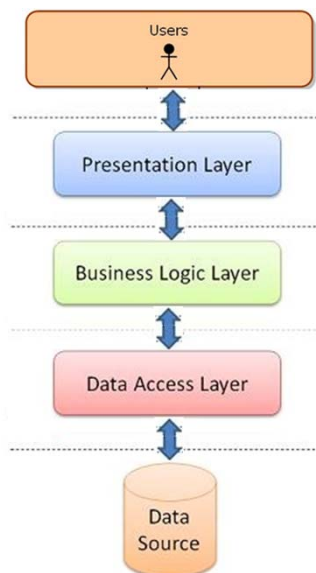


Xây dựng phần mềm

Mô hình nhiều tầng (n tiers)

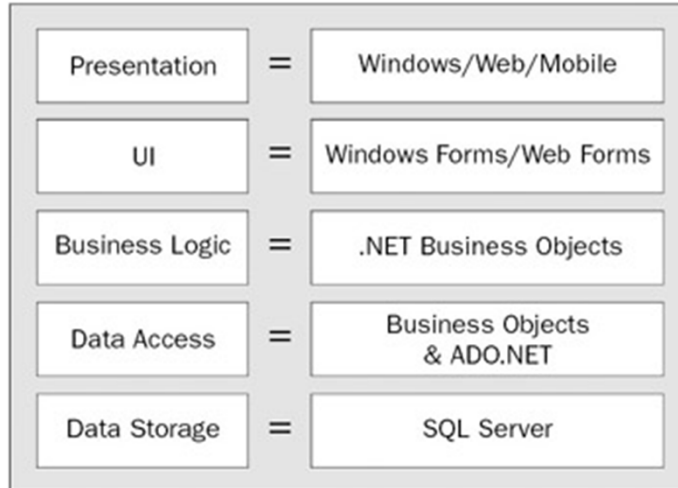
Th.S Cao Thái Phương Thanh
caothaiphuongthanh@gmail.com

3-layer model



3-layer model

Logic model → design classes

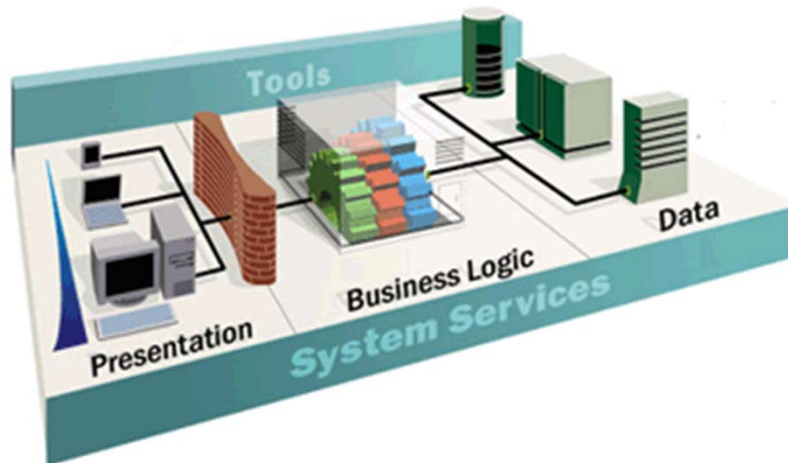


Layers and Tiers

- ✦ Logical Separation
- and
- ✦ Physical Separation

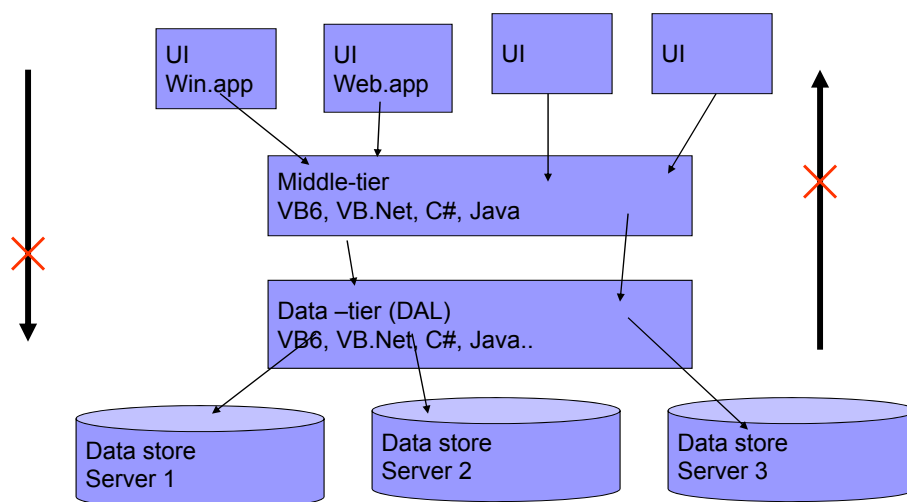
Tiers

✦ Physical Separation



Software architecture tiers

✦ Software architecture in logic and physic

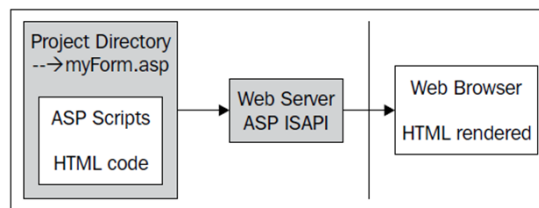


n-Tier m-Layer Software Architecture

- ✦ This lesson uses Microsoft ASP.NET project to illustrate the tier, layer software architecture.
- ✦ By default, web applications have a build-in 3-tier architecture:
 - The client-side browser : the presentation tier
 - The code : the application tier
 - The physical database: the data tier
- We focus on the application tier (the code architecture)

1-Tier 1-Layer

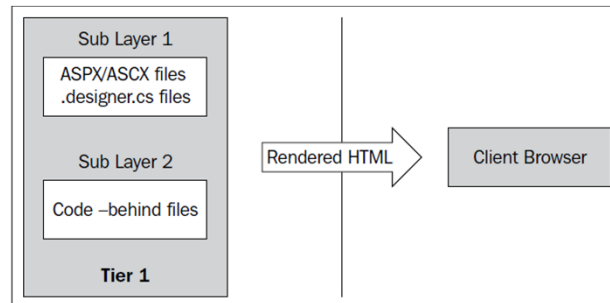
- ✦ The classic ASP (or PHP) style: inline coding



- .NET Project (ASPX project) compiling into a DLL in the /bin folder and under a single namespace: MyApp
- No. of project files: 1
- No. of namespaces: 1
- Example (notice the asp:Repeater control)

1-Tier 1-Layer

✦ ASP.NET: code behind model

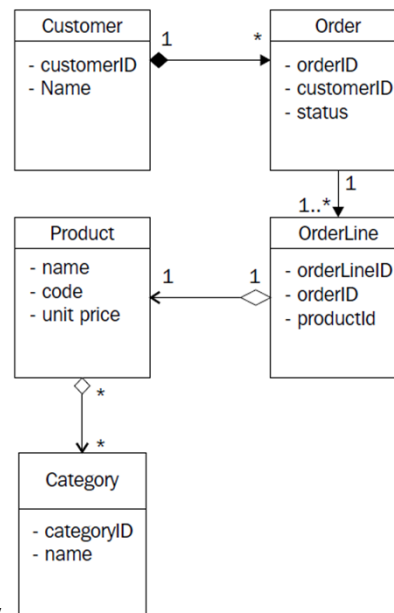


- We still have one single physical DLL, but the UI code itself is logically separated into two layers — one in the markup code and the other in the code-behind file

→ Example

Example

✦ Another sample database:



1-Tier 2-Layer

+ 2 layers:

- UI-layer with ASPX and code-behind classes
- Data access classes under a different namespace and in the folder separate Code
- All files are in the same project

→ Example

Notice:

- + Web.config: connection string
- + ASPX data source control: Quick but less control over data access.

1-Tier 3-Layer

+ 3 layers:

- UI-layer with ASPX and code-behind classes
- Business classes under a different namespace and folder (BL) but in the same project
- Data access classes under a different namespace and folder (DAL) but in the same project

→ Example

Notice:

- SQLHelper: utility class
- Code 3 lớp: tự điều khiển (showall.aspx) → cách code
- ObjectDataSource (datasource-customers.aspx) : cách dùng ?

2-Tier 3-Layer

The solution will have:

- + ASP.NET Web Project having Presentation Layer coding in ASPX and ASCX files, under the namespace, MyApp.Web (Tier 1)
- + A class library project having two folders (Tier 2):
 - Business: for business logic code, with namespace MyApp.Code.Business (Layer 1)
 - DAL: for data access code, with namespace MyApp.Code.DAL (Layer 2)

→ Example

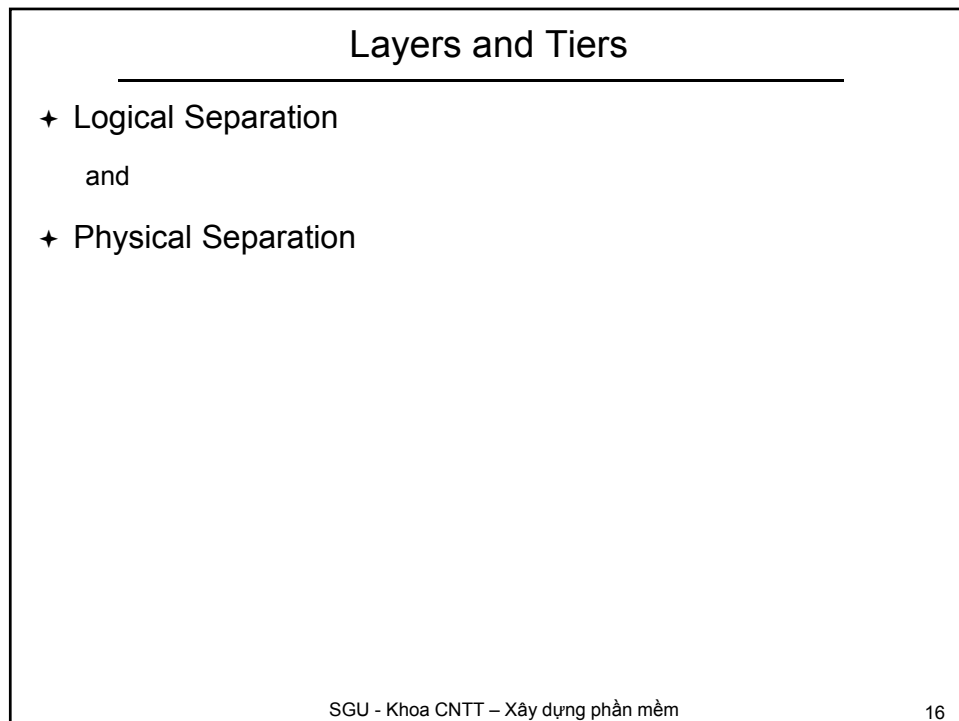
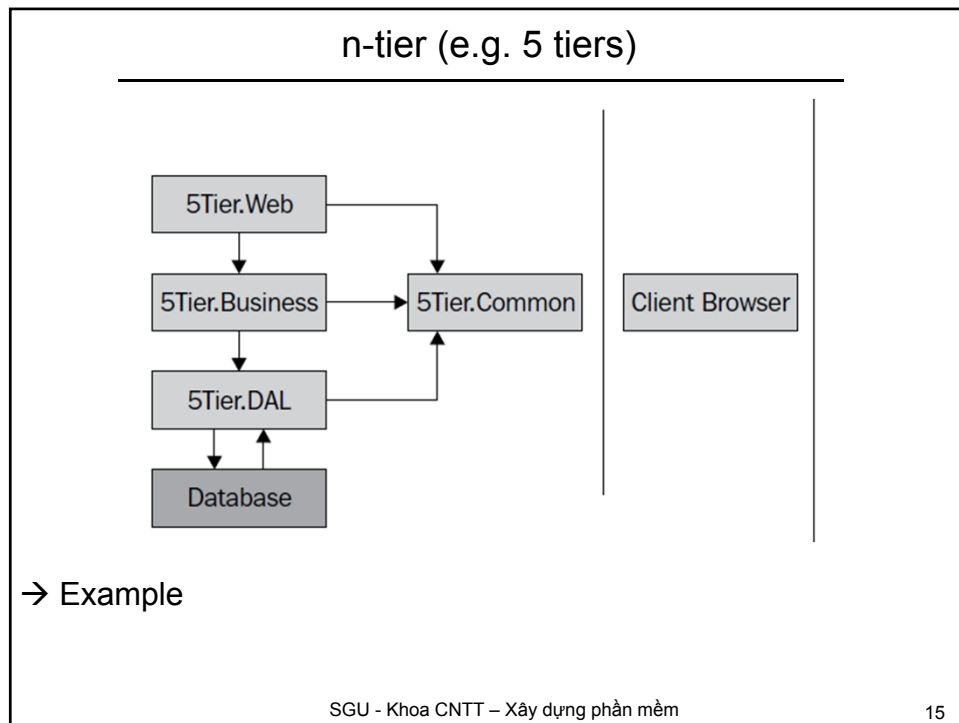
Notice: add reference

3-Tier 3-Layer

The solution will have:

- + ASP.NET Web Project having Presentation Layer coding in ASPX and ASCX files, under namespace MyApp.Web (Tier 1)
- + A class library project having business logic code, with namespace, MyApp.Code.Business (Tier 2)
- + A class library project DAL for data access code, with namespace, MyApp.Code.DAL (Tier 3)

→ Problem ?



n-Layer software architecture

- ✦ Tư tưởng chính của mô hình phân lớp: chia để trị
- ✦ Cách code phân lớp không phải là cách code duy nhất
 - Cũng không phải là cách code luôn đúng
 - Nhưng chắc chắn là một cách code **tốt**
 - ...

Bài tập lớn

- ✦ Các chức năng chính của bài tập:
 1. Giới thiệu, tìm kiếm, quản lý sản phẩm
 2. Khách hàng đặt hàng online
 3. Xử lý đơn hàng đã đặt
 4. Nhập kho sản phẩm
 5. Báo cáo thống kê