



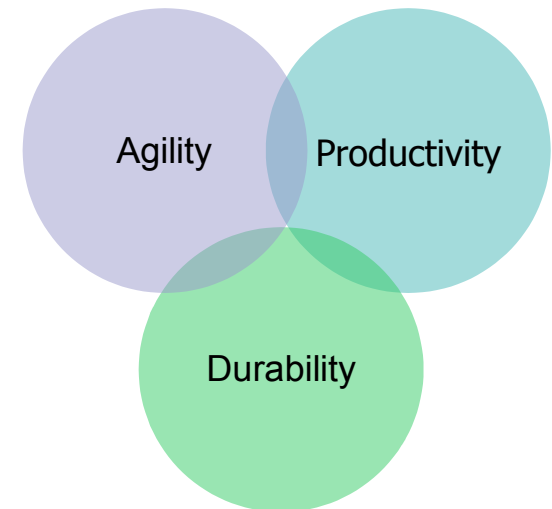
SOA & MDA

with



- Who / Why
- MDA and Acceleo
- Modeling SOA

- Let's switch to industrial tools and methods :
 - **Acceleo** : Industrialise software development
 - **Agility** : Retro-engineering & migration
- Obeo : software vendor and consultancy
- Our goals :
 - Create software factories for pragmatic solutions
 - Reduce the **gap** between theory and reality
- A real OpenSource business model



Some references

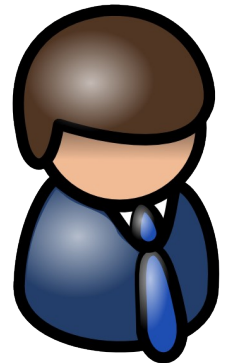


How to reduce **delay** of my projects ?

Where are my **up-do-date** specifications ?

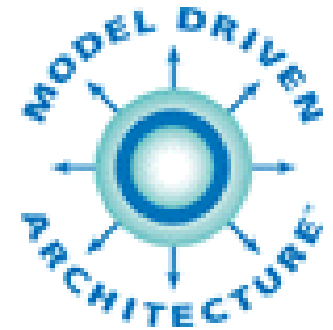
How **capitalise** the knowledge of my teams ?

How to be more **agile** with **evolutions** ?





- Innovate with an MDA approach
 - Design to provide a high level conception
 - **Automatise** projection model - code
 - Integrate inside existing process
 - Be simple and customized !!!!



Procedural



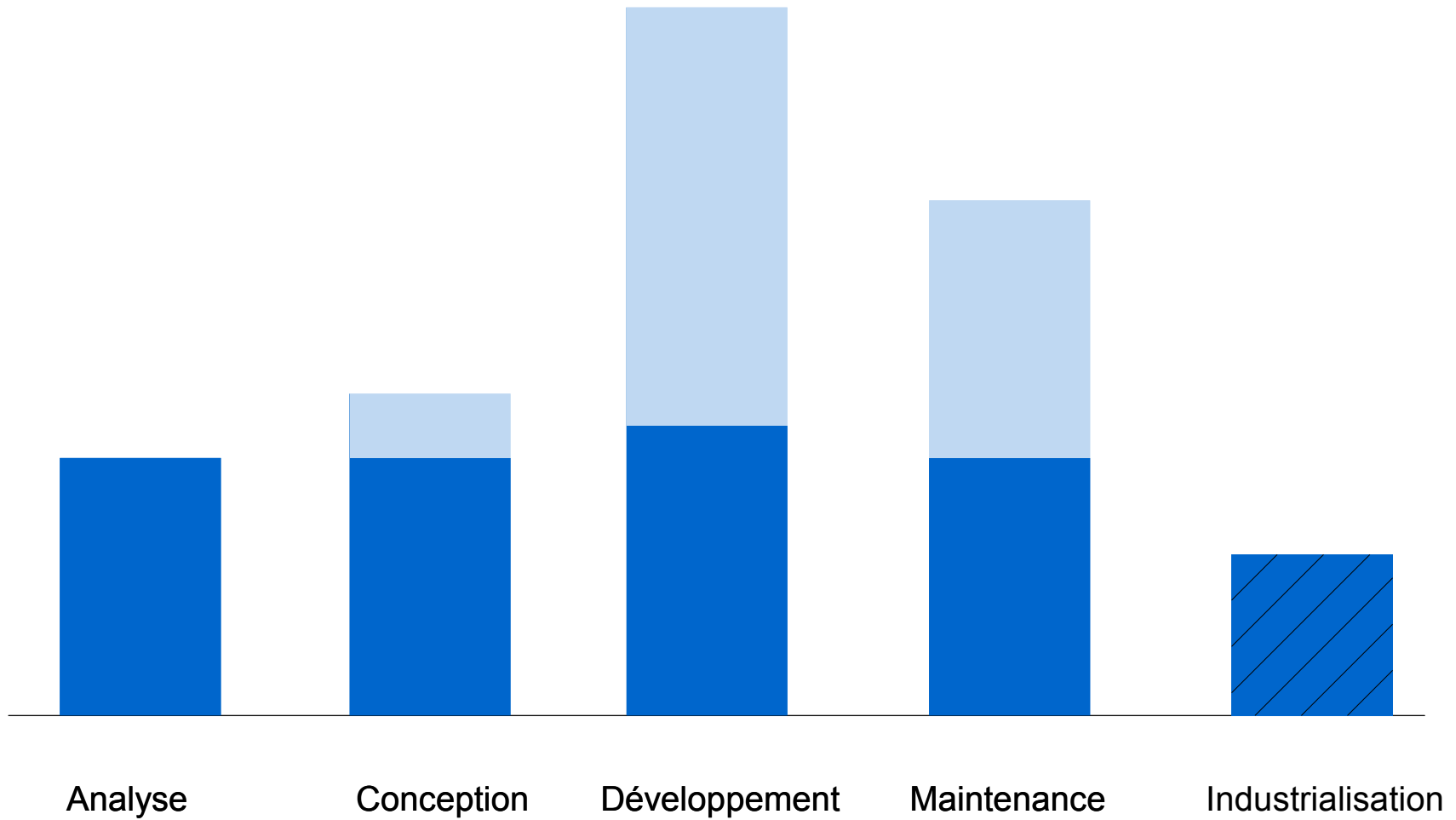
Object

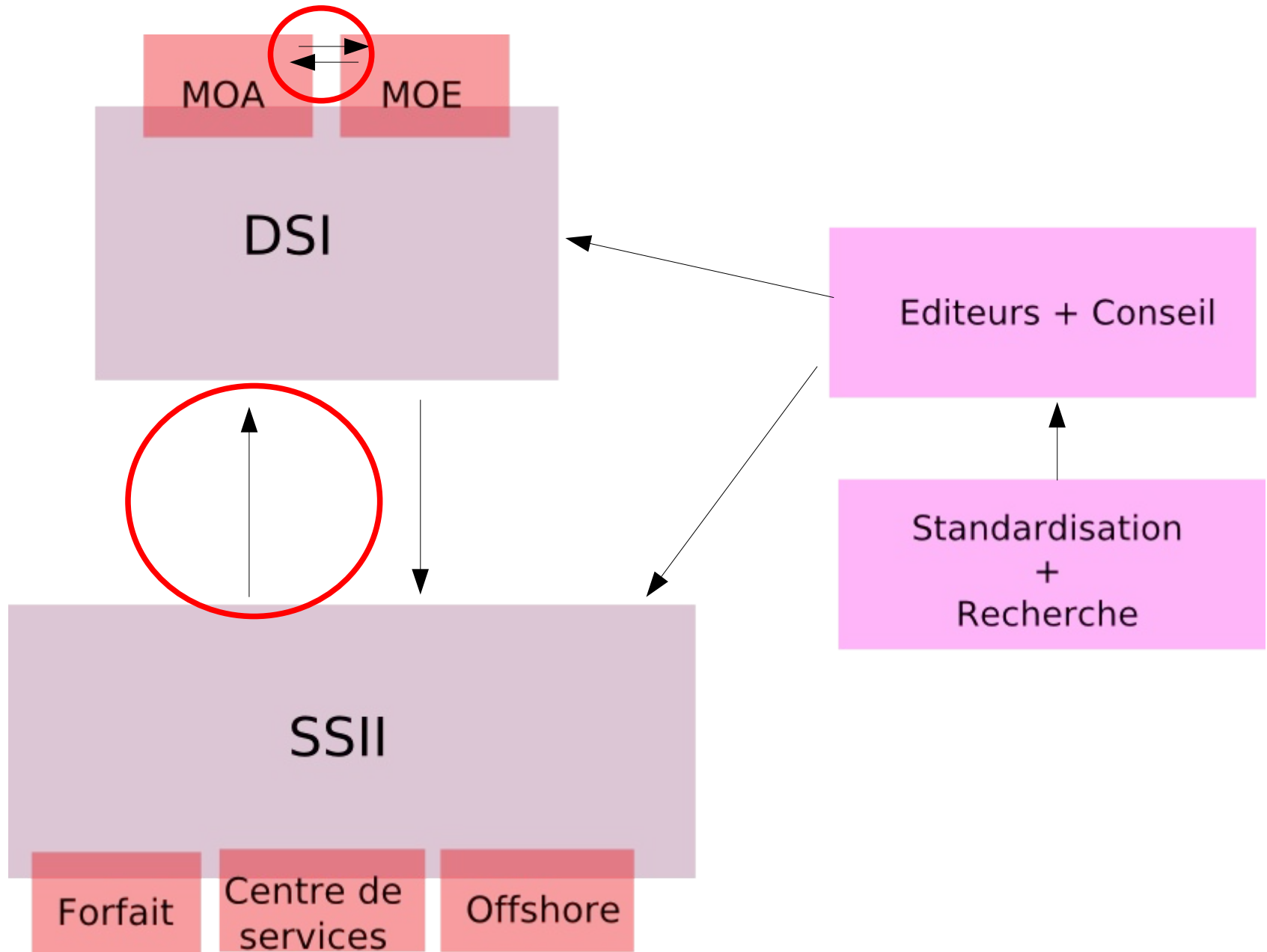


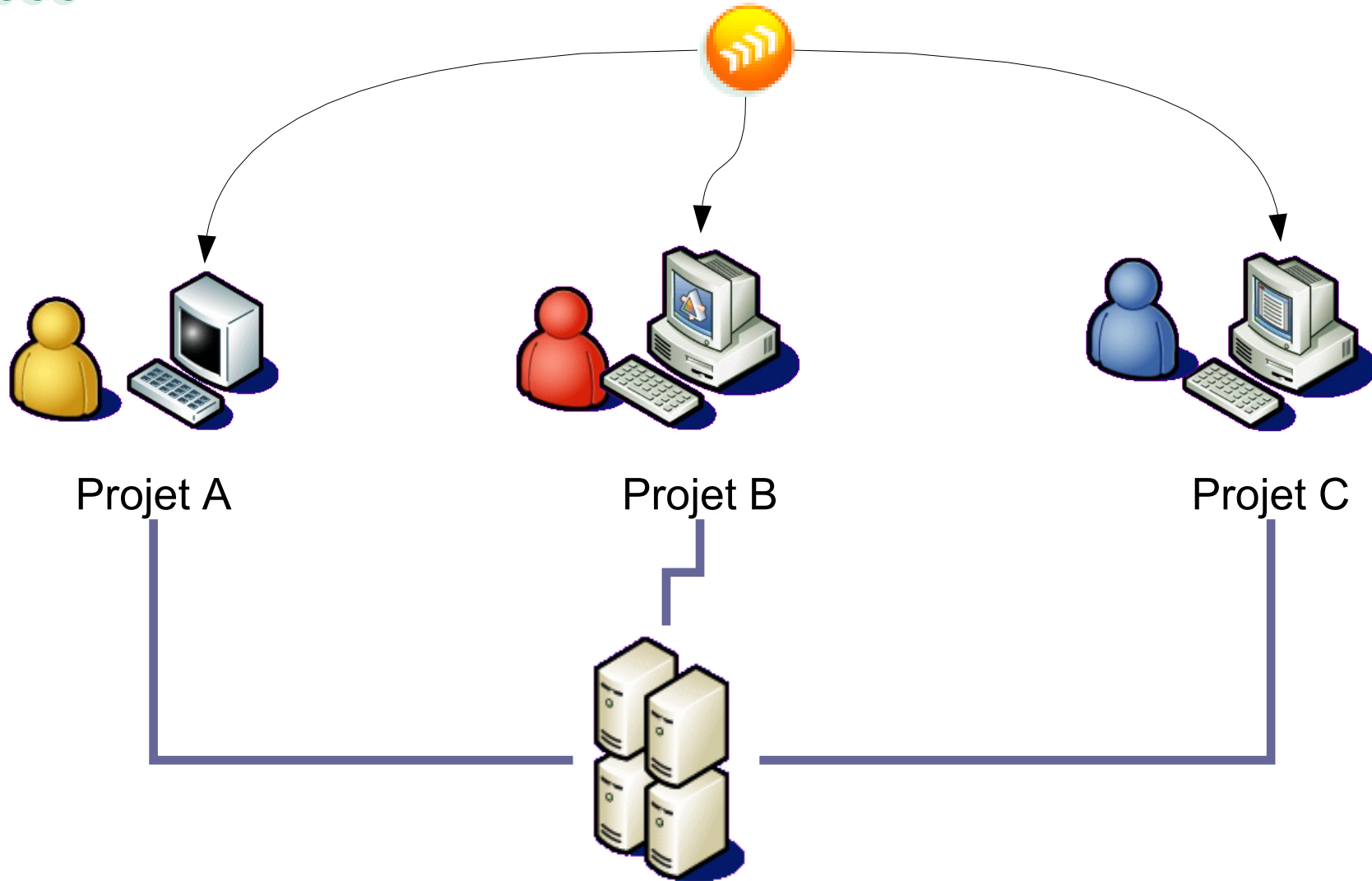
Component



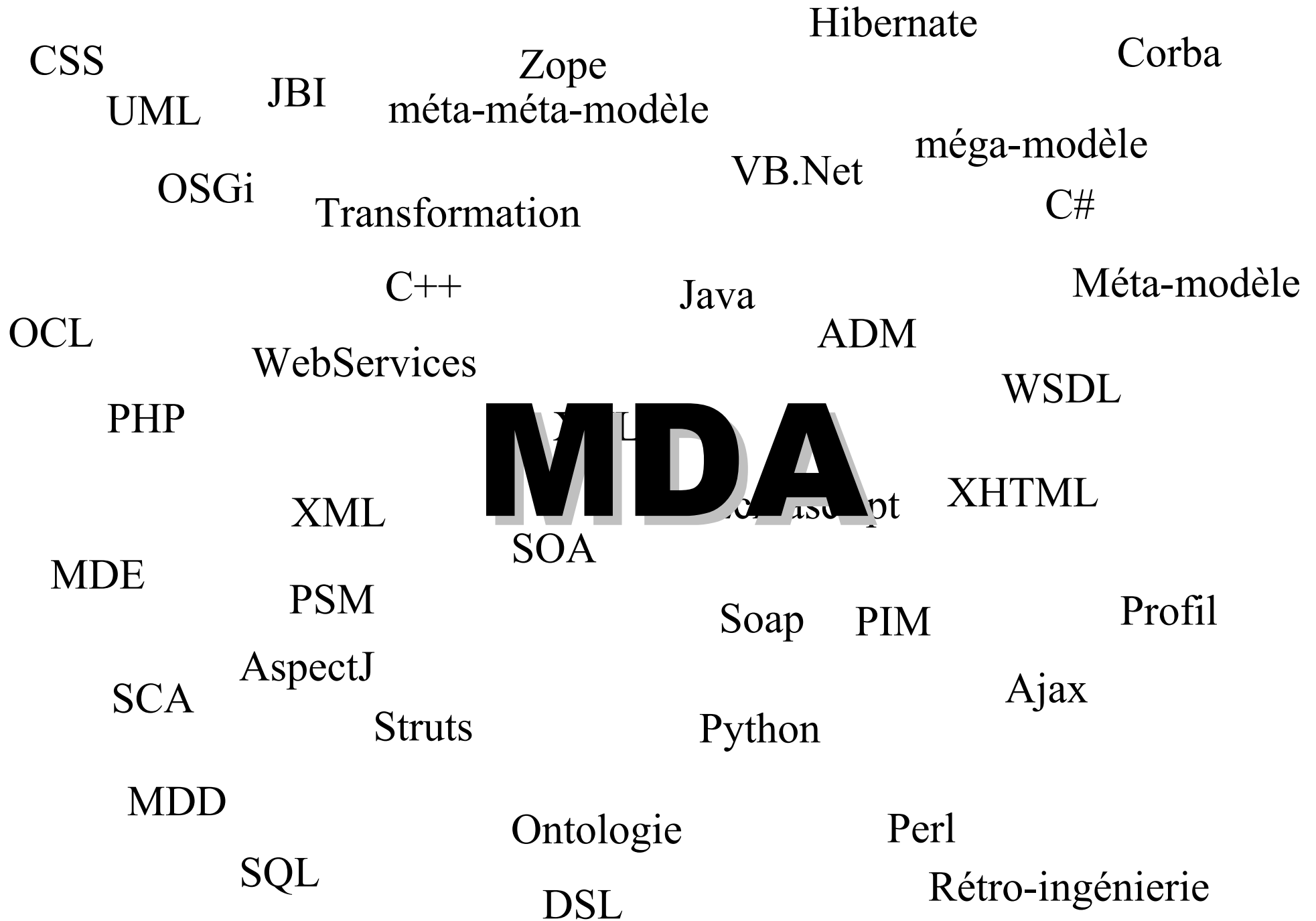
Model



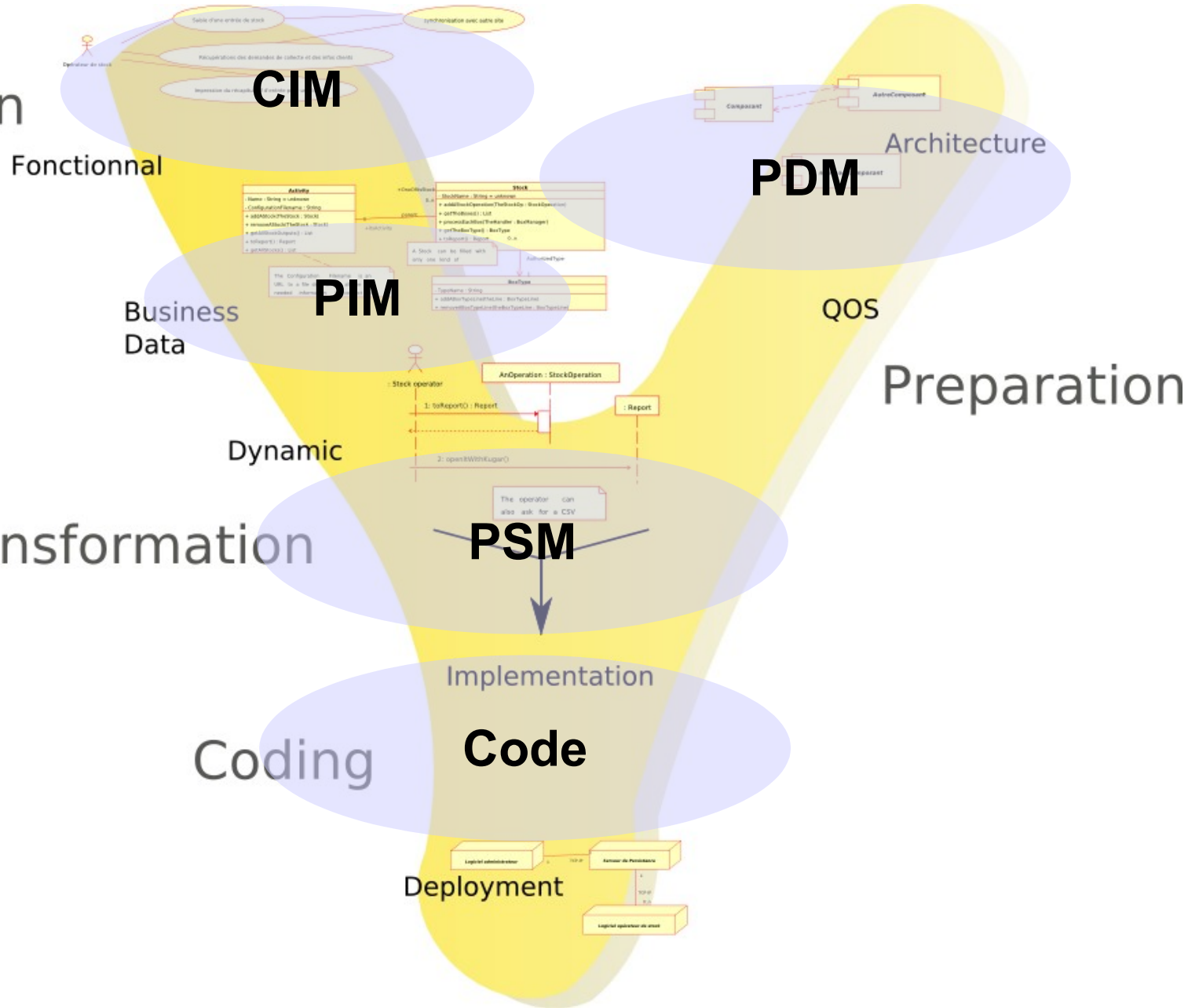




- Capitalisation => Reliability + Rationalisation + Evolutivity



Modelisation



Fonctionnel

CIM

PDM

Architecture

Business Data

PIM

QOS

Preparation

Dynamic

PSM

Transformation

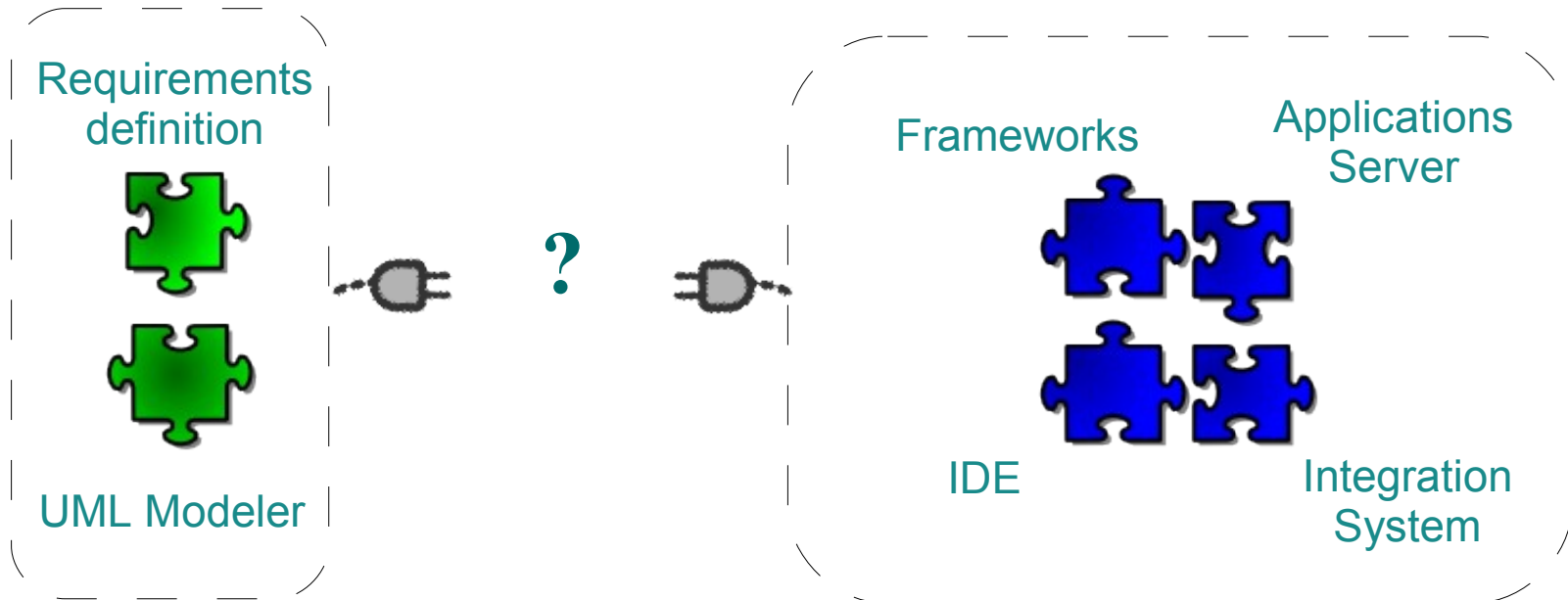
Implementation

Coding

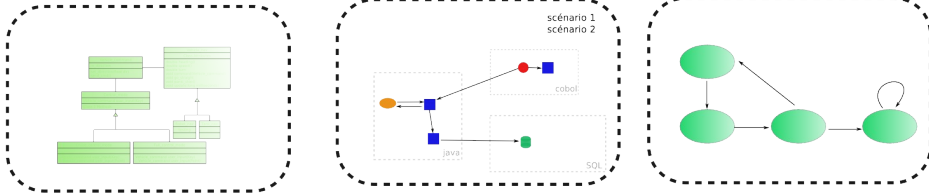
Code

Deployment

- 1 : Pragmatic
- 2 : Traceability
- 3 : Model Industrialisation

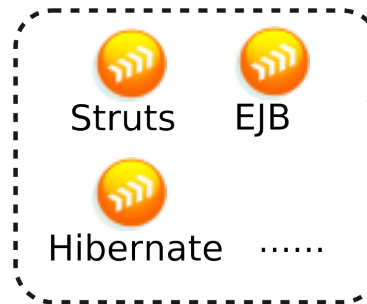


Modèles



Business Models
Agility of evolutions
Urbanisation view
Interoperability (UML, XMI, ...)

Modules



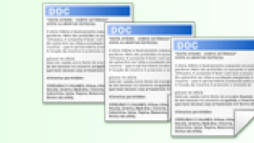
Durability of best practices
« White Box »
Design driven by architecture



Struts - JSF



Spring - EJB



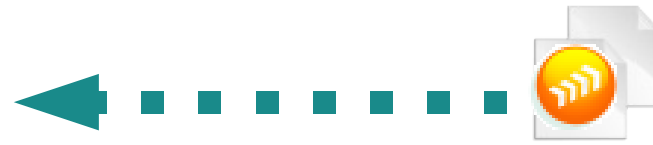
HIBERNATE

Massive generation (~70%)
Delay reduction (~30%)
Better quality

Class diagram



Struts module



Struts

Actions,
Forms



Tests
JUnit



Jsp,
Navigation

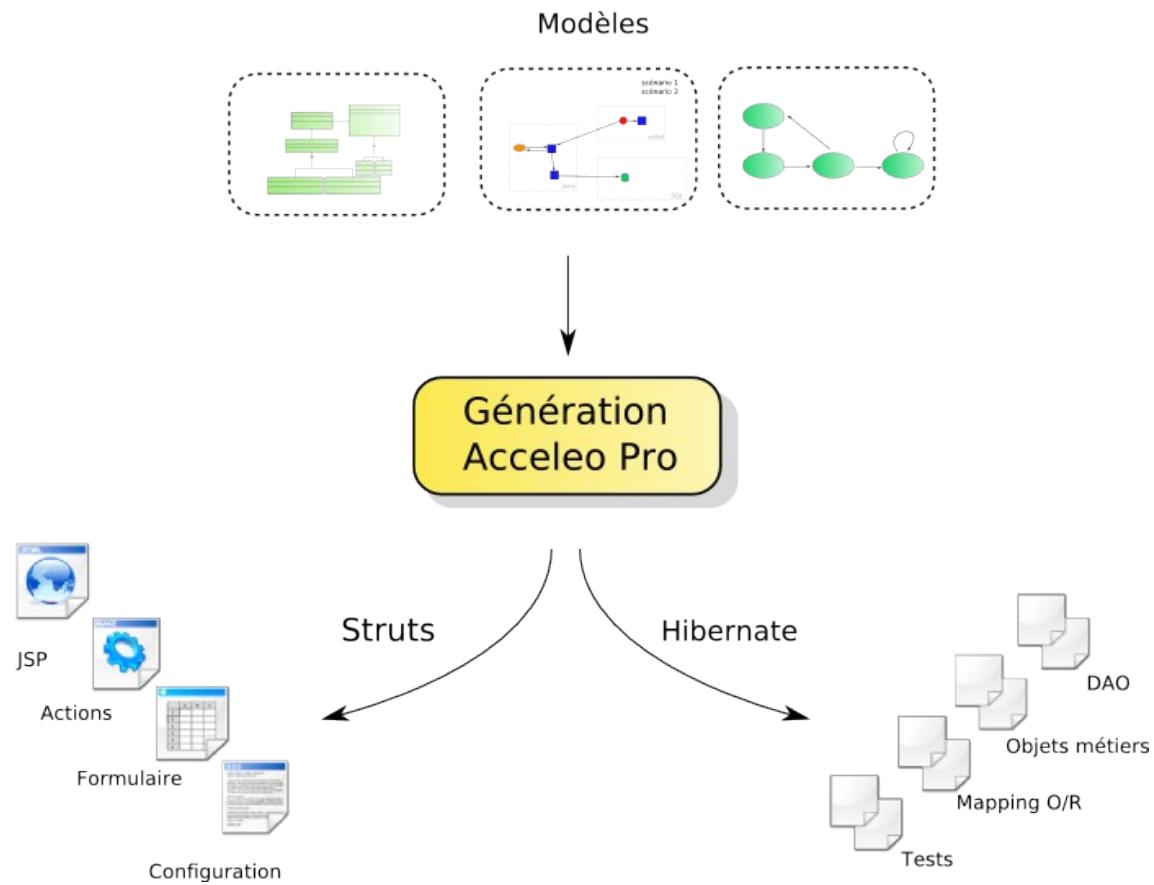


Struts-config,
tiles, validation...



1 UML class = 12 generated files

Demo (without SOA)



Acceleo Pro Traceability



Be **professional** with Acceleo !

Cahier des charges



- **Consistency** validation code / models
- Iterative process
- Impact analysis, round-tripping...



Modèle



Code
généré

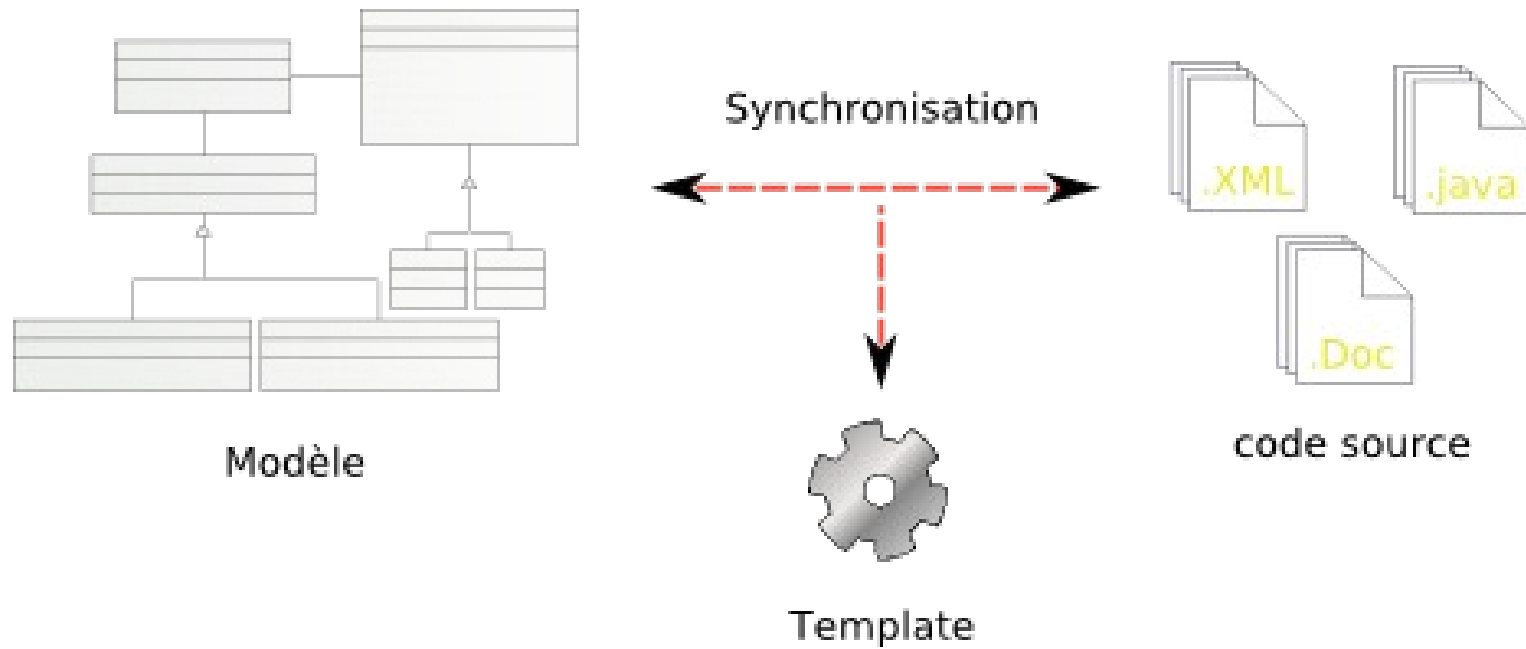


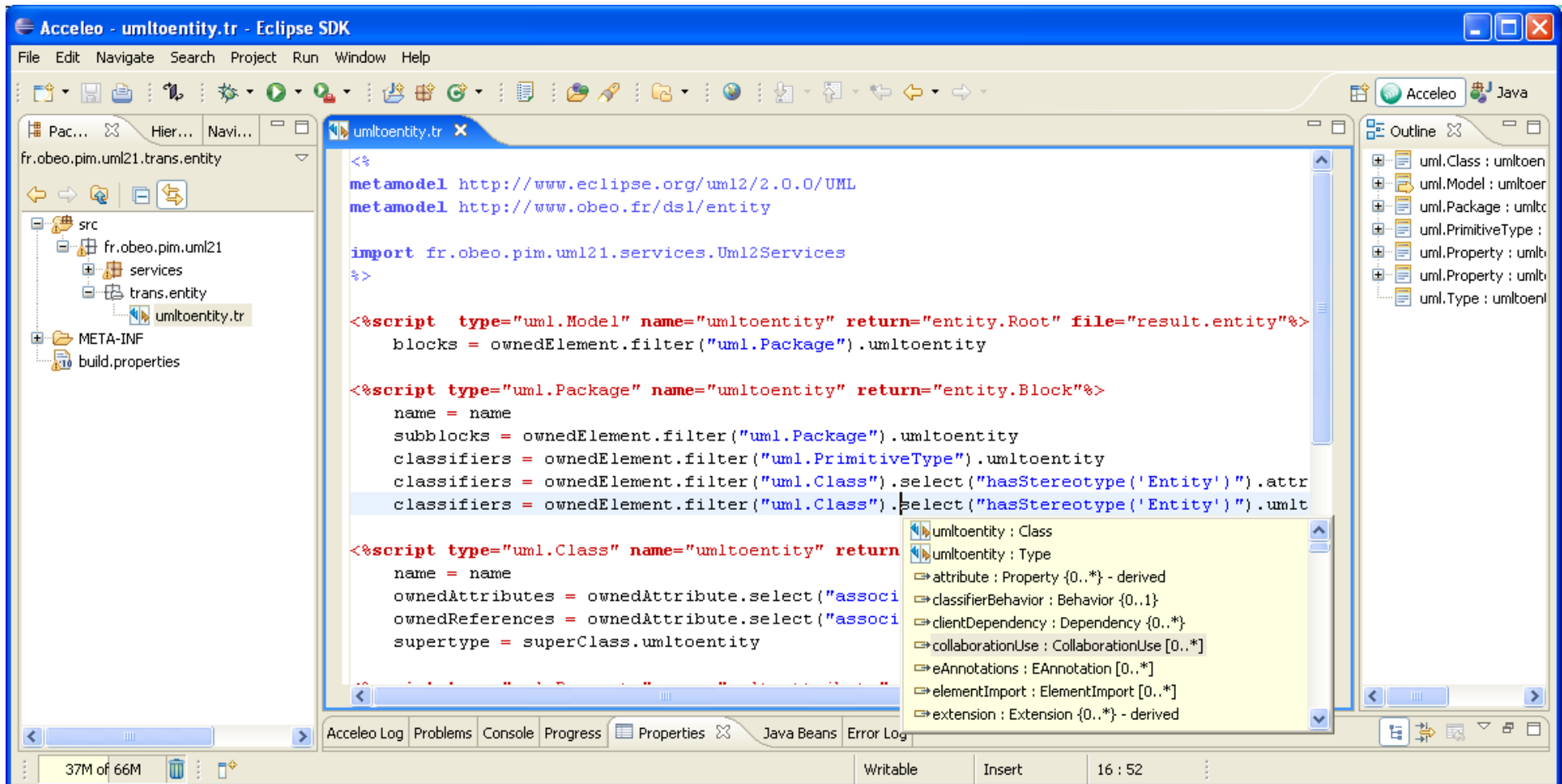
Code
complet

Modules
Acceleo

Frameworks

— *Acceleo Pro Traceability*



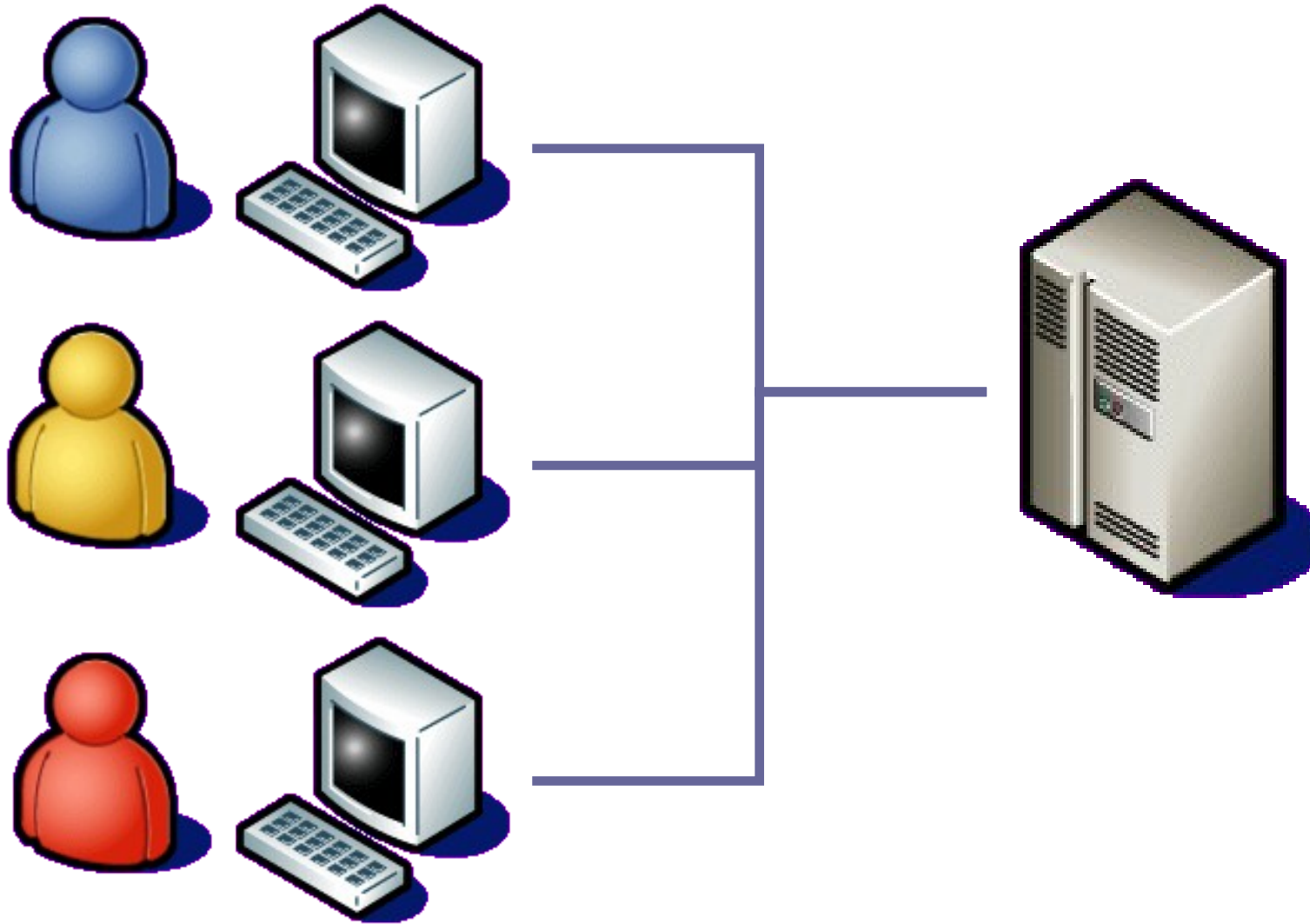


The screenshot shows the Eclipse IDE with the following components:

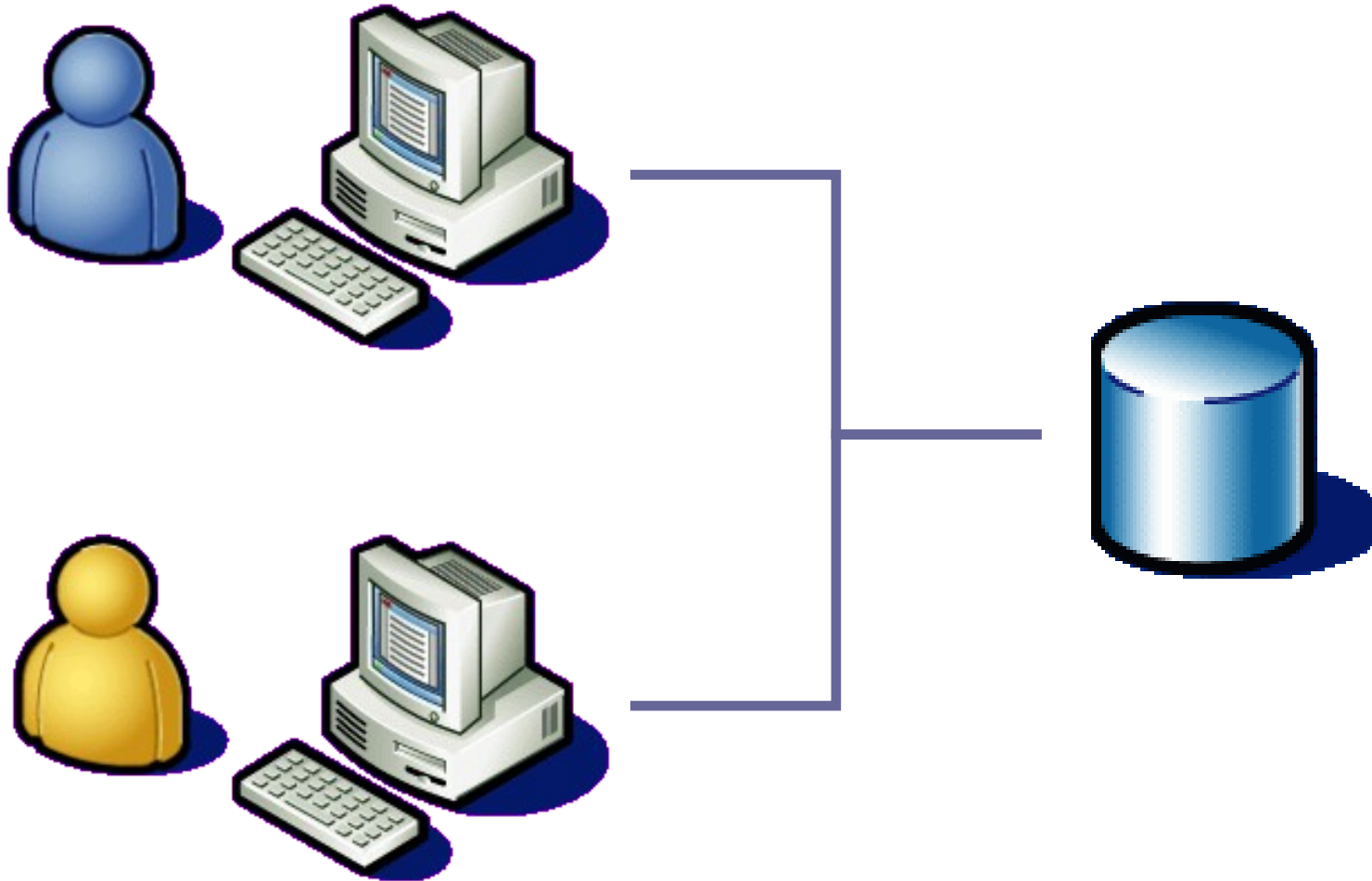
- Project Explorer:** Shows the project structure for `fr.obeo.pim.uml21.trans.entity`, including `src`, `services`, `trans.entity`, `umltoentity.tr`, `META-INF`, and `build.properties`.
- Editor:** Displays the Xtext DSL code for `umltoentity.tr`. The code defines a metamodel, imports services, and uses Xtext scripts to generate UML models, packages, and classes.
- Outline:** Lists the generated UML elements: `uml.Class : umltoen`, `uml.Model : umltoer`, `uml.Package : umlto`, `uml.PrimitiveType :`, `uml.Property : umlt`, and `uml.Type : umltoen`.
- Tooltip:** A tooltip is shown over the `umltoentity : Type` line, displaying its inheritance hierarchy and features:
 - `umltoentity : Class`
 - `umltoentity : Type`
 - `attribute : Property {0..*} - derived`
 - `classifierBehavior : Behavior {0..1}`
 - `clientDependency : Dependency {0..*}`
 - `collaborationUse : CollaborationUse [0..*]`
 - `eAnnotations : EAnnotation [0..*]`
 - `elementImport : ElementImport [0..*]`
 - `extension : Extension {0..*} - derived`
- Bottom Panel:** Shows the `Properties` view with the following values: `Writable`, `Insert`, and `16 : 52`.

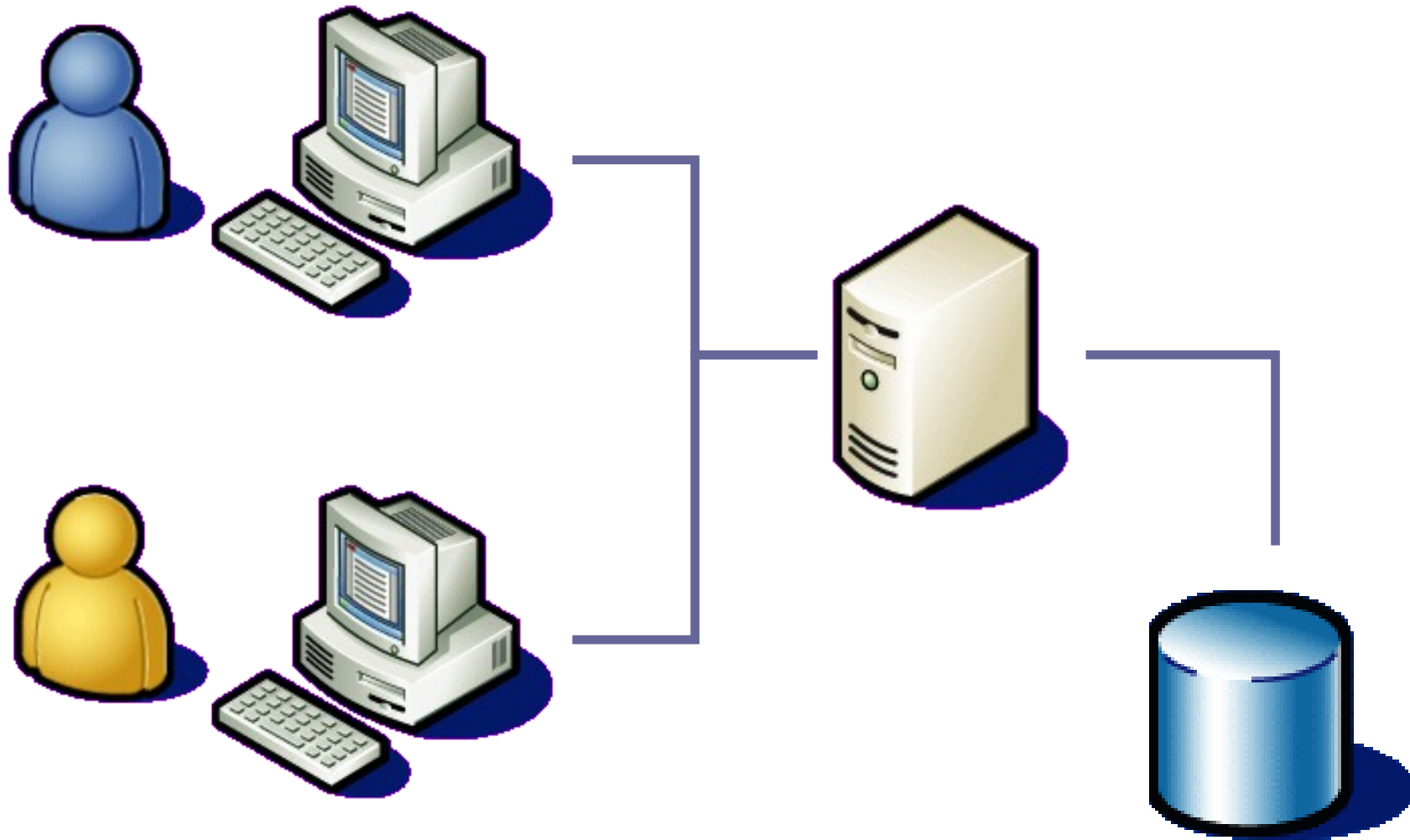
MDA and SOA

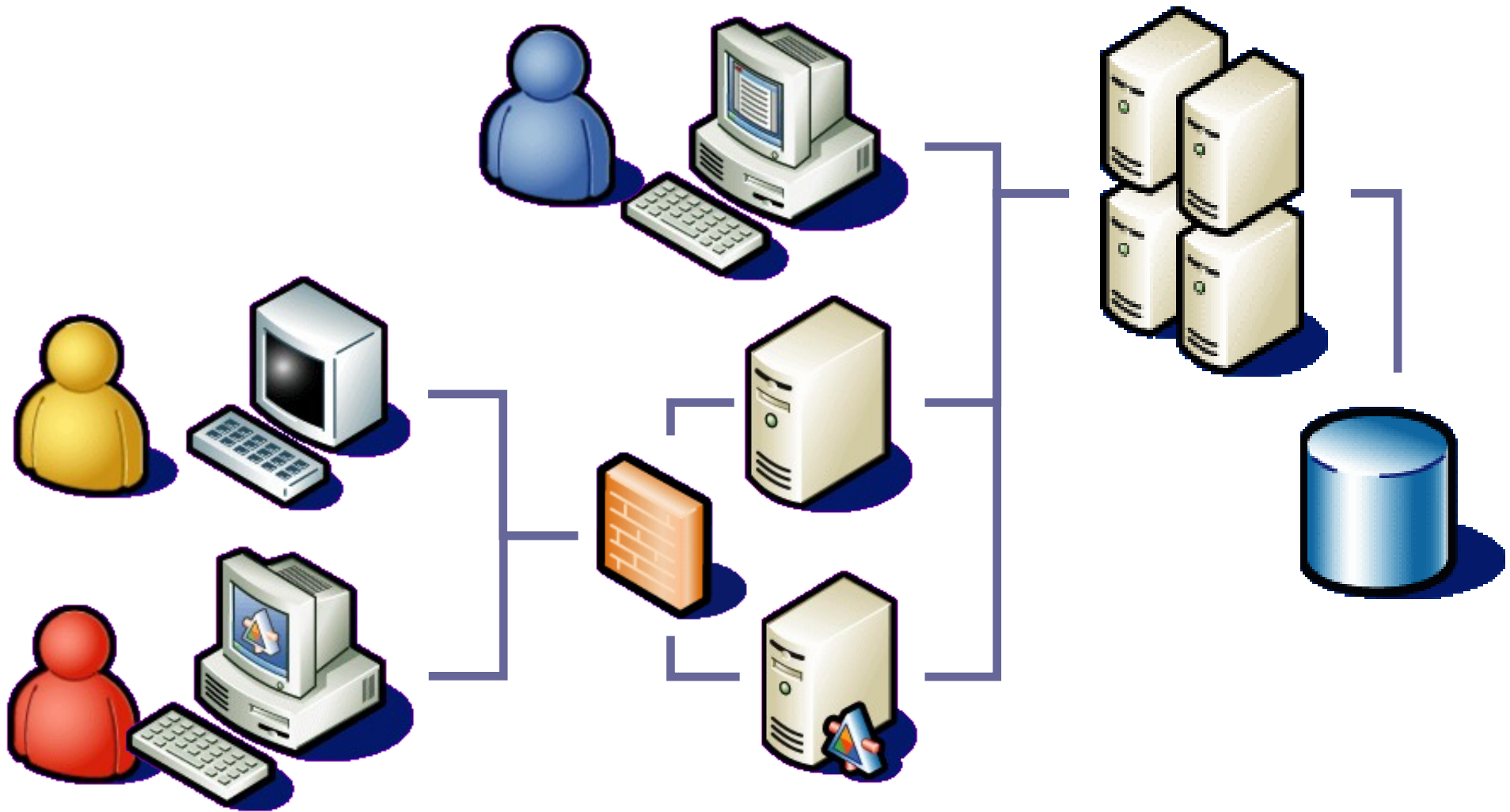
Centralized Architecture



Client-Server Architecture



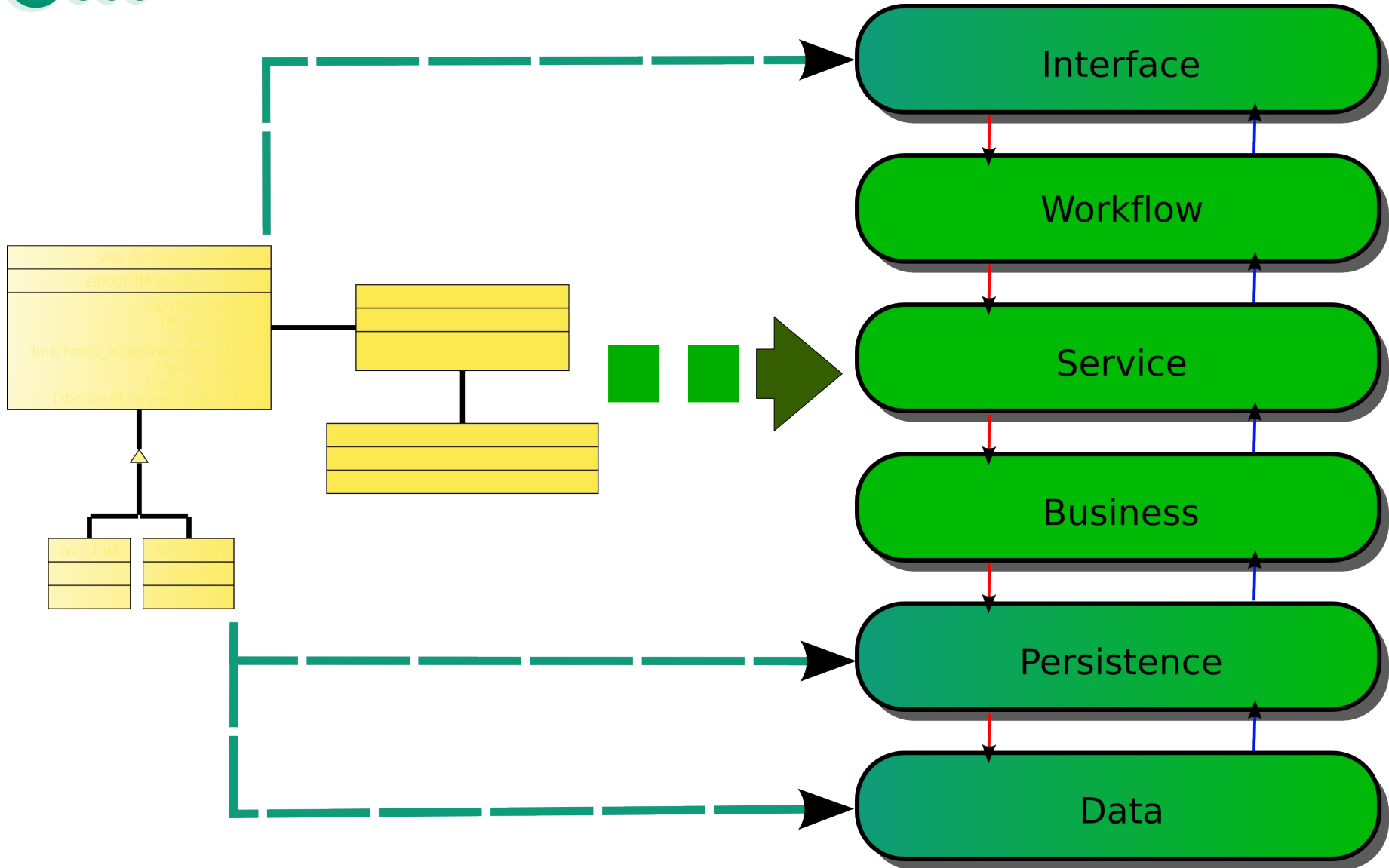




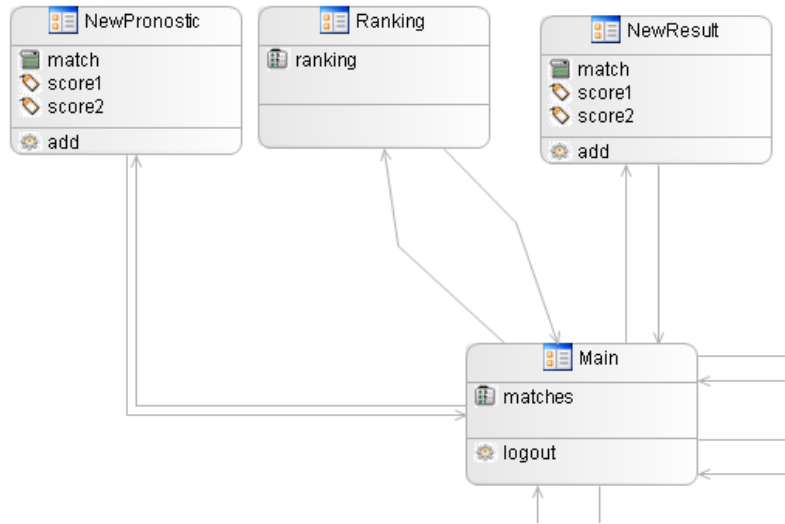
- But the reality is different :
many kind of architecture co-exist and must work together

- Why do we need SOA ?
 - IS aligned to company business
 - But what about functional / technical urbanisation ?
 - But what about tools and methodology ?
 - Be more evolutive
 - But what about IS global view ?
 - But where is my business knowledge ?
 - Be more reliable
 - But what about simplicity ?
 - But what about agility ?

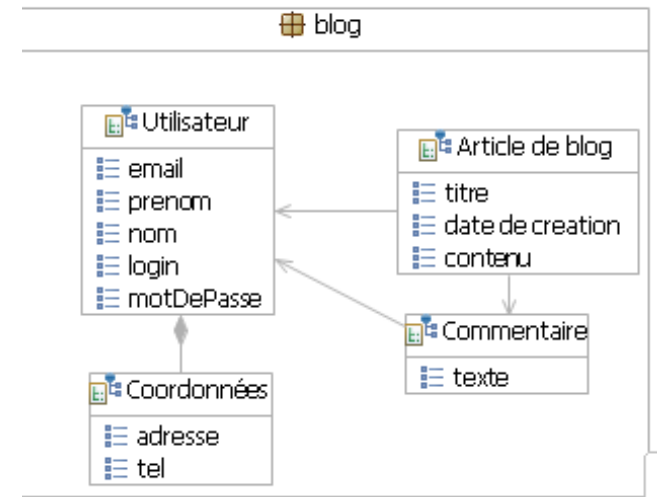
Manage several layers



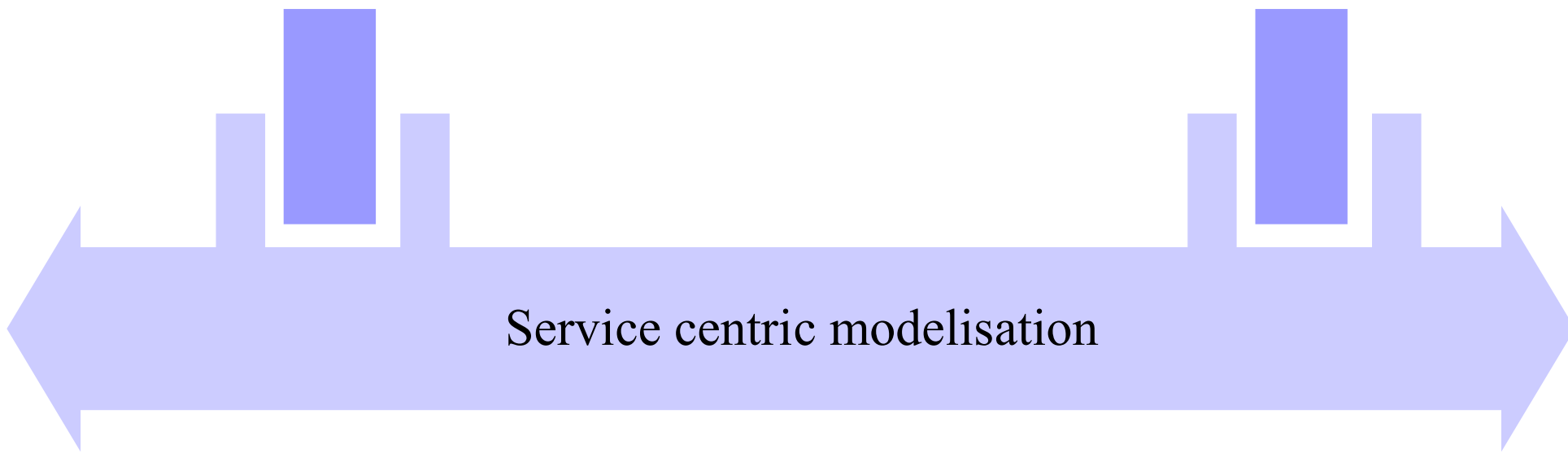
→ Acceleo



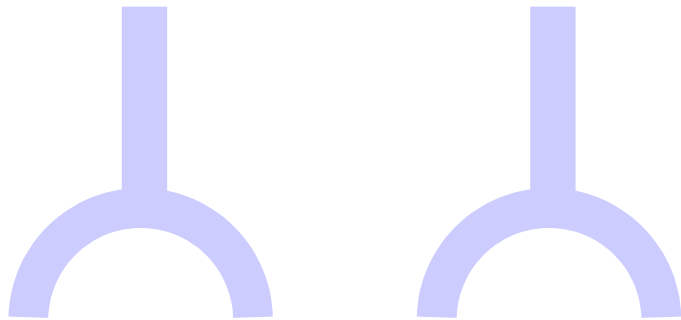
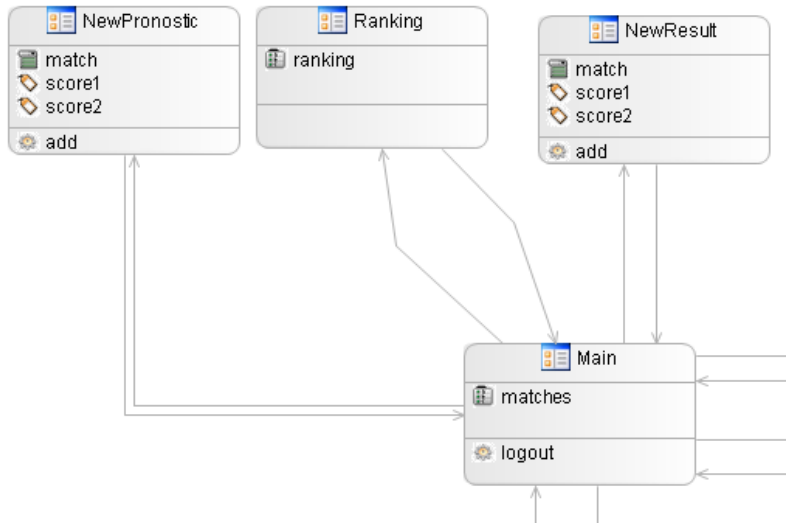
User centric modelisation



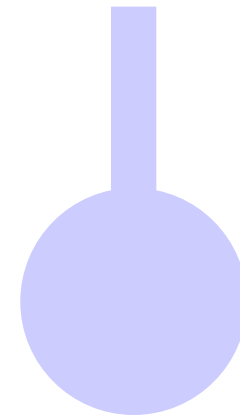
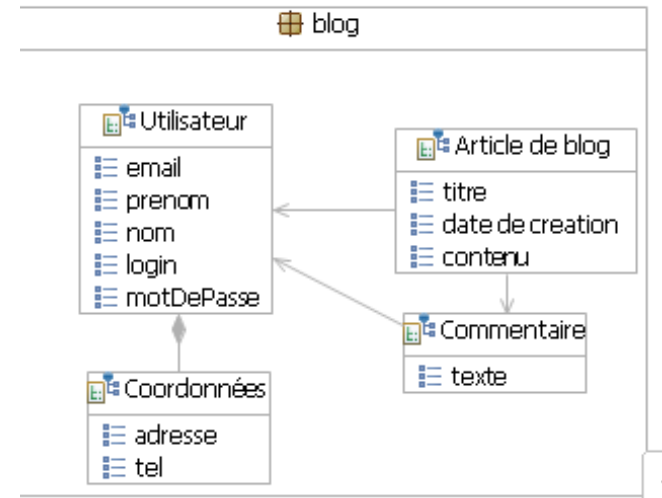
Business centric modelisation



Service centric modelisation

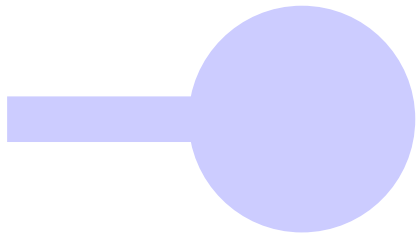


What is needed ?

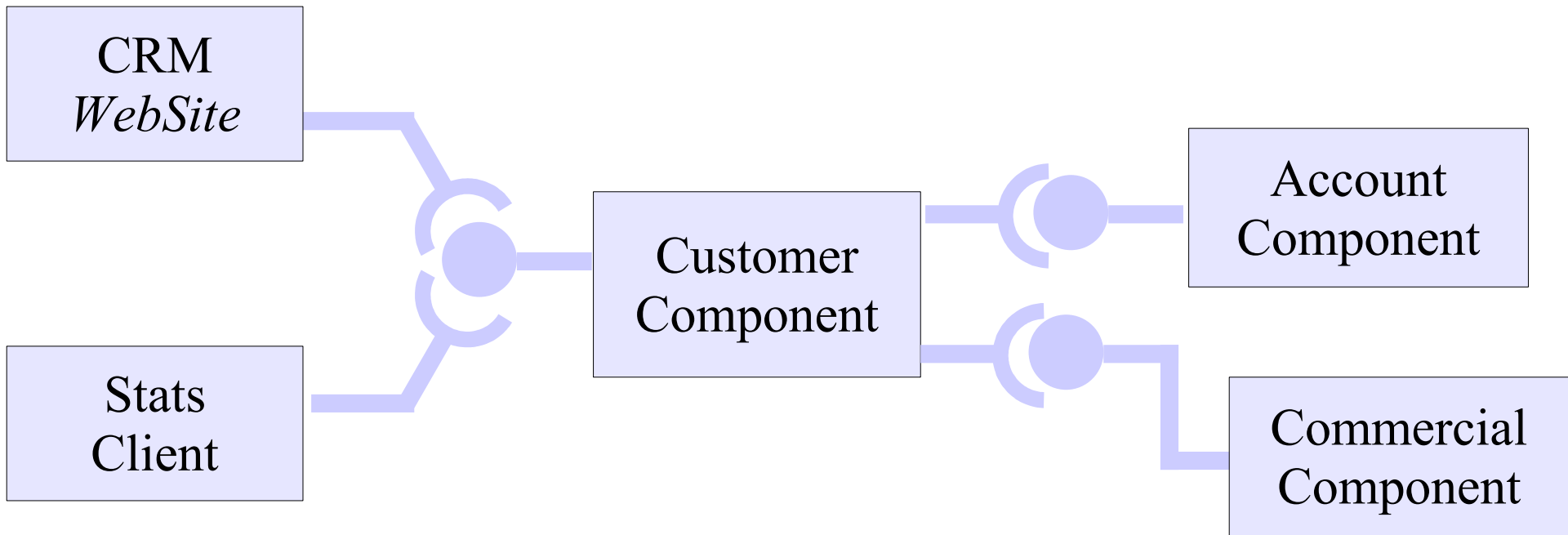


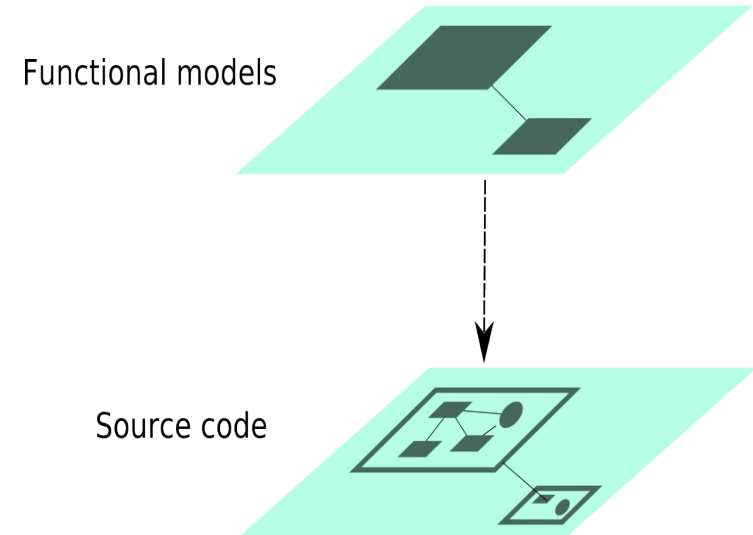
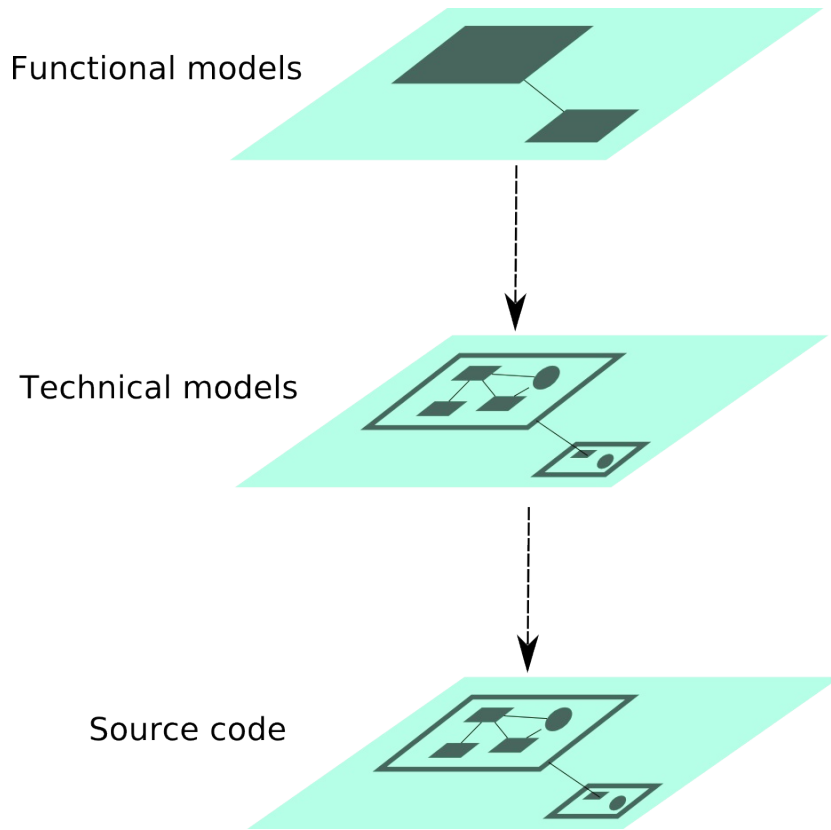
What is provided ?

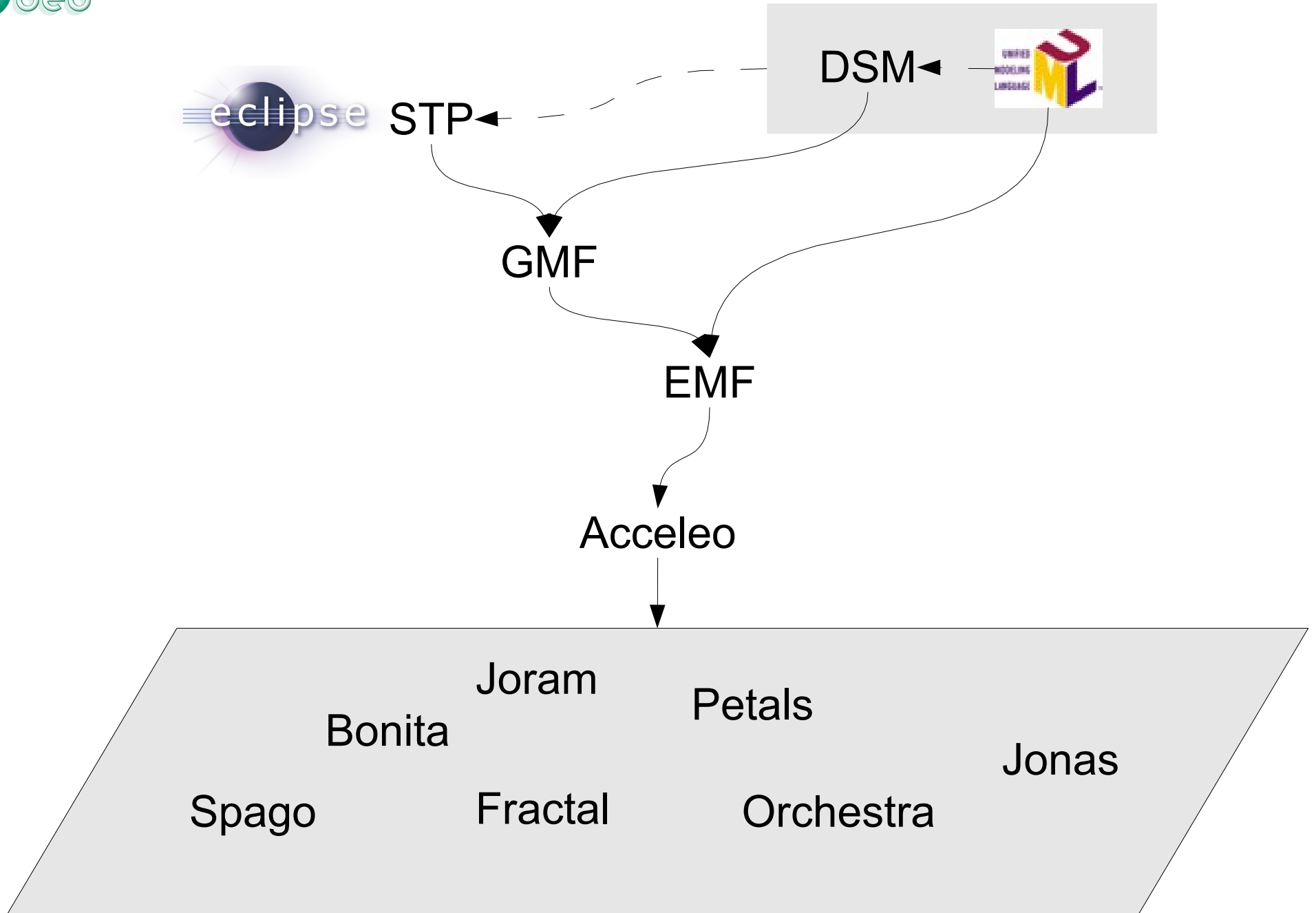




- A service is a contract
- It is described with :
 - A name
 - List of exchanged datas
- A Service can be :
 - Interface oriented
 - Message oriented
- But how describe technicals parameters ?







Architecture et
Méthodologie MDA

- patterns d'architecture
- prototypage
- méthodologie de conception
- définition des méta-modèles

Industrialisation

- réalisation sur mesure de
 - o modules de génération
 - o modeleurs
- adaptation aux contextes projets

Réalisation

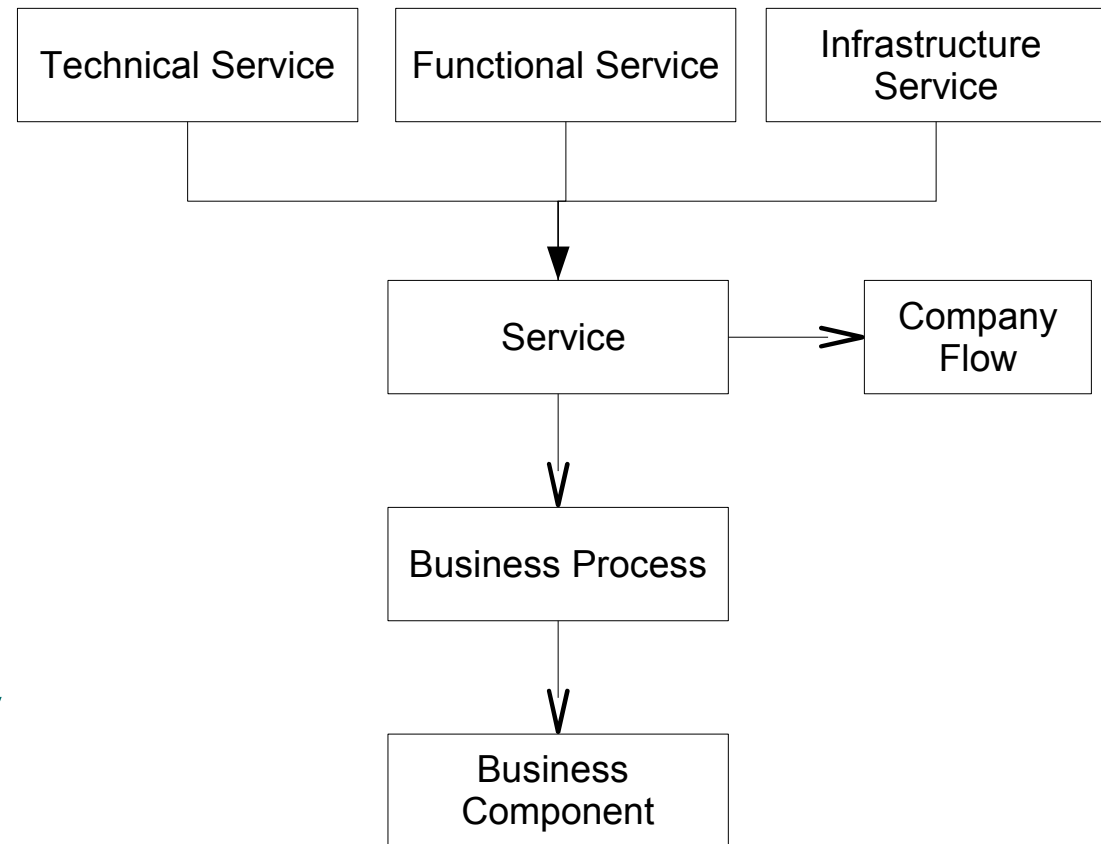
- formations
- support dédié
- traçabilité des évolutions
- synchronisation modèle/code

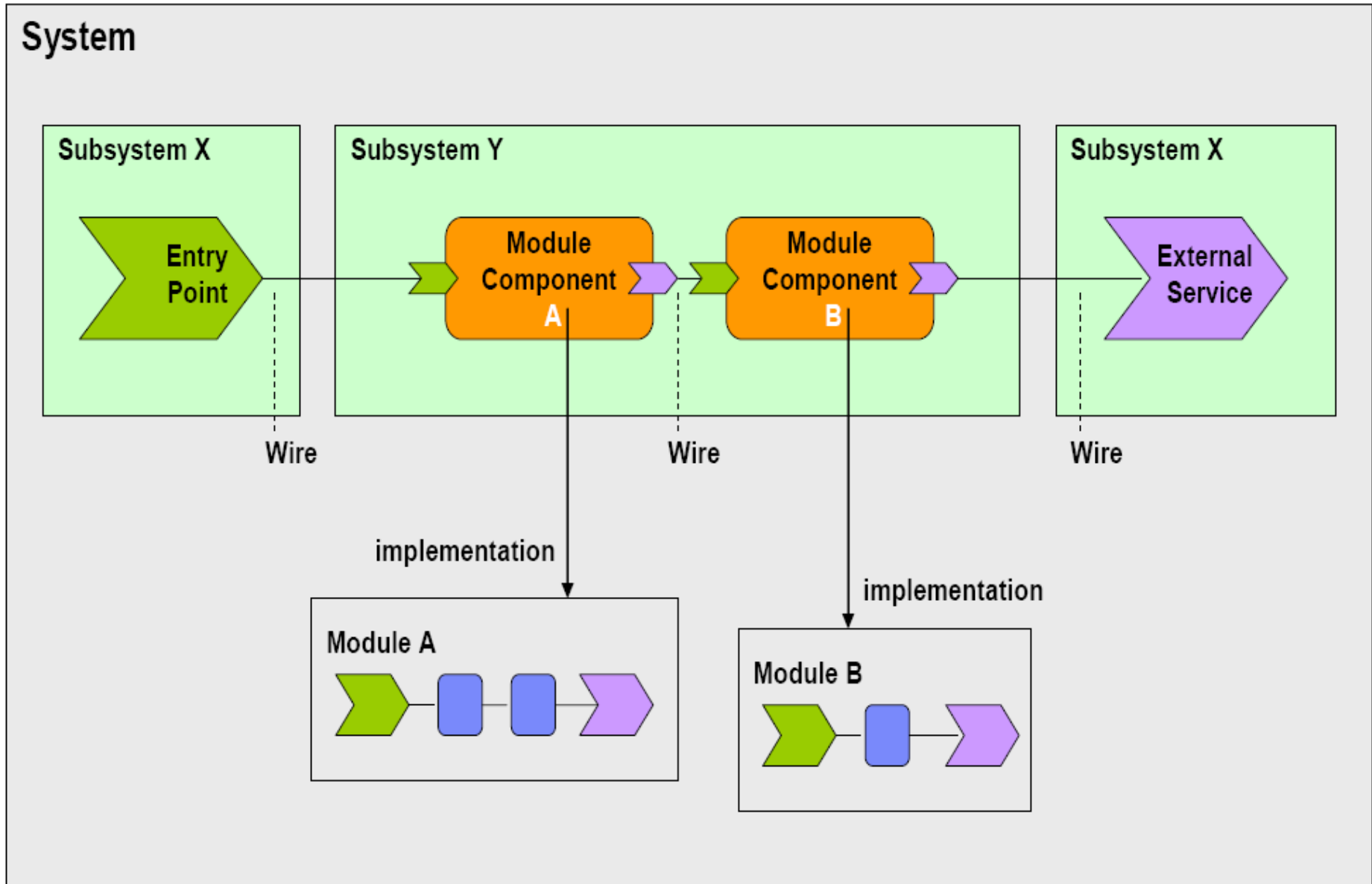
“How should I use models to be efficient ?”

STEP 1 : DESIGN

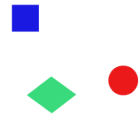
“How to design using MY semantics ?”

- Create **your** repository
- Be Unified : UML
 - Profil + Stereotypes
- Be Specific : DSM
 - Metamodel
- EMF bring interoperability
- Eclipse STP compatibility



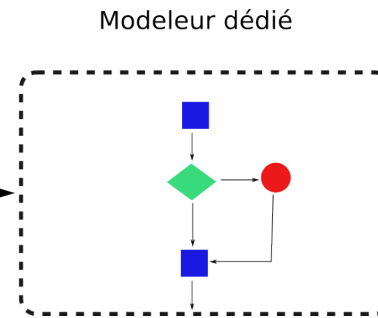


Définition des concepts



Mapping graphique

Règles de validation



Goals :

– Simplify

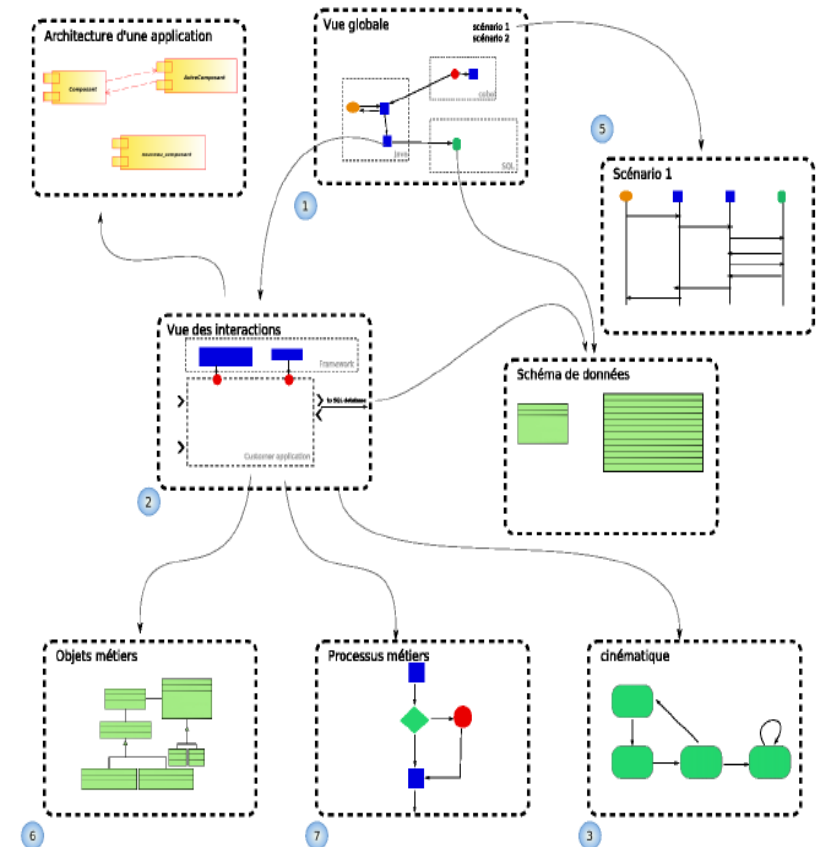
- design
- tools

– Improve

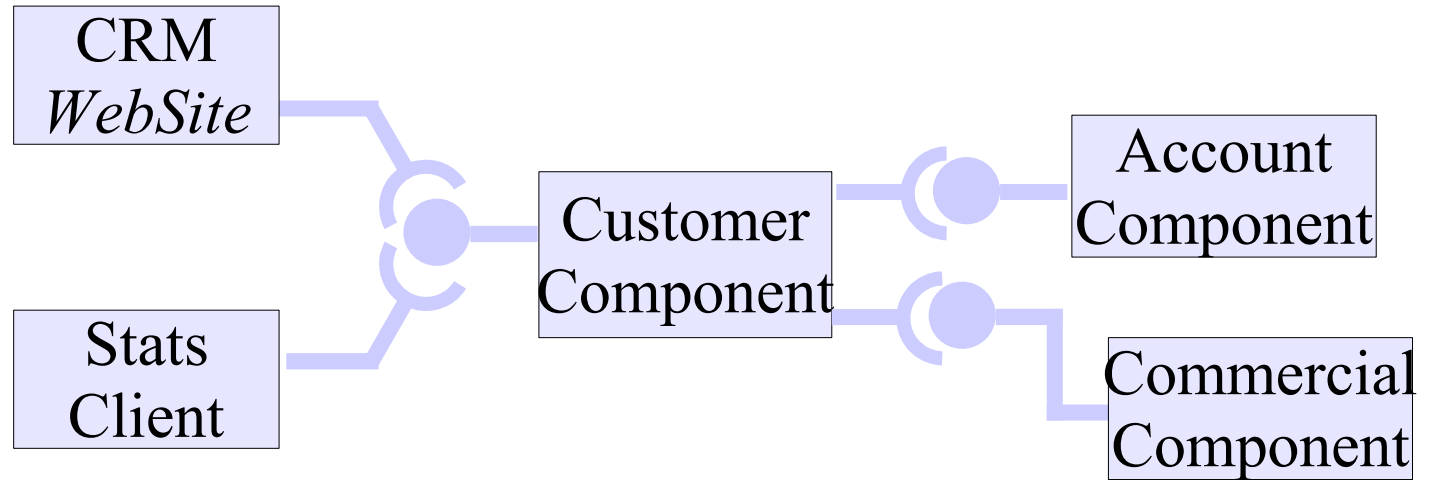
- not limited to **UML** concepts
- full **model based** (EMF)

– Accelerate

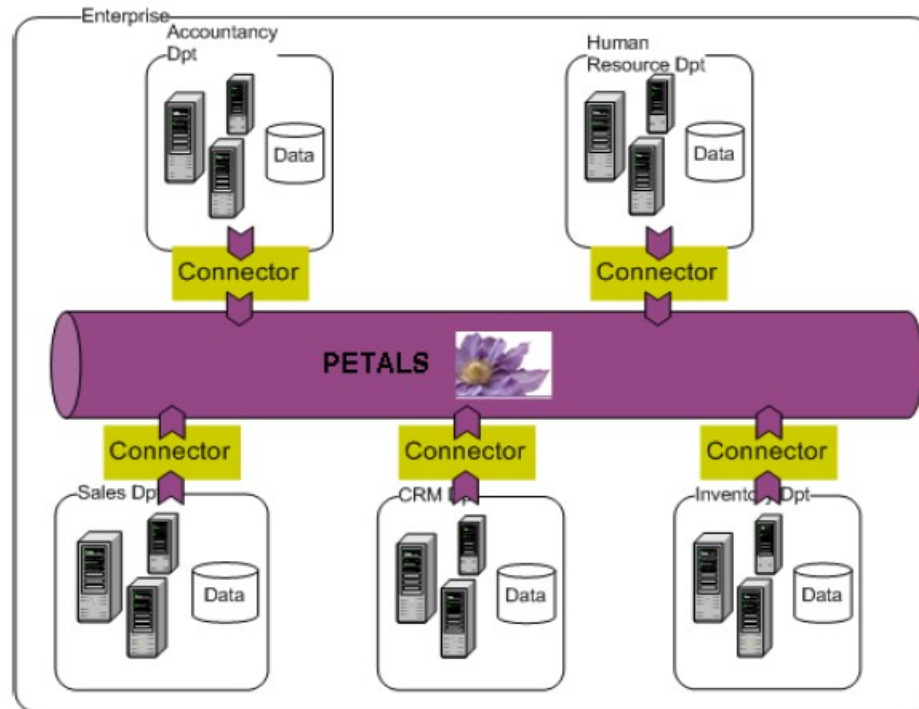
- integration with Eclipse and Acceleo
- productivity



Logical View



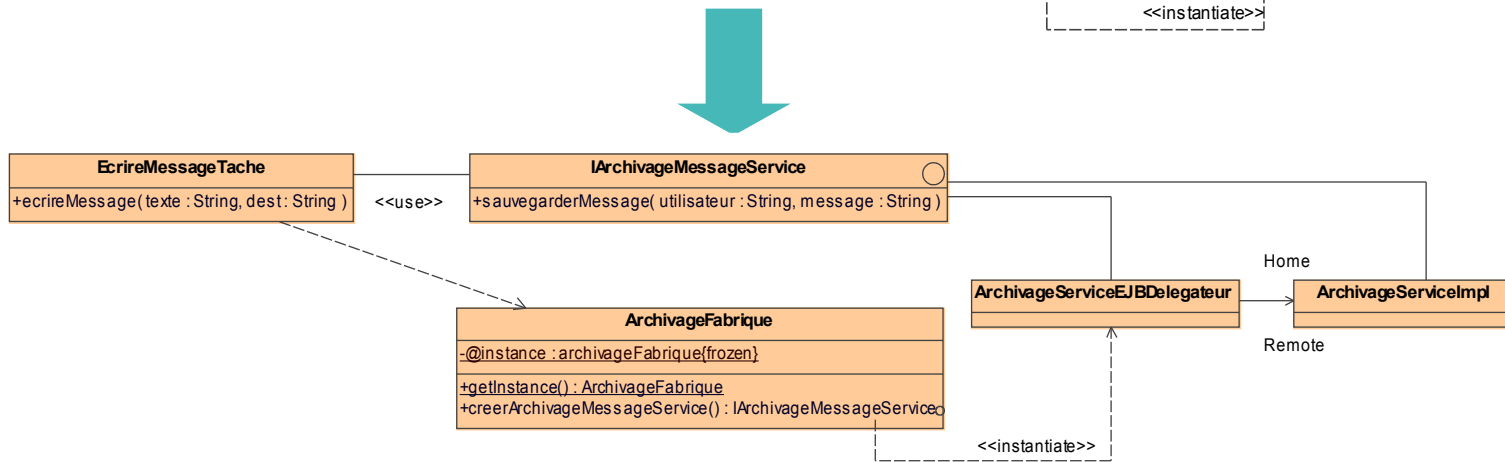
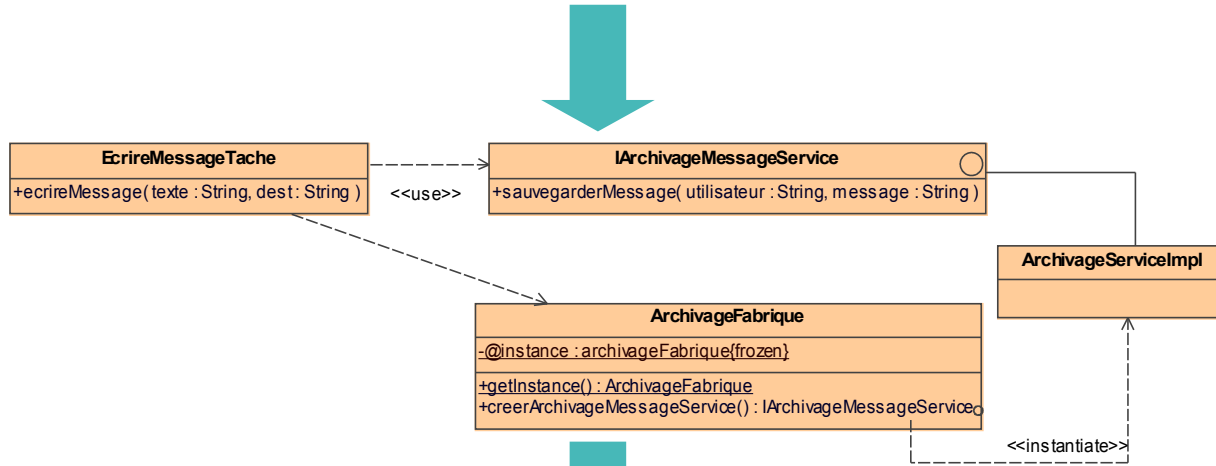
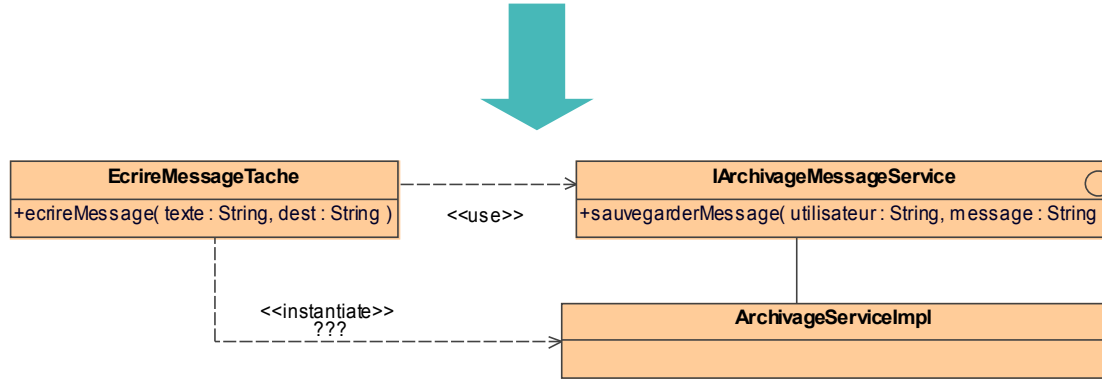
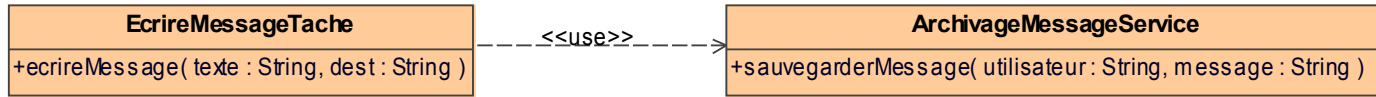
Technical View

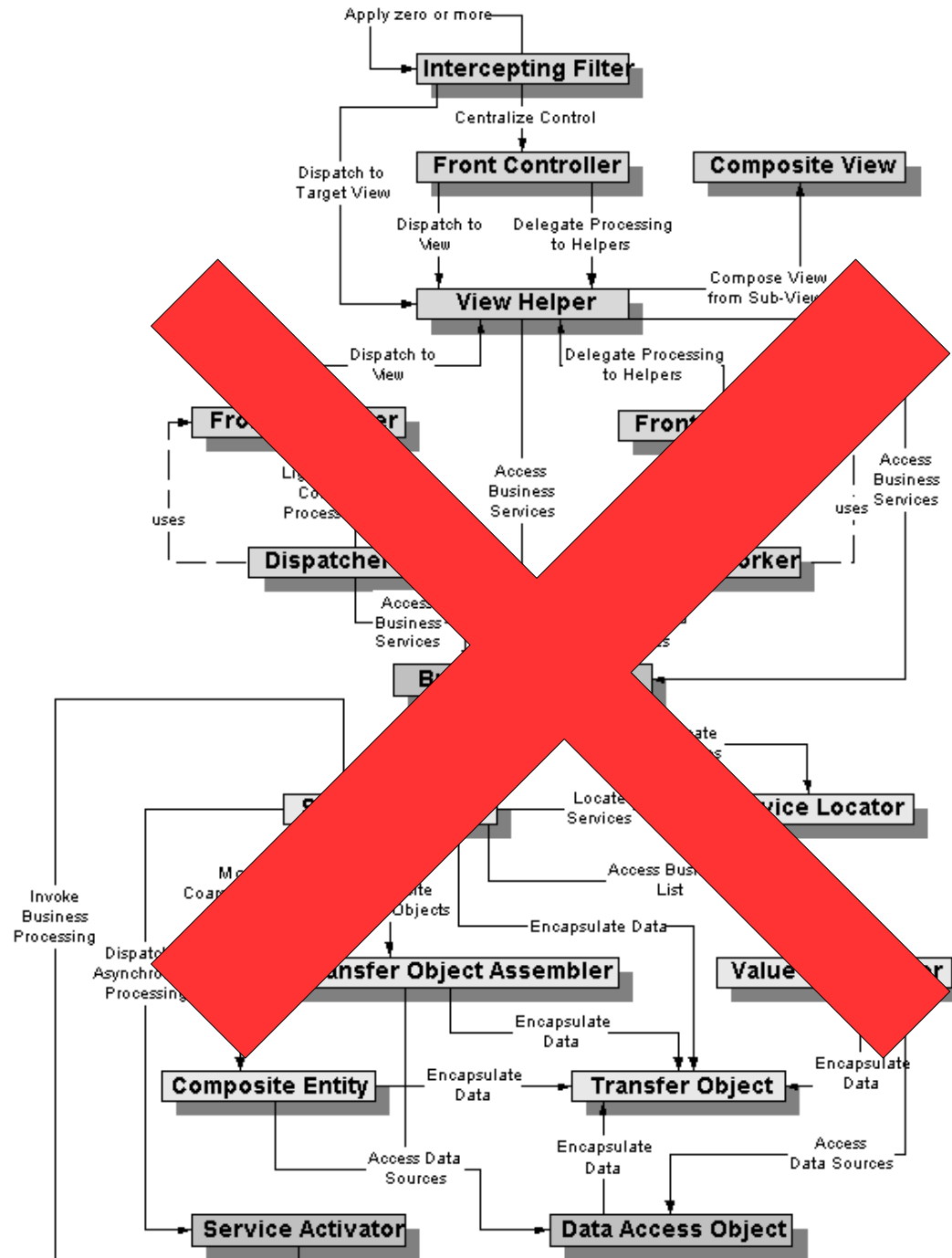


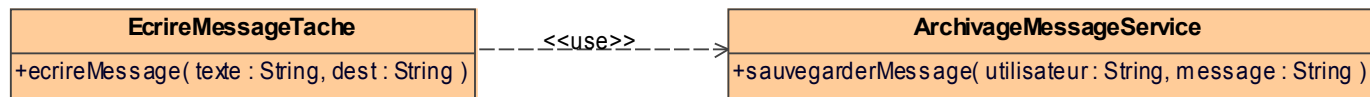
“How keep it simple ?”

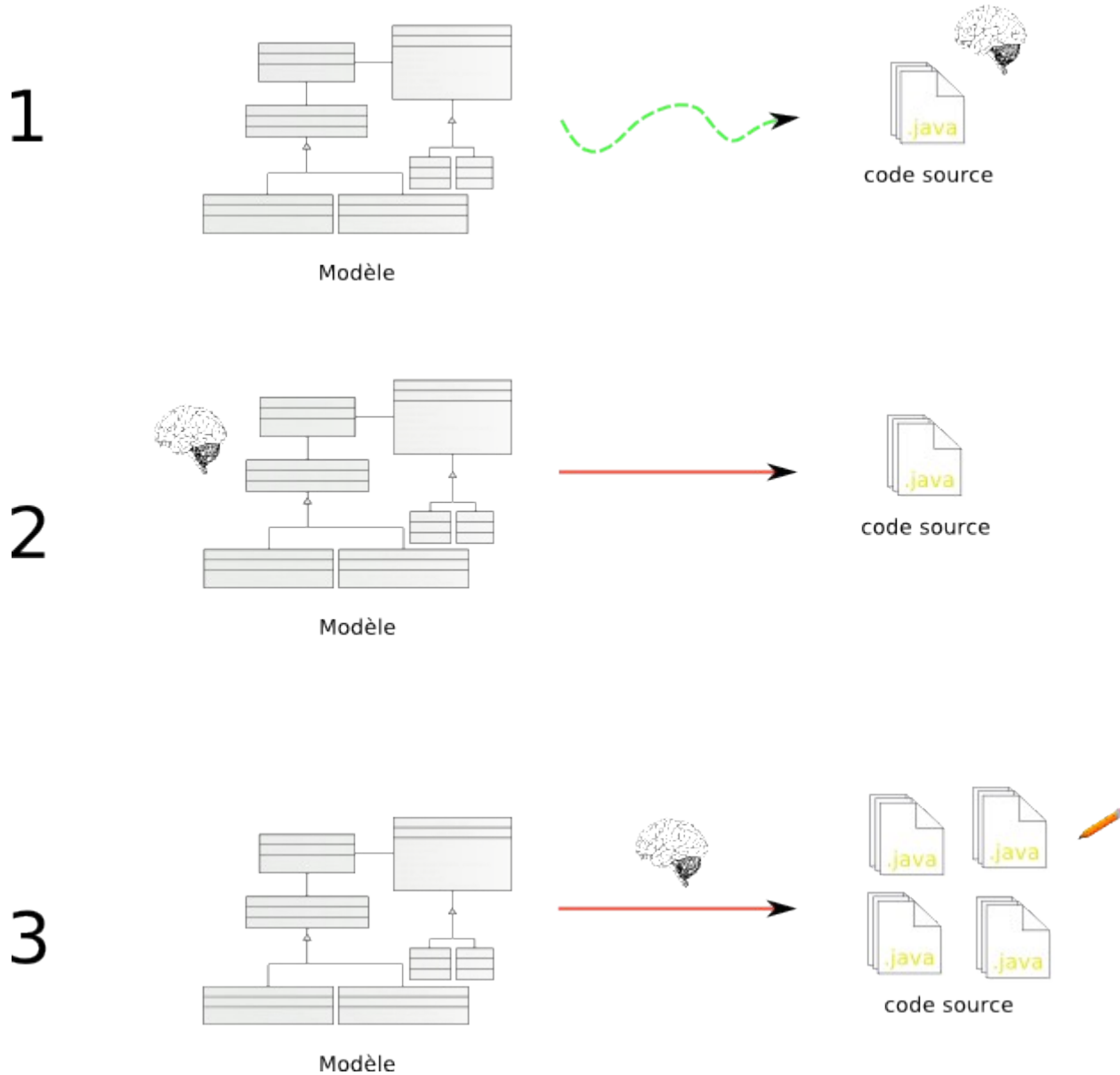
STEP 2 : INDUSTRIALISATION

“Where is my technical knowledge ?”





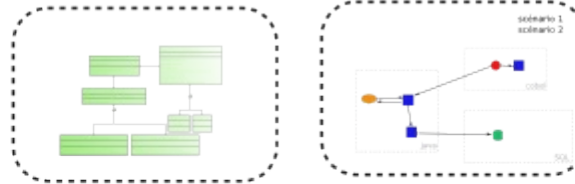




STEP 3 : GENERATION

“Is MDA miraculous ?”

Modèles



Architecture SOA
Composants distribués
Objets métiers



**Génération
Acceleo Pro**

WebServices

EJB

Couches services
Java
WSDL

DAO
Objets métiers
EJB Entity
Services distribués
EJB Stateless
Bus EAI
EJB MDB

- MDA can be simple
- Functional <-> Acceleo <-> Technical
- Acceleo can be used to target Petals, Jonas, Telosys, Spago, ... : *Acceleo Community Modules*
- Acceleo help urbanisation and governance
- ScorWare / Eclipse STP / Acceleo

Acceleo : MDA generator - Home - Mozilla Firefox

Fichier Edition Affichage Aller à Marque-pages Outils ?

http://localhost/acceleo.org/pages/home/en

>> Français



Accelerate your developments
Effective MDA !

Menu

- Home
- Presentation
- Features
- Documentation
- Demonstration
- Download
- Support



Acceleo
OpenSource
GPL



Download Acceleo

Welcome to Acceleo's website

Acceleo is the most powerful code generator tool of its generation. It has been designed to improve software development productivity.

Acceleo brings to everyone the MDA approach applied to development industrialization.



```

    graph TD
      A[Step 1 : Modeling] --> B[Acceleo]
      C[Step 2 : Industrialization] --> B
      B --> D[Step 3 : Generation]
      E[Modules] --- C
    
```

Acceleo helps you to :

- use your models in a productive way
- apply the best architecture patterns
- shorten project deadlines
- improve reliability

Acceleo, a simple and efficient solution :

Acceleo + MDA + eclipse
= **Productivity + Durability + Agility**

Acceleo assets :

Terminé

Adblock