

Reverse and Forward Transformation Chain



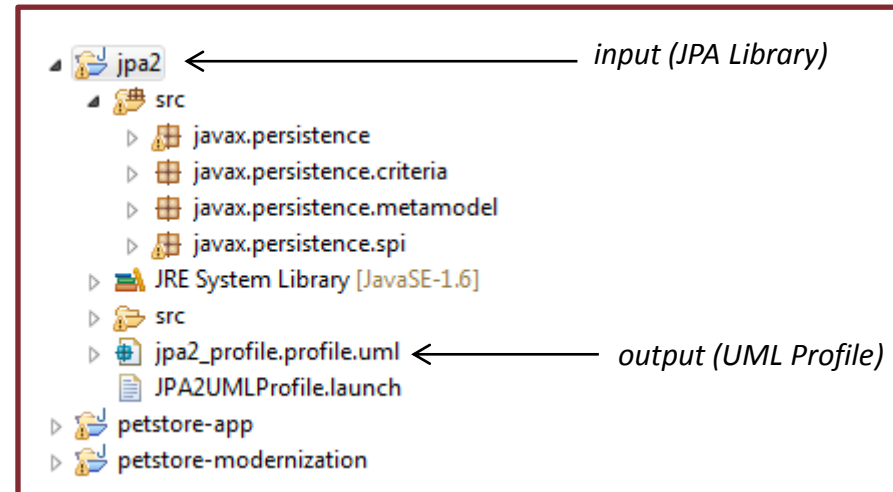
9th Symposium and Summer School On Service-
Oriented Computing
Heraklion, Crete, Greece
01-07-15

JUMP: Discover Libraries



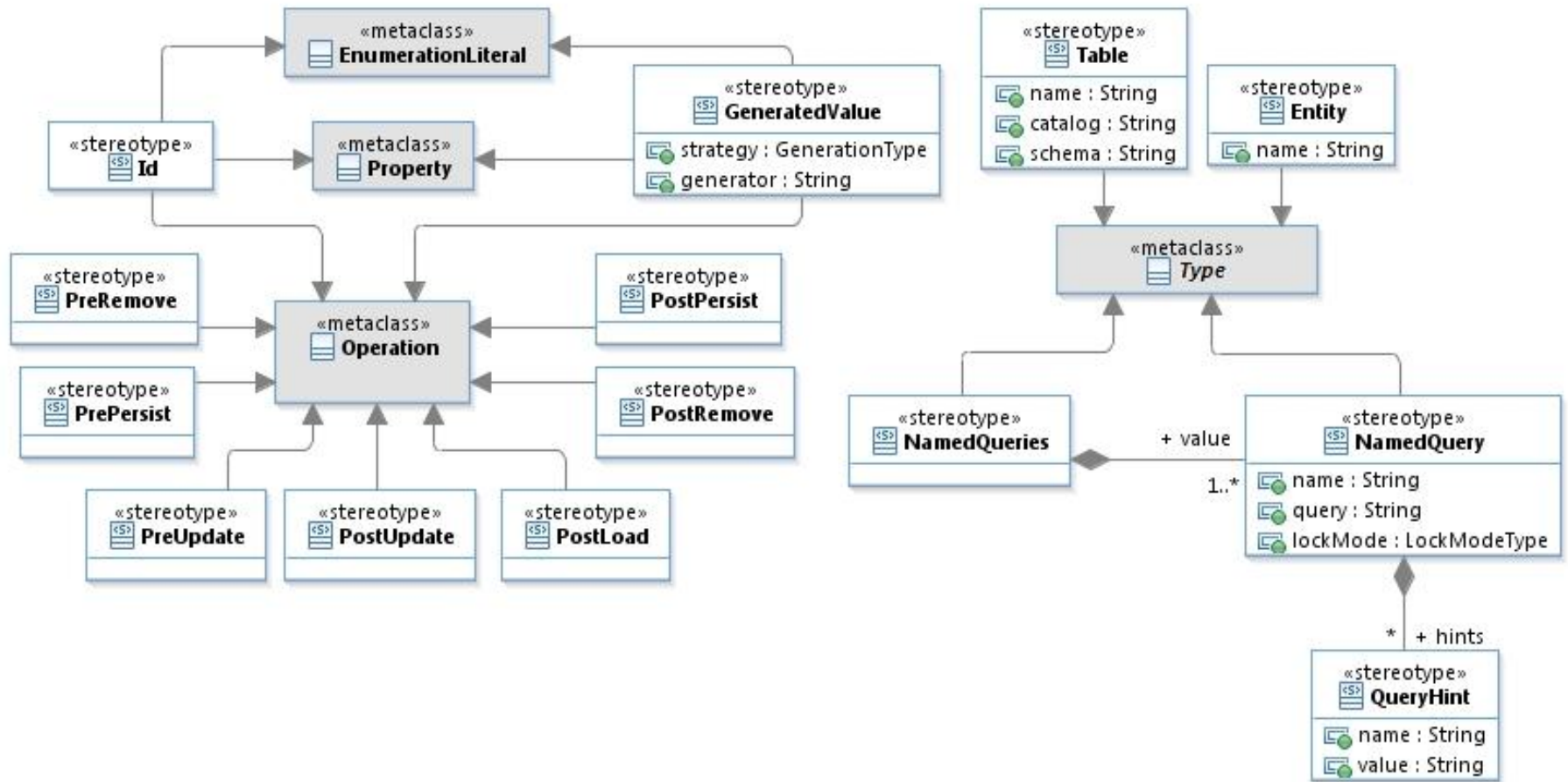
- It allows to extract UML Profiles from Java annotations

- Example shown: JPA Library
- To execute it: right click on the jpa2 project and select *Model Discovery Toolbox -> UML Profile Discovery*



- More UML Profiles have been discovered by JUMP
 - They can be installed as Eclipse plugins
 - <https://code.google.com/a/eclipselabs.org/p/uml-profile-store/source/browse/#git%2FUMLProfiles>

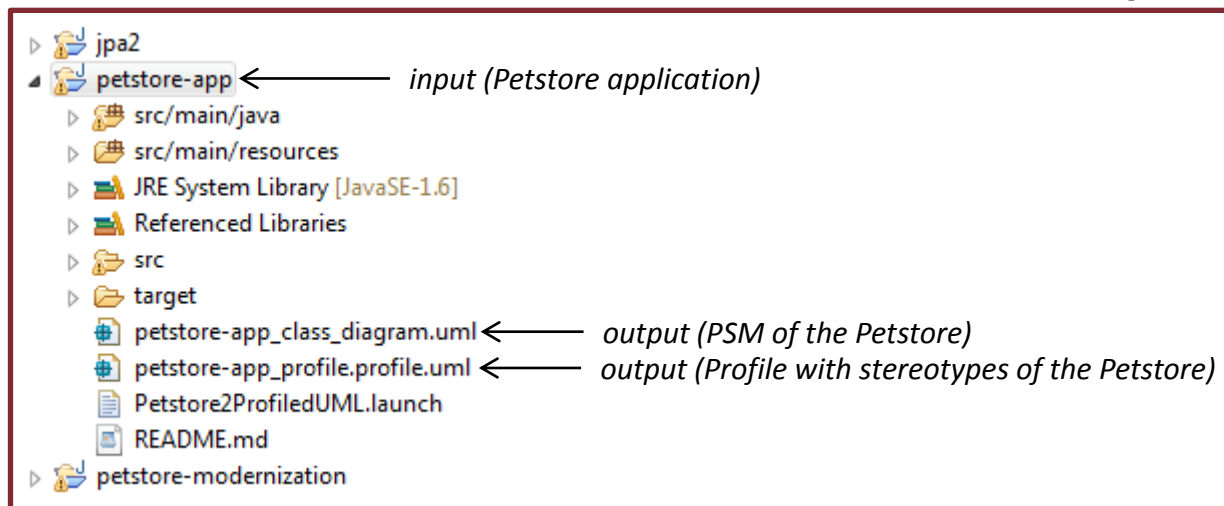
Snapshot of Collected UML Stereotypes - JPA



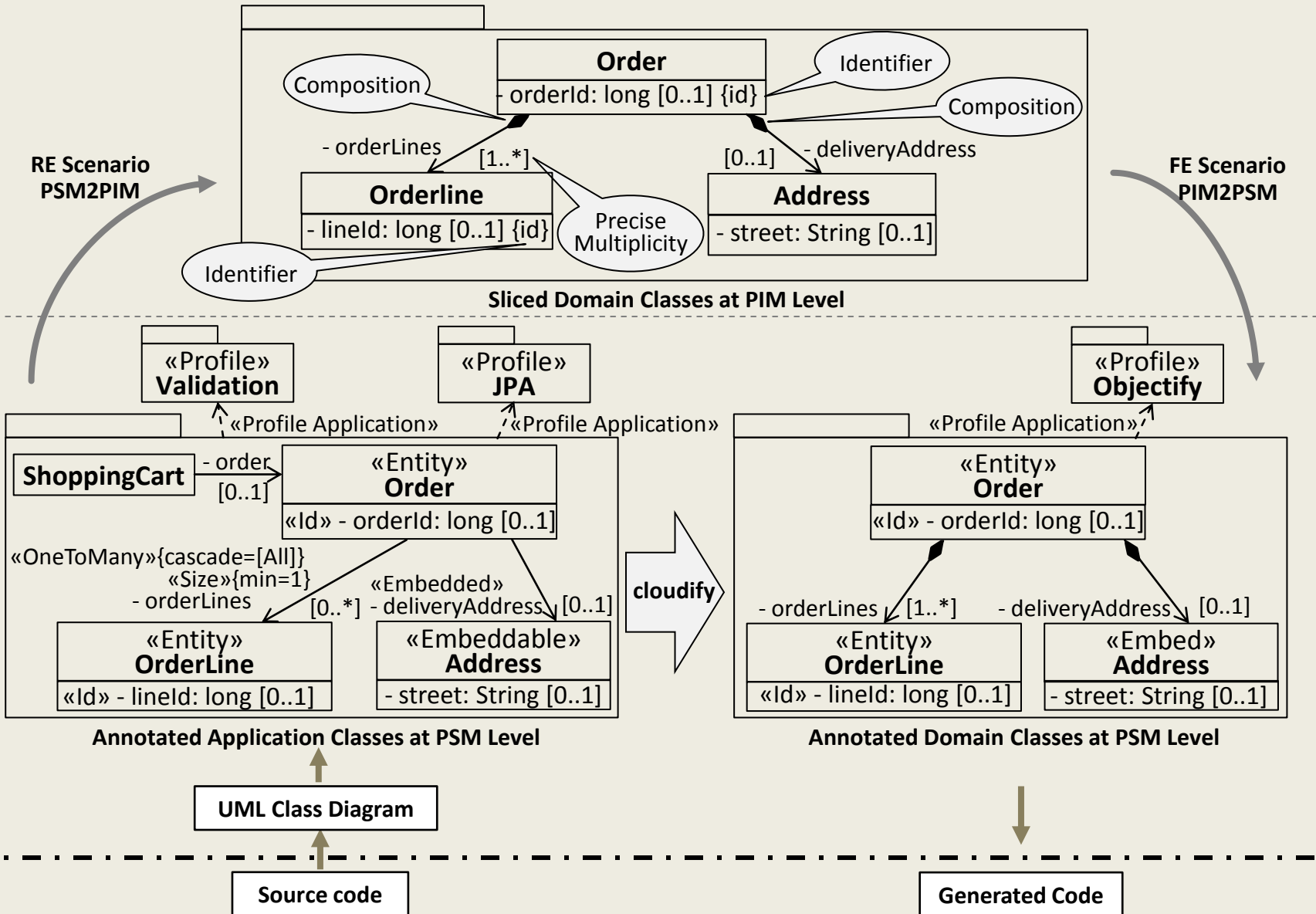
JUMP: Discover PSM + Libraries (RE)



- Execute JUMP on the *petstore-app*
 - In order to execute it, right click on the *petstore-app* and select *Model Discovery Toolbox -> Profiled UML Model Discovery*



Cloudifying through JUMP



Cloudifying through JUMP

UML Class Diagram



Source code

1. Understanding (I) (RE)



- Based on the JPA annotations, we are interested only in the domain model

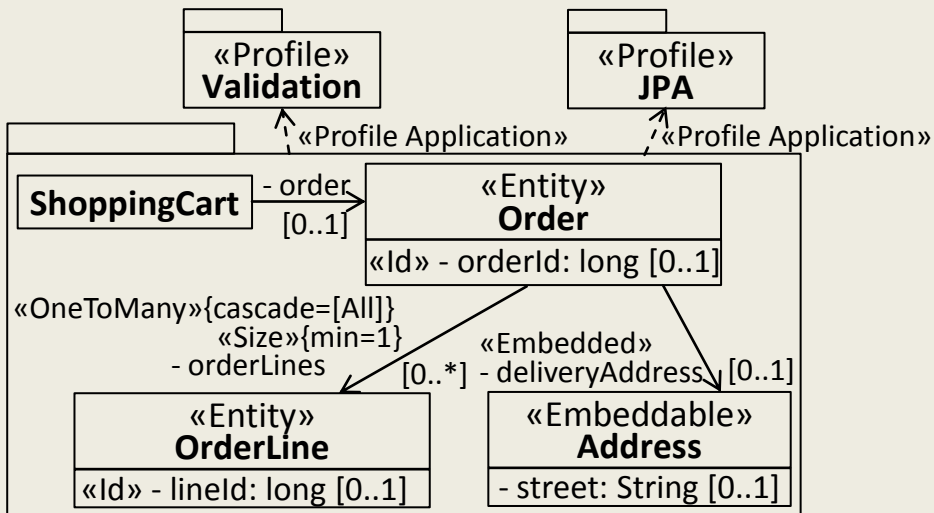
The screenshot shows a project explorer with the following structure:

- ▶ jpa2
 - ▶ petstore-app
 - ▶ src/main/java
 - ▶ src/main/resources
 - ▶ JRE System Library [JavaSE-1.6]
 - ▶ Referenced Libraries
 - ▶ src
 - ▶ target
 - ▶ petstore-app_class_diagram.uml
 - ▶ petstore-app_profile.profile.uml
 - ▶ Petstore2ProfiledUML.launch
 - ▶ README.md
 - ▶ petstore-modernization
 - ▶ src
 - ▶ src-gen
 - ▶ JRE System Library [JavaSE-1.7]
 - ▶ Referenced Libraries
 - ▶ deployment
 - ▶ feasibility_reports
 - ▶ lib
 - ▶ models
 - ▶ deployment
 - ▶ discovery
 - ▶ goals
 - ▶ modernisation
 - ▶ technical_feasibility
 - ▶ understanding
 - ▶ petstore-domain_class_diagram_PSM.uml
 - ▶ profiles
 - ▶ trafo
 - ▶ trafo-launch
 - ▶ 01_Annotation-based-Slicer.launch
 - ▶ 02_Annotation-based-Analysis.launch
 - ▶ 03_PIM2ObjectifiedPSM.launch
 - ▶ 04_ObjectifiedPSMWithServiceClasses.launch
 - ▶ 05_GenerateCode-Petstore.launch

Annotations on the right side of the image:

- input (PSM of the Petstore)
Some profiles of the UML Profile Store are also taken as input
- output (Sliced PSM of the Petstore containing the domain model)
- launch configuration

Cloudifying through JUMP (1)



Annotated Application Classes at PSM Level

UML Class Diagram

Source code

2. Understanding (II) (RE)



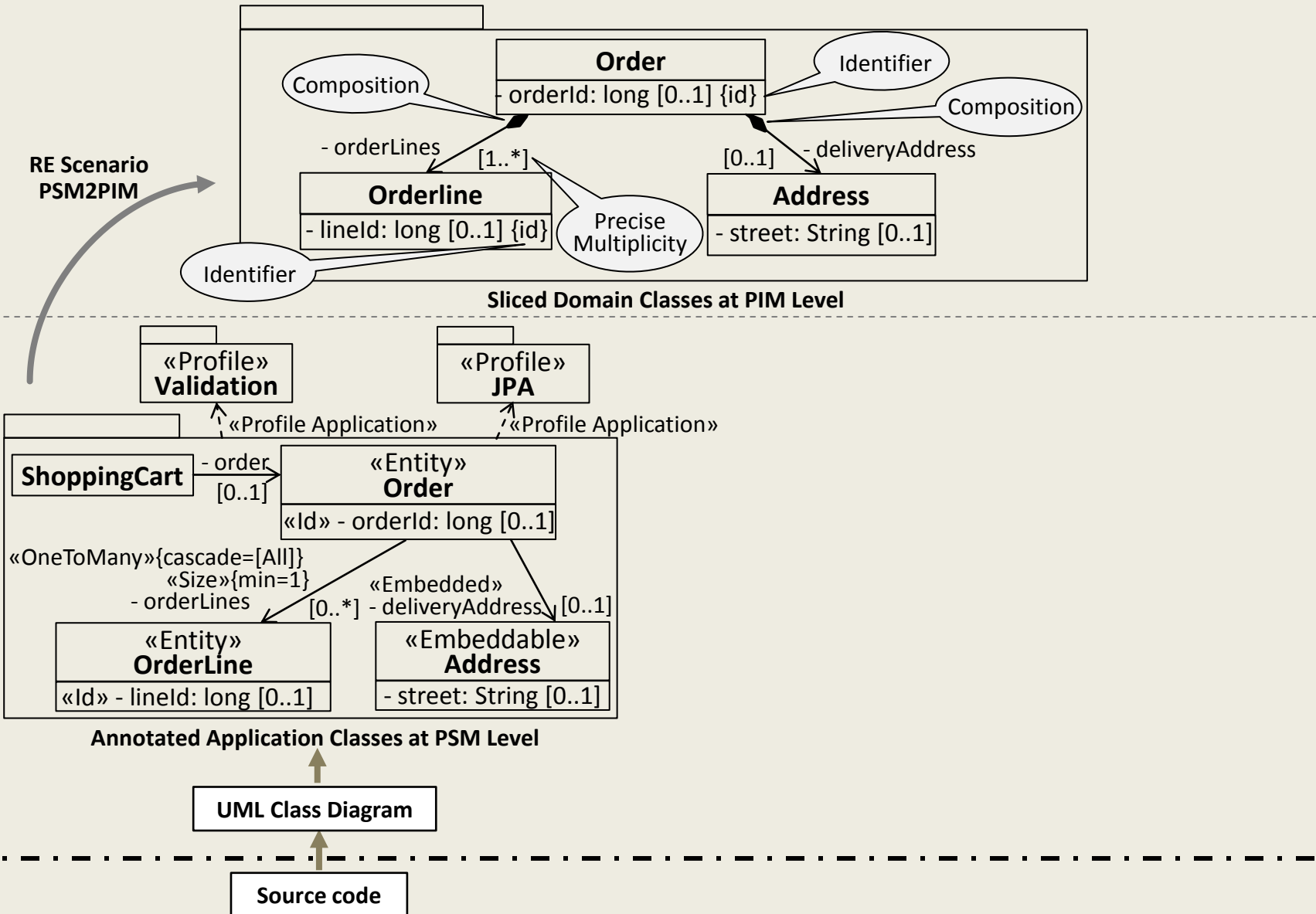
- We want to obtain a PIM from the PSM
 - Remove all stereotypes
 - Turn some plain associations into composition associations

Output (PIM of the Petstore containing the domain model)

Input (Sliced PSM of the Petstore containing the domain model)

launch configuration

Cloudifying through JUMP (2)



3. Modernisation (I) (FE)



- Refine the PIM into a PSM cloud-compatible
 - Addition of the Objectify Profile

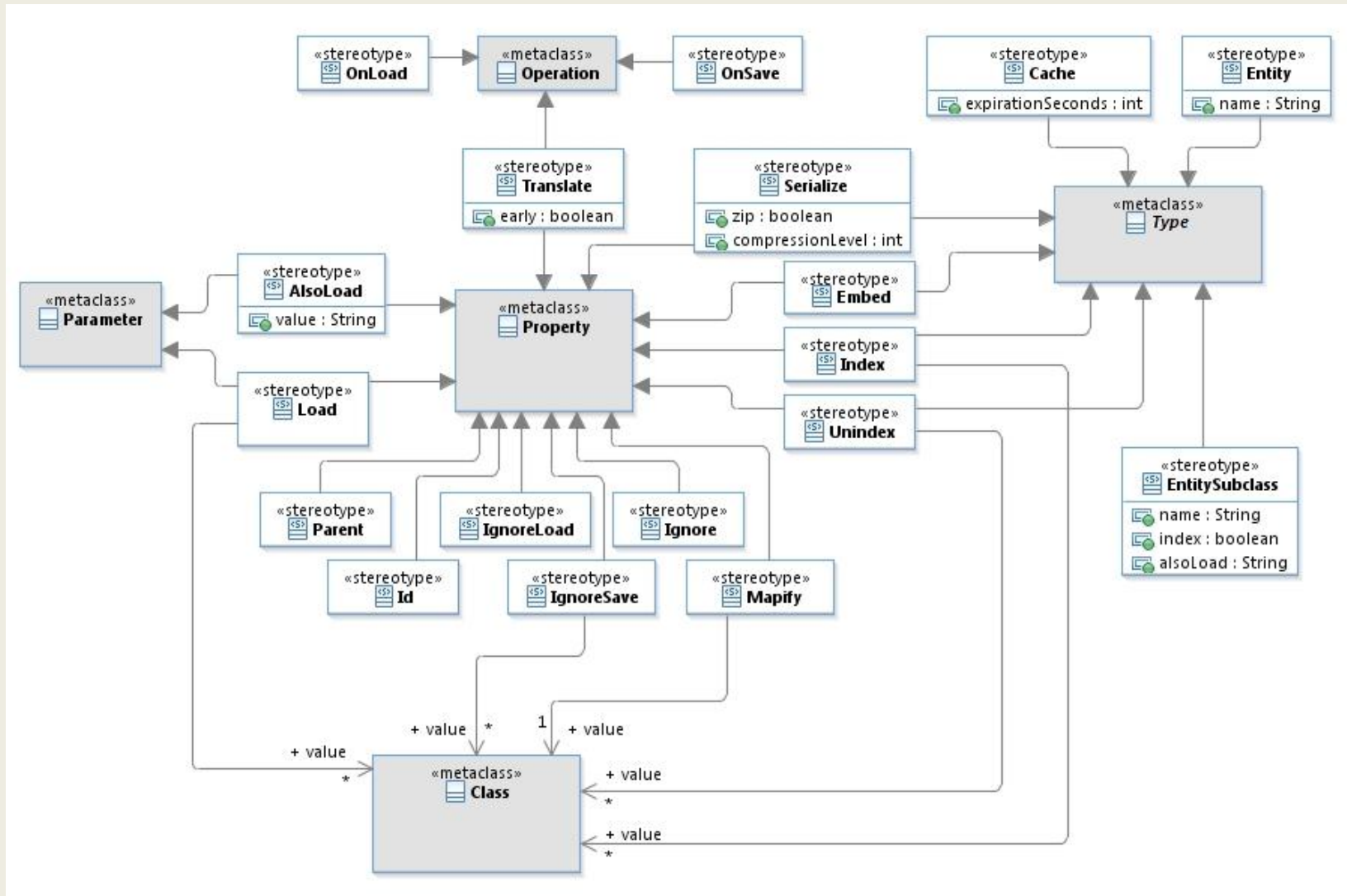
The screenshot shows a project tree for 'petstore-modernization'. The tree structure is as follows:

- petstore-modernization
 - src
 - src-gen
 - JRE System Library [JavaSE-1.7]
 - Referenced Libraries
 - deployment
 - feasibility_reports
 - lib
 - models
 - deployment
 - discovery
 - goals
 - modernisation
 - petstore-domain_class_diagram_PSM.uml
 - technical_feasibility
 - understanding
 - petstore-domain_class_diagram_PIM.uml
 - petstore-domain_class_diagram_PSM.uml
 - profiles
 - trafo
 - trafo-launch
 - 01_Annotation-based-Slicer.launch
 - 02_Annotation-based-Analysis.launch
 - 03_PIM2ObjectifiedPSM.launch
 - 04_ObjectifiedPSMWithServiceClasses.launch
 - 05_GenerateCode-Petstore.launch

Annotations on the right side of the image:

- An arrow points from the text "Output (PSM of the Petstore domain model with stereotypes from Objectify)" to the file 'petstore-domain_class_diagram_PSM.uml' in the 'modernisation' folder.
- An arrow points from the text "Input (PIM of the Petstore containing the domain model)" to the file 'petstore-domain_class_diagram_PIM.uml' in the 'understanding' folder.
- An arrow points from the text "launch configuration" to the file '03_PIM2ObjectifiedPSM.launch' in the 'trafo-launch' folder.

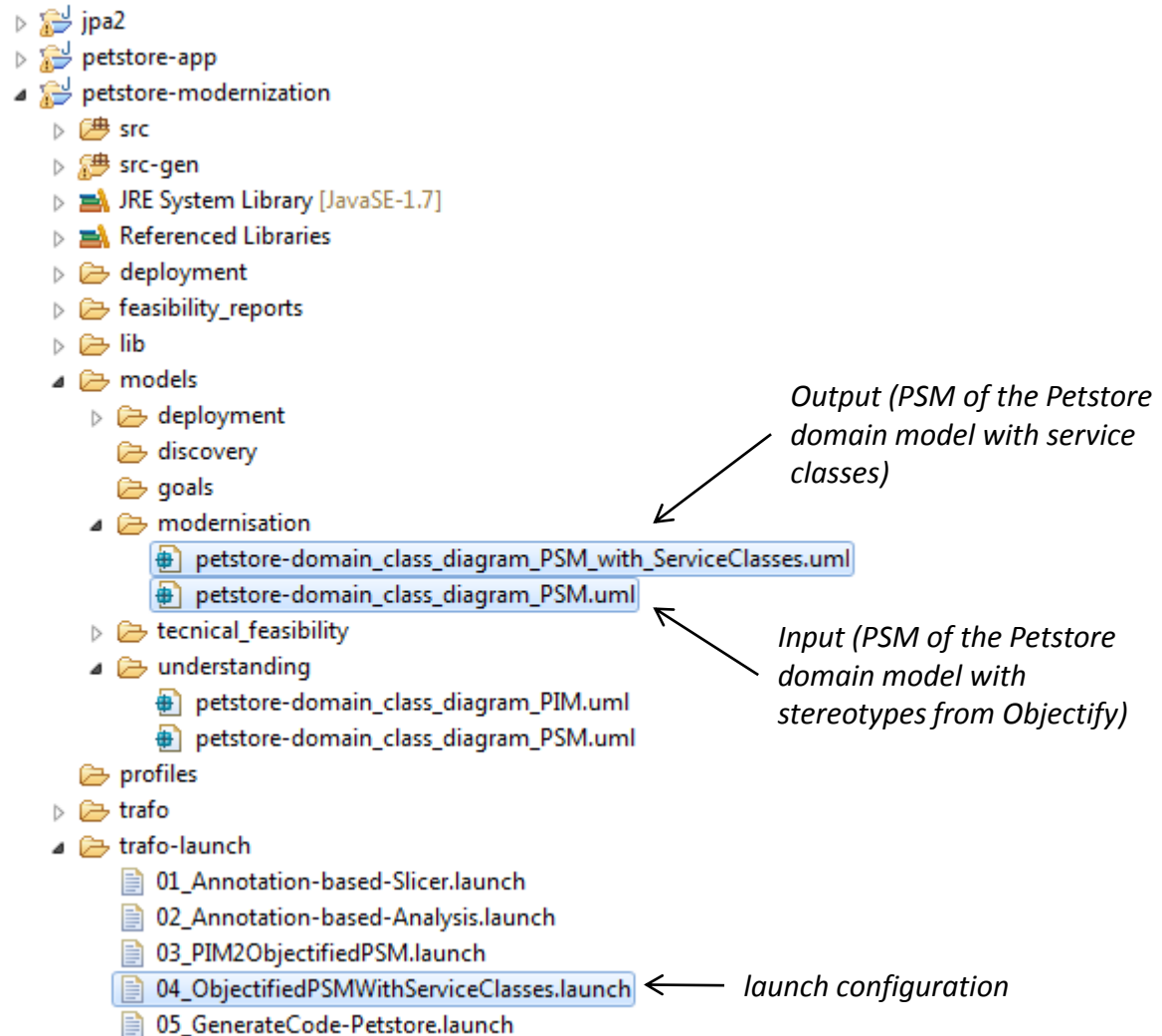
Snapshot of Collected UML Stereotypes - Objectify



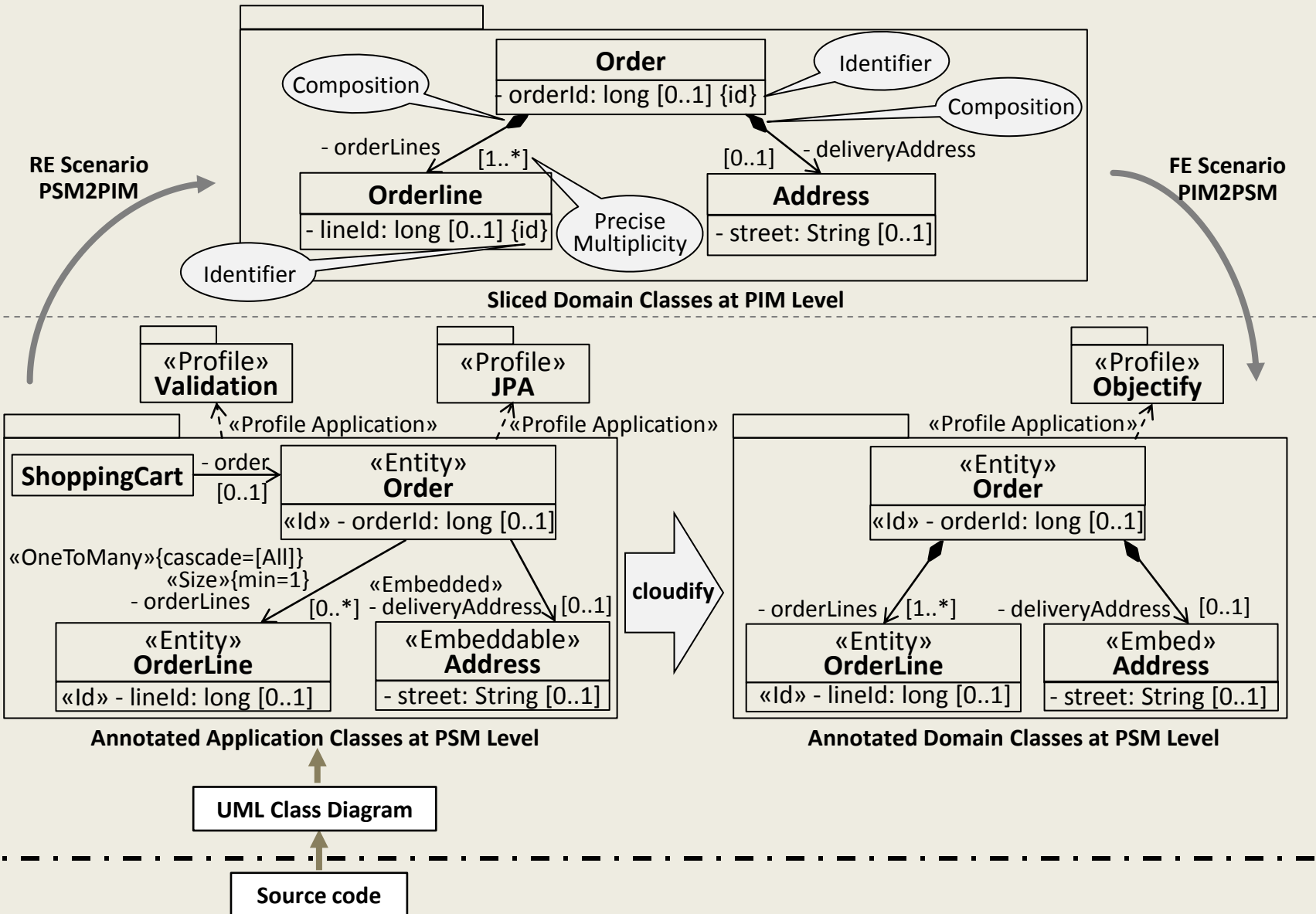
4. Modernisation (II)(FE)



- The PSM is further refined by adding service classes
 - They contain Objectify instructions



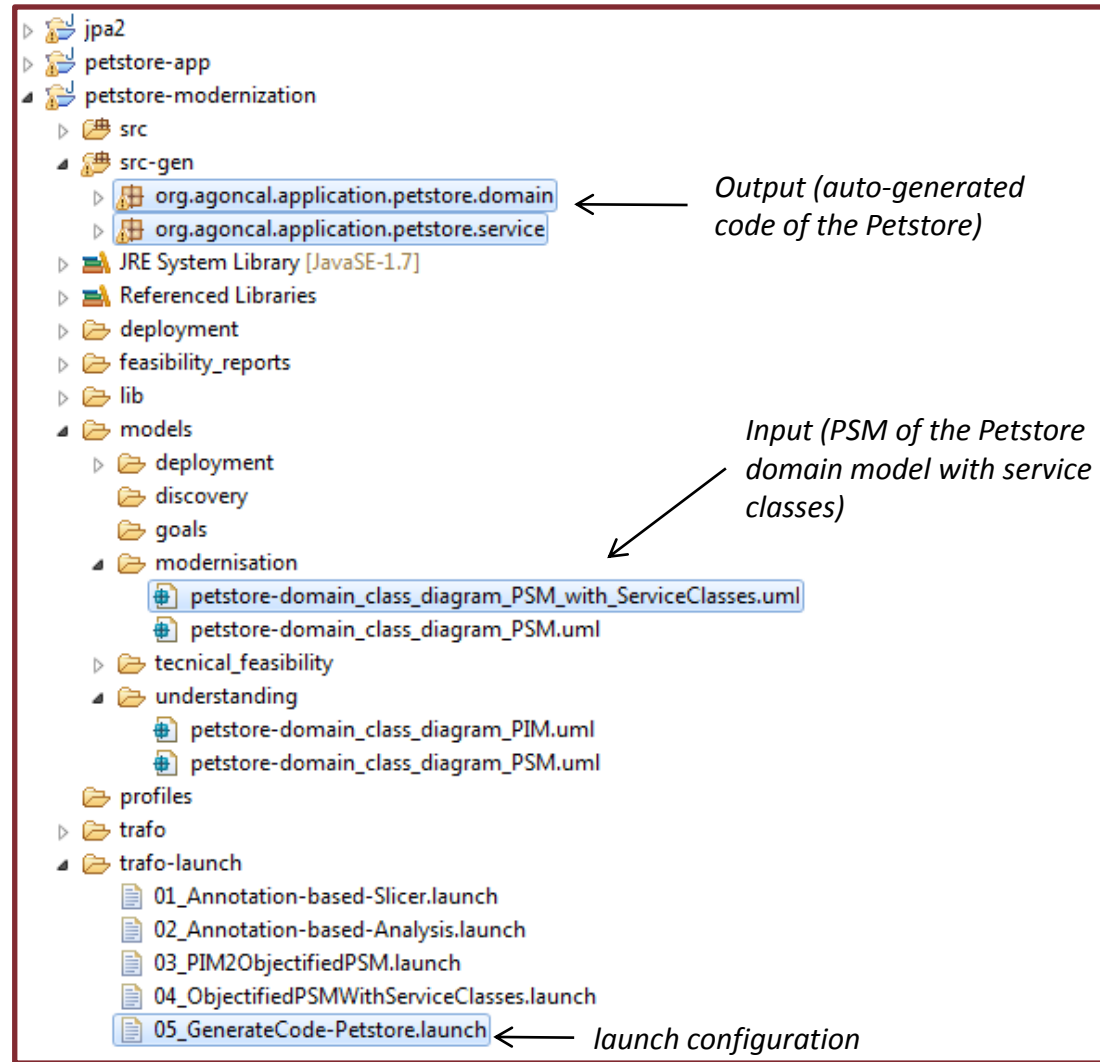
Cloudifying through JUMP (3-4)



5. Code Generation (FE)



- Extension of the Acceleo UML2 to Java code generator



Cloudifying through JUMP (5)

