Taxonomy of Flexible Linguistic Commitments

Dr. Vadim Zaytsev aka @grammarware

FlexMDE 2015

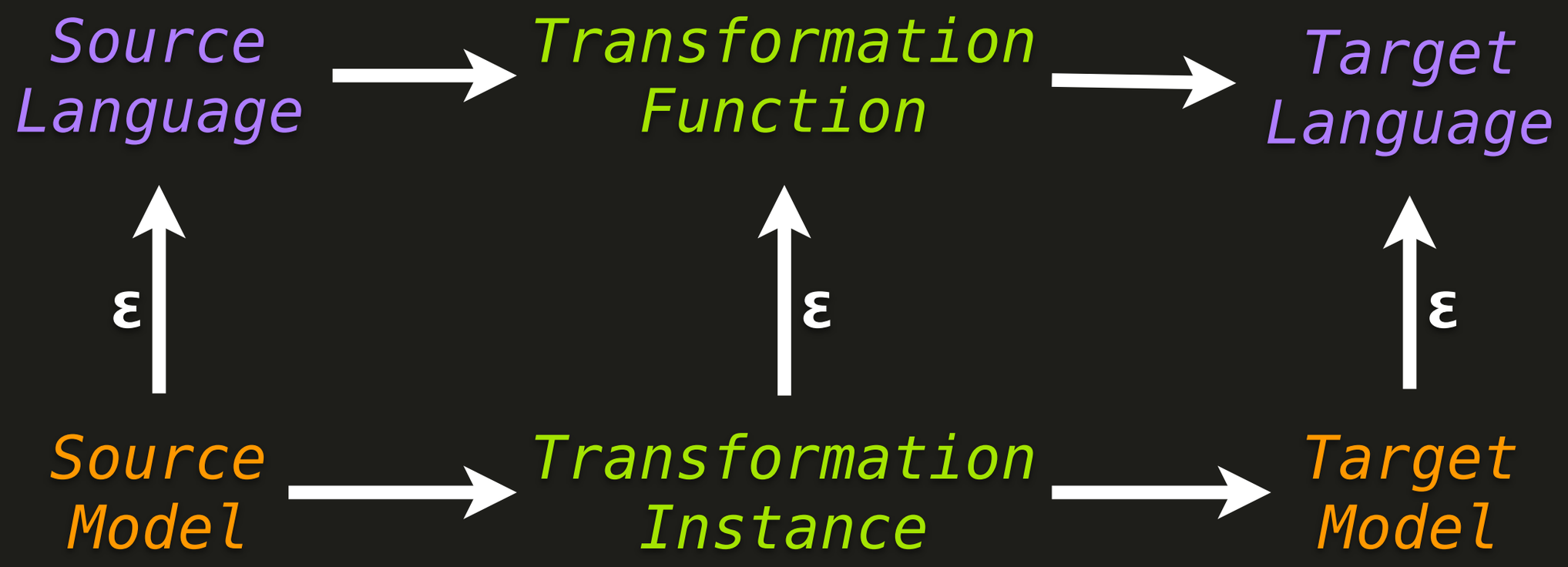
What if you miss?

http://cheezburger.com/635909/funny-memes-images-that-prove-design-isnt-for-everybody

Mega start

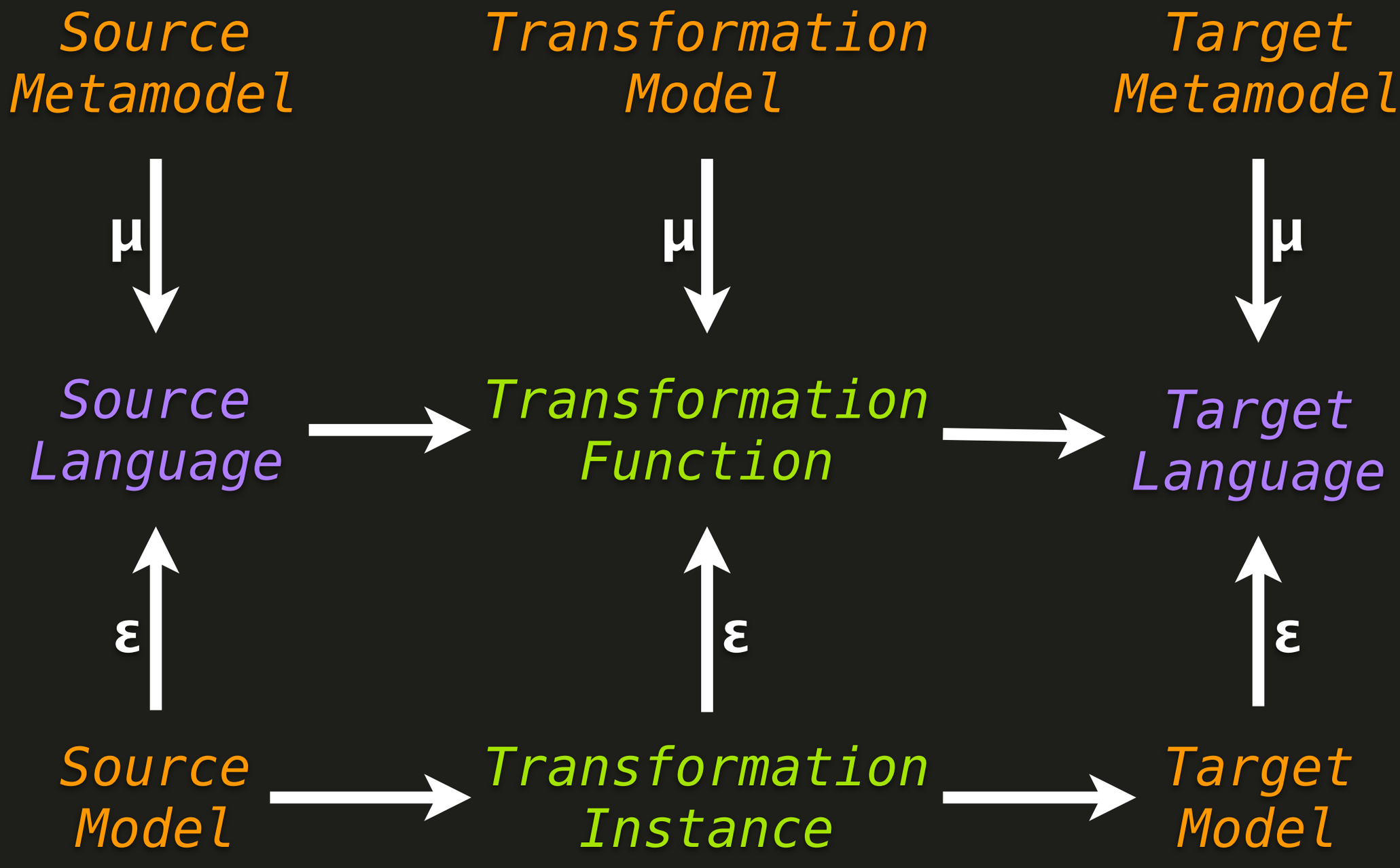
Jean-Marie Favre, Towards a Basic Theory to Model MDE, 2004.

Mega start



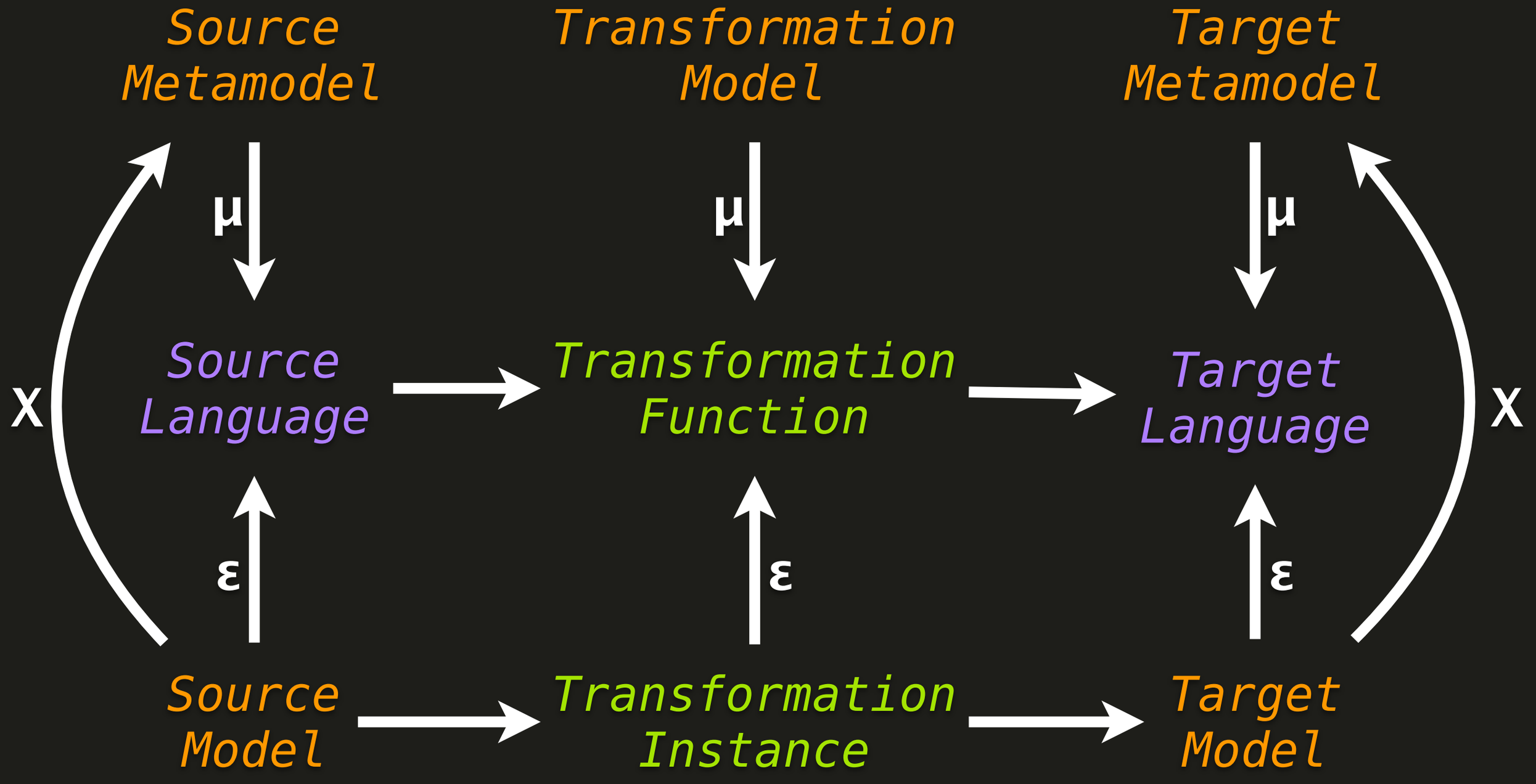
Jean-Marie Favre, Towards a Basic Theory to Model MDE, 2004.

Mega start



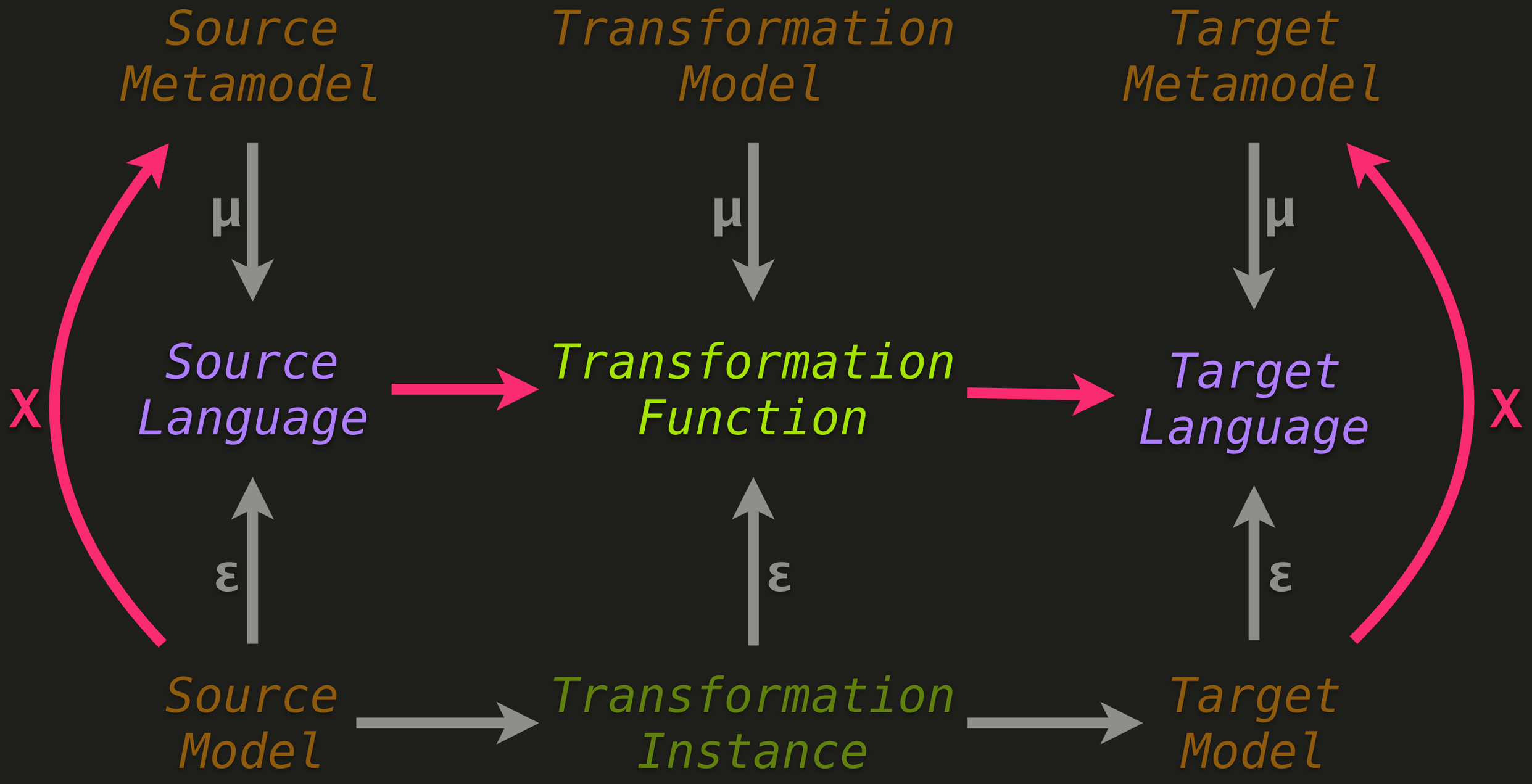
Jean-Marie Favre, Towards a Basic Theory to Model MDE, 2004.

Mega start



Jean-Marie Favre, Towards a Basic Theory to Model MDE, 2004.

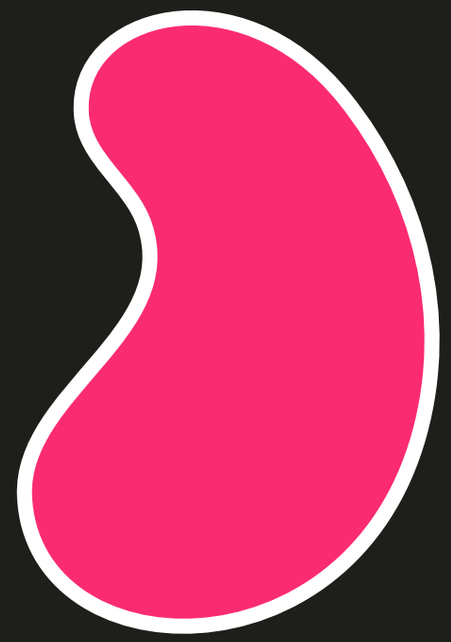
Focus


Source Metamodel
Transformation Model
Target Metamodel
μ
μ
μ
Source Language
Transformation Function
Target Language
X
X
ε
ε
ε
Source Model
Transformation Instance
Target Model

Problem 0:

Precision

Two languages

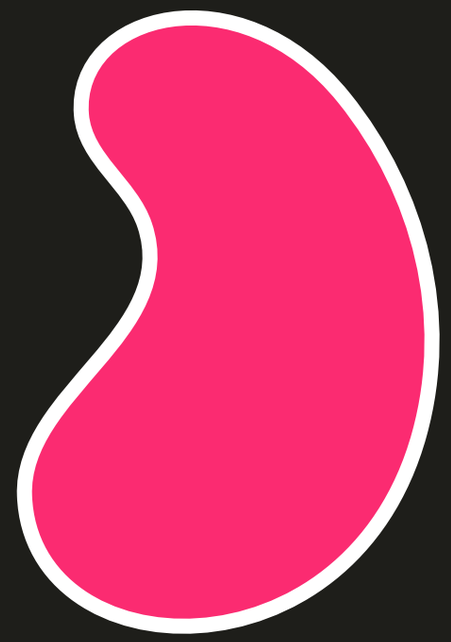


One language

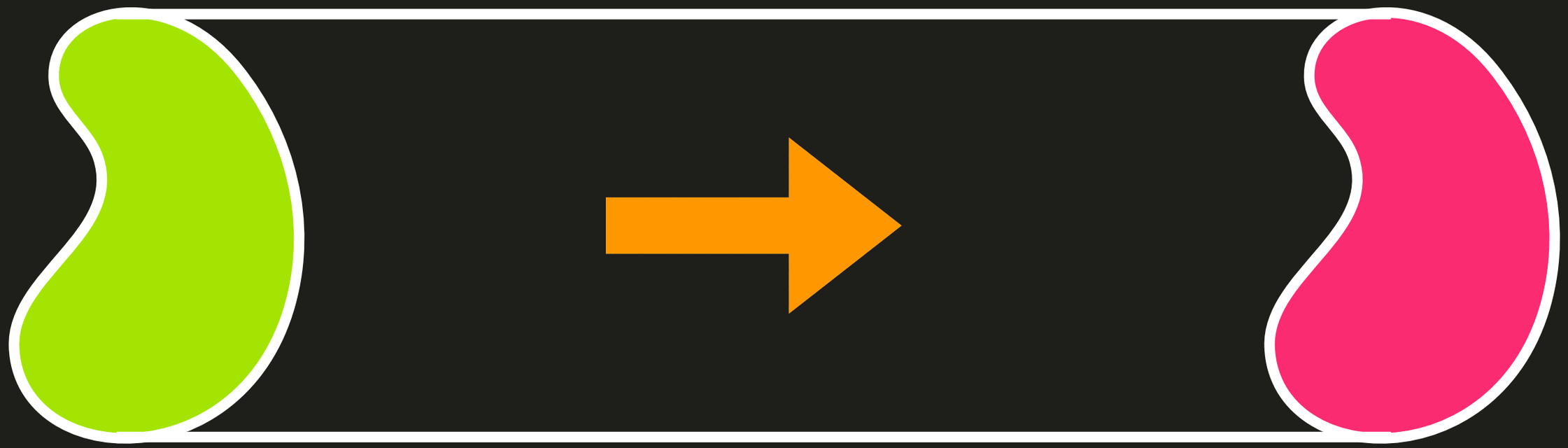
Tell me about your refactoring

- ✓ You accept
  - ✓ all of Java
  - ✓ nothing else
- ✓ Transform it
- ✓ Produce
  - ✓ nothing besides
  - ✓ use all features

Two languages



Assumed commitment



Partial applicability

Language subset

Conservative mapping

Liberal mapping

Robust mapping

Antirobust mapping

Fault recovery

Fault tolerance

Overtolerance

Shotgun effect

Shotgun

http://langsec.org/

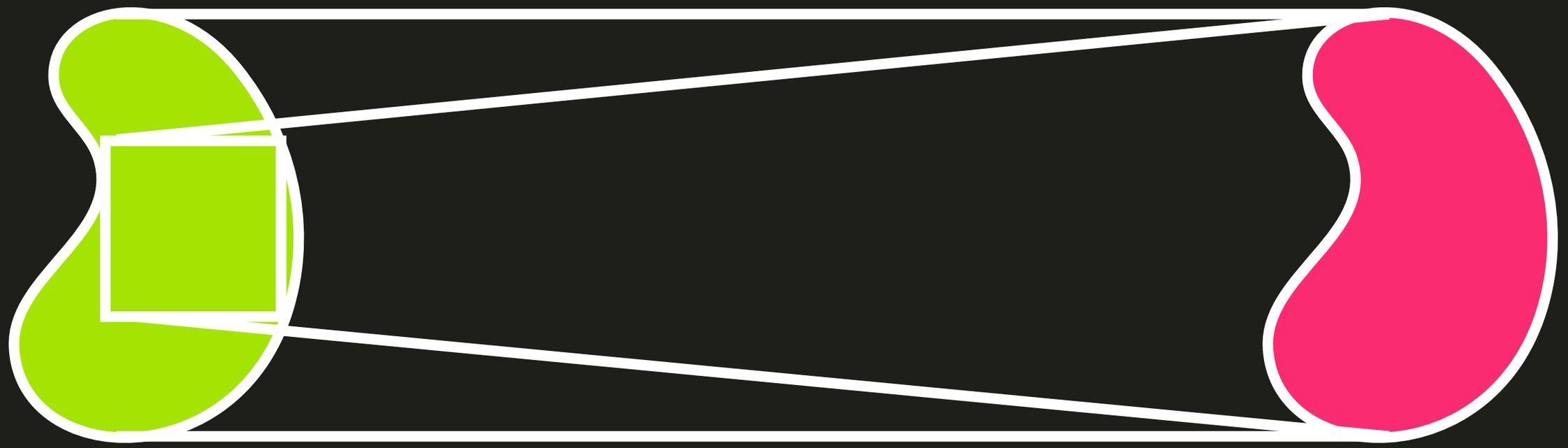
Problem I:

Application

Function extension

Function extension

Goal is clear



Function extension

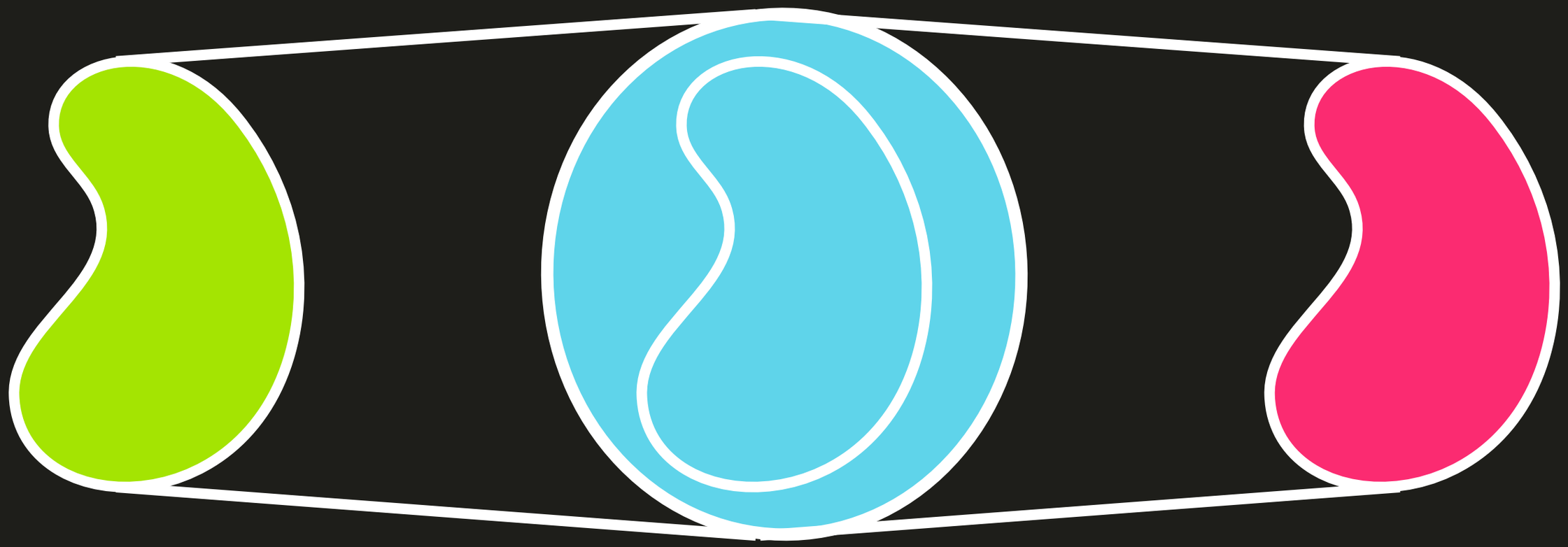
Function extension

Goal is unclear

Problem II:

Composition

Liberal + conservative = ?



might not be the same extension!

Streamliners

=

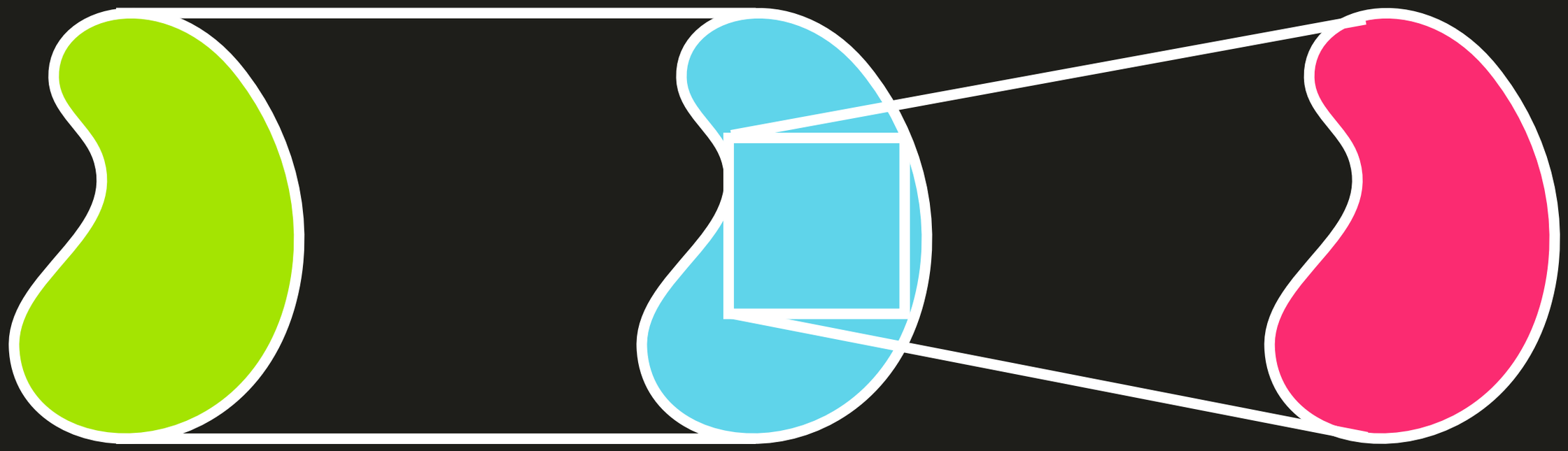
identity

Use of =

partial to complete applicability

canoniser

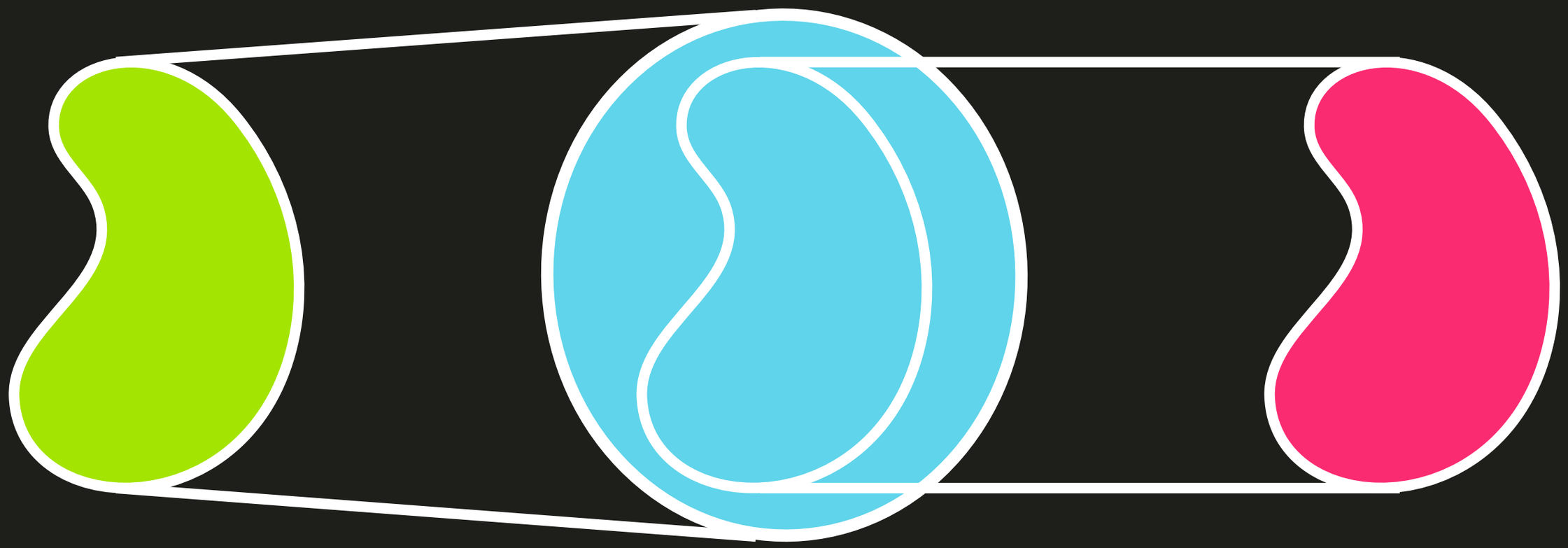
Use of ⨟



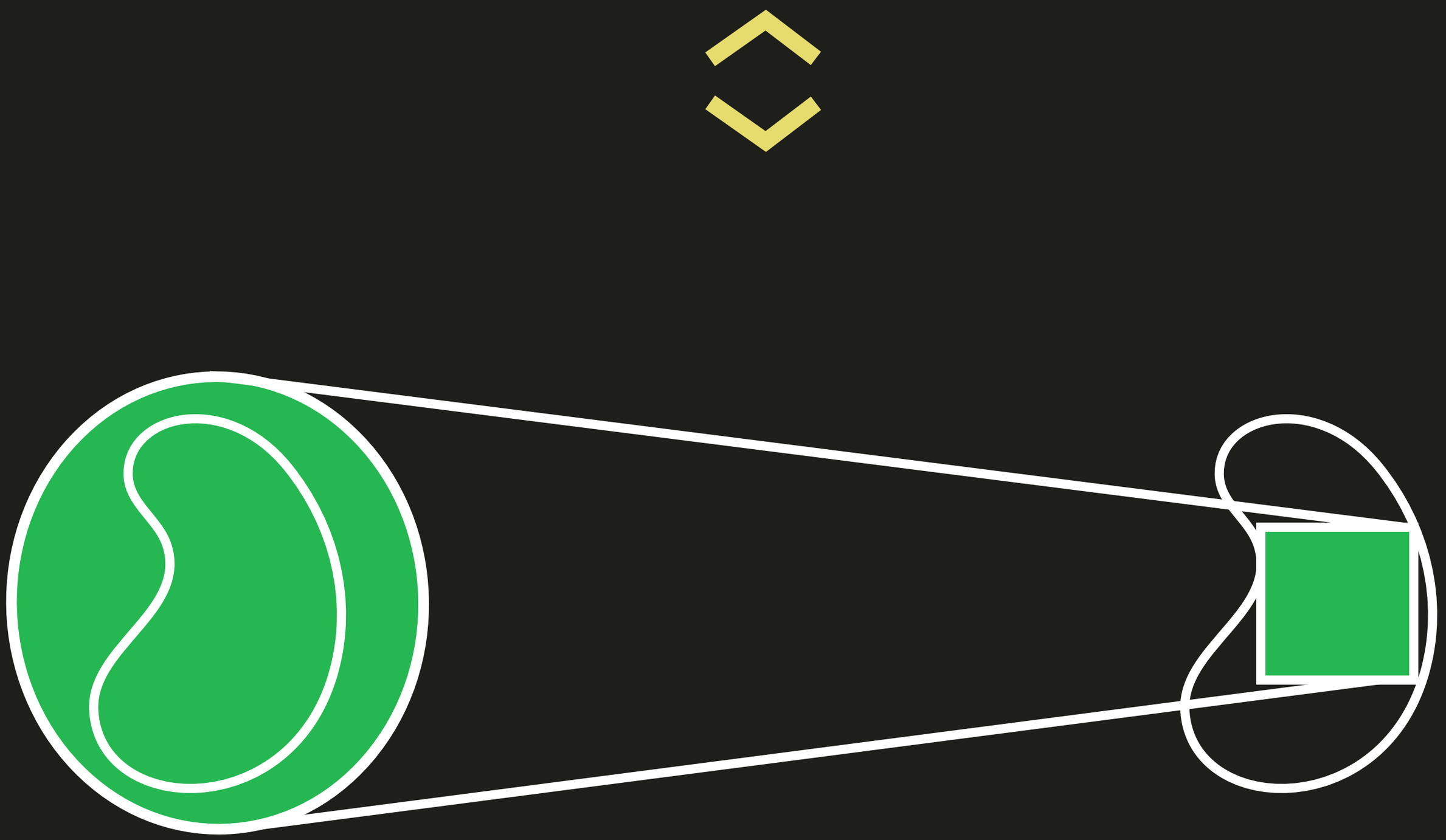
function composition

codifier

Use of ≏

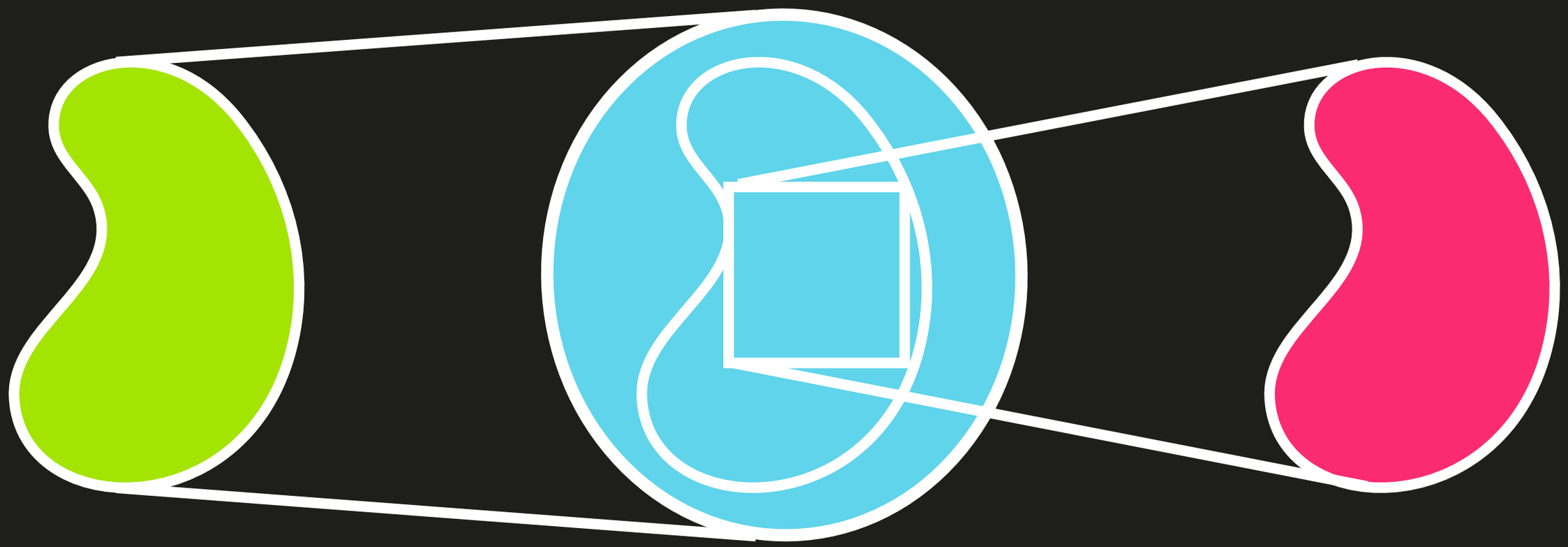


preventing the shotgun effect



normaliser

Use of <>



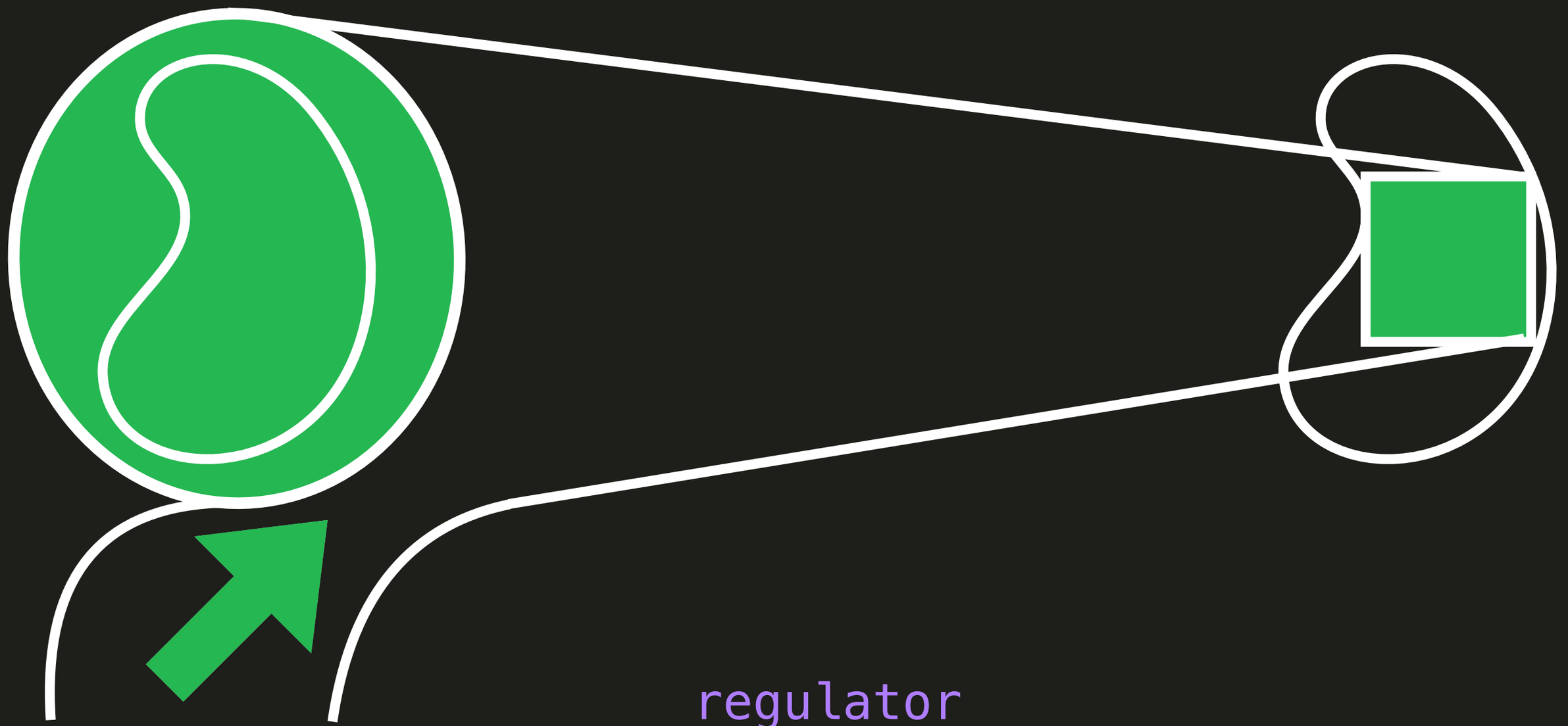
function composition

Problem III:

Calibration

12

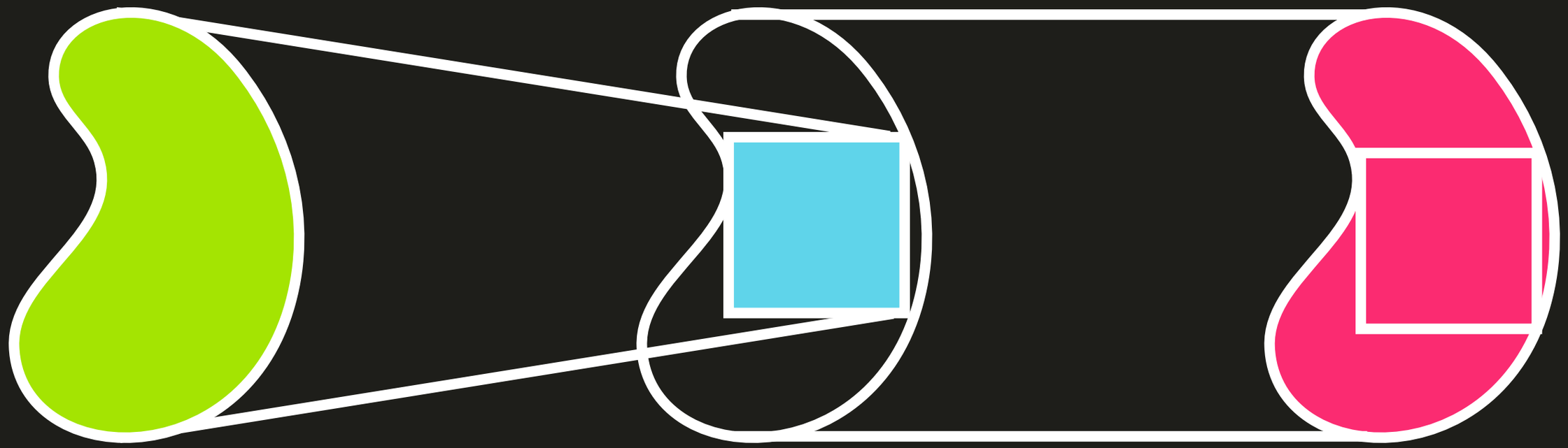
calibrator


regulator

Problem IV:

Overapproximation

$$(L \to \check{L}) \to (L \to L) \to (L \to L)$$



we might get a subset in the end

id || (Ľ→Ľ) is (L→L)

the only case where streamliners do not help

Conclusion

- ✓ Flexible commitments everywhere
- ✓ Can be considered precisely
- ✓ Mapping extension is not trivial
- ✓ Composition with streamliners
- ✓ Calibration is still not trivial
- ✓ Occasional overapproximation
- ✓ Demo at 16:15!
- ✓ Questions?