

# THE ONION ARCHITECTURE

# JEFFREY PALERMO

## MANAGING PARTNER & CEO



## ПОЛЕЗНЫЕ ССЫЛКИ

- [JEFFREYPALERMO.COM](http://JEFFREYPALERMO.COM)
- [DEVELOP.COM/ONIONARCHITECTURE](http://DEVELOP.COM/ONIONARCHITECTURE)
- [HABRAHABR.RU/POST/233747/](http://HABRAHABR.RU/POST/233747/)



# ПЛАН НА СЕГОДНЯ

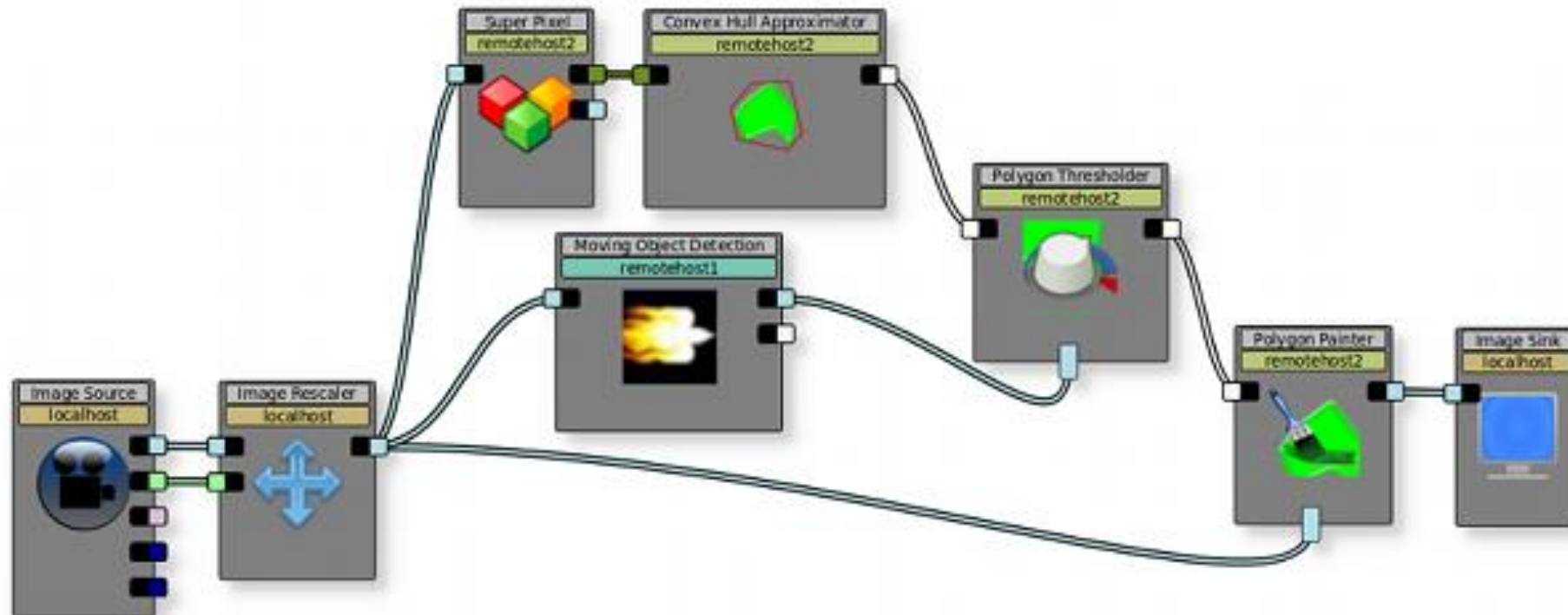
- ВВОДНАЯ ИНФОРМАЦИЯ + ТРАДИЦИОННЫЙ ПОДХОД
- ЛУКОВАЯ АРХИТЕКТУРА
- СРАВНЕНИЕ

# ПОСЛЕДСТВИЯ





# MODULAR PROGRAMMING



# INTERFACES





# JAVA-ПРИМЕР

The screenshot shows an IDE with a project named 'untitled1'. The project structure includes a package 'kazakov' with sub-packages 'animals' and 'forest'. The 'animals' package contains classes 'Animal', 'Bear', and 'Hedgehog'. The 'forest' package contains the class 'FitTree'. The 'FitTree.java' file is open, showing the following code:

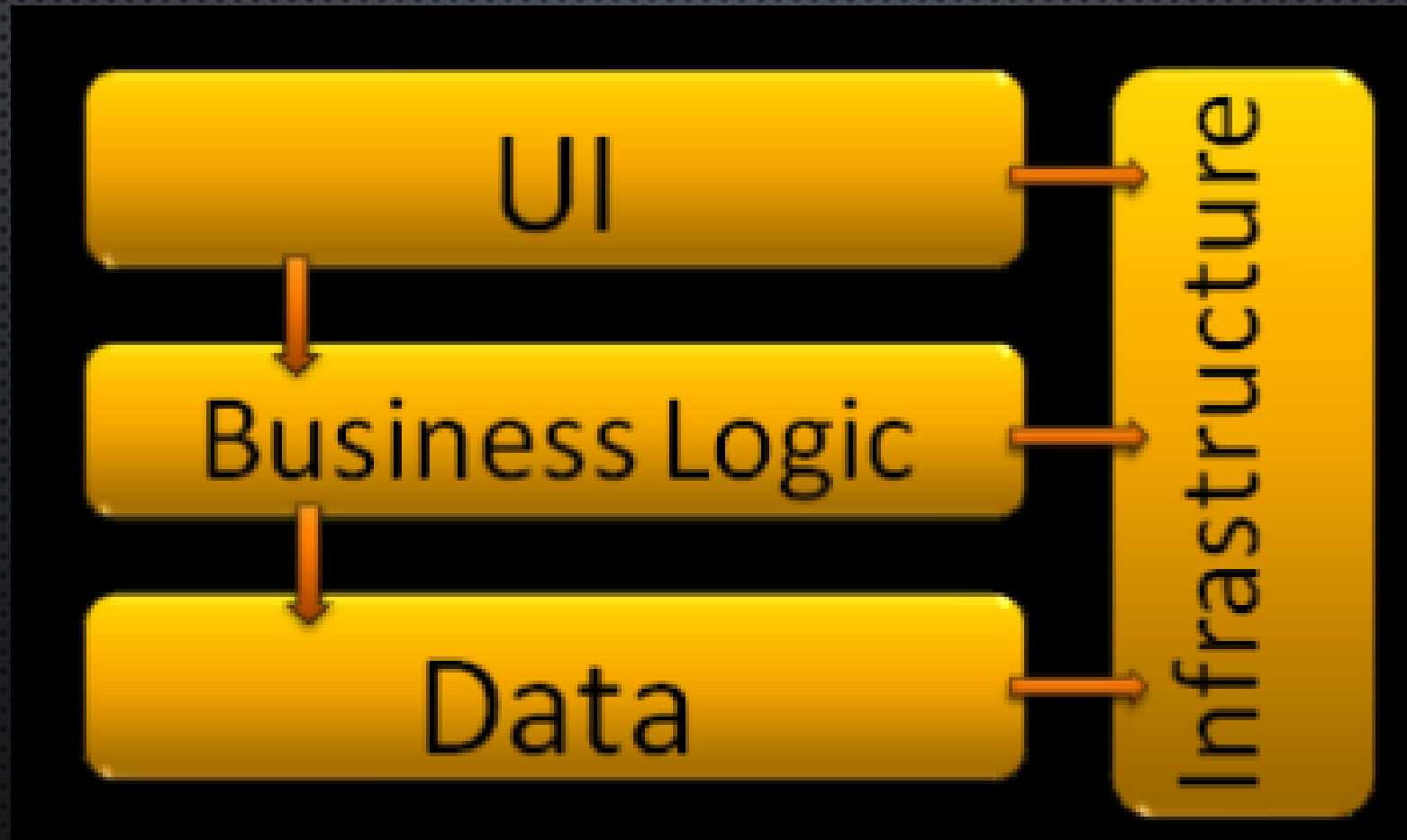
```
public static void main(String[] args) {  
    Animal animal = new Bear();  
    System.out.println("bear: " + animal.conHited());  
  
    animal = new Hedgehog();  
    System.out.println("hedgehog: " + animal.conHited());  
}
```

The output window shows the execution results:

```
"C:\Program ...  
bear:      Мишка рассердился и ногою - топ!  
hedgehog:  DEAD  
  
Process finished with exit code 0
```

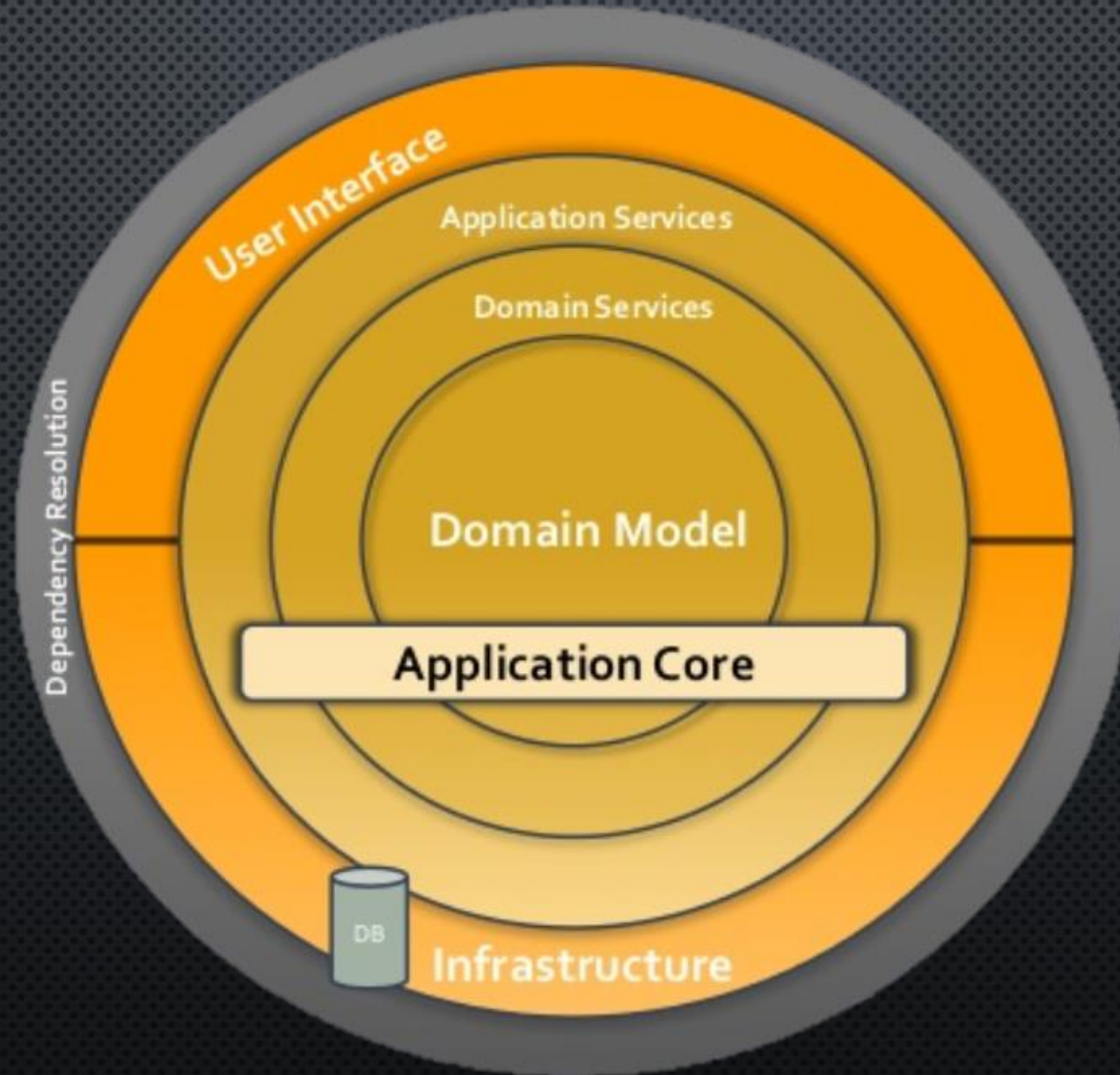
The status bar at the bottom indicates that all files are up-to-date (5 minutes ago) and shows the current time as 17:1.

# TRADITIONAL LAYERED ARCHITECTURE

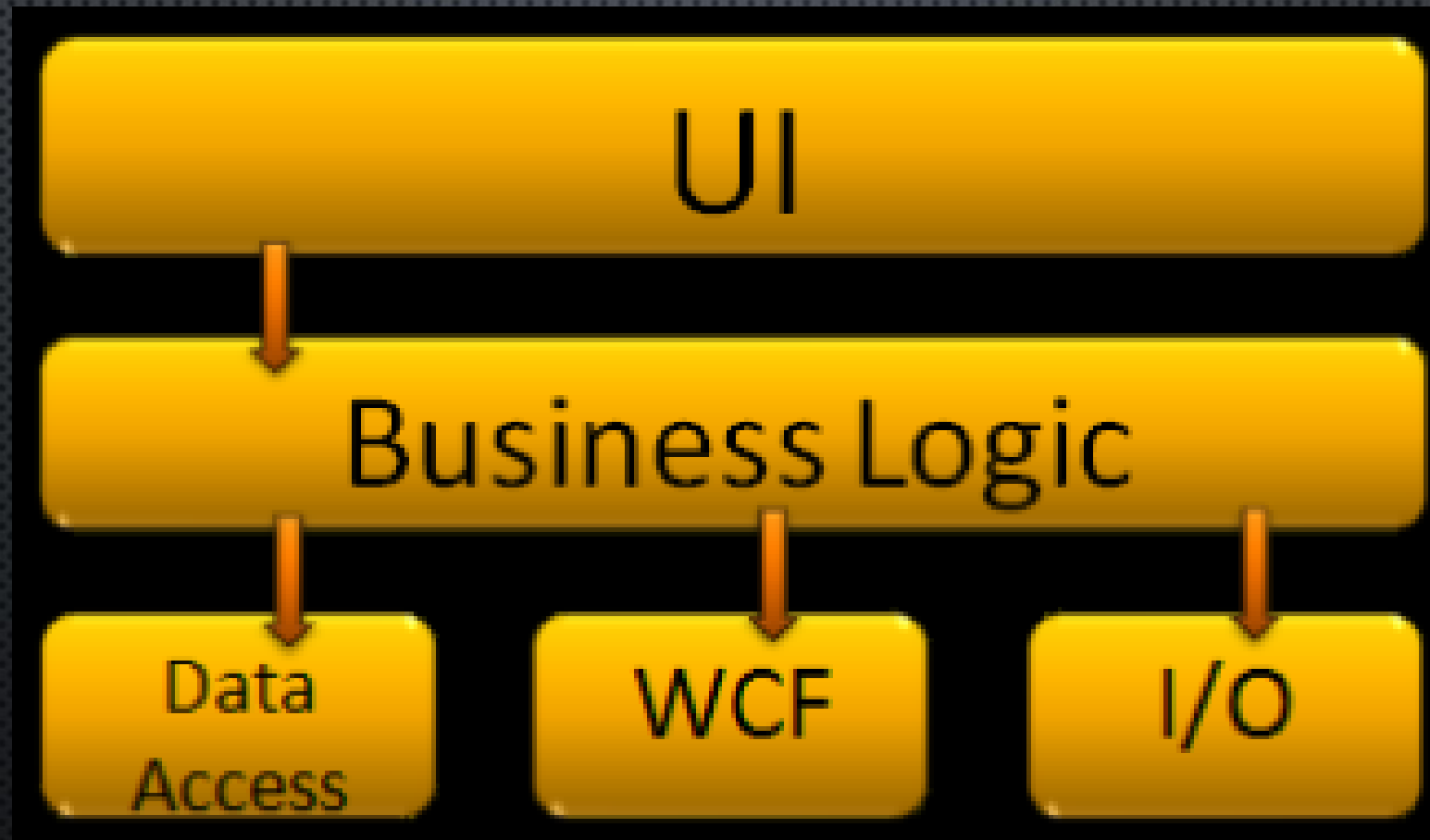




# THE ONION ARCHITECTURE

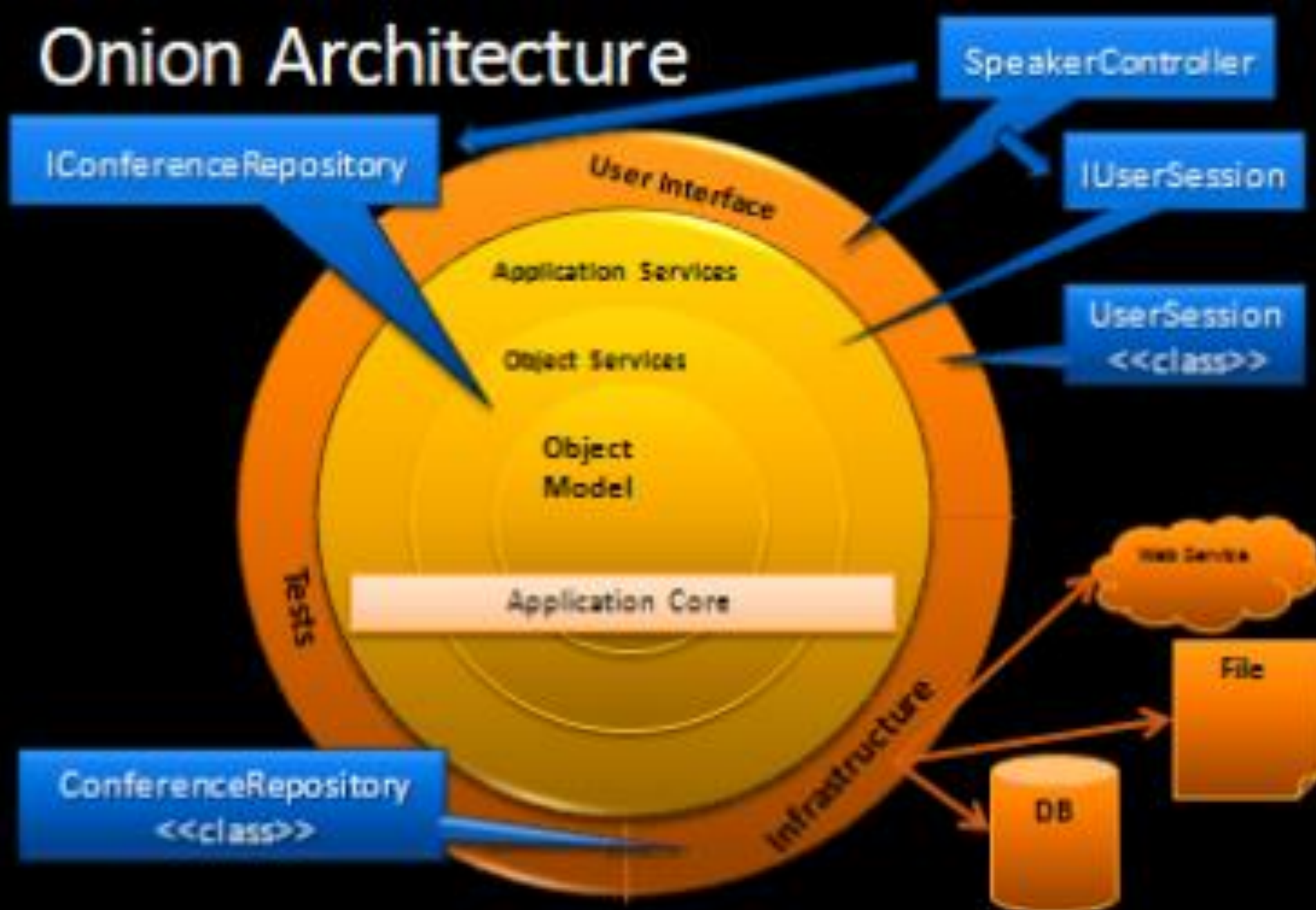


# TRADITIONAL LAYERED ARCHITECTURE

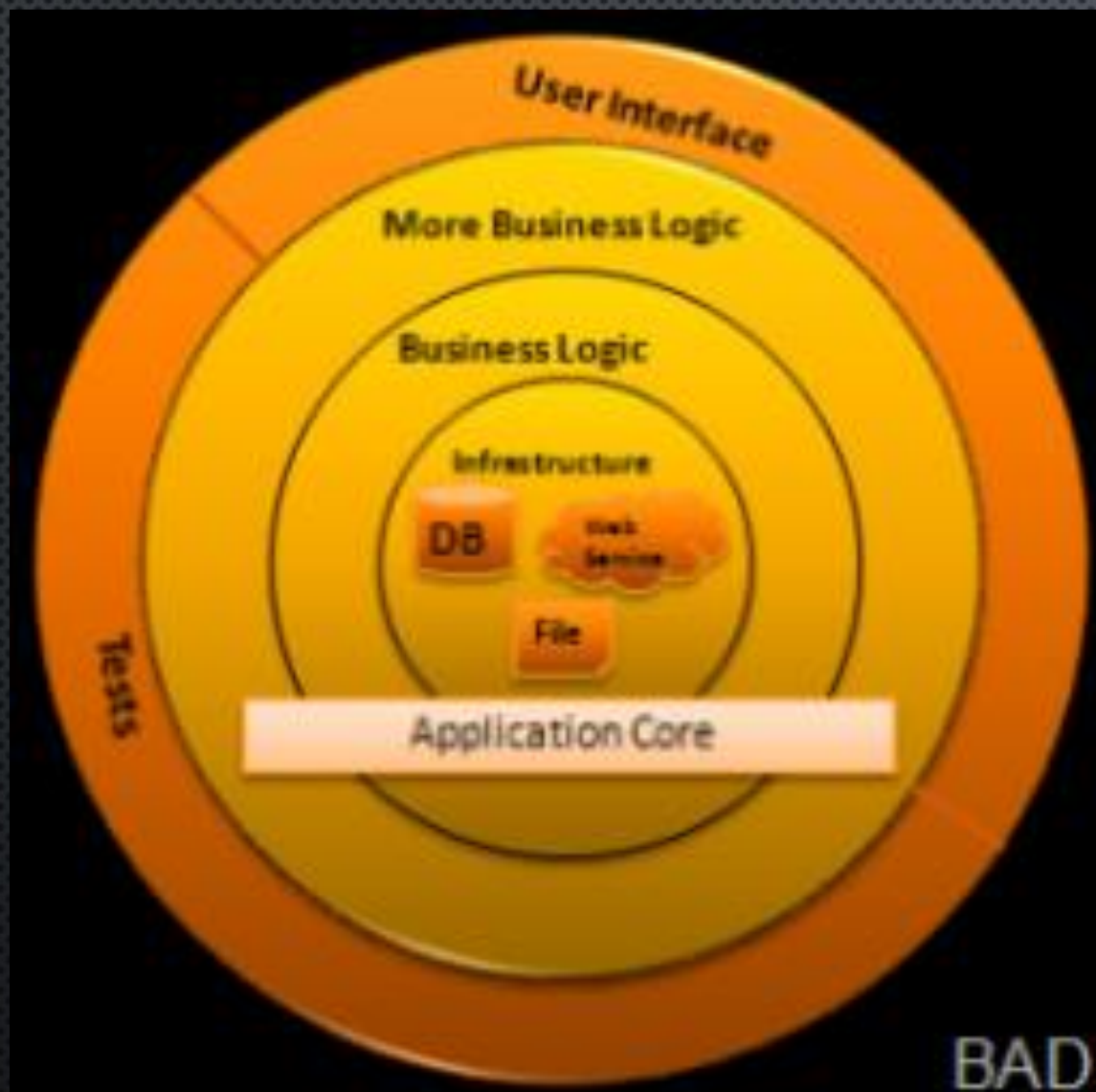




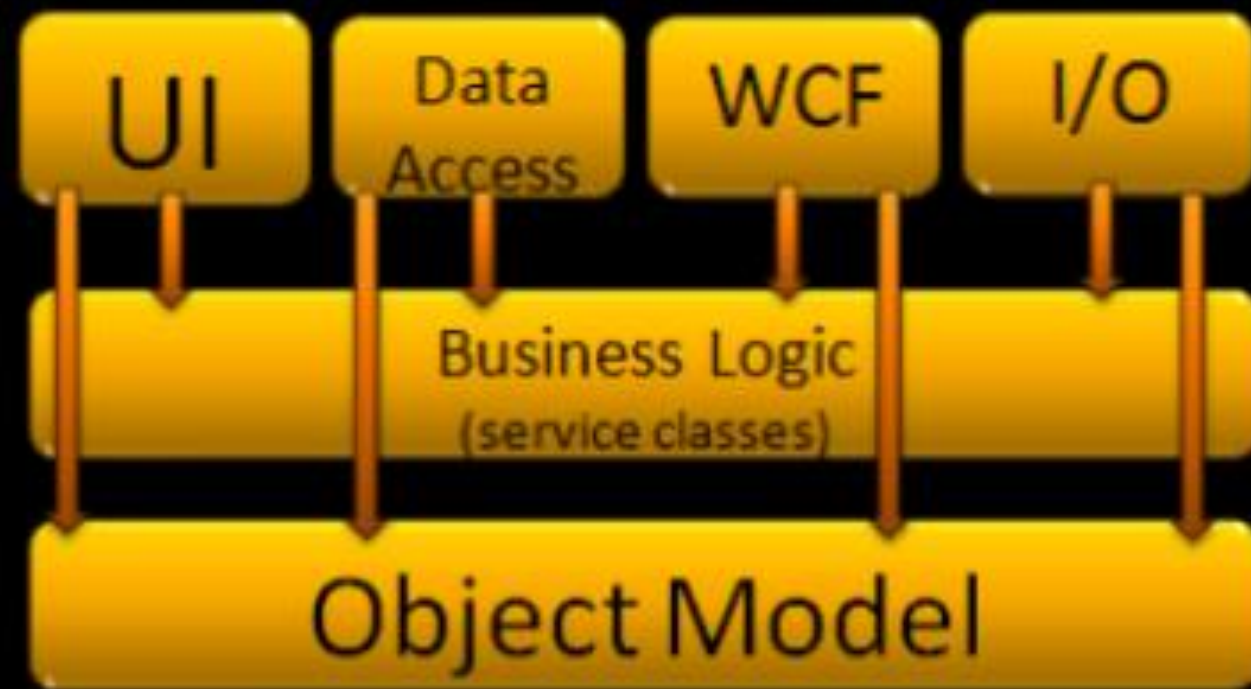
# Onion Architecture



# НОВЫЙ ВЗГЛЯД



## Onion Architecture (flattened)





# КЛЮЧЕВЫЕ ПРИНЦИПЫ ONION ARCHITECTURE

- ПРИЛОЖЕНИЕ СТРОИТСЯ ВОКРУГ НЕЗАВИСИМОЙ ОБЪЕКТНОЙ МОДЕЛИ
- ВНУТРЕННИЕ СЛОИ ОПРЕДЕЛЯЮТ ИНТЕРФЕЙСЫ. НАРУЖНЫЕ СЛОИ РЕАЛИЗУЮТ ИХ
- СВЯЗНОСТЬ РАСПРОСТРАНЯЕТСЯ К ЦЕНТРУ
- ВСЕ КОД ЯДРА ПРИЛОЖЕНИЯ МОЖЕТ БЫТЬ СКОМПИЛИРОВАН И ЗАПУЩЕН БЕЗ РЕАЛИЗОВАННОЙ ИНФРАСТРУКТУРЫ

БЛАГОДАРЮ ЗА ВНИМАНИЕ