

## JAX 2013 – Modeling Day

---

### Service Repository for Model-Driven SOA

Design and Implementation Aspects

Thomas Stahl, b+m Informatik AG / engineering dept.

Stefan Zeug, b+m Informatik AG / engineering dept.



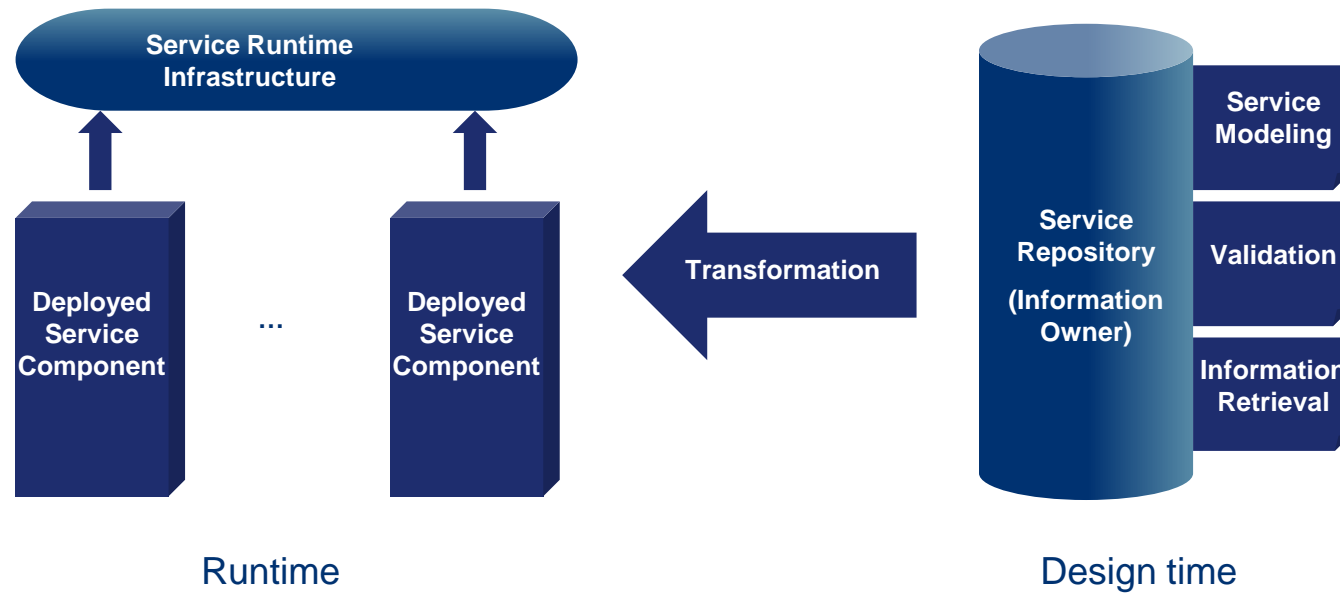
# Agenda

---

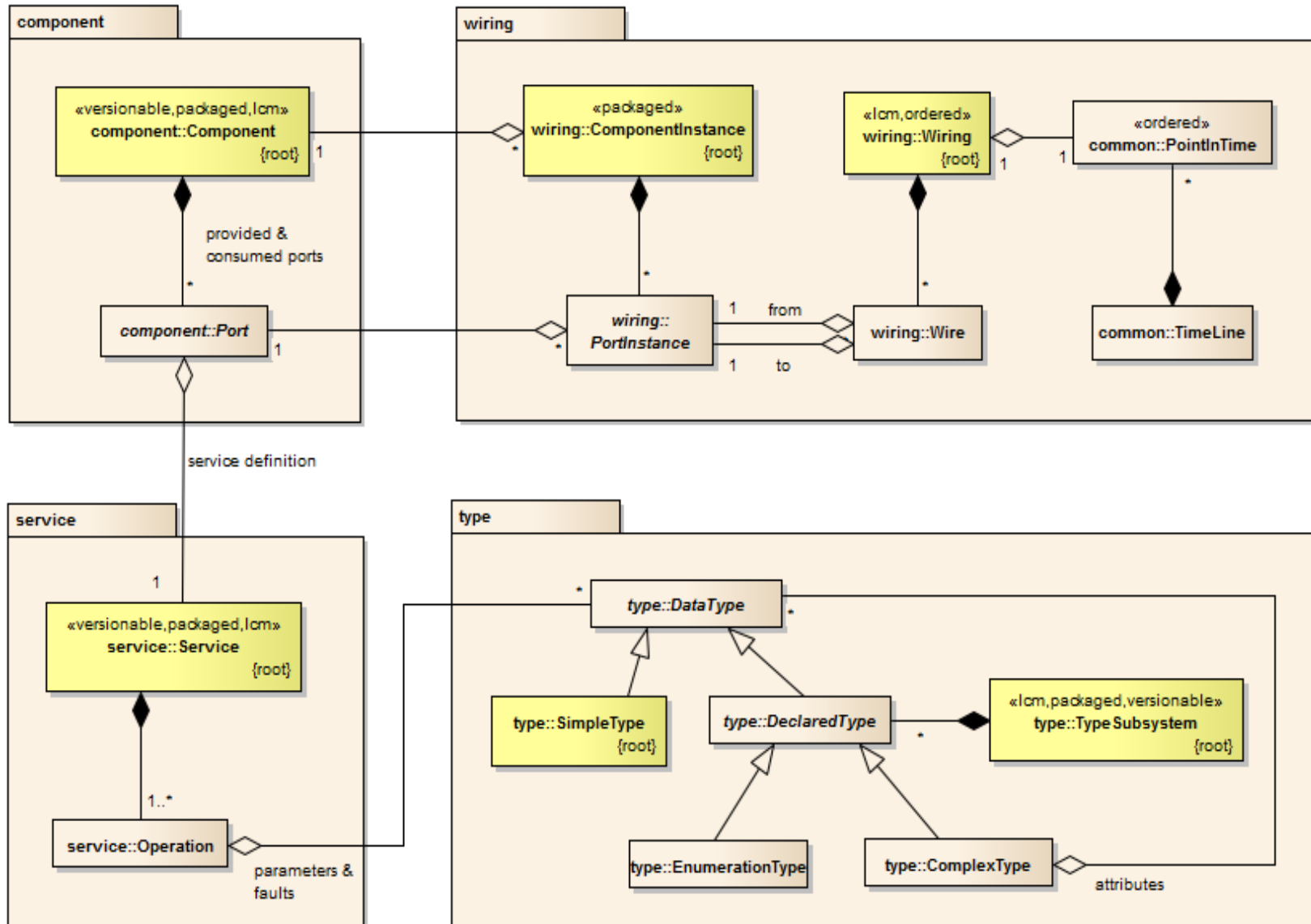
- **Context**
- **Architectural Decisions and Design Principles**
- **Specific Aspects**
  - Discussion of Design- and Implementation-Perspective
- **Prospect**

# Context

- SOA is an almost comprehended domain in the sense of DSL-Engineering and Model-Driven Software Development
- The tools of the Eclipse Modeling Project provide a good basis for the technical infrastructure of a SOA – especially a Service Repository
- This talk presents a (prototypically validated) target vision which was developed in the project MAIA of the Swiss Mobiliar (see last track)



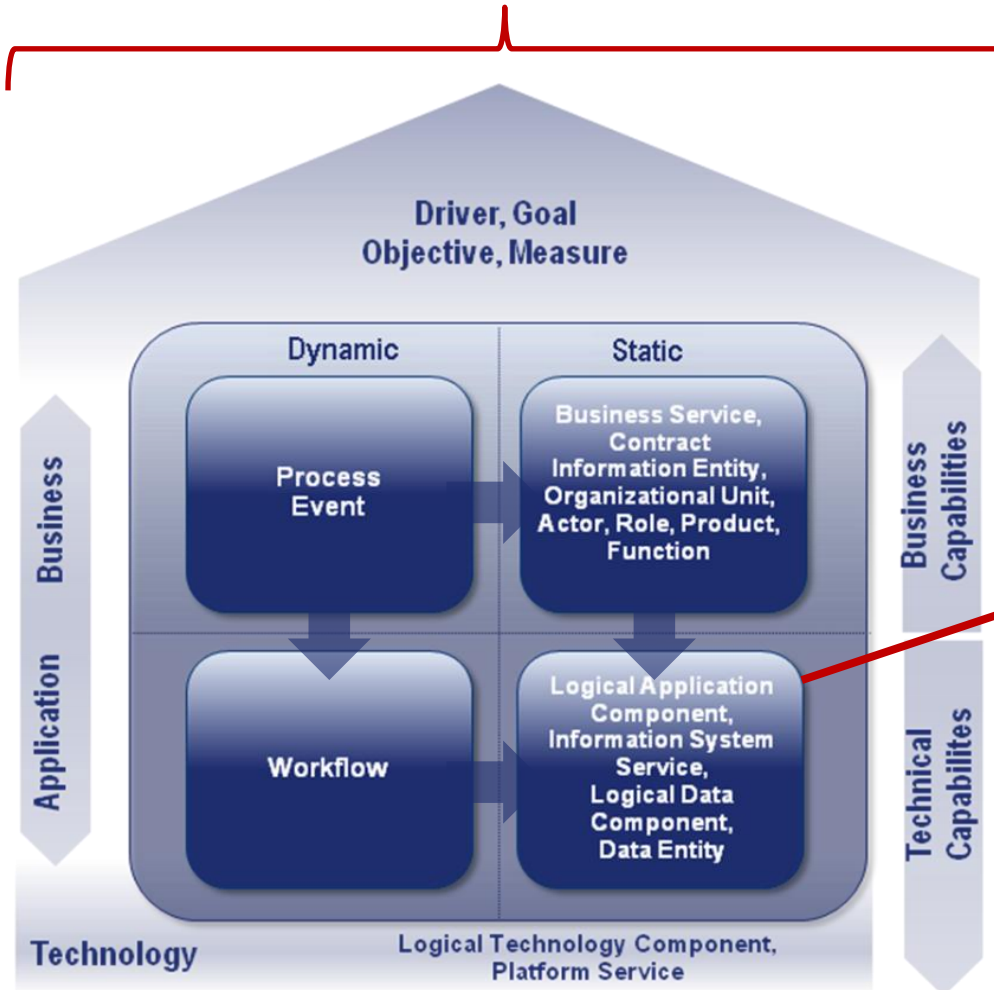
# Recap: Status Quo Scope of Technical SOA Domain Model from the MAIA Project



# Scope / Target Vision

Wide scope (Vision):

Enterprise-Repository - compatible with TOGAF content metamodel



**Narrow scope:  
Service-Repository -  
managing technical  
services and schemas to  
enable SOA micro  
governance**

# Some Usage Scenarios

---

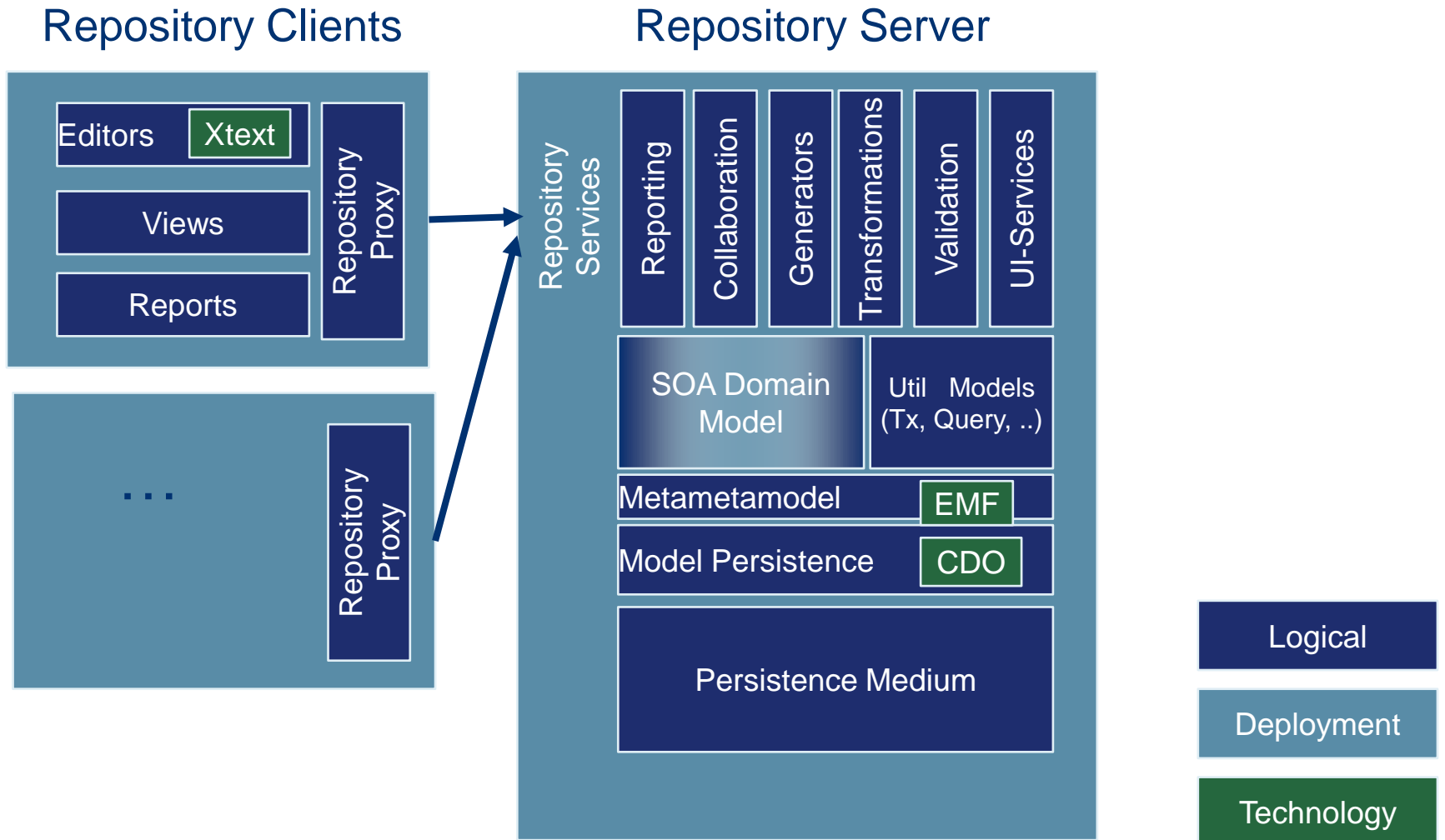
- **(Service-) Reuse**  
Contract publishing, searching and finding, reporting, navigation to associated elements and artifacts relevant for software development (like generated service stubs)
- **(Service-) Evolution**  
Life cycle management, technical versioning (snapshots), quality gates (constraint sets)
- **Dependency Management**  
Impact analysis
- **SOA Governance rules and processes**  
Design time policies, model-diffs, workflow support for life cycle management
- **Deployment Automation (Forward Engineering)**  
Consistency between deployed (service-)versions  
Consistency between (service-)model and deployed artifacts
- **Support for Business IT Alignment**  
Manage dependencies between technical services, business services, business processes and technical workflows

# Architectural Decisions and Design Principles

- **MDD is an established paradigm in the SOA context**
- **Start Small: Evolution of the Service Domain Model must be supported – it's a model on its own**
- **We see a SOA-Repository as a special case of a generic Model-Repository. The typical usage scenarios of a SOA-Repository can thus be mapped to**
  - Collaborative editing of (service) models
  - Persistence of (service) models
  - (Service) model diffs & validation
  - (Service) model transformations and integration of generators
  - Navigation, views, queries, reports for (service) models
- **We build upon the technologies of the Eclipse Modeling Project**
  - EMF/Ecore as metameta model
  - CDO as persistence layer
  - Xtext as DSL infrastructure



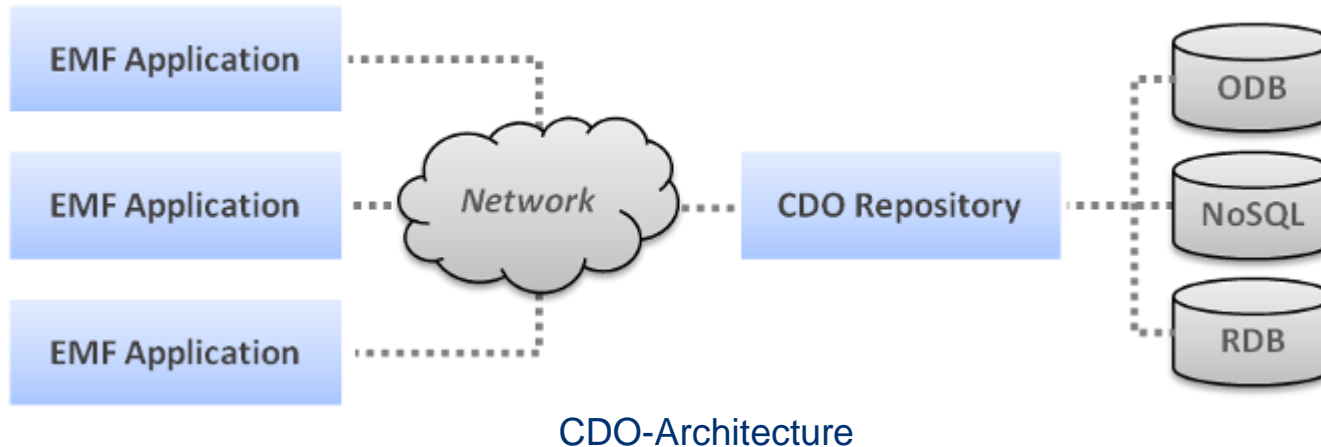
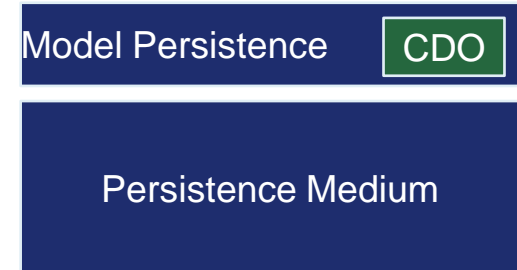
# Architectural Blueprint





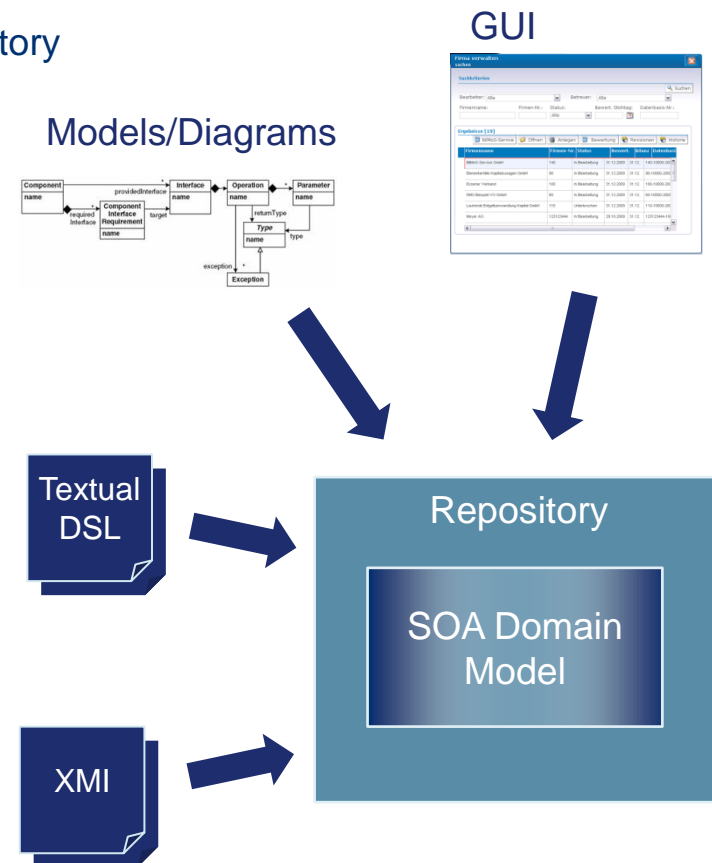
# Model Persistence

- Data ownership is located by the repository (master)
- Separation of DSL-Metamodel and Domain-Model
- Replaceable persistence layer
- CDO integration, database model, CDO session
- Prototyping results



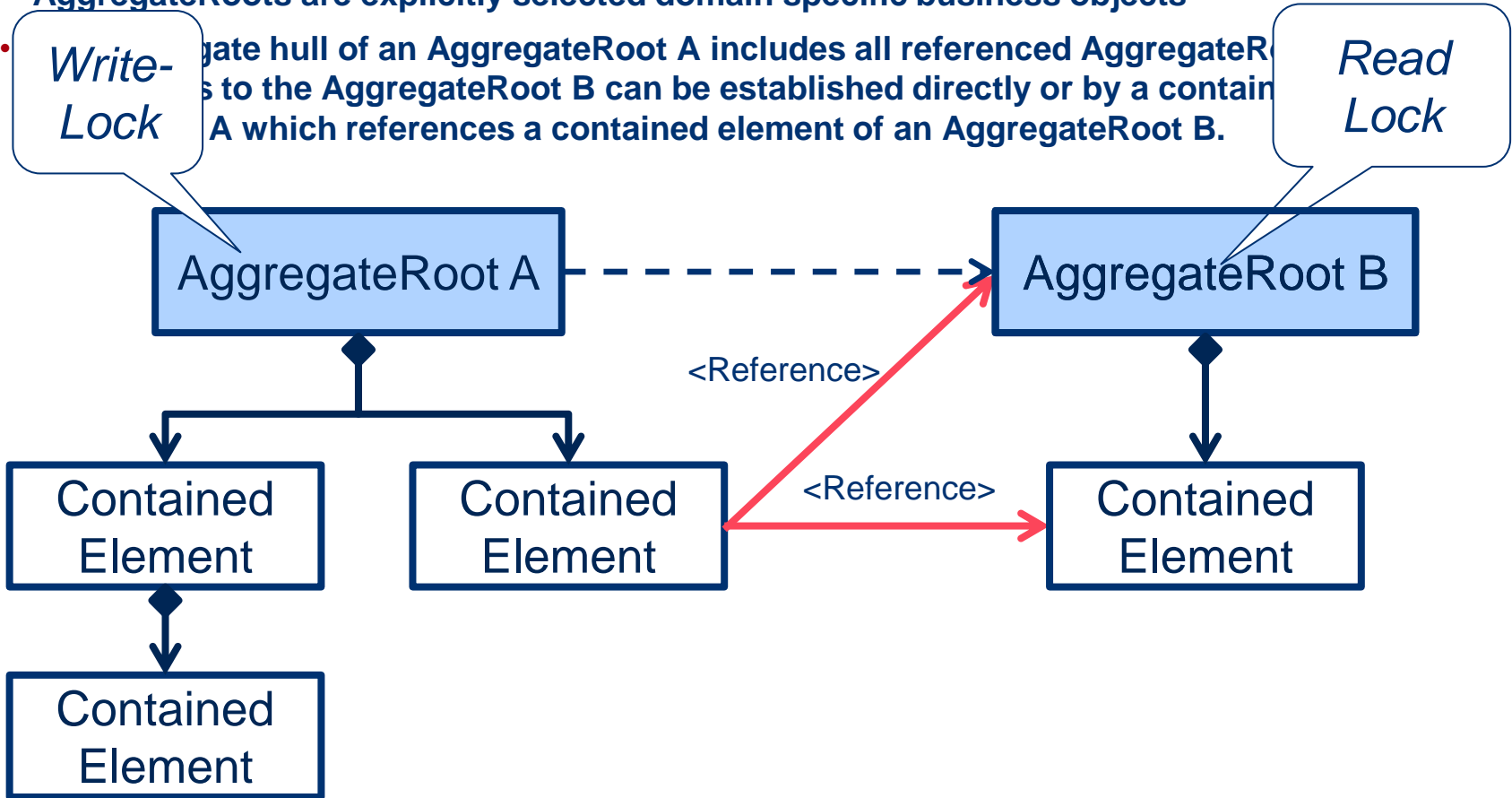
# Collaboration and Editing

- No essential changes to the existing edit use cases (textual SOA-DSL of project MAIA)
- Collaboration
  - Transaction/locking model within the repository
  - Logical transactions & locking
  - Check in / check out paradigm
  - Locking granularity
  - Conflict detection and handling
- Editing
  - Textual DSL representation as snapshots
  - Fan in principle (channels)



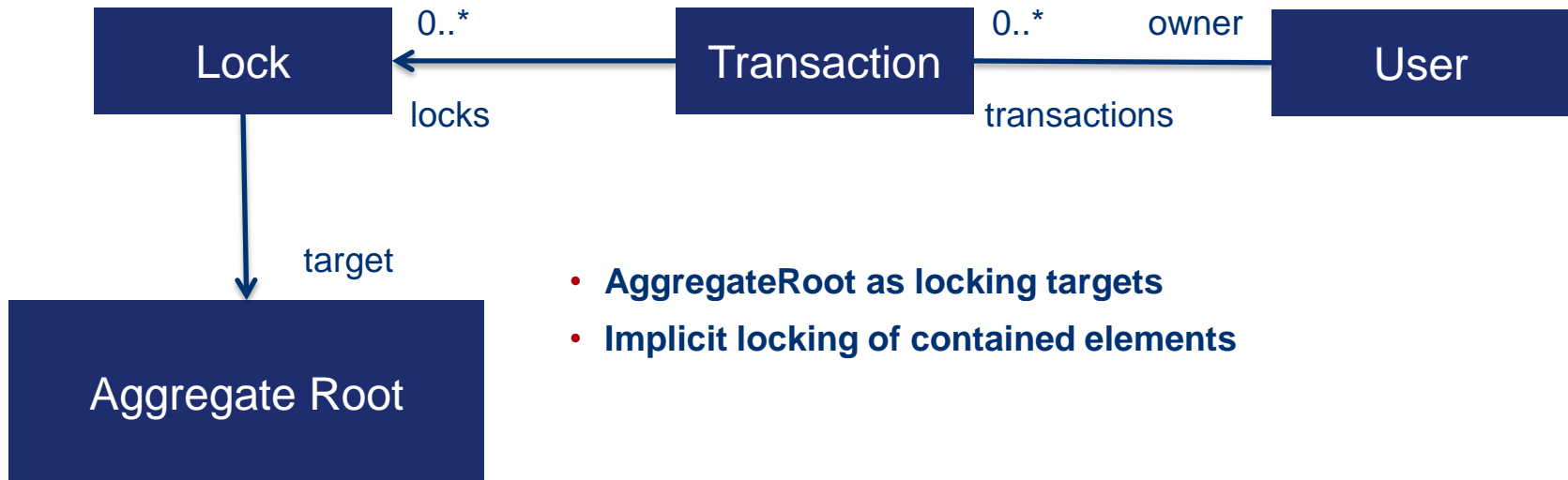
# Collaboration and Editing – „Aggregate Hull“

- AggregateRoots are explicitly selected domain specific business objects
- The aggregate hull of an AggregateRoot A includes all referenced AggregateRoots. A direct reference from A to the AggregateRoot B can be established directly or by a contained element of A which references a contained element of an AggregateRoot B.



# Collaboration/Transaction Metamodel

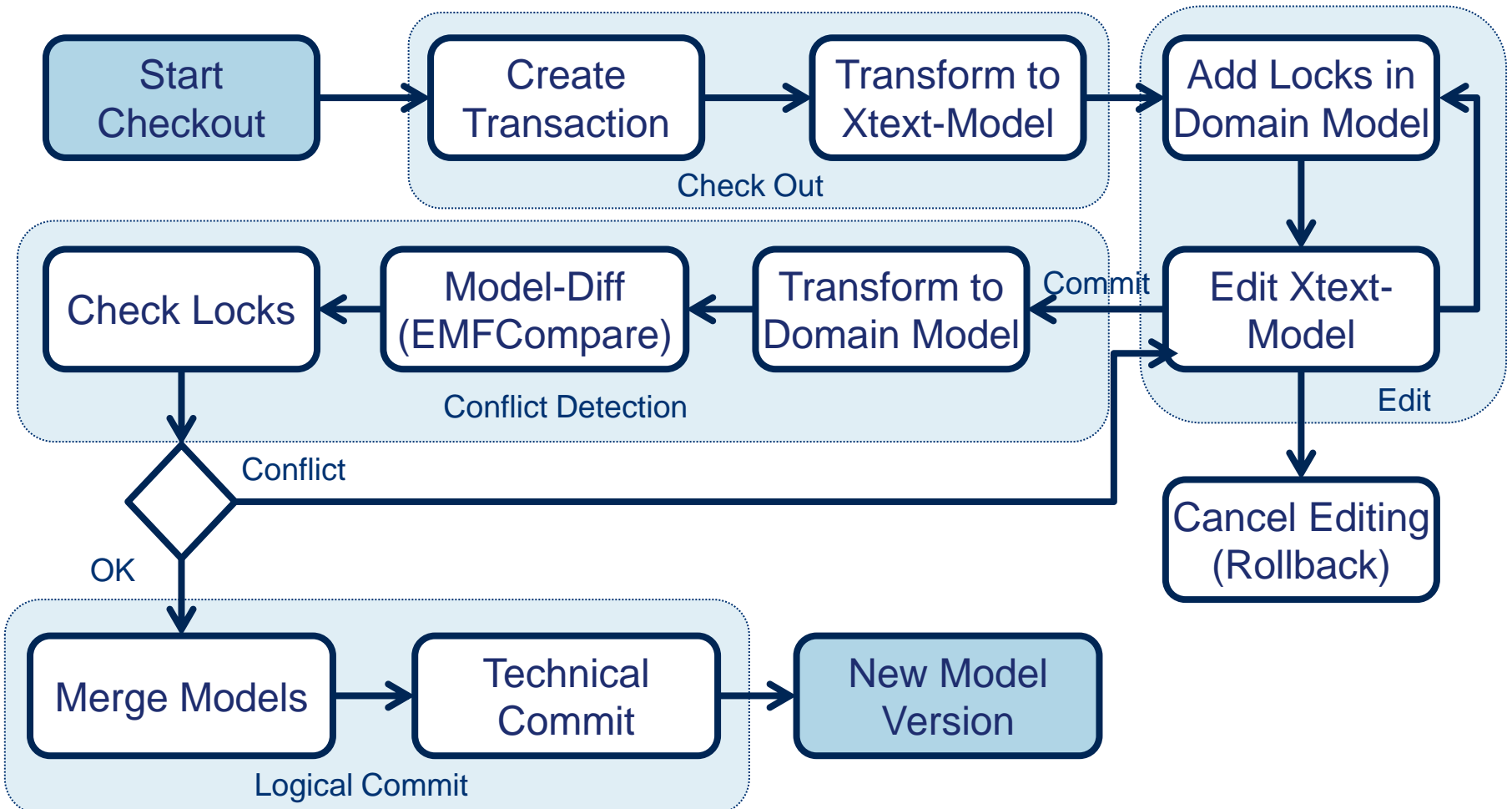
- **Main elements of a simple metamodel for collaboration/transaction**
  - Transaction with status (open/closed)
  - Lock with type (write / shared-read)
    - Only pessimistic locking
  - User management



- **AggregateRoot as locking targets**
- **Implicit locking of contained elements**

# Checkout-Edit-Commit

Edit model:



# Aggregate Root View

- Transaction management
- Locking

Add Locks

The screenshot shows the 'Aggregate Root View (Chuck)' application window. At the top, there is a toolbar with buttons for 'Init FT', 'Login', 'Open Transaction', 'Close Transaction', and 'Add Lock'. There are also input fields for 'Name Filter' and 'Type Filter' (set to 'all'). Below the toolbar is a table with three columns: 'Name', 'Type', and 'Package'. The table contains various components and services, some of which are highlighted in green or cyan. A callout box labeled 'Write-Lock' points to the 'Decorator for LifeContractRelation (v1.0)' row. Another callout box labeled 'Shared-Read-Lock' points to the 'LifeContractService (v4.0)' row. A third callout box labeled 'Add Locks' points to the 'Add Lock' button in the toolbar.

Name	Type	Package
LifeContract (v1)	Component	<root>.ch.mobi.auskunftssystem
LifeContractRelation (v1)	Component	<root>.ch.mobi.auskunftssystem
Decorator for LifeContractRelation (v1.0)	ComponentDecoration	
Decorator for LifeContractRelation (v1)	ComponentDecoration	
default	ComponentInstance	<root>.ch.mobi.auskunftssystem
default	ComponentInstance	<root>.ch.mobi.auskunftssystem
~FeatureSoap:~soapNSMajor:ch.mobi.mai...	FeatureSoap	
LifeContractRelationService (v1.0)	Service	<root>.ch.mobi.Auskunftssystem
LifeContractService (v4.0)	Service	<root>.ch.mobi.auskunftssystem
LifeContractService (v3.0)	Service	<root>.ch.mobi.Auskunftssystem
LifeOfferRelationService (v1.0)	Service	<root>.ch.mobi.Auskunftssystem
Decorator for LifeOfferRelationService (v1.0)	ServiceDecoration	
Decorator for LifeContractRelationService (v1...	ServiceDecoration	
Decorator for LifeContractService (v3.0)	ServiceDecoration	
Decorator for LifeContractService (v4.0)	ServiceDecoration	

# Transaction and Collision View

- **Transaction View:**

- Overview of transaction and locks in Domain Model
- Locks: Status (write / shared-read) and referenced AggregateRoots

**Transactions**

Name	Time	Status
nT-Elvis-1363803245874	(20....	open
nT-Chuck-13638033261...	(20....	open

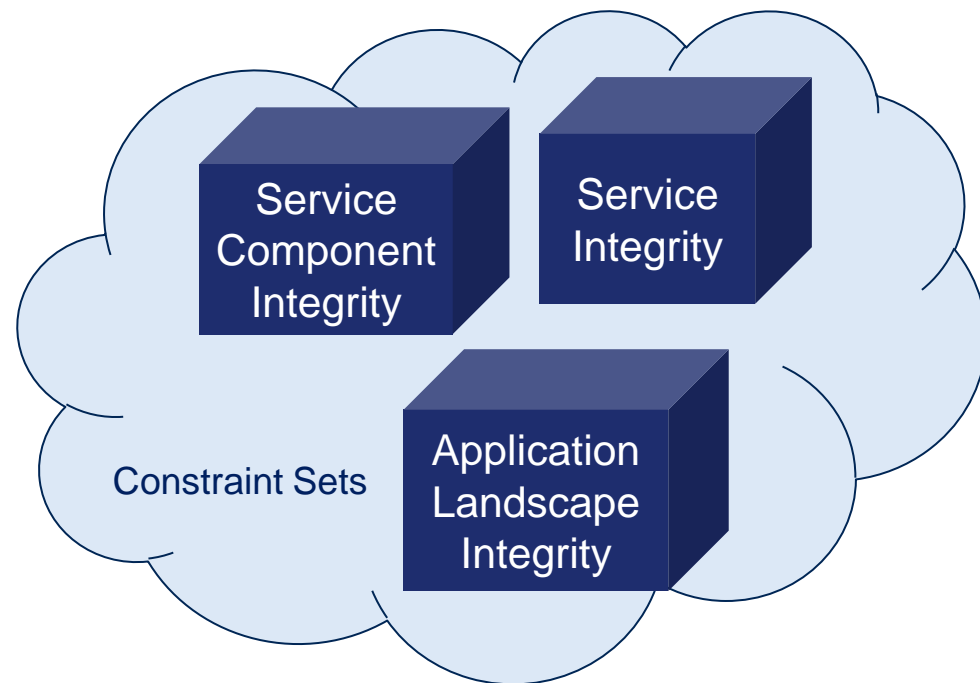
**Locks**

Name	Type	Target	Target Type
(:SimpleType)-1363803326216	read	Date	SimpleType
(:ServiceDecoration)-1363803326216	read	Decorator for LifeContractS...	ServiceDecoration
(:TypeSubsystem)-1363803326216	read	Commons (v2.0)	TypeSubsystem
(:SimpleType)-1363803326216	read	Boolean	SimpleType
(:SimpleType)-1363803326216	read	String	SimpleType
(:SimpleType)-1363803326216	read	Short	SimpleType
(:TypeSubsystem)-1363803326216	read	LifeContract (v4.0)	TypeSubsystem
(:Service)-1363803326216	write	LifeContractService (v4.0)	Service
(:Service)-1363807228384	read	LifeContractRelationService...	Service
(:ComponentDecoration)-1363807...	read	Decorator for LifeContractR...	ComponentDecoration
(:Component)-1363807228384	write	LifeContractRelation (v1)	Component

- **Collision View:**

- Show conflicts if any
- Show type of conflicts (write/write, write/read, read/write)

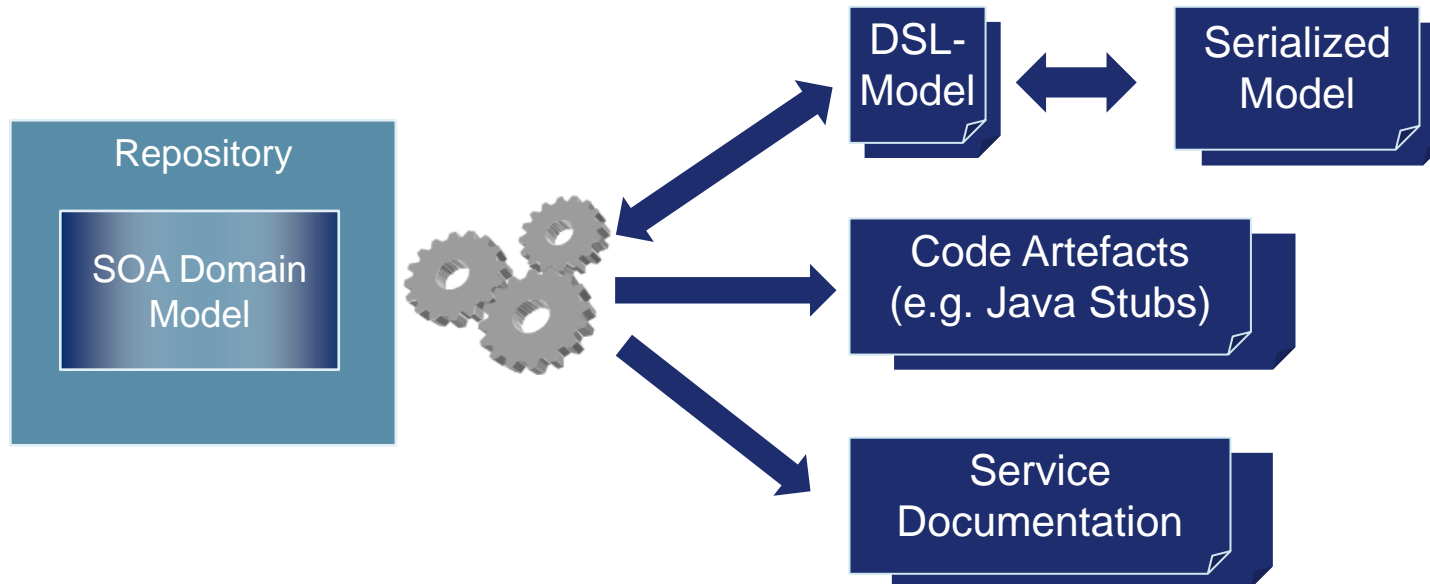
- Distinguished constraint sets with different grades of wellformedness
- Validation is based on Domain-Model - including rules referring previous (service) versions (life cycle management) - independent of the textual DSL
- Constraint language
  - Specific DSL vs. OCL
  - Technical integration





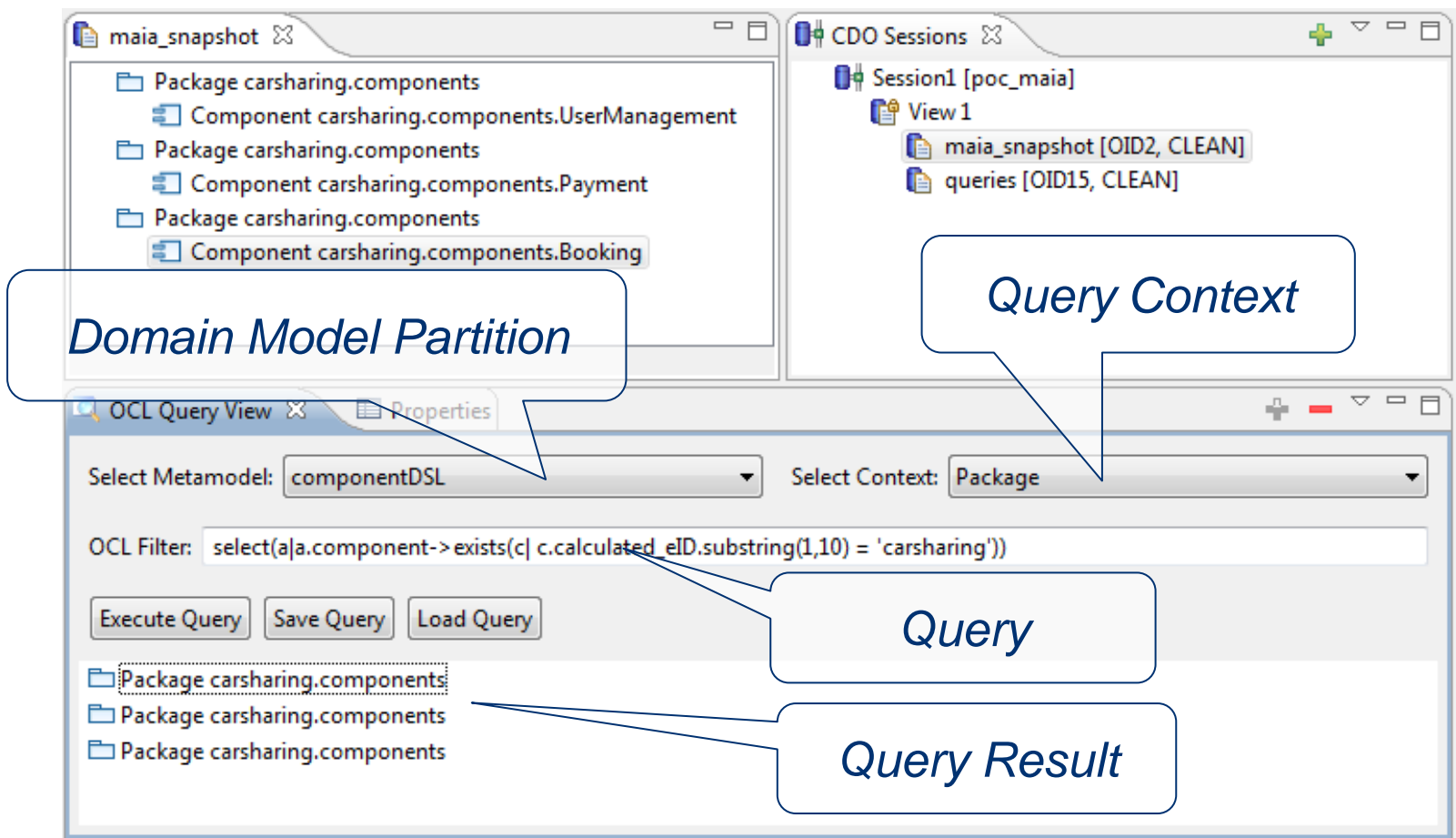
# Transformations & Generators

- Bidirectional transformation between DSL-Model and Domain-Model
- Fan out principle
- Generator integration (prototype vs. target vision)



# Queries & Reports (1/2)

- Generic query language as a base (e.g. OCL)
- Ad-hoc queries based on Domain-Model
- Queries can be stored in the repository (saved queries)



The screenshot displays a software interface with several windows and callouts:

- maia\_snapshot** window: Shows a tree view of a domain model partition with packages and components like `carsharing.components.UserManagement` and `carsharing.components.Booking`. A callout labeled *Domain Model Partition* points to this window.
- CDO Sessions** window: Shows a session named `Session1 [poc_maia]` with a view containing `maia_snapshot [OID2, CLEAN]` and `queries [OID15, CLEAN]`. A callout labeled *Query Context* points to this window.
- OCL Query View** window: Contains a dropdown for `Select Metamodel: componentDSL`, a dropdown for `Select Context: Package`, and an `OCL Filter` text box with the query: `select(a|a.component->exists(c) c.calculated_eID.substring(1,10) = 'carsharing'))`. Below the text box are buttons for `Execute Query`, `Save Query`, and `Load Query`. A callout labeled *Query* points to the OCL Filter text box.
- Properties** window: Shows a tree view of package instances, with the top item `Package carsharing.components` highlighted. A callout labeled *Query Result* points to this window.

# Queries & Reports (2/2)

- Table visualization of elements connected by a metamodel path

**OCL Path Definition:**  
`self.members->select(m|m....`

Dependencies Table View

Select Metamodel: `de.bmiag.dynamod.repository.systemmodel.vb6.structure`

X-Axis: `Form`

Y-Axis: `Form`

OCL Path: `self.members->select(m|m.oclsTypeOf(Operation)).oclAsType(common::structure::ExecutableElement).staticCallees->select(c|c.oclsKindOf(vb6::structure::Member)).oclAsType(vb6::structure::Member).containingModule`

Dependencies	frm_Findbuch	frm_SuchenErse	Target	frmBENABT1	frmBENUTZER	frmBENUTZER2	frmIndexakte	frmKLASSIFIKA	frmKLASSIFIKA'	frmKopy	frmLFDNR	frmLogin	frmMain
frm_Findbuch	✓	✗											
frm_SuchenErset	✗	✓		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
frmAbout	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
frmABTEILUNG	✗	✗		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
frmABTEILUNG1	✗	✗		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
frmAKTE	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
frmLFDNR	✗	✗		✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
frmLogin	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
frmMain	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
frmMisch	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
frmSammel	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
frmSplash	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
frmSTAT	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
frmSTGES	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
frmSTKL	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
frmSUCH	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
frmSucherg	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
frmtabindex	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

**OCL Path:**  
**Matching Elements**

CSV-Export

# Conclusion & Prospect

---

- **Conclusion**

- Advantages of modeling technologies in the context of a SOA repository
  - Reuse of generic building blocks => efficiency
  - Consistency from model to runtime and documentation
  - Metamodel oriented construction => flexibility, adaptability
- Shift of information ownership from textual model representation into the repository (model runtime)
  - => Enhanced support of usage scenarios
  - => Enhanced support of different users / roles
  - => Improved governance support
  - => Multi channel editing (fan in)

- **Prospect**

- Schema Evolution
- Alternative persistence media
  - GraphDB, ...



## Contact

---

**Tom Stahl, Stefan Zeug**

**b+m Informatik AG**  
Rotenhofer Weg 20  
24109 Melsdorf

[t.stahl@bmiag.de](mailto:t.stahl@bmiag.de), [stefan.zeug@bmiag.de](mailto:stefan.zeug@bmiag.de)

T +49 4340 404- 0  
F +49 4340 404-111

