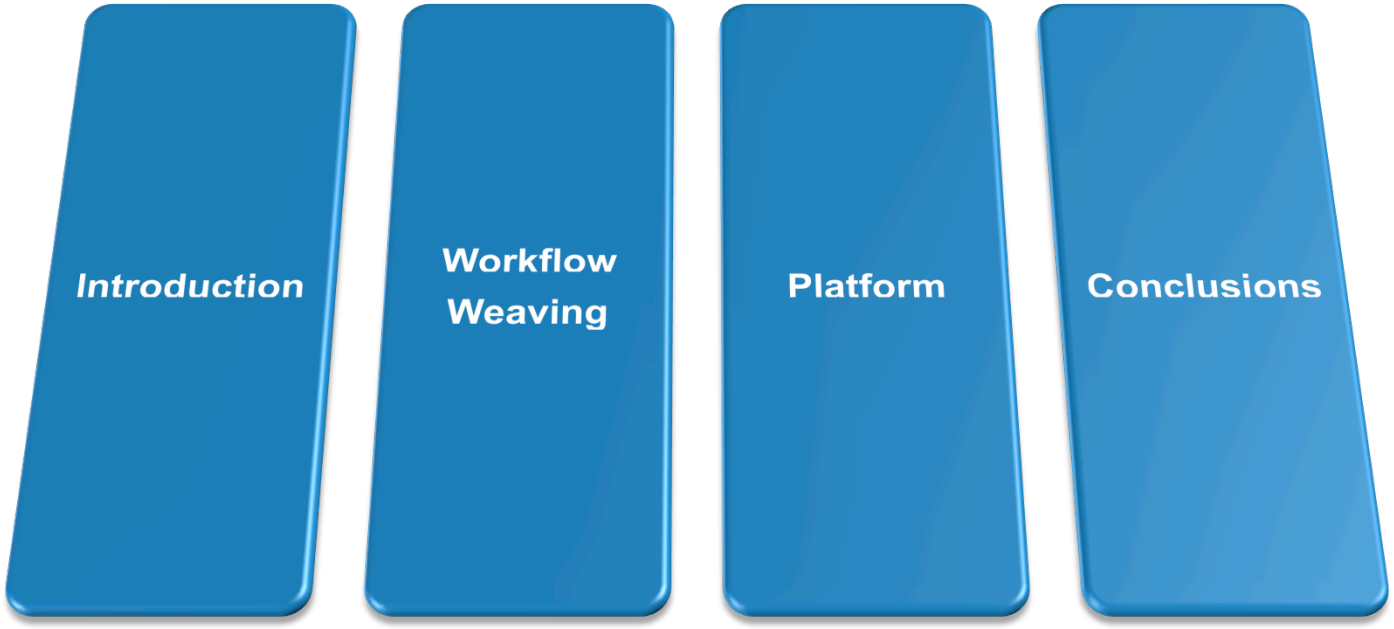
Implicit BPM
Business Process Platform for Transparent Workflow Weaving

Rubén Mondéjar, Pedro García, Carles Pairot, and Enric Brull

BPM Round Table
Tarragona

Contents



Context

- Building **applications** from the **ground up** is **no** longer an **acceptable** business **practice** and it is certainly **not cost effective**

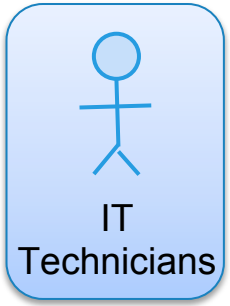
- **Current** application **exploitation** demands a **high** degree of **integration,** which **implies** a **costly connection** development **work** in every **software** piece

BPM

- Can be seen as a **mechanism** for **integrating** systems and **developing** new **applications**
- A **discipline** where **business** and **technical** users should work **together**
- However, **business analysts** are **not aware** of the end **applications** or **services**, and their **interoperations**

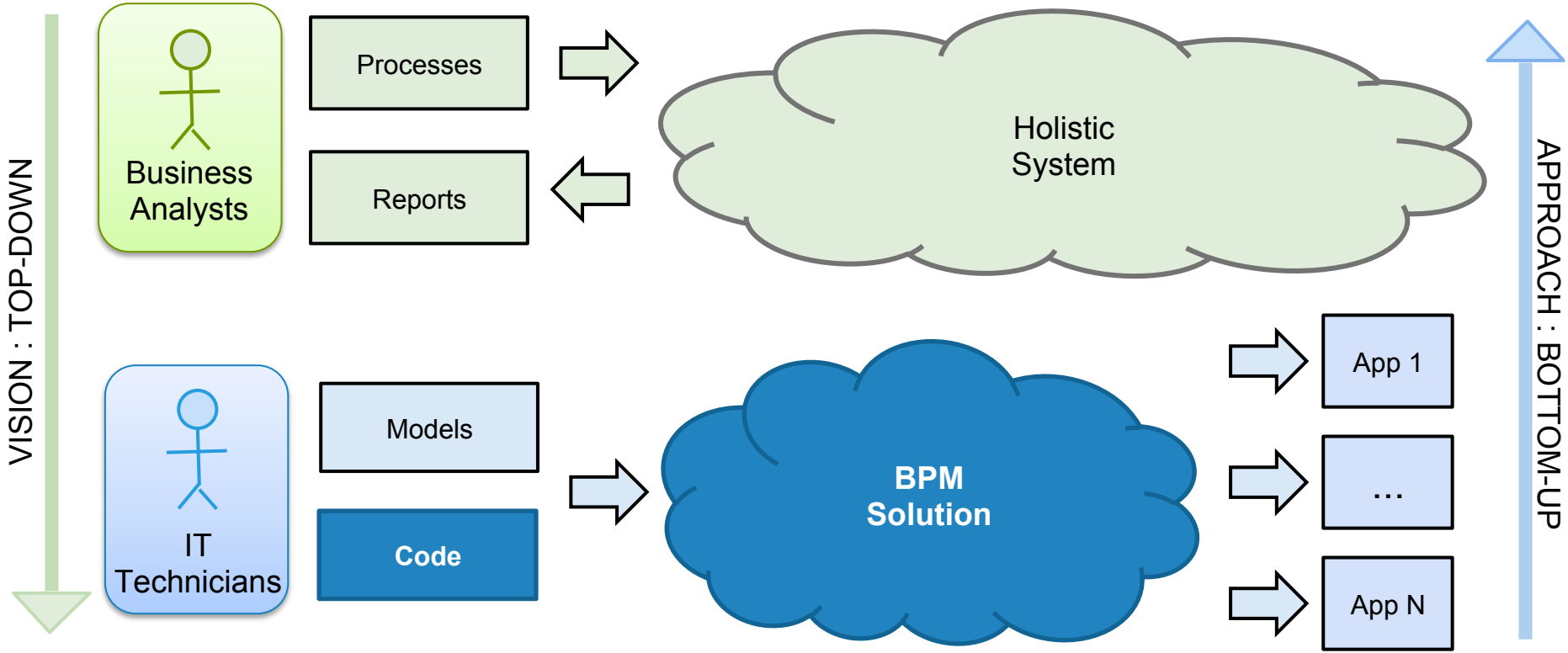
Key Questions

Are we able to **automate processes** and directly install and **execute** them in a **holistic** system?



How can we **integrate** business **processes transparently** into the organizational **applications**?

Conceptual View



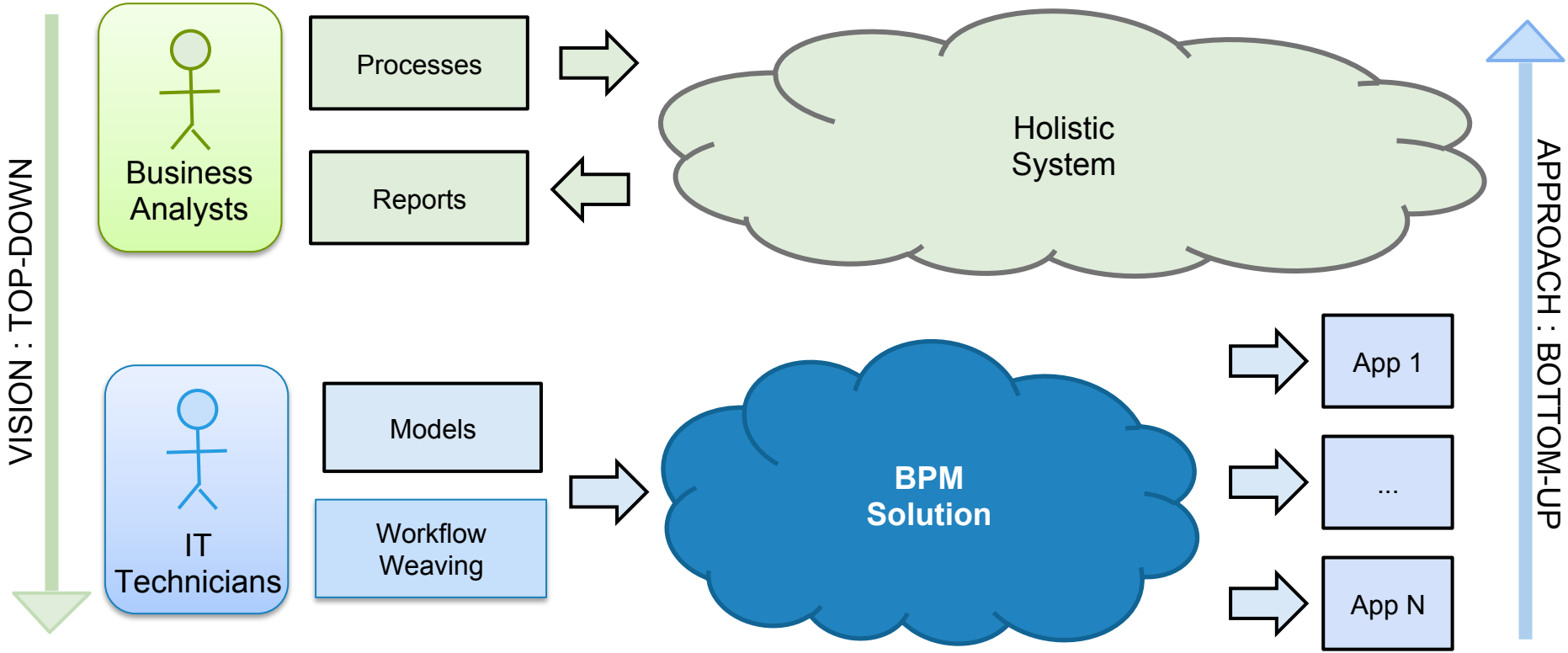
Explicit Solutions

- Traditional **BPM** approaches are well-known to **implement workflows** on top of **software systems** applicable to a **business**
- **Supporting** the **separation** of the business **process** from the core **application**, but **not transparently** and presenting important **drawbacks**
- Can be **perceived** as being **expensive** and really **complex** to deploy

Drawbacks

- Understanding existing **legacy code** through **reengineering** is a **challenging** task that may **consume** a lot of **resources**
- Some **recent works** propose **BPM reengineering** techniques
- It is **not easy** to apply **without tools** that help to **understand** the application **behaviour**

Proposal


VISION : TOP-DOWN
Business Analysts
Processes
Reports
Holistic System
APPROACH : BOTTOM-UP
IT Technicians
Models
Workflow Weaving
BPM Solution
App 1
...
App N

Workflow Weaving

- Enabling **integration** of business **processes** with **heterogeneous** corporate web **applications transparently**
- Avoiding **access**, **modification**, and detailed **knowledge** of the source **code** of the targeted **applications**
- Provides an **intuitive notation** to both **IT technician** and **business analyst** users to **represent** complex **integration** semantics and interactions

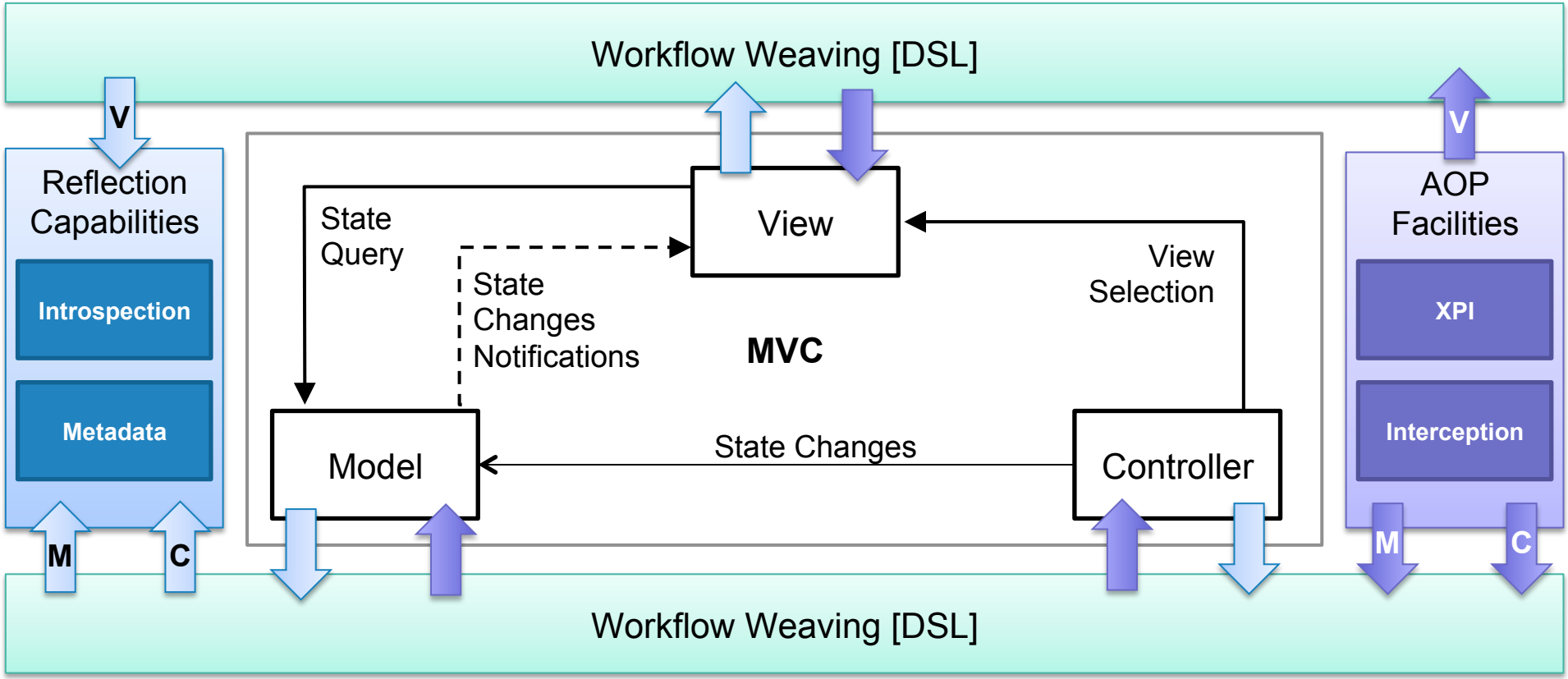
Contributions

1) **MVC blackboxing**
   - wrapping the MVC **pattern** used by **modern** web **applications**

2) **AOP/XPI approach**
   - a clean and **generic** code **introspection** and **interception** of **any** MVC **framework**

3) **DSL abstraction**
   - a **high-level** domain **language** to simplify the **integration** specification

Diagram


Workflow Weaving [DSL]
V
Reflection Capabilities
Introspection
Metadata
M
C
State Query
View
State Changes Notifications
MVC
View Selection
Model
State Changes
Controller
V
AOP Facilities
XPI
Interception
M
C
Workflow Weaving [DSL]

Aspects

- The **interception** is performed whenever the **code** execution **reaches** an aspect **pointcut**
- Consequently, an aspect **advice** is executed (**before**, **during**, or **after**)
- Some **AOP facilities** allow dynamic **weaving** of crosscutting concerns in **load-time** or **runtime**

XPI Solution

- **Each** MVC **implementation** requires its **own pointcuts**

➢ **Crosscut Programming Interfaces** (XPI) are explicit interfaces that specify **pointcut declarations**

✓ Providing a **clear abstraction** between the **aspects** in the **platform** and the intercepted **MVC framework**

MVC XPI

```
public abstract aspect MvcXpi {

    public pointcut inModel(): within(*.persistence.Entity);
    public pointcut saveMethod(): inModel() &&
                            execution(public Object save( . .)) ;
    //More CRUD Methods (...)
```

M

```
    public pointcut inView(): within(*..gsp * gsp);
```

V

```
    public pointcut inController(): within(*..*Controller);
    public pointcut controllerAction(): inController() &&
                                execution(@Action public * *.*(..));
}
```

C

DSL Approach

- **AOP** paradigm has **not** been **widely adopted** due to its inherent **complexity**

➢ A **simple** and **understandable** human readable **language** is required to **abstract** the **interception** facilities

✓ Our **specification** provides the manner to **formalize** an abstract **descriptor** for the **Workflow Weaving technique**

DSL Grammar

```
DSL = string '{' {weaver} '}' ;
weaver = in application ':' {act ',' behaviour} '; ' ;
application = characters;
act = when variable element [from controller ] ;
when = Before | Instead of | After ;
variable = '"'characters'"' ;
element = action | view | event | task | domain | attribute ;
controller = characters ;
behaviour = connector variable element [ from controller ]
            [ by variable ] [{and behaviour}] ;
connector = perform | find | save | render |
            trigger | start | sets in ;
```

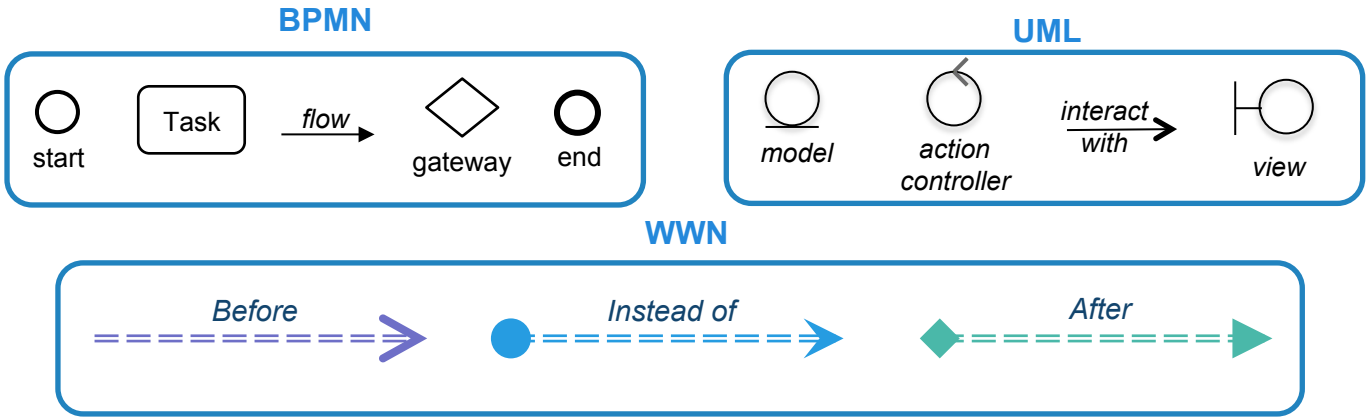
ACT

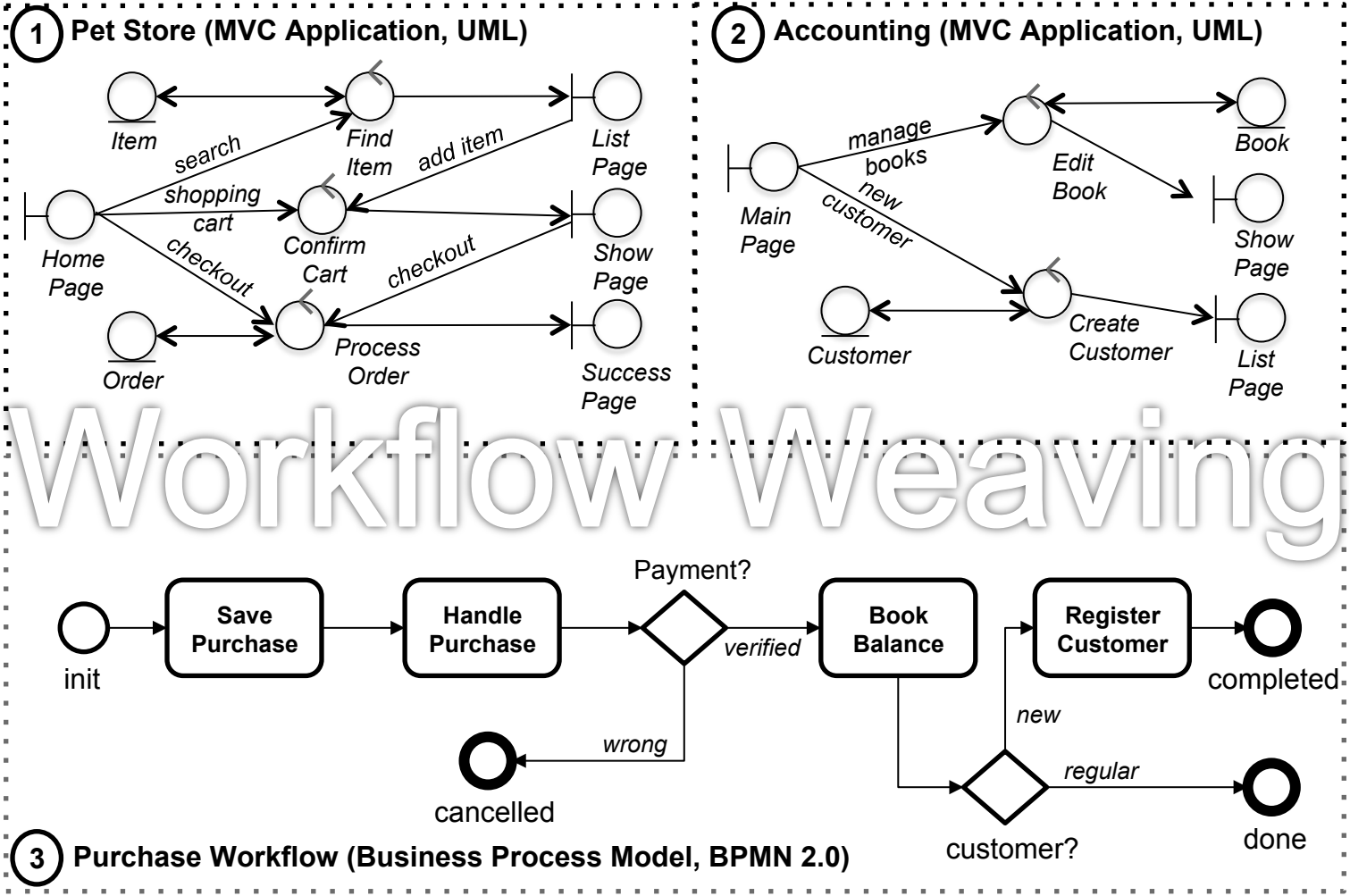
BHS

WEAVER

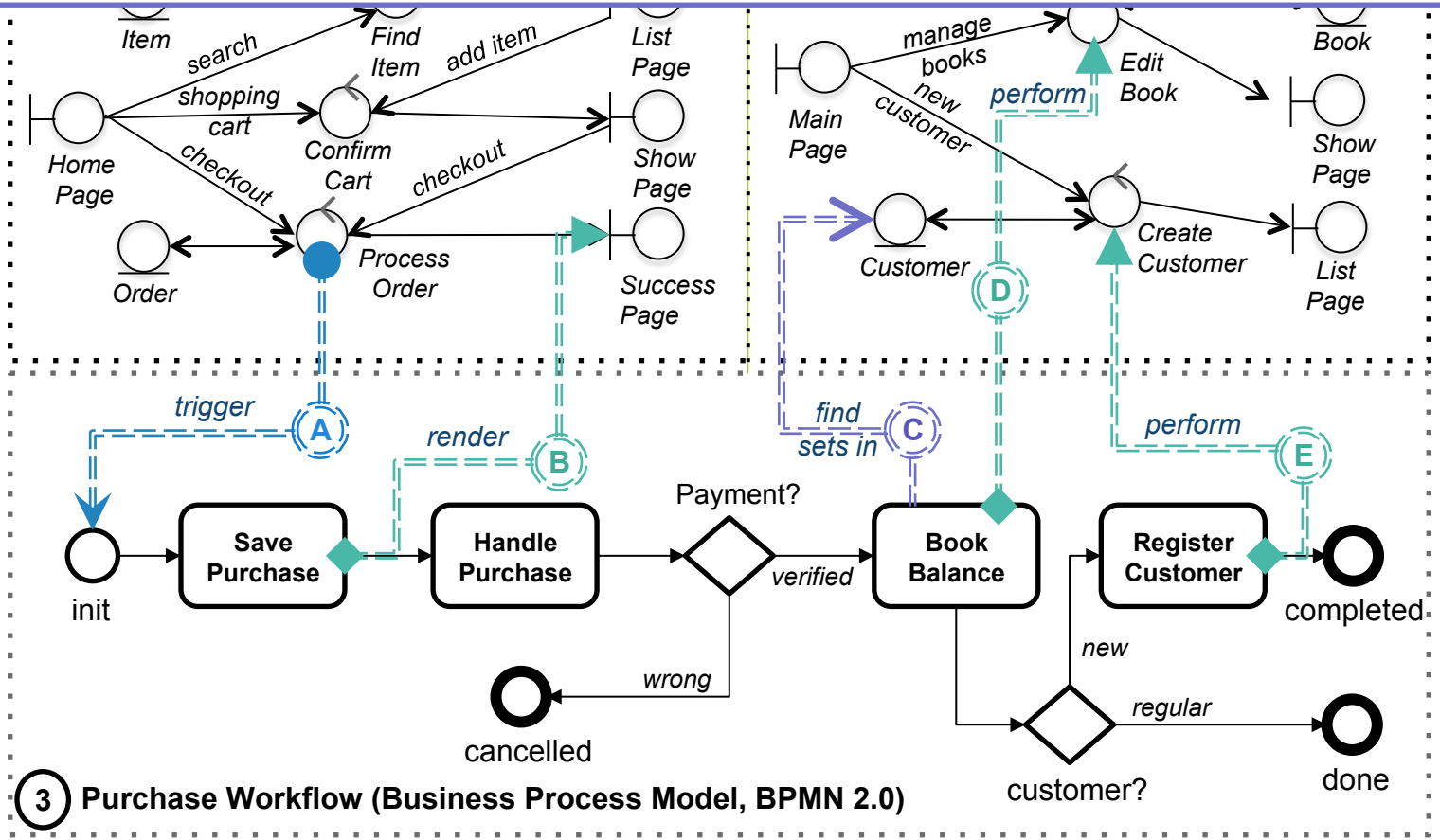
Graphical Notation

- **BPMN** is the **standard** for specifying **business processes models**
- **UML** notation to illustrate the **MVC components** and **interactions** among the **applications**
- **Workflow Weaving Notation (WWN)** uses **dashed arrows** that represent **interception** and **reaction** between **processes** and **applications**




1 Pet Store (MVC Application, UML)
Item
search
Find Item
add item
List Page
shopping cart
Home Page
Confirm Cart
Show Page
checkout
checkout
Order
Process Order
Success Page
2 Accounting (MVC Application, UML)
manage books
Book
Edit Book
Main Page
new customer
Show Page
Customer
Create Customer
List Page
Workflow Weaving
Payment?
Save Purchase
Handle Purchase
verified
Book Balance
Register Customer
completed
init
wrong
new
regular
cancelled
customer?
done
3 Purchase Workflow (Business Process Model, BPMN 2.0)

```
(C) Before "Book Balance" task, find "customer" domain by "nin"
                                        and sets in "exists" attribute;
```

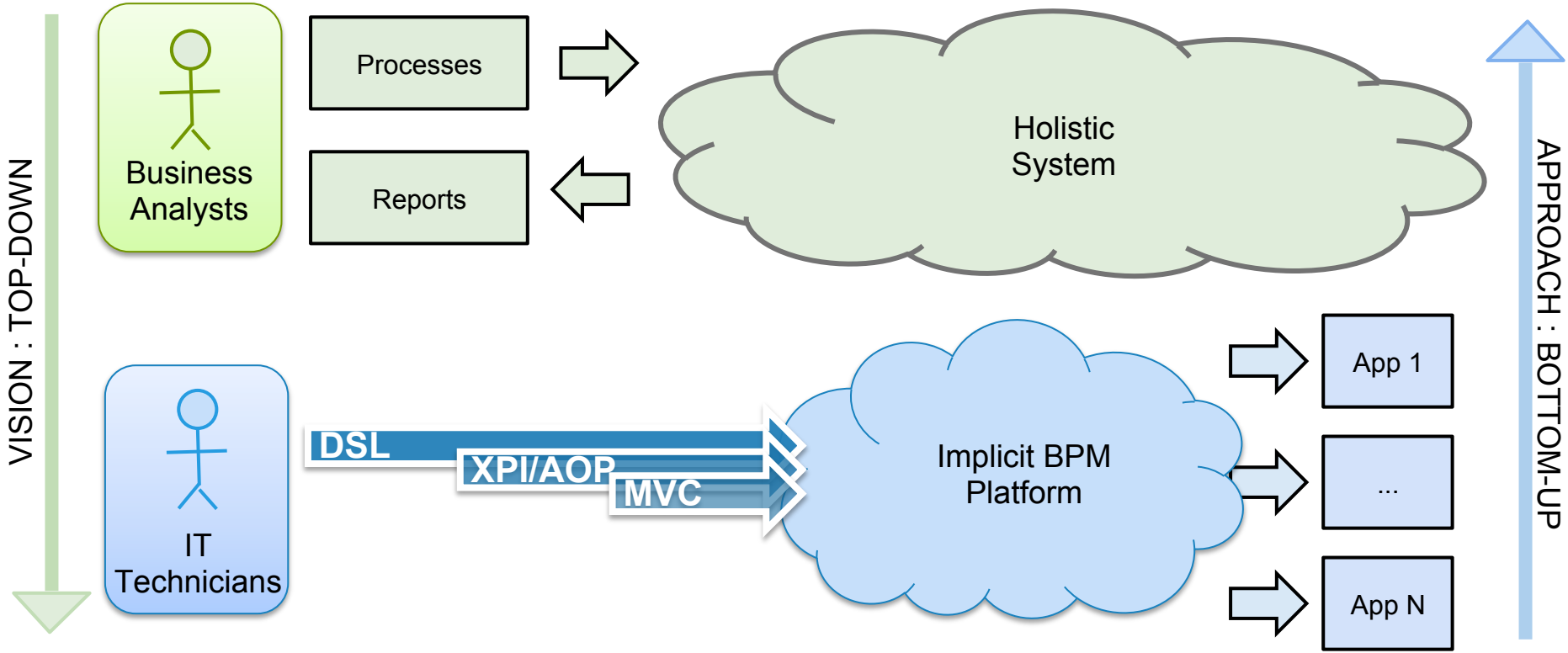


Use Case DSL

```
PurchaseWorkflow {
  in PetStore :
  A Instead of "process" action from Order, trigger "init" event;
  B After "Save Purchase" task, render "success" view;

  in Accounting :
  C Before "Book Balance" task, find "customer" domain by "nin"
                                and sets in "exists" attribute;
  D After "Book Balance" task, perform "update" action from Book;
  E After "Register Customer" task, perform "create" action from Customer;
}
```

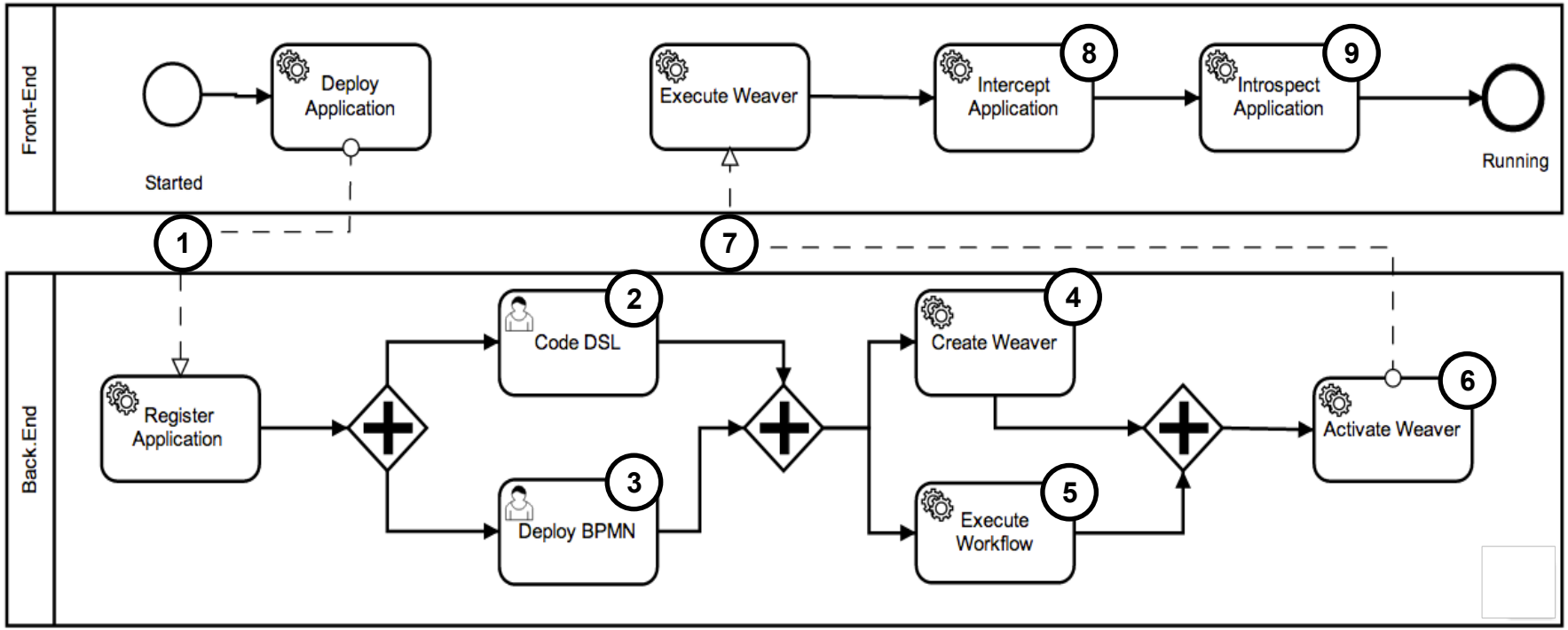
Platform


VISION : TOP-DOWN
Business Analysts
Processes
Reports
Holistic System
APPROACH : BOTTOM-UP
IT Technicians
DSL
XPI/AOP
MVC
Implicit BPM Platform
App 1
...
App N

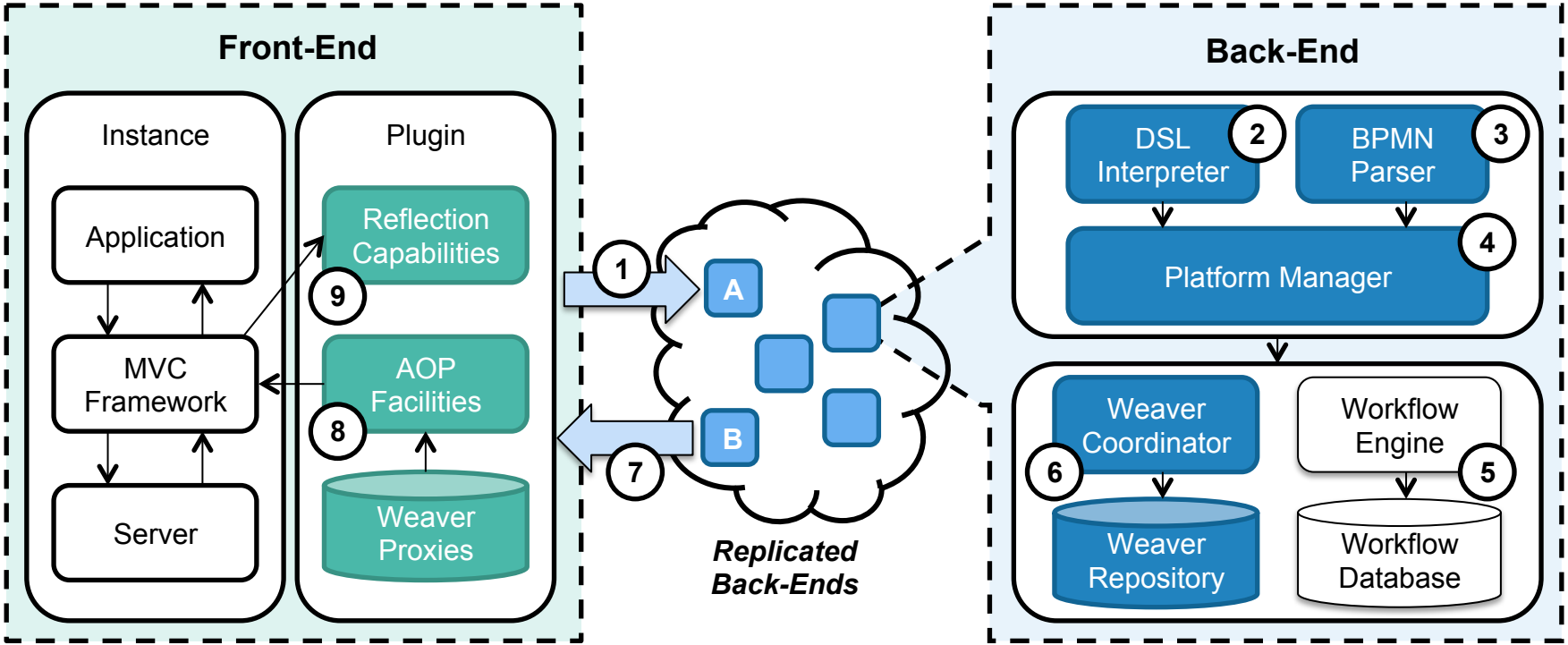
Implicit BPM

- Our **platform** has already been implemented as a **distributed architecture**
- Which **consists** of two separate parts: the **Front-End** and the **Back-End** systems
- Each part is connected via **web standard** mechanisms **exposing** its own **API** to allow **remote** method **invocation**

Life Cycle


Front-End
Deploy Application
Started
Execute Weaver
Intercept Application
8
Introspect Application
9
Running
1
7
Back.End
Register Application
Code DSL
2
Deploy BPMN
3
Create Weaver
4
Execute Workflow
5
Activate Weaver
6

Architecture



Distributed Weavers

- Acts are **detected** and **intercepted** by the **AOP facilities**
- **Locally** (Front-End) are launched via the **XPI pointcuts**
  - Example : `After "create" action from Item`
- **Remotely** (Back-End) are **notified** to the **applications**
  - Example : `Instead of "Confirm Request" task`

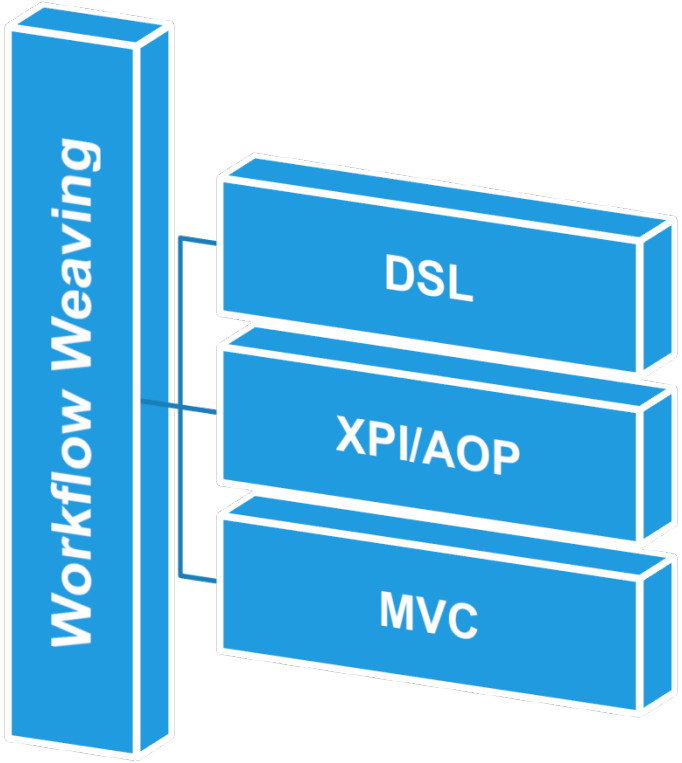
Previous Approaches

- Use **explicit** or **clear-box** techniques, requiring **detailed knowledge** of the **targeted** applications
- **Rewriting** systems or **reengineering** them for BPM **integration**
- These solutions **involve** expensive development **costs**

Presented Solution

- The **novelty** of the **Implicit BPM** is to perform a new **integration** technique of **processes** and **applications** transparently

- Since this **technique** presents such **high potential**, we plan to apply it in different **prospective fields**



Prototype

- **Available** at **http://implicit-bpm.sf.net** under a LGPL license

- Supported by **Open Source** projects

Future Work

- **DSL** Extensions
  - Example : Complex **Queries**
- **Real Uses** and Scenarios
  - Example : **ITIL v3**
- PaaS **Cloud** Deployment

Questions?

Implicit BPM: a Business Process Platform for Transparent Workflow Weaving