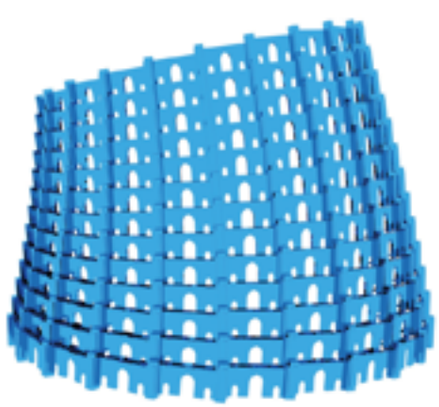
A Coupled Transformations Chrestomathy Based on Compilable and Executable Megamodels

Ralf Lämmel
Software Languages Team
University of Koblenz-Landau, Germany
http://www.softlang.org/



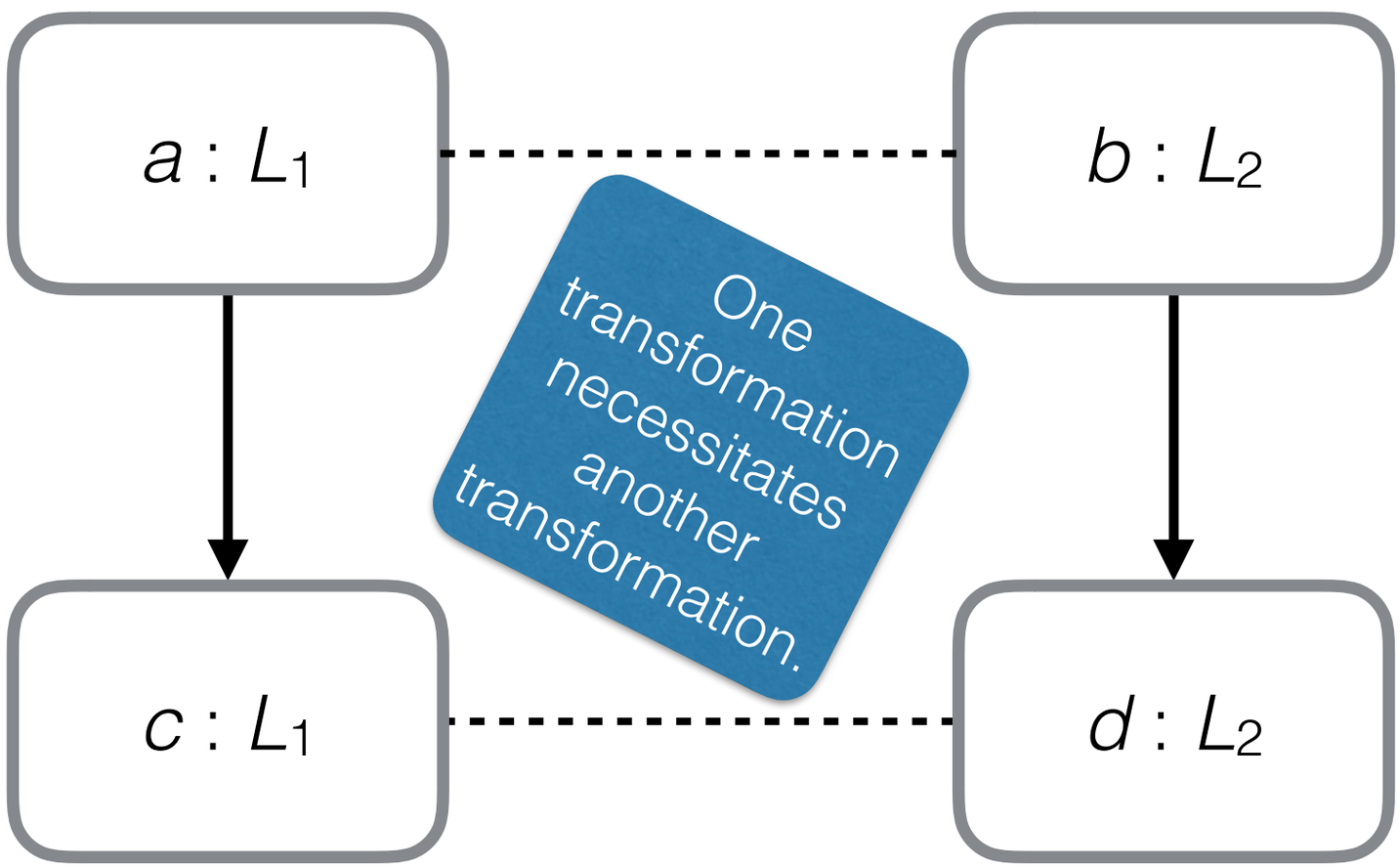
softlang

What's a *coupled transformation*?

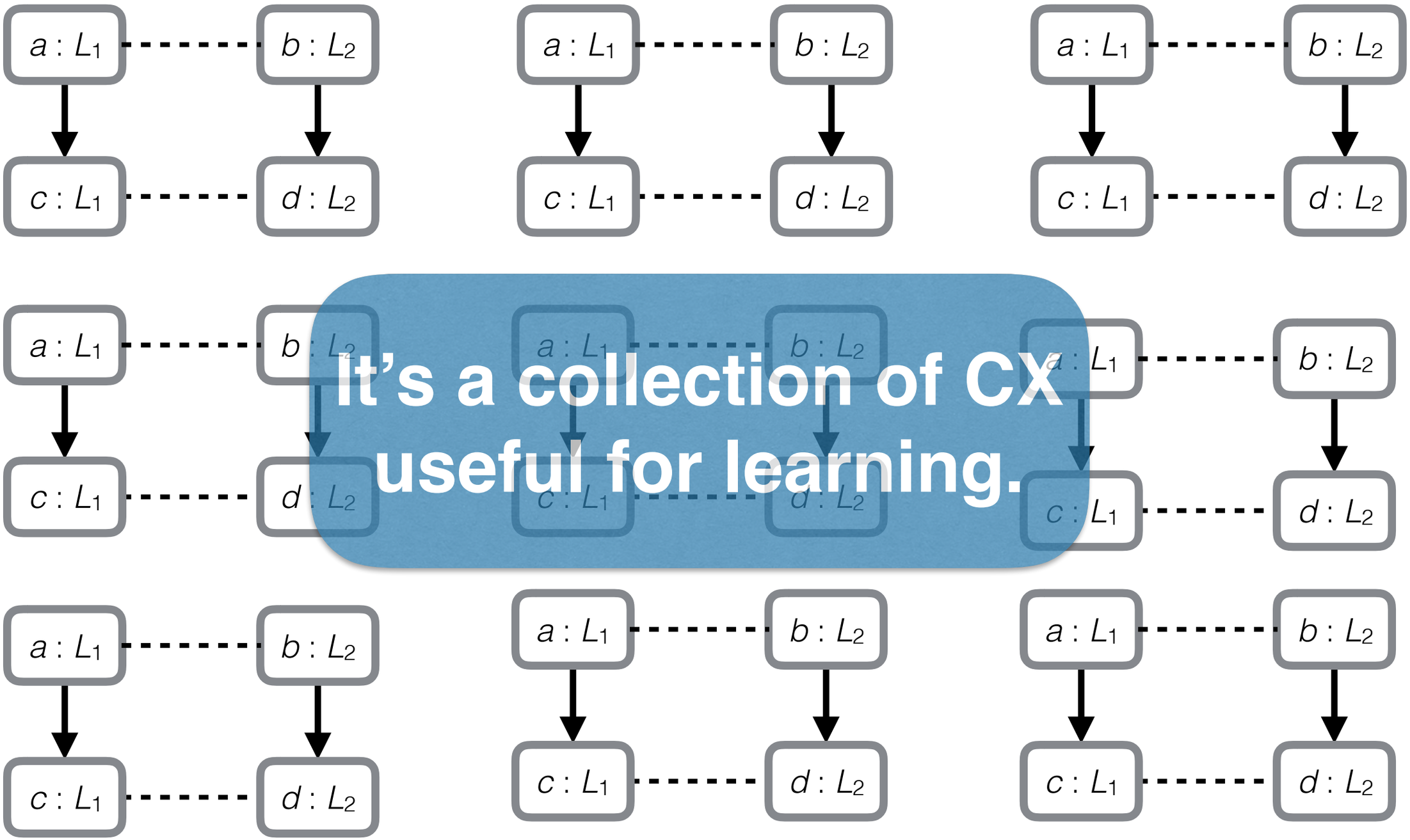
- - - - - - - - - Consistency, e.g., conformance

——→ Transformation, often in the sense of **evolution**

$x : L$ Artifacts 'typed' by languages



What's a coupled transformation *chrestomathy*?



It's a collection of CX useful for learning.

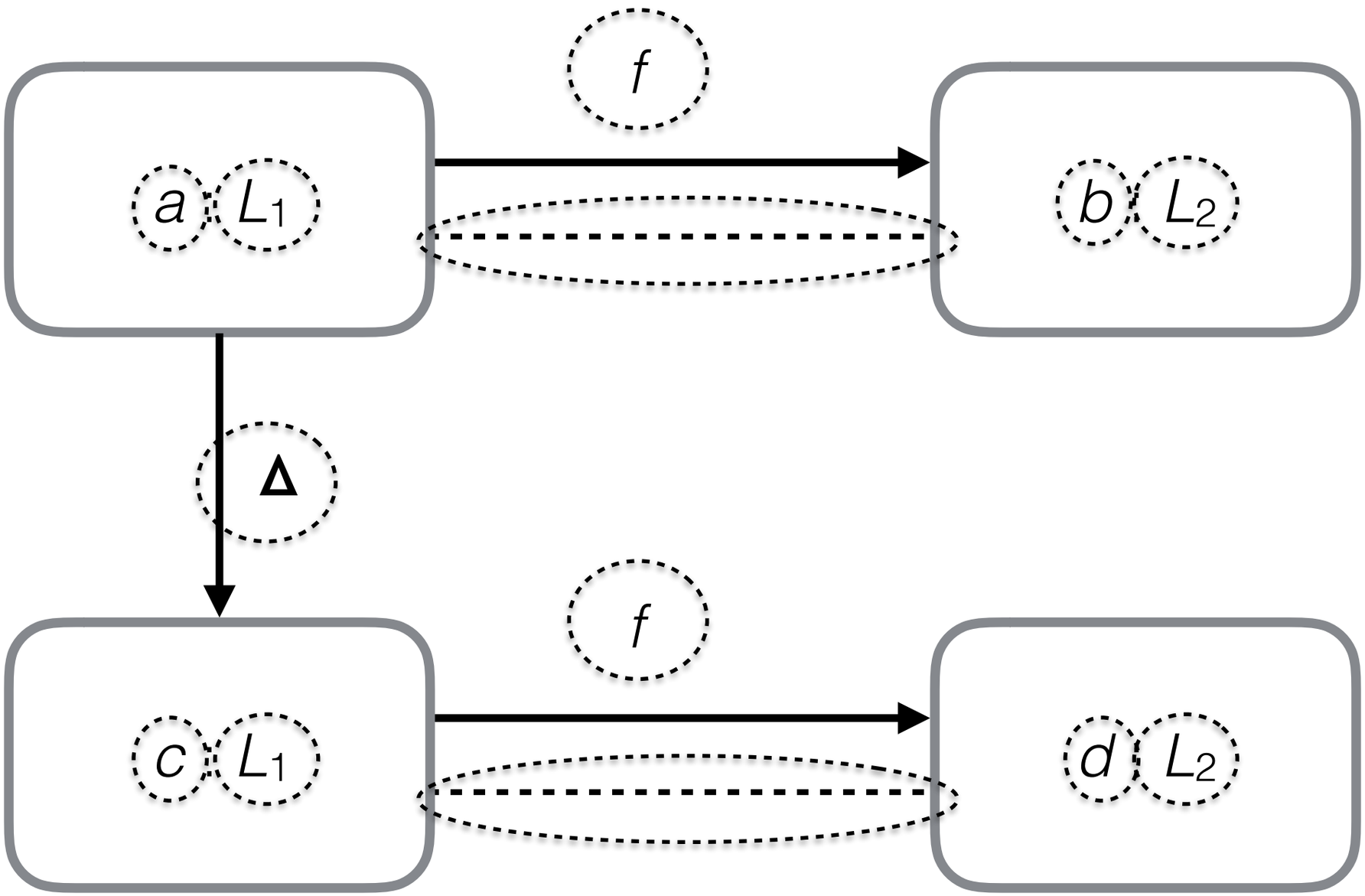
...

What's a megamodel and **why in the world** do we need to compile it?

Let's see later.

CX by *mapping*

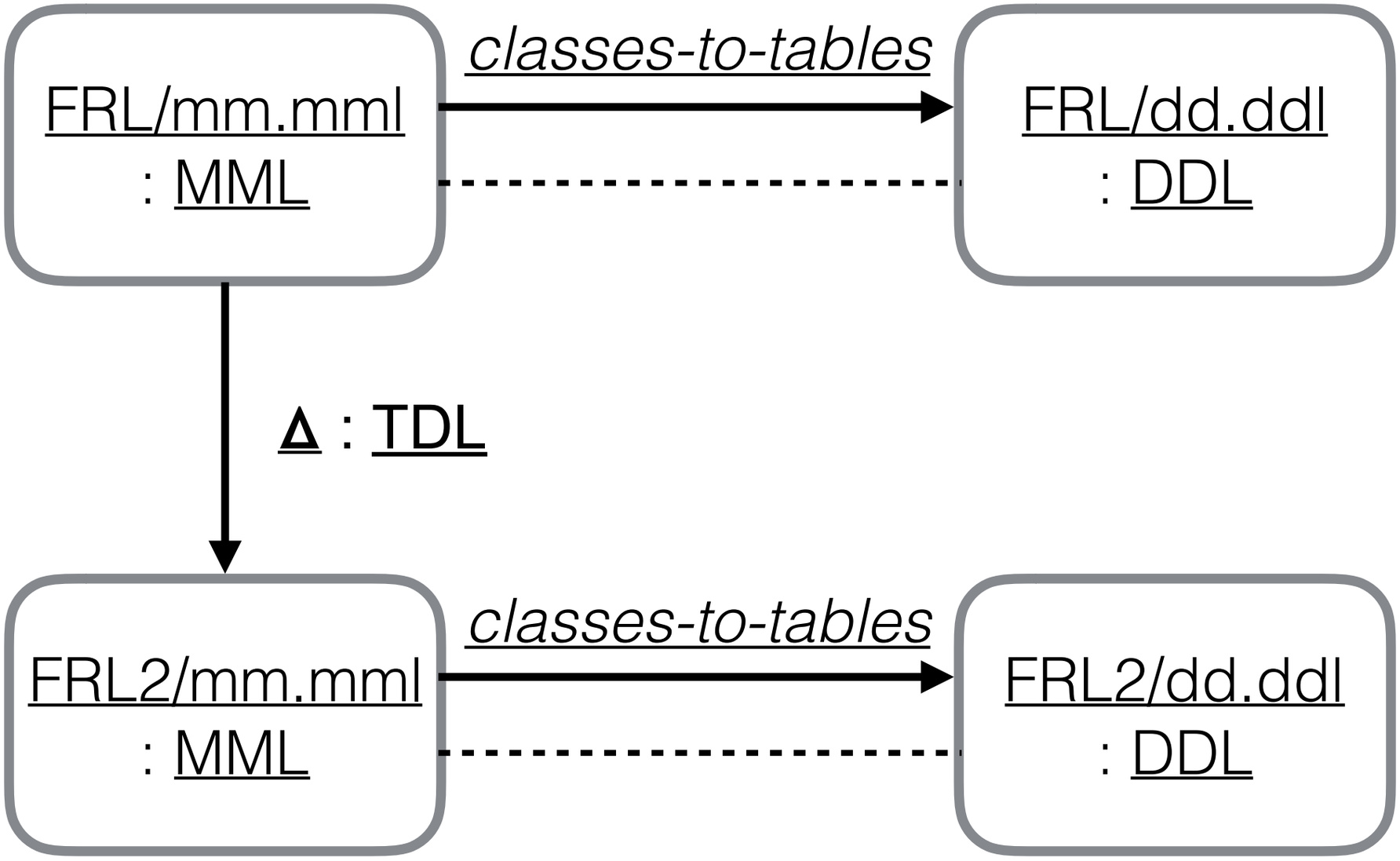
Let's instantiate the megamodel!



CX by *mapping*

Everything is linked to artifacts!

FRL — Family ("by reference") Language
MML — Meta Modeling Language
DDL — Data Definition Language
TDL — Term Difference Language



Why such a chrestomathy?

- ❖ CX are relatively complex transformations.
- ❖ Notions covered: deltas, incrementality, consistency, …
- ❖ The chrestomathy helps capturing the essence.
- ❖ Different CX forms and scenarios can be compared.

How to do such a chrestomathy?

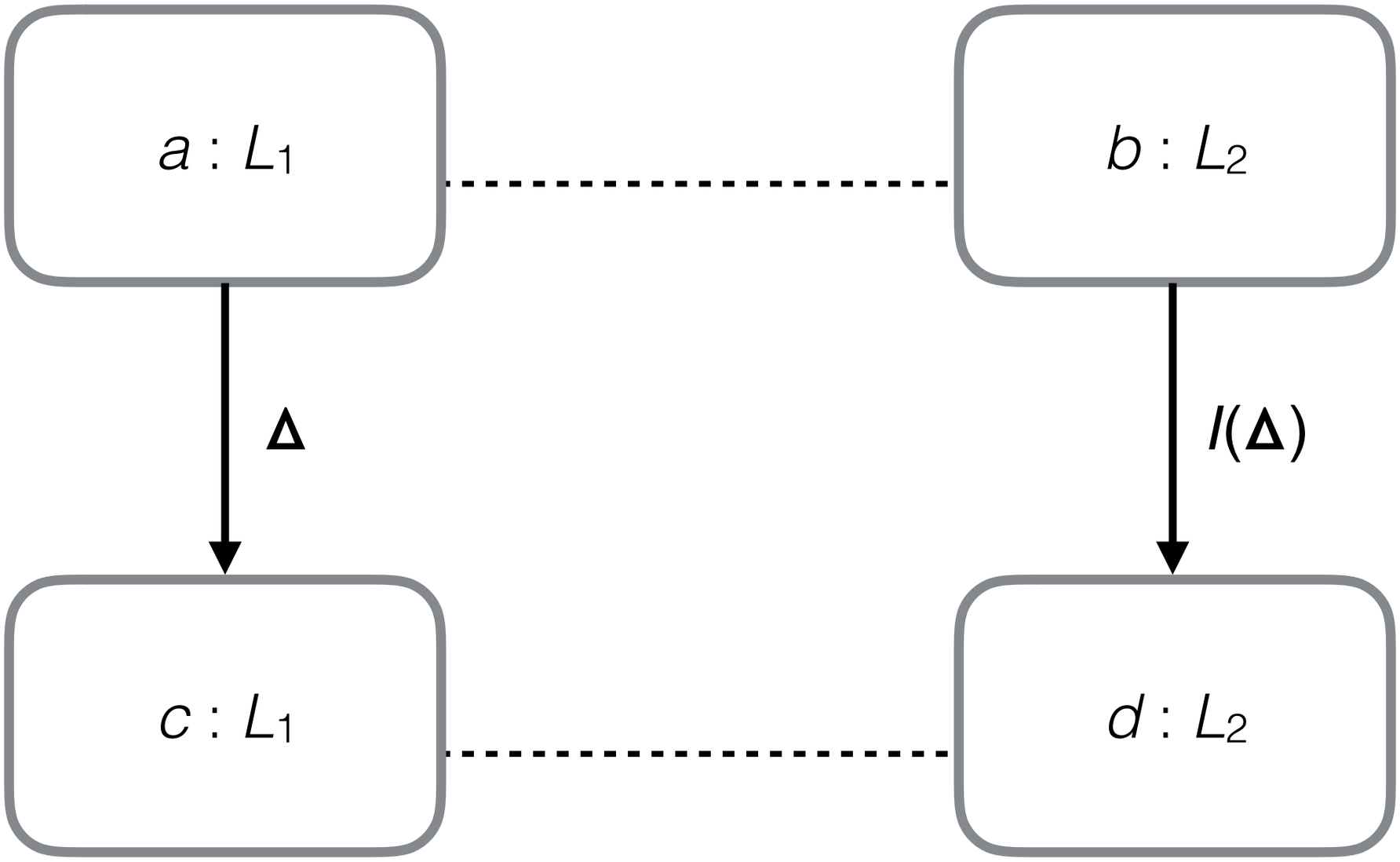
- ✦ Use a software language repository (SLR).
- ✦ Use technology-neutral metaprogramming.
- ✦ We use the (Prolog-centric) YAS: Yet Another SLR.
- ✦ We use megamodeling for executable patterns of CX.

Demo time

```
git clone https://github.com/softlang/yas.git
cd yas
make // if you have SWI-Prolog installed
make view // if you have GraphViz/dot installed
find . -name "*.egl" // This lists EGL grammars.
...
```

CX by *incremental mapping*

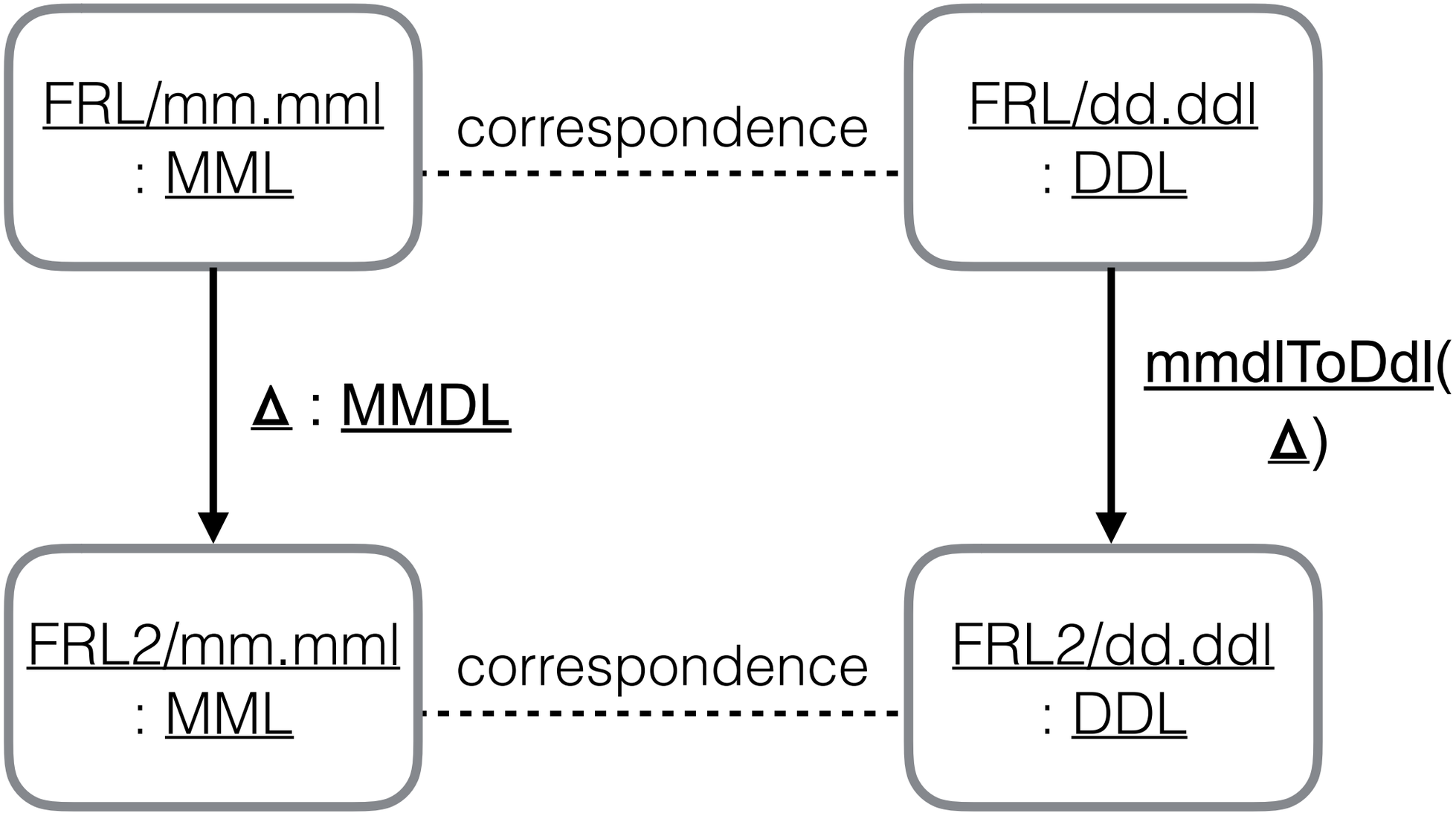
Let's instantiate the megamodel!



CX by *incremental mapping*

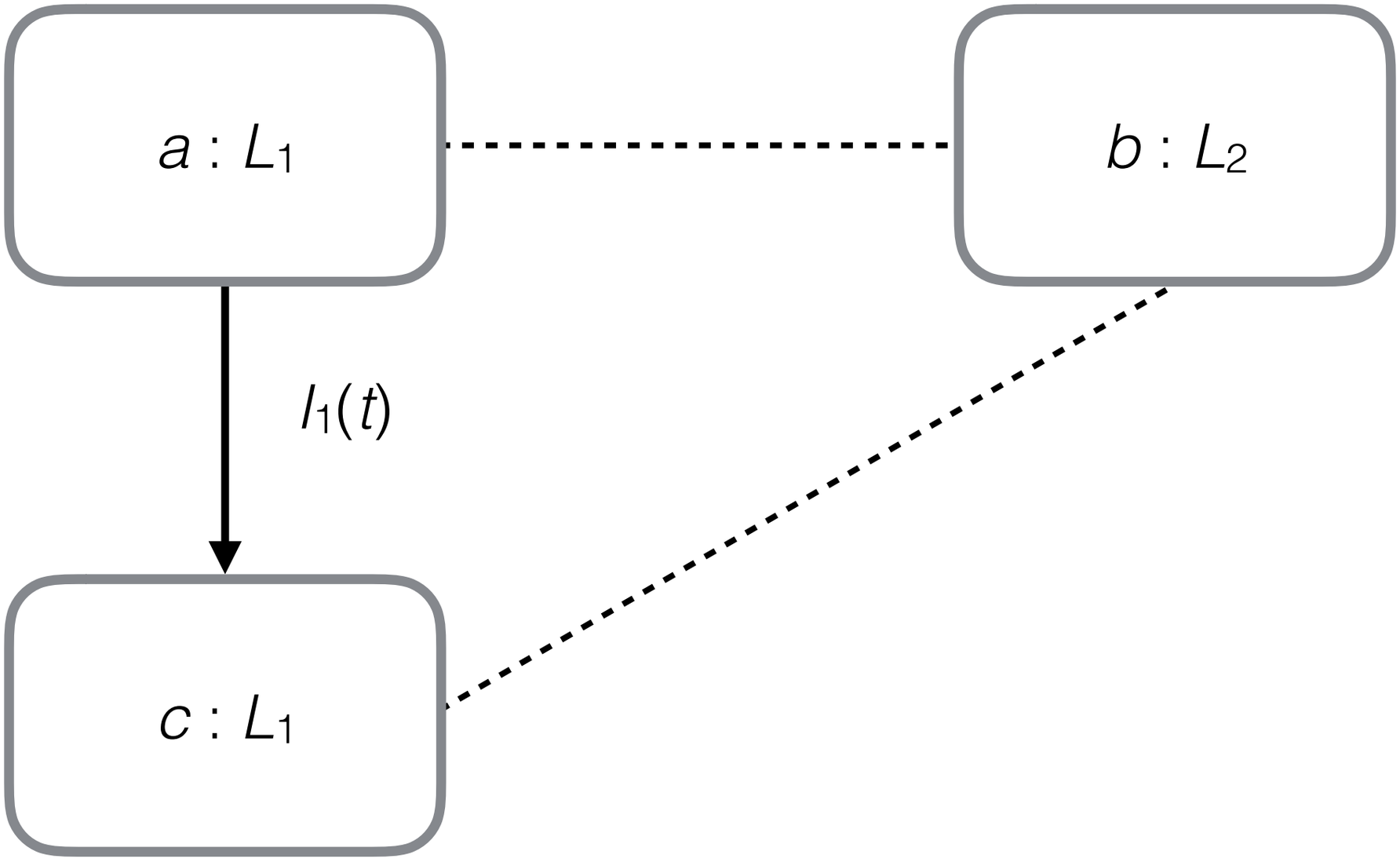
Everything is linked to artifacts!

FRL — Family ("by reference") Language
MML — Meta Modeling Language
DDL — Data Definition Language
MMDL — MetaModel Difference Language



CX by *invariant consistency*

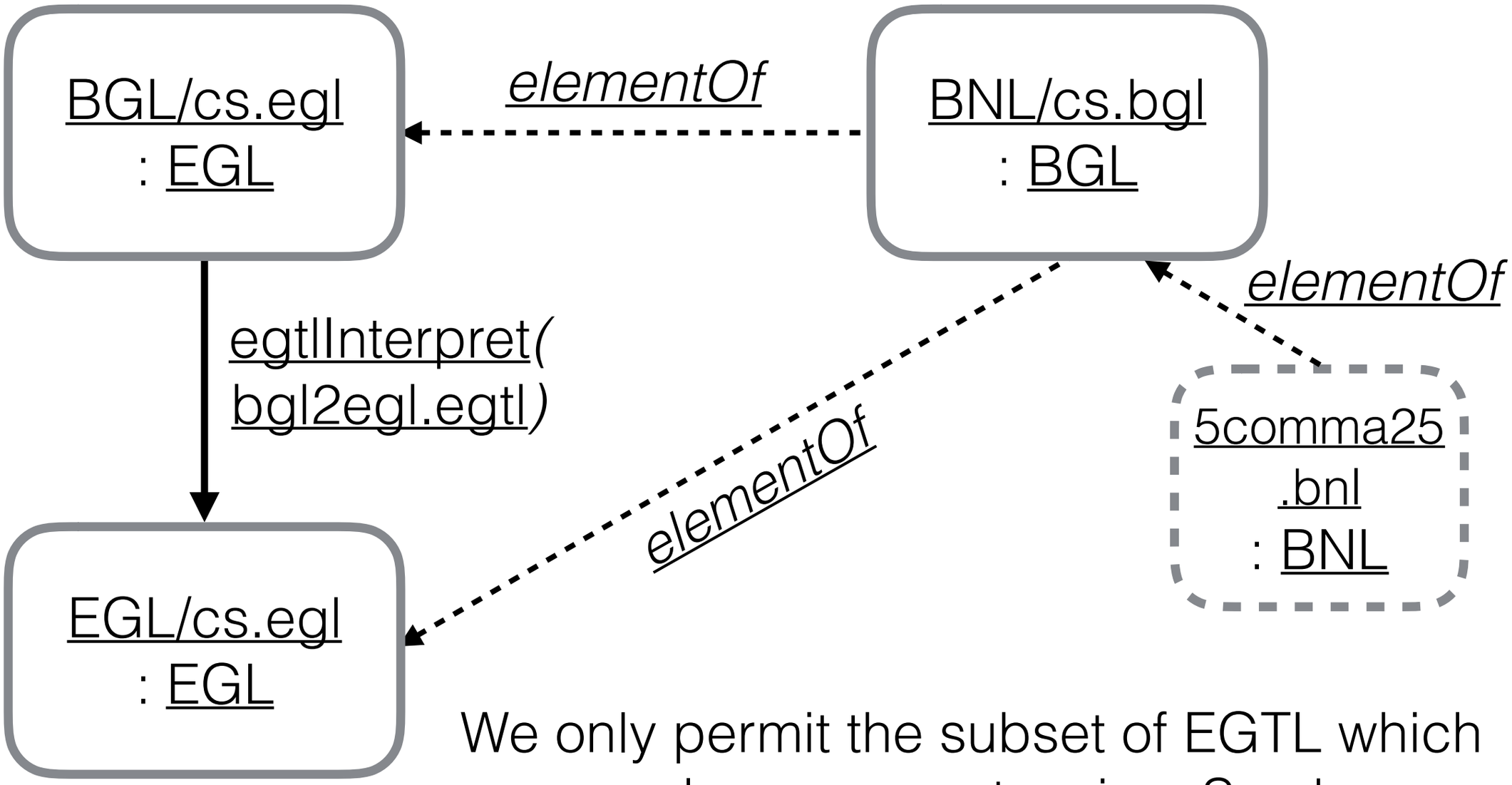
Let's instantiate the megamodel!



CX by *invariant consistency*

Everything is linked to artifacts!

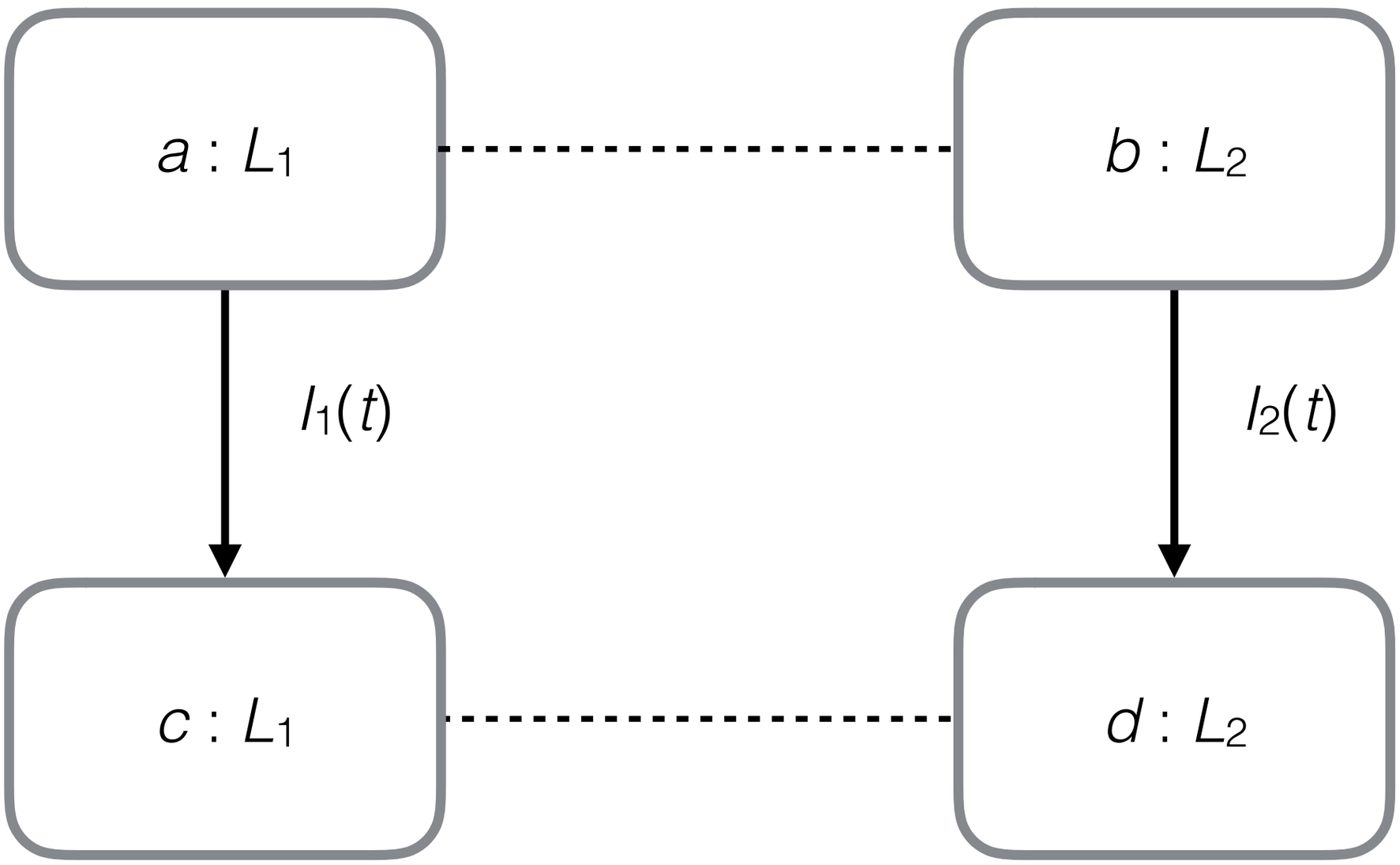
BNL — Binary Number Language
BGL — Basic Grammar Language
EGL — Extended Grammar Language
EGTL — Extended Grammar Transformation Language



We only permit the subset of EGTL which serves language extension. See here.

CX by *co-transformation*

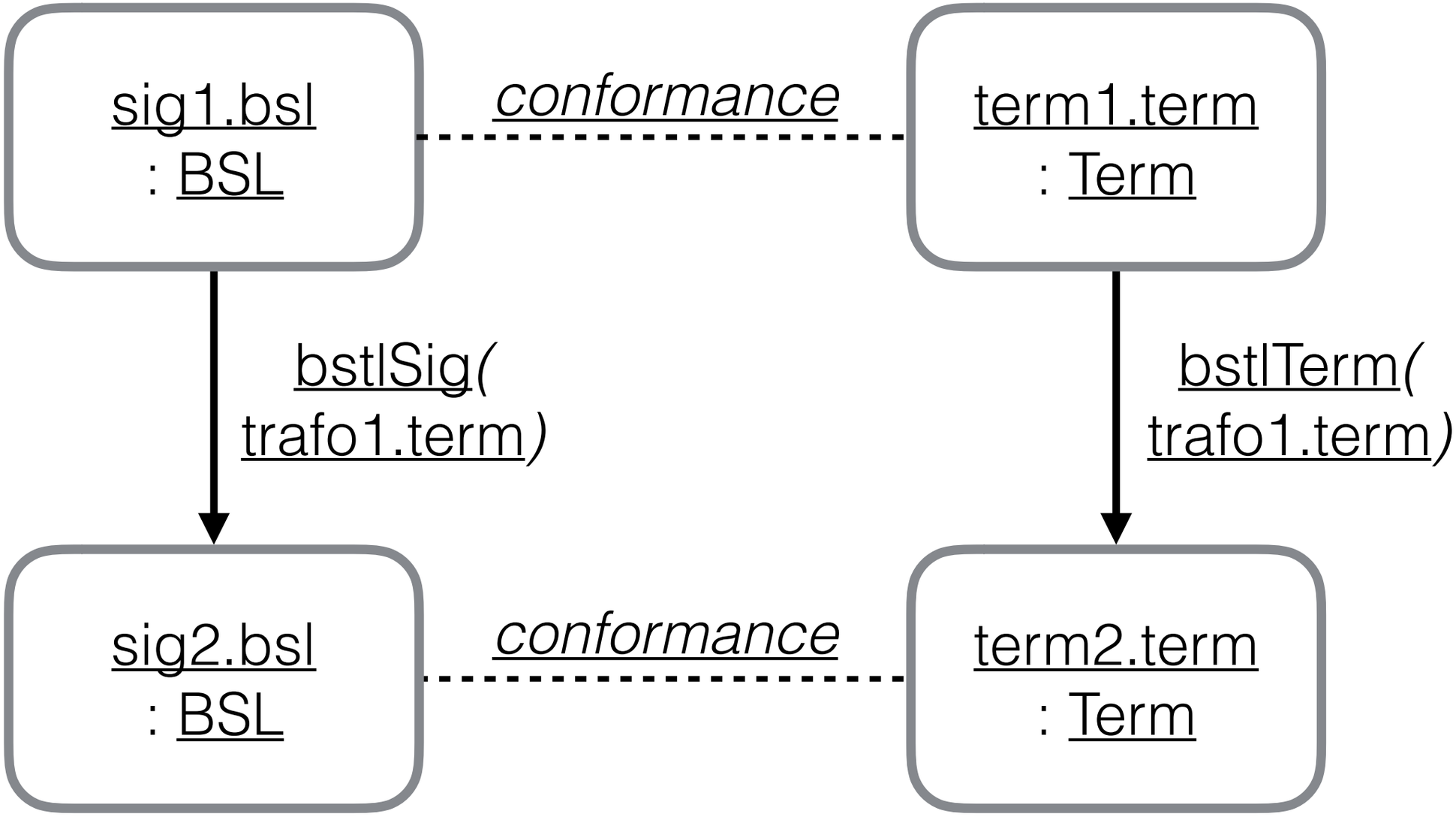
Let's instantiate the megamodel!



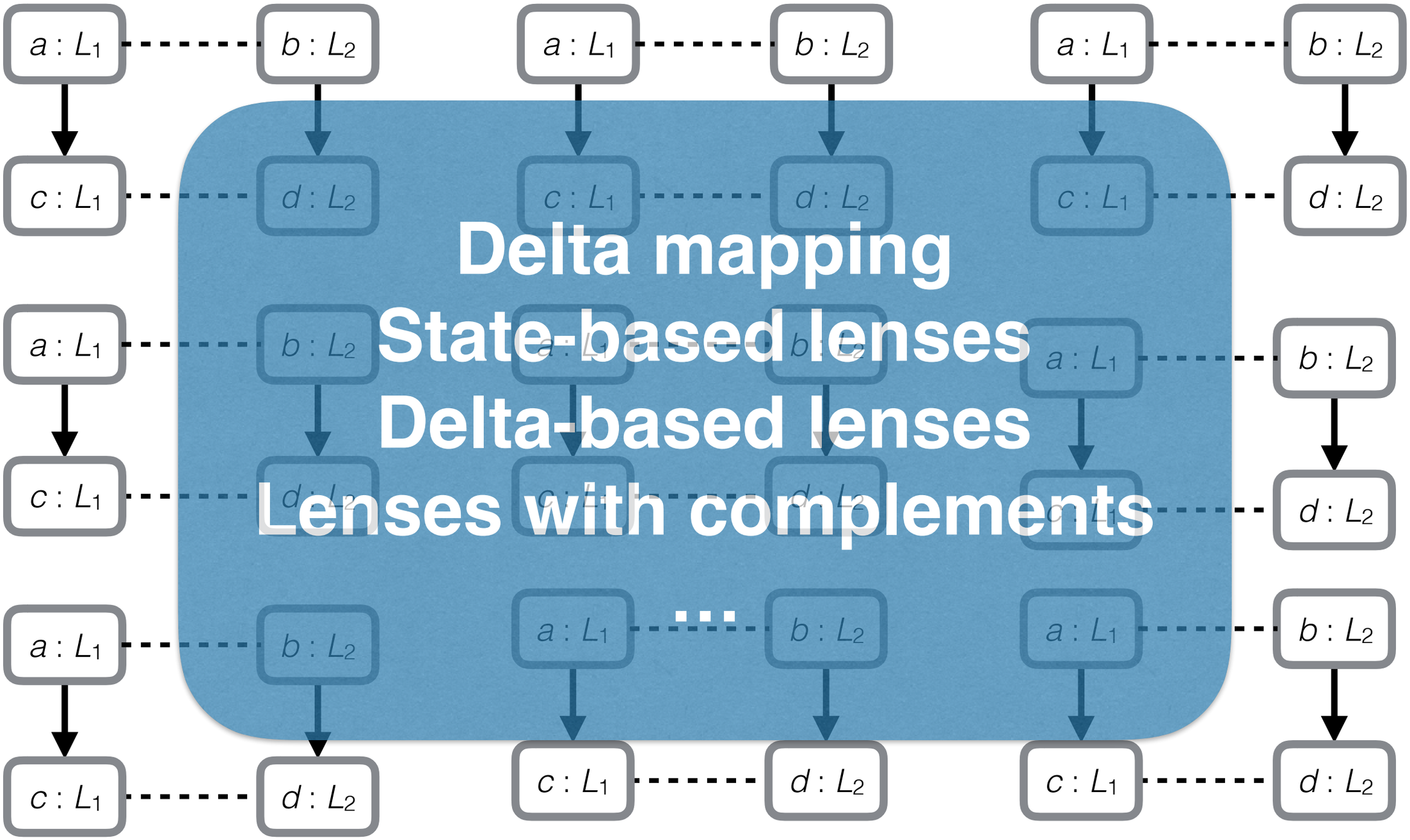
CX by *co-transformation*

Everything is linked to artifacts!

BSL — Basic Signature Language
Term — Terms conforming to signature
BSTL — Basic Signature Transformation Language



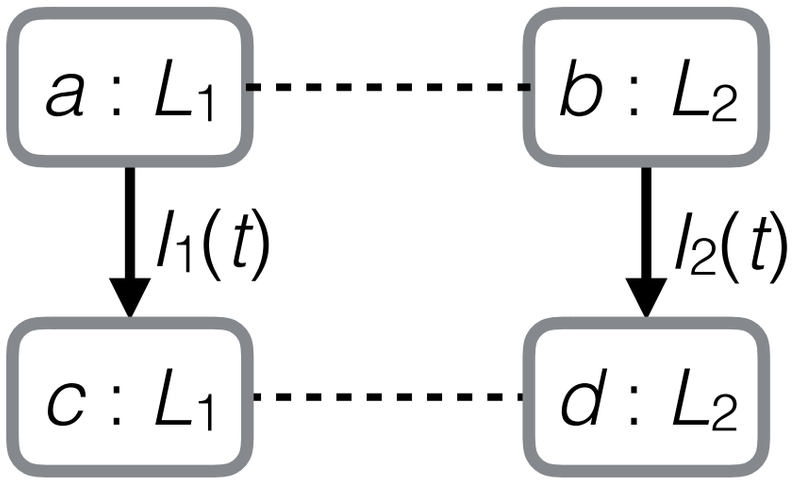
More CX



Delta mapping
State-based lenses
Delta-based lenses
Lenses with complements

...

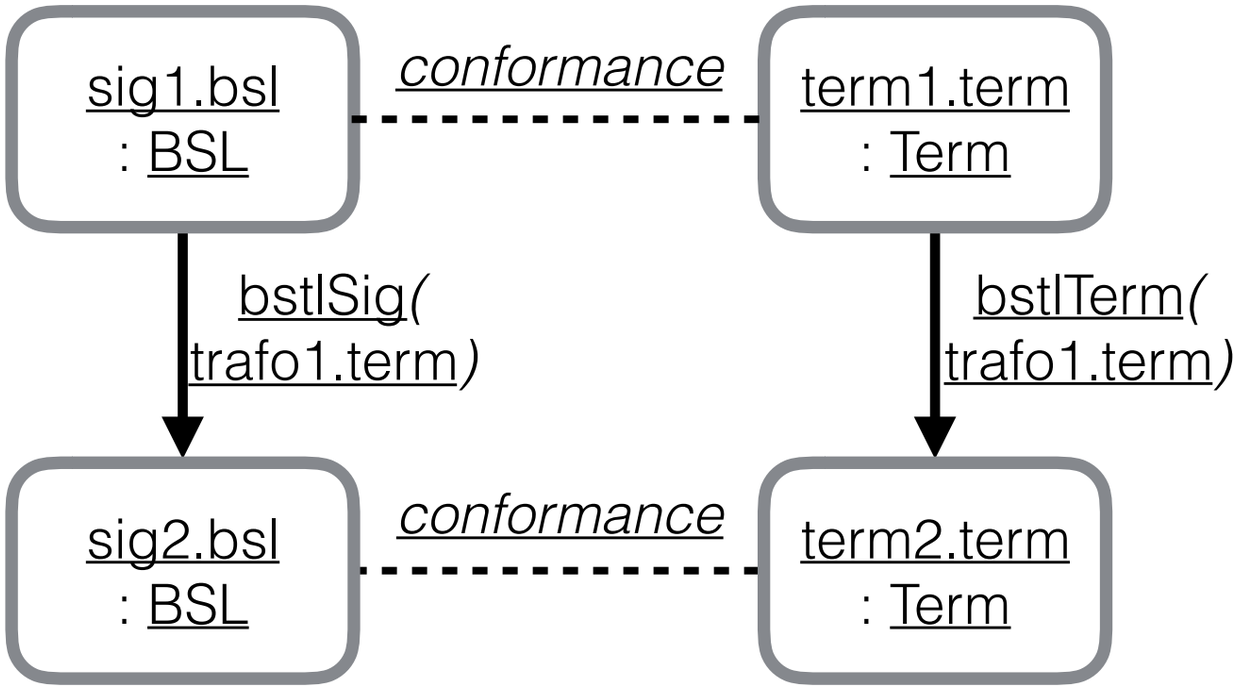
High-level megamodel CX by *co-transformation*



LAL megamodel <u>cx.transformation</u>

```
reuse coupling
reuse interpretation [ L2 ↦ L1, Any2 ↦ Any1 ]
reuse interpretation [ L1 ↦ L2, Any1 ↦ Any2 ]
axiom consistency { ∀ t ∈ XL. ∀ a, c ∈ L1. ∀ b, d ∈ L2.
  consistent(a, b)
  ∧ interpret(t, a) = c
  ∧ interpret(t, b) = d ⇒ consistent(c, d) }
```

Low-level megamodel CX by *co-transformation*



Ueber megamodel BSTL/tests/trafo1.ueber

```
[ elementOf('trafo1.term',bstl(term)),
  elementOf('term1.term',term),
  elementOf('term2.term',term),
  elementOf('sig1.term',bsl(term)),
  elementOf('sig2.term',bsl(term)),
  relatesTo(conformsTo,['term1.term','sig1.term']),
  mapsTo(interpret,['trafo1.term','term1.term'],['term2.term']),
  mapsTo(interpret,['trafo1.term','sig1.term'],['sig2.term']),
  relatesTo(conformsTo,['term2.term','sig2.term']) ].
```

Megamodel compilation

- Forall becomes exists
- Implication becomes conjunction
- …
- Instantiate languages, artifacts, functions, relations.
- Rely on interpretations at low level.

End of Talk — Thanks!

Work on megamodeling is joint work at softlang with:

- Andrei Varanovich
- Marcel Heinz
- Lukas Härtel
- Johannes Härtel

Thanks also to the SLE course 2016, specifically:

- Lukas Debald
- Christopher Held
- Philipp Seifer