Migration des SI vers le Cloud

Modernisation du patrimoine logiciel par les modèles

Olivier Le Goaer
olivier.legoaer@univ-pau.fr

Agenda

- Problématique
- Modernisation par les modèles
- ADM : La boîte à outils de l'OMG
- Projet Européen Remics

Problématique

Modernisation du SI : contexte

- Prépondérance des SI
  - Grands comptes (banques, assurance, mutuelle, …)
  - SI devenus stratégiques, voir critiques pour le métier
- Situations d'urgence
  1. Cycles technologiques courts
  2. Évolutions rapides des entreprises (structure/réglementation)
  3. Départ des « sachant » des entreprises
- Défi industriel et scientifique majeur
  - Coûts importants, ROI difficile à évaluer
  - Verrous technologiques

Logiciels patrimoniaux (legacy)

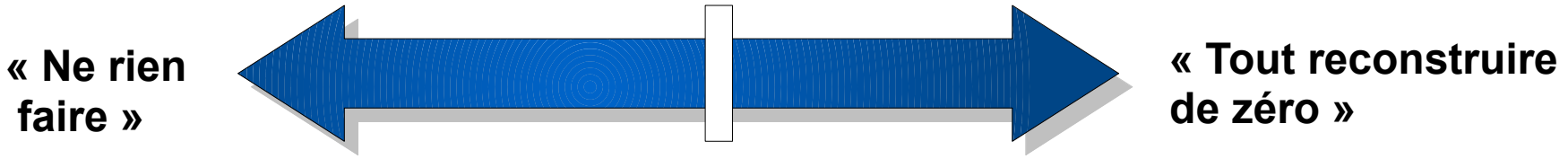
- Patrimoine logiciel recouvre 2 notions :
  - Valeur : du point de vue comptable, le SI est un **actif** de l'entreprise
  - Temps : le SI date historiquement et est devenu **obsolète**
- Modernisation du patrimoine logicielle
  - Conserver la valeur du SI tout en le mettant au goût du jour
- Projets de modernisation du SI
  - Migration du SI : le projet vise à re-localiser le SI sur une plateforme technologique récente. Typiquement iso-fonctionnelle.
  - Refonte du SI : le projet vise à rebâtir le SI pour repartir sur de meilleures bases. Les fonctionnalités peuvent être repensées.

La réalité de la modernisation

- Le rendez-vous manqué
  - La recherche fondamentale s'est concentré sur des technologies pas si veille que cela (XML, Java, …)
  - Le véritable besoin de l'industrie porte sur les gros systèmes codés en COBOL (Banques) ou en C (Telecom)
- Les SI « dinosaures »
  - Passage à l'échelle : millions de lignes de code
  - Abstraction : faiblesse de structuration (green screen, flat files, ...), manque de vue synthétique sur le système
  - Connaissance : connaissance partagée entre plusieurs intervenants, et parfois perdue

Se préparer à évoluer

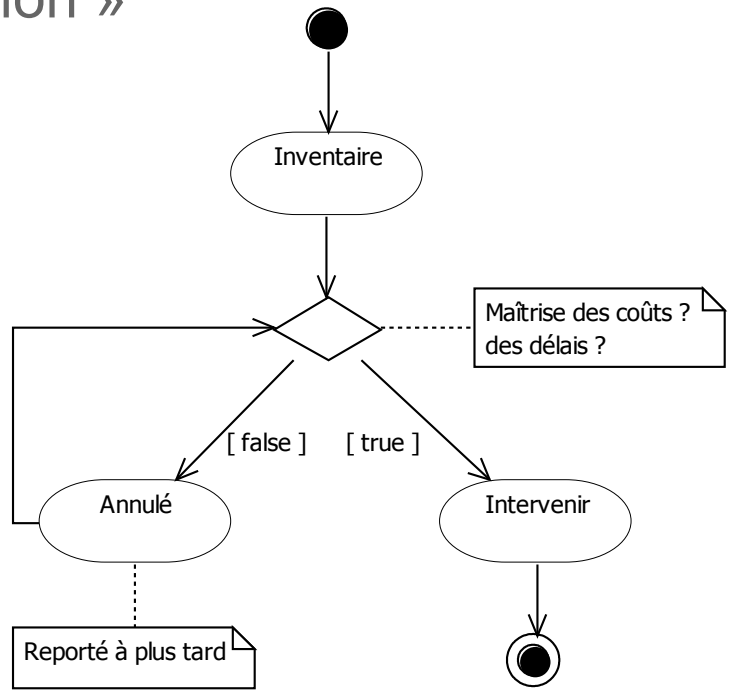
- Une option raisonnable



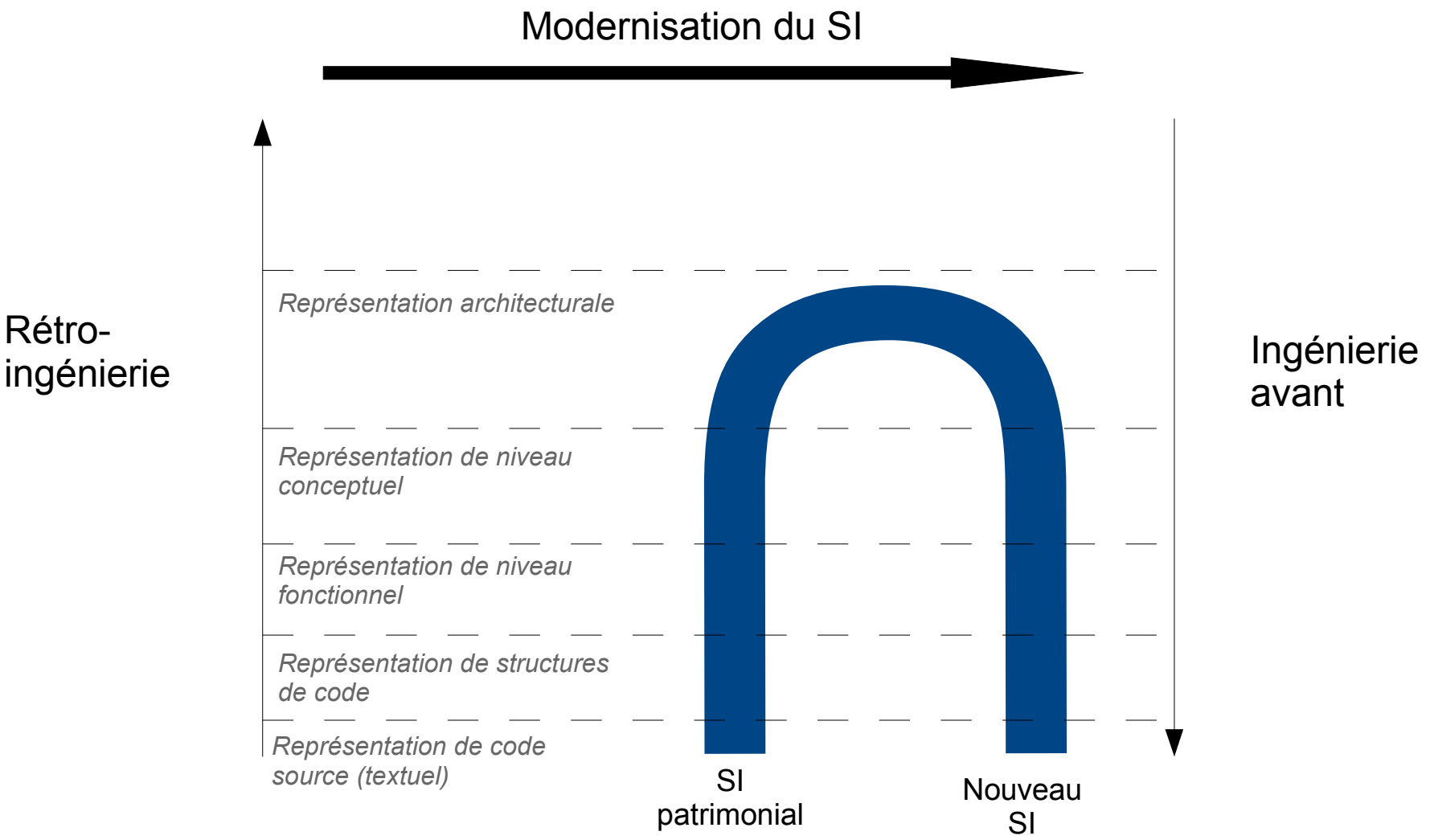
- Faire le point sur la situation
  - Inventorier le portefeuille des logiciels de chaque entreprise
  - L'occasion de (re)découvrir ce qui se cache sous l'empilement de couches logicielles (*« Software Archeology »*)
- Prendre une décision
  - Comprendre en vue d'estimer la complexité, les coûts
  - Avis d'experts, d'analystes

Outils CASE de modernisation

- Logiciels pour assister les projets de modernisation
  - « Computer-aided software modernization »
  - Fournir un support automatisé (en %)
- A chaque étape clé
  - Inventaire
  - Décision
  - Intervention
- Bénéfices attendus
  - Plus sûr, plus rapide, plus économique



Modèle du « fer à cheval »

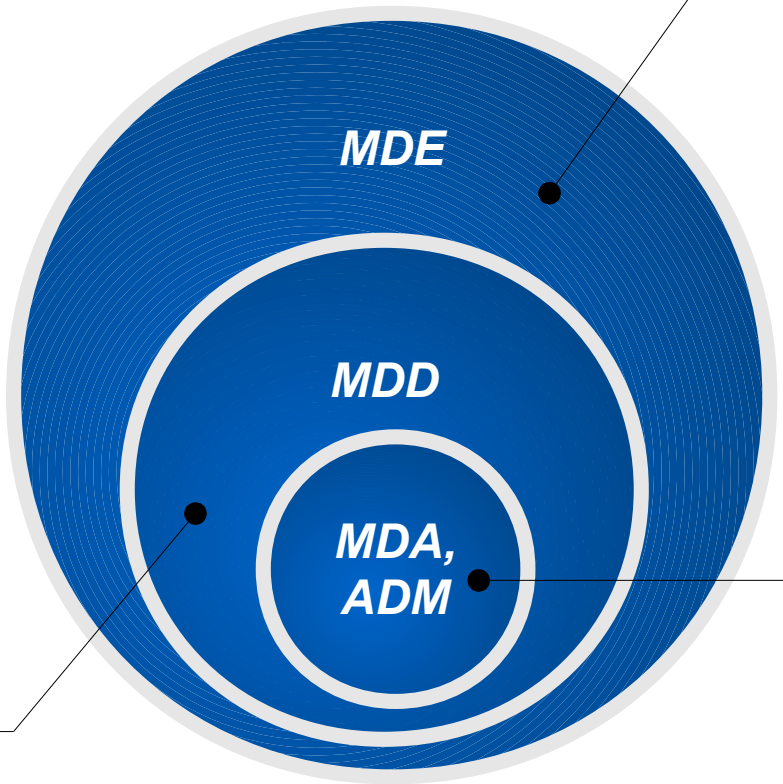


Point de vue ingénierie

1. Retro-ingénierie (*reverse-engineering*)
   - Objectifs : inventorier les artefacts du patrimoine logiciel
   - Actions : auditer, fouiller
2. Ré-ingenierie (*re-engineering*)
   - Objectifs : revitaliser ces artefacts
   - Actions : améliorer, nettoyer, refactoriser, remplacer
3. Ingénierie avant (*forward-engineering*)
   - Objectifs : cibler de nouvelles plateformes technologiques, intégrer des notions nouvelles
   - Actions : générer du code, documenter, tester

Modernisation par les modèles

Le MD*

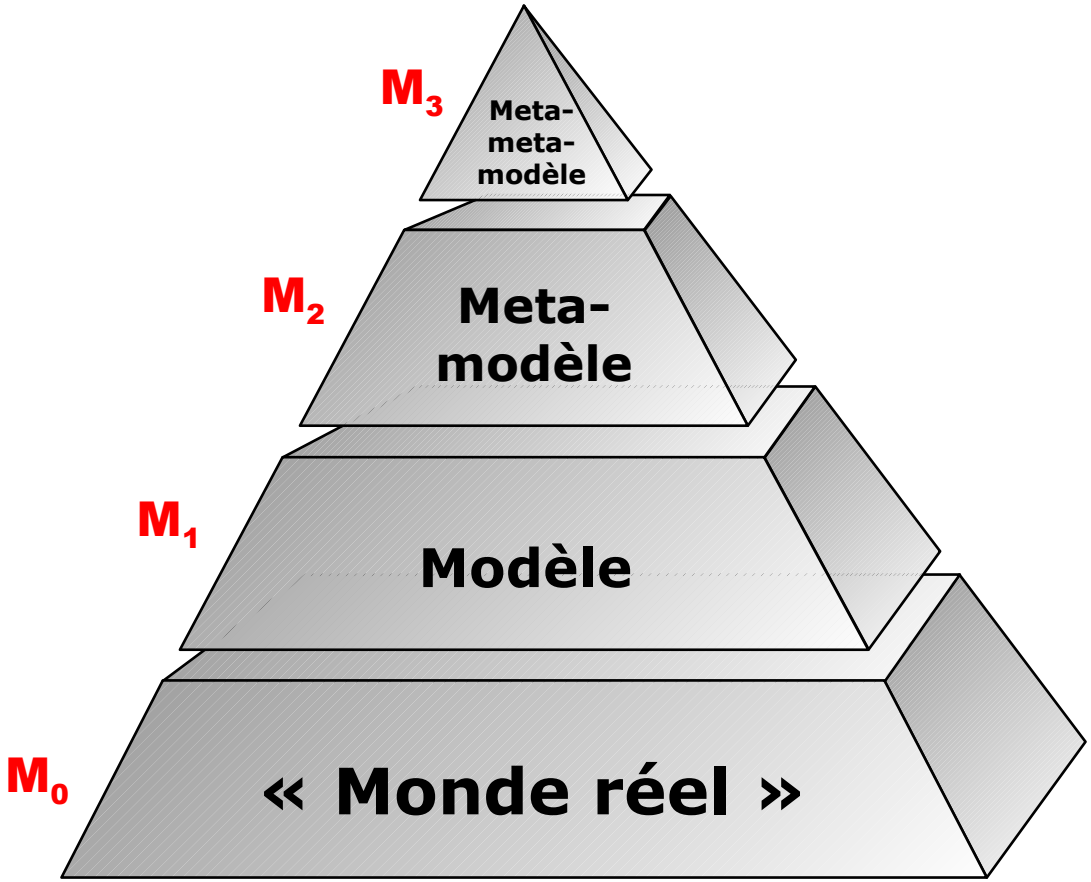


Toute activité qui considère les modèles comme citoyens de première classe (à traiter en priorité)

Vision de l'*OMG* : raffinement des modèles (PIM/PSM)

Les modèles couvrent le cycle de développement et de maintenance des systèmes logiciels

La pile de métamodélisation



Le MOF

Le métamodèle UML et d'autres métamodèles

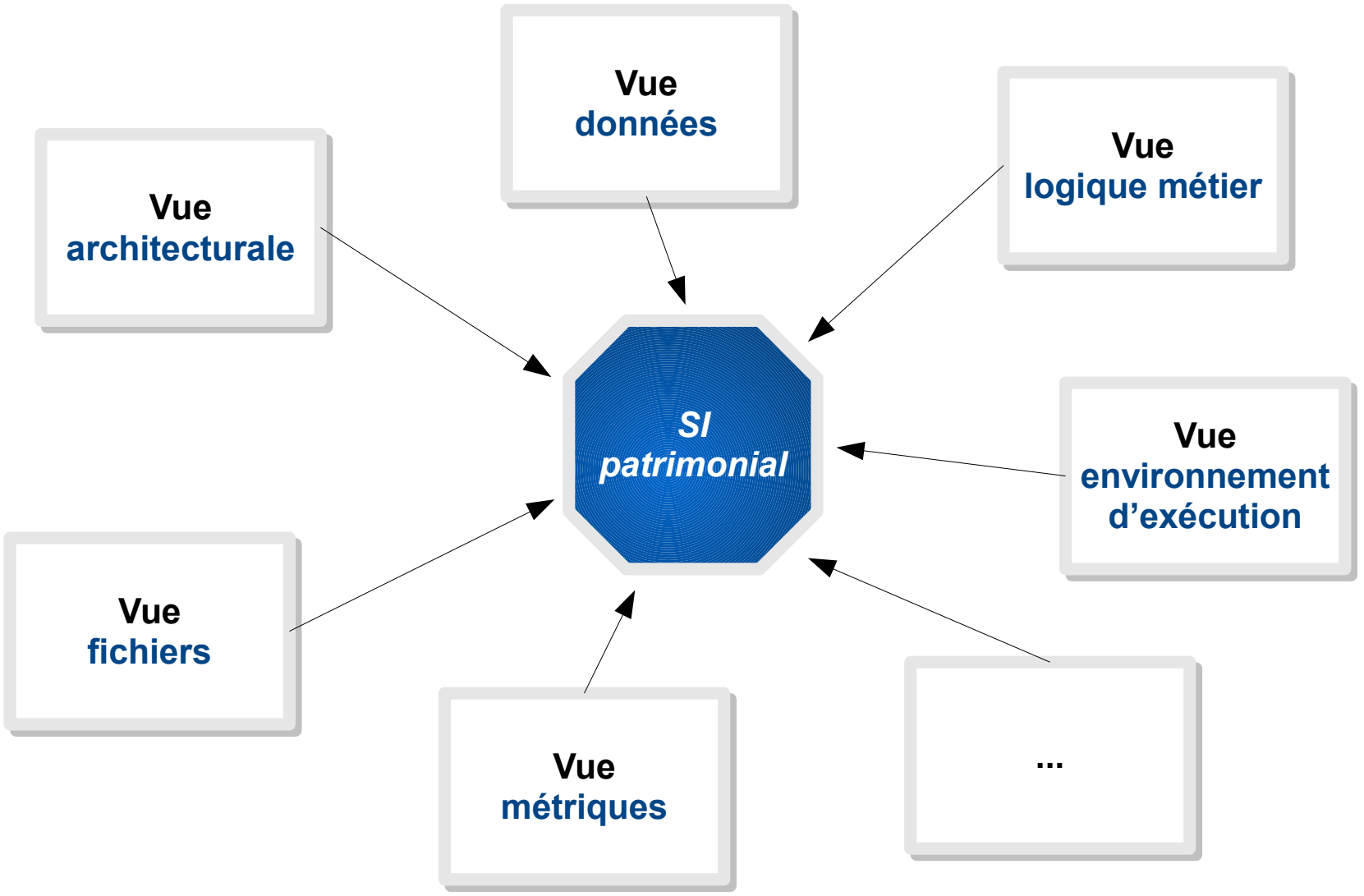
Des modèles UML et d'autres modèles

Usages variés de ces modèles

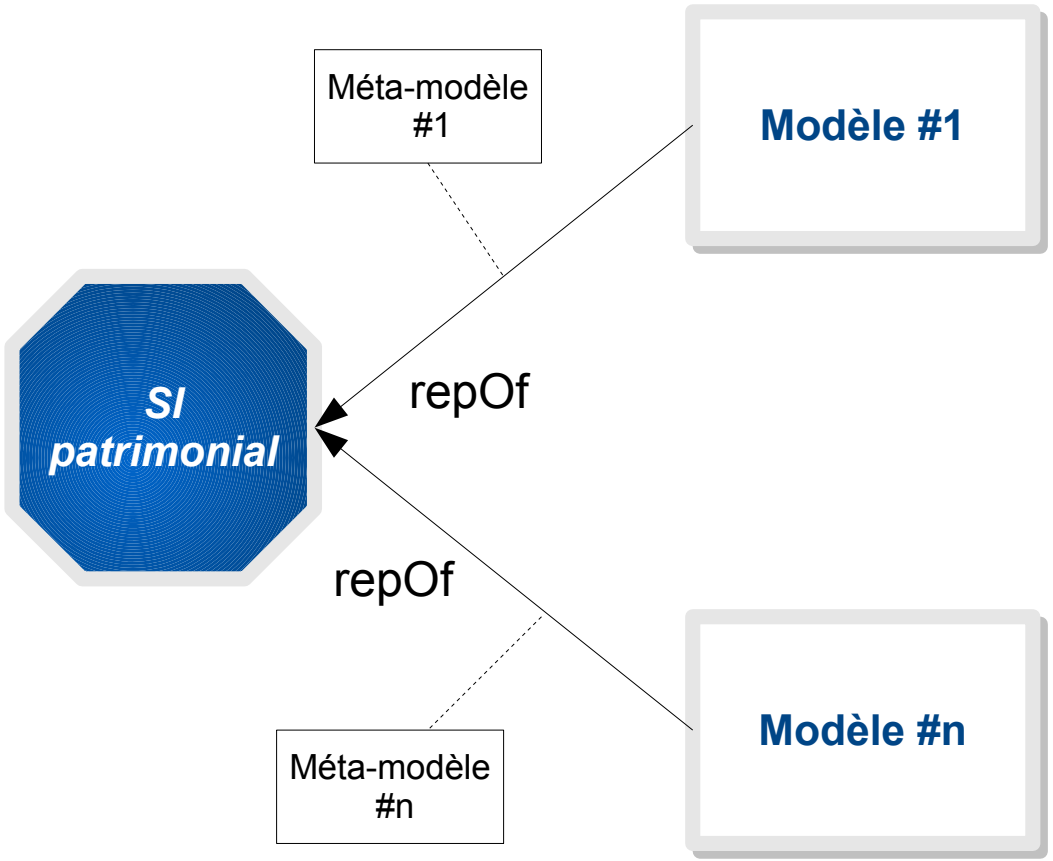
Model-driven Modernization

- La modernisation est un processus complexe
  - Nécessite une forte montée en abstraction !
- L'IDM comme technologie idéale
  - Capturer un point de vue sur le SI sous la forme d'un **modèle** (conforme à un **méta-modèle**)
  - Le traiter sous la forme d'une **transformation**
- Un projet de modernisation fera intervenir de nombreux de modèles qu'il faudra gérer
  - Méga-modélisation
  - Bus à modèles
  - ...

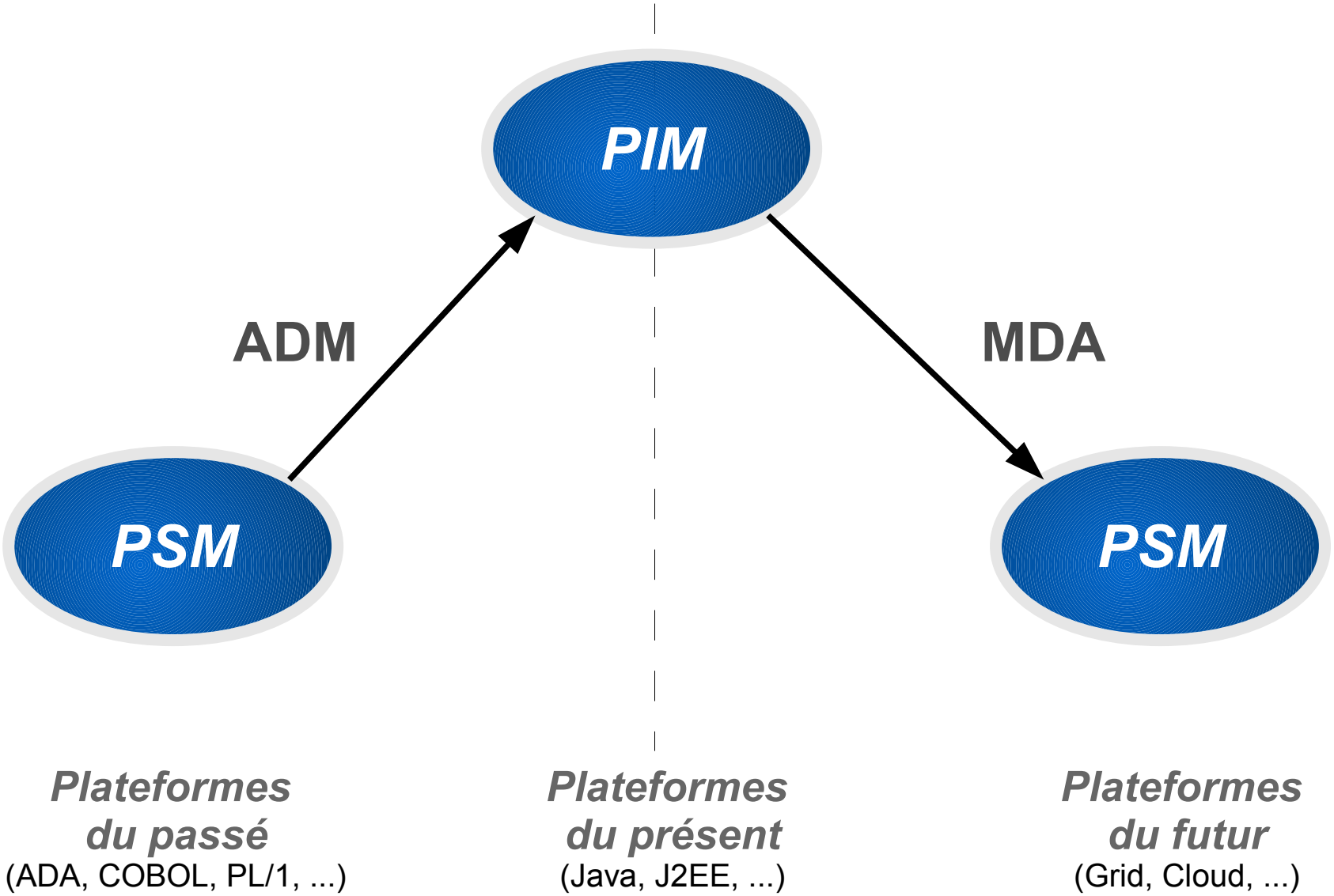
Approche multi-vues



Multi-vues par les modèles



La modernisation selon l'OMG

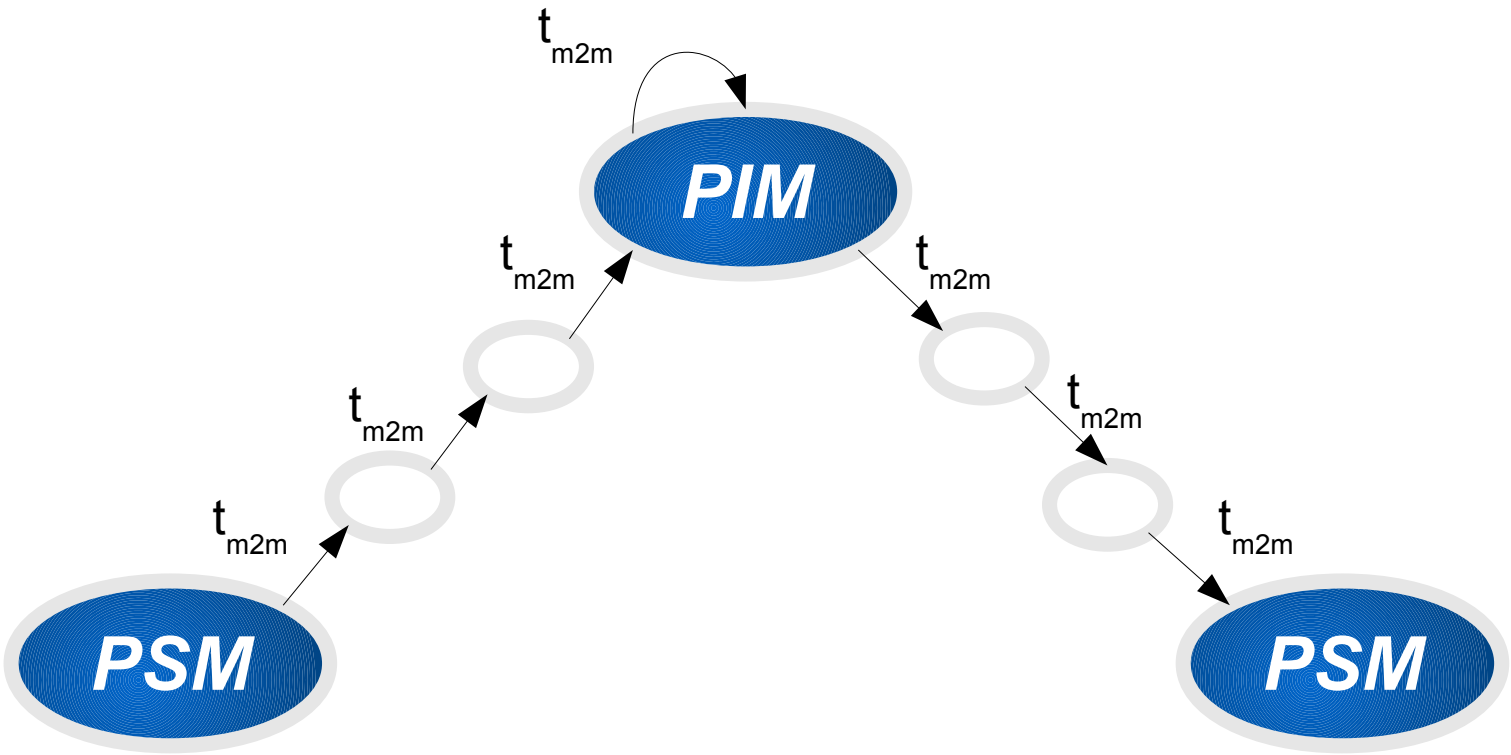


OMG - ADM | MDA

- Initiatives complémentaires de l'OMG
  - ADM : en lien avec le *reverse-engineering*
  - MDA : en lien avec le *forward-engineering*
- L'OMG comme cadre normatif
  - Spécifie (entre autres) des métamodèles
  - Leur implémentation n'est pas fournie
- Bénéfices à court-termes
  - Standardisation
  - Interopérabilité entre les outils, échange des modèles

Discrétisation du processus

- Chaîne de transformations
- Modèles (et méta-modèles) intermédiaires



ADM : La boîte à outils de l'OMG pour le reverse

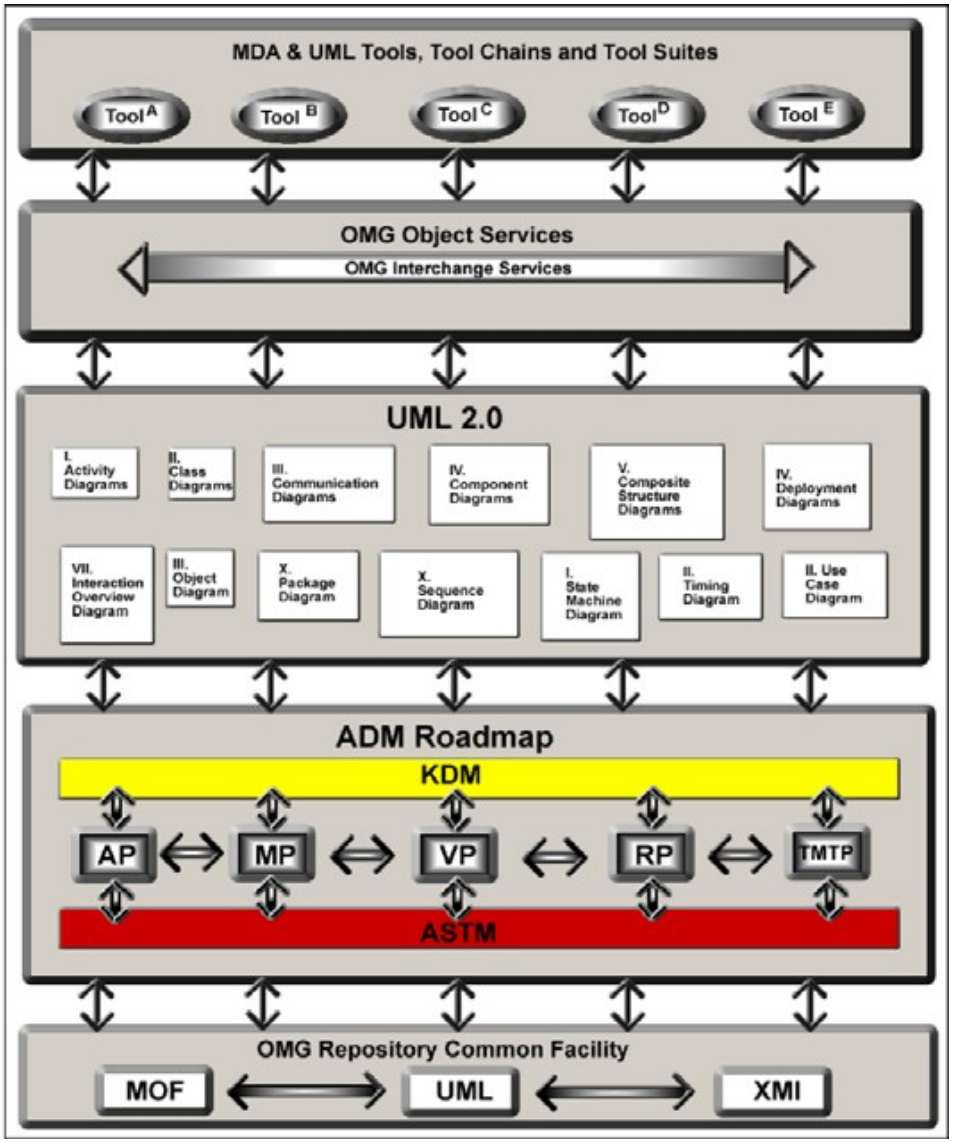
OMG ADM Task Force

- Groupe créé en 2003
- Mission
  - « Create specifications and promote industry consensus on modernization of existing applications »
- Plan d'attaque
  - 7 paquetages (*Request For Proposal*)
  - 12 scénarios de modernisation
- Pilotage
  - Djenana Campara
  - William Ulrich

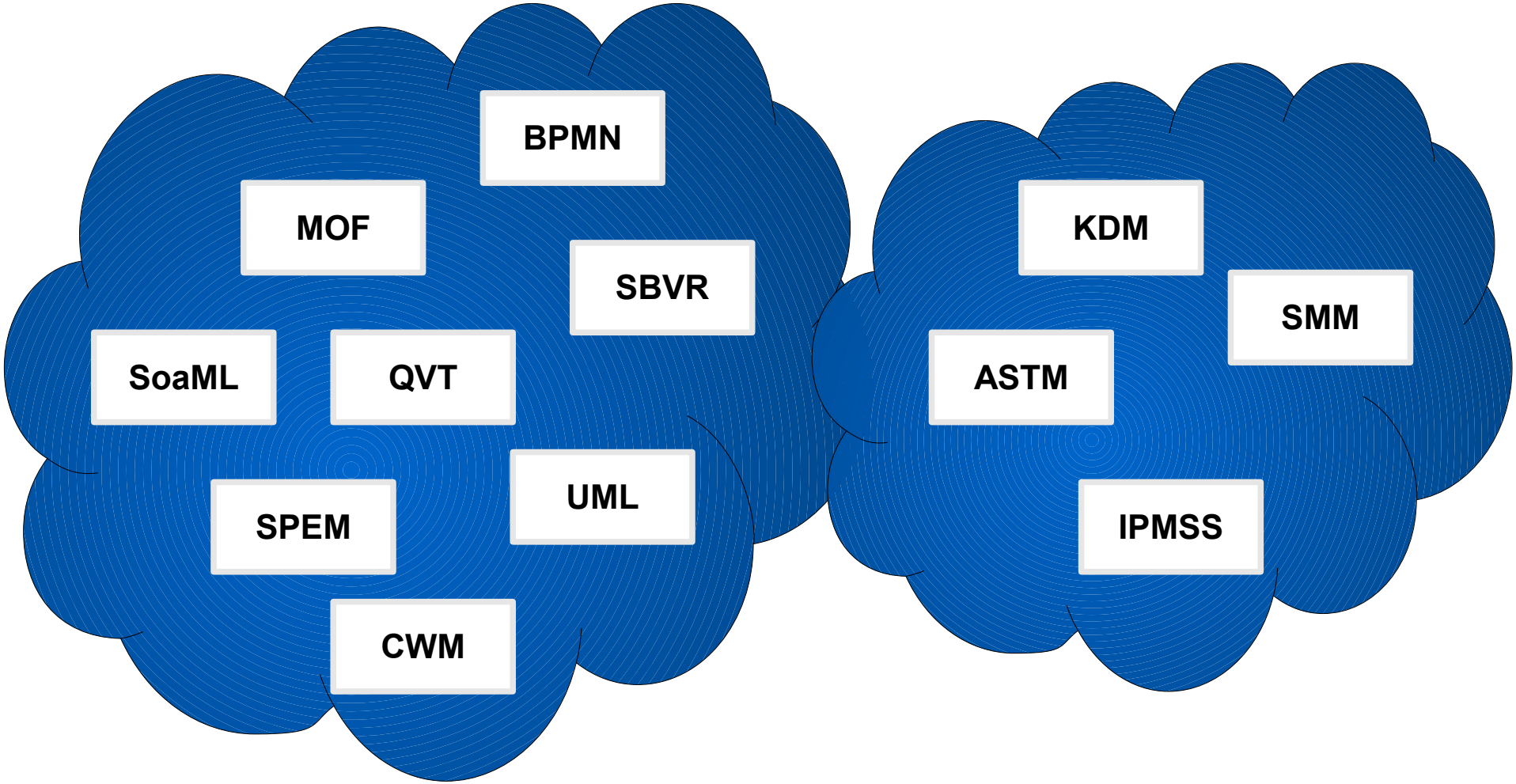
ADM Roadmap

- RFP #1: Knowledge Discovery Meta-Model (KDM) Package
- RFP #2: Abstract Syntax Tree Meta-Model (ASTM) Package
- RFP #3: Analysis Package (AP)
- RFP #4: Metrics Package (MP)
- RFP #5: Visualization Package (VP)
- RFP #6: Refactoring Package (RP)
- RFP #7: Target Mapping & Transformation Package (TMTP)

Interopérabilité selon l'OMG



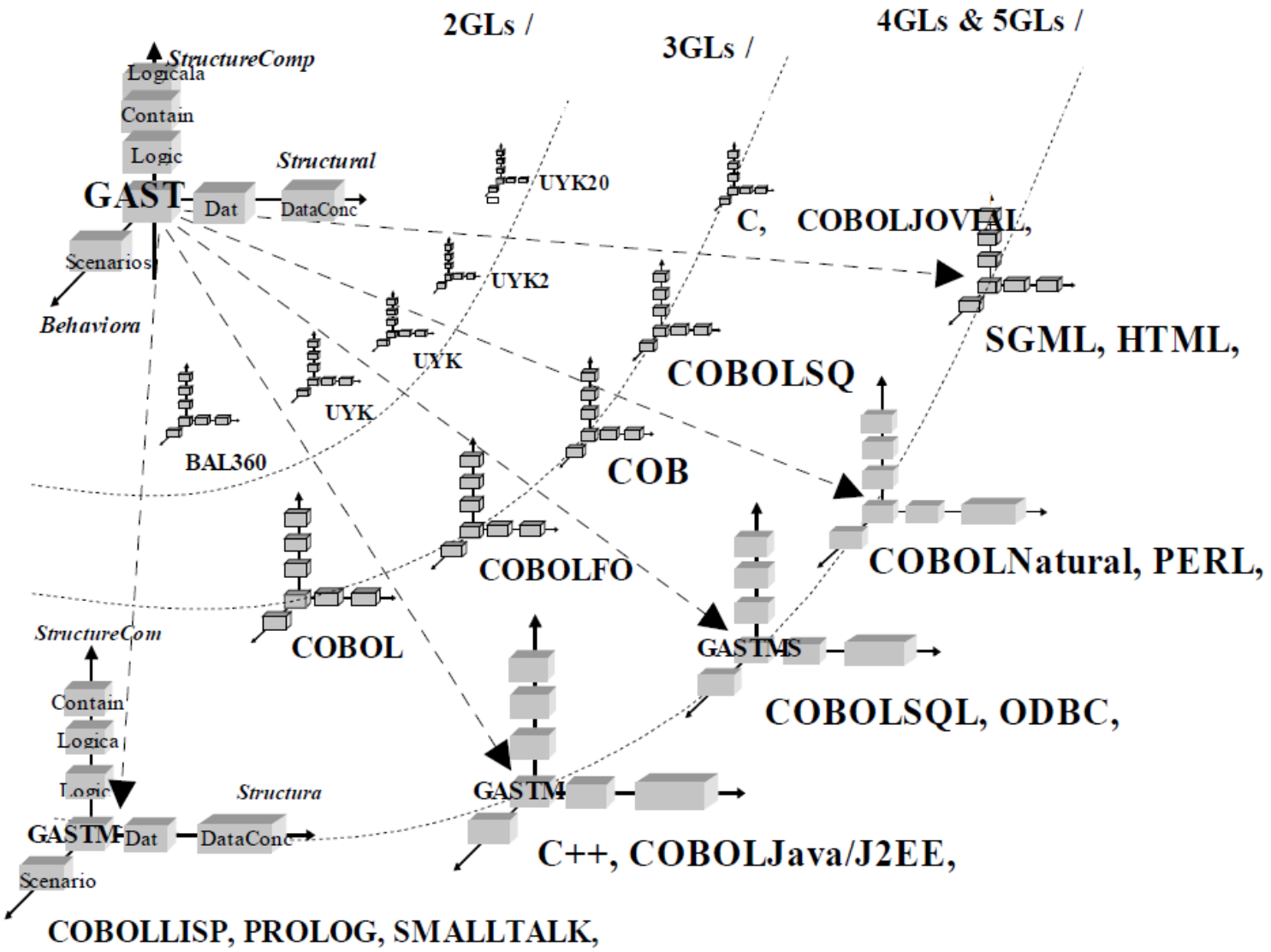
Écosystème OMG (metamodèles)



Focus sur ASTM

- Abstract Syntax Tree Metamodel (ASTM)
  - Prévu pour une modélisation bas-niveau, fidèle au code source
  - Supporte différentes familles de langages : programmation essentiellement, d'interrogation, de transformation, ...
- ASTM = GASTM + SASTMs
  - GASTM (Generic ASTM) : metamodèle commun pour représenter un code source
    - *Méta-types : DeclarationOrDefinition, Expression, Literal, etc...*
  - SASTM (Specific ASTM) : metamodèle dédié à chaque langage
    - *Méta-types : TernaryOperator (Java), MoveStatement (COBOL), etc*

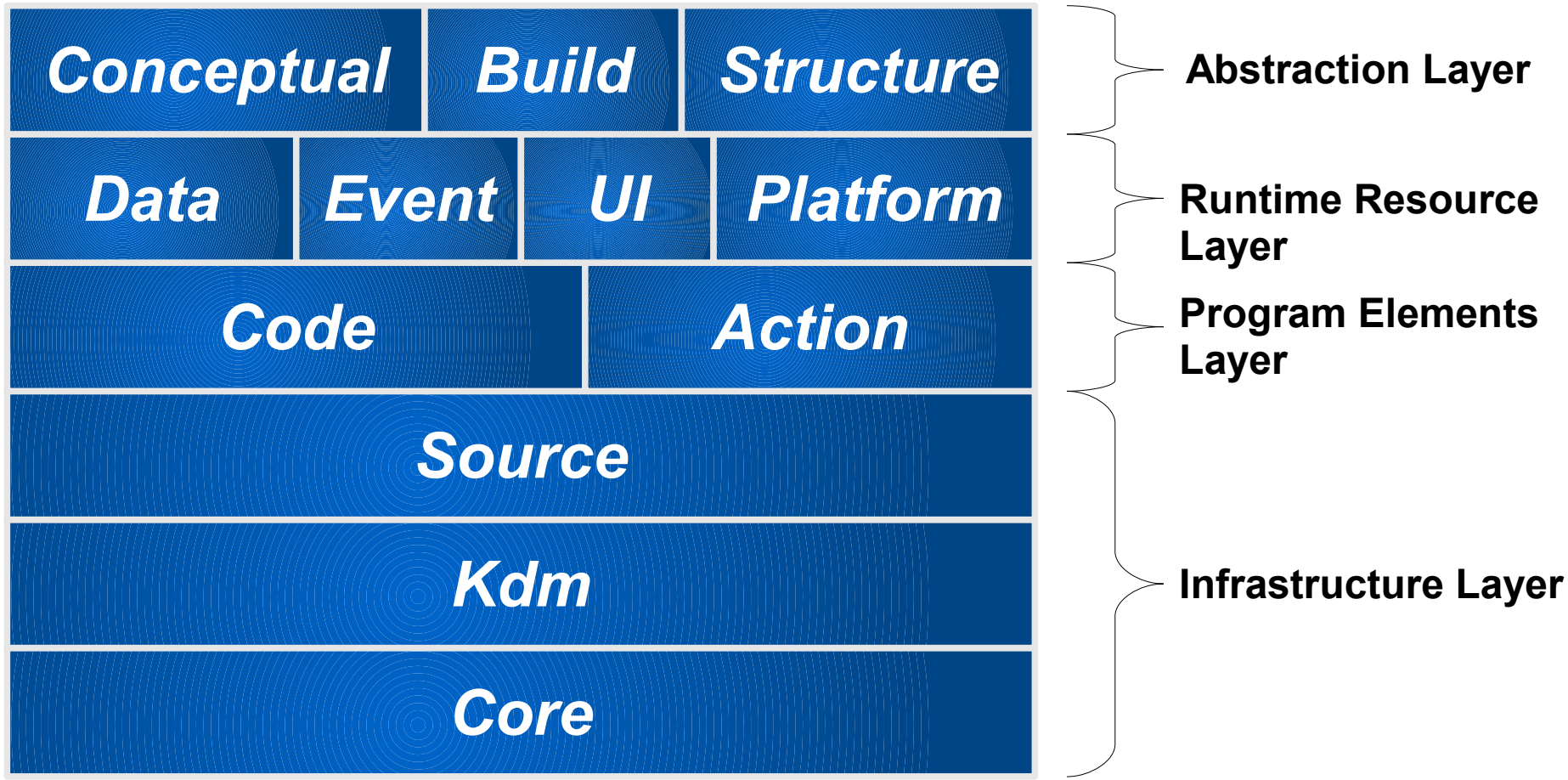
ASTM et langages



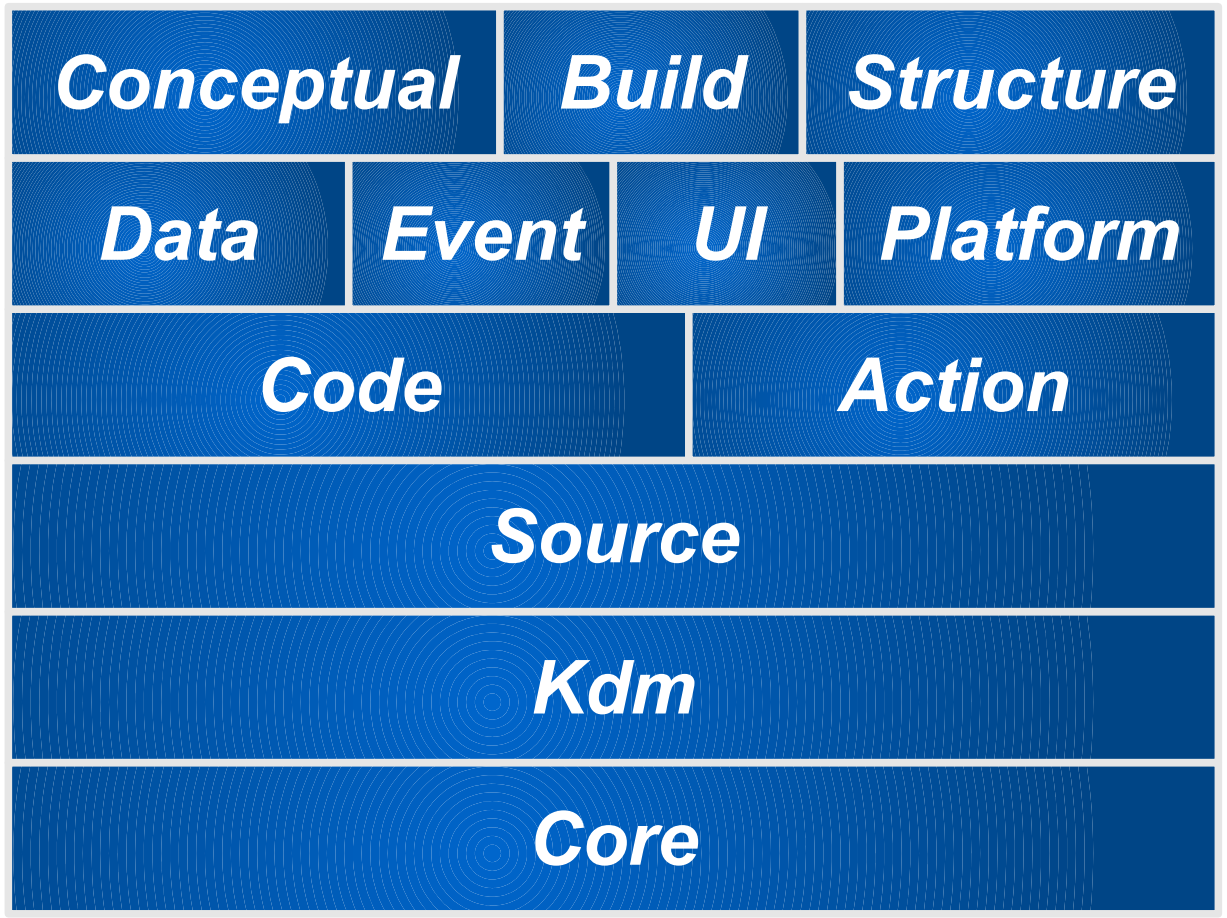
Focus sur KDM

- Knowledge Discovery Metamodel (KDM)
  - Prévu pour une modélisation haut-niveau, une « cartographie » du SI patrimonial selon plusieurs points de vues (cf. diapo 17)
  - Fournit différents « micro-langages » couvrant les aspects comportement, structure, et données du SI
  - Sert de modèle pivot à partir duquel on va dériver de nouveaux modèles interopérables (i.e. des PIMs)
    - *UML2, SOAML, BPMN, SBVR, …*
- Implémentation connue
  - KDM Analytics (http://www.kdmanalytics.com/)

Architecture de KDM



Degré d'exploitation de KDM



Haut-niveau, implicites, jugement d'experts/analystes

Primitives, explicites, extraction automatique

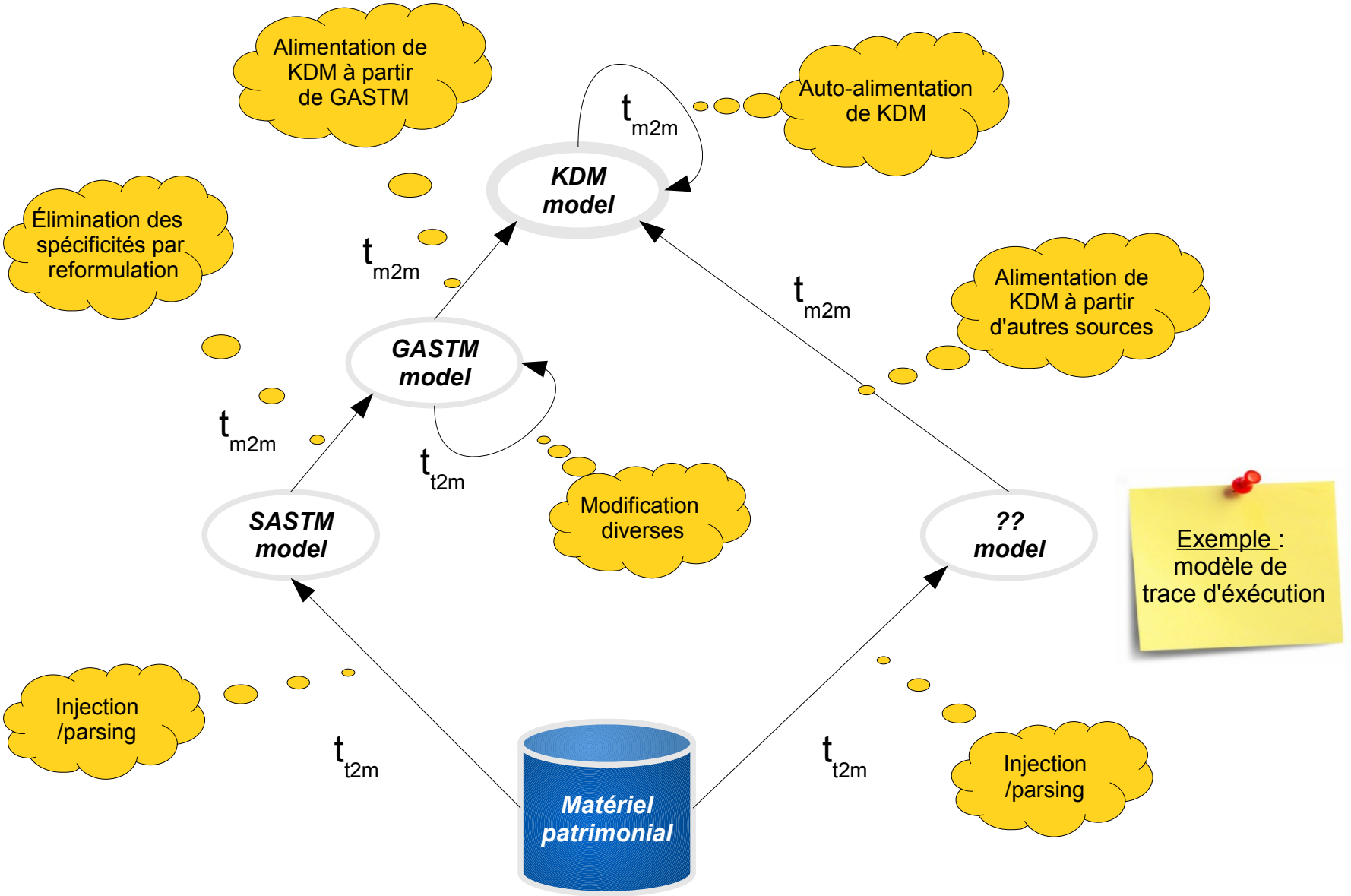
Framework

Méta-modélisation

SASTM, GASTM, et KDM

- SASTM et GASTM
  - Chaque SASTM étend GASTM
  - Une transformation SASTM2GASTM peut être nécessaire pour éliminer les spécificités du langage
- Complémentarité de ASTM et KDM
  - ASTM (bas-niveau d'abstraction) sert à alimenter KDM (haut-niveau d'abstraction)
  - Le paquetage `Code` de KDM n'a pas été prévu pour une modélisation du code en dessous du niveau procédure
  - ASTM est prévu pour conserver la syntaxe concrète/de surface aux niveau des nœuds de l'arbre

Reverse « ADM-compliant »

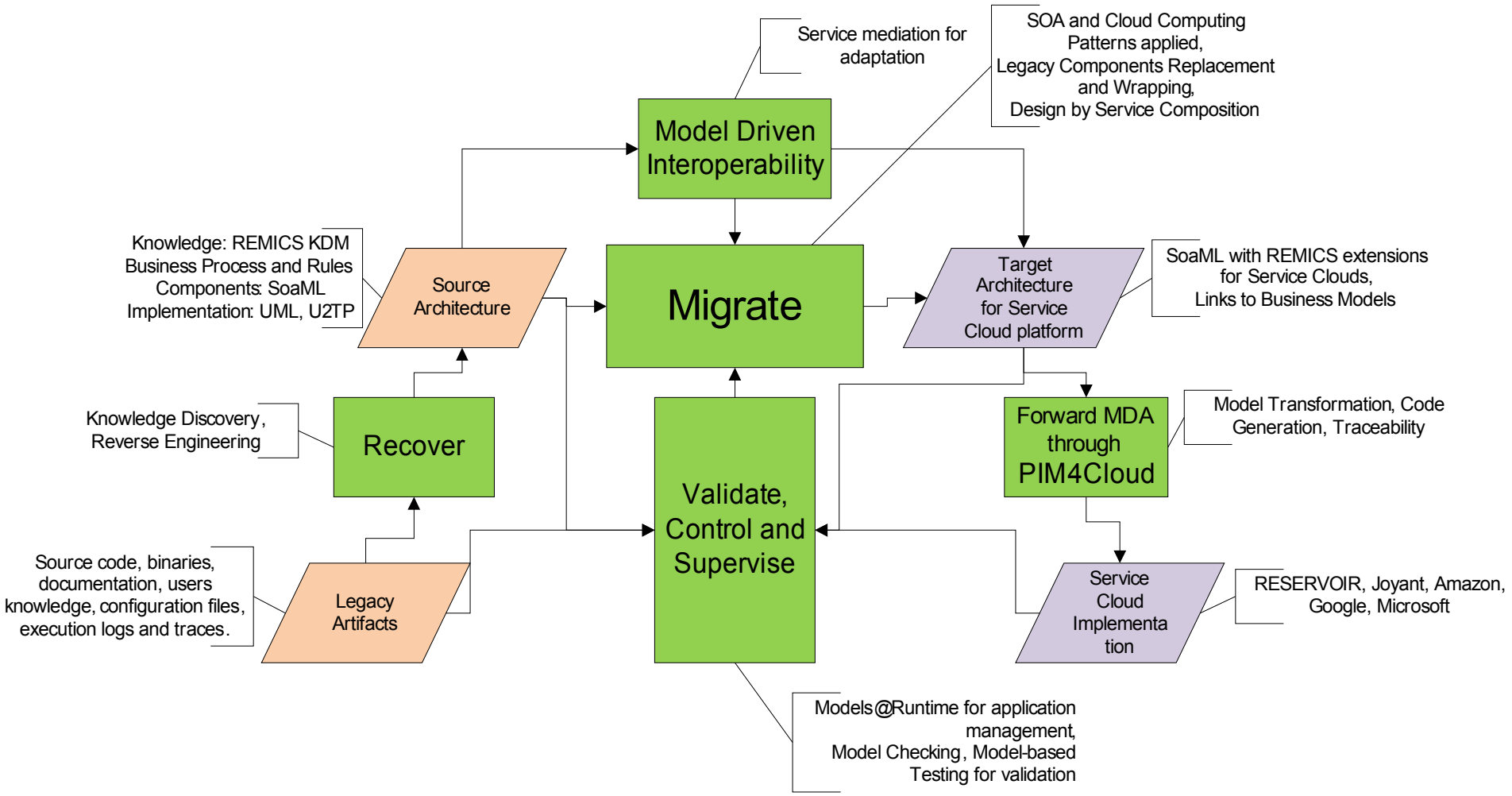


Projet Européen Remics

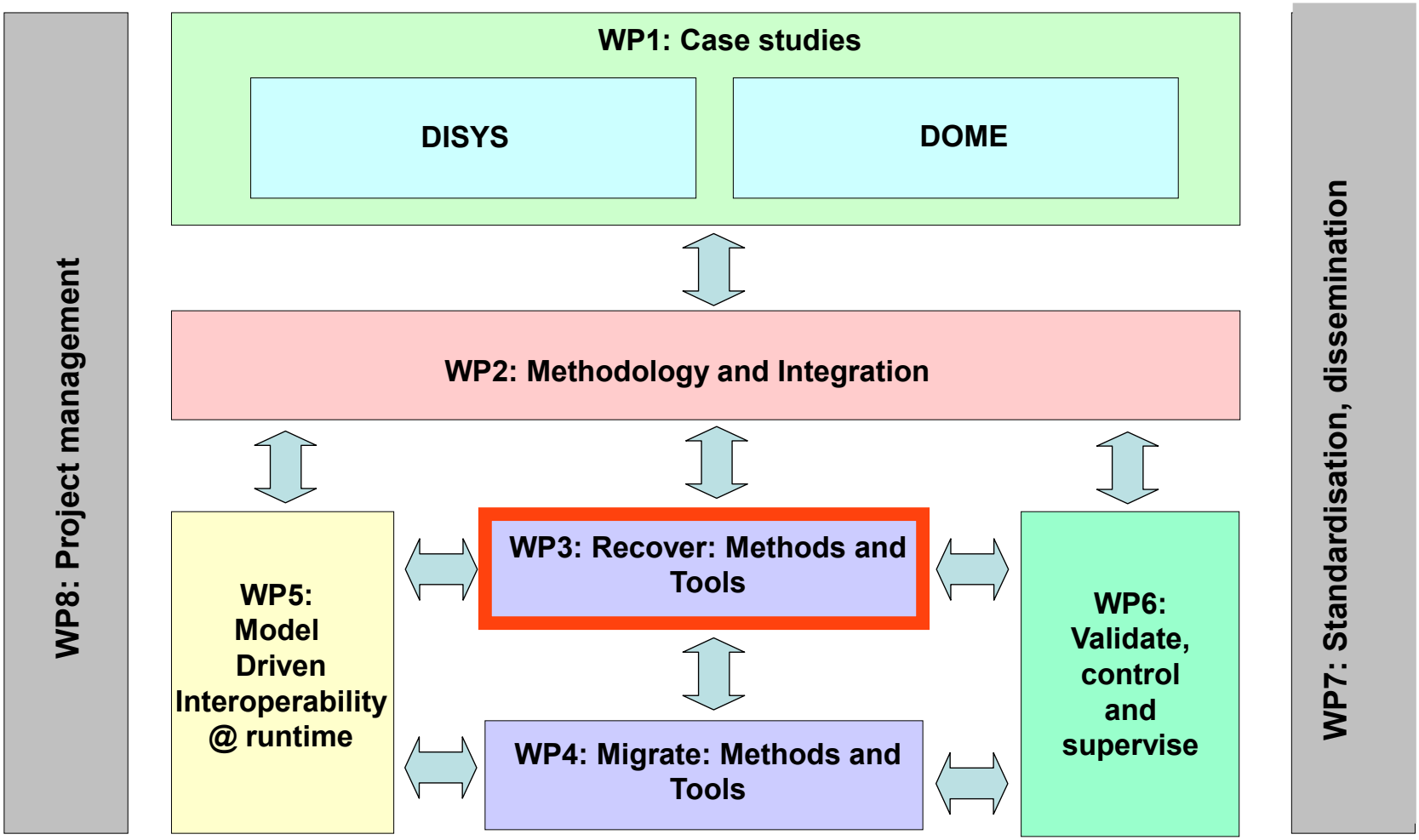
Projet Remics

- Remics
  - **Re**use and **M**igration of legacy applications to **I**nteroperable **C**loud **S**ervices
  - www.remics.eu/
- Participants
  - Consorsium industriels/académiques
- 8 Work Packages (WP), dont :
  - WP3 : Recover (*Netfective Technology*)
  - WP4 : Migrate

Projet Remics



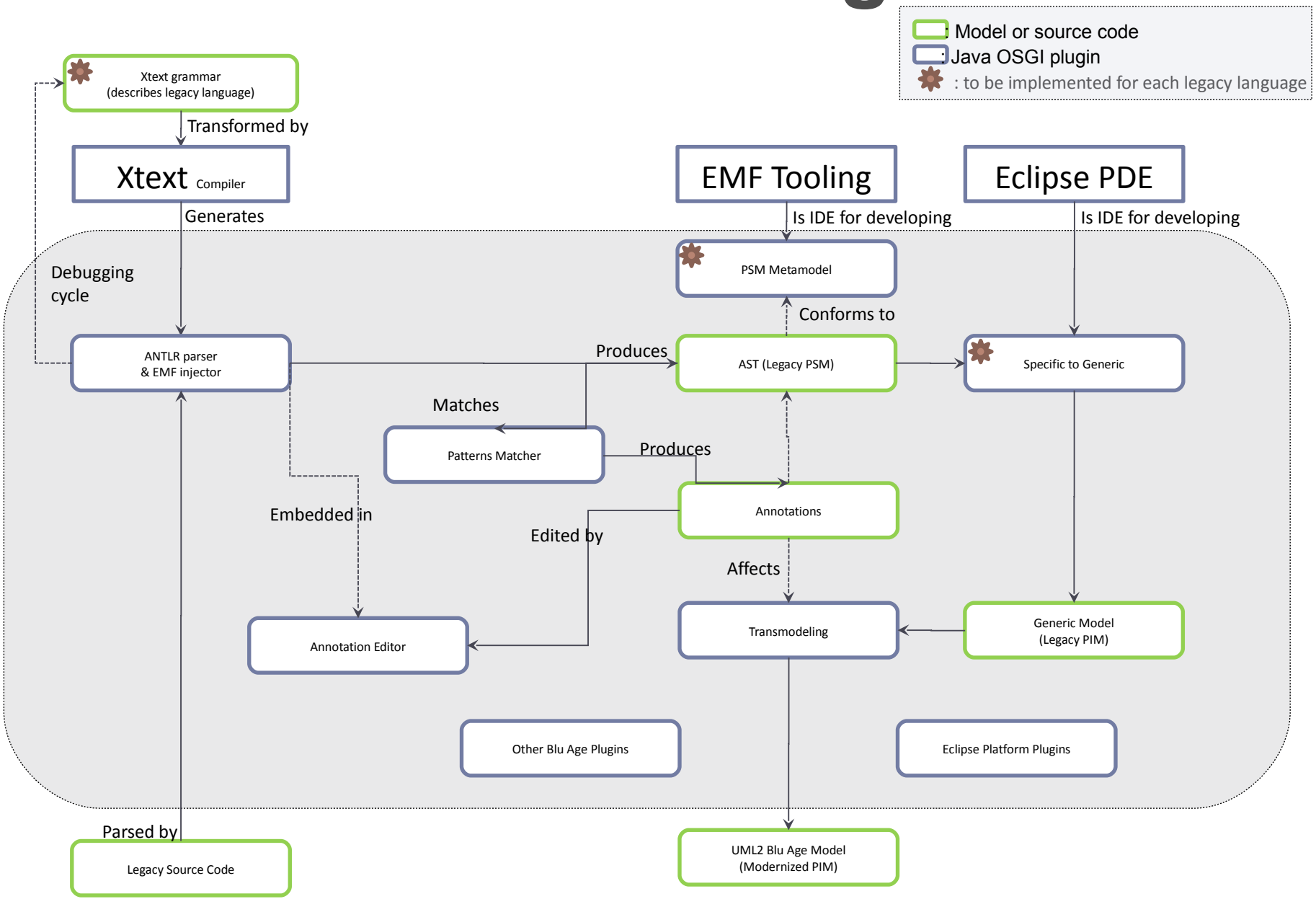
Structure des WPs



Netfective BluAge®

- Module BluAge® Reverse
  - Phase de rétro-ingénierie
  - Produit des modèles UML2 en sortie
- Module BluAge® Analyst
  - Phase décisionnelle
  - Produit des modèles de «dashboard» en sortie
- Module BluAge® Design&Generate
  - Phase d'ingénierie avant
  - Prend des modèles UML2 en entrées

Architecture de BluAge® Reverse



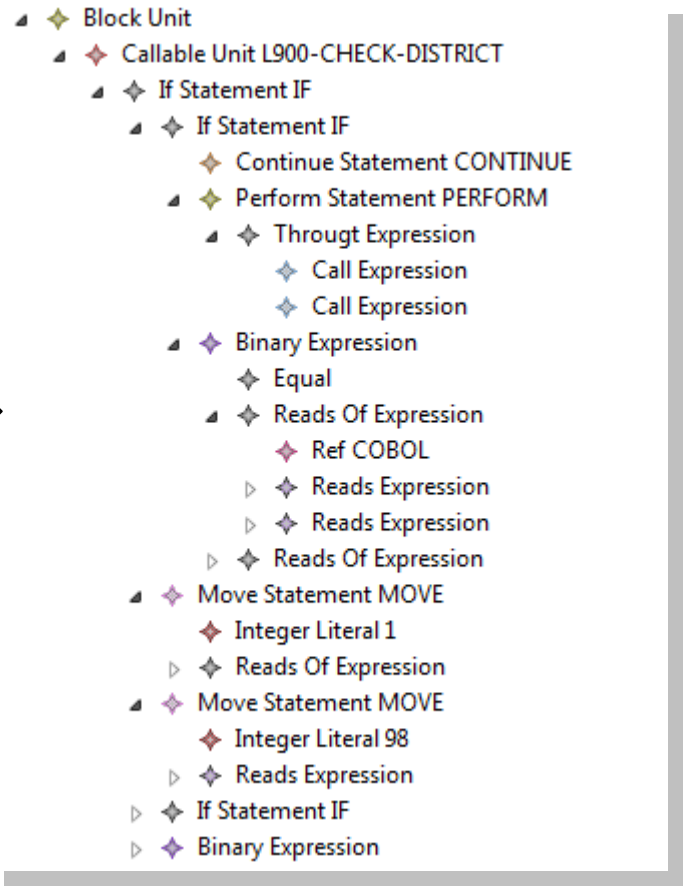
Code cobol → SASTM cobol

```
L900-CHECK-DISTRICT.

    IF ENTRY-DISTRICT     OF RPL-OUT > ZEROS
        IF ENTRY-DISTRICT OF RPL-OUT = PREV-DISTRICT OF RPL-OUT
            CONTINUE
        ELSE
            PERFORM L910-CHECK-MFLOCNBT  THRU L910-EXIT
        END-IF
    ELSE
        MOVE 1  TO SH-ENTRY-DISTRICT OF IPM-PRICE-BASIS-INCLUSIONS OF RPL-OUT
        MOVE 98 TO REPLY-CODE

        IF IP-HEADQUARTERS-USER       OF MSG-IN
        OR IP-SERVICE-CENTER-USER     OF MSG-IN
        OR IP-CONTROL-LOCATION-USER   OF MSG-IN
            MOVE MSG-MUST-ENTER-DISTRICT
              TO ERROR-MESSAGE-KEY  OF ERROR-MESSAGE-LINKAGE
        ELSE
            MOVE MSG-IP-DISTRICT-EMPTY
              TO ERROR-MESSAGE-KEY  OF ERROR-MESSAGE-LINKAGE
        END-IF
    END-IF
    .
L900-EXIT.  EXIT.
```

t_{t2m}




```
MOVE returns cobolStatement::MoveStatement:
    name=C_Move sourceElement=(CORRESPONDING|Expression) C_TO (targetElement+=basicExpressionNoComma ','?)*
;
```

SASTM cobol → GASTM

Specific:

- Move Statement MOVE
  - Integer Literal 1
  - Reads Of Expression
    - Ref COBOL
    - Reads Expression
      - Ref COBOL
    - Reads Of Expression
      - Ref COBOL
      - Reads Expression
        - Ref COBOL
      - Reads Expression
        - Ref COBOL

t_{m2m}

Generic:

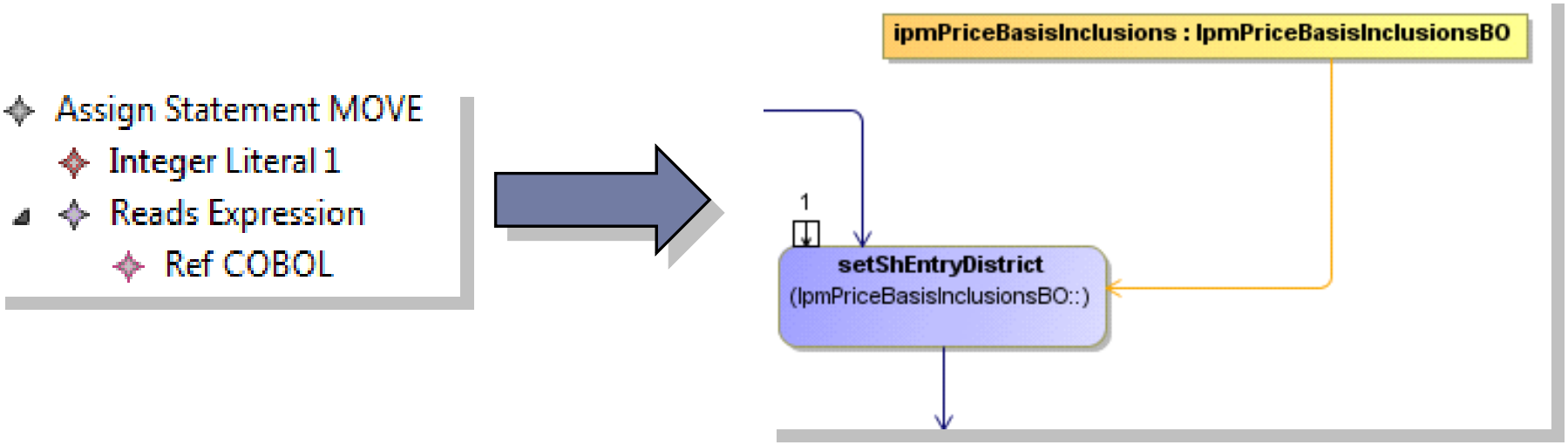
- Assign Statement MOVE
  - Integer Literal 1
  - Reads Expression
    - Ref COBOL

Autre Exemple :
opérateur ternaire
de Java
ré-écrit en if/else

Original code:

```
MOVE 1  TO SH-ENTRY-DISTRICT OF IPM-PRICE-BASIS-INCLUSIONS OF RPL-OUT
```

GASTM→UML2 + Profil



PIMs pour le Cloud

- BluAge Profile
  - Profil UML2 développé par Netfective et supporté par BluAge® Design&Generate
  - Introduit les notions d'architectures de services
- SoaML
  - Spécification de l'OMG
  - Sous-ensemble d'UML2 qui fait un usage spécialisé du diagramme de collaboration
- CloudML
  - Extension de SoaML pour le Cloud (RFP OMG)
  - Doit identifier les concepts communs à toutes les plateformes Cloud (Amazon, Windows Azure, ...)