

KIẾN TRÚC 3 TẦNG

THE THREE-LAYER ARCHITECTURE

Nguyễn Hoàng Anh

nhanh@fit.hcmuns.edu.vn

Nội dung trình bày

1 Generics

2 Kiến trúc 3 tầng

3 Demo



Generics



None Generics

```
public class Main {  
  
    public static void main(String[] args) {  
        // TODO code application logic here  
        List list=new LinkedList();  
        list.add("0312143");  
        list.add("0312234");  
        list.add("0312532");  
  
        String first=(String)list.get(0);  
    }  
}
```



None Generics

```
public class EmployeeDAO {  
  
    public static ArrayList selectEmployeeAll() {  
        ArrayList arr = new ArrayList();  
        // Add EmployeeDTOs to arr  
        return arr;  
    }  
    public static void main(String[] args) {  
        ArrayList arr = EmployeeDAO.selectEmployeeAll();  
        for (int i = 0; i < arr.size(); i++) {  
            EmployeeDTO emp = (EmployeeDTO) arr.get(i);  
            System.out.println(emp.toString());  
        }  
    }  
}
```



None Generics

```
ArrayList arr = new ArrayList();
arr.add("Java");
```

Compile Time:BUILD SUCCESSFUL

```
//.....
```

```
EmployeeDTO emp= (EmployeeDTO) arr.get(0);
```

Runtime: Error

Generics

```
public class Main {  
  
    public static void main(String[] args) {  
        // TODO code application logic here  
        List<String> list=new LinkedList();  
        list.add("0312143");  
        list.add("0312234");  
        list.add("0312532");  
  
        String first=list.get(0);  
    }  
}
```



Generics

```
public class EmployeeDAO {  
  
    public static ArrayList<EmployeeDTO> selectEmployeeAll() {  
        ArrayList<EmployeeDTO> arr = new ArrayList<EmployeeDTO>();  
        //Add EmployeeDTOs to arr  
        return arr;  
    }  
    public static void main(String[] args) {  
        ArrayList arr = EmployeeDAO.selectEmployeeAll();  
        for (int i = 0; i < arr.size(); i++) {  
  
            System.out.println(arr.get(i).toString());  
        }  
    }  
}
```



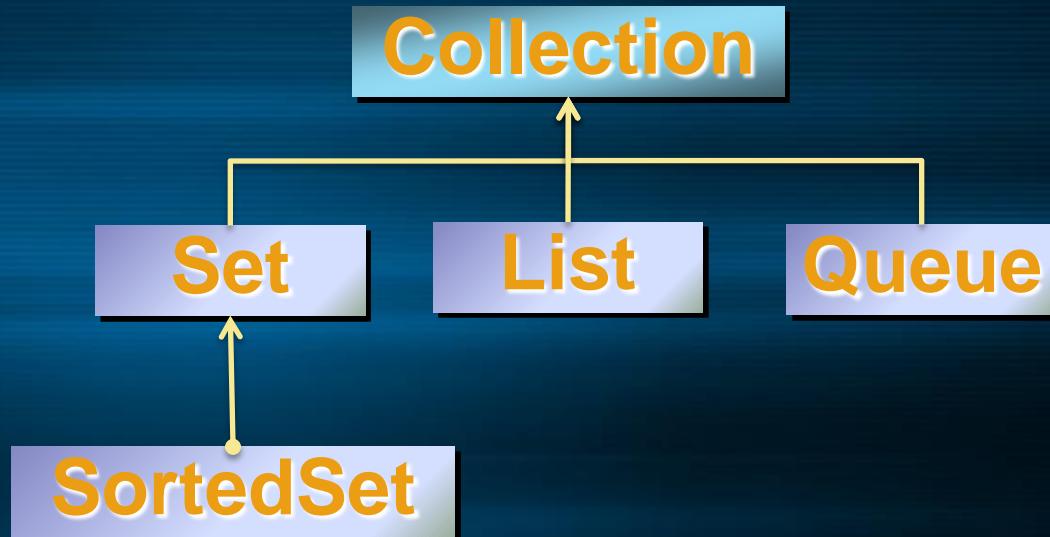
Generics

```
public class EmployeeDAO {  
  
    public static ArrayList<EmployeeDTO> selectEmployeeAll() {  
        ArrayList<EmployeeDTO> arr = new ArrayList<EmployeeDTO>();  
        arr.add("0312143");  
        return arr;  
    }  
    public static void main(String[] args) {  
        ArrayList arr = EmployeeDAO.selectEmployeeAll();  
        for (int i = 0; i < arr.size(); i++) {  
            System.out.println(arr.get(i).toString());  
        }  
    }  
}
```

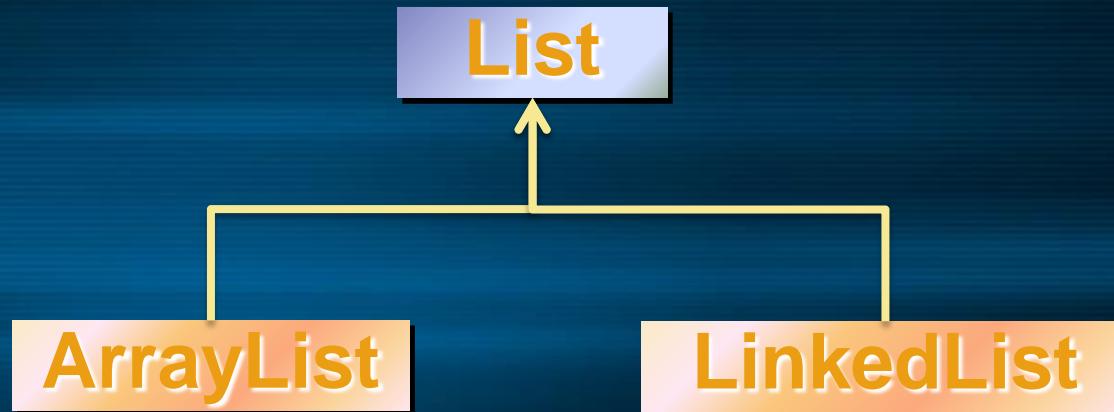
Compile Time: BUILD FAILED

Runtime: SAFE

Generics – Collection Interfaces



Generics – ArrayList, LinkedList



Kiến trúc 3 tầng – The Three-Layer Architecture



Hướng tiếp cận phân tầng trong phần mềm

1

Chia phần mềm thành các tầng (layer)

2

Mỗi tầng có vai trò đặc trưng riêng

3

Quyết định sự tương tác giữa các tầng như thế nào.

Vai trò của hướng tiếp cận phân tầng trong phần mềm

1

Sẵn sàng cho việc mở rộng hệ thống

2

Thuận tiện cho việc dùng lại

3

Hệ thống dễ bảo trì

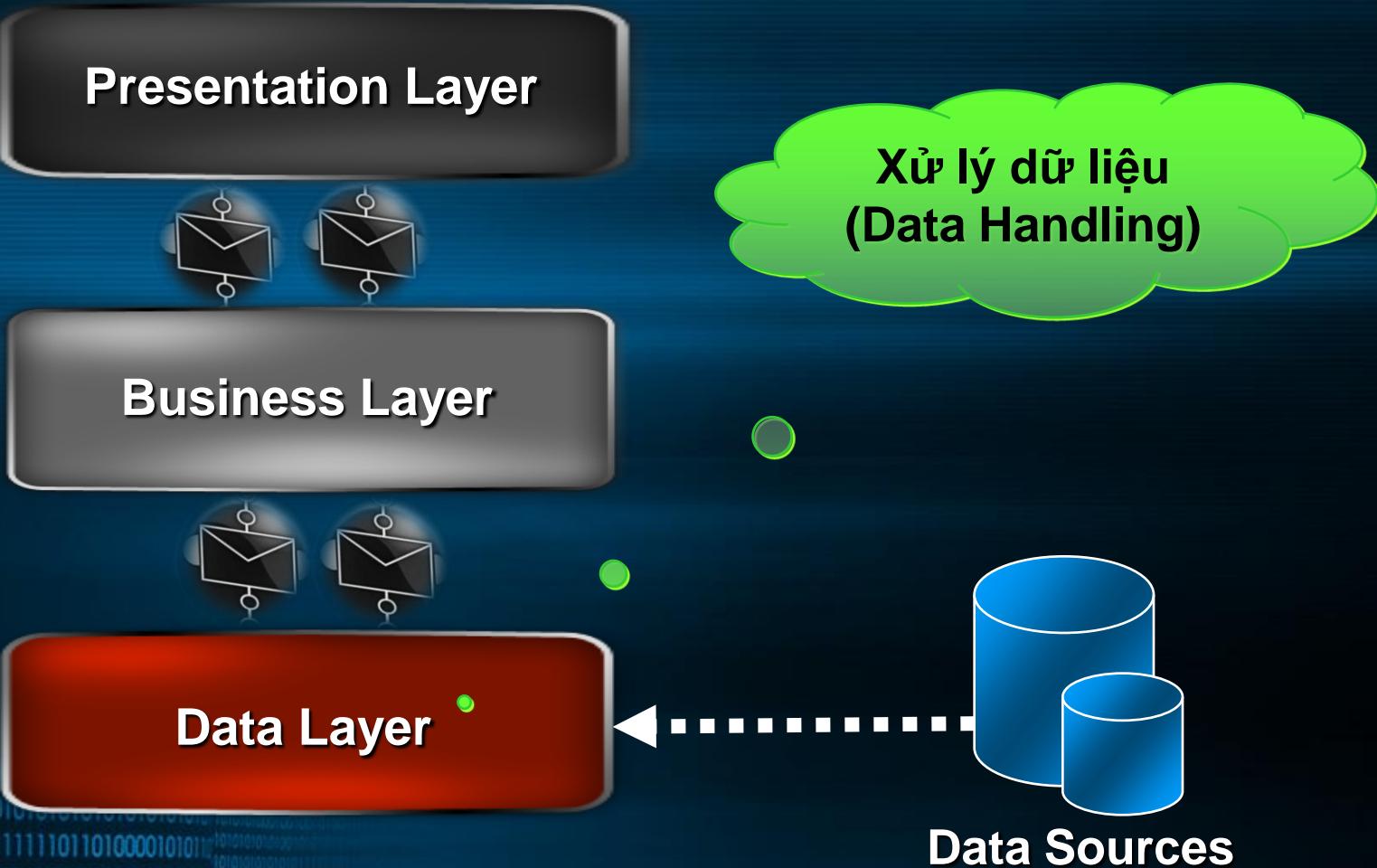
4

Dễ dàng phân chia công việc

Kiến trúc 3 tầng (The Three-Layer Architecture)



Data Layer



Tầng dữ liệu – Data Layer

Presentation Layer



Business Layer



Data Layer

Xử lý dữ liệu
(Data Handling)

Inserting Data

Updating Data

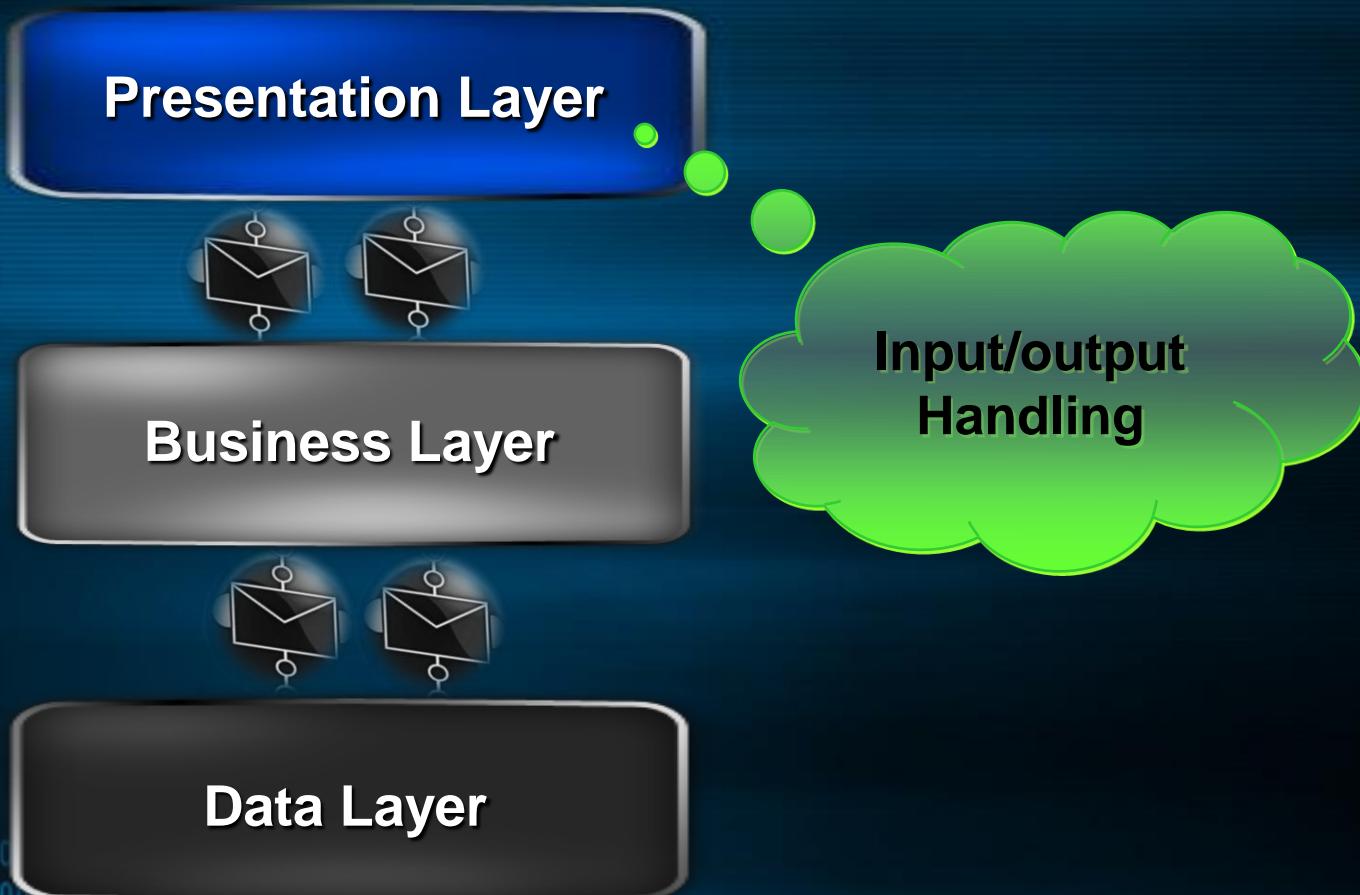
Deleting Data

Retrieving Data

Vai trò của Business Layer



Kiến trúc 3 tầng (The Three-Layer Architecture)



Kiến trúc 3 tầng (The Three-Layer Architecture)



Đối tượng trao đổi
dữ liệu
Data Transfer Object

Tầng dữ liệu – Data Layer



Tầng dữ liệu – Data Layer

Presentation Layer



Business Layer



Data Layer

Xử lý dữ liệu
(Data Handling)

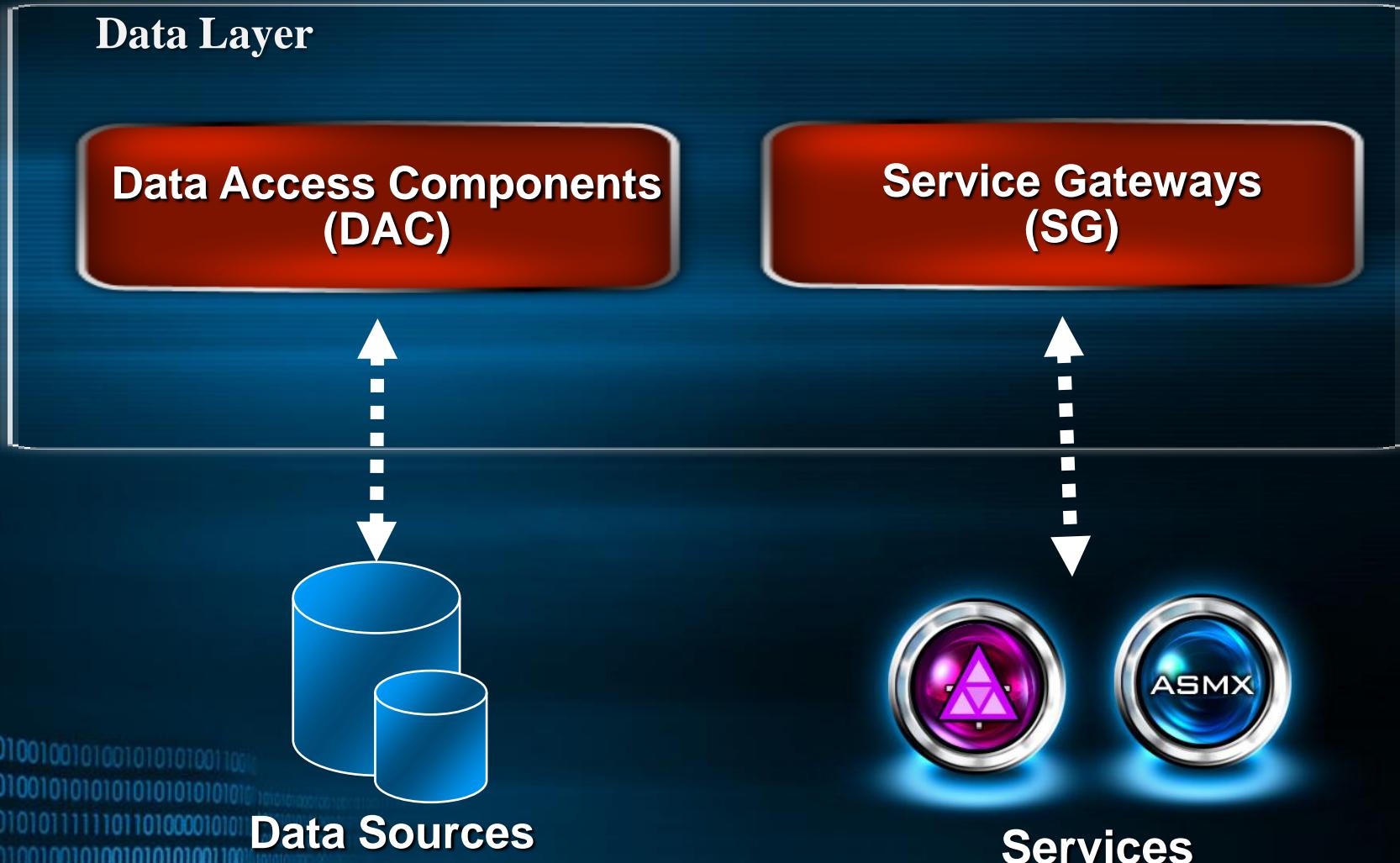
Inserting Data

Updating Data

Deleting Data

Retrieving Data

Tầng dữ liệu – Data Layer

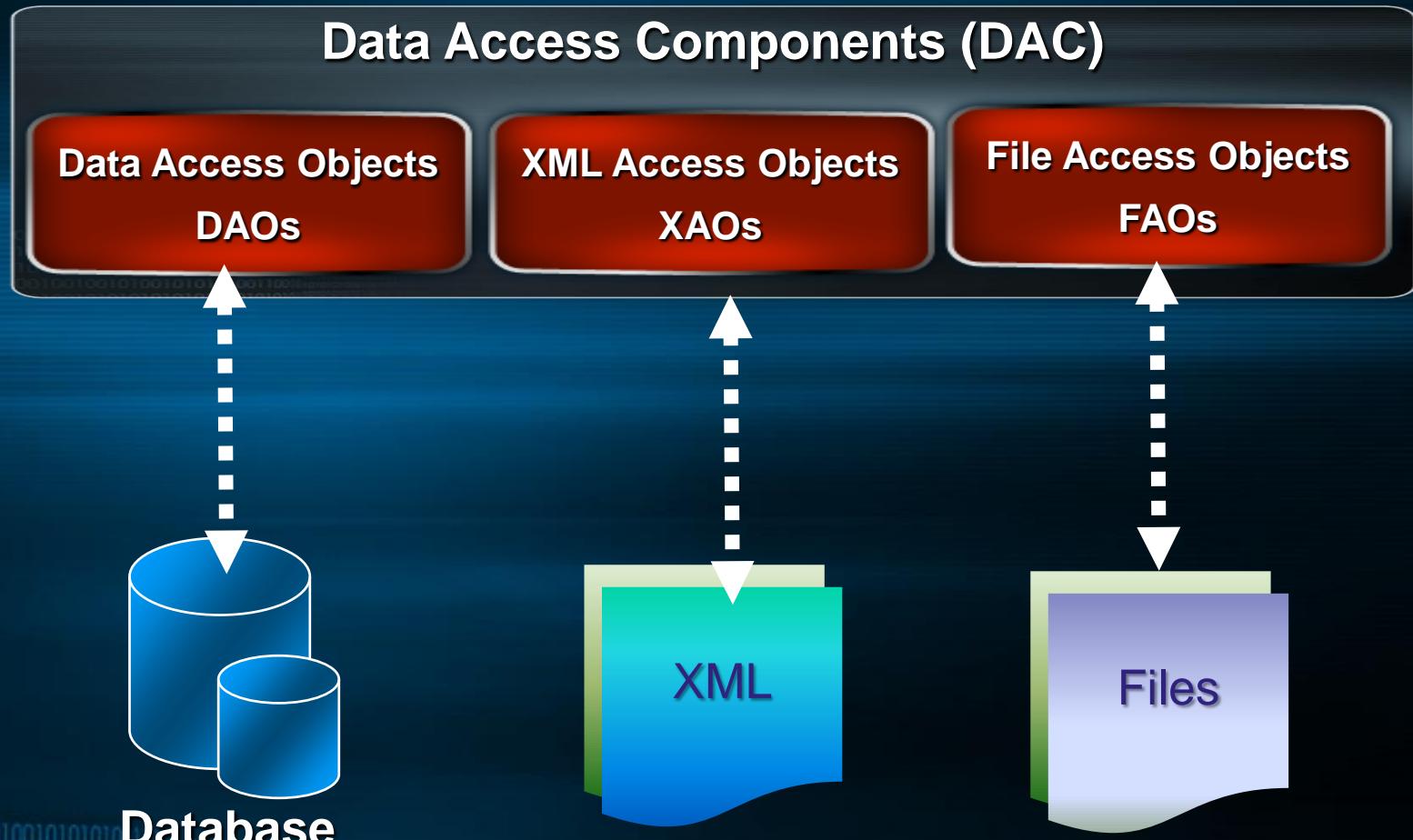


Tầng dữ liệu – Data Layer

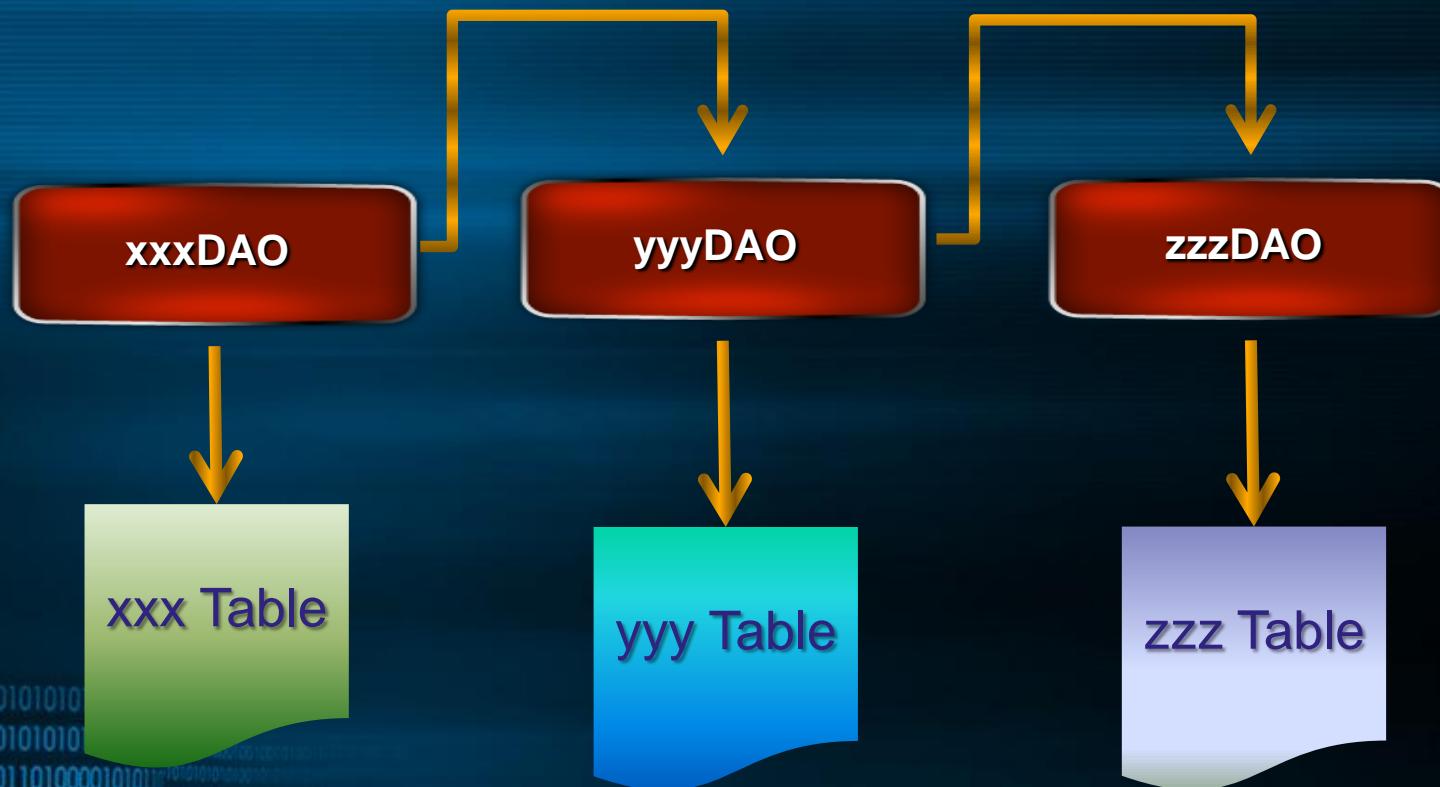
Data Access Components (DAC): là thành phần chính chịu trách nhiệm lưu trữ và truy xuất từ các nguồn dữ liệu (Data Sources).

Service Gateways (SG): là những thành phần trợ giúp việc truy xuất các dịch vụ bên ngoài một cách dễ dàng.

Tầng dữ liệu - Data Layer



Thiết kế Tầng dữ liệu với mẫu đơn giản



Đối tượng kết xuất của Data Layer

Presentation Layer



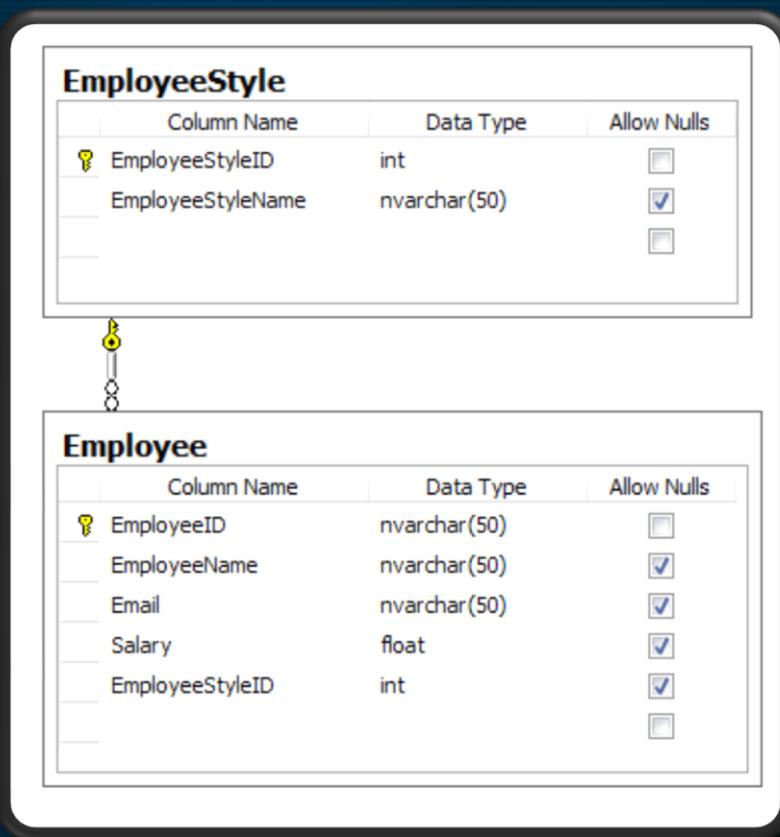
Business Layer



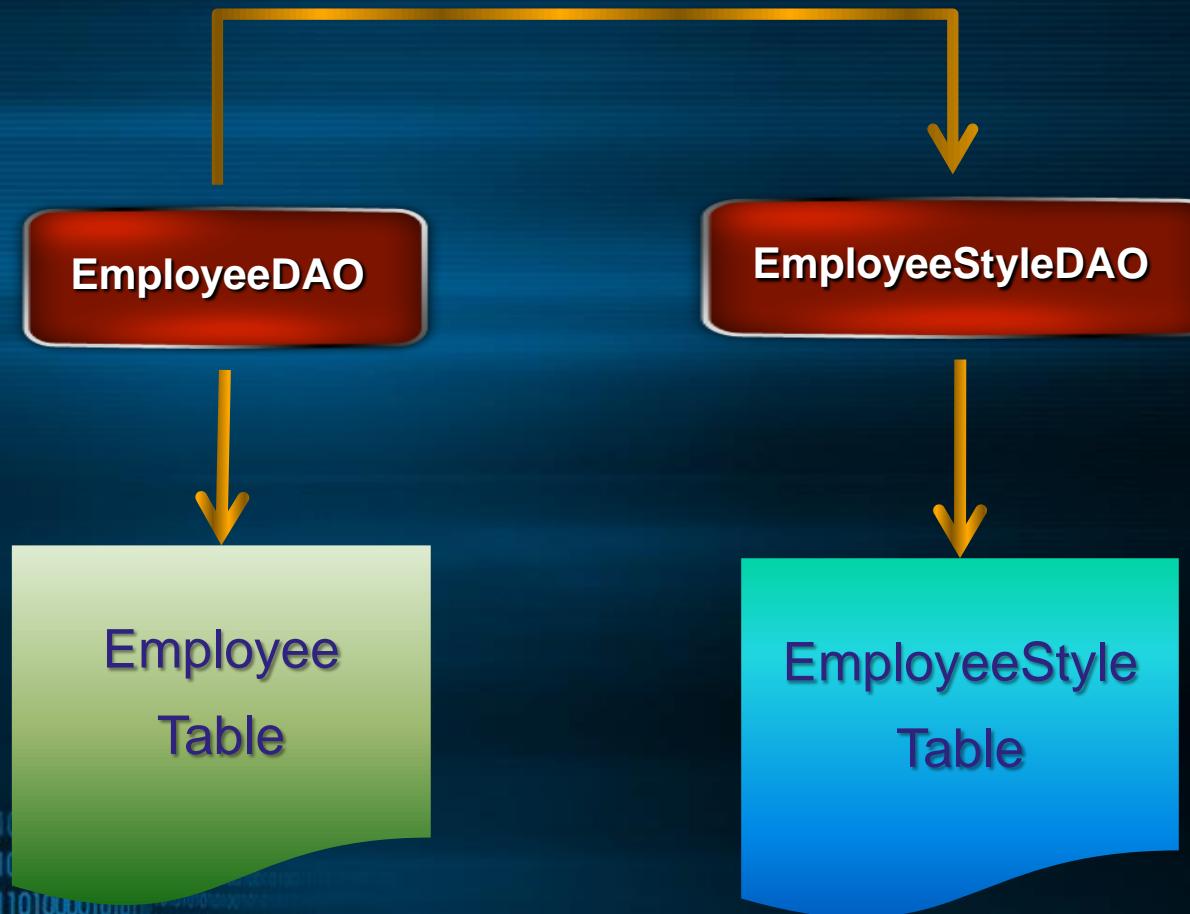
Data Layer

Đối tượng trao đổi
dữ liệu
Data Transfer Object

Data Layer – HRM



Data Layer – HRM - Java



Data Layer – HRM - Java

```
/**  
 *  
 * @author Nguyen Hoang Anh  
 */  
  
public class EmployeeDAO {  
    //Inserting  
    public static boolean insertEmployee(EmployeeDTO emp) { ... }  
    //Deleting  
    public static boolean deleteEmployee(String employeeID) { ... }  
    //Updating  
    public static boolean updateEmployee(EmployeeDTO emp) { ... }  
    //Retrieving  
    public static ArrayList<EmployeeDTO> selectEmployeeAll() { ... }  
  
    public static EmployeeDTO selectEmployeeByID(String employeeID) { ... }  
  
    public static ArrayList<EmployeeDTO> selectEmployeeByEmployeeStyleID(int employeeStyleID) { ... }  
  
    public static boolean checkEmployeeByID(String employeeID) { ... }  
  
    public static ArrayList<EmployeeDTO> searchEmployeeByName(String employeeName) { ... }  
}
```

Data Layer – HRM - Java

```
/*
 *
 * @author Nguyen Hoang Anh
 */

public class EmployeeStyleDAO {
    //Inserting
    //Updating
    //Deleting
    //Retrieving
    public static ArrayList<EmployeeStyleDTO> selectEmployeeStyleAll() {...}

    public static boolean checkEmployeeStyleID(int employeeStyleID) {...}
}
```

Data Tier



Thiết kế tầng nghiệp vụ – Business Layer



Tầng nghiệp vụ - Business Layer

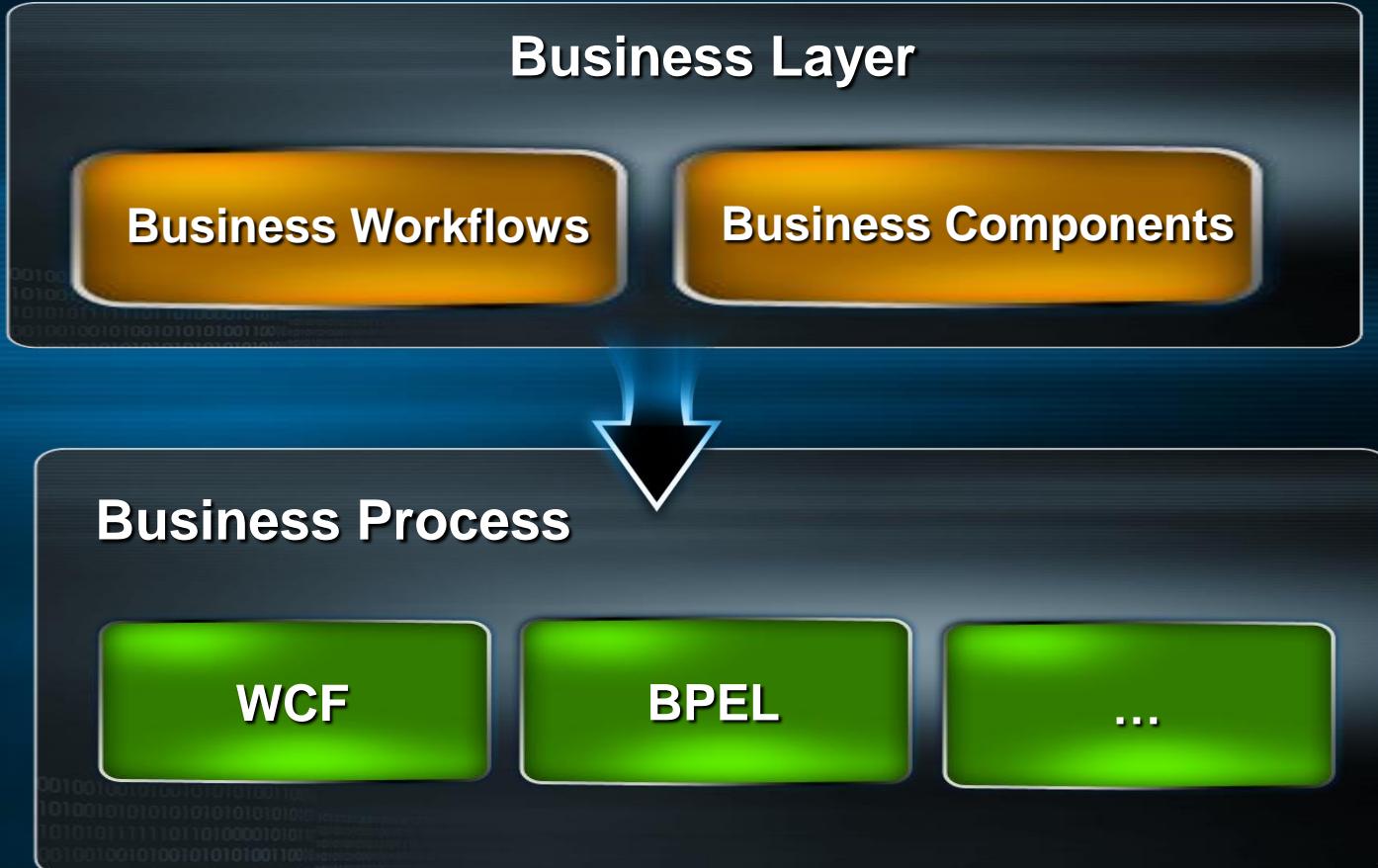


Tầng nghiệp vụ - Business Layer

Tầng này thực hiện các nghiệp vụ chính của hệ thống, sử dụng các dịch vụ do tầng Data cung cấp, và cung cấp các dịch vụ cho tầng Presentation.

Tầng này có thể sử dụng các dịch vụ của các nhà cung cấp thứ 3 để thực hiện công việc của mình. Ví dụ: sử dụng dịch vụ của cổng thanh toán trực tuyến như Paypal.

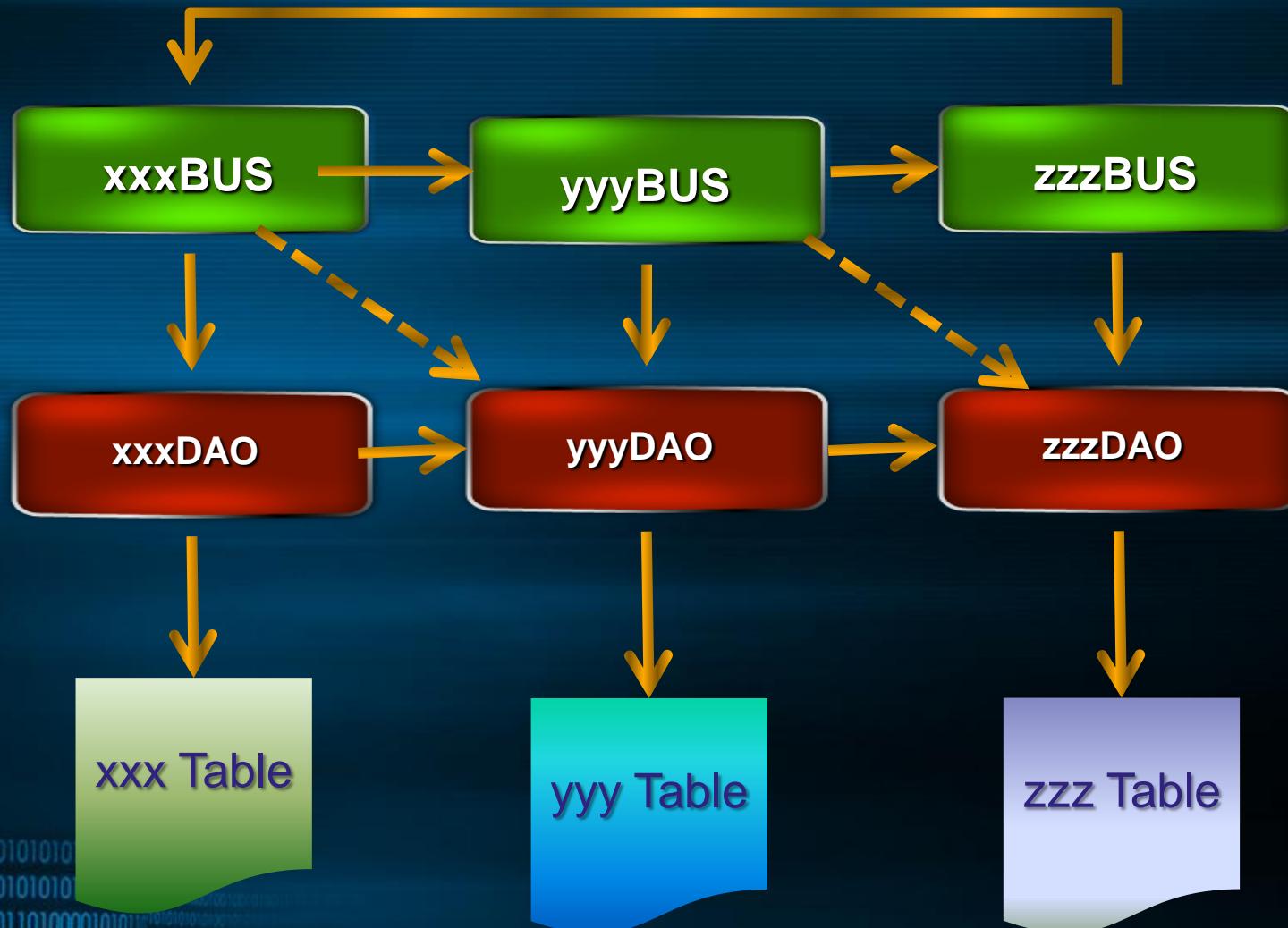
Business Layer – Tầng nghiệp vụ



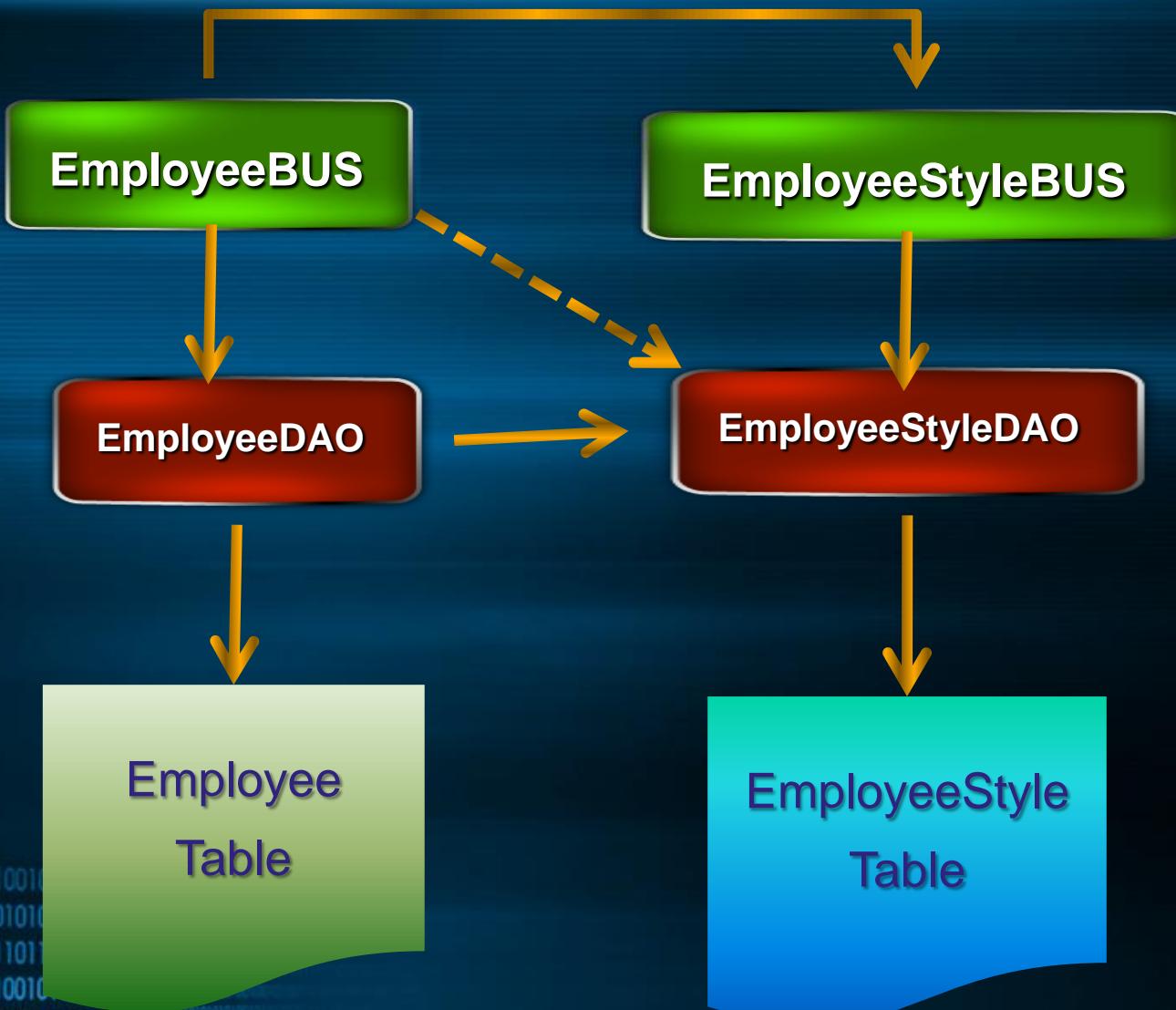
Business Layer – Tầng nghiệp vụ

Business Components (BC): là những thành phần chính thực hiện các dịch vụ mà Service Interface cung cấp, chịu trách nhiệm kiểm tra các ràng buộc logic (constraints), các quy tắc nghiệp vụ (business rules), sử dụng các dịch bên ngoài khác để thực hiện các yêu cầu của ứng dụng.

Thiết kế Tầng nghiệp vụ với mẫu đơn giản



Business Layer – HRM - Java



Business Layer – HRM - Java

```
public class EmployeeBUS {

    public static boolean insertEmployee(EmployeeDTO emp) {
        boolean result = false;
        if (EmployeeDAO.checkEmployeeByID(emp.getEmployeeID())) {
            result = false;
        } else {
            result = EmployeeDAO.insertEmployee(emp);
        }
        return result;
    }

    public static boolean deleteEmployee(String employeeID) { ... }

    public static boolean updateEmployee(EmployeeDTO emp) { ... }

    public static ArrayList<EmployeeDTO> selectEmployeeByEmployeeStyleID(int employeeStyleID) {
        if (EmployeeStyleDAO.checkEmployeeStyleID(employeeStyleID)) {
            return EmployeeDAO.selectEmployeeByEmployeeStyleID(employeeStyleID);
        } else {
            return new ArrayList<EmployeeDTO>();
        }
    }

    public static ArrayList<EmployeeDTO> selectEmployeeAll() { ... }

    public static ArrayList<EmployeeDTO> searchEmployeeByName(String employeeName) { ... }
}
```

Business Layer – HRM - Java

```
package BUS;

import DAO.EmployeeStyleDAO;
import DTO.EmployeeStyleDTO;
import java.util.ArrayList;

/**
 *
 * @author Nguyen Hoang Anh
 */
public class EmployeeStyleBUS {

    public static ArrayList<EmployeeStyleDTO> selectEmployeeStyleAll() {
        return EmployeeStyleDAO.selectEmployeeStyleAll();
    }
}
```

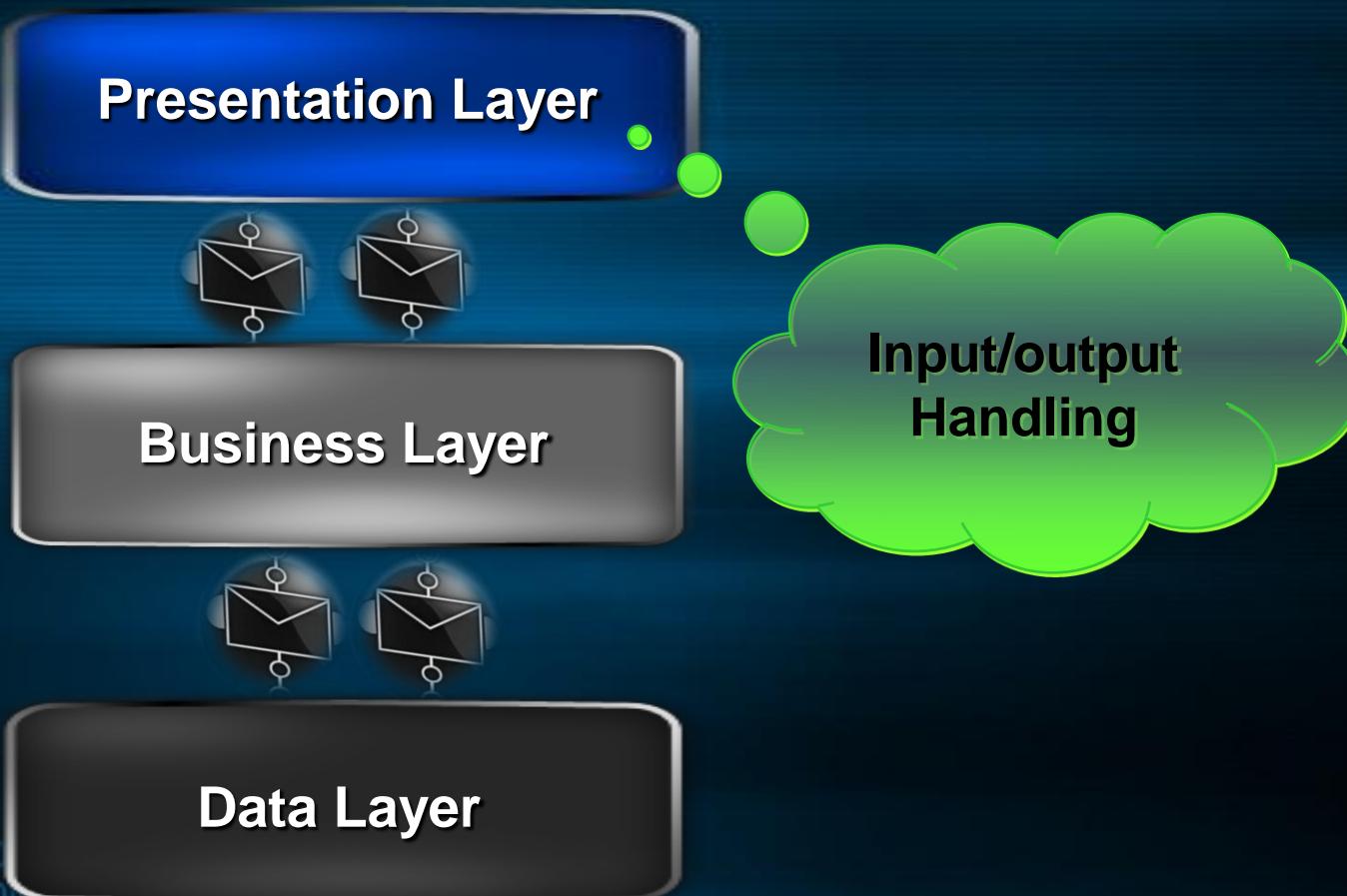
Business Tier



Tầng giao diện – Presentation Layer



Tầng giao diện – Presentation Layer



Tầng giao diện – Presentation Layer

Tầng này làm nhiệm vụ giao tiếp với người dùng để thu thập dữ liệu và hiển thị kết quả/ dữ liệu thông qua các thành phần trong giao diện người dùng. Tầng này sử dụng các dịch vụ do tầng Business cung cấp.

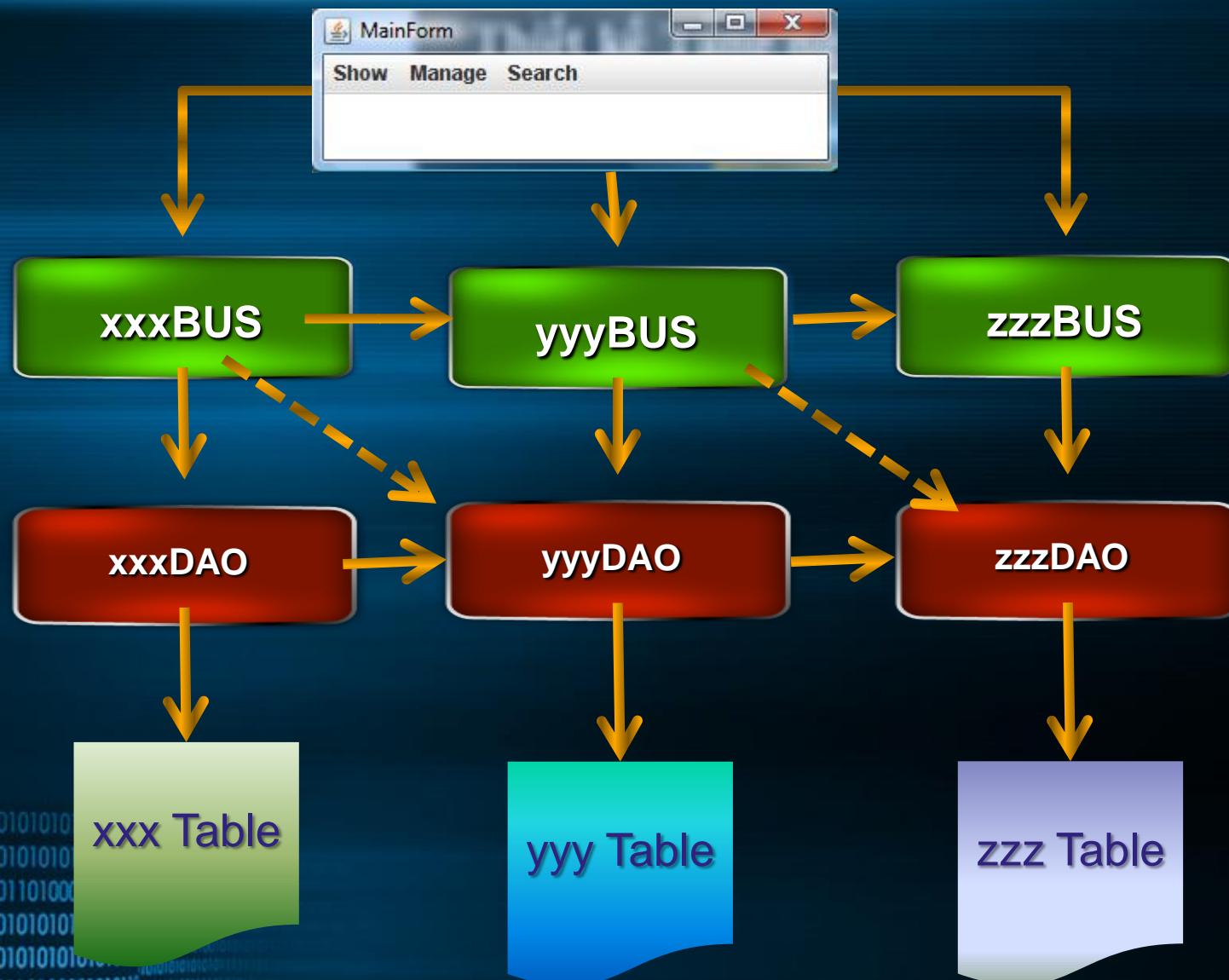
Tầng giao diện – Presentation Layer

Windows Forms

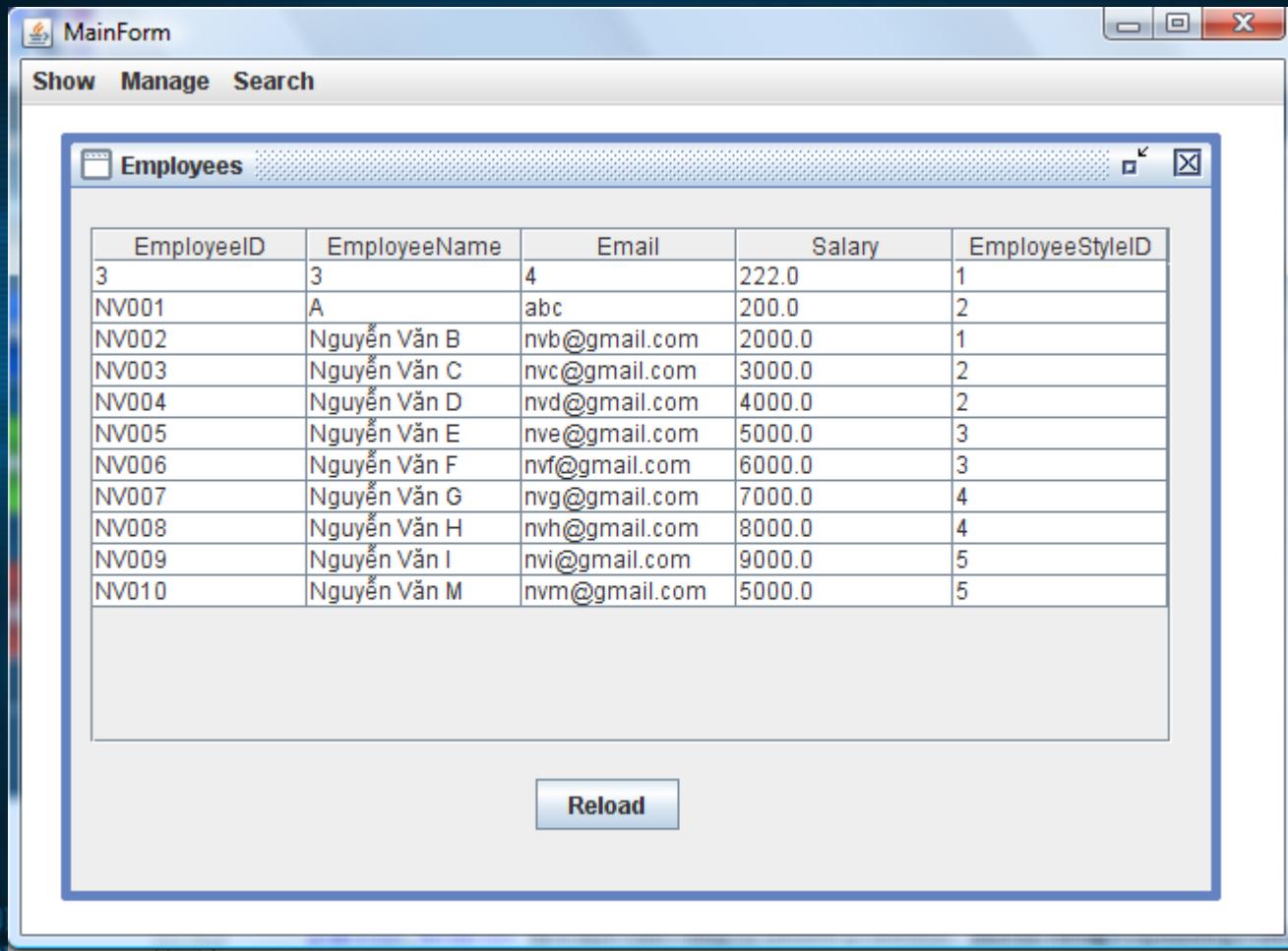
Web Forms

Mobile Forms

Tầng nghiệp giao diện – Presentation Layer



Tầng giao diện – HRM – Java Swing



Tầng giao diện – HRM – Java Swing

MainForm

Show Manage Search

Manage Employees

EmployeeID:	NV007	EmployeeName:	Nguyễn Văn G
Email:	nvg@gmail.com	Salary:	7000.0
EmployeeStyle:	Kinh Doanh		

EmployeeID	EmployeeName	Email	Salary
NV007	Nguyễn Văn G	nvg@gmail.com	7000.0
NV008	Nguyễn Văn H	nvh@gmail.com	8000.0

Insert Update Delete Reset

Đối tượng trao đổi dữ liệu – Data Transfer Object

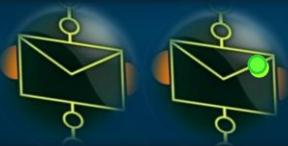


Đối tượng trao đổi dữ liệu – Data Transfer Object(DTO)

Presentation Layer



Business Layer



Data Layer

Đối tượng trao đổi
dữ liệu
Data Transfer Object

Đối tượng trao đổi dữ liệu với mẫu đơn giản



Các thuộc tính

Các Port
Property

Data Transfer Object – HRM - Java

```
package DTO;

/**
 *
 * @author Nguyen Hoang Anh
 */
public class EmployeeDTO {
    protected String employeeID;
    protected String employeeName;
    protected String email;
    protected float salary;
    protected int employeeStyleID;
    public String getEmail() { ... }
    public void setEmail(String email) { ... }
    public String getEmployeeID() { ... }
    public void setEmployeeID(String employeeID) { ... }
    public String getEmployeeName() { ... }
    public void setEmployeeName(String employeeName) { ... }
    public int getEmployeeStyleID() { ... }
    public void setEmployeeStyleID(int employeeStyleID) { ... }
    public float getSalary() { ... }
    public void setSalary(float salary) { ... }
}
```

Data Transfer Object – HRM - Java

```
package DTO;

/*
 *
 * @author Nguyen Hoang Anh
 */
public class EmployeeStyleDTO {

    protected int employeeStyleID;
    protected String employeeStyleName;

    public int getEmployeeStyleID() { ... }

    public void setEmployeeStyleID(int employeeStyleID) { ... }

    public String getEmployeeStyleName() { ... }

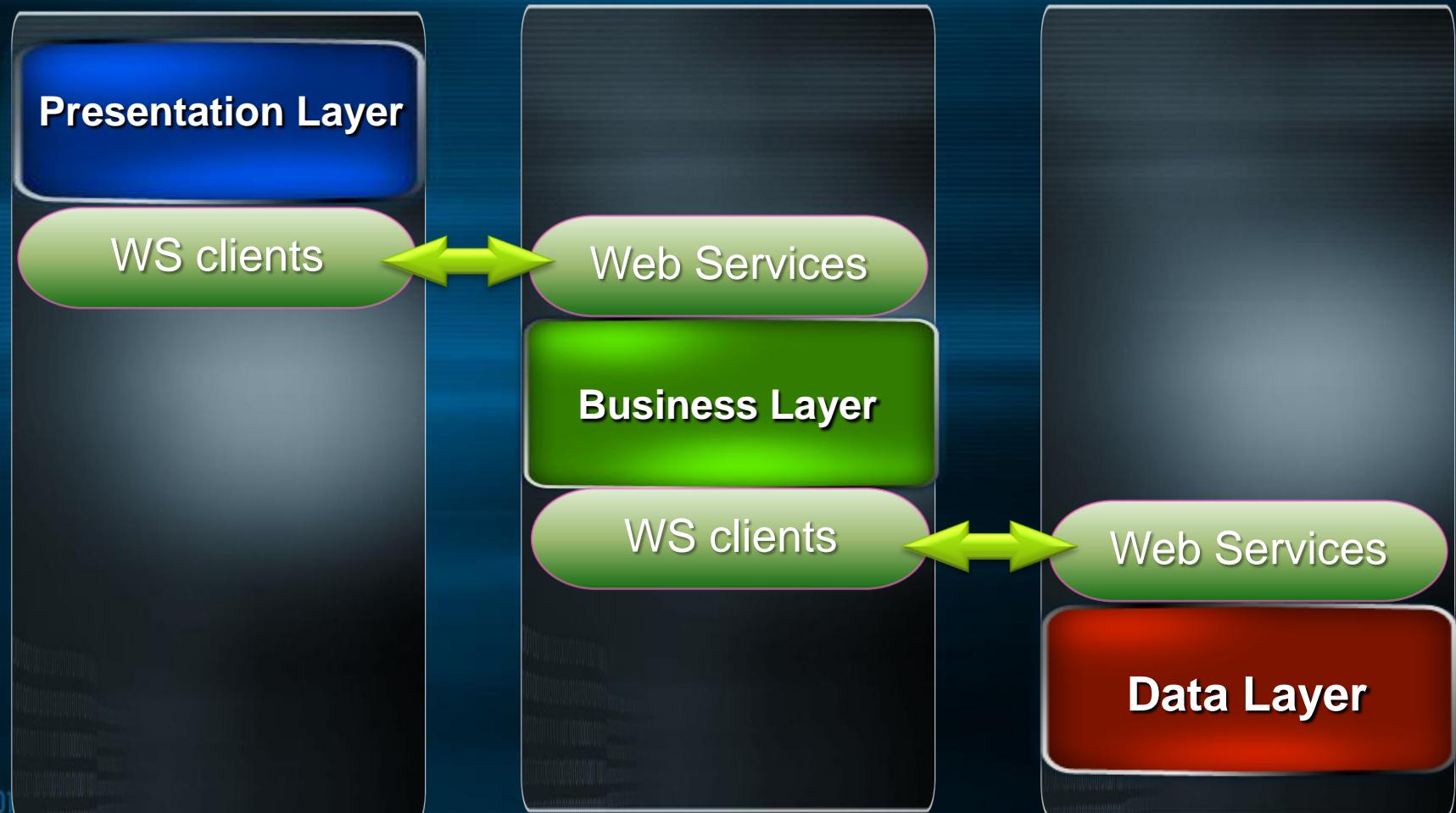
    public void setEmployeeStyleName(String employeeStyleName) { ... }
    @Override
    public String toString() { ... }
}
```

```
}
```

Kiến trúc 3 tầng (The Three-Layer Architecture)



Three Tier



Three Layer vs Three Tiers

Three Layer là một thể hiện của kiến trúc phần mềm ở dạng logic.

Three Tier là một thể hiện của kiến trúc phần mềm ở dạng vật lý (physical).

DEMO

The Three-Layer Architecture

Software Engineering

