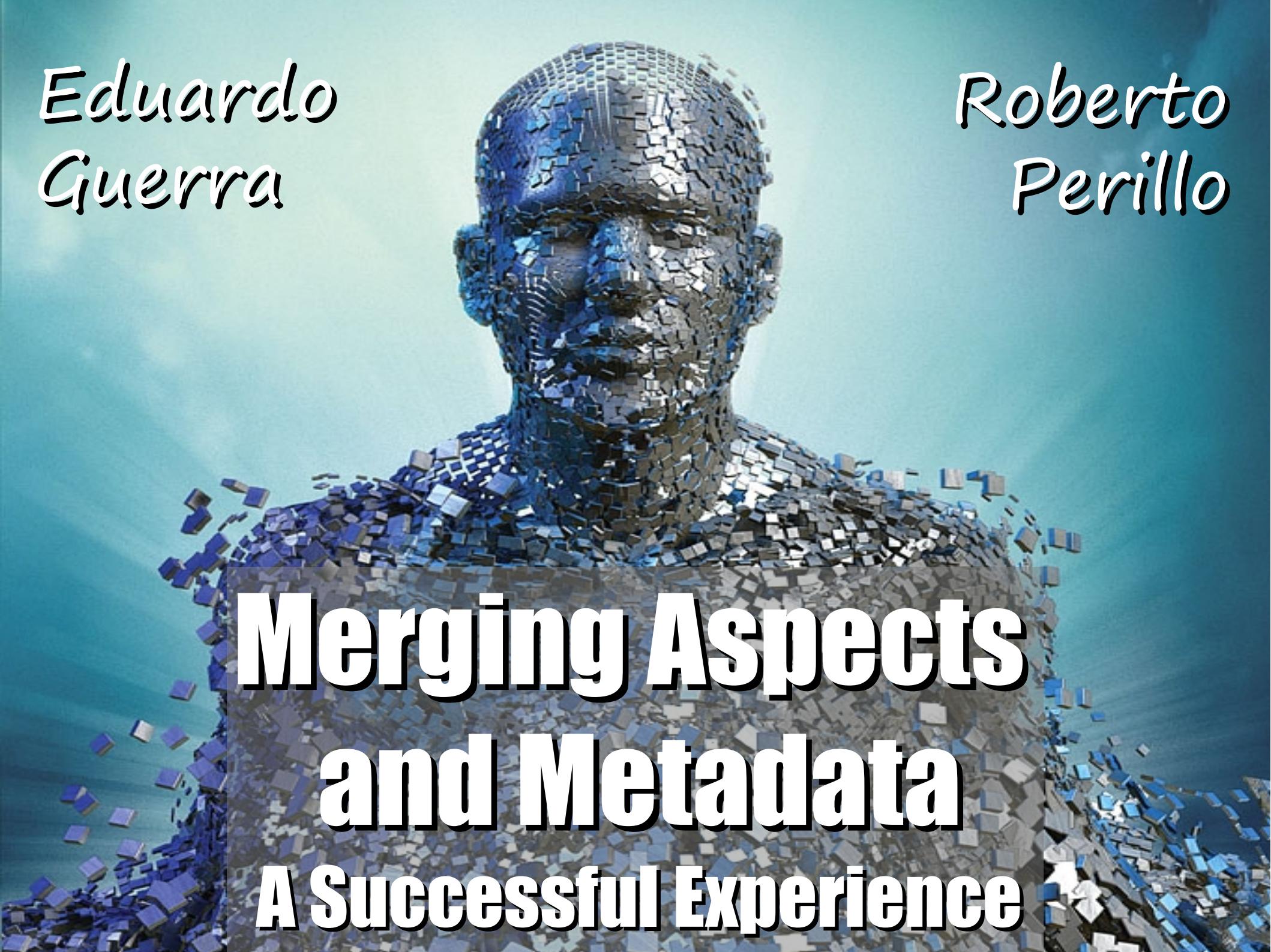


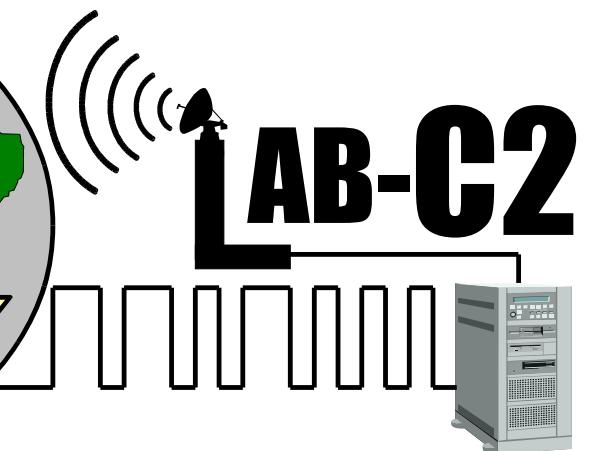
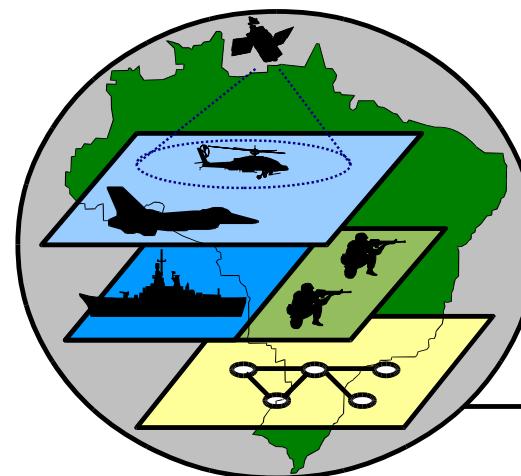
*Eduardo  
Guerra*

*Roberto  
Perillo*



# Merging Aspects and Metadata A Successful Experience

# THANKS!





# *Important Concepts*



## *Application Requirements*



## *Software Architecture*

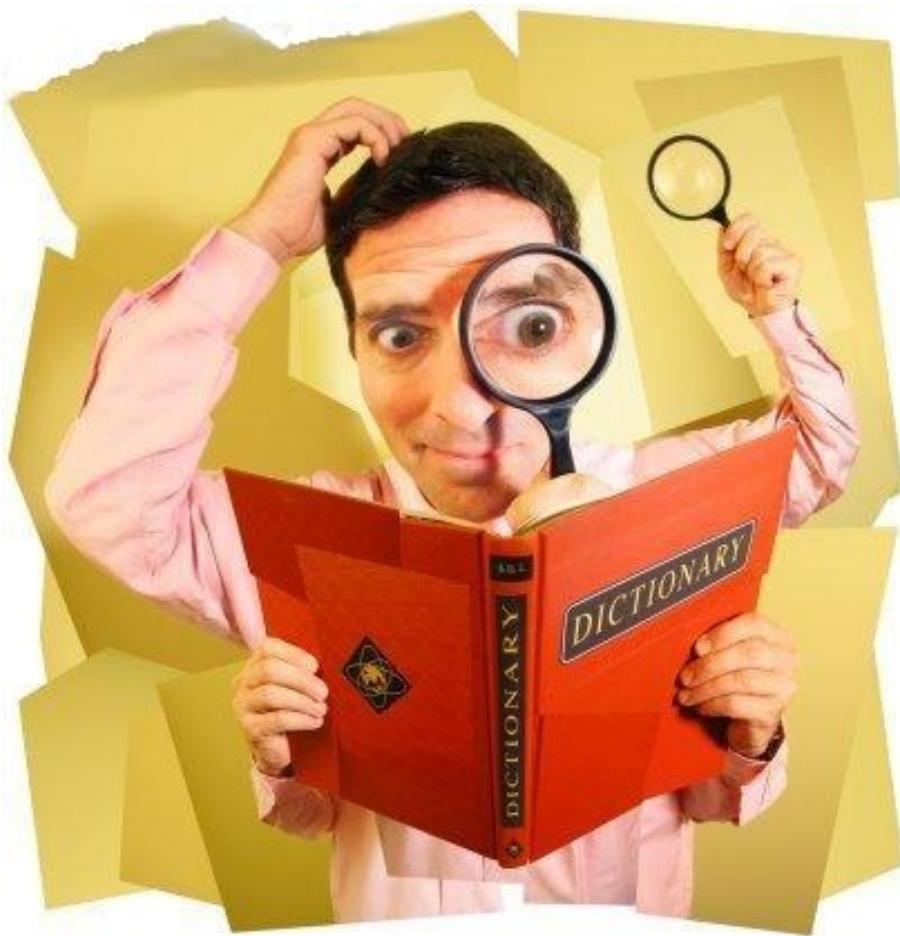


## *Implementation Details*



## *Impacts*

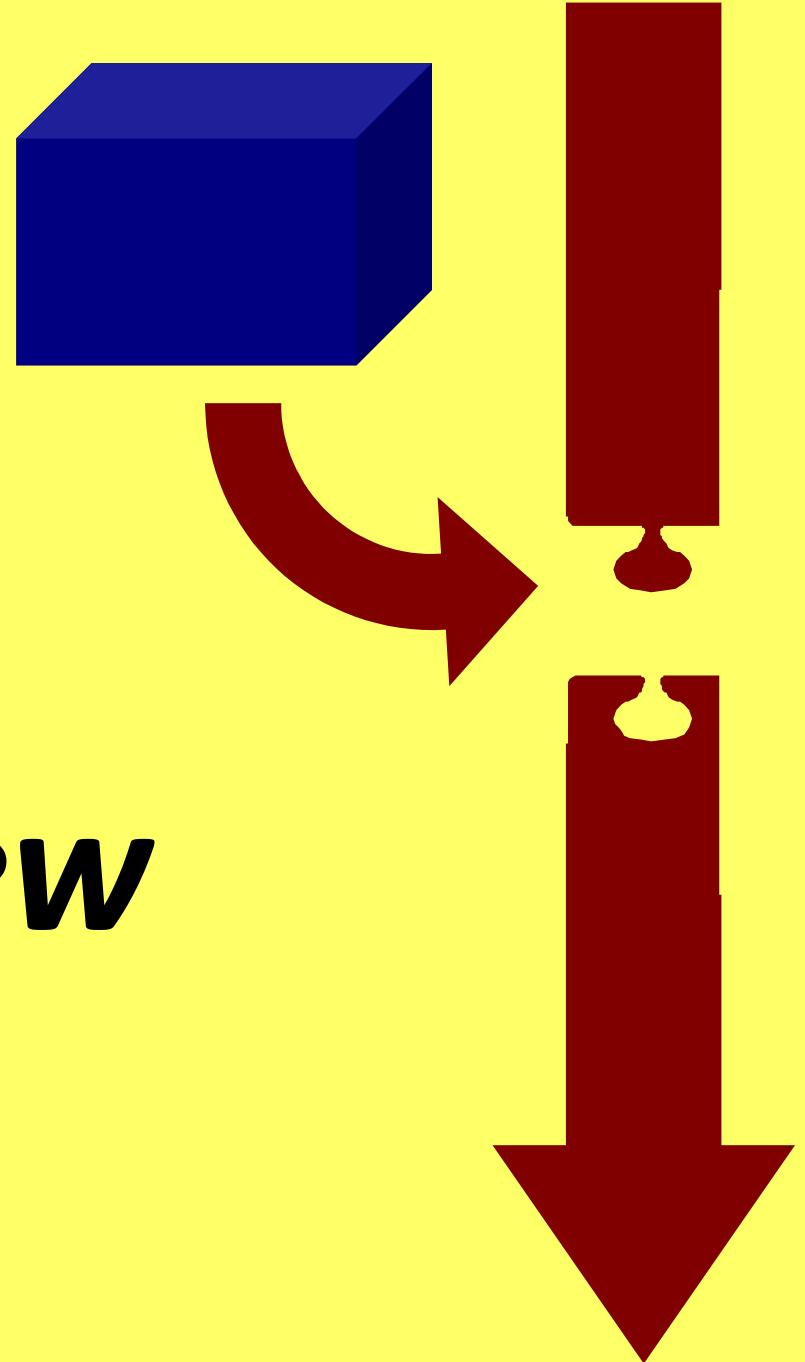
# Important Concepts



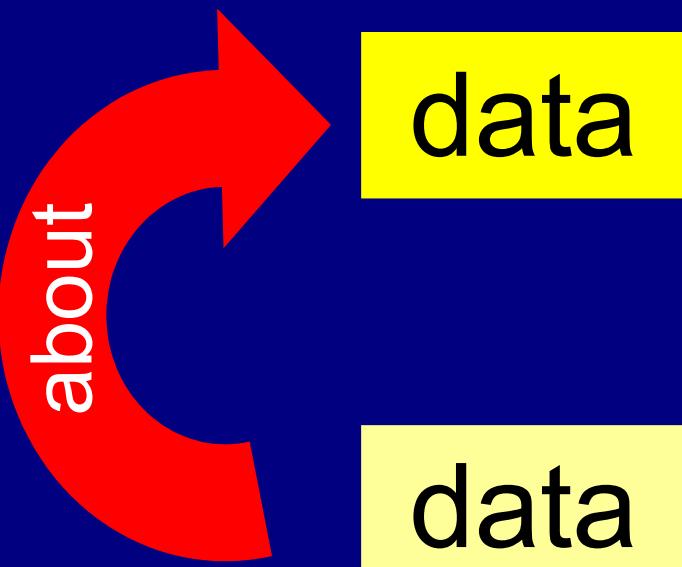
*Aspect-oriented programming is a programming paradigm which aims to increase **modularity** allowing the separation of **crosscutting concerns**.*



*Aspects can intercept the execution of methods to introduce a new crosscutting behavior.*



# metadata =



## domain-specific metadata

@Good

public

String

method(

int n ) ;

## intrinsic metadata

@Annotation

database

metadata  
definition

*conventions*

<XML>

programatic()

# Metadata



*Aspects can use custom metadata to identify where it should act and to customize its behavior.*

# **FACT**



Aspects and  
custom metadata  
are rarely  
applied in real  
applications!



# **Afraid of new technologies**

## **Lack of Knowledge**



## **Not enough successful cases**



# Requirements



# **Web application**



# **Critical**



# **Average number of users**



## **Information should reach fast its destination**



# Reverse AJAX

# Granular Access Control



# **Instance validation**



## **Logging Changes**



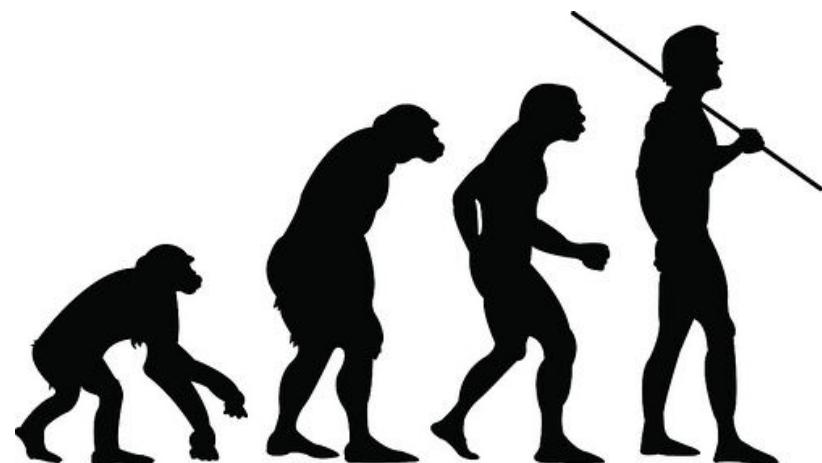
# **Integration with 3 Other Applications**

# Challenge

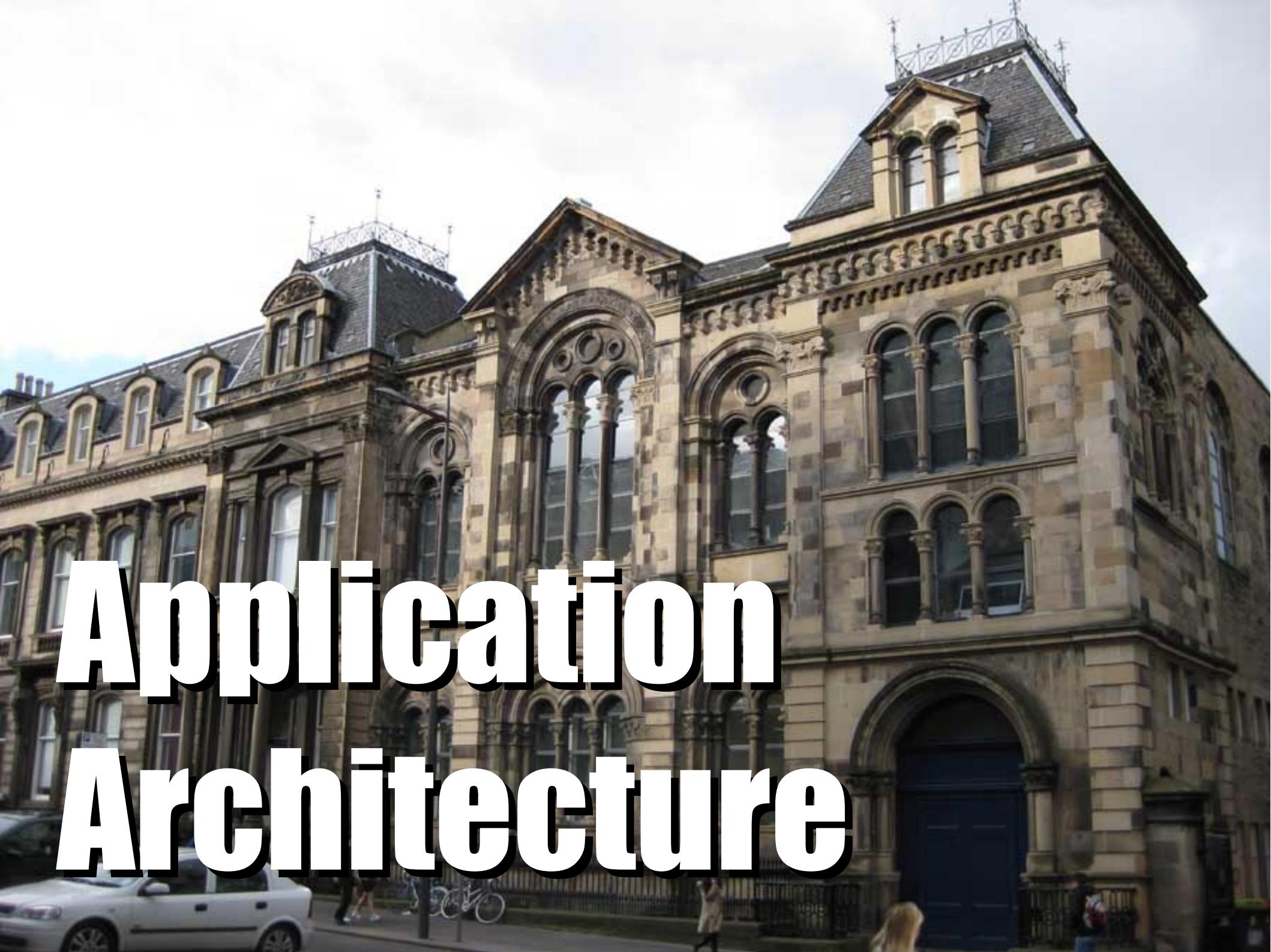


*Productivity*

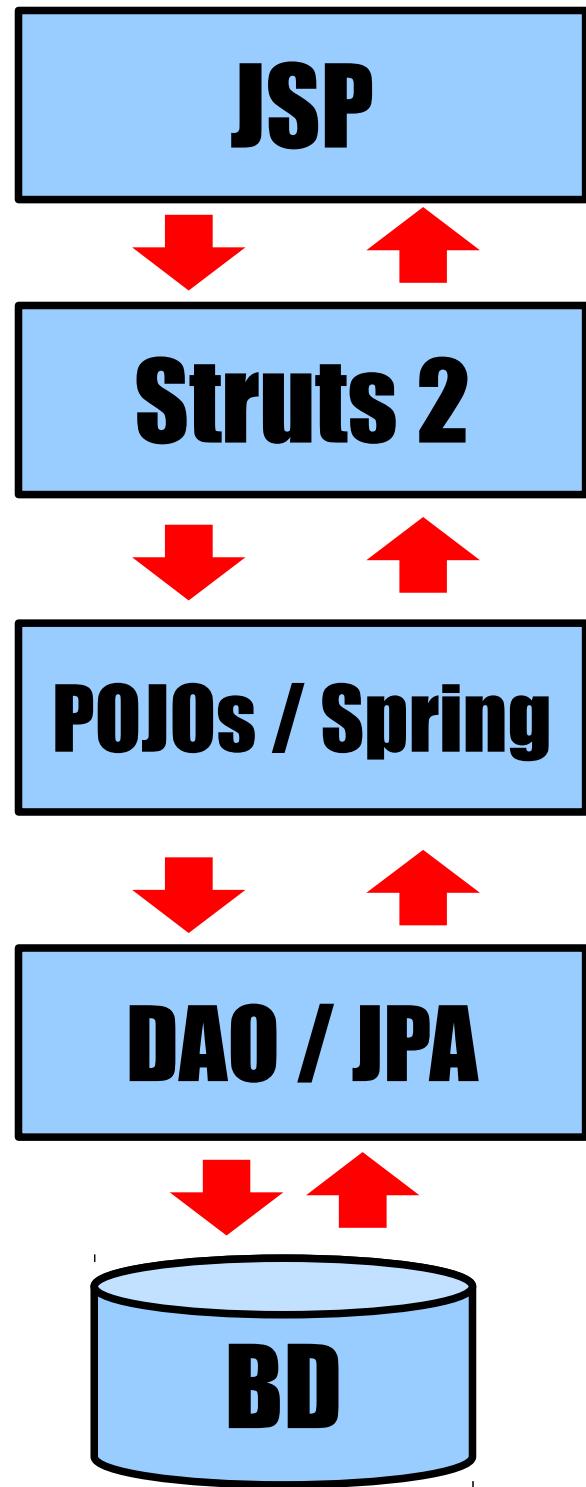
*Evolutionary  
Architecture*



# Application Architecture

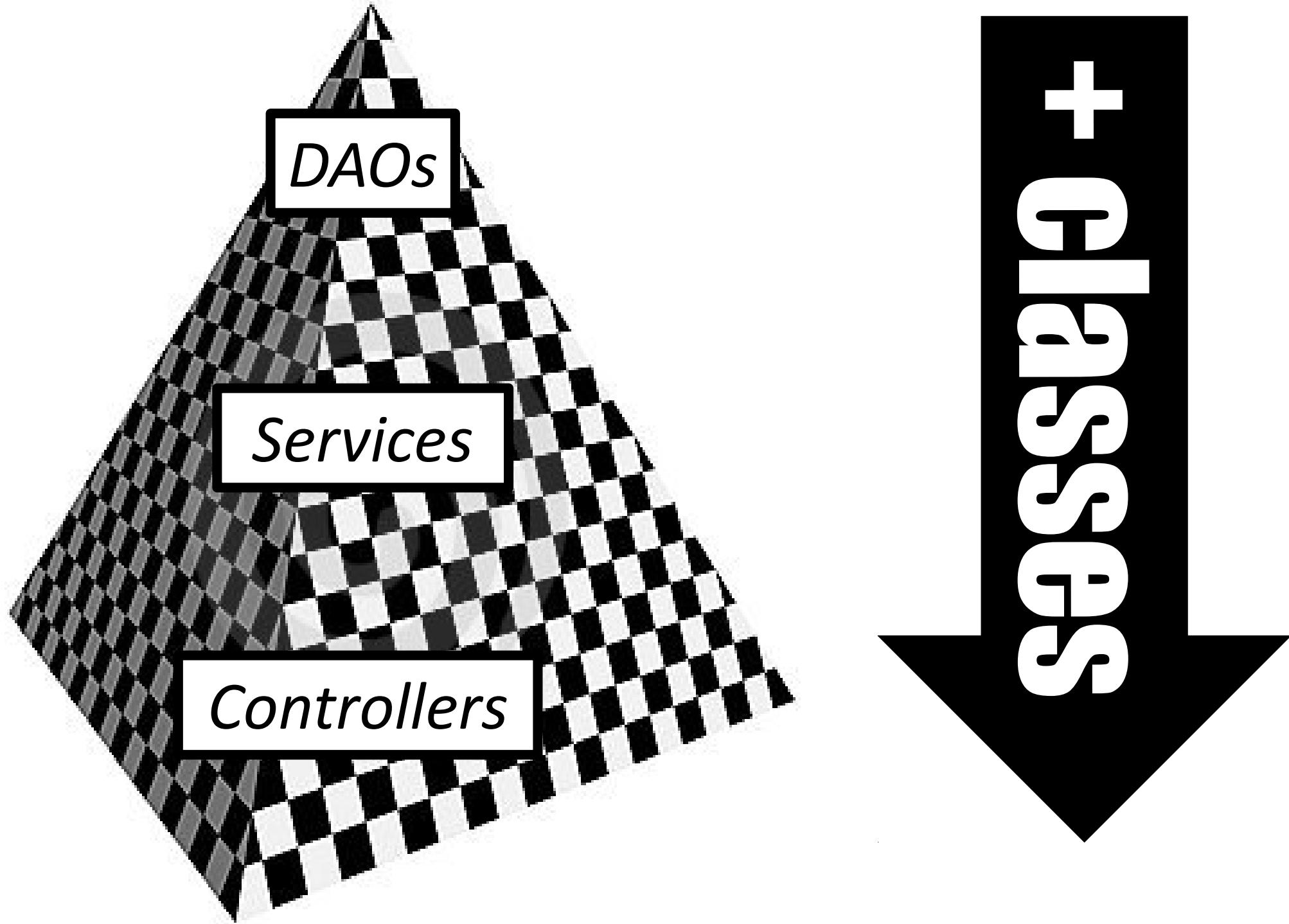


# Application Architecture



**DomainClass**

-atribute1
-atribute2
-atribute3
+getAttribute1()
+setAttribute1()
+getAttribute2()
+setAttribute2()
+getAttribute3()
+setAttribute3()



JSP



Struts 2

POJOs / Spring

DAO / JPA

BD



Struts  
Interceptors

AspectJ

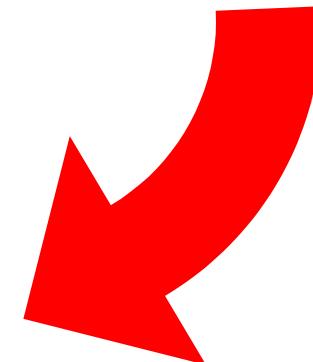


+



# Metadata

*Define how each service should be handled when intercepted by the aspects.*

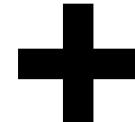




# *General and reusable services*

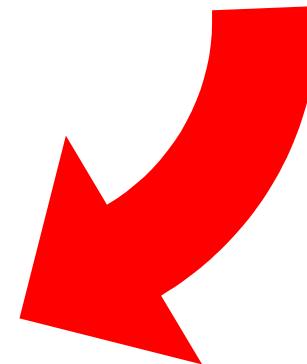
How to differentiate aspect behavior for each domain class used by the service?

DomainClass
-atribute1
-atribute2
-atribute3
+getAttribute1()
+setAtribute1()
+getAttribute2()
+setAtribute2()
+getAttribute3()
+setAtribute3()



# Metadata

*Describe specific information that define how the aspect should handle an interception including that domain class.*



# Implementation Details



# Finding Domain Class

1. *Search for @DomainClass annotation*

```
@DomainClass (Entity.class)  
public void m(Object o, Class c);
```

A diagram illustrating the search for a domain class. It shows a code snippet with annotations. A blue arrow points upwards from the 'Object' parameter to the '@DomainClass' annotation. A green arrow points downwards from the 'Class' parameter to the variable 'c'. The code itself is:  
`@DomainClass (Entity.class)  
public void m(Object o, Class c);`

2. *Search for an object or a class  
that represents a domain*



# Example: **Reverse AJAX**

*Needed in many functionalities*

*Hardworking to create*

*Deal with a complex framework*



change



DomainClass
-attribute1
-attribute2
-attribute3
+getAttribute1()
+setAttribute1()
+getAttribute2()
+setAttribute2()
+getAttribute3()
+setAttribute3()

update

CZ | PL | EN

Přihlášený uživatel: demouser  
Demo společnost  
Polní 5  
74601 Opava  
[Odhlásit](#)

### Objednávky

Filtr

Stav zpracování: Vše  
16 - Nová  
18 - Potvrzená  
19 - Odmítnutá

Datum vystavení od:   Datum dodání od:

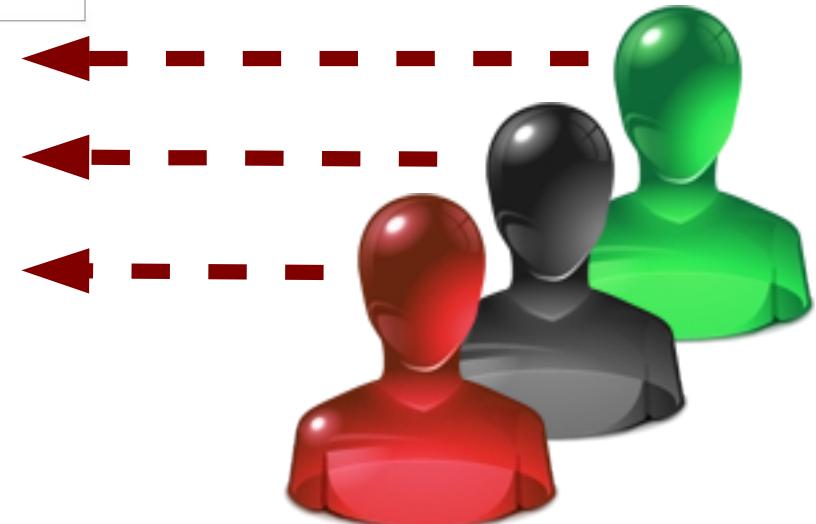
Fakturované:  Odběratel:   
Přečtené:  Místo dodání:

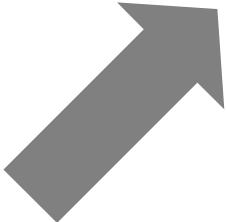
Zobrazit [Vyčistit formulář](#)

Aktuální výběrová podmínka:								
Číslo dokladu	Datum vystavení	Odběratel	Stav dokladu	Datum dodání	Místo dodání	Poznámka	Fatura	Akce
000028	05.09.2008	8546963740017	18	21.03.2002	8546963740017	Kontakt: Dalibor Kre...		
000021	05.09.2008	8546963740017	18	21.03.2002	8546963740017	Kontakt: Dalibor Kre...		
000027	05.09.2008	8546963740017	18	21.03.2002	8546963740017	Kontakt: Dalibor Kre...		
000012	05.09.2008	8546963740017	18	21.03.2002	8546963740017	Kontakt: Dalibor Kre...		
000019	05.09.2008	8546963740017	18	21.03.2002	8546963740017	Kontakt: Dalibor Kre...		
000013	05.09.2008	8546963740017	18	21.03.2002	8546963740017	Kontakt: Dalibor Kre...		
000020	05.09.2008	8546963740017	18	21.03.2002	8546963740017	Kontakt: Dalibor Kre...		
000015	05.09.2008	8546963740017	18	21.03.2002	8546963740017	Kontakt: Dalibor Kre...		
000014	05.09.2008	8546963740017	18	21.03.2002	8546963740017	Kontakt: Dalibor Kre...		
000016	05.09.2008	8546963740017	18	21.03.2002	8546963740017	Kontakt: Dalibor Kre...		

Strana 2 z 4, celkem zaznamů: 33

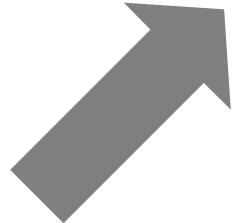
[<] [<] [2] [>] [>]





*Service method should indicate that it modifies the domain entity*

```
@DataModification  
public E save(E p) { ... }
```



*Domain class should indicate which page and region should be updated*

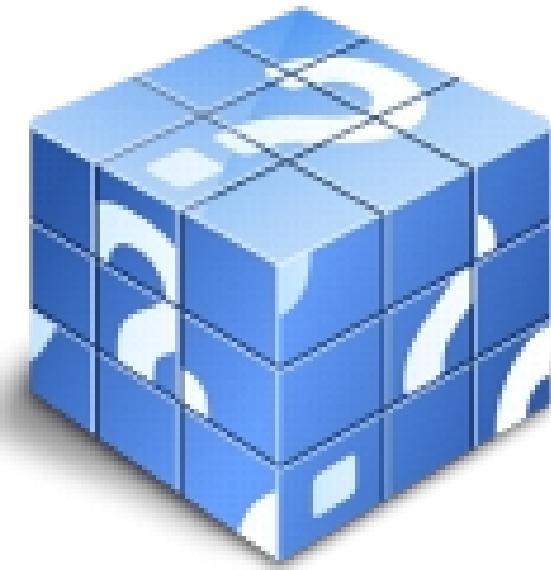
```
@NotifyPage(page="page",  
            topics="region")  
public class Pessoa { ... }
```

The background features a large, glowing orange planet with a textured surface and a bright yellow/orange ring of debris or dust particles trailing off to the right. Below it is a smaller, dark blue planet with a visible horizon and a thin blue ring. The overall scene is set against a dark, star-filled space.

**IMPACT**



# Metadata



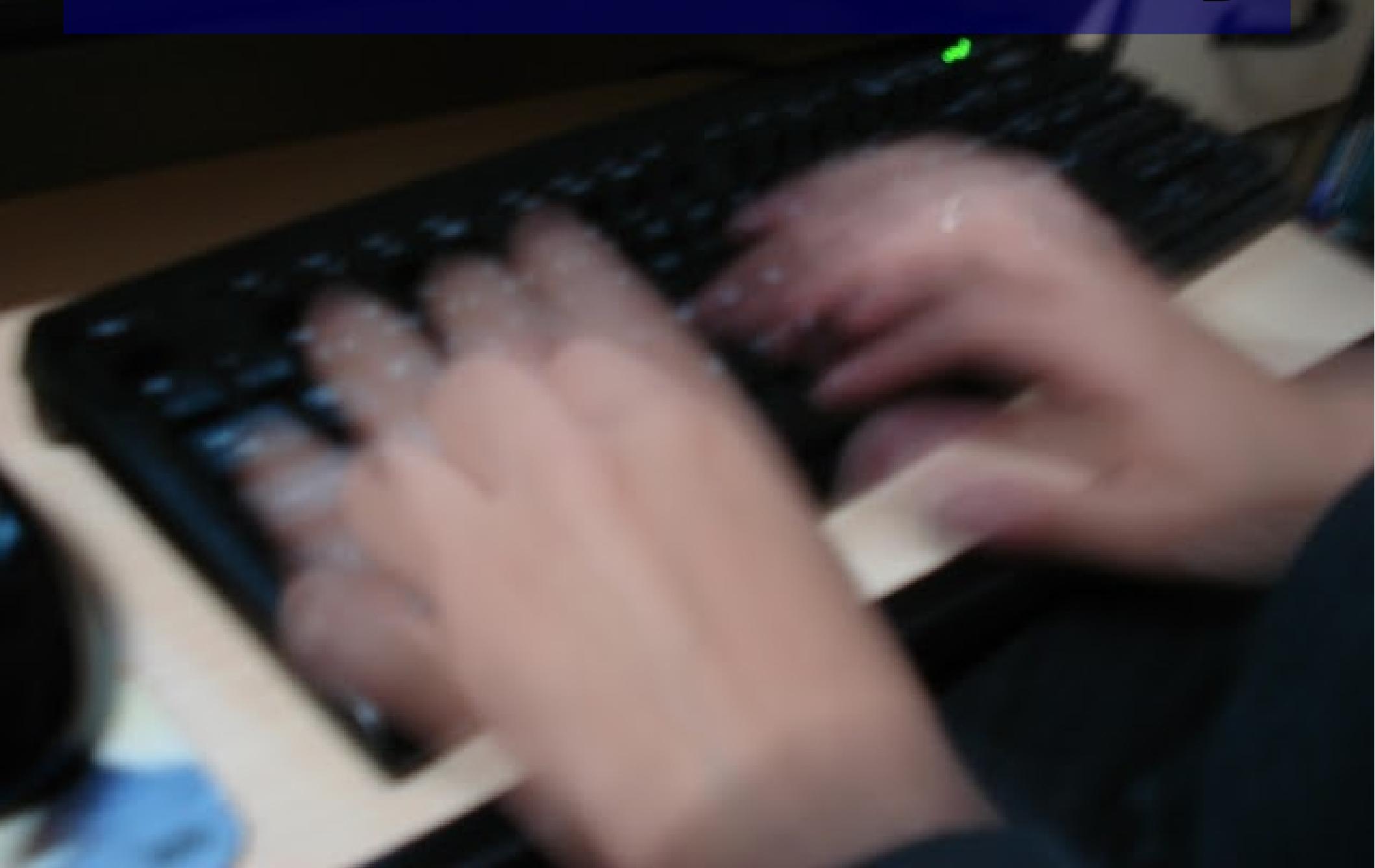
# Aspects

# **Application Domain**



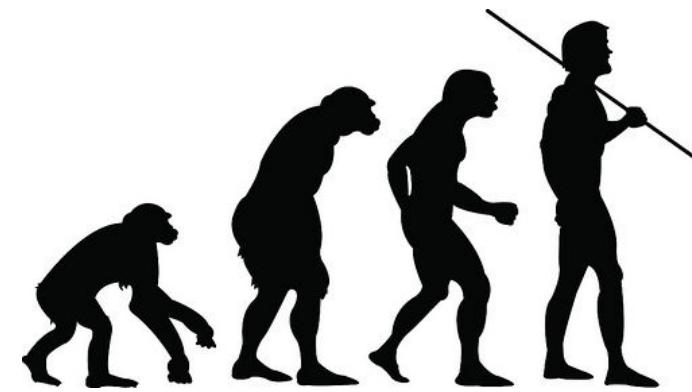
# **Architecture**

# Increased Productivity



**1**

*Reverse AJAX  
Instance Validation*



**2**

*Granular Access Control  
Conditional Reverse AJAX  
Web Service Integration (1 app)*

**3**

*Auditing  
+ Web Service Integration (2 app)*

*“Metadata is used by the  
developers as if they  
didn't know that there is  
an aspect processing them”*