



# Maestría en Ingeniería

## Curso de Arquitectura de Software

### Sesión 9: MDA

*Fernando Barraza A.  
fbarraza@javerianacali.edu.co*

# Sesion 9



- Objetivo: Presentar al estudiante los conceptos y aplicaciones sobre MDA.
- Temas:
  - Motivación, definición, pros y contras de MDA
  - Estructura de MDA – Modelos.
  - Procesos y desarrollo con MDA.

# Model Driven Architecture (MDA)



- Propuesta del OMG (Object Management Group)
- Una aproximación para la especificación de sistemas de software que separa la especificación de la funcionalidad del sistema de la implementación sobre una tecnología particular o plataforma.
- “Diseñe una vez, construya sobre cualquier plataforma”

# Motivación para MDA



- Un tema recurrente en la evolución del software es el uso de lenguajes formales más abstractos para modelar soluciones.
- De esta forma se pretende que se incremente la productividad al trasladar por medio de herramientas automáticas los modelos desde lo abstracto a lo concreto en lenguajes como Java o C#.

# Para que sirve MDA a las empresas?



- Integración a sistemas B2B/ B2C
- Preservar la inversión en IT cuando la compañía crece o sus tecnologías evolucionan
  - EJB 1.1 --> EJB 2.0
  - XML --> XML Schema
  - MTS --> COM+
  - CORBA 2.0 --> CORBA 3.0

# Solución propuesta por MDA



- Aislar la información y la lógica de procesamiento de las particularidades de la tecnología
- Construir dos modelos para estructurar explícitamente el sistema:
  - Platform Independent Model (PIM)
  - Platform Specific Model (PSM)

# Origen de MDA



## Propuesta del OMG (Object Management Group)

### Miembros OMG:

- AT&T
- BEA
- Borland
- Boeing
- CA
- Citigroup
- Compaq
- Ericsson
- Ford
- Fujitsu
- Glaxo
- HP
- Hitachi
- Hyperion
- IBM
- IONA
- io Software
- Kabira
- KC
- John Deere
- Microsoft
- MITRE
- MSC Soft
- NASA
- NEC
- NetGenics
- NTT
- OASIS
- Oracle
- Pfizer
- Rational
- SAGA
- SAP
- SAS
- Secant
- Siemens
- Sprint
- Sun
- Unisys
- Vertel

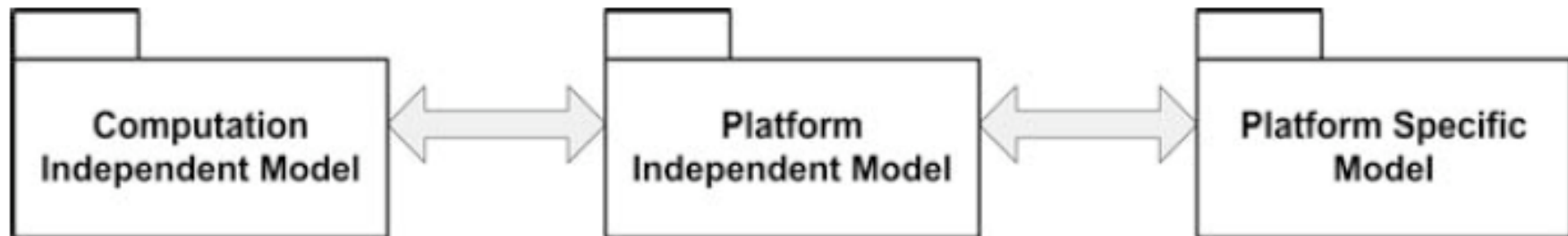
# Estructura de MDA



- Es un framework para guiar el proceso de desarrollo de software basado en la arquitectura que se apoya en UML, MOF, CWM.
- Abstrae los elementos funcionales del sistema de los tecnicismos de las plataformas



# Tres modelos de MDA



# Términos claves en MDA



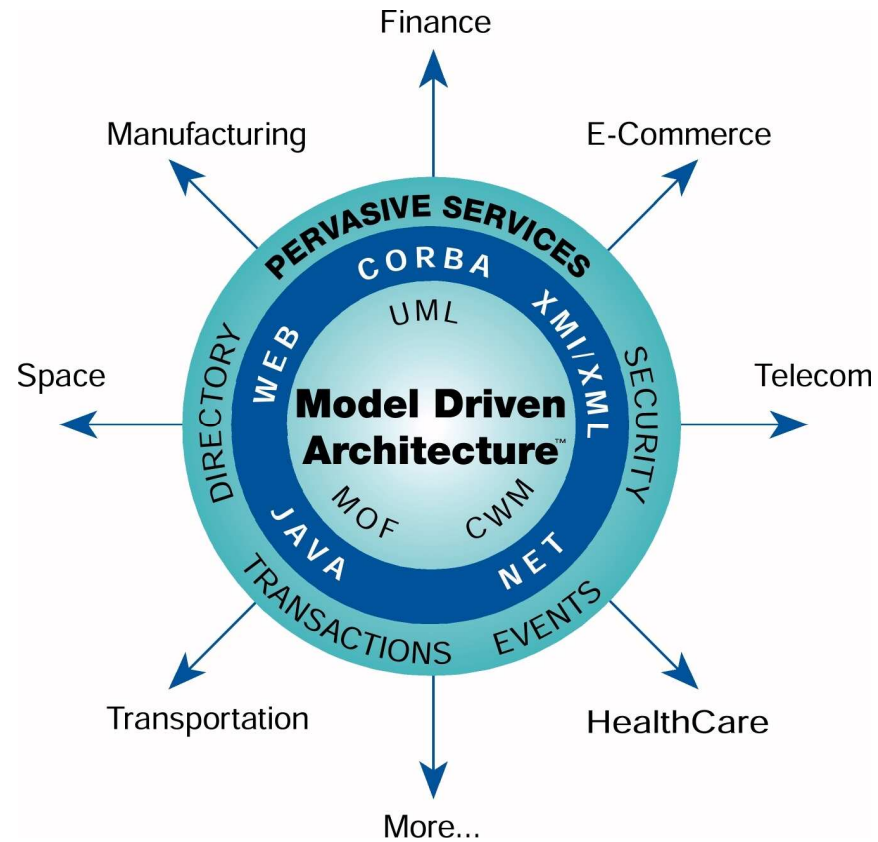
- **Modelo**
  - Una especificación formal de una función, estructura o comportamiento de un sistema (En UML por ejemplo)
- **Plataforma**
  - Detalles técnicos y de ingeniería que son irrelevantes a la funcionalidad principal de un componente de software.
- **Computation Independent Model (CIM)**
  - También conocido como el modelo de dominio. Se enfoca en el entorno y requerimientos del sistema escondiendo detalles de implementación
- **Platform Independent Model (PIM)**
  - Una especificación formal de una estructura y una función de un sistema que abstrae detalles técnicos (Ej: Una transferencia de fondos)
- **Platform Specific Model (PSM)**
  - Los detalles técnicos (CORBA, SOAP, EJB)

# Pros y contras



- Ventajas:
  - Permite portabilidad
  - Ayuda a validar la correctitud de la arquitectura
  - Las herramientas generan una buena parte del código
- Desventajas
  - Los mecanismos de mapeos pueden estar estandarizados lo que puede generar dependencias de los modelos con el código

# Esquema de MDA



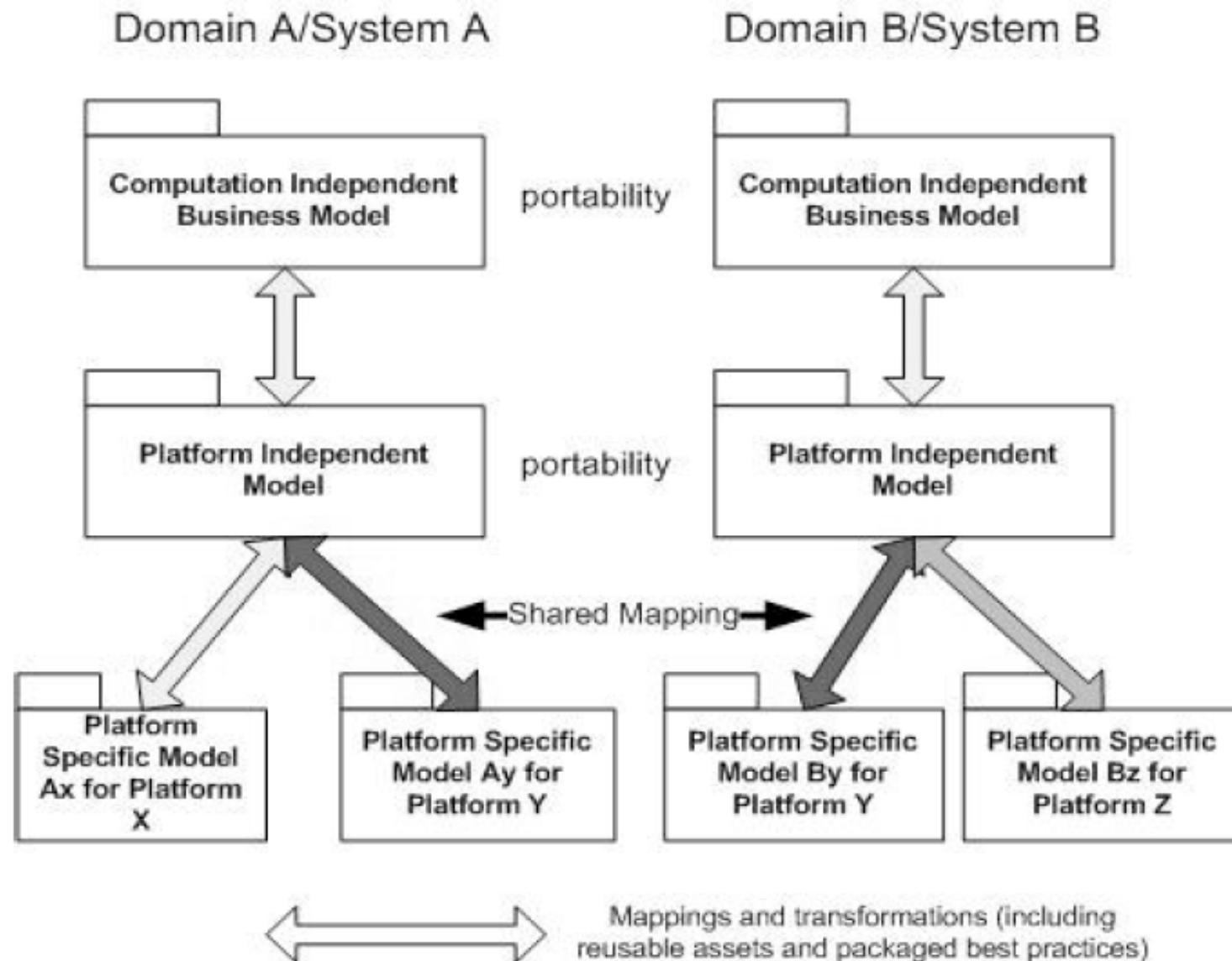
- Las plataformas objetivo en MDA:
  - CORBA
  - Java/J2EE
  - C#/.NET
  - XML/SOAP
- MDA provee perfiles UML para dichas plataformas

# Servicios dominantes



- Servicios esenciales para la mayoría de aplicaciones
  - Transacciones
  - Servicio de directorio
  - Seguridad
  - Persistencia
  - ...
- Se definen en el modelo PIM

# Proceso en MDA



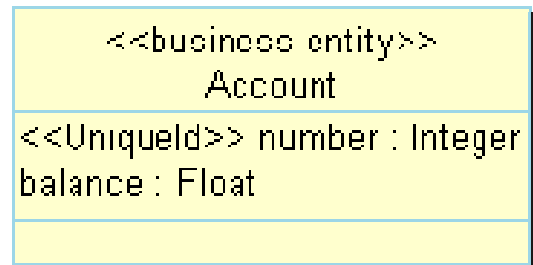
# Platform Independent Model (PIM)



- Es una especificación “formal” de la estructura y función de un sistema que abstrae los detalles técnicos.
- Se expresa generalmente utilizando UML
- Se pueden utilizar otros lenguajes (basados en MOF)



# PIM: Ejemplo



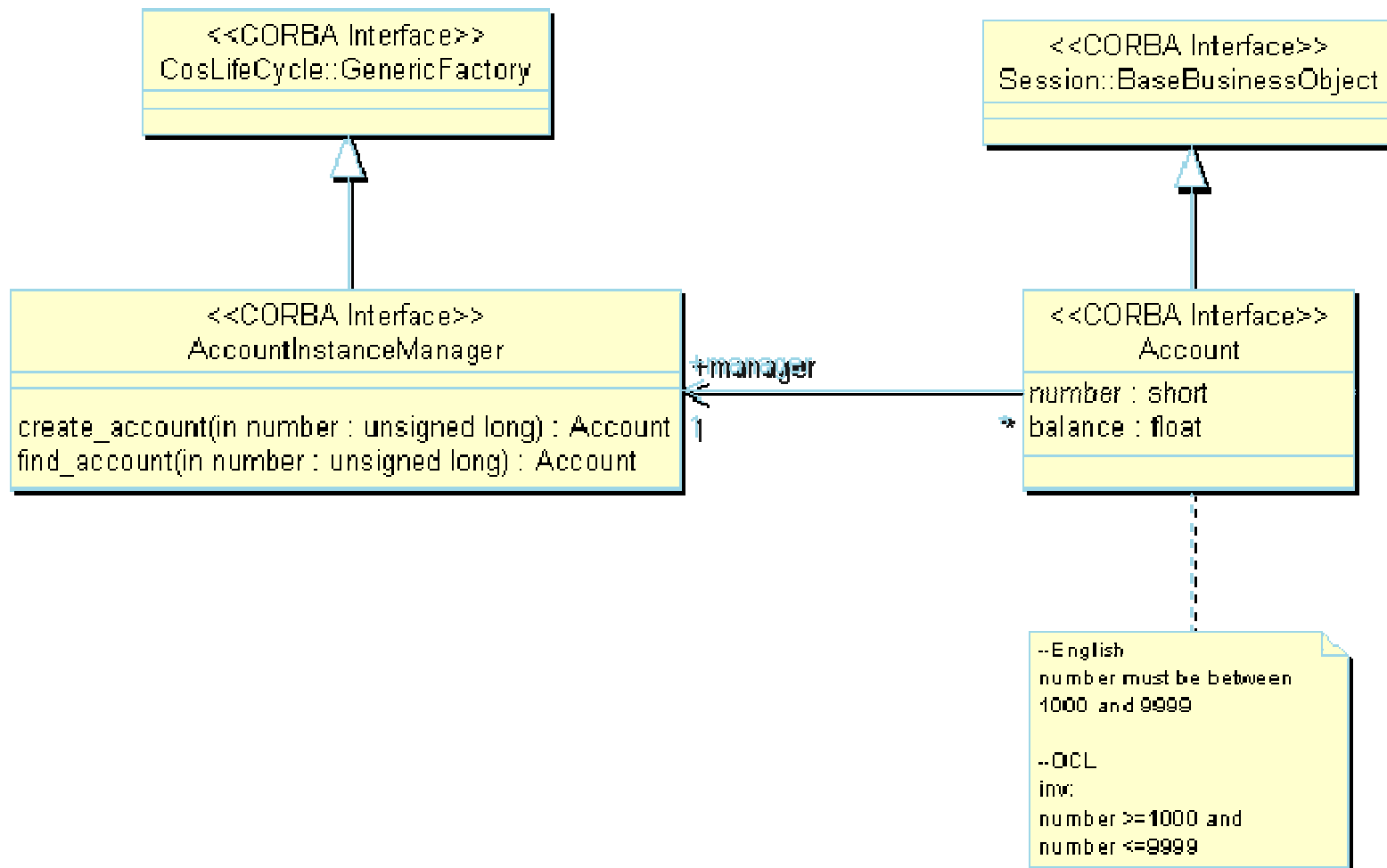
```
--English  
number must be between  
1000 and 9999  
  
--OCL  
Inv:  
number >= 1000  
and  
number <= 9999
```

# Platform Specific Model (PSM)

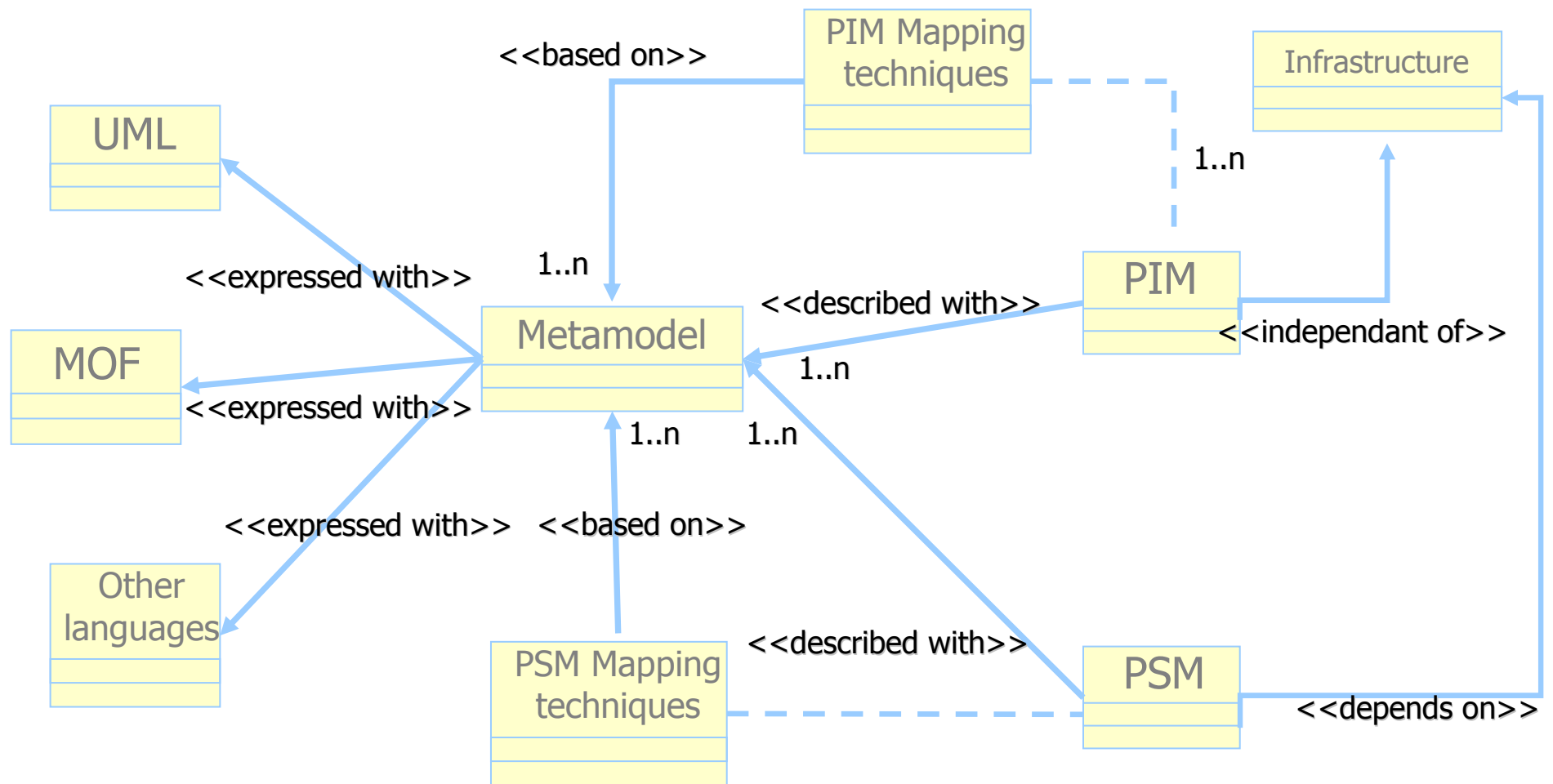


- Especifica como la funcionalidad definida en PIM es realizada en una plataforma particular
- Expresado usando UML extendido con perfiles UML específicos a la plataforma

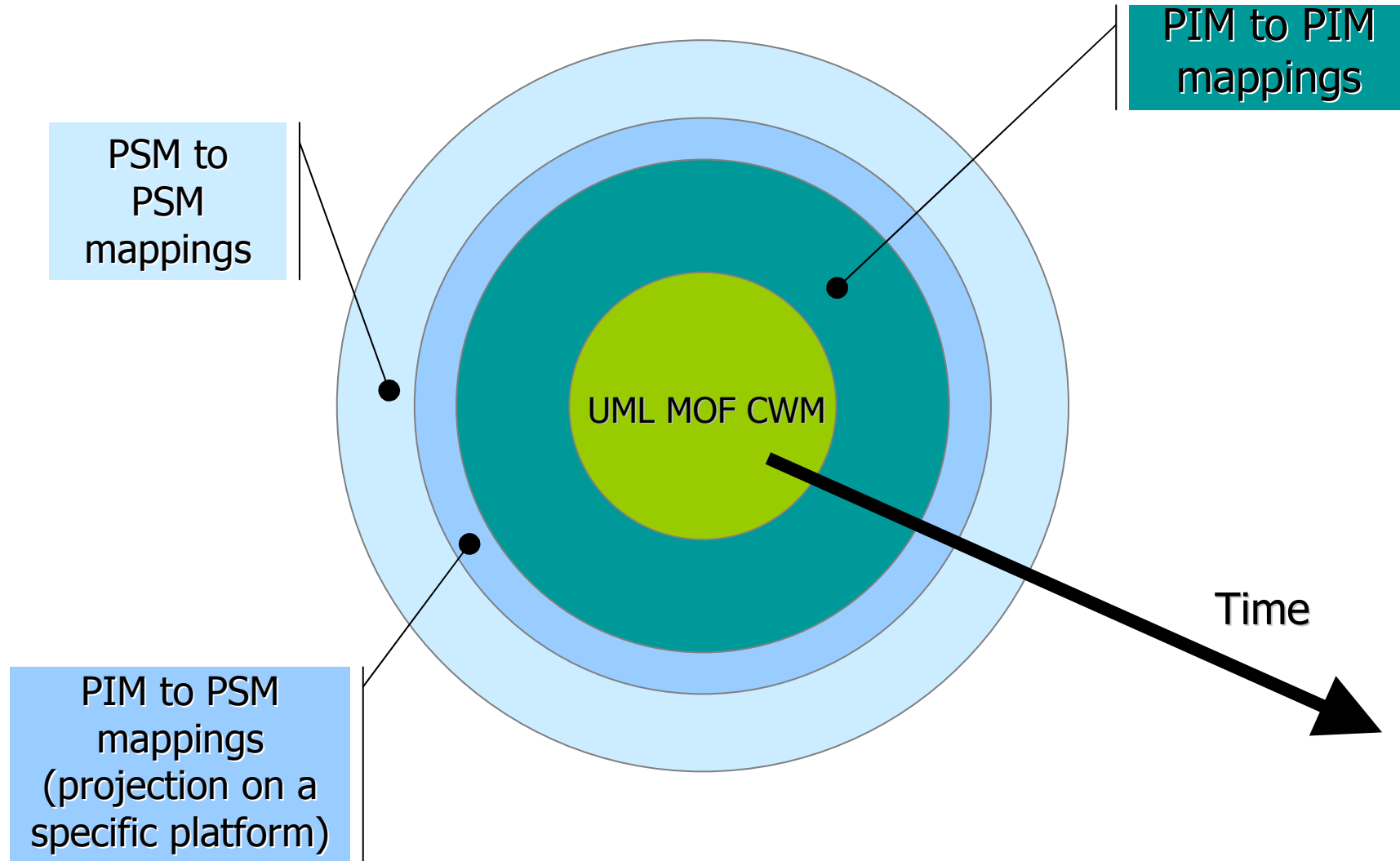
# PSM: Ejemplo



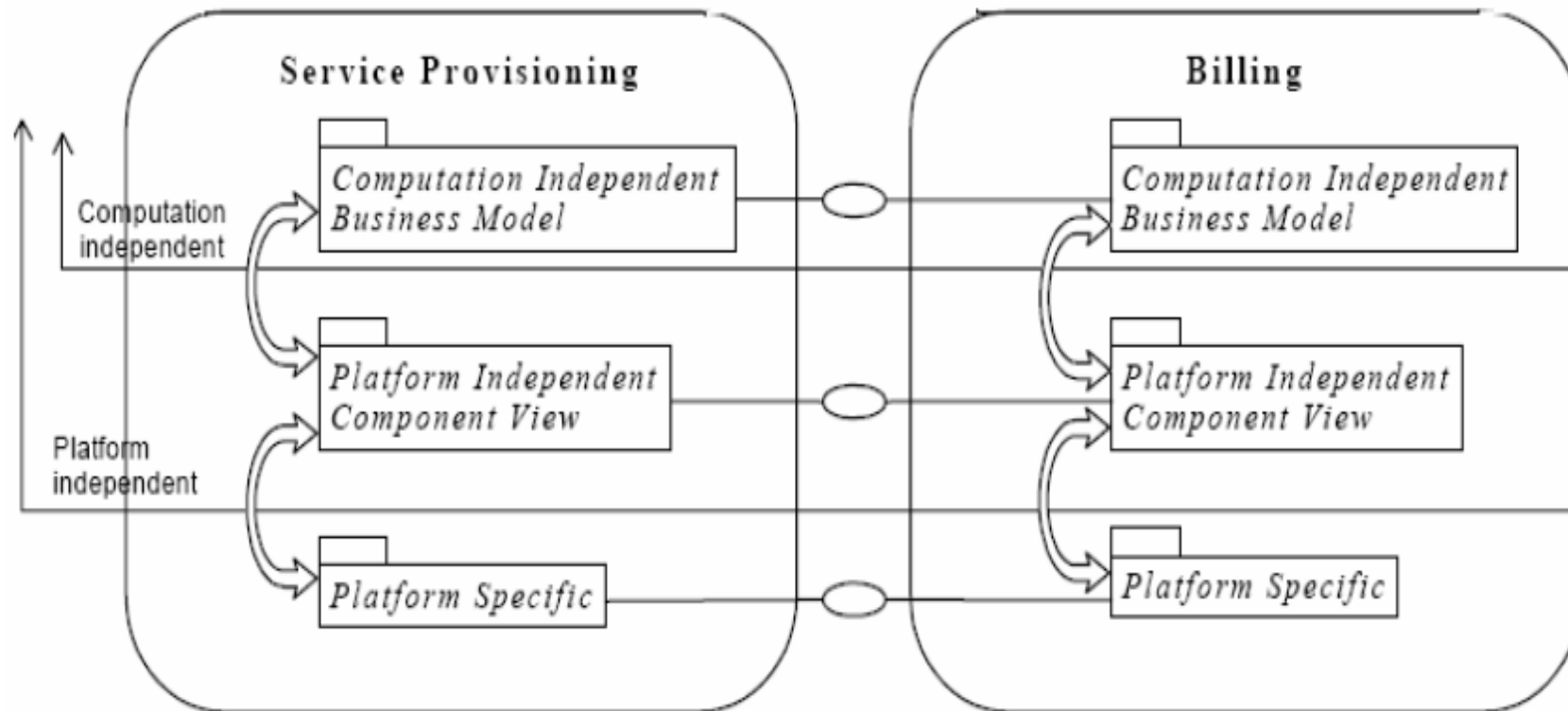
# Metamodelo MDA



# Ciclo de vida en MDA



# Implementación de MDA



# MDA y AS



- La mayoría de modelos en MDA son esencialmente representaciones de una AS.
- Los modelos de dominio y de sistema son diferentes vistas del modelo de AS.
- El código generado posee características de la arquitectura que van con los detalles de implementación.
- El código puede ser usado en ingeniería en reversa para reconstruir la arquitectura de la aplicación.

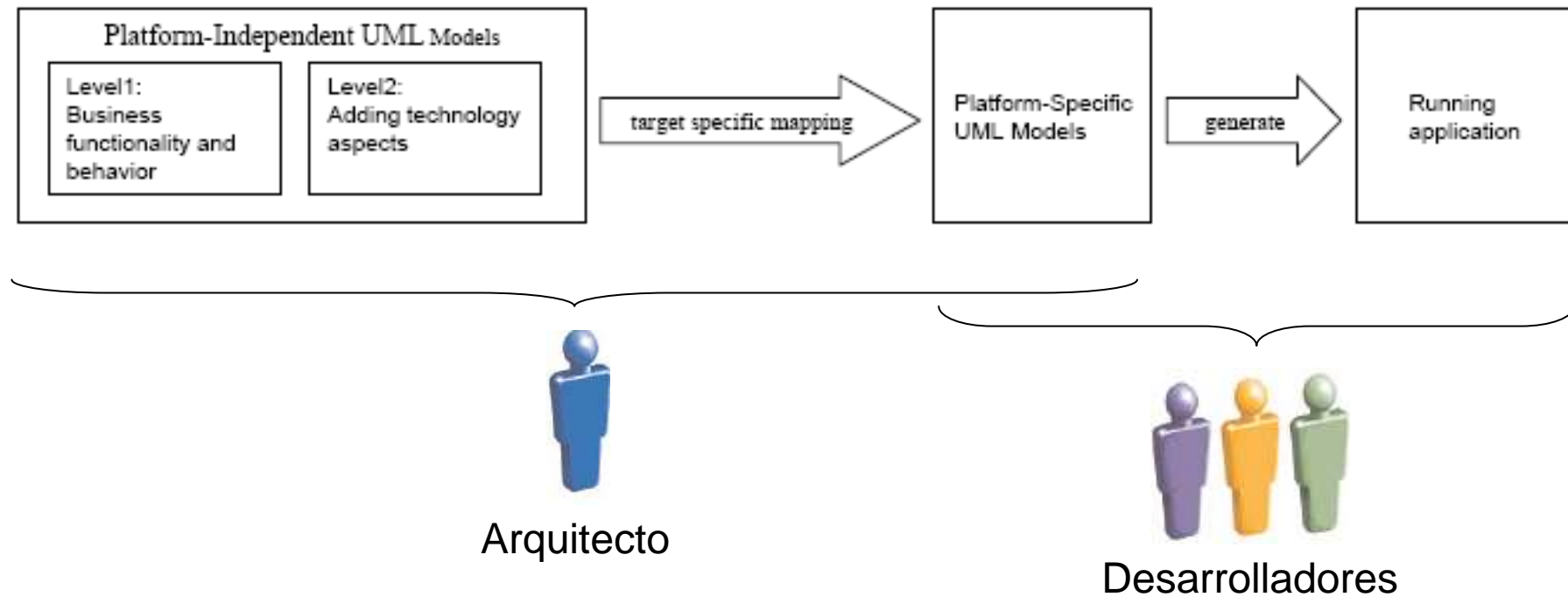
# Descripcion de Modelos en MDA



- Los modelos son descritos usando un ADL.
- UML es usado en MDA como un ADL, aunque ciertos formalismos y características dinámicas de ciertos ADL's no pueden ser expresados de forma completa por UML.
- Sin embargo esta es un área de rápido crecimiento donde las herramientas de modelado se han ido especializando en diferentes modelos de dominio. AndroMDA es un ejemplo de esto.



# Desarrollando con MDA



# Desarrollando con MDA



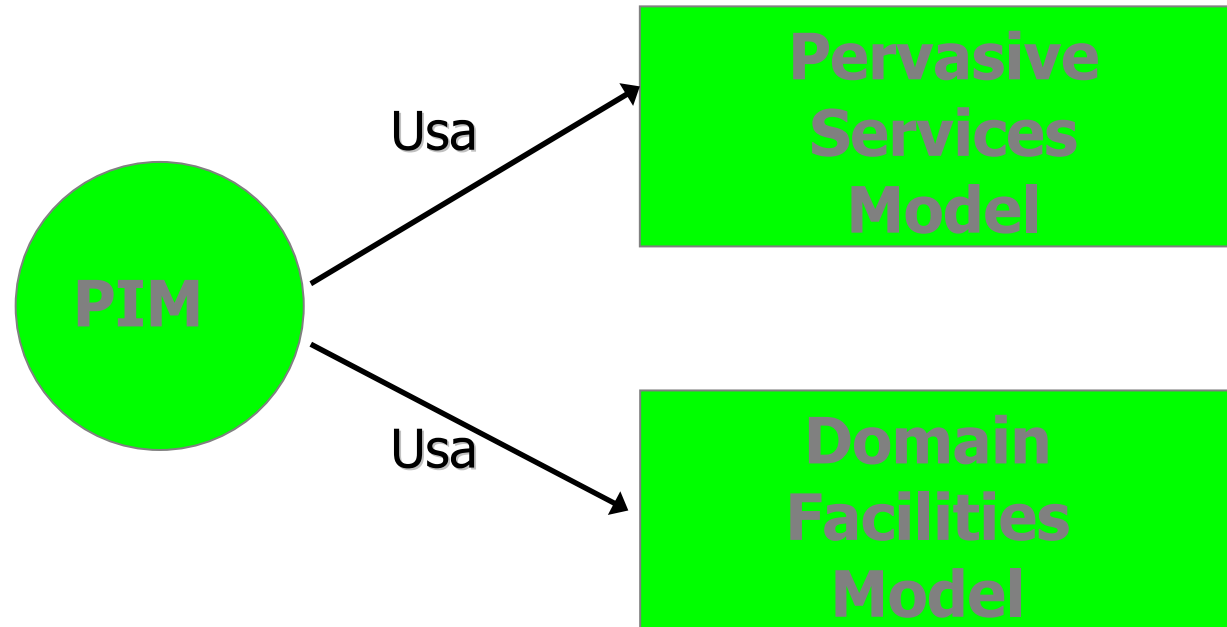
- Acompaña el proceso de desarrollo escogido
- Obtiene retroalimentación de hallazgos
- Guía la revisión de aspectos de calidad del sistema
- Apoya el proceso de documentación técnica del software
- Se recomienda apoyarse en una herramienta para modelado con MDA

# Paso 1: Definir PIM (1/2)



- Es el primer paso en todo proyecto MDA
- Representa la funcionalidad del negocio sin poner atención a detalles de implementación
- Algunas herramientas ofrecen representaciones del dominio y servicios dominantes las cuales facilitan el uso y la incorporación en el modelo

# Paso 1: Definir PIM (2/2)

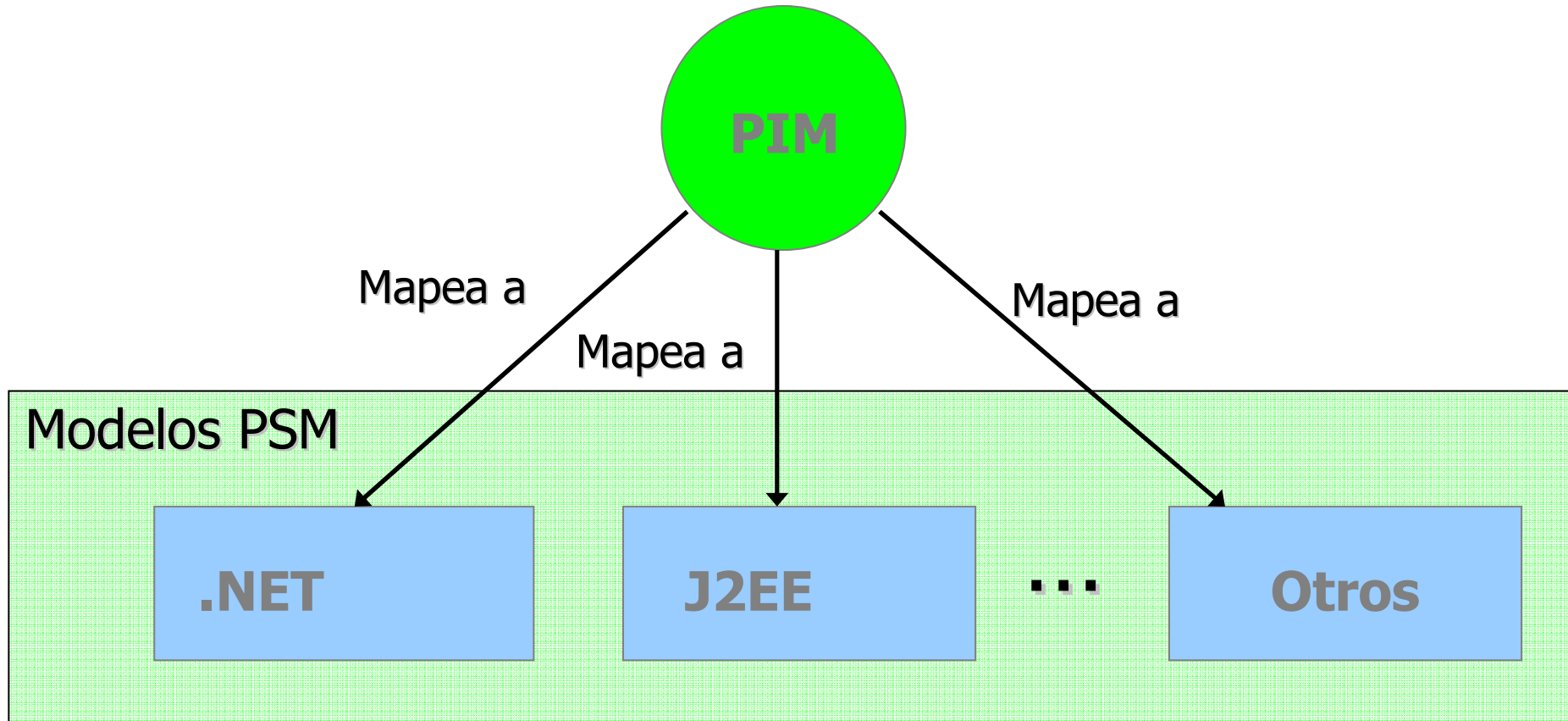


## Paso 2: Definir PSM (1/2)



- Utilizar el PIM como entrada a los mecanismos de mapeo para generar el código que sea automático. Otro será escrito manualmente.
- El PIM podrá ser mapeado a una plataforma única o a múltiples plataformas

# Paso 2: Definir PIM (2/2)

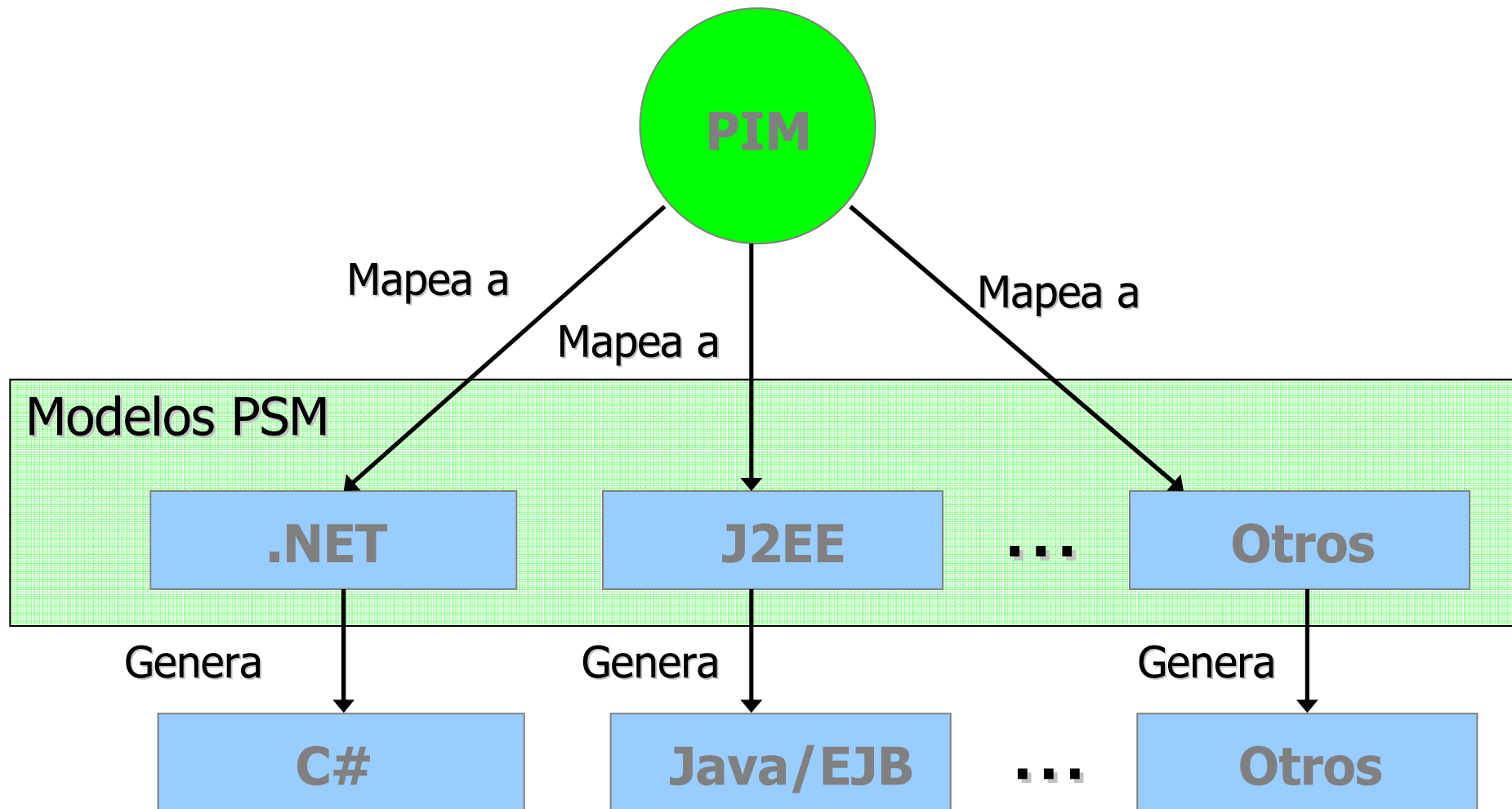


## Paso 3: Generar la Aplicación (1/2)



- Una herramienta MDA genera la mayoría del código de implementación para la tecnología seleccionada por el desarrollador
- Se puede hacer ingeniería en reversa para reutilizar los modelos en otras plataformas

# Paso 3: Generar la Aplicación (2/2)



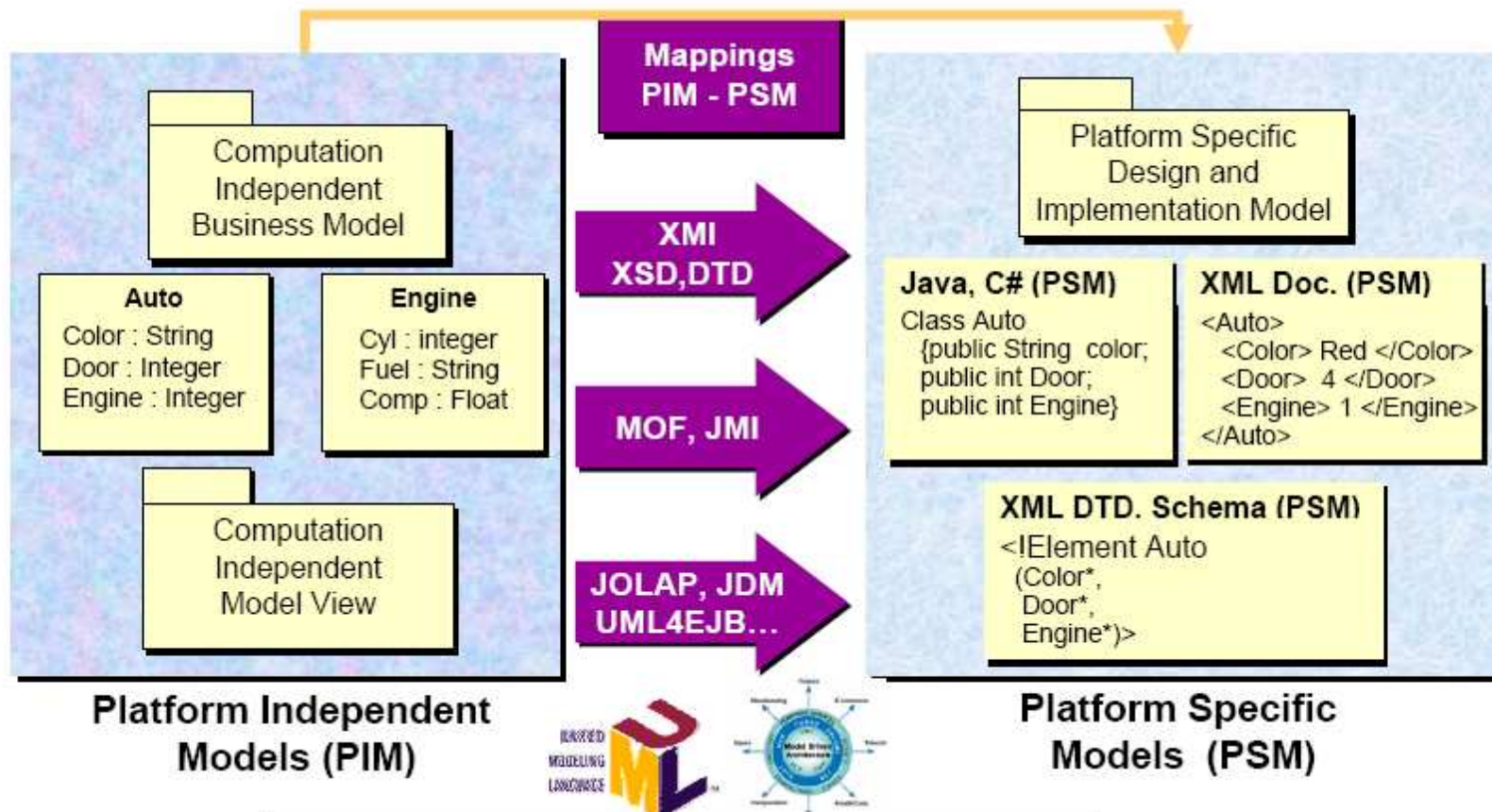


# Mapeo de modelos



- Se define como un conjunto de reglas y técnicas utilizadas para modificar un modelo y así producir otro
- Los mapeos en MDA son descritos utilizando UML

# Mapeo en MDA



# Mapeo PIM a PIM



- Usado cuando los modelos son ampliados o refinados durante el ciclo de vida de desarrollo sin necesidad de información de la plataforma específica.

# Mapeo PIM a PSM



- Es una proyección basada en las características de la plataforma
- Estas características son descritas usando UML
- Se pueden apoyar en perfiles UML para describir conceptos comunes de la plataforma
- Puede ser hecha manualmente (humano) o por un algoritmo (herramienta)

# Créditos



- Essential Software Architecture. Ian Gorton
- MDA, Model Driven Architecture. Lea Kutnoven.
- OMG's Model Driven Architecture. Davide Buscaldi.