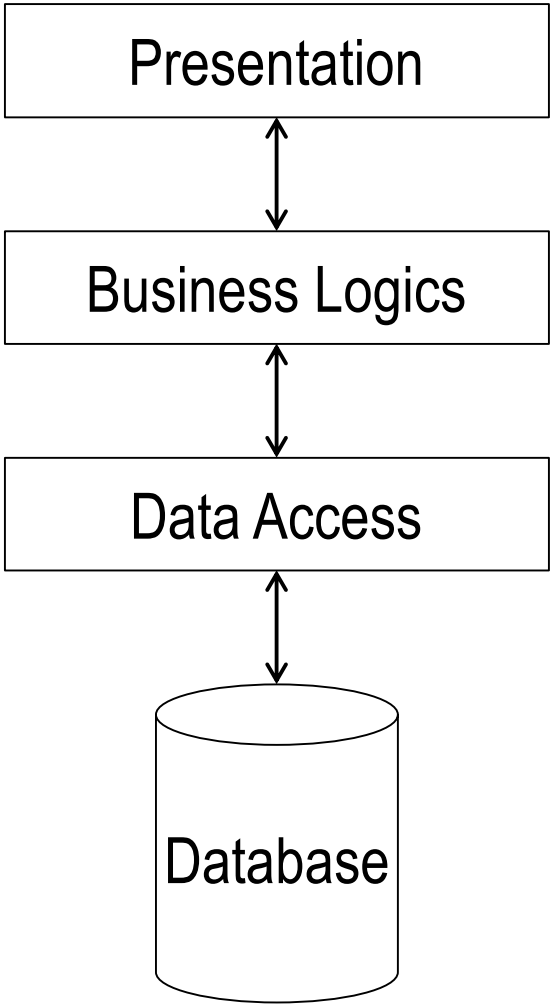
Remote Procedure Call, RPC

- A procedural invocation from a process on one machine (client) to a process on another machine (server)
- Similar technology – Remote Method Invocation (RMI)



Architectural Layers / Tiers

Thin Client (different machines)



Note: There are subtle differences between layer and tier.
However, for the time being, let us treat them as the same.

Why different layers?

Adv

- Each layer has its own responsibility
- Promote good design principles
- Easier to maintain
- Scale easily

Disadv

- Need good communications to make it work
- More difficult to program
- More complicated to deploy

Business Logic Layer, BLL

- The collection of classes that handle business rules of an application
- Classes in BLL
  - interact with DAL classes to determine whether the business rules are followed
  - act as the controller of the business application

Business Object

- An object / class that models the business processes
  - Business logic
  - Business rules
  - Algorithms
  - Workflow
- Example
  - accessing bank account
  - verifying credit card details
  - preparing an invoice

Classes in BLL

- Session Bean (a type of Enterprise JavaBean)
  - StateLess Session Bean, SLSB
  - StateFul Sesion Bean, SFSB
  - Singleton Session Bean, SSB

Lifetime of a Business Object

- The lifetime of a session or the lifetime of the client code that is calling the session object
  - The time of a browser window is open
  - The time of your web page is running
  - A standalone client application is open
  - Another business object using your business object
- The enterprise server will destroy business objects if clients time out

General Issues

- A Business object cannot be shared between clients
- Business objects do not represent data in a database

Different types of Business Objects

- Stateless business object
- Stateful business object

Stateless Business Object

- A business object that is designed to service business processes that span ***only one*** method request (from client) or transaction
- Example: Banking system
  - Transfer money from one account to another
- Example: Credit Card Payment
  - Payment via Credit Card

References

- Java EE Tutorial