



Web-разработка на C# и платформе Microsoft .NET

Основы трехслойной архитектуры



Классическая архитектура



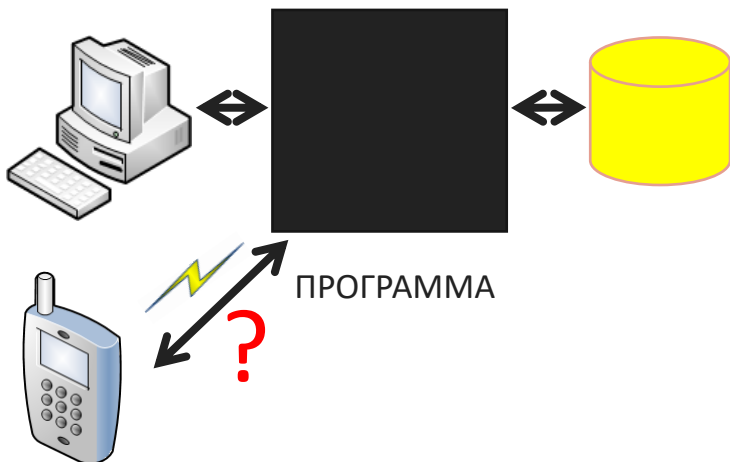
```
protected void ddSelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        DateTime rollbackDate = new DateTime(year, month, day);
        var copyFiles = from file in copyDirectory.GetFiles()
                        where file.CreationTime < rollbackDate
                        select file;

        FileInfo[] existingFiles = directory.GetFiles();
        foreach (FileInfo file in existingFiles)
        {
            var fileCopies = from item in copyFiles
                            where item.Name.StartsWith(file.Name)
                            select item;
            var fileCopiesArr = fileCopies.ToArray();
            var lastCopy = fileCopiesArr[0];
            for (int i = 1; i < fileCopiesArr.Length; i++)
            {
                if (fileCopiesArr[i].CreationTime > lastCopy.CreationTime)
                {
                    lastCopy = fileCopiesArr[i];
                }
            }

            string destinationFile = System.IO.Path.Combine(path, lastCopy.Name);
            string bcpFile = System.IO.Path.Combine(path, file.Name.Replace(destinationFile, bcpFile));

            file.CopyTo(bcpFile);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Введено некорректное значение даты");
    }
}
```

Классическая архитектура

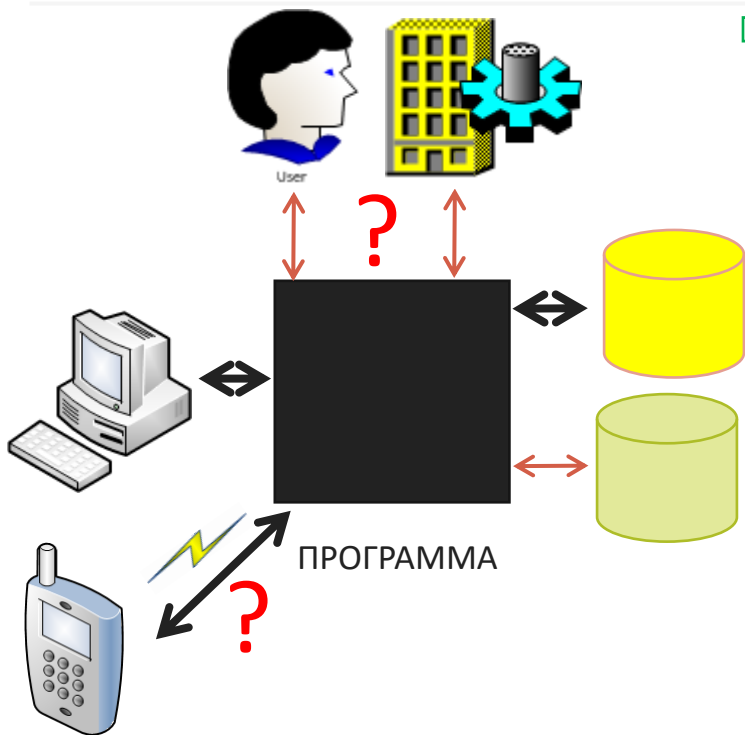


```
protected void ddSelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        DateTime rollbackDate = new DateTime(year, month, day);
        var copyFiles = from file in copyDirectory.GetFiles()
                        where file.CreationTime < rollbackDate
                        select file;

        FileInfo[] existingFiles = directory.GetFiles();
        foreach (FileInfo file in existingFiles)
        {
            var fileCopies = from item in copyFiles
                            where item.Name.StartsWith(file.Name)
                            select item;
            var fileCopiesArr = fileCopies.ToArray();
            var lastCopy = fileCopiesArr[0];
            for (int i = 1; i < fileCopiesArr.Length; i++)
            {
                if (fileCopiesArr[i].CreationTime > lastCopy.CreationTime)
                {
                    lastCopy = fileCopiesArr[i];
                }
            }

            string destinationFile = System.IO.Path.Combine(path, lastCopy.Name);
            string bcupFile = System.IO.Path.Combine(path, lastCopy.Name + ".bcup");
            File.Copy(file.FullName, destinationFile, true);
            File.Copy(file.FullName, bcupFile, true);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Введено некорректное значение даты");
    }
}
```

Классическая архитектура



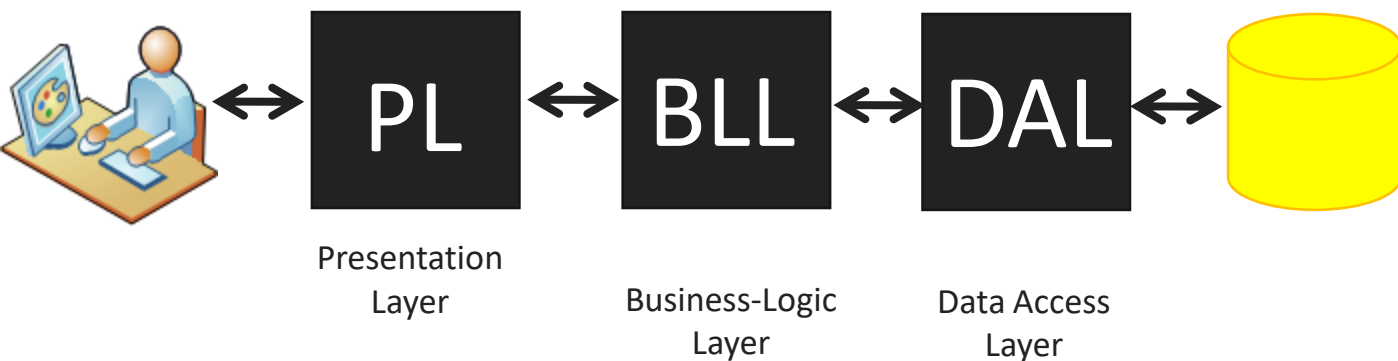
```
protected void ddSelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        DateTime rollbackDate = new DateTime(year, month, day);
        var copyFiles = from file in copyDirectory.GetFiles()
                        where file.CreationTime < rollbackDate
                        select file;

        FileInfo[] existingFiles = directory.GetFiles();
        foreach (FileInfo file in existingFiles)
        {
            var fileCopies = from item in copyFiles
                            where item.Name.StartsWith(file.Name)
                            select item;
            var fileCopiesArr = fileCopies.ToArray();
            var lastCopy = fileCopiesArr[0];
            for (int i = 1; i < fileCopiesArr.Length; i++)
            {
                if (fileCopiesArr[i].CreationTime > lastCopy.CreationTime)
                {
                    lastCopy = fileCopiesArr[i];
                }
            }

            string destinationFile = System.IO.Path.Combine(path, lastCopy.Name);
            string bcupFile = System.IO.Path.Combine(path + "\\Backup", lastCopy.Name);

            File.Copy(destinationFile, bcupFile, true);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Введено некорректное значение даты");
    }
}
```

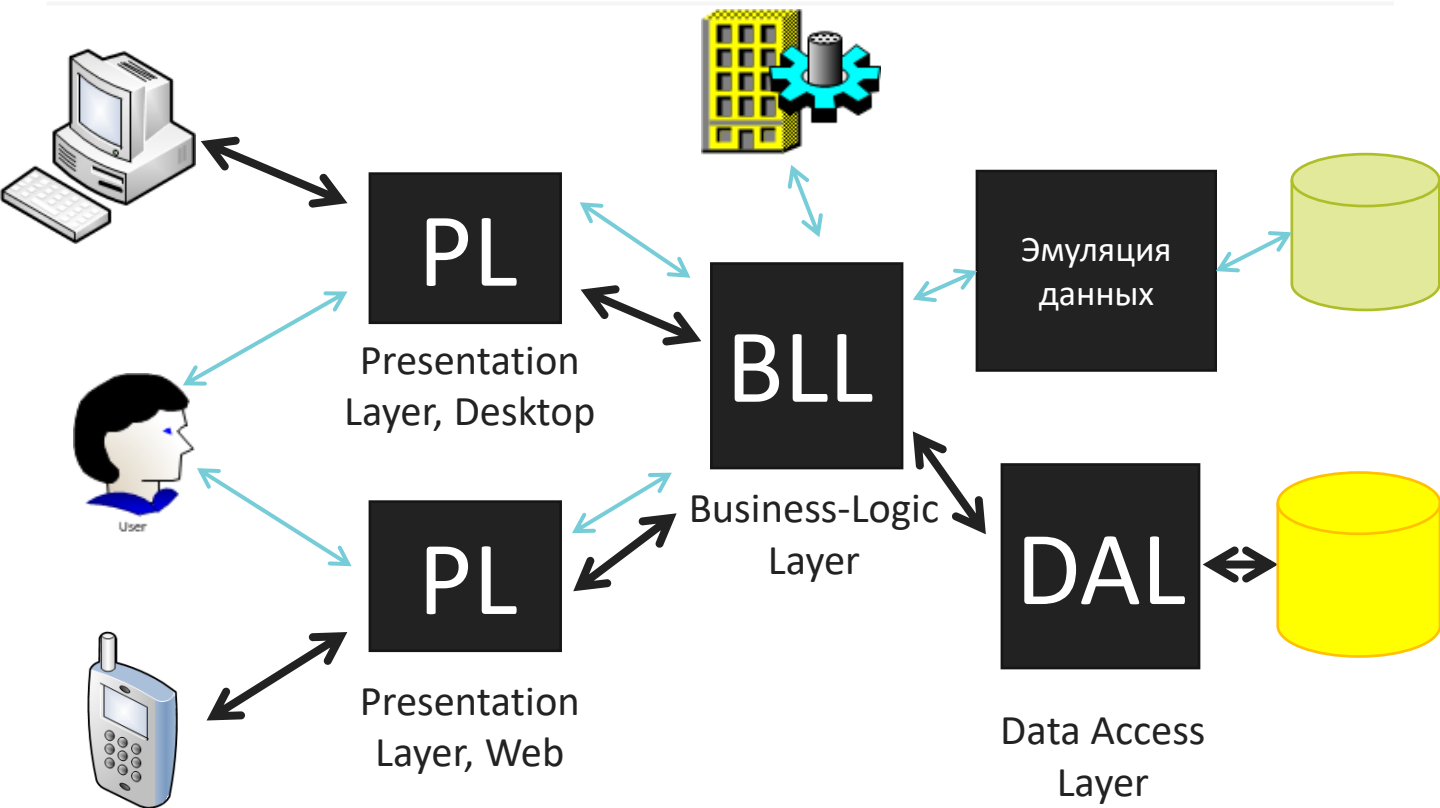
Трёхслойная архитектура



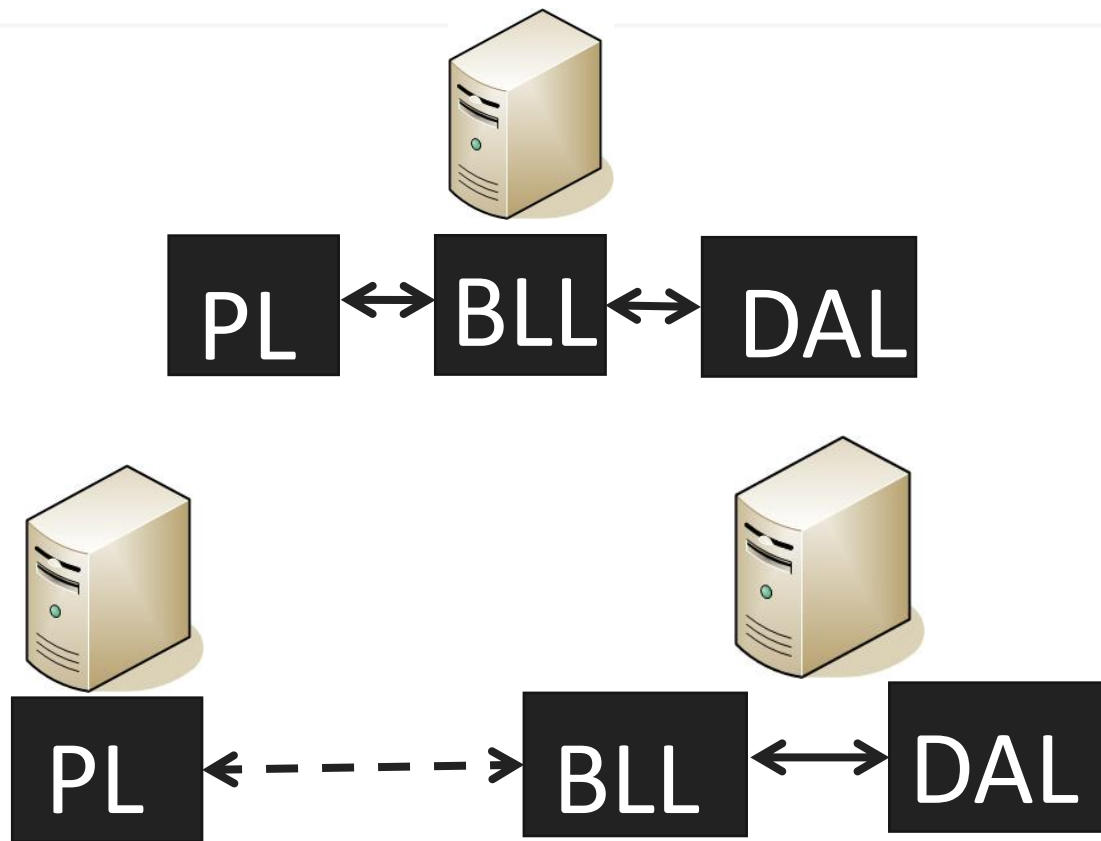
Где используется

- Приложение имеет несколько интерфейсов
- Возможно последующее изменение способа хранения данных, или интерфейса без существенного изменения функциональности
- Необходимо сократить нагрузку на сервер, разнеся ее по нескольким машинам

Изменение способа хранения



Масштабирование



СПАСИБО ЗА ВНИМАНИЕ