Bridging Microsoft Oslo and Eclipse EMF

Seminar Modellgetriebene Softwareentwicklung
Abschlusspräsentation

Stanley Hillner

Microsoft Codename „Oslo“

- **Microsofts neuestes Werkzeug für MDSD**
- **Heißt jetzt „MS SQL Server Modeling Services“**

- **3 wesentliche Bestandteile**
  - Oslo Modeling Language „M“
  - Oslo Repository
  - „Quadrant“

- **Oslo verfolgt typische Ziele des MDSD**
  - Modelle nehmen im gesamten SW-Lebenszyklus zentrale Rolle ein
  - Steigerung der Produktivität
  - Steigerung der SW-Qualität

- **Direkte Anbindung an .NET Plattform**

Microsoft Codename „Oslo“ – „M“

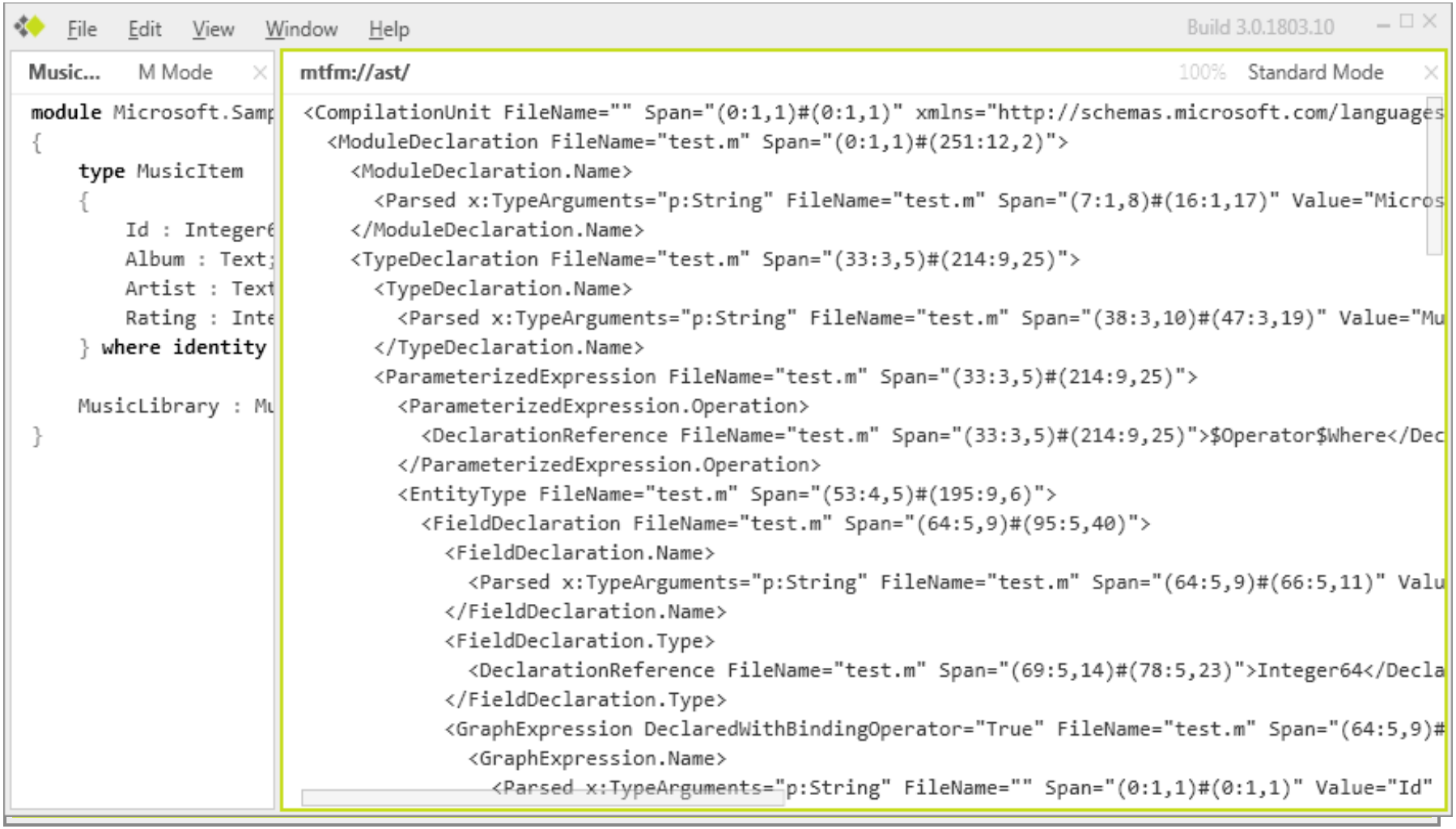
- **Generische Modellierungssprache (3 Bestandteile)**
  - MSchema: Definition von Datenschemata (Metamodelle)
  - MGraph: Definition von Dateninstanzen (Modelle)
  - MGrammar: Definition von Grammatiken für textuelle DSLs

- **Oslo stellt auch Editor für „M“ zur Verfügung („Intellipad“)**
  - Modellierung aller drei Sprachbestandteile
  - Verschiedene Sichten auf die Schemata, Modelle, Grammatiken, usw.

- **Beispiel für Modellierung in „M“ unter Nutzung von „Intellipad“**
  - Musikbibliothek
    - **Alben**
    - **Künstler**
    - **Bewertungen**

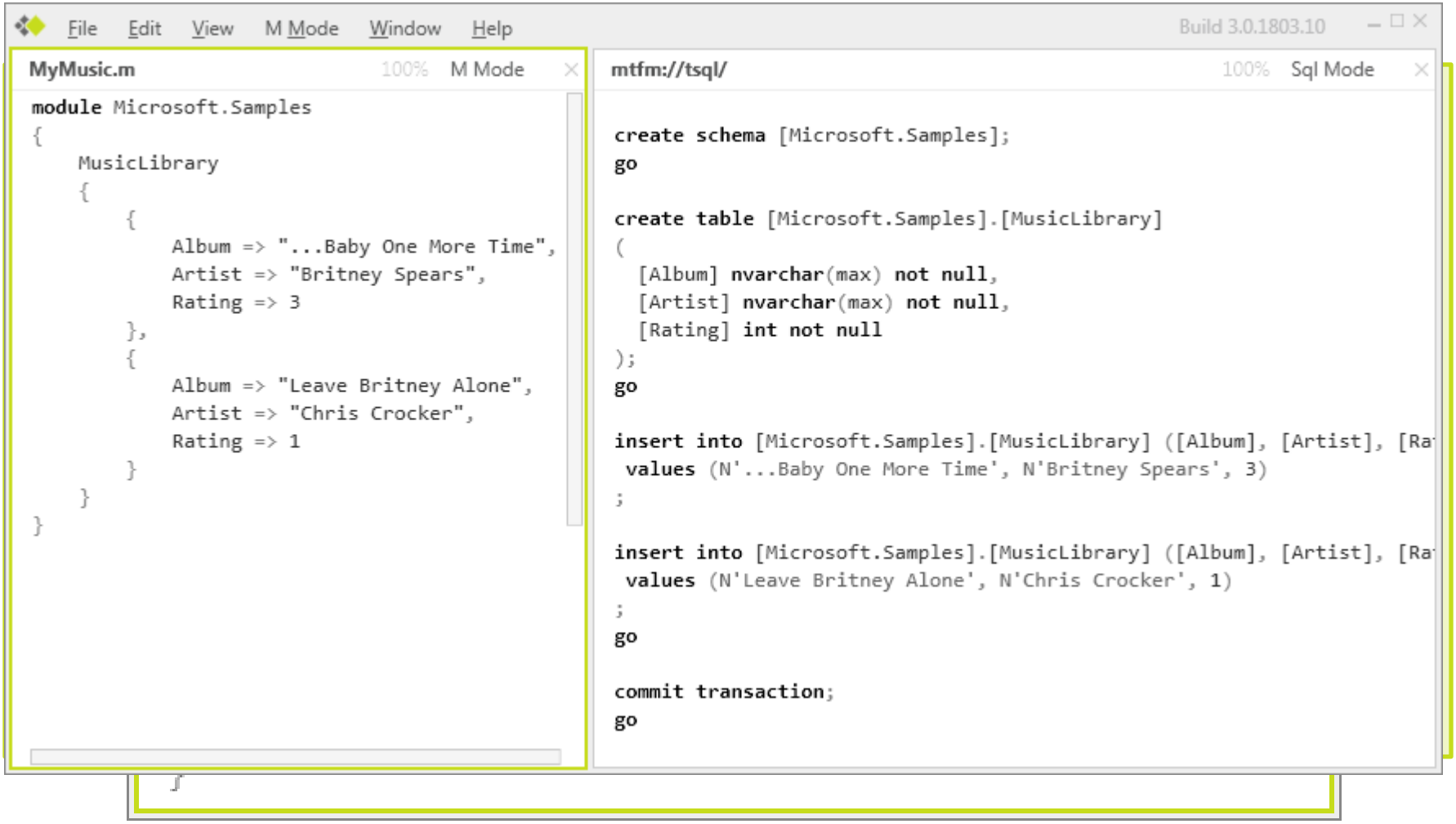
Microsoft Codename „Oslo“ – MSchema

- **Definition des Schemas (Bibliothek, Alben, Interpreten, Ratings)**

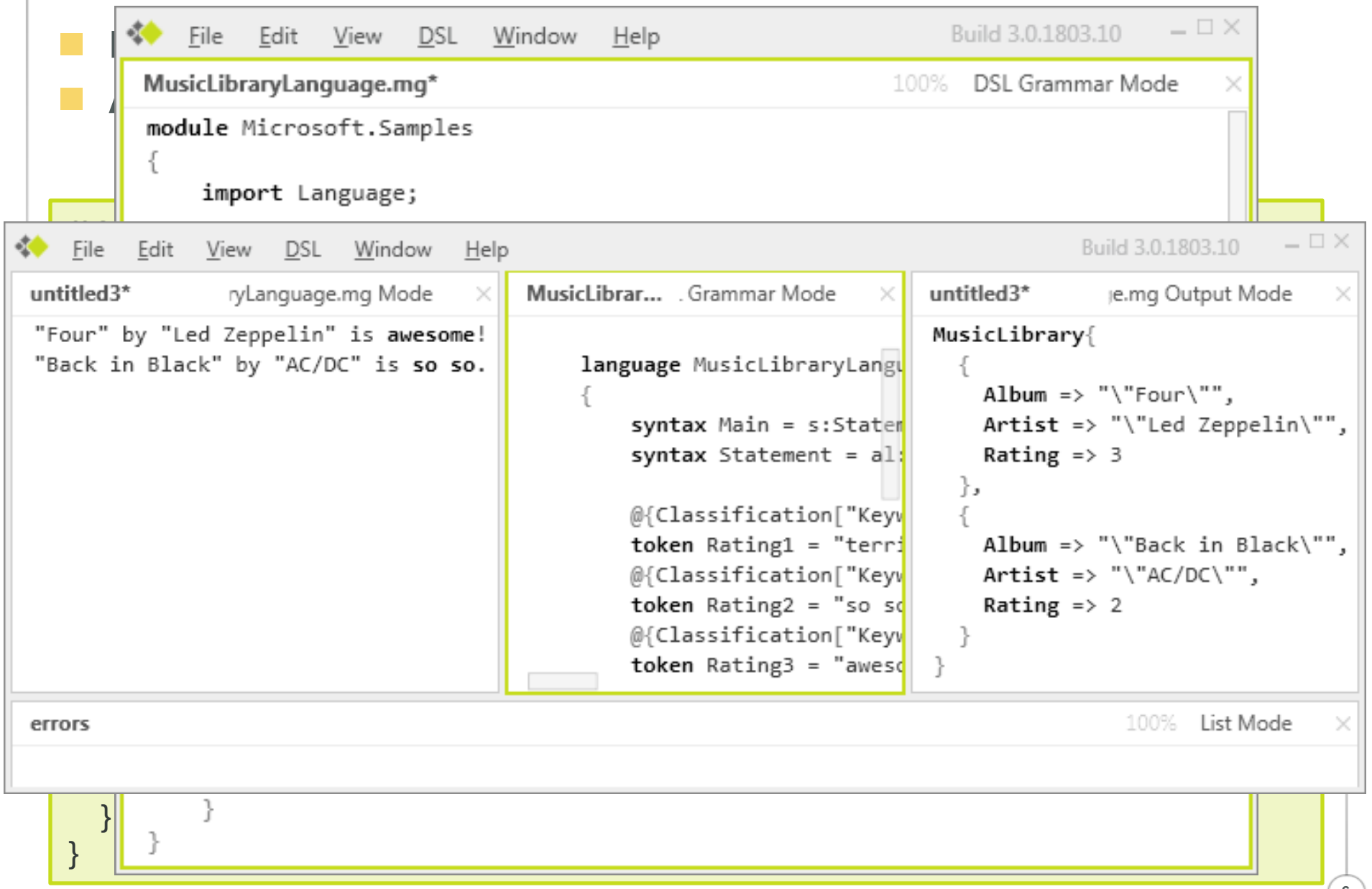


Microsoft Codename „Oslo“ – MGraph

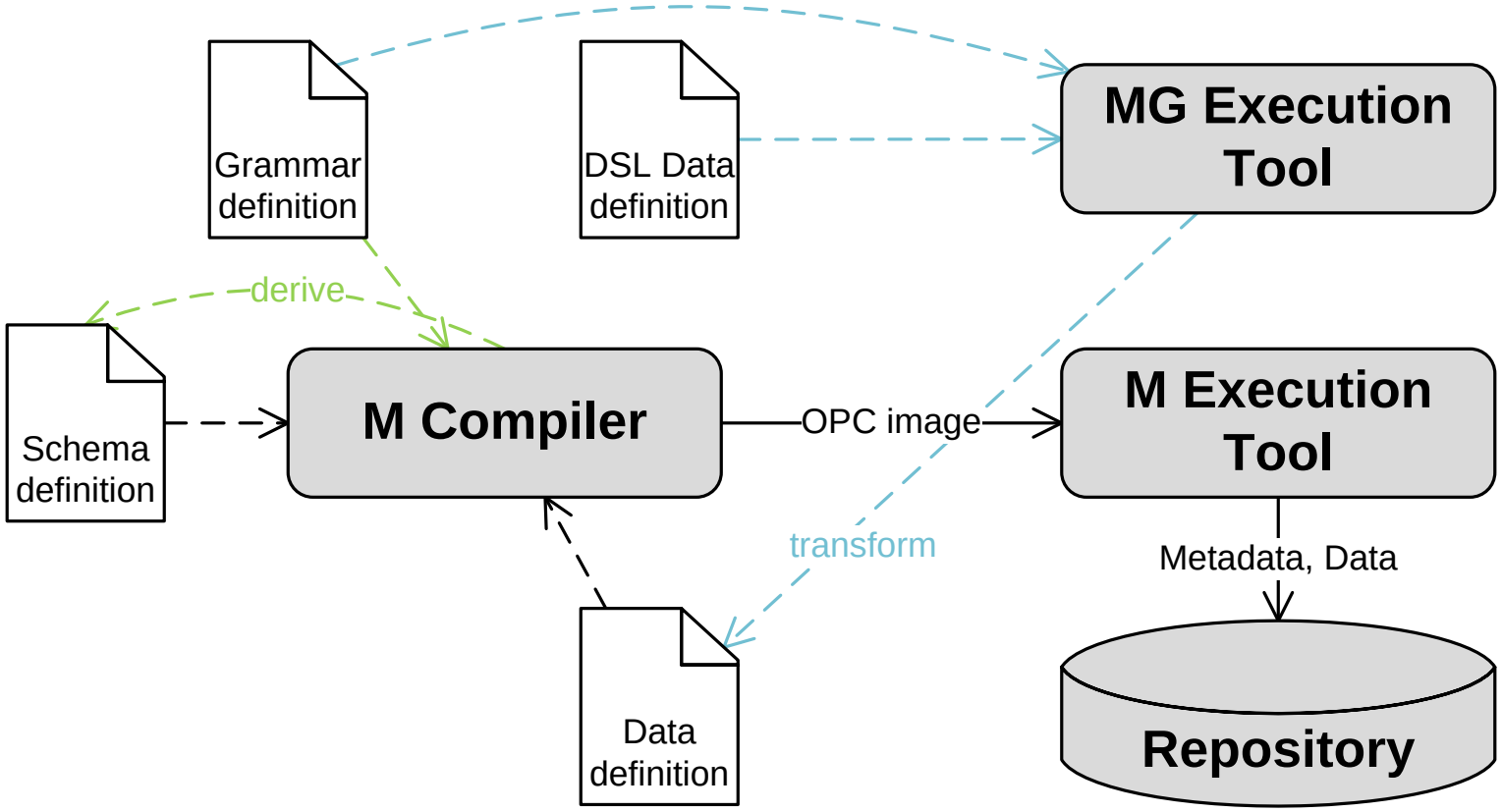
- **Definition von Dateninstanzen (konform zum Schema)**



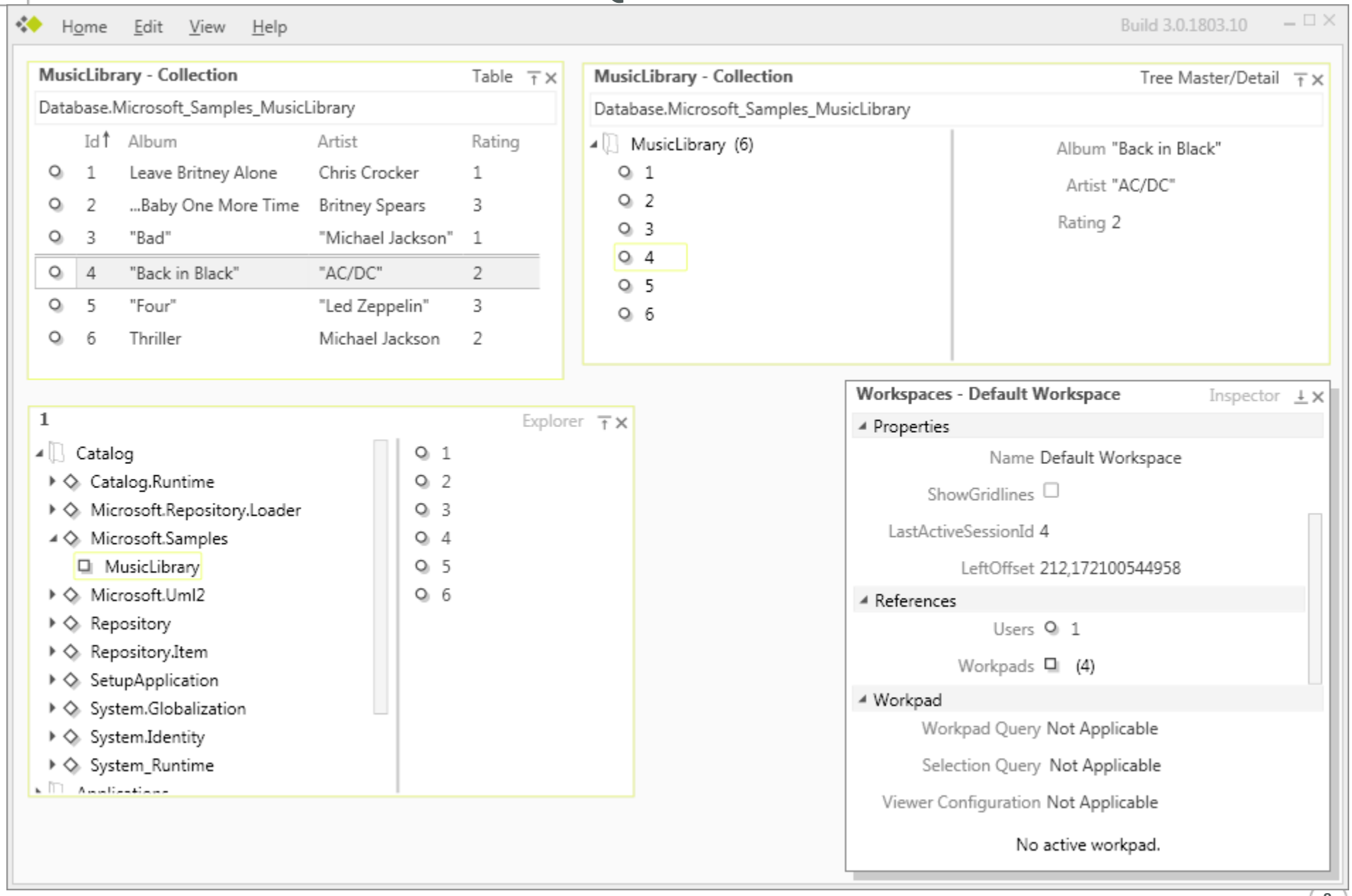
Microsoft Codename „Oslo“ – MGrammar



Microsoft Codename „Oslo“ – Modellierungsprozess



Microsoft Codename „Oslo“ – Quadrant



Eclipse Modeling Framework

- **Open-Source Java-Framework zur Erstellung von Anwendungen**
- **Arbeitet mit strukturierten Modellen**

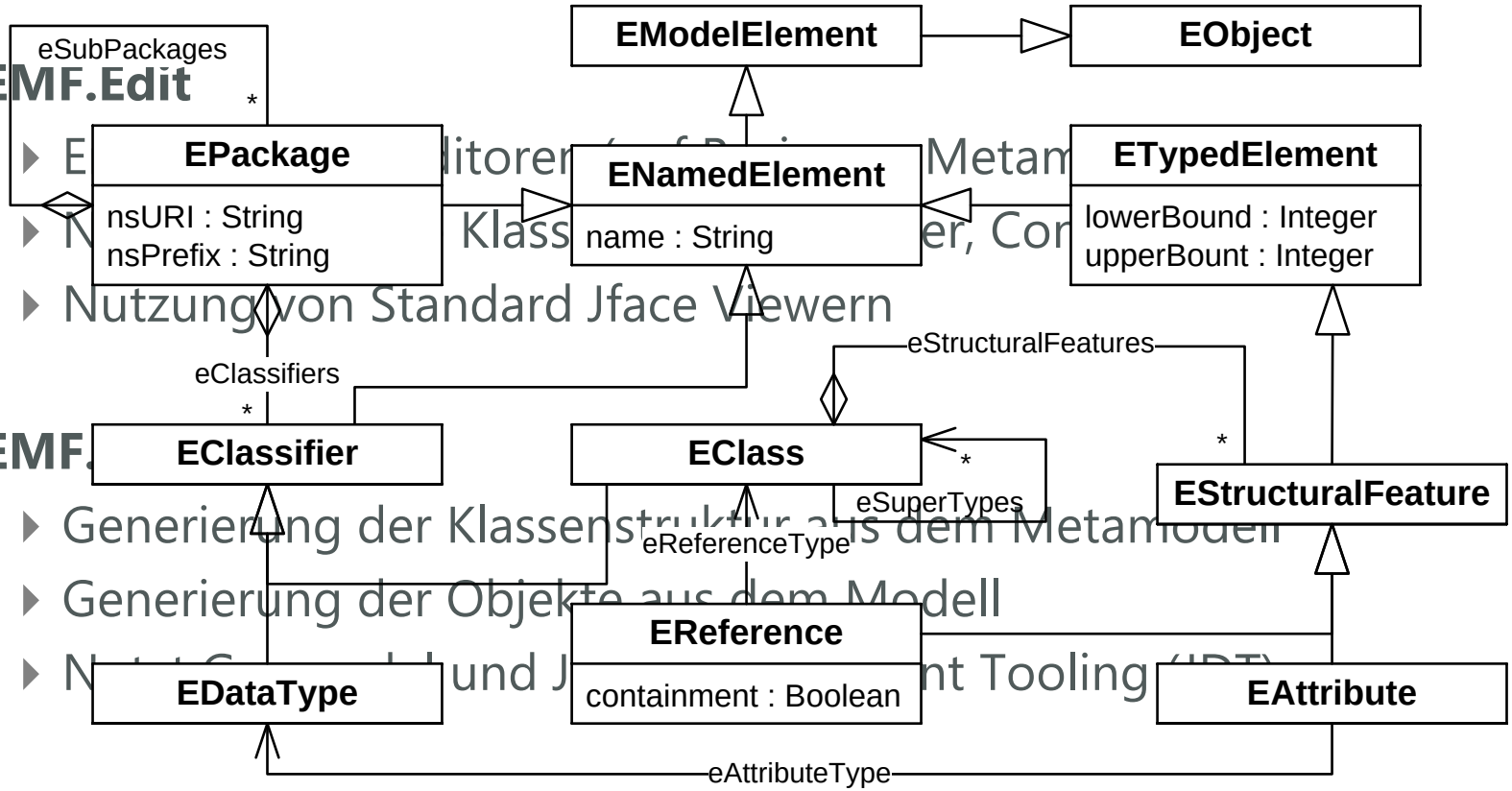
- **Wesentliche Bestandteile**
  - EMF Core → Ecore als Metametamodell (MOF-Implementierung)
  - EMF.Edit → Erstellung von Editoren für Metamodelle
  - EMF.Codegen → Generierung von Code für (Meta)Modelle

- **EMF ist leicht und schnell zu erlernen**
  - Trotzdem sehr mächtig

- **Große Akzeptanz und Verbreitung**
  - Ist Open-Source
  - Vielzahl an Werkzeugen vorhanden, für Aufgaben des MDSD

Eclipse Modeling Framework – Komponenten

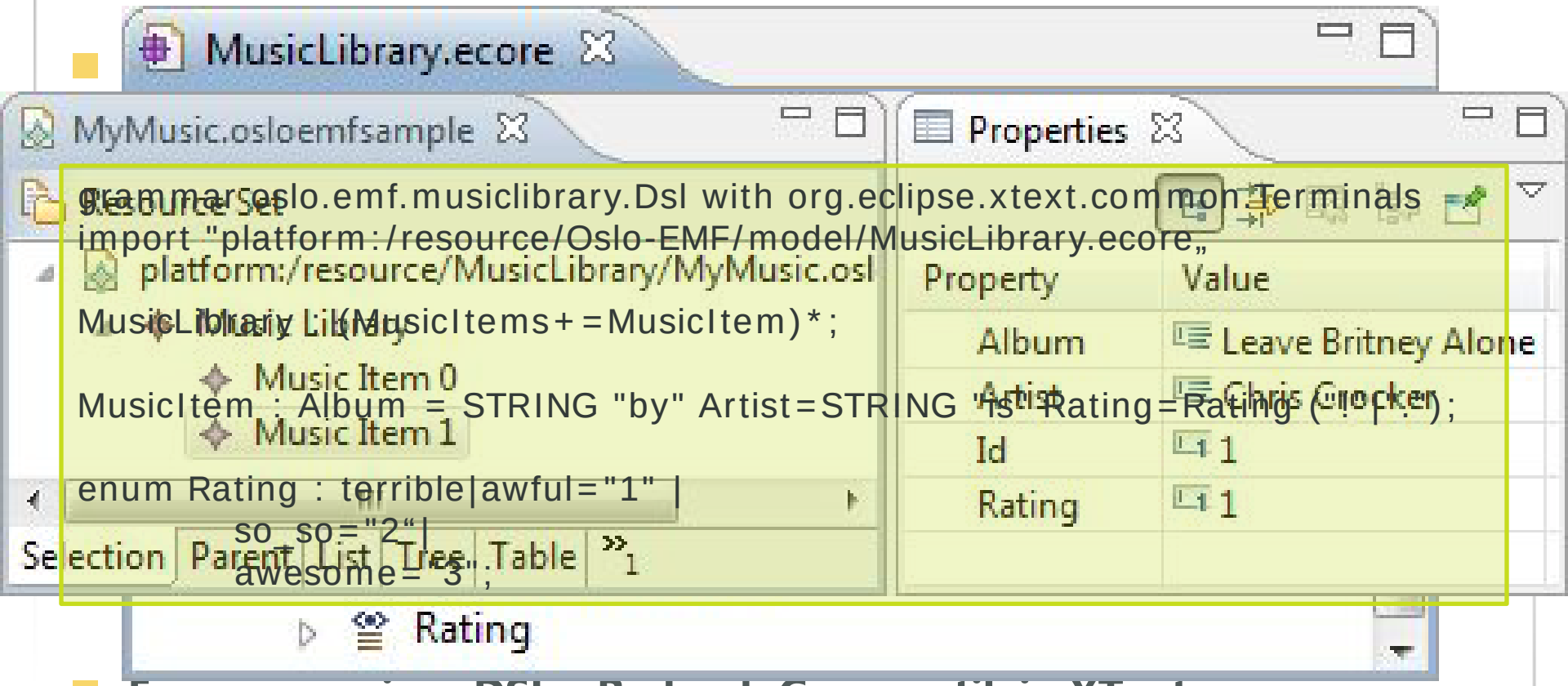
- **EMF Core**
  - Ecore als Metametamodell
  - Implementiert MOF-Spezifikation der OMG
  - Aufbau siehe Diagramm



- **EMF.Edit**
  - E ditoren (auf Basis von Metam
  - N Klass er, Cor
  - Nutzung von Standard Jface Viewern
- **EMF.**
  - Generierung der Klassenstruktur aus dem Metamodell
  - Generierung der Objekte aus dem Modell
  - N und J nt Tooling (JDT)

Eclipse Modeling Framework – (Meta)Modellierung

- **Erstellung eines Metamodells (konform zu Ecore)**
  - Nutzung des Ecore model editor, graphisch, aus XSD, Rose-Modell
- **Erzeugung einer DSL z.B. durch Grammatik in XText**



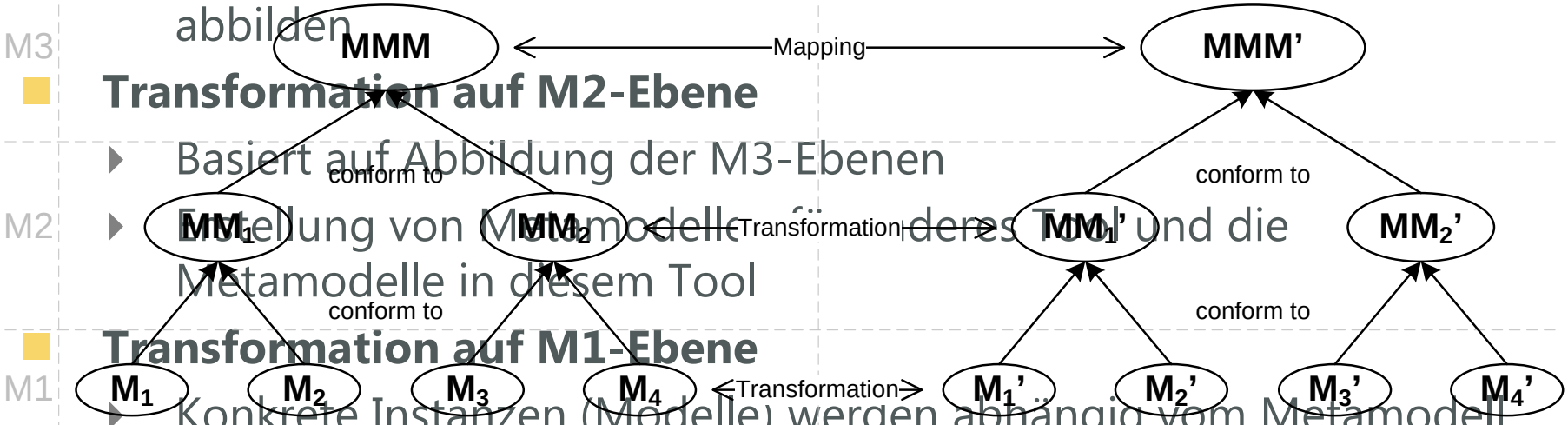
Mögliche Vorteile einer Oslo-EMF Brücke

- **EMF ist bereits Hub für Bridging von MDSD Tools**
  - Bereits viele Brücken vorhanden (Visio, Aris, MetaEdit, DSL Tools, GME, usw.)
  - Verwendung existierender Brücken zur Verringerung des Aufwandes (transitive Brücken)
  - Dadurch höhere Effizienz und Qualität
- **Bereitstellung von Funktionalität**
  - EMF ist Open-Source → große Community → Viele Tools
  - Passendes Werkzeug für (fast) jede Aufgabe
  - Oslo hat auch Vorzüge (Repository, Quadrant)
- **Wechsel ermöglichen**
  - EMF→Oslo
  - Oslo→EMF

Das M3-level based Bridging pattern

- **Setzt Vorhandensein einer Metamodell-Hierarchie voraus**

- **Mapping auf M3-Ebene**
  - Konstrukte des einen auf Konstrukte des anderen Werkzeugraums abbilden

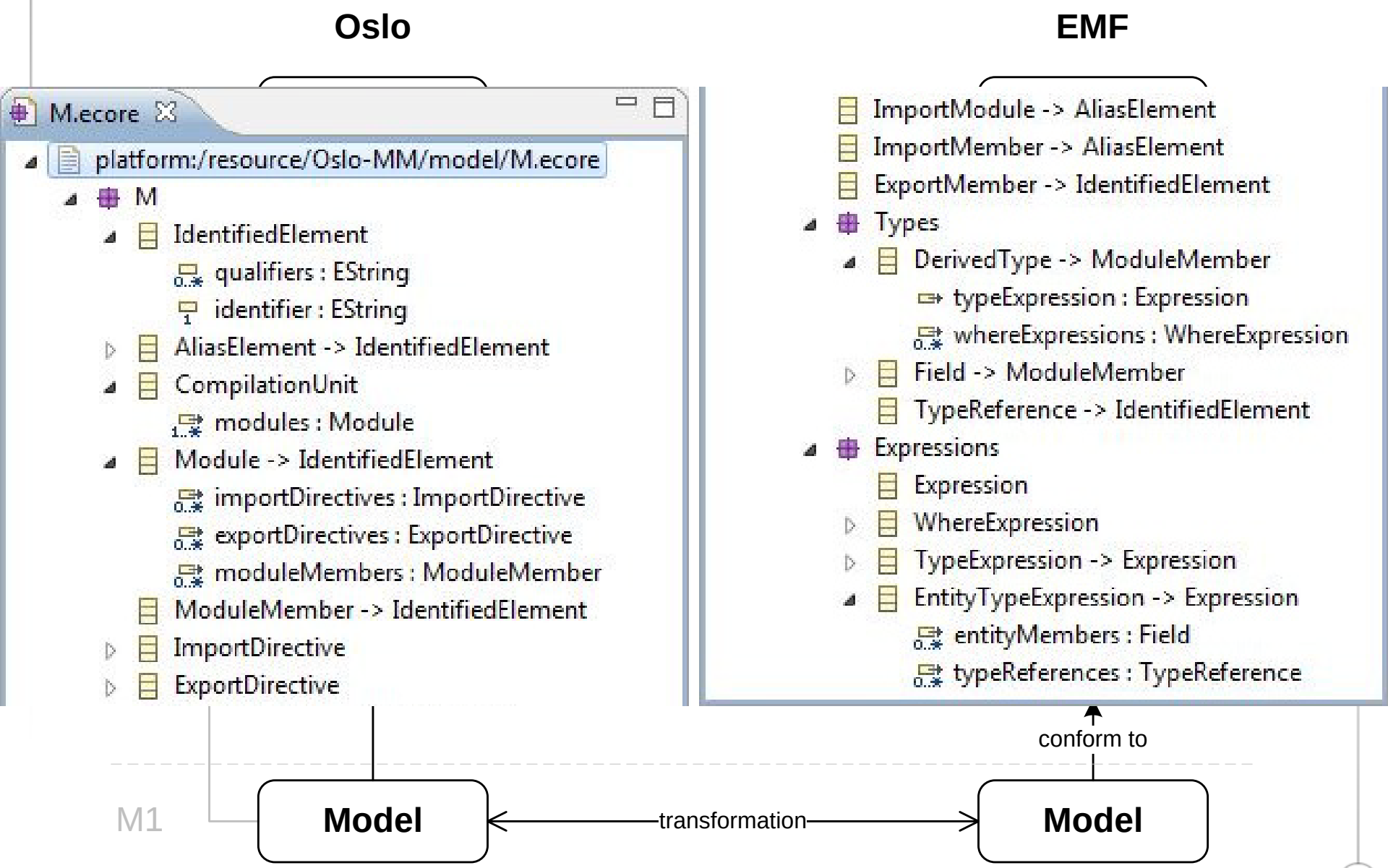


- **Transformation auf M2-Ebene**
  - Basiert auf Abbildung der M3-Ebenen
  - Erstellung von Metamodellen für dieses Tool und die Metamodelle in diesem Tool

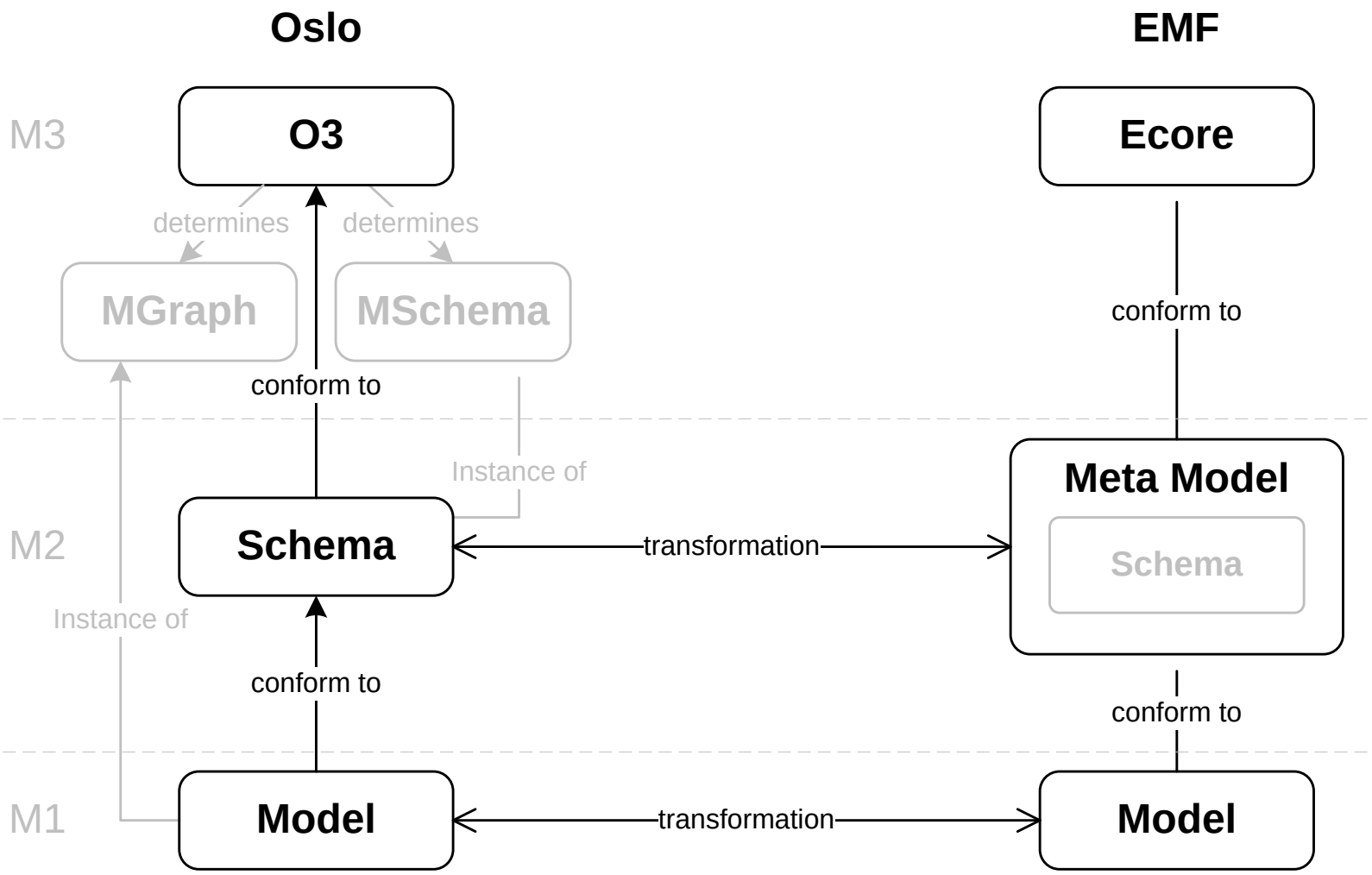
- **Transformation auf M1-Ebene**
  - Konkrete Instanzen (Modelle) werden abhängig vom Metamodell transformiert

- **Problem:**
  - Metamodelle bzw. Modelle in generischem Format

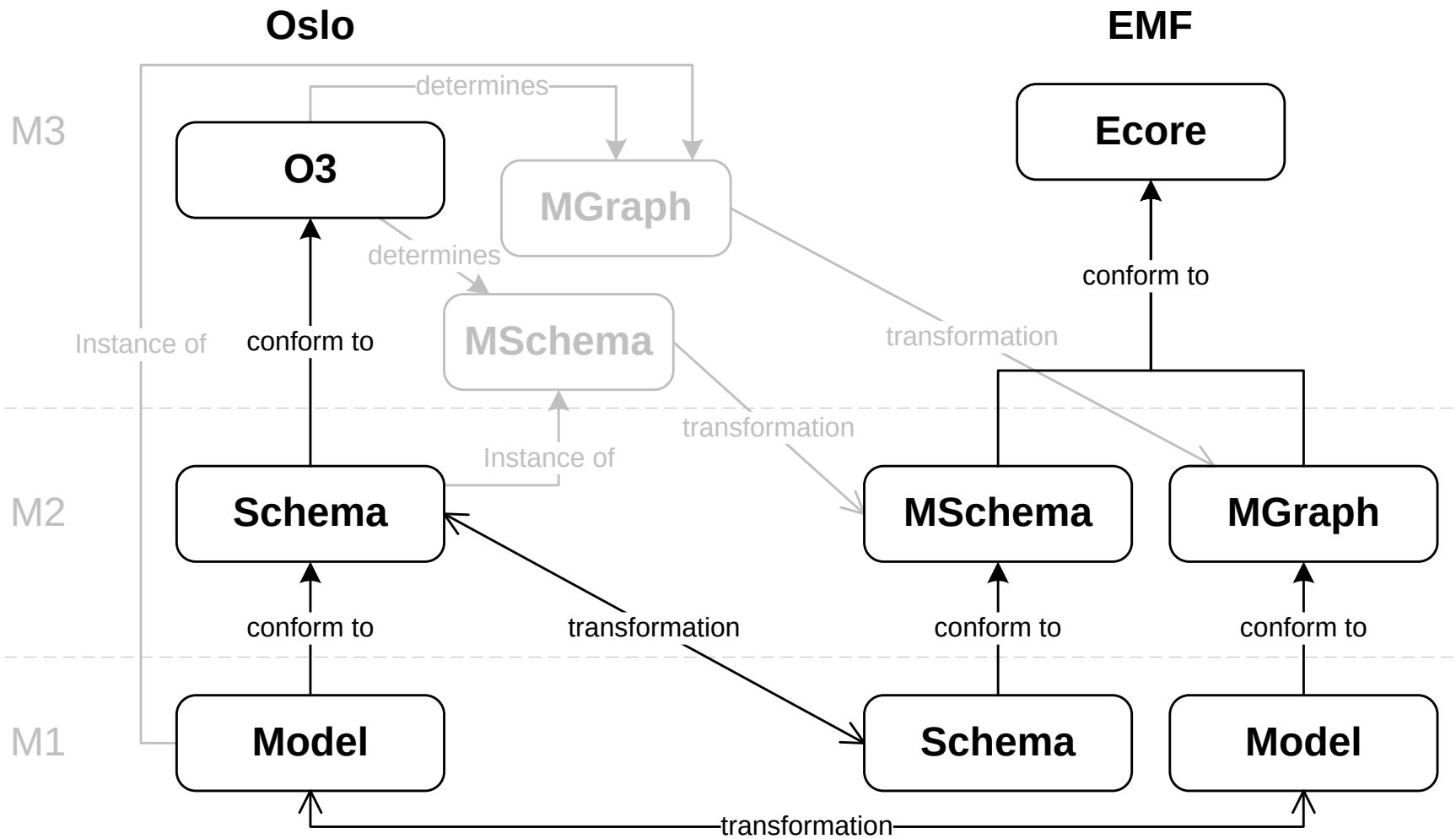
Bridging-Ansatz (M3B)



Bridging-Ansatz (M2B)



Bridging-Ansatz (Vertikale Transformationen)



Fragen?