GEFÖRDERT DURCH DIE DFG Deutsche Forschungsgemeinschaft

Karlsruhe Institute of Technology

Test, Validation, Analysis

CONVIDE CRC 1608

Combining Variability and Consistency

Dirk Neumann • 25.03.2026

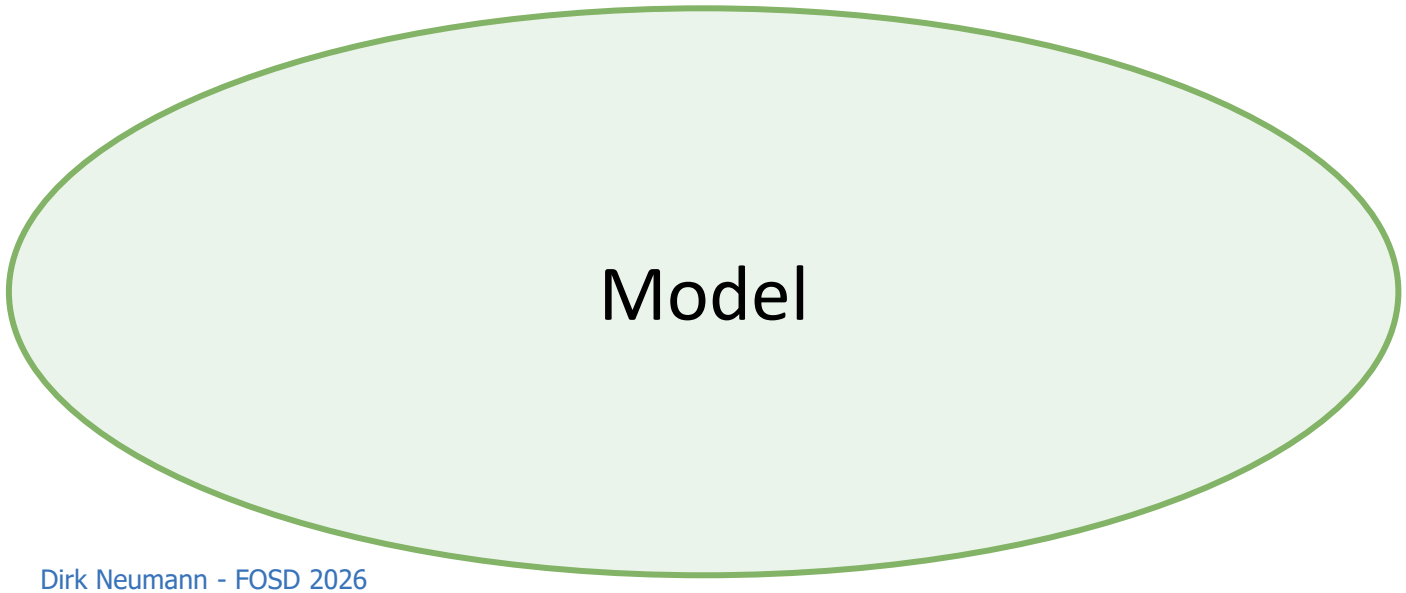
Questions

How to combine variability and consistency?

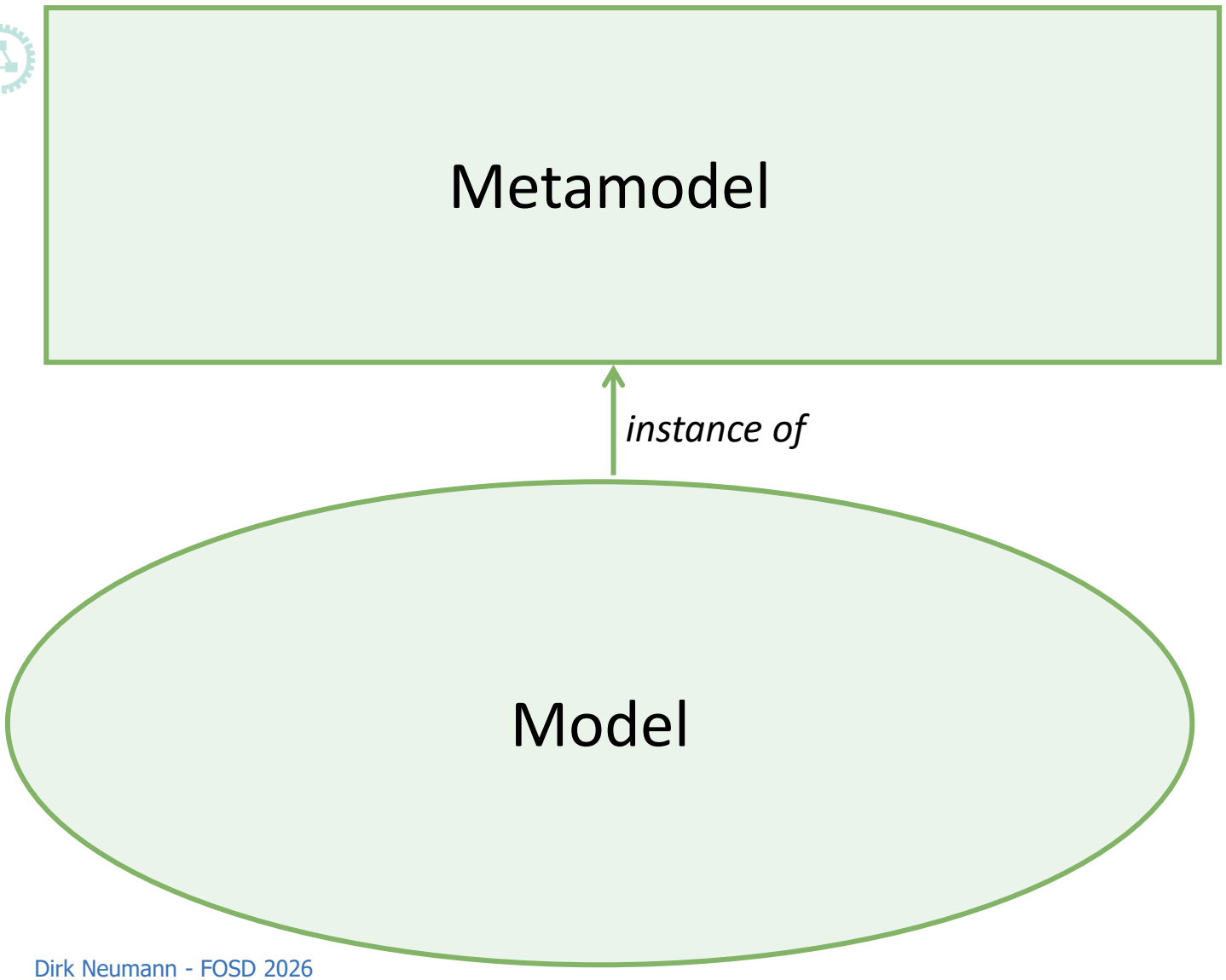
How to work with a possible combination?

How can consistency in variable systems be categorized?

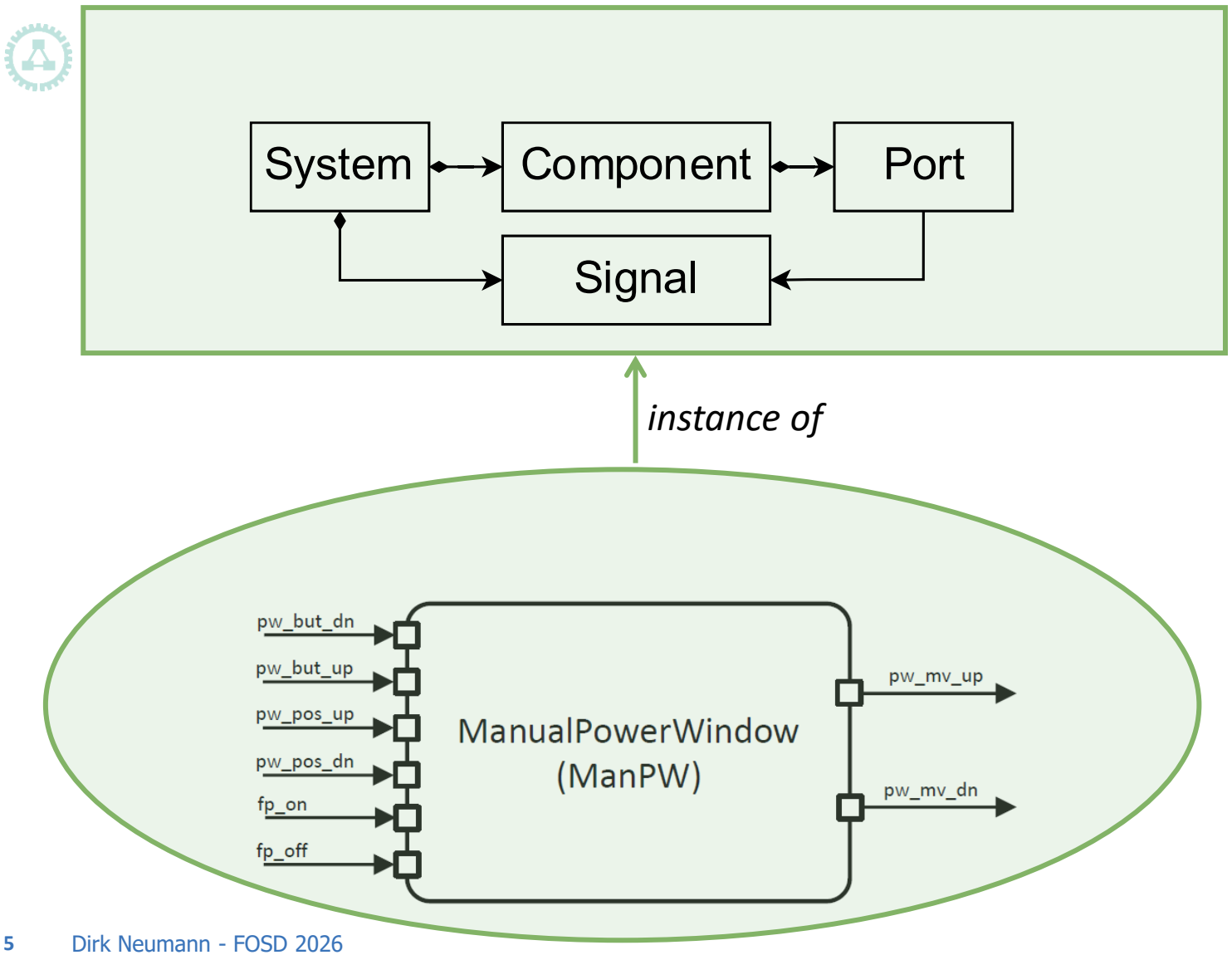
Model-Driven (Software) Engineering



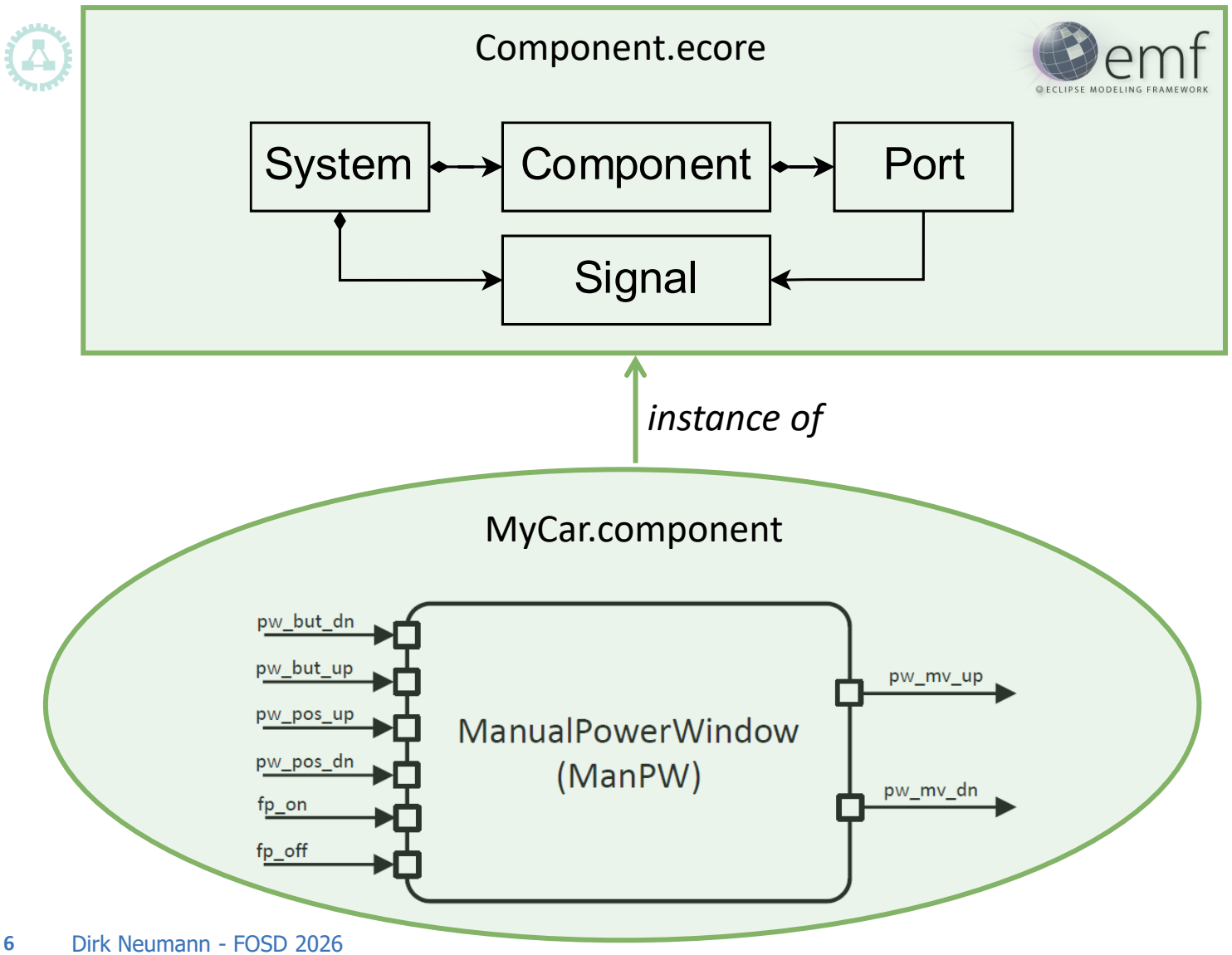
Model-Driven Engineering



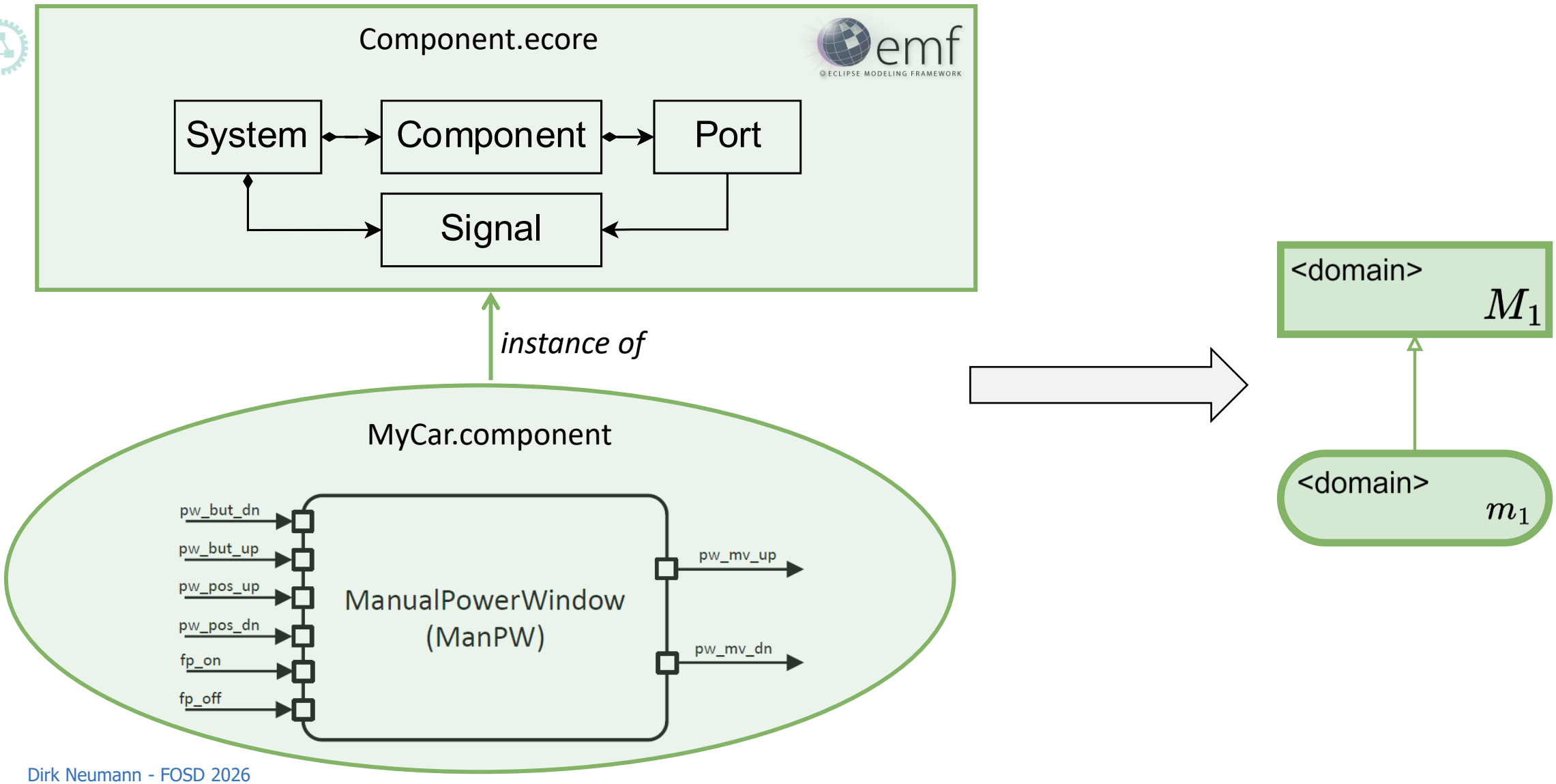
Model-Driven Engineering



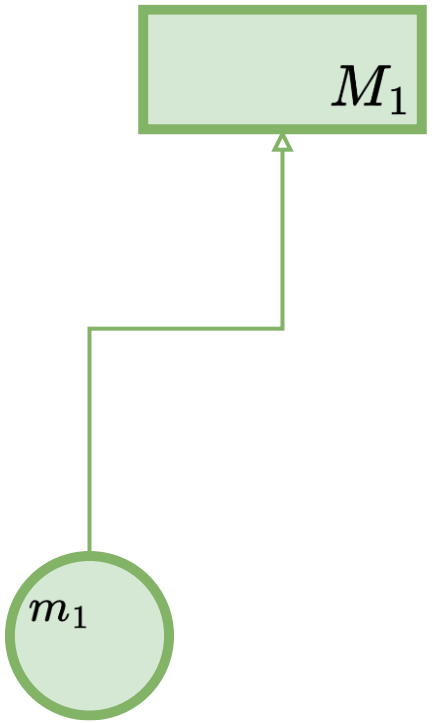
Model-Driven Engineering



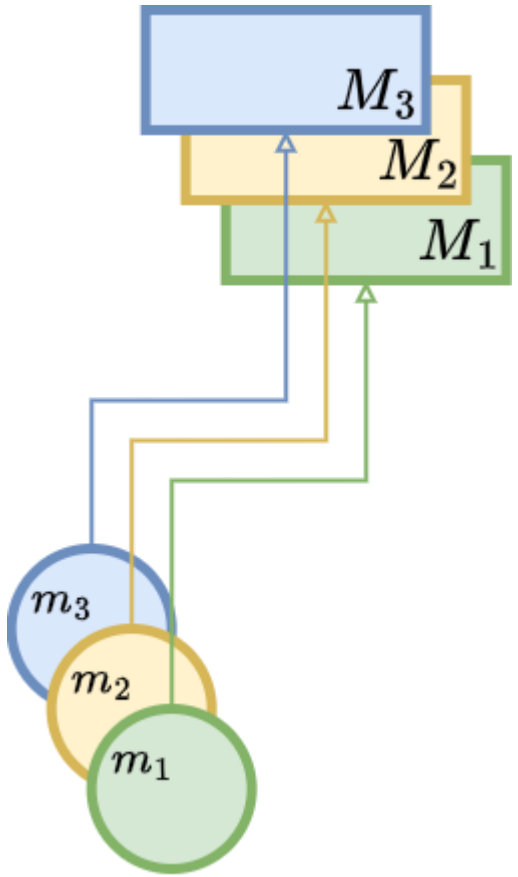
Model-Driven Engineering



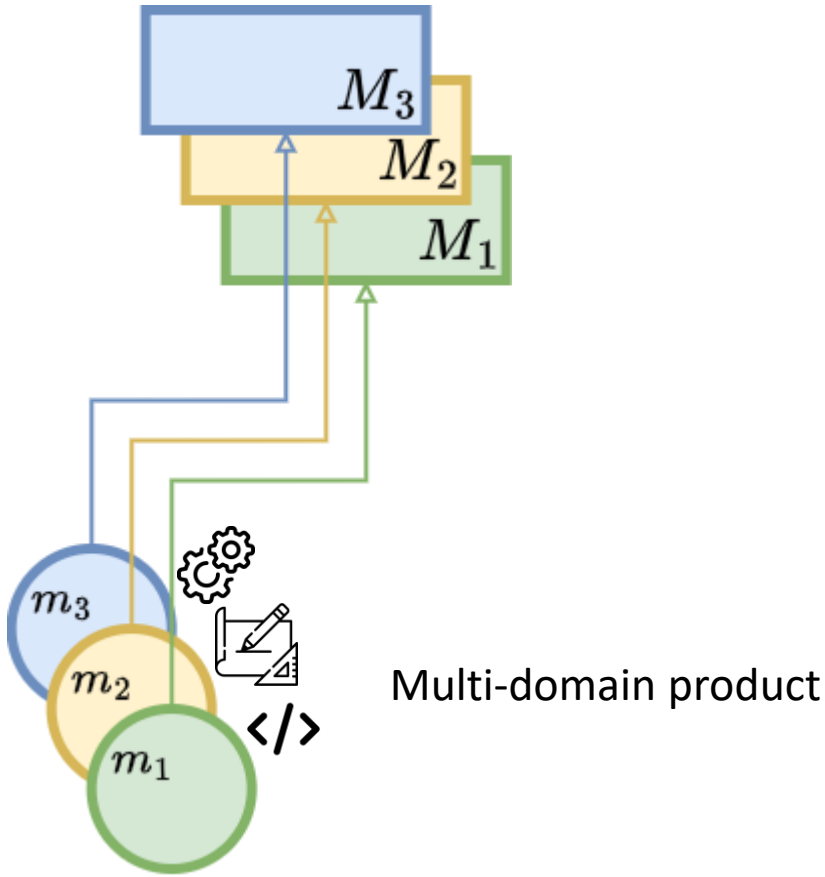
Multiple Domains



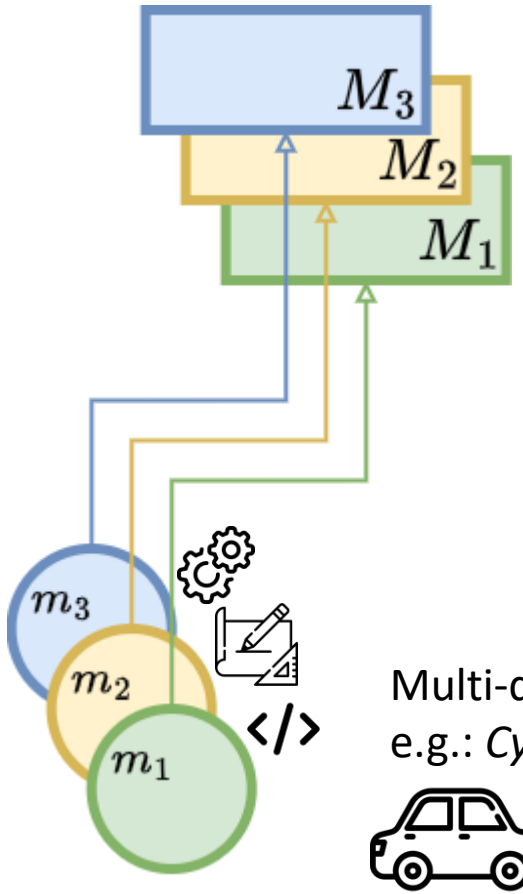
Multiple Domains



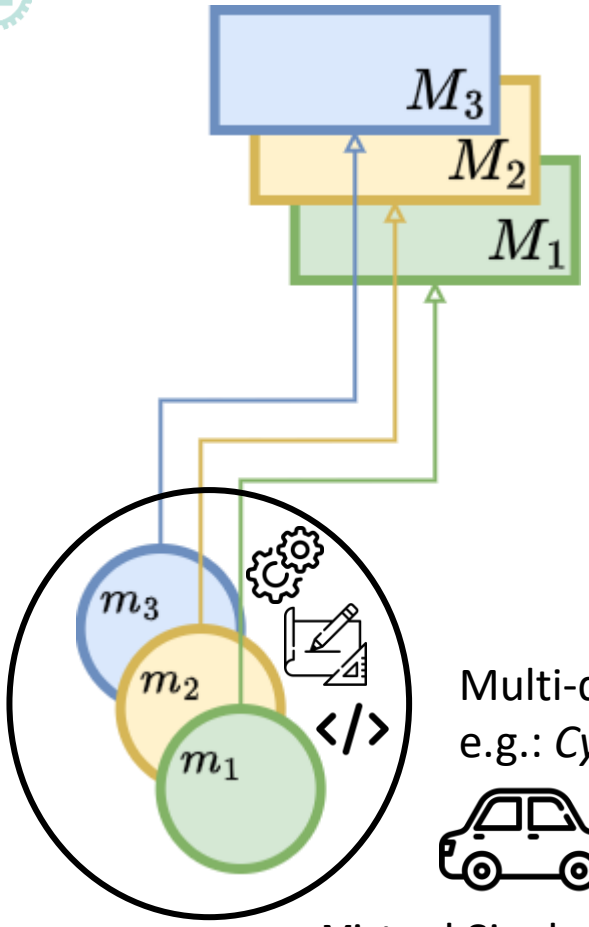
Multiple Domains



Multiple Domains

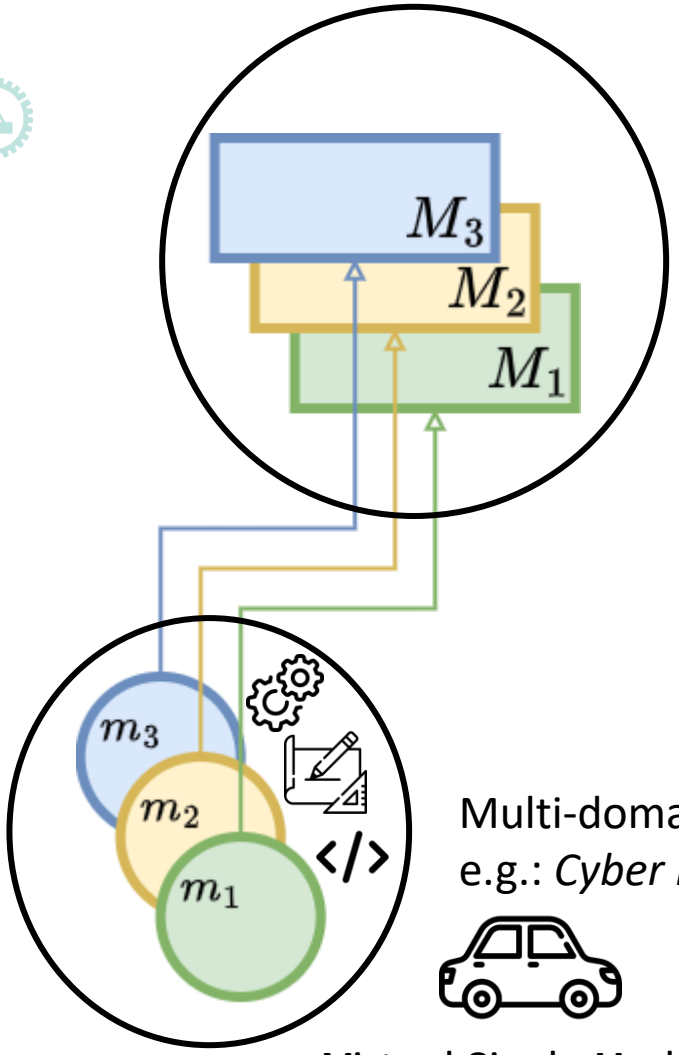


Multiple Domains



Virtual **S**ingle **U**nderlying **M**odel (V-SUM)

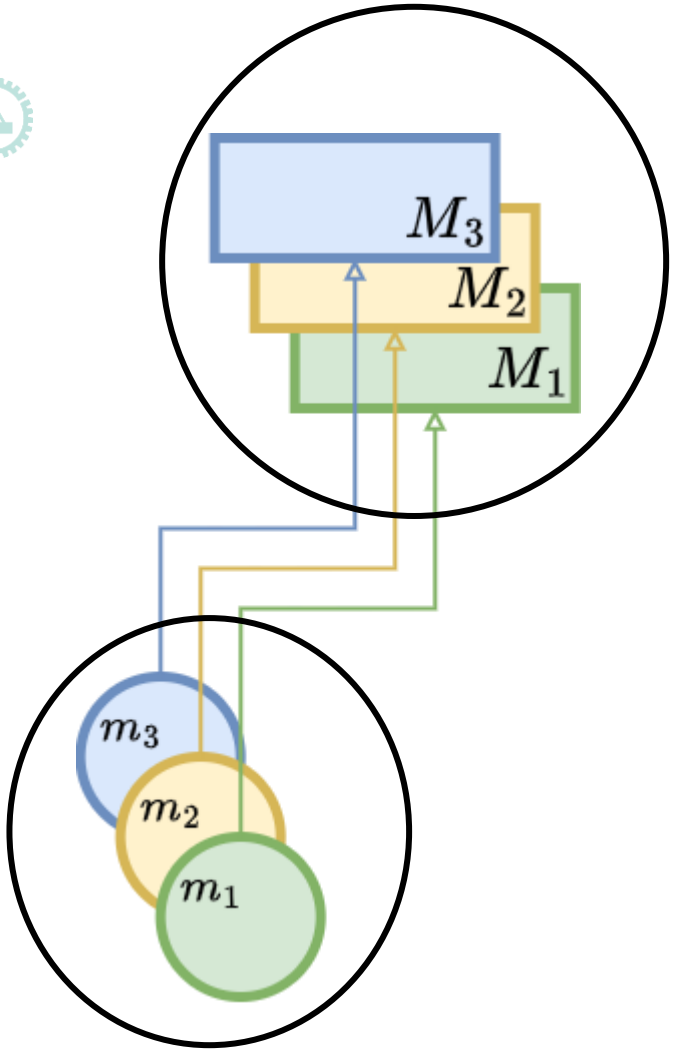
Multiple Domains



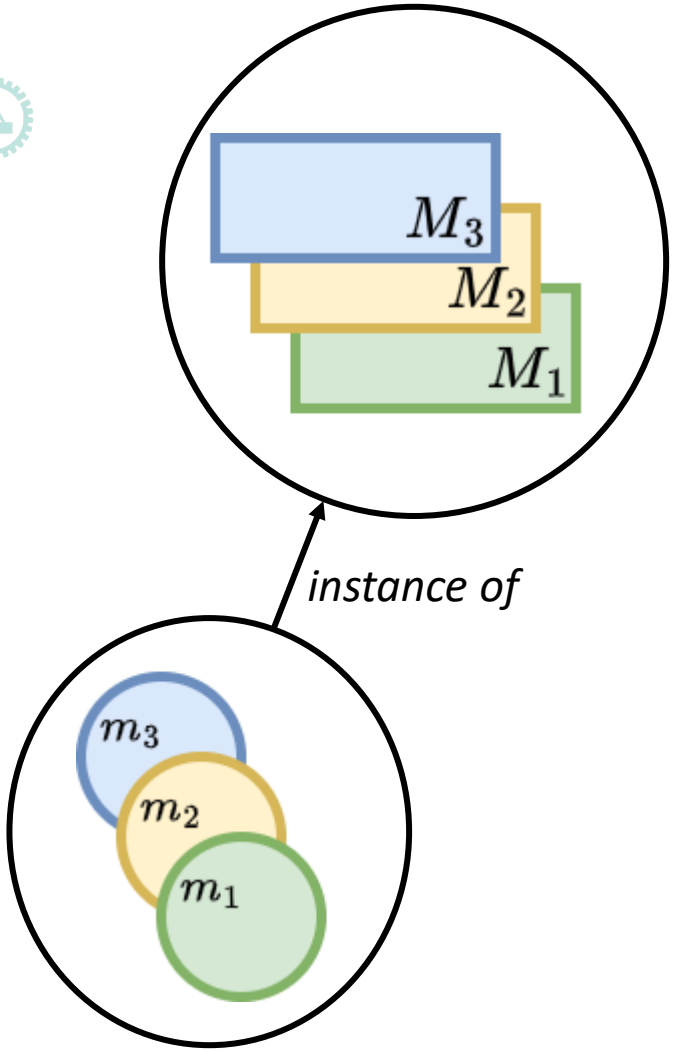
Virtual **S**ingle **U**nderlying **M**eta**m**odel (V-SUMM)

Virtual **S**ingle **U**nderlying **M**odel (V-SUM)

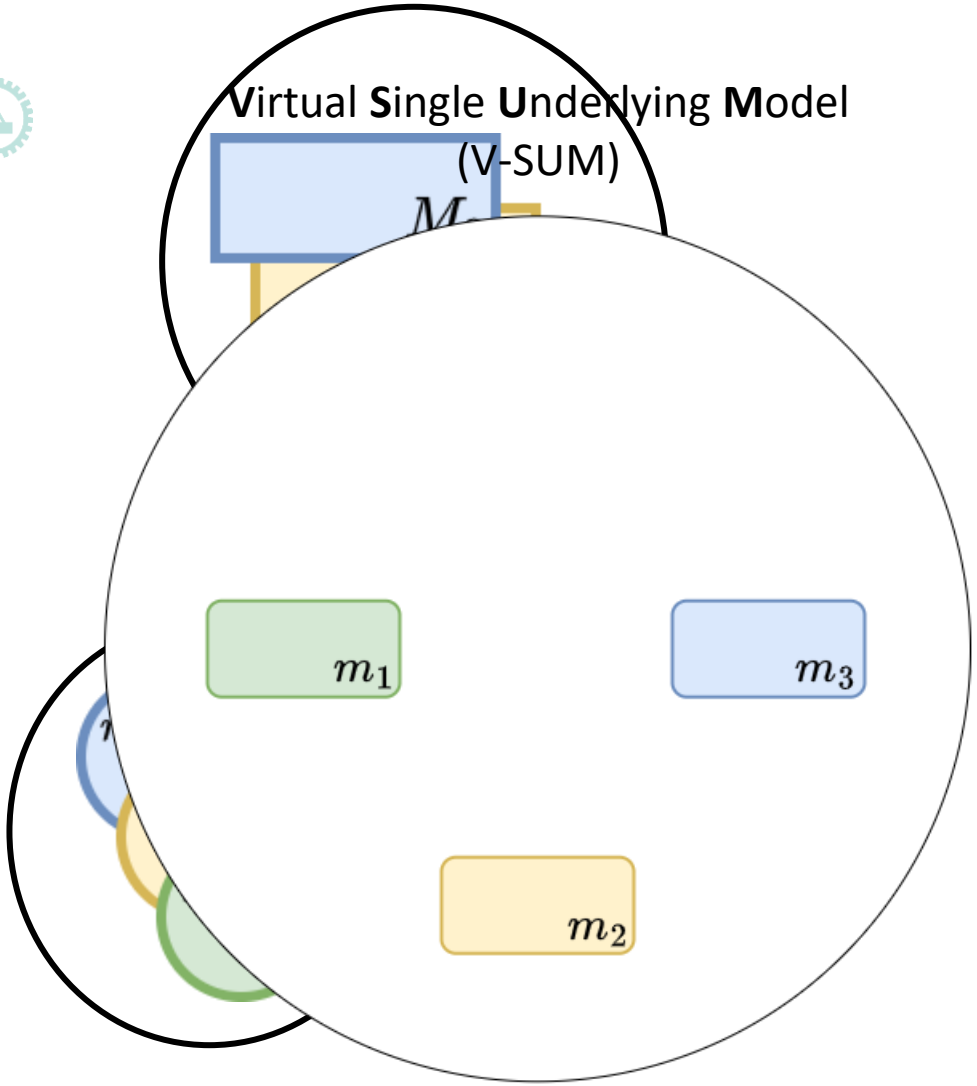
Multiple Domains

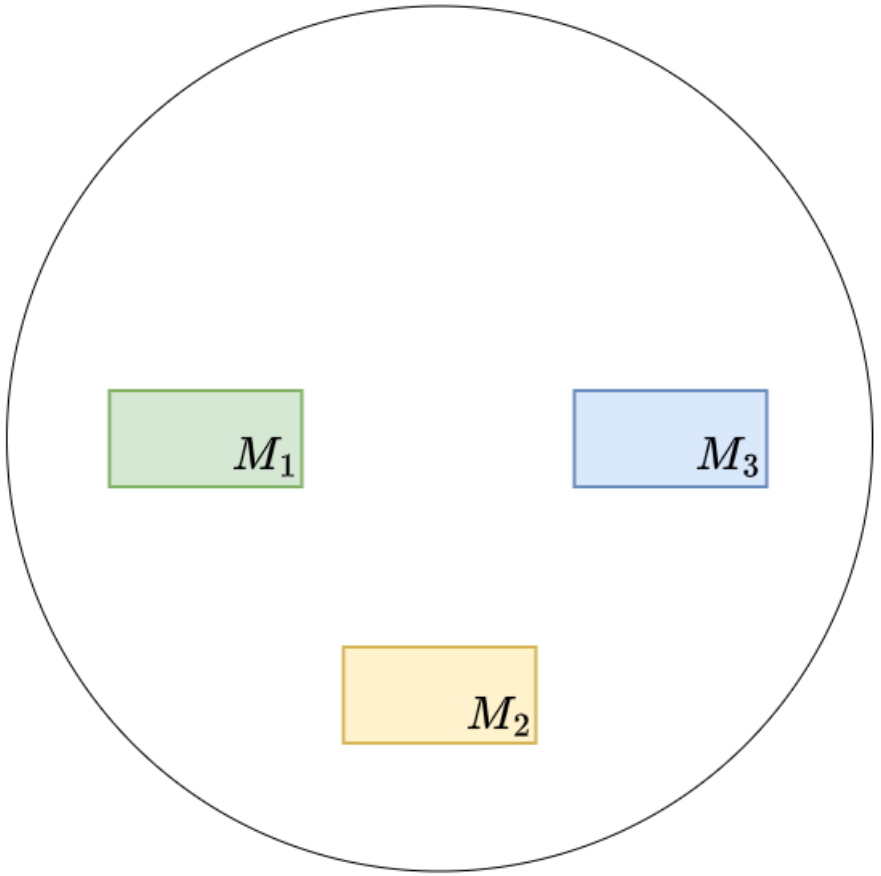


Multiple Domains



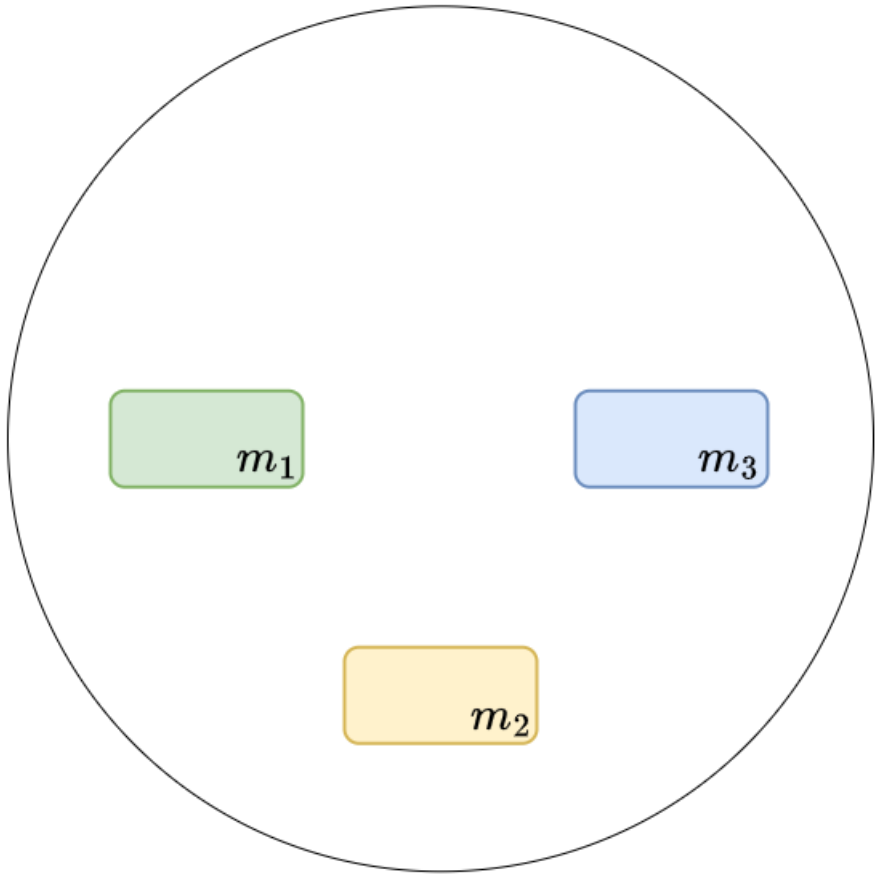
Multiple Domains



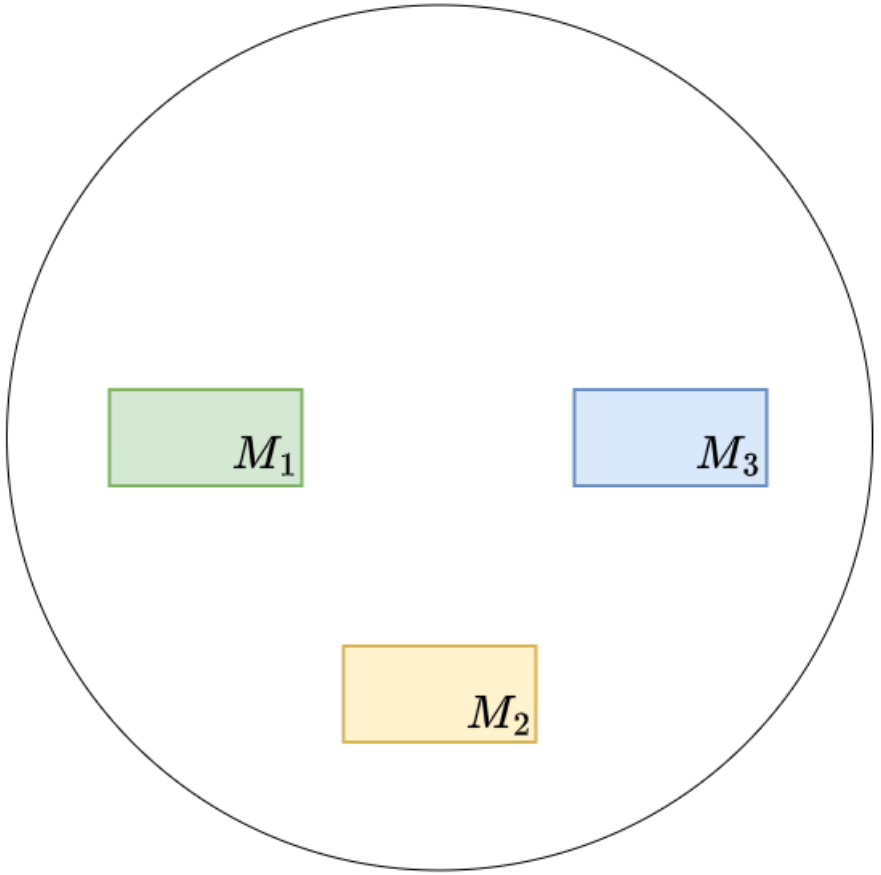


Part 1: Consistency

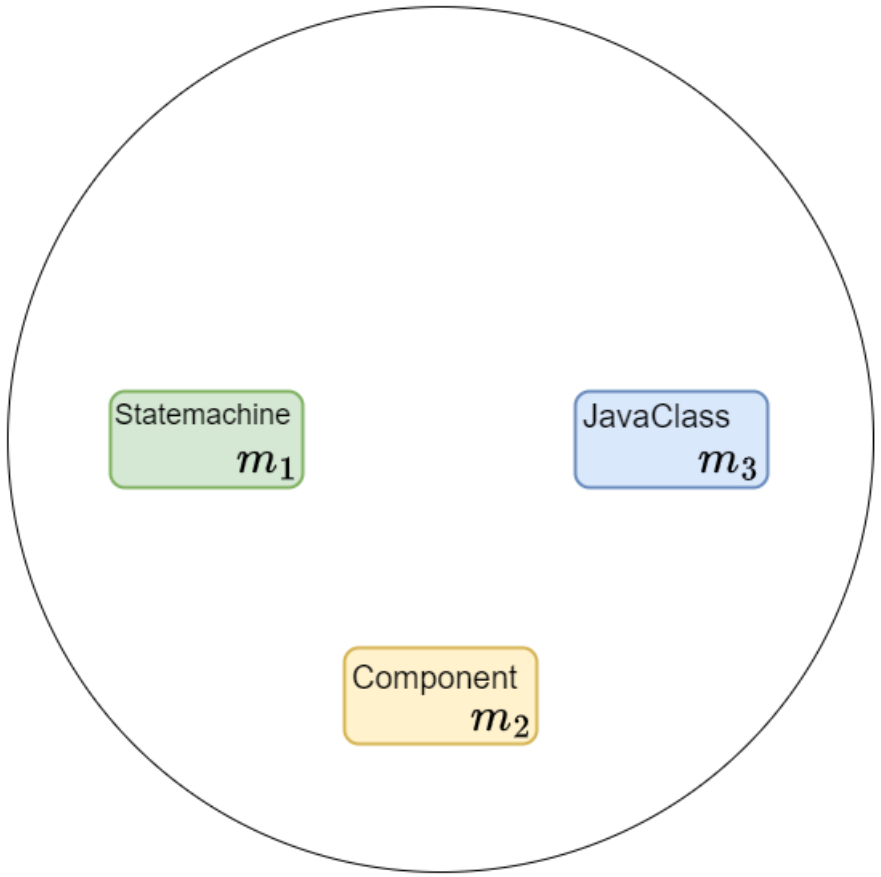
Virtual **S**ingle **U**nderlying **M**odel
(V-SUM)

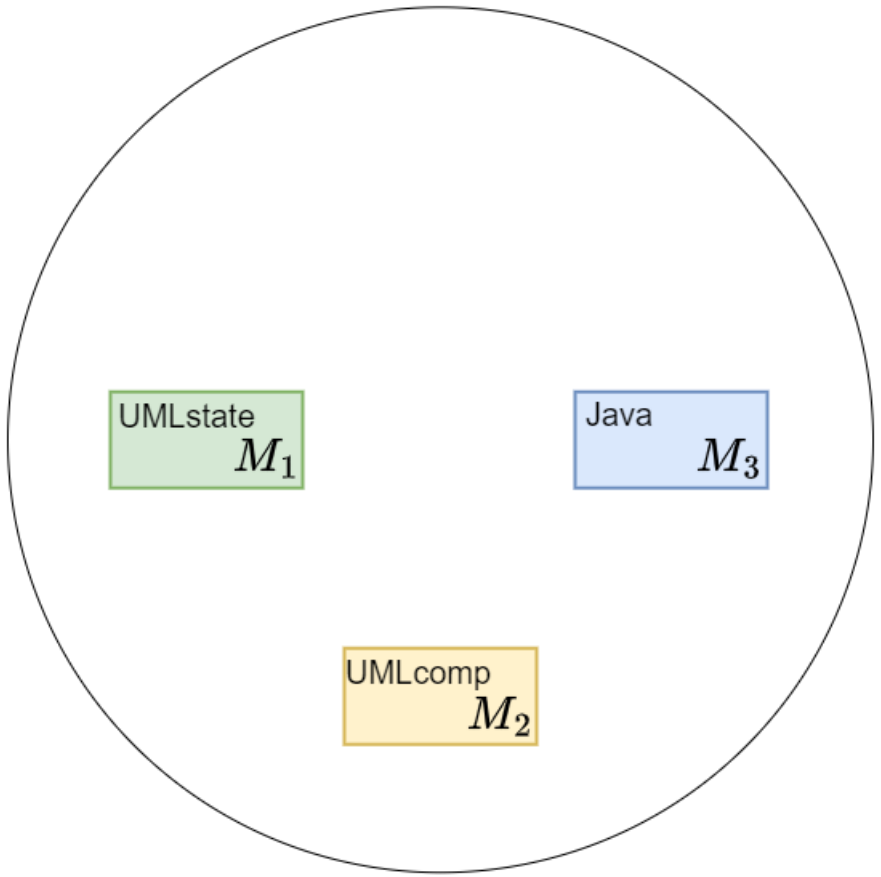


Virtual **S**ingle **U**nderlying **M**eta**m**odel
(V-SUMM)



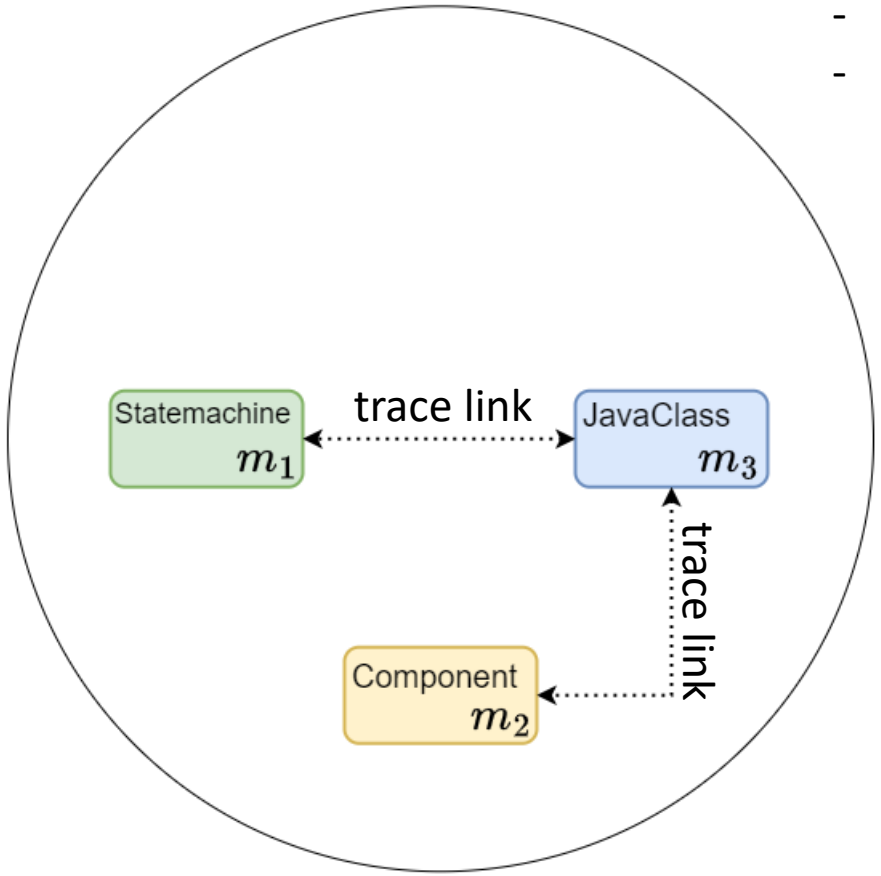
Part 1: Consistency





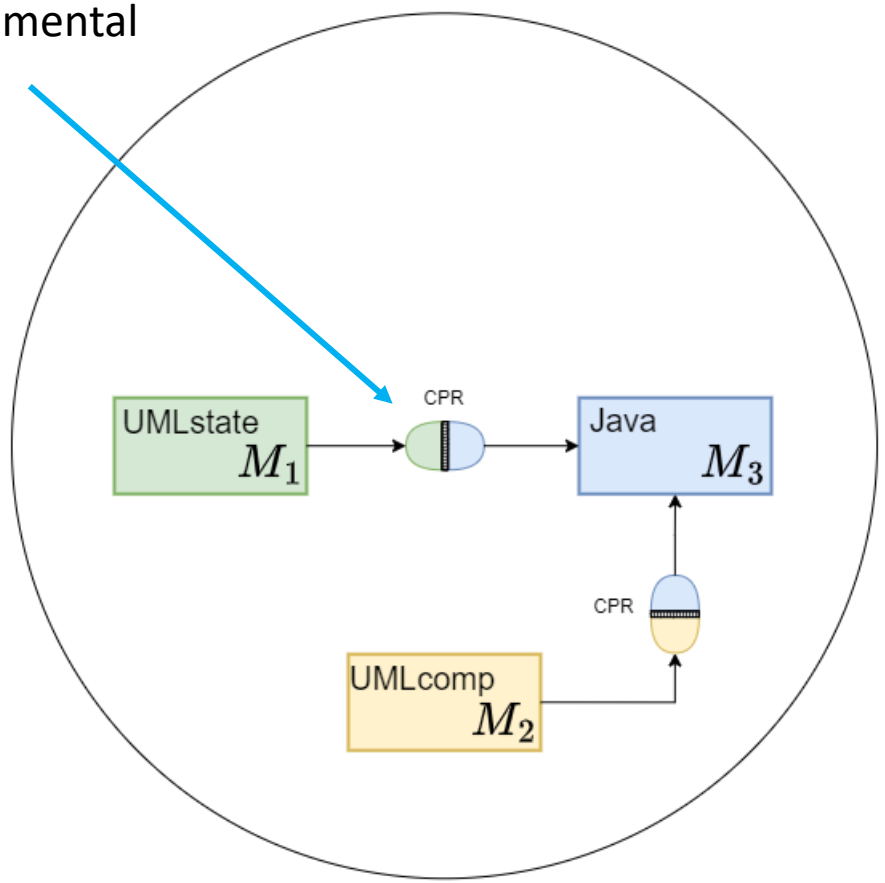
Part 1: Consistency

V-SUM

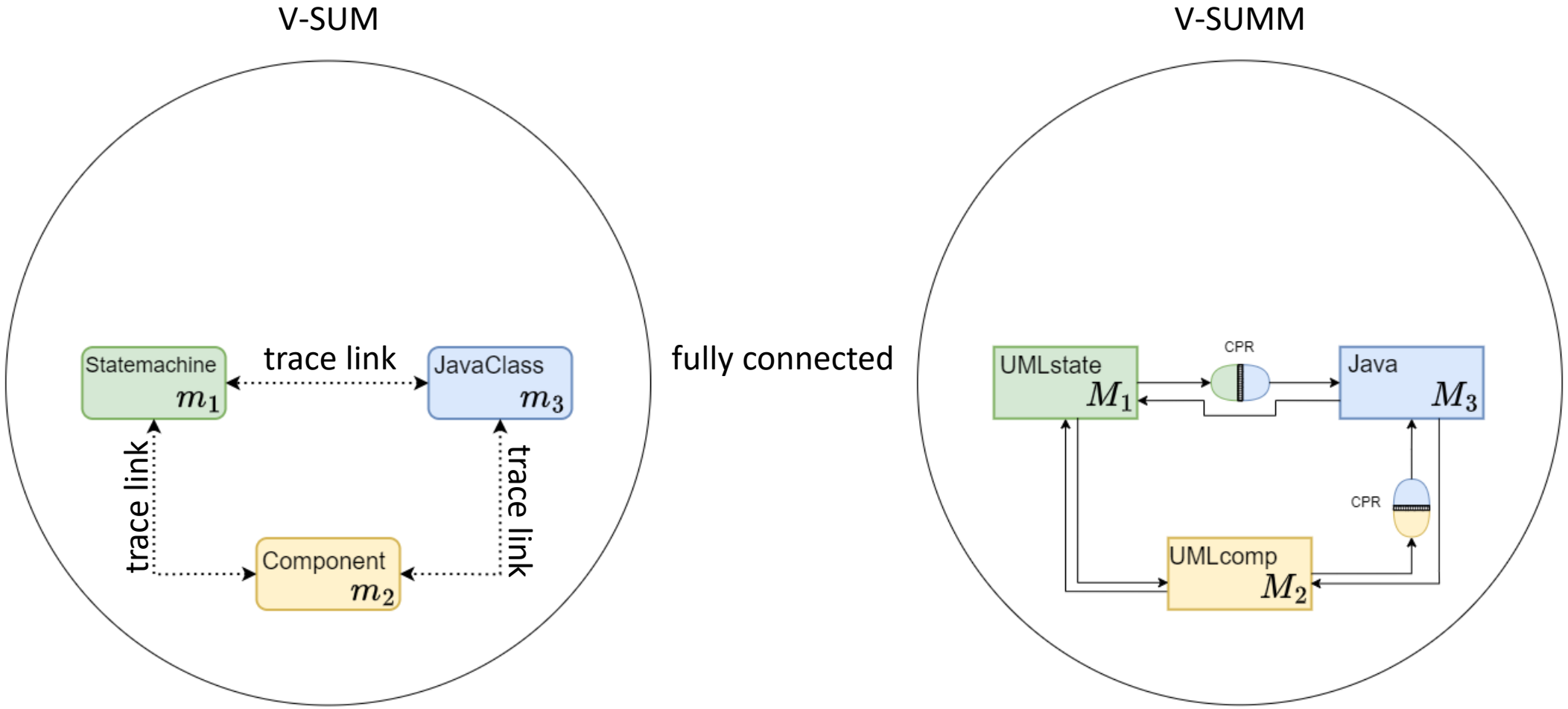


Consistency Preservation Rules
- model transformations
- Imperative, incremental
- Uni-directional

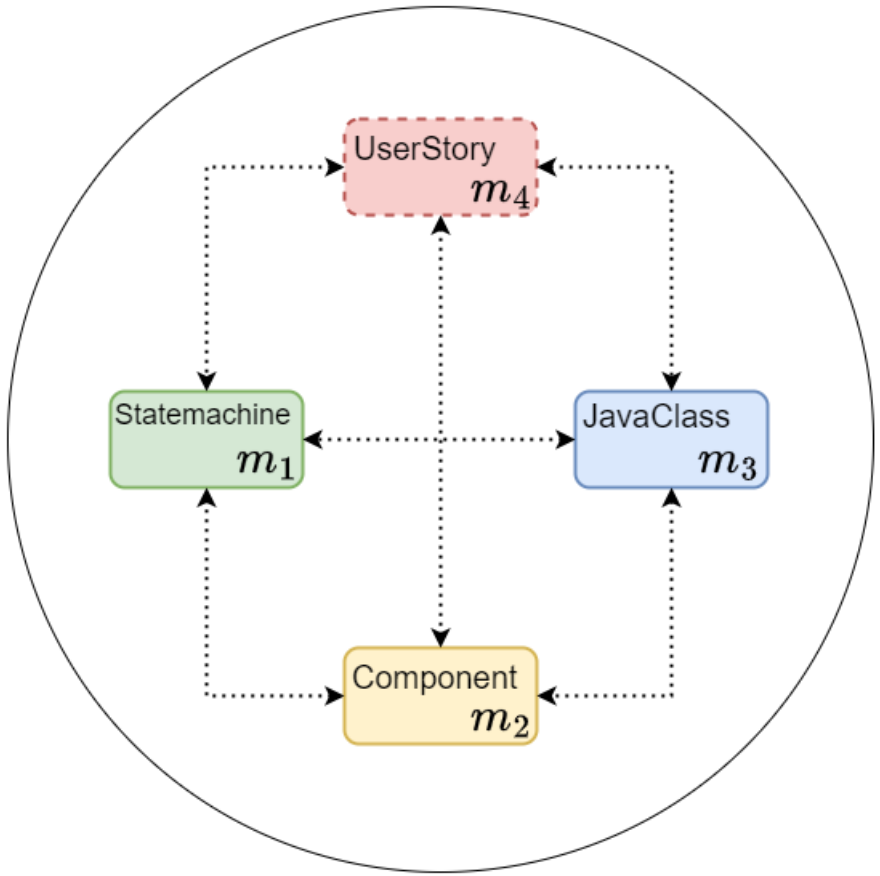
V-SUMM

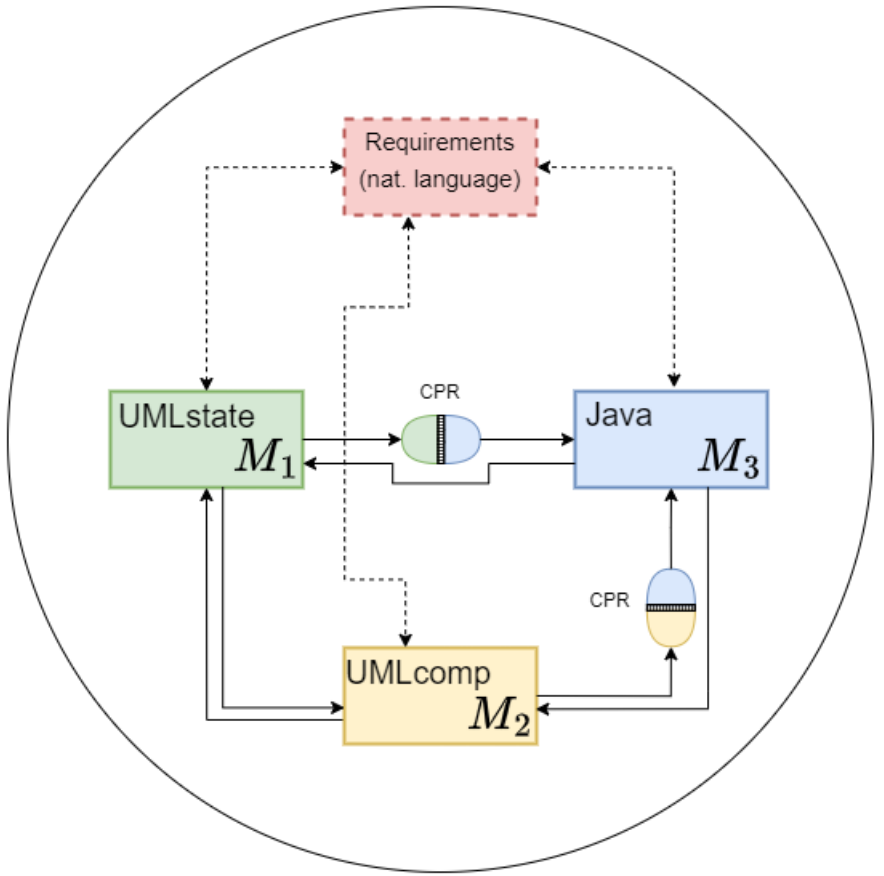


Part 1: Consistency

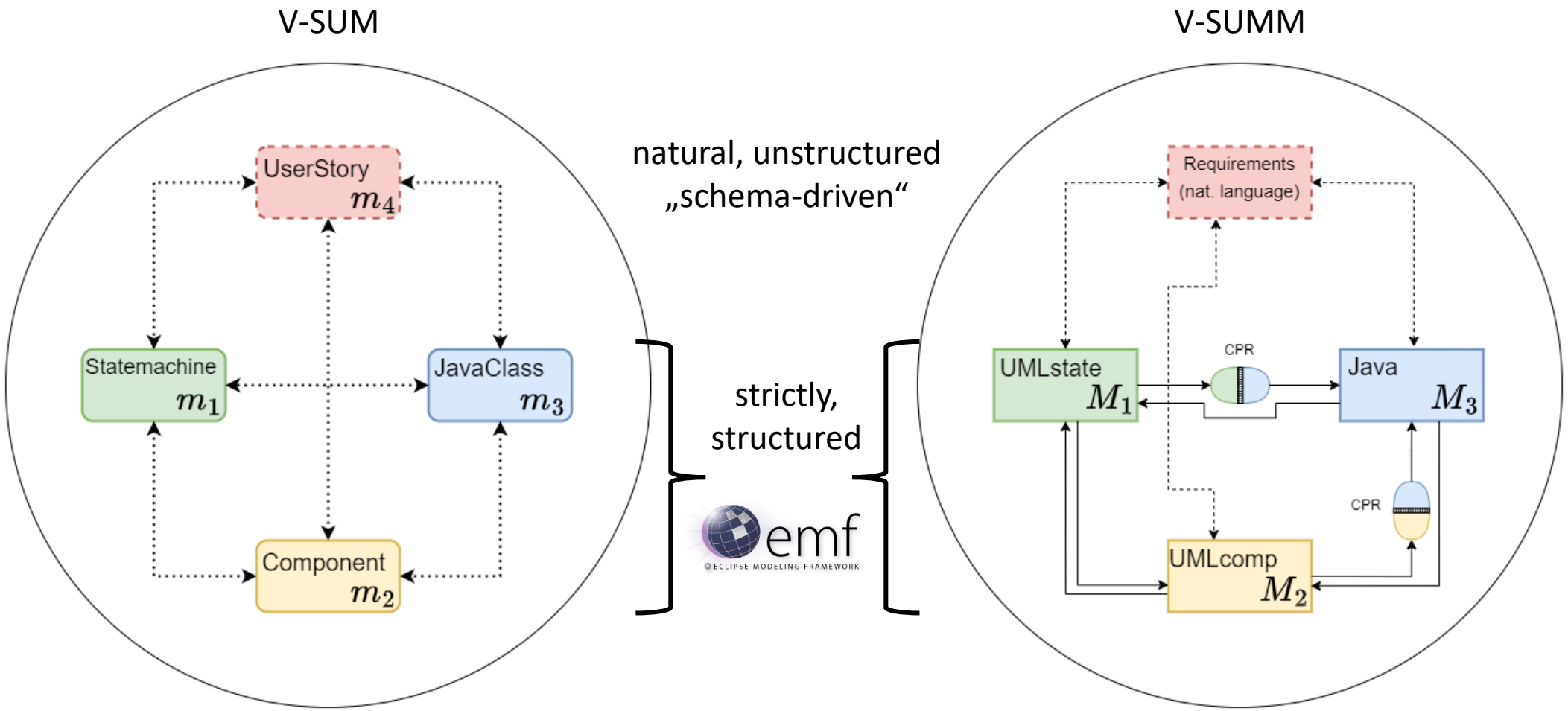


Part 1: Consistency



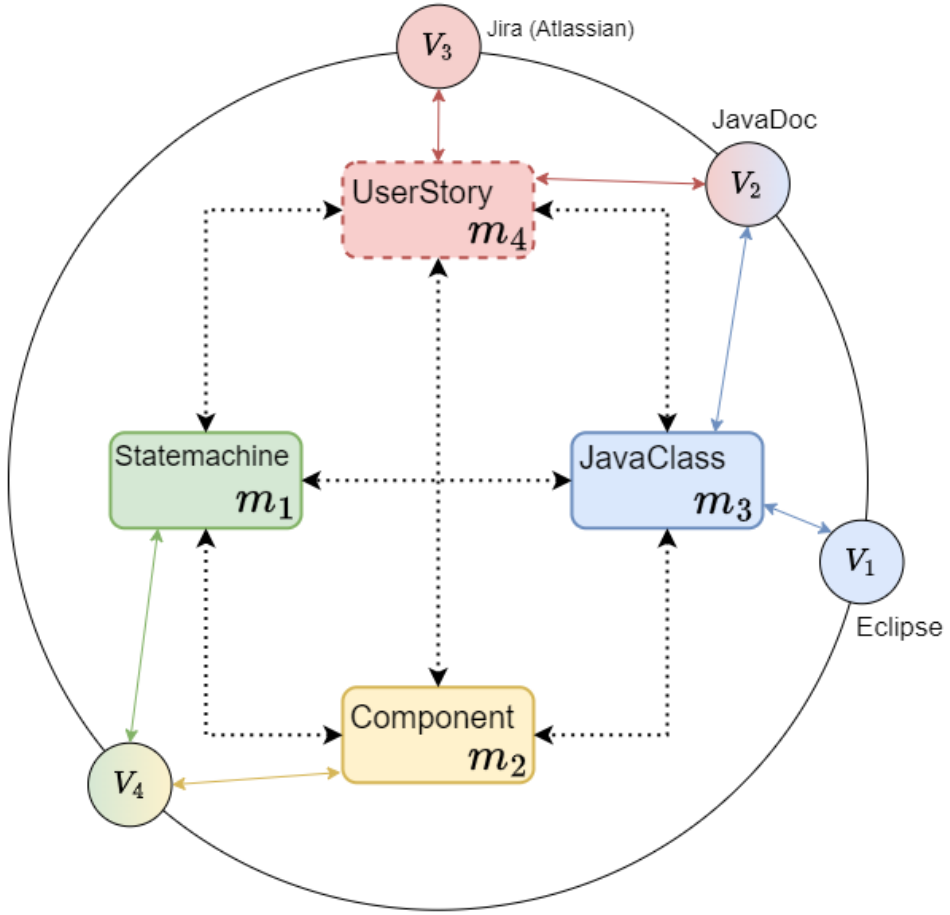


Part 1: Consistency

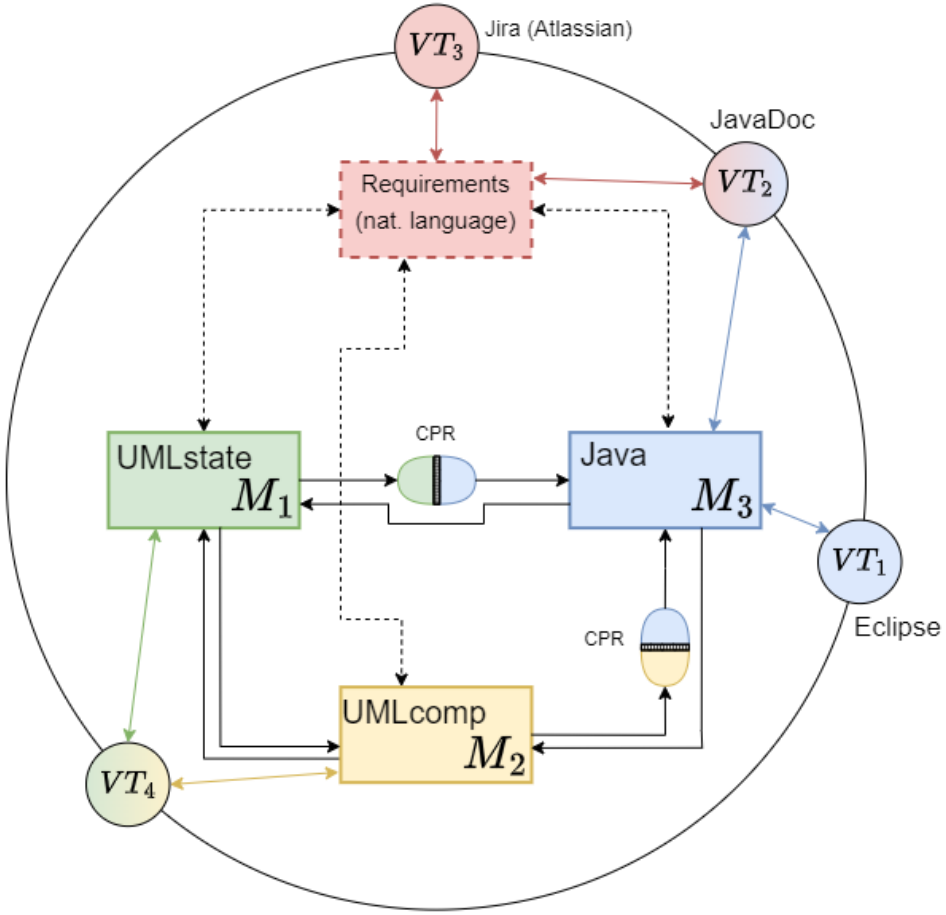


Part 1: Consistency

V-**Single**-UM



V-**Single**-UMM

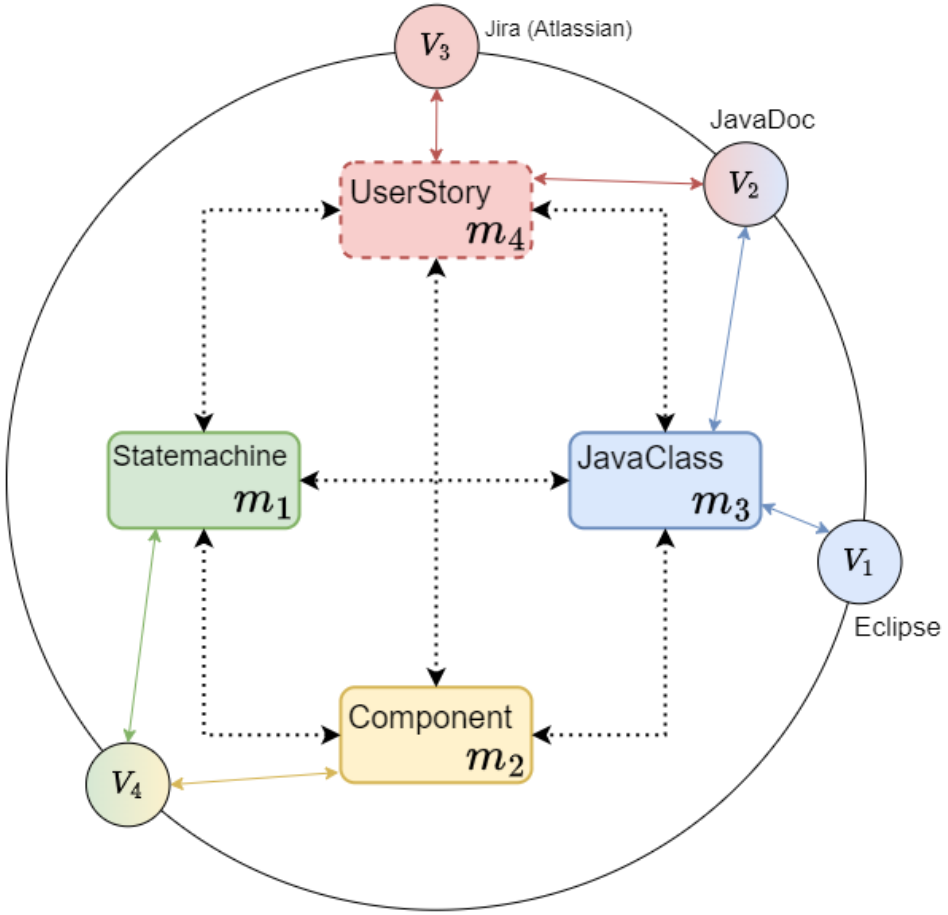


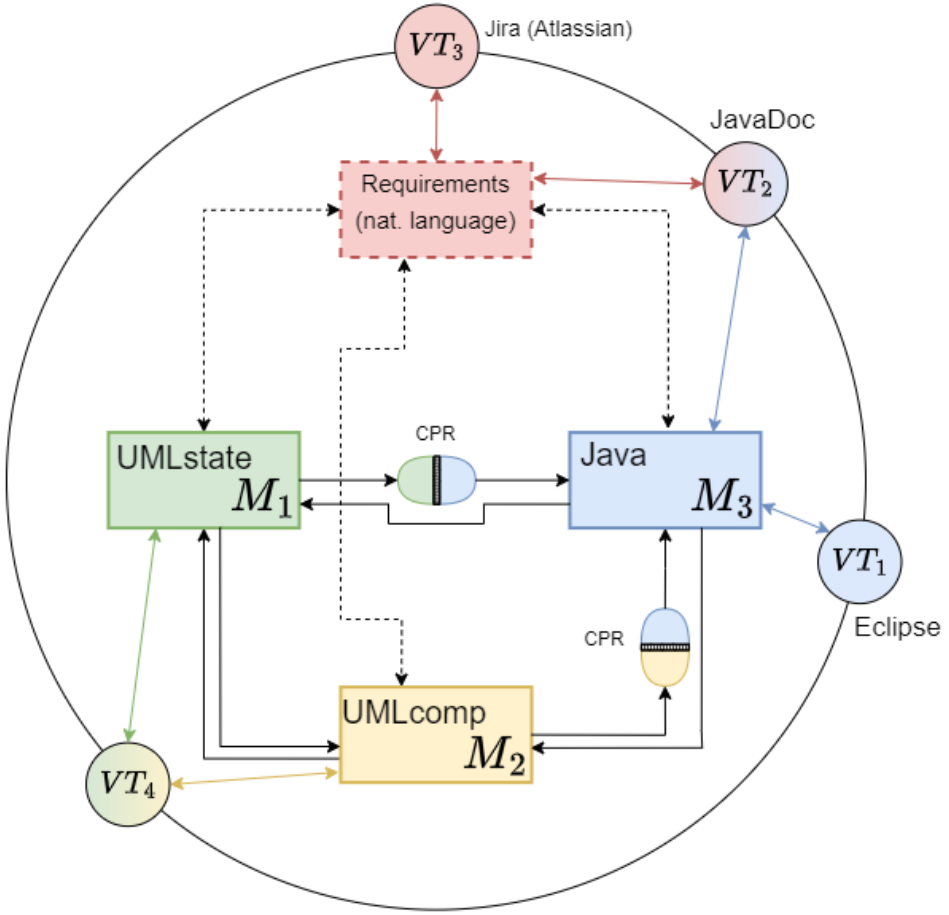
Part 1: Consistency

V-**Single**-UM

V-**Single**-UMM

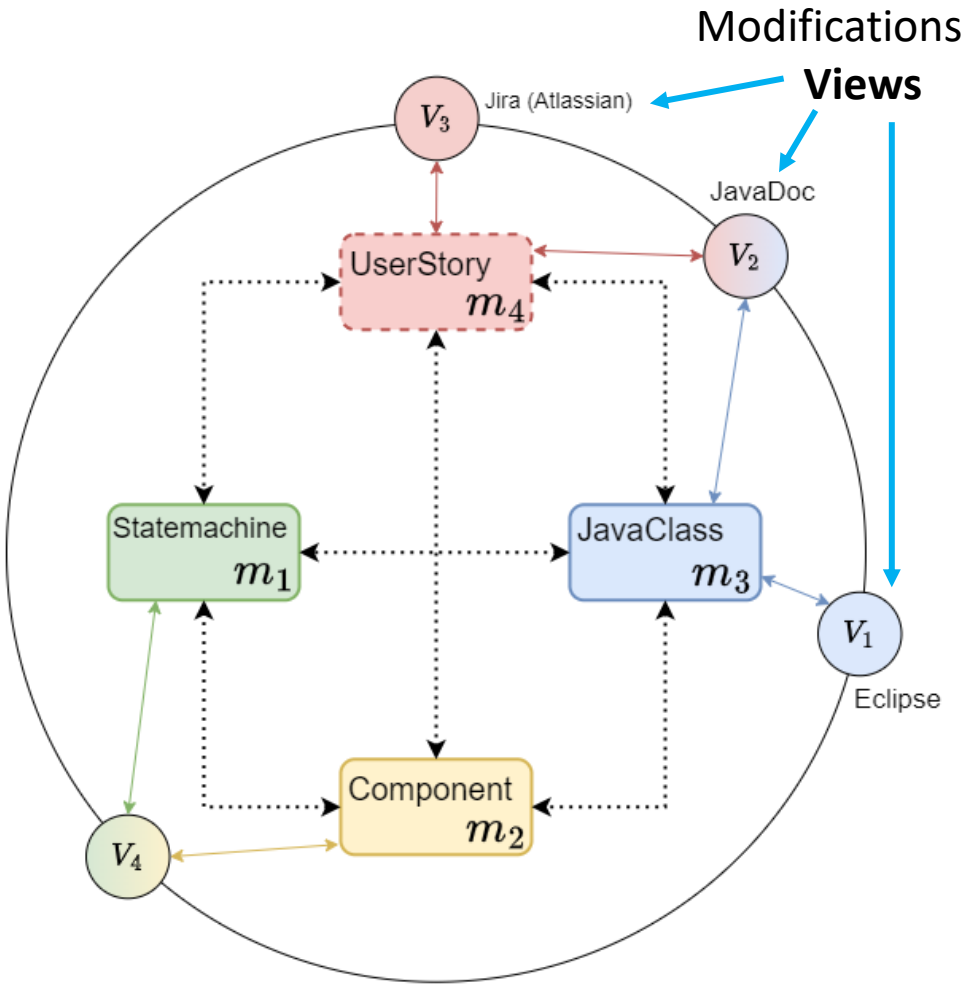
We do not want everybody to always work on the complete big picture.



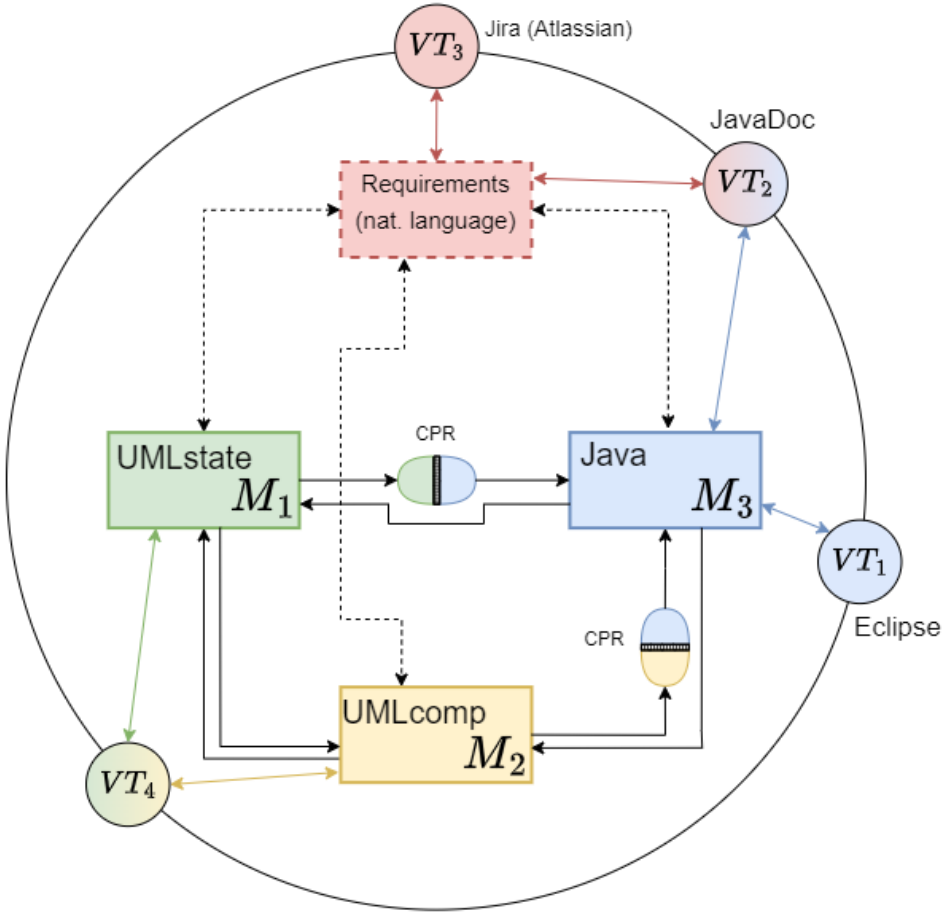


Part 1: Consistency

V-**Single**-UM



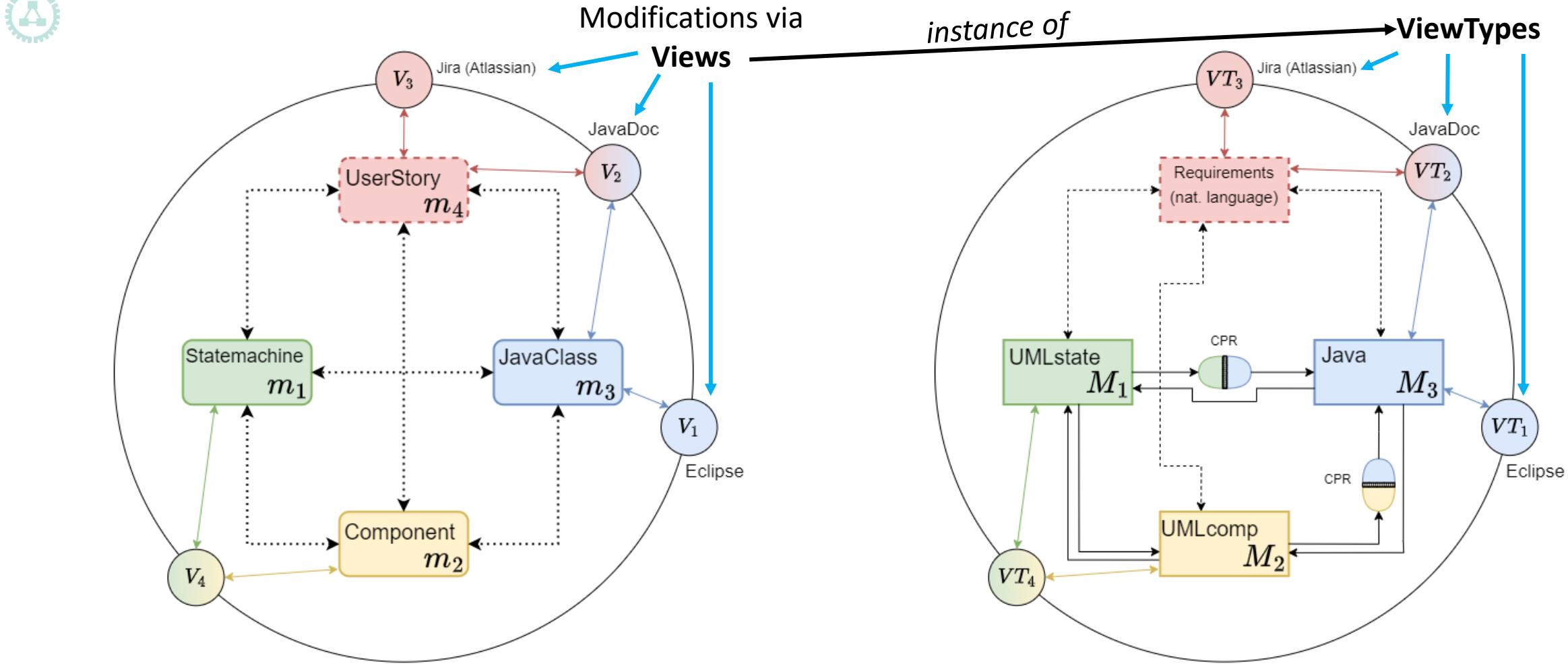
V-**Single**-UMM



Part 1: Consistency

V-**Single**-UM

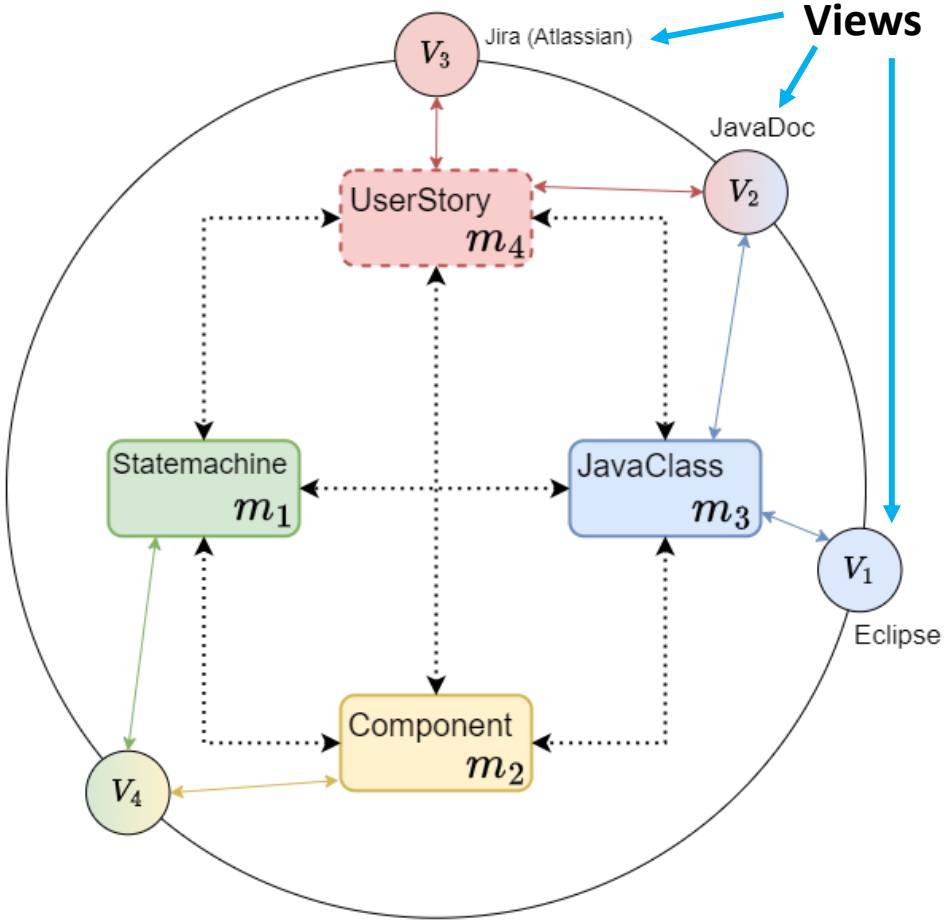
V-**Single**-UMM



Part 1: Consistency

V-**Single**-UM

Modifications via **Views**

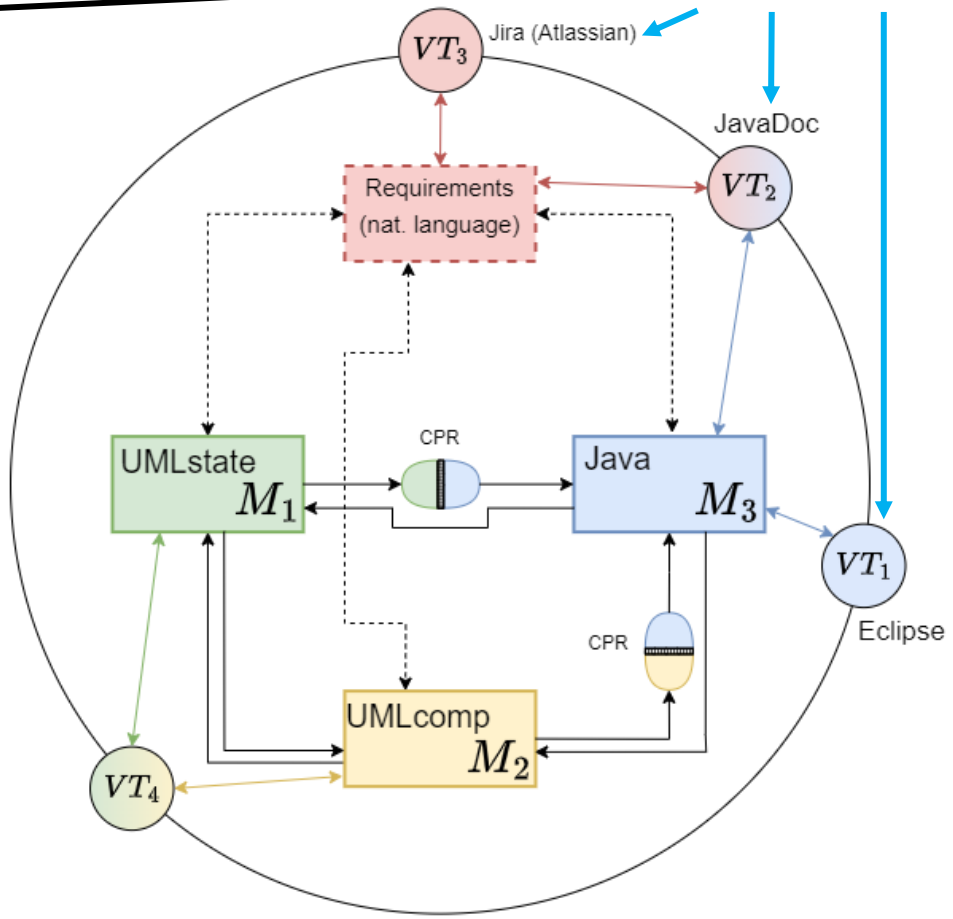


instance of

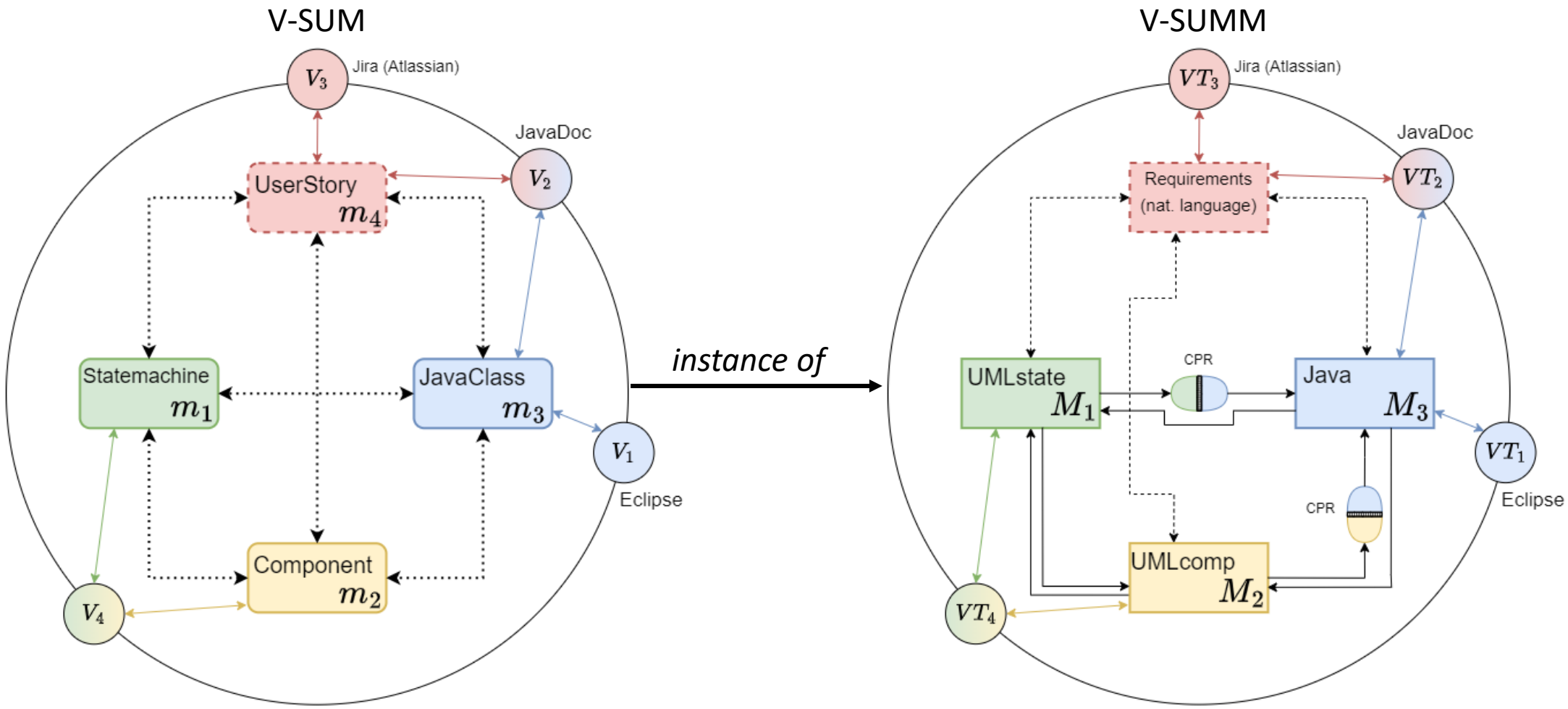
V-**Single**-UMM

(„subsets of VSUMM“)

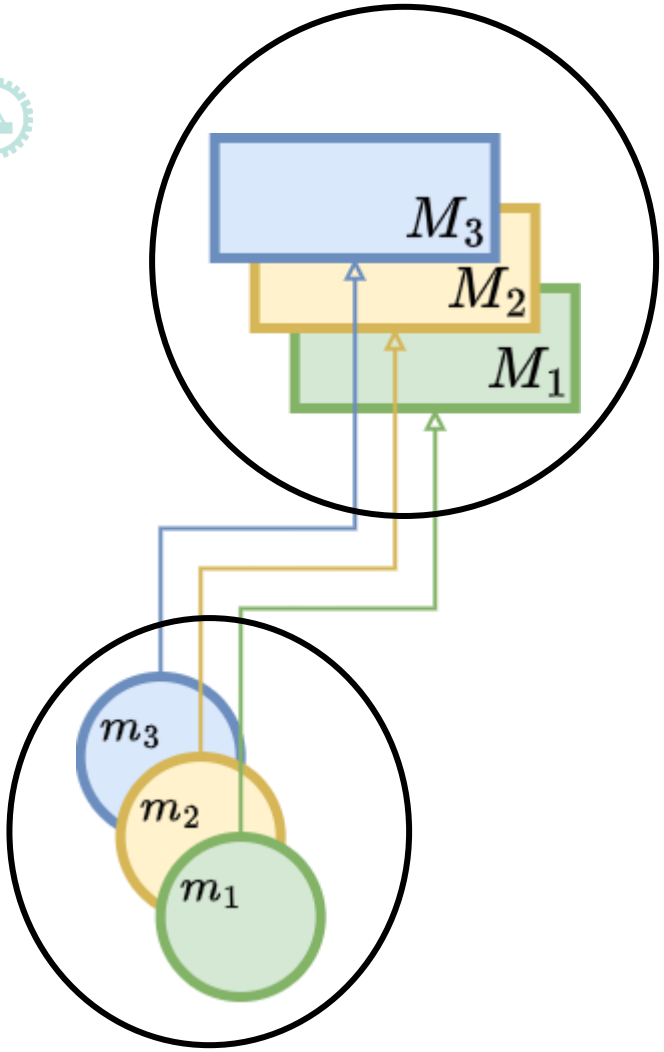
ViewTypes



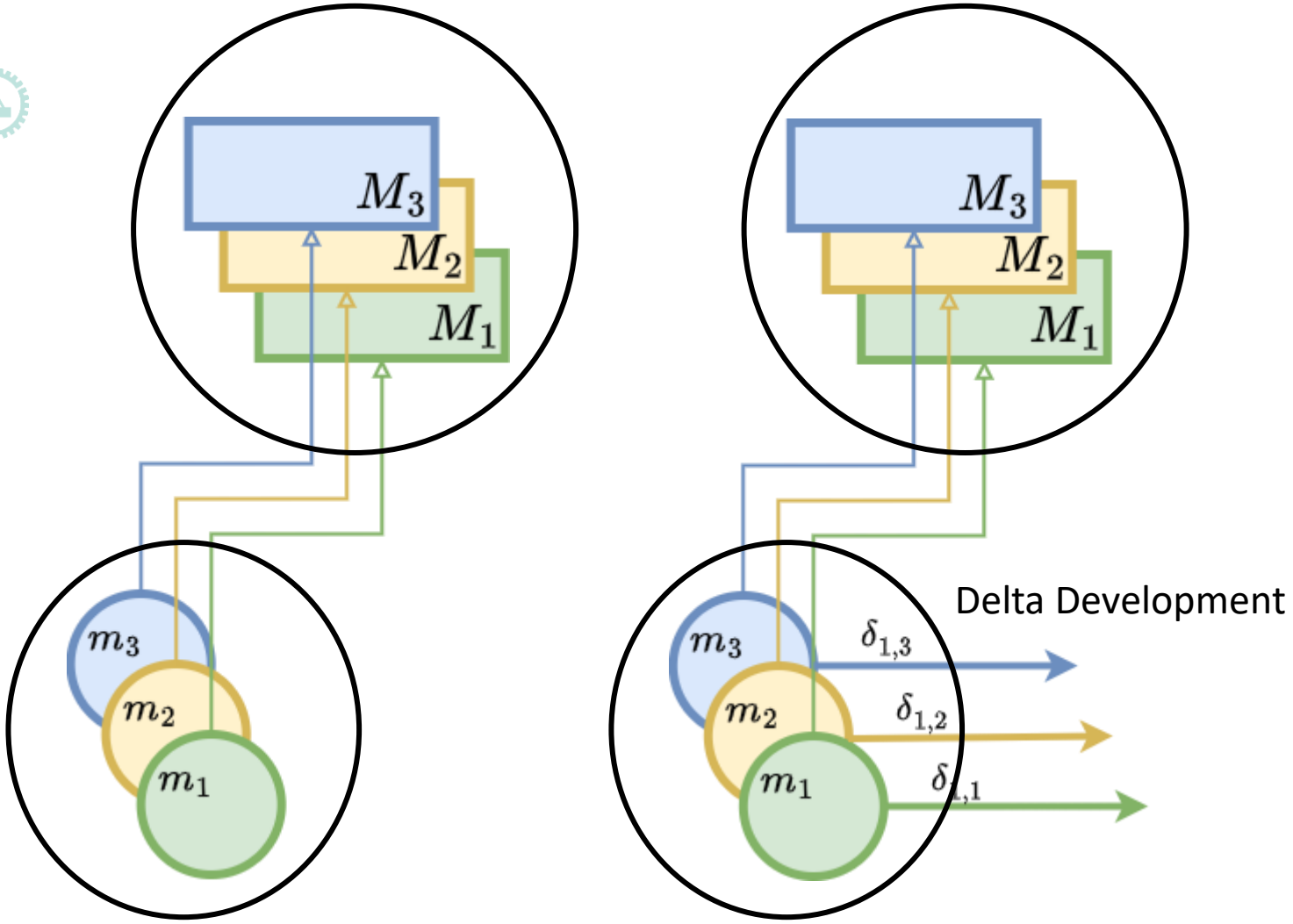
Part 1: Consistency



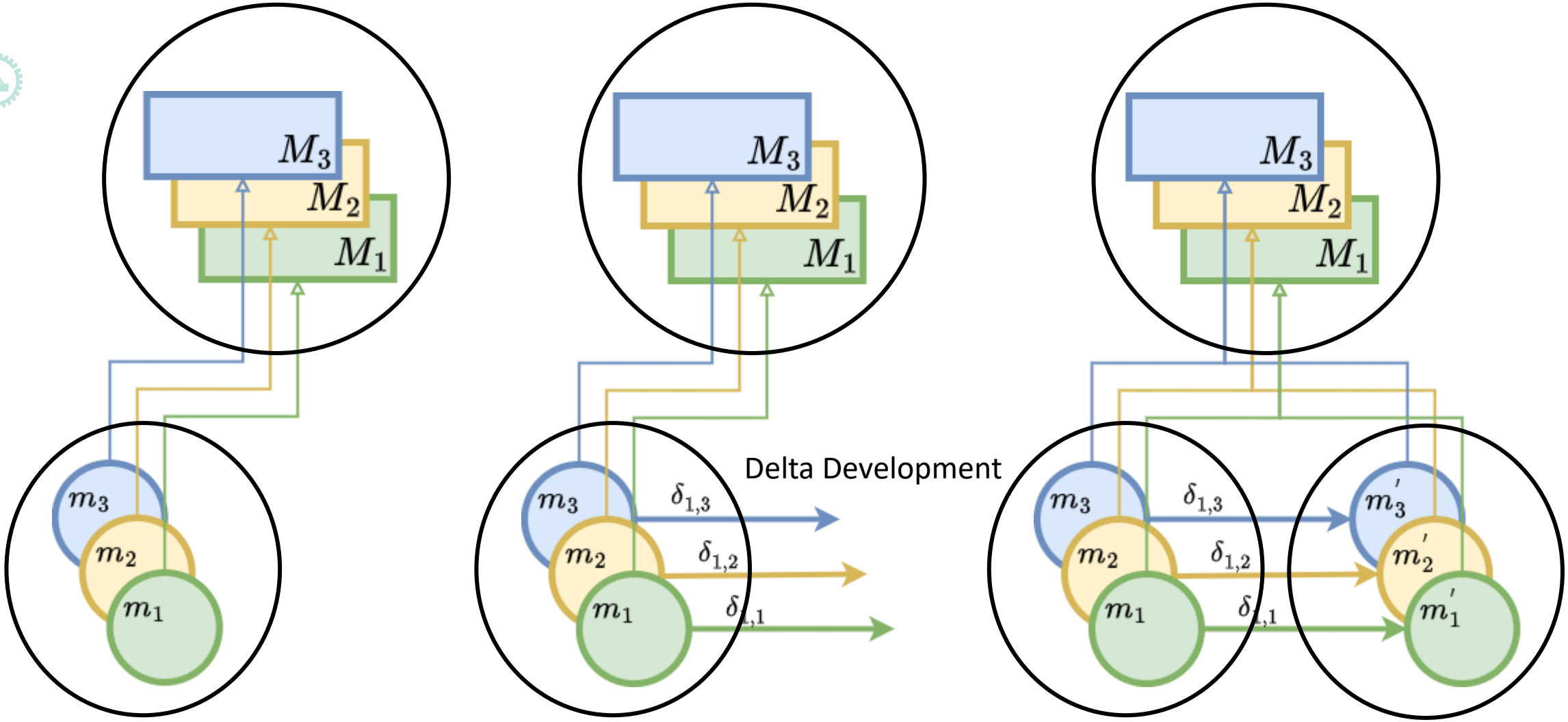
Part 2: Variability



Part 2: Delta-oriented Variability



Part 2: Delta-oriented Variability

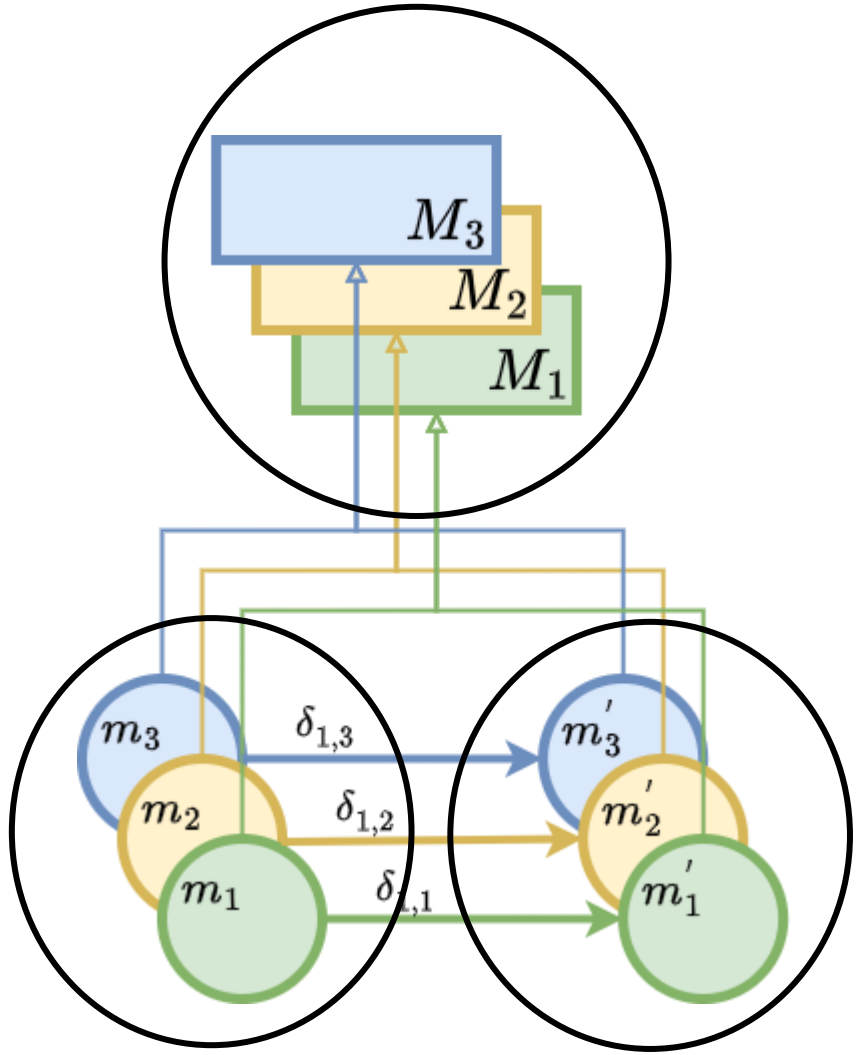


GEFÖRDERT DURCH DIE DFG Deutsche Forschungsgemeinschaft

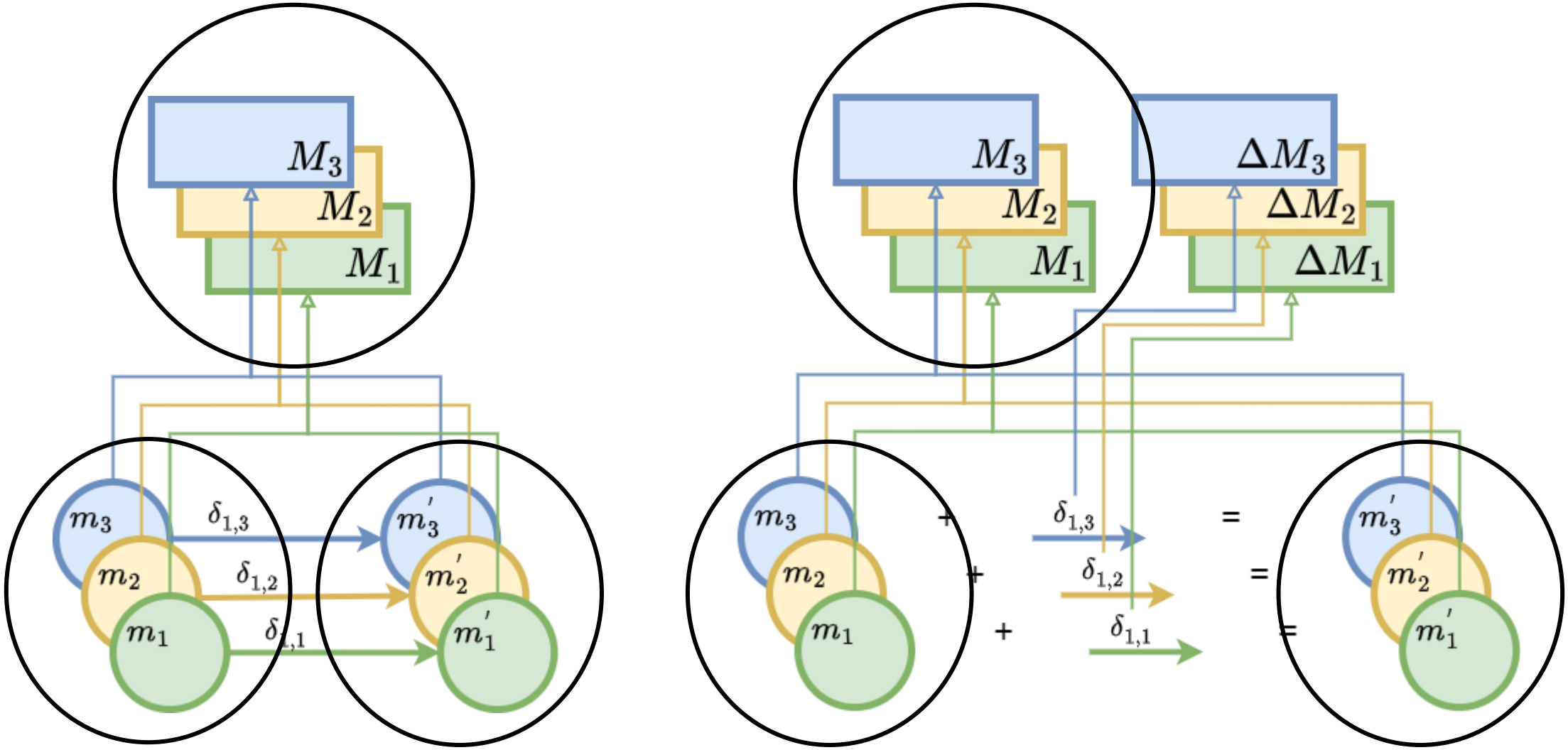
CONVIDE CRC 1608

How to combine variability and consistency?

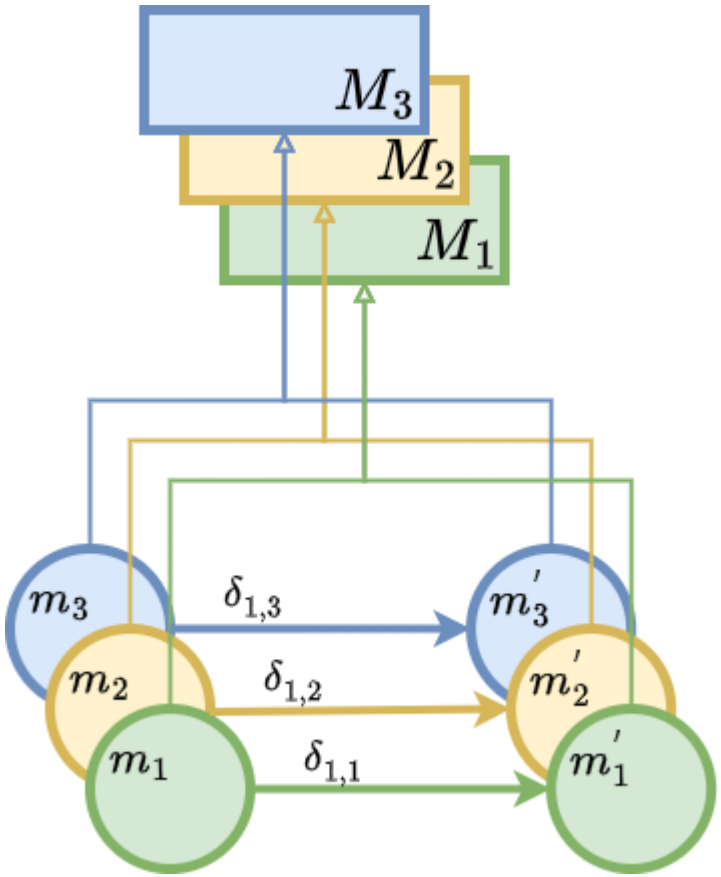
Part 2: Delta-oriented Variability

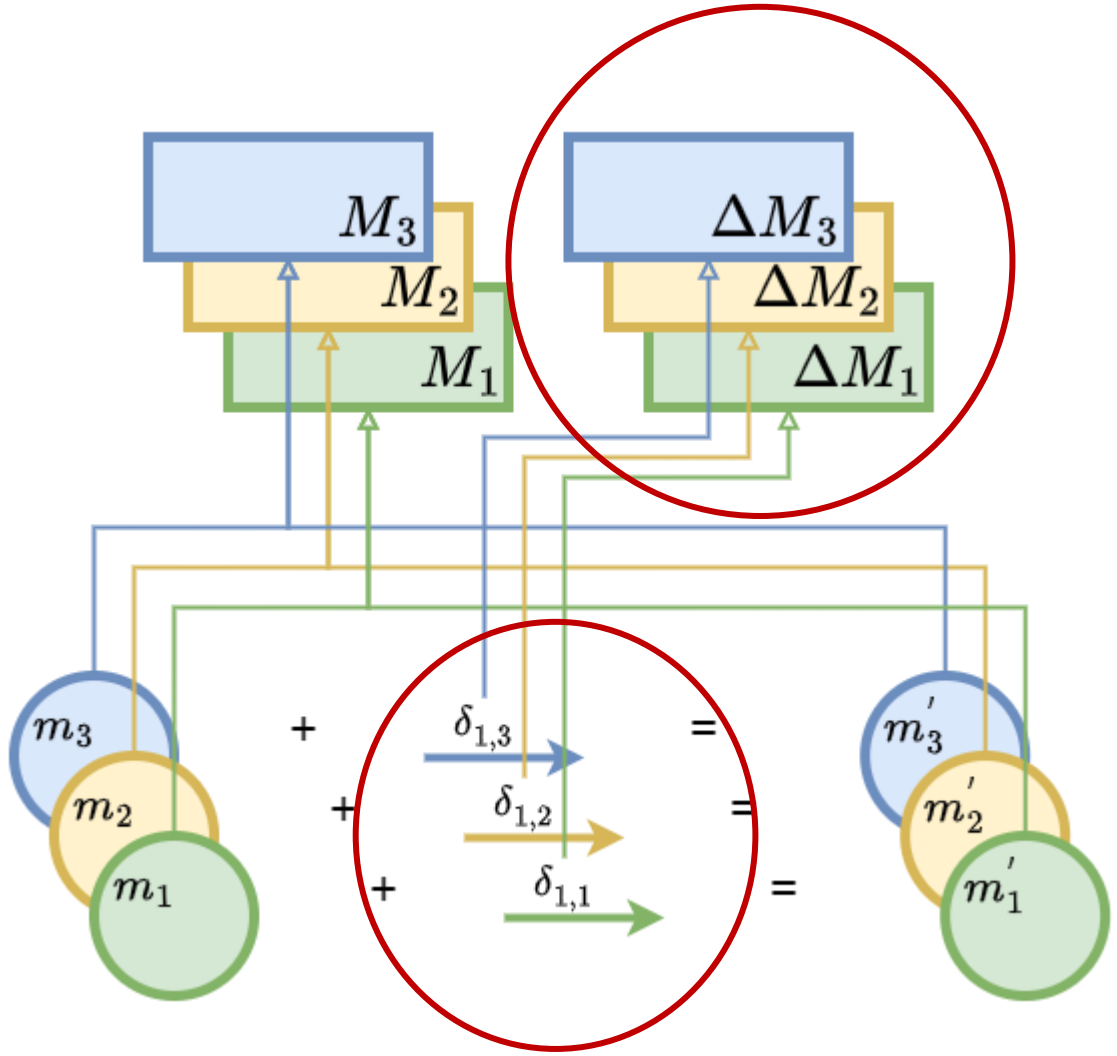


Part 2: Delta-oriented Variability **(Meta-)Modeling**

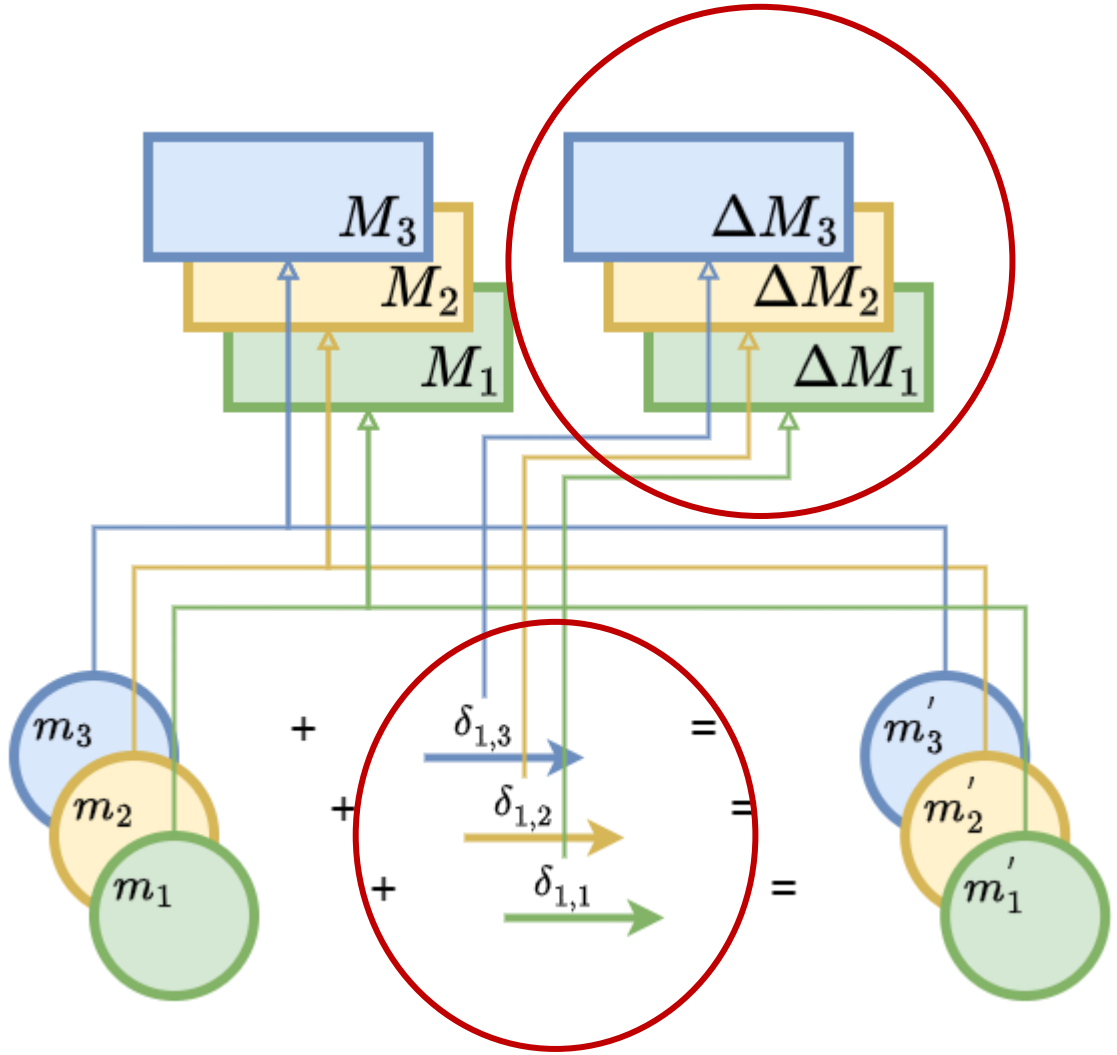


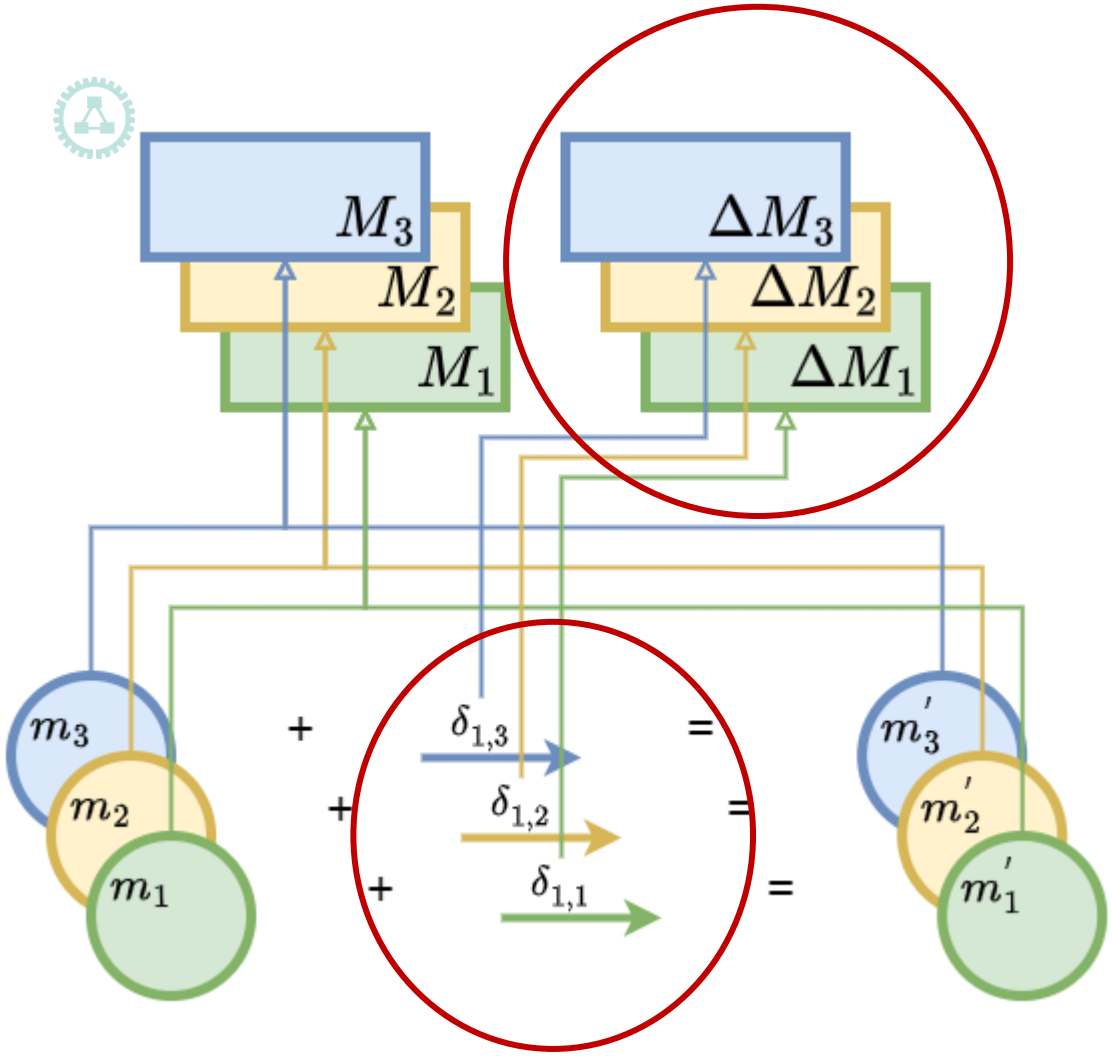
New: **Consistent** Delta-oriented Variability (Meta-)Modeling



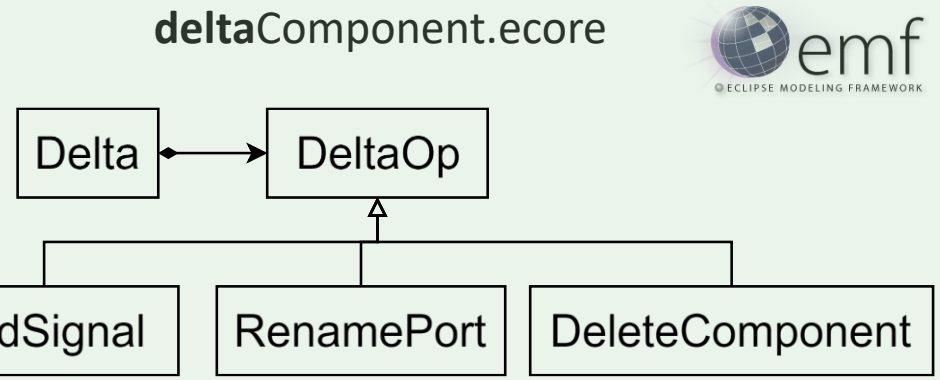


New: **Consistent** Delta-oriented Variability (Meta-)Modeling





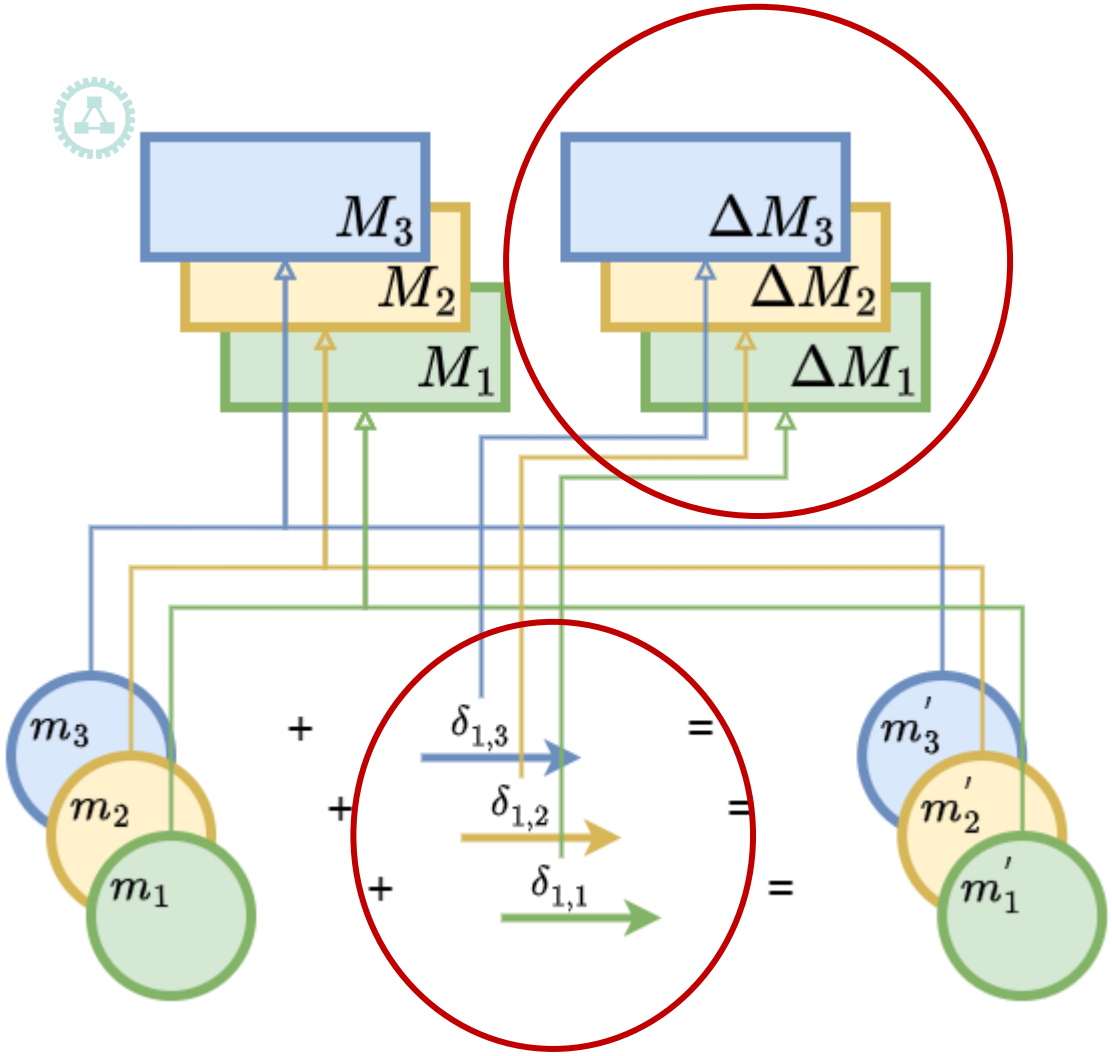
deltaComponent.ecore



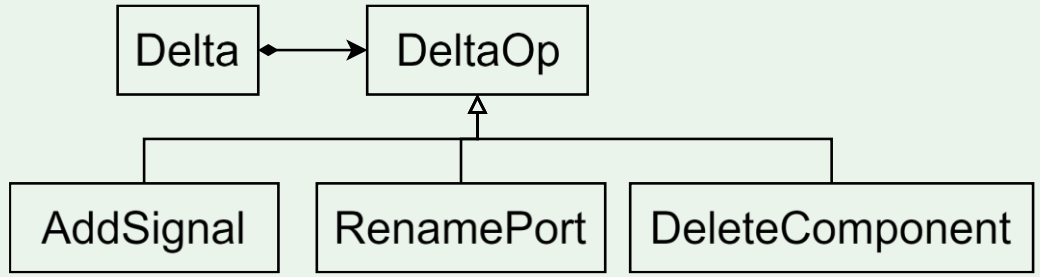
instance of

My.component**Delta**

Delta „AddingFingerProtection“:
- AddComponent(„FingerProtection“)
- AddPort(„activated“)
- ConnectComponent(ManPW,FingerProtection)
- AddSignal(„FingerProtectionOn“)
- SetOutputSignal(„activated“, „FingerProtectionOn“



deltaComponent.ecore



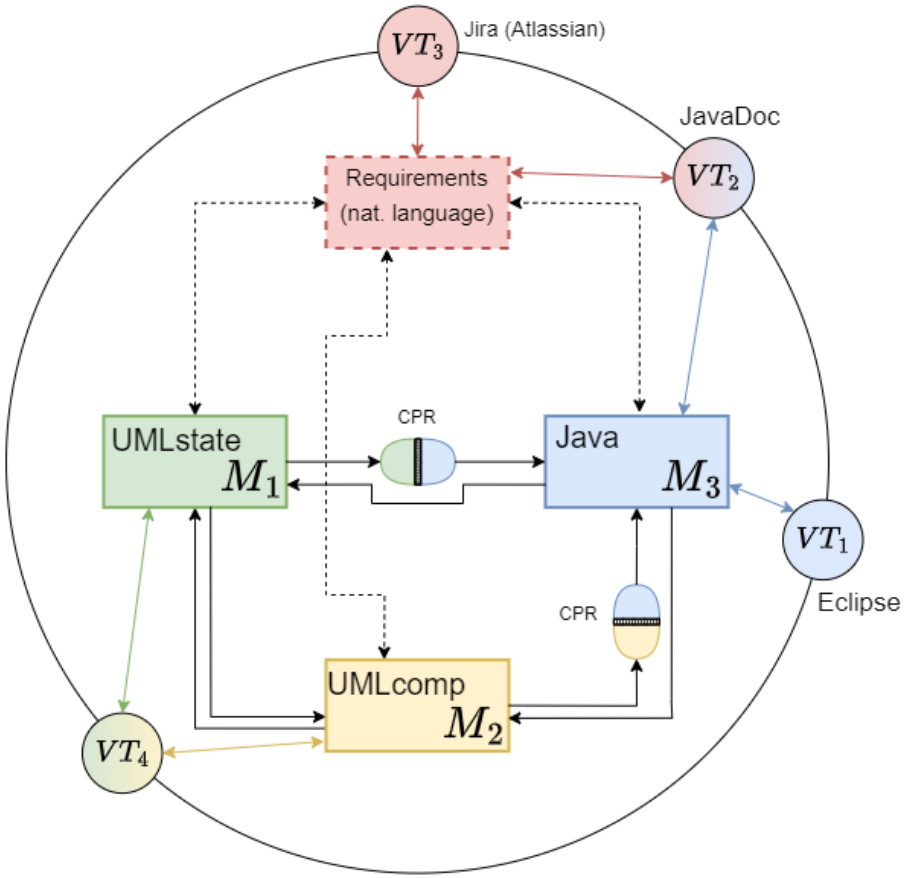
instance of

My.component**Delta**

Delta „AddingFingerProtection“:
- AddComponent(„FingerProtection“)
- AddPort(„activated“)
- ConnectComponent(ManPW,FingerProtection)
- AddSignal(„FingerProtectionOn“)
- SetOutputSignal(„activated“, „FingerProtectionOn“

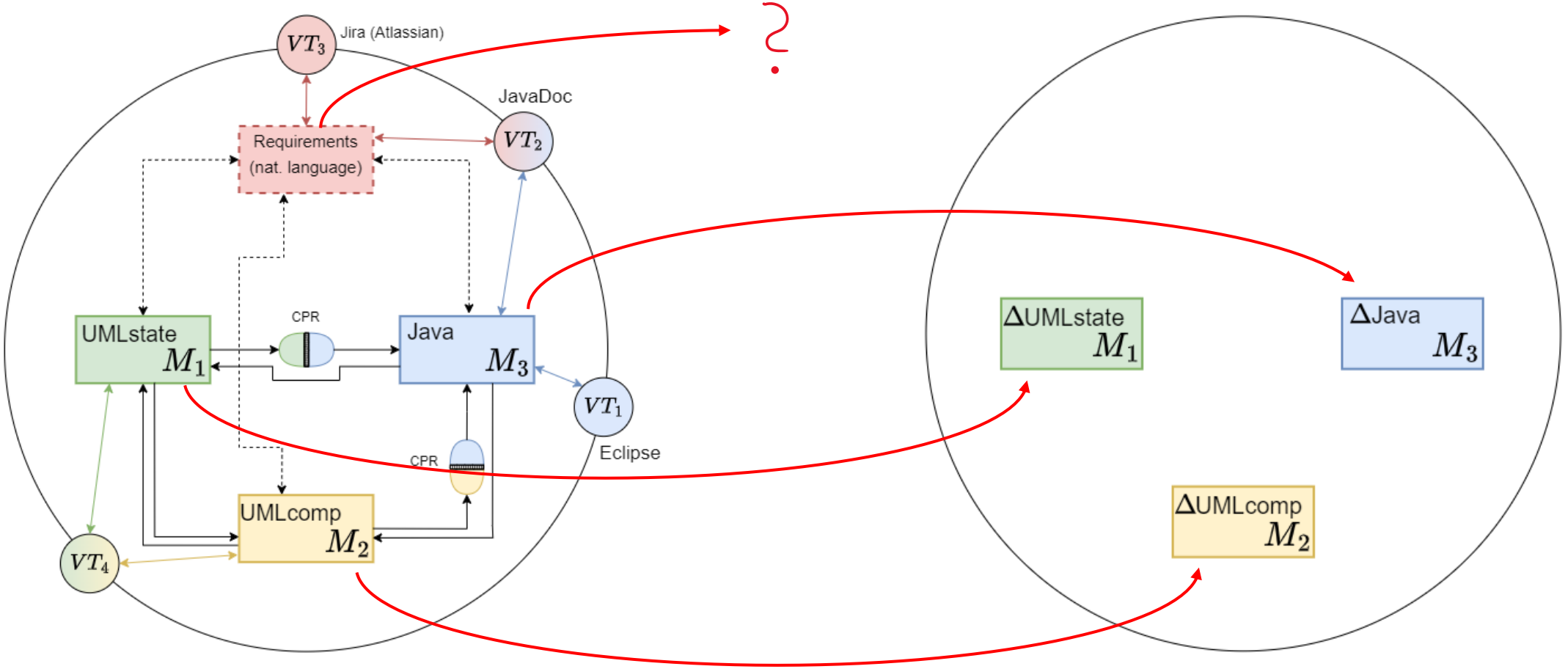
Constructing a Var-V-SUMM from a V-SUMM

Virtual Single Underlaying Metamodel (V-SUMM)

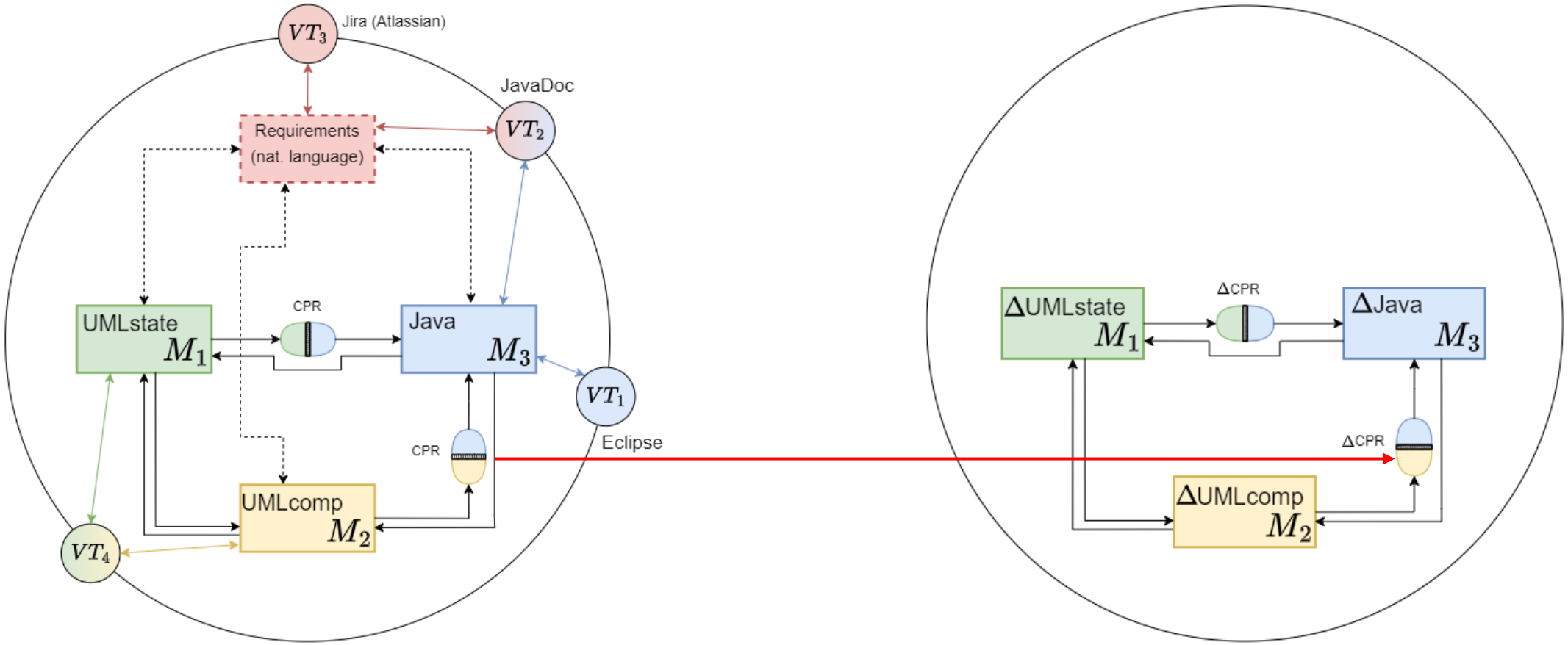


Variability-aware Virtual Single Underlaying Metamodel (**Var**-V-SUMM)

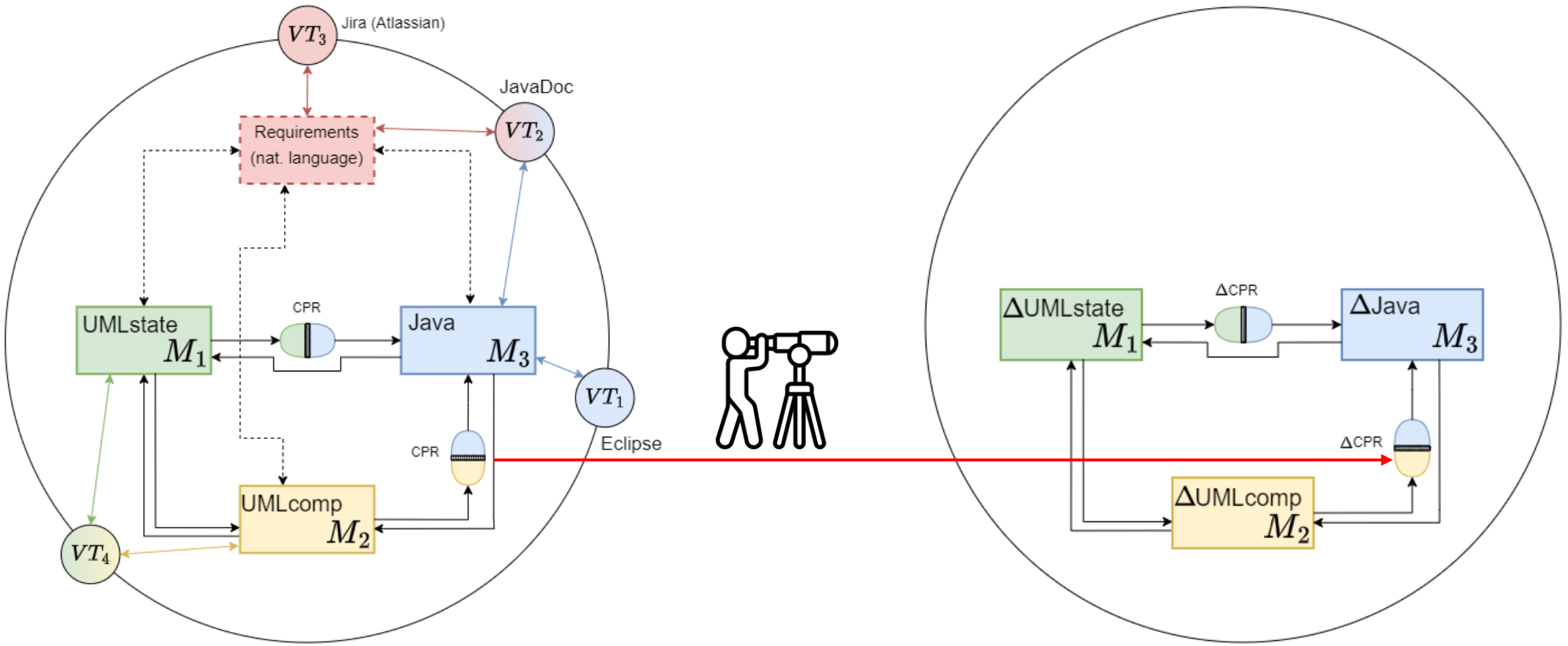
Constructing a Var-V-SUMM from a V-SUMM



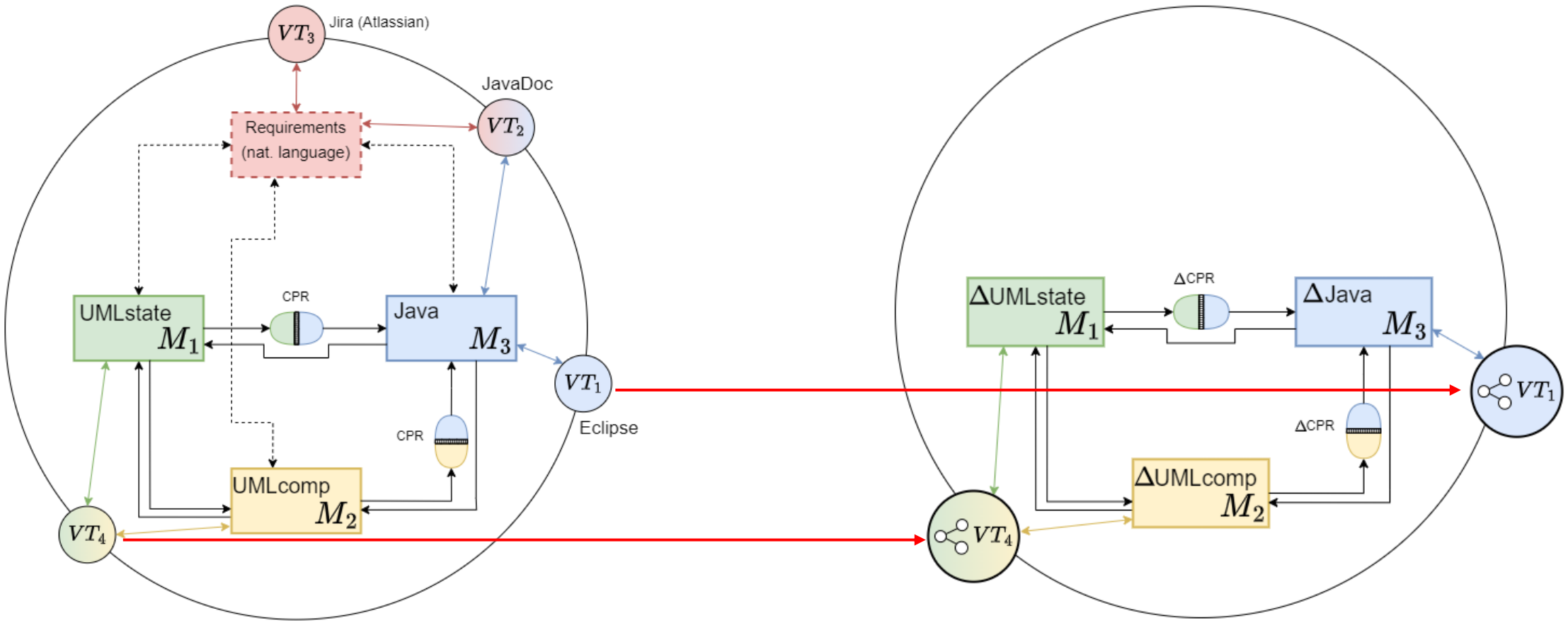
Constructing a Var-V-SUMM from a V-SUMM



Constructing a Var-V-SUMM from a V-SUMM

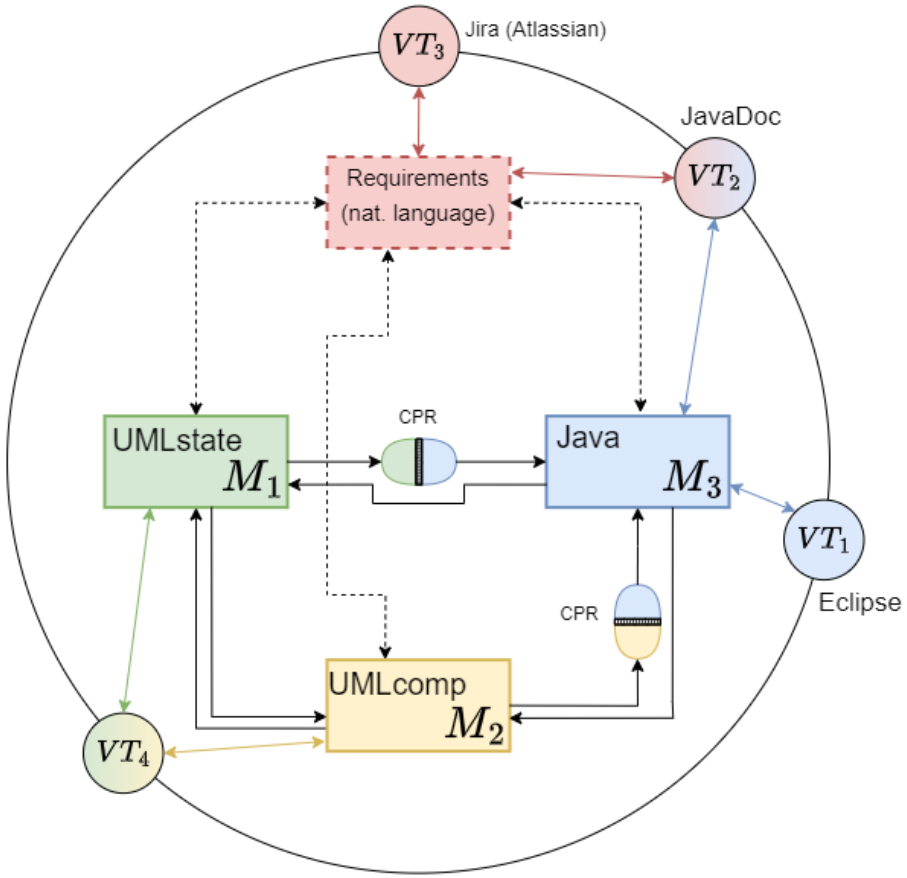


Constructing a Var-V-SUMM from a V-SUMM



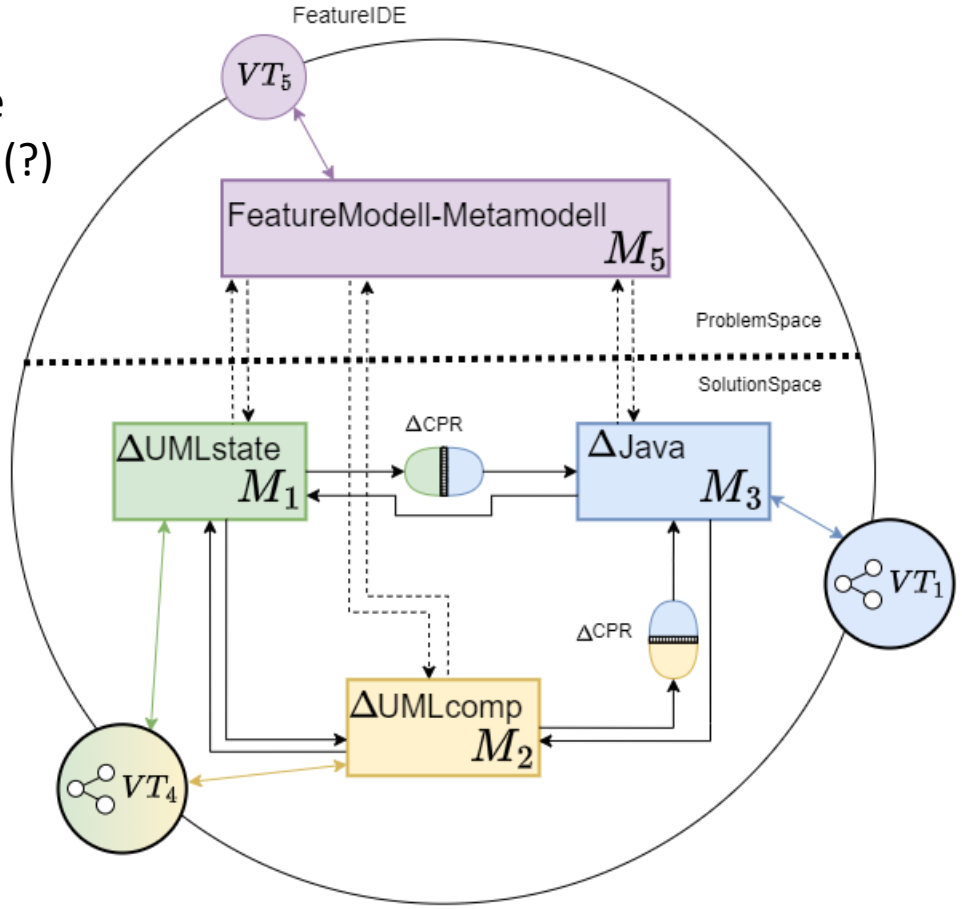
Constructing a Var-V-SUMM from a V-SUMM

V-SUMM

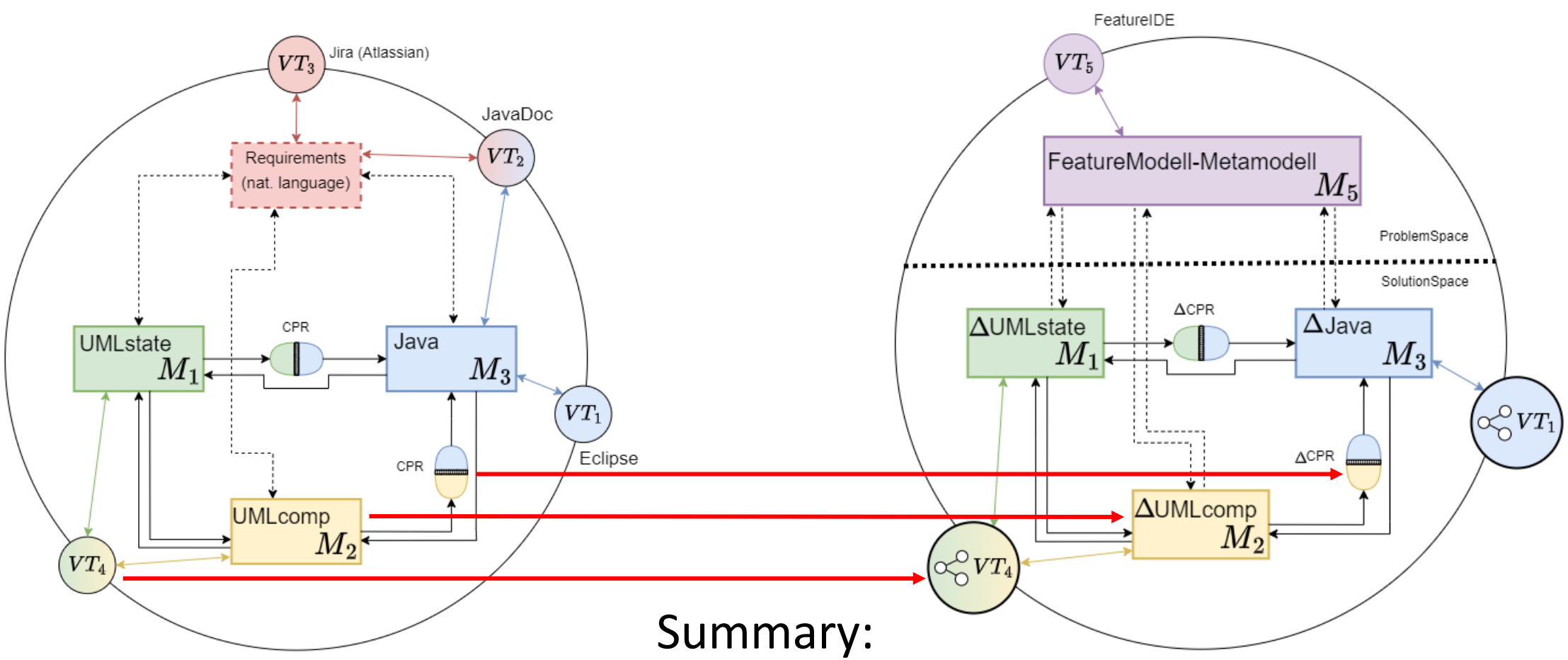


Including the Problem Space (?)

Var-V-SUMM



How to derive a Var-V-SUMM from a V-SUMM?



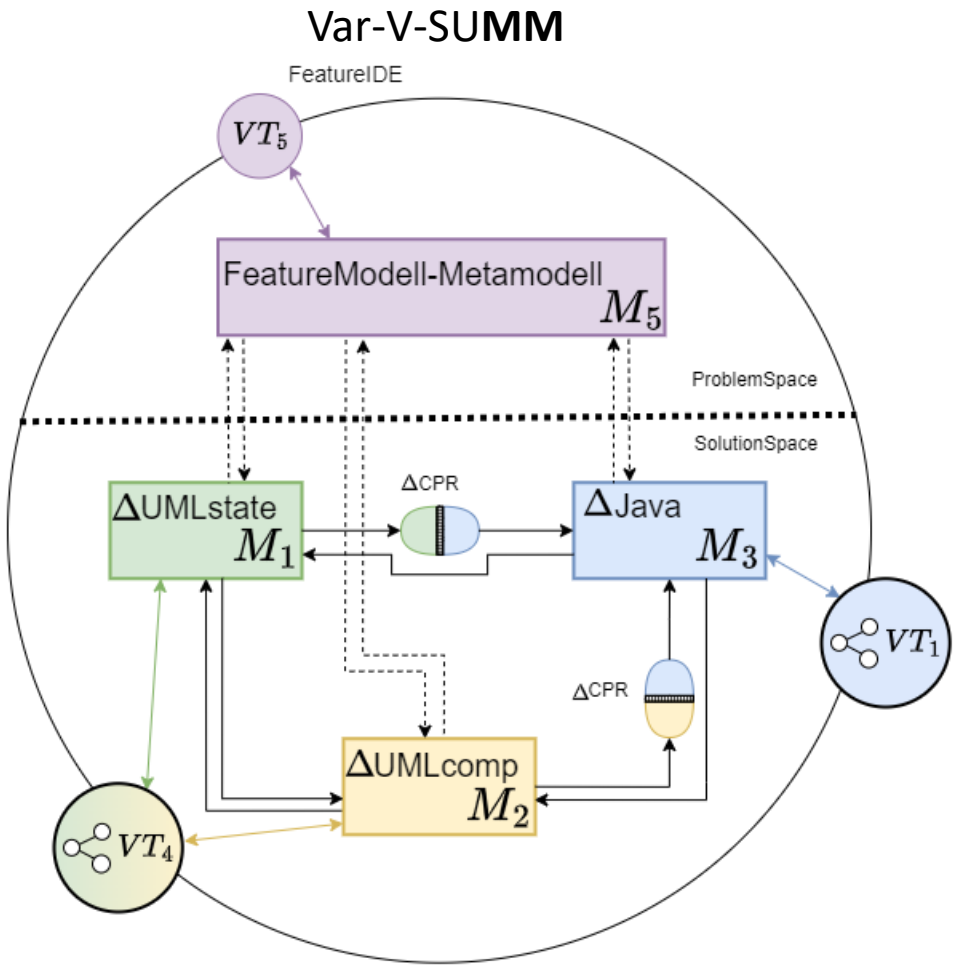
Summary:

3 „conversion steps“

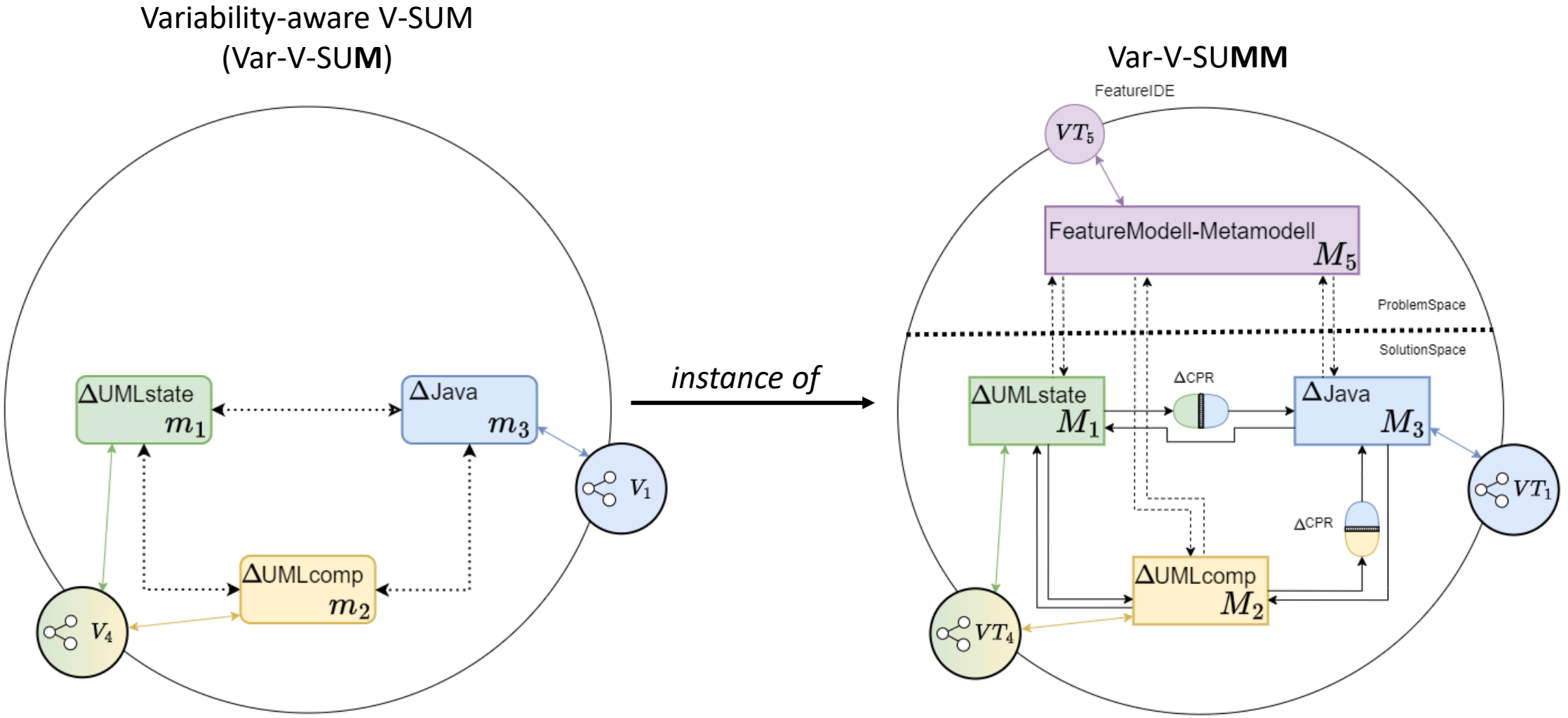
Instantiating the Var-V-SUM

?

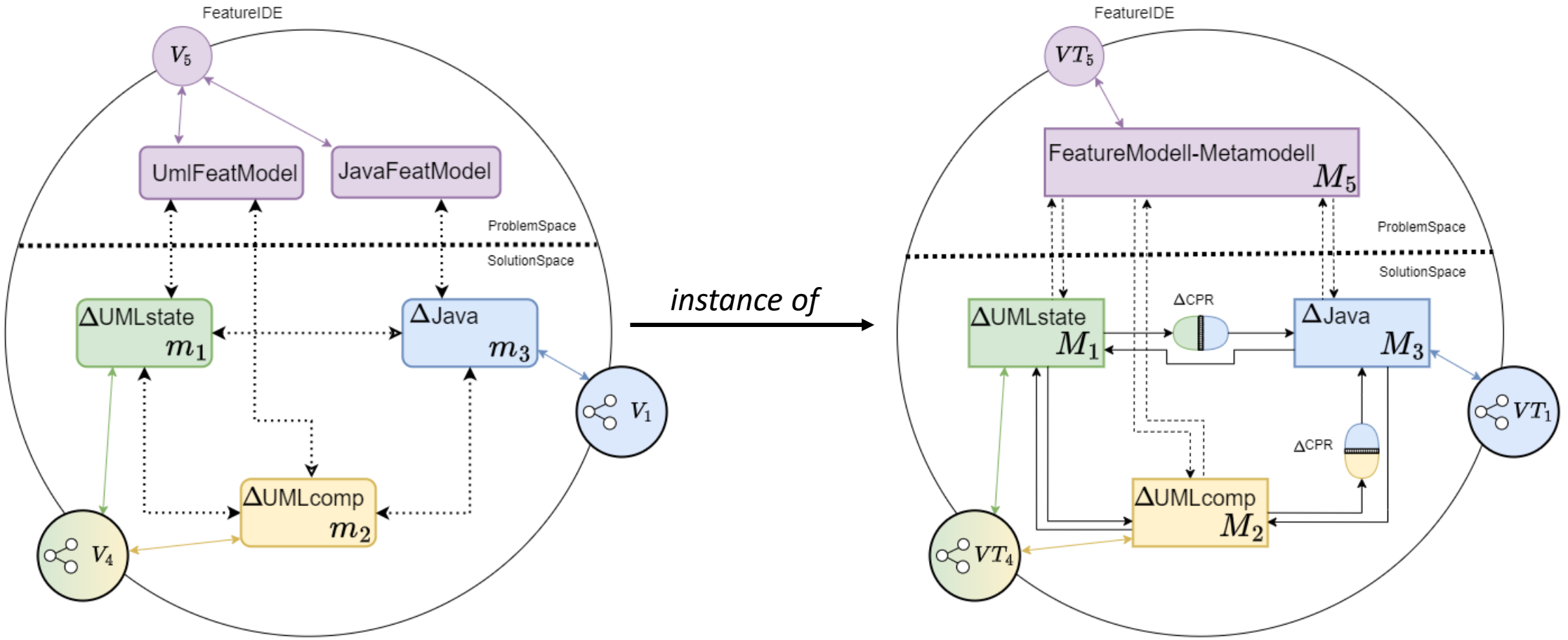
instance of



Instantiating the Var-V-SUM



Instantiating the Var-V-SUM



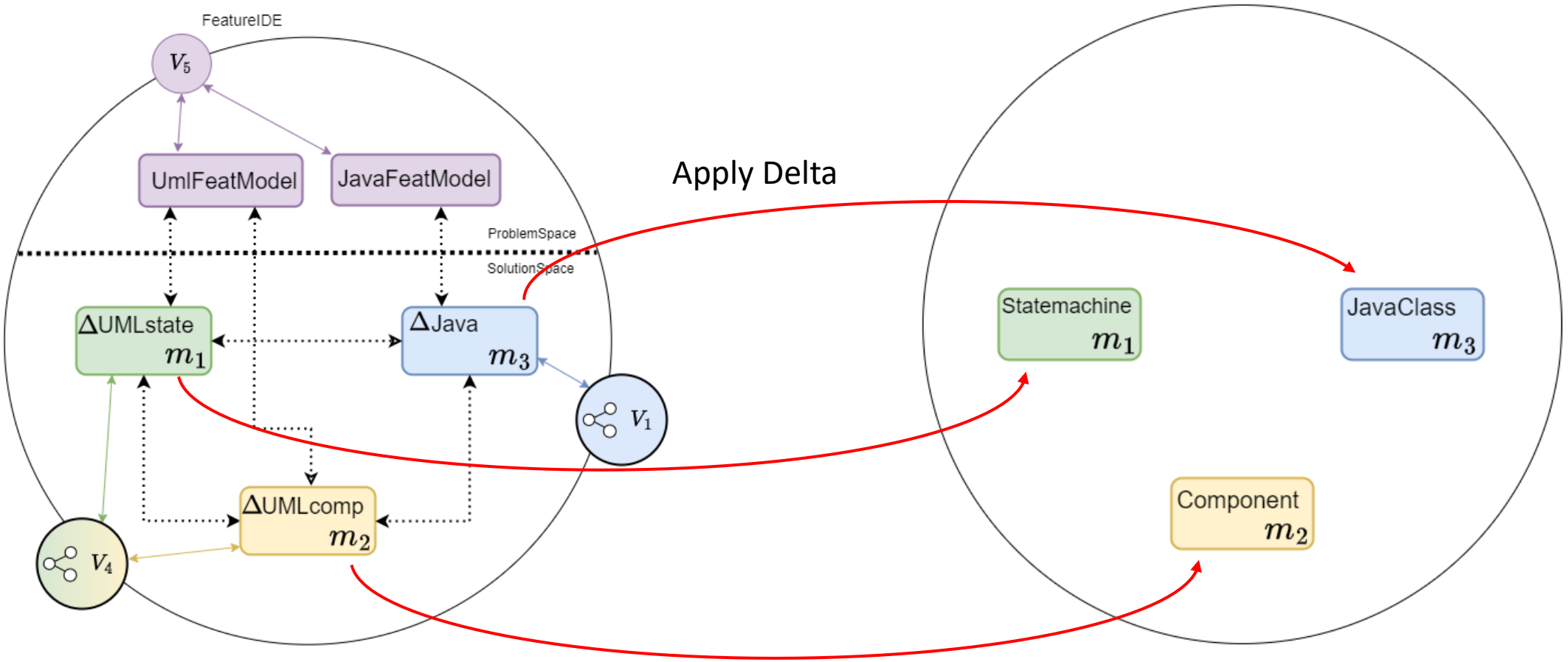
GEFÖRDERT DURCH DIE DFG Deutsche Forschungsgemeinschaft

CONVIDE CRC 1608

How to work with a (Var-)V-SU(M)M?

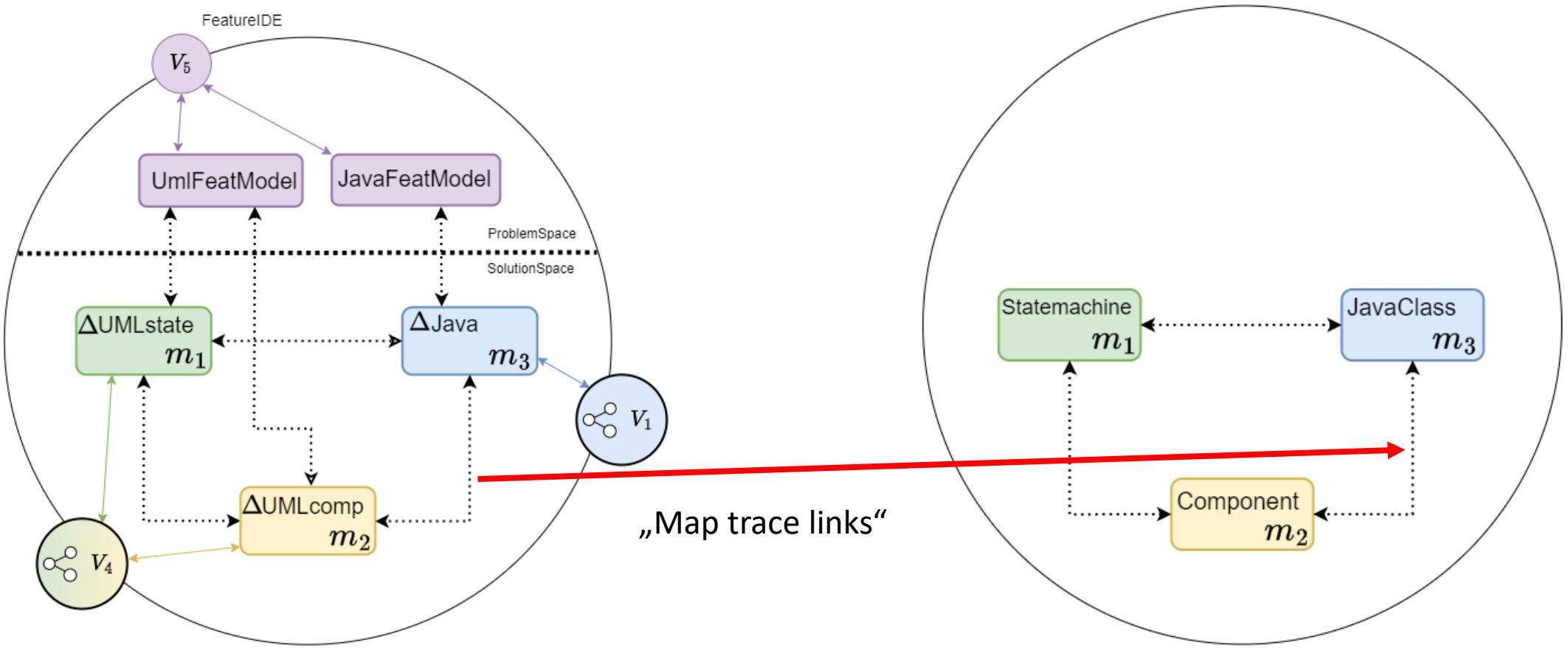
Product-oriented development

Deriving and working on a single product



Product-oriented development

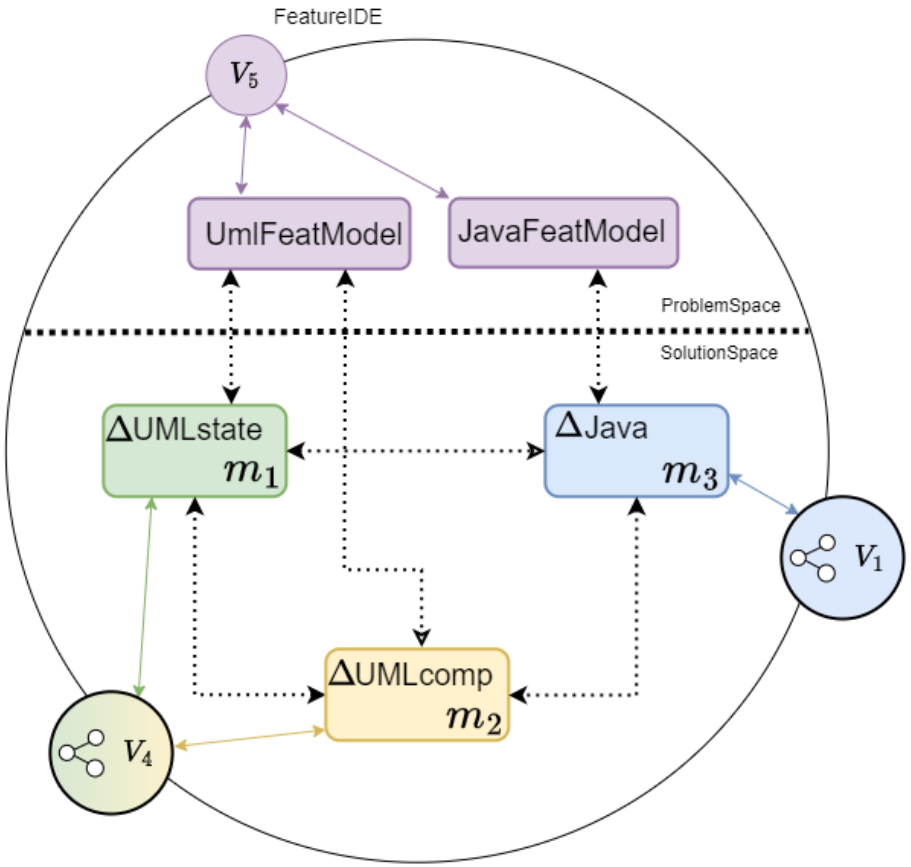
Deriving and working on a single product



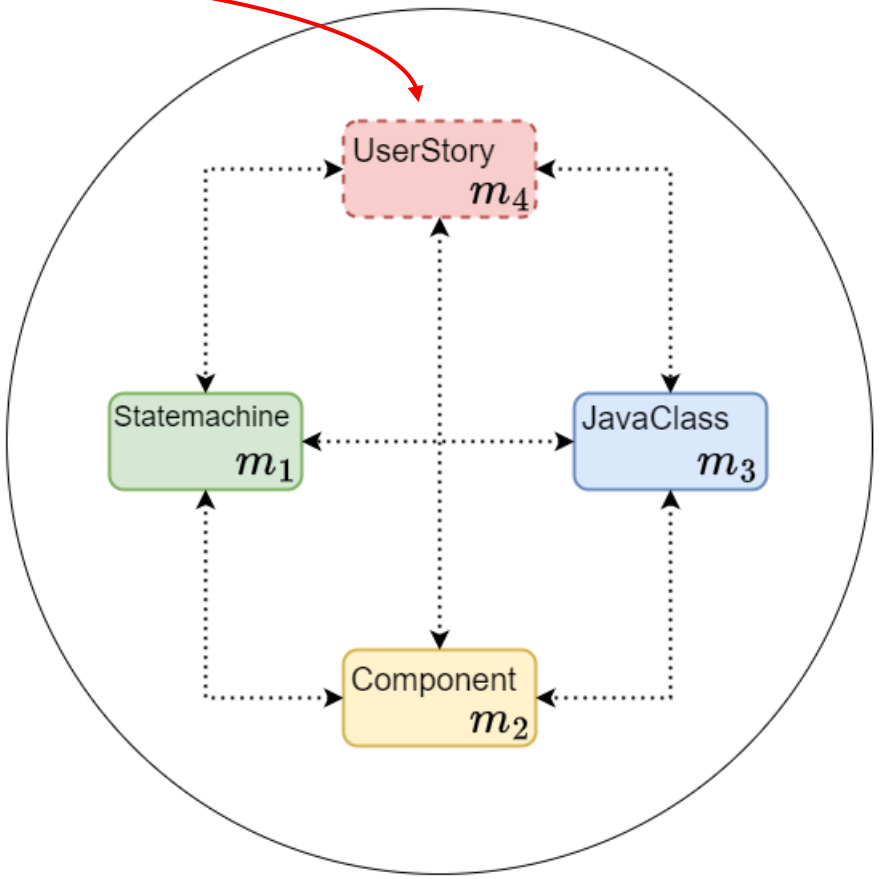
Product-oriented development

Deriving and working on a single product

Var-V-SUM

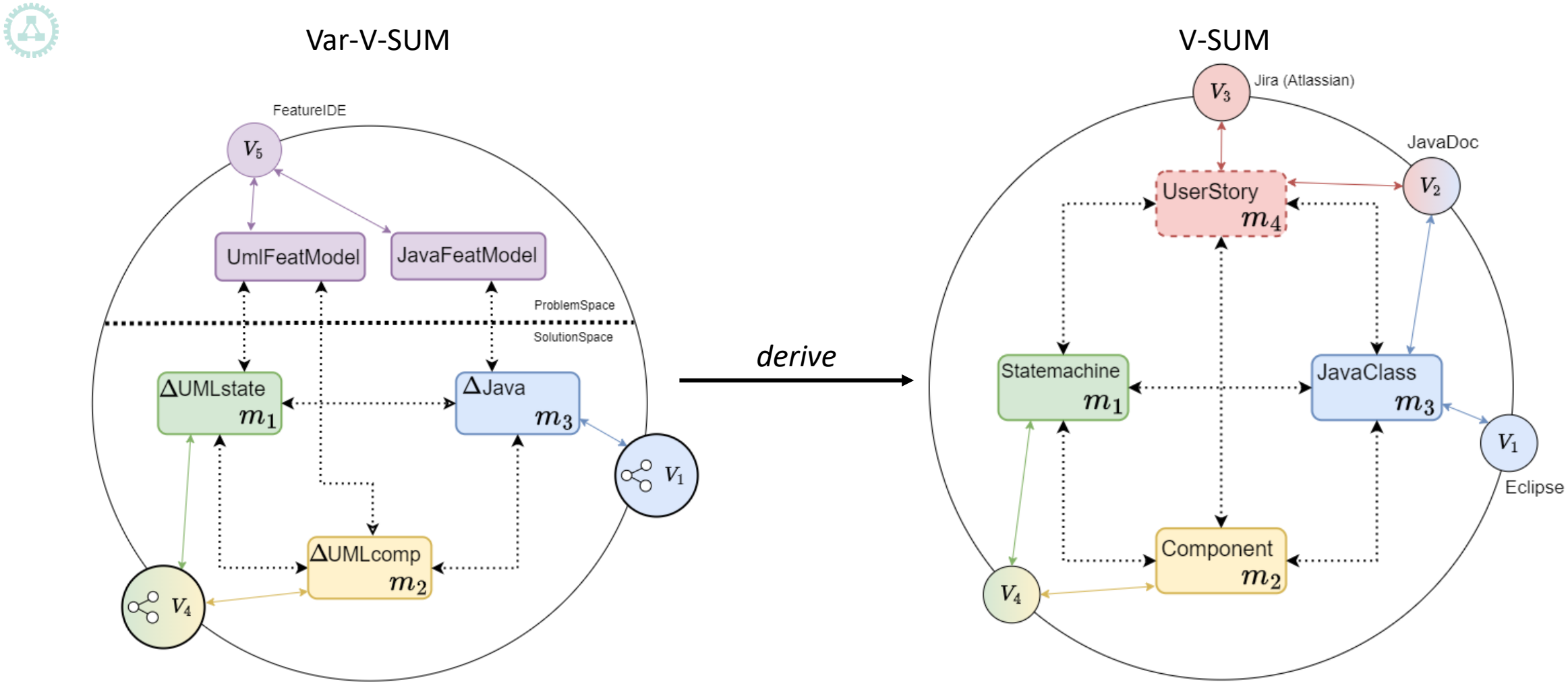


V-SUM



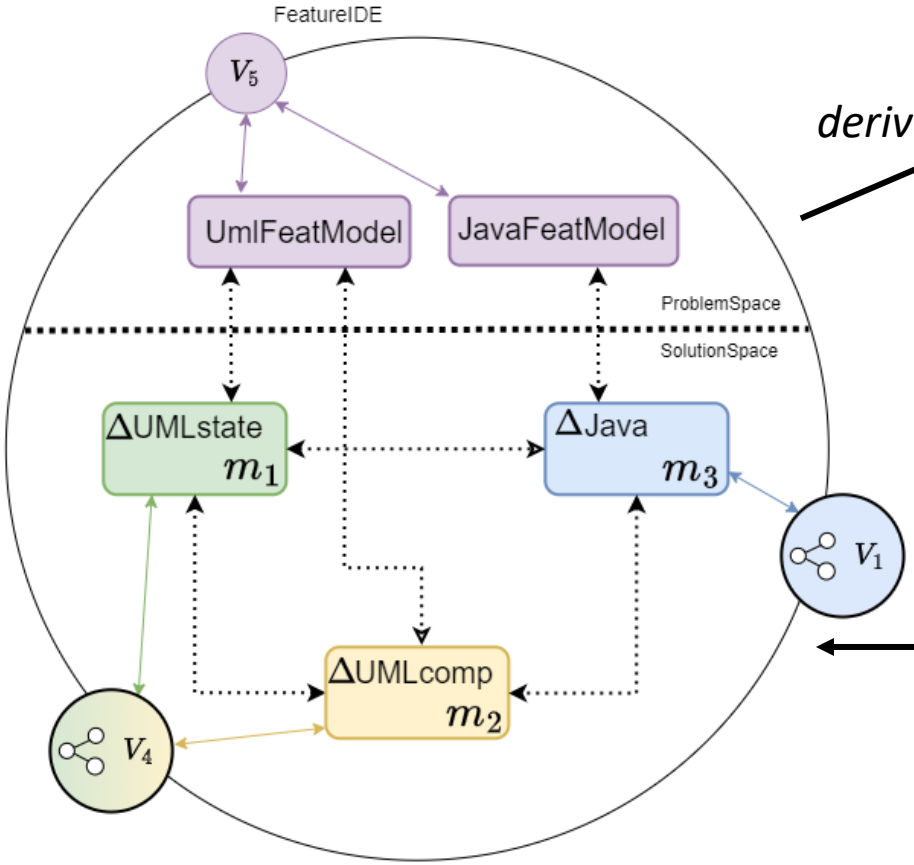
Product-oriented development

Deriving and working on a single product

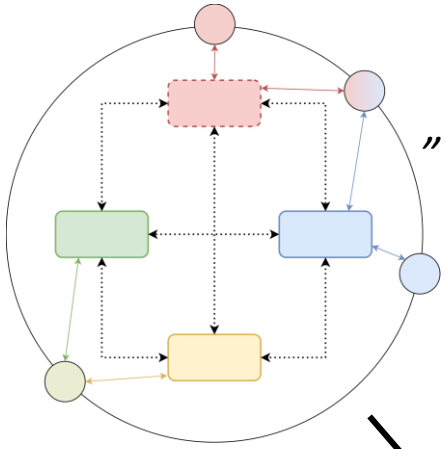


Product-oriented development

Deriving and working on a single product



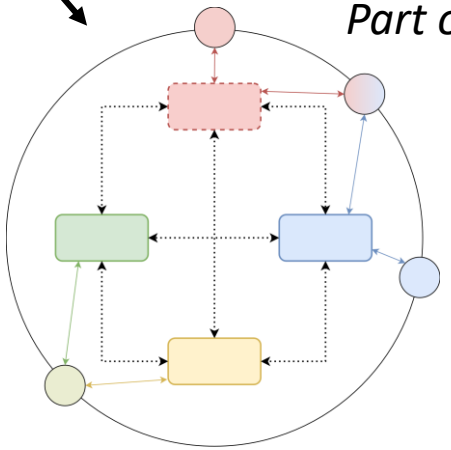
derive



„Part of the PL“

modify

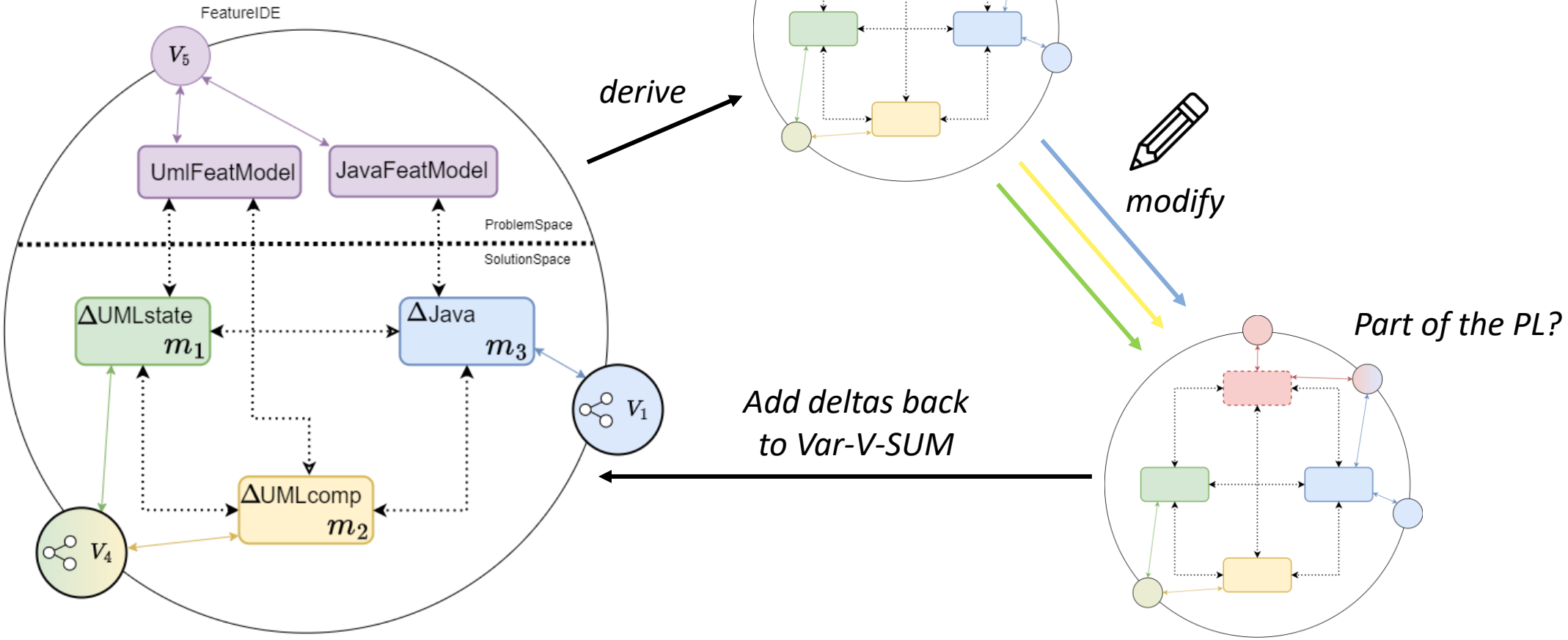
Part of the PL?



internalize

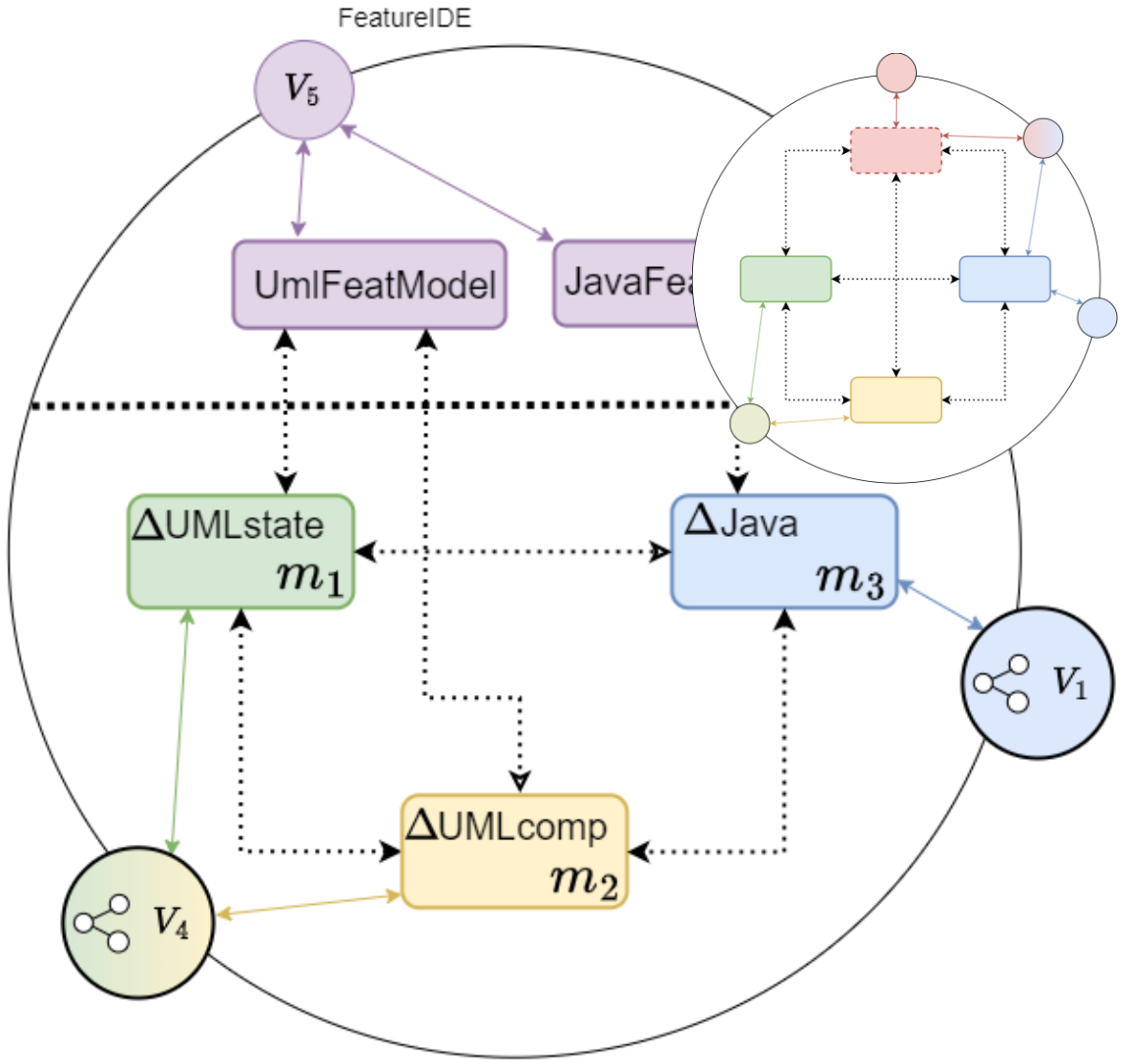
Product-oriented development

Deriving and working on a single product



Product-oriented development

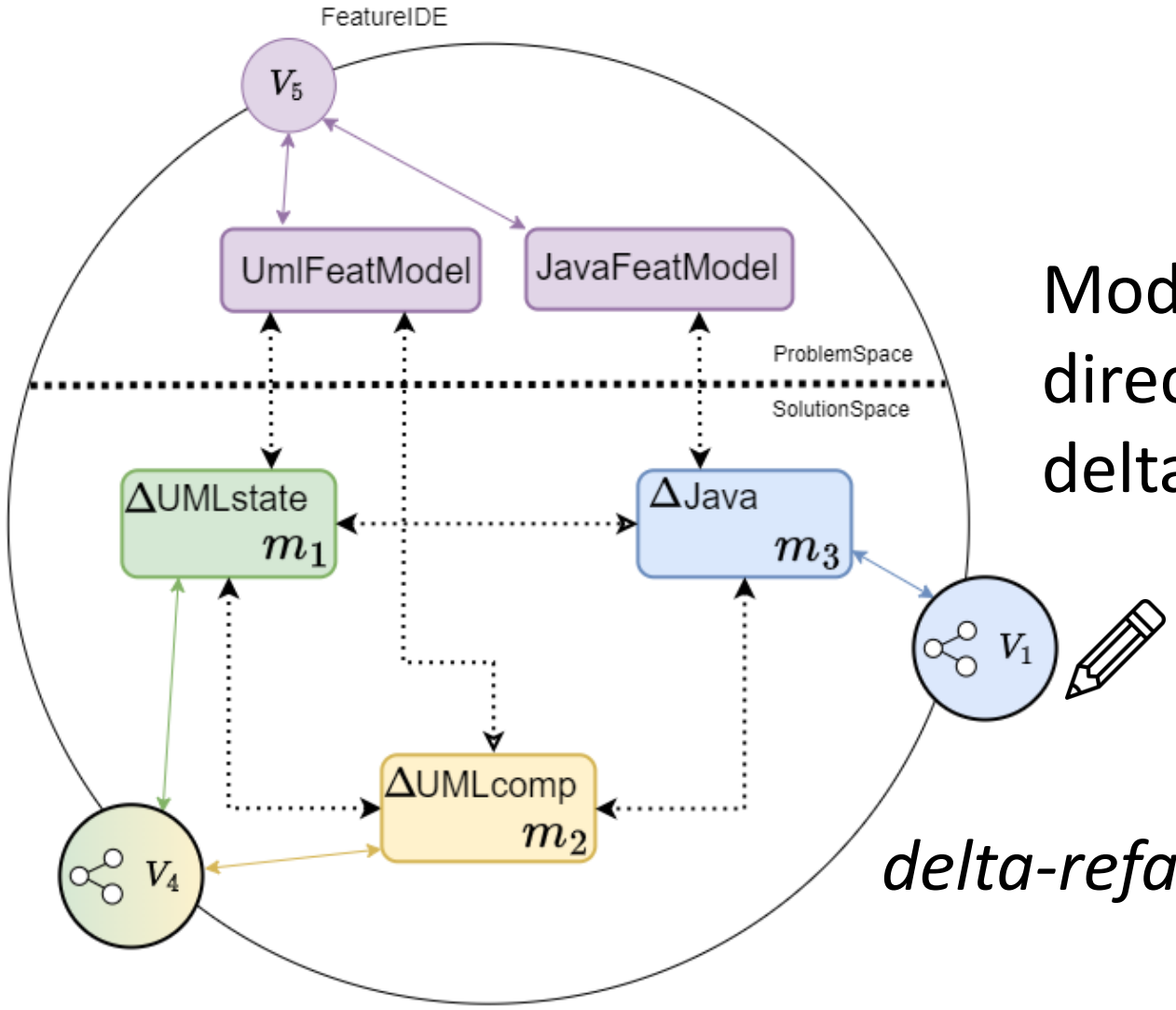
Deriving and working on a single product



The V-SUM as a view of the Var-V-SUM

Plattform-oriented development

Deriving and working on a single product



Modify the product line directly. Delta-CPRs keep the delta-models consistent.

higher-order deltas

delta-refactoring

safe PL evolution

GEFÖRDERT DURCH DIE DFG Deutsche Forschungsgemeinschaft

CONVIDE CRC 1608

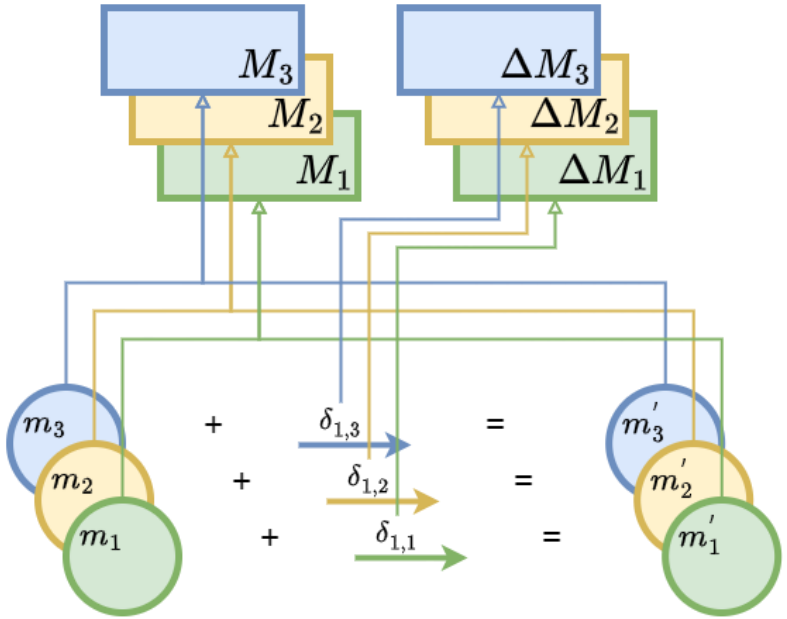
Can all this be better categorized?

Consistency Categories

	FeatureModel	Metamodel	Model	DeltaLanguage	DeltaModel
FeatureModel	?				
Metamodel	?	?			
Model	?	?	?		
DeltaLanguage	?	?	?	?	
DeltaModel	?	?	?	?	?

Consistency Categories

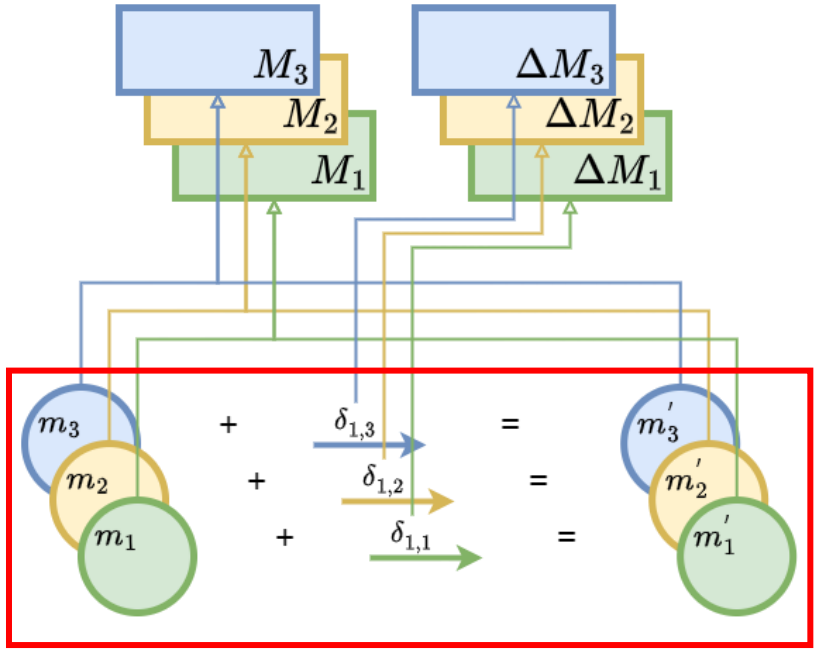
	FeatureModel	Metamodel	Model	DeltaLanguage	DeltaModel
FeatureModel	?				
Metamodel	?	?			
Model	?	*instance of*	?		
DeltaLanguage	?	?	?	?	
DeltaModel	?	?	?	*instance of*	?



1. How is consistency defined?
2. How can consistency established initially?
3. How can consistency preserved if artifacts change due to variability?

Consistency Categories

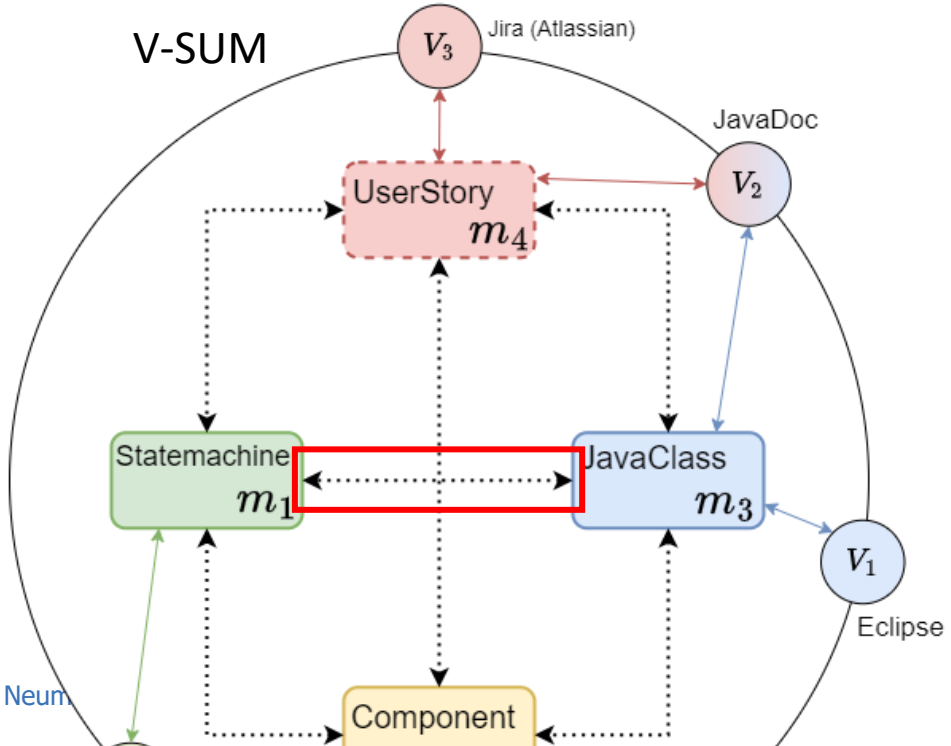
	FeatureModel	Metamodel	Model	DeltaLanguage	DeltaModel
FeatureModel	?				
Metamodel	?	?			
Model	?	*instance of*	?		
DeltaLanguage	?	?	?	?	
DeltaModel	?	?	apply-to/derive	*instance of*	?

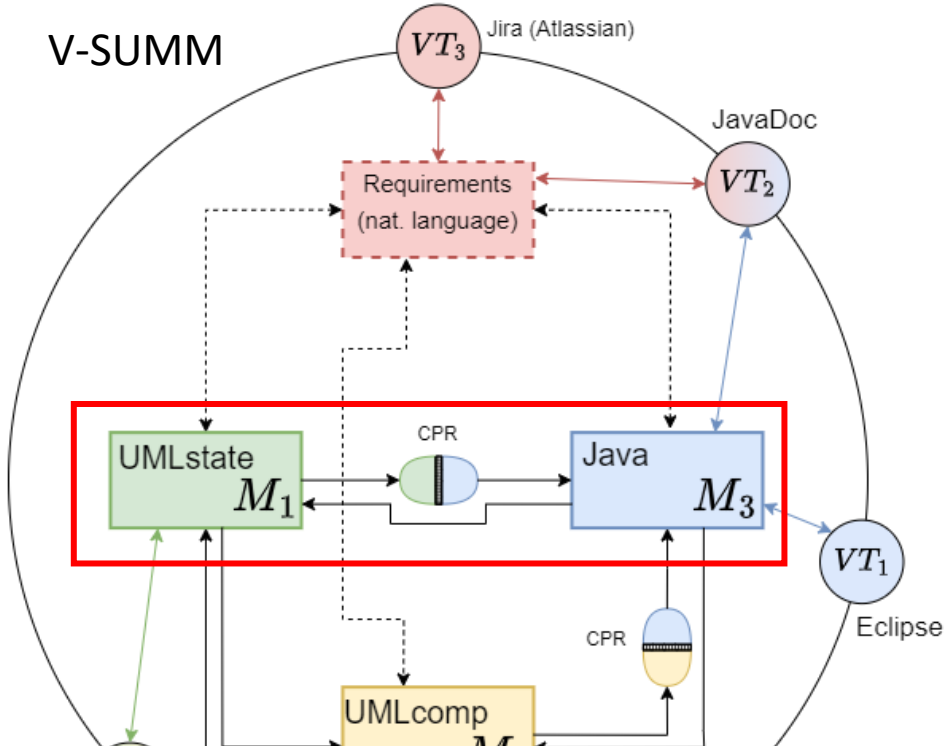


1. How is consistency defined?
2. How can consistency established initially?
3. How can consistency preserved if artifacts change due to variability?

Consistency Categories

	FeatureModel	Metamodel	Model	DeltaLanguage	DeltaModel
FeatureModel	?				
Metamodel	?	Product-CPR			
Model	?	*instance of*	Trace Links		
DeltaLanguage	?	?	?	?	
DeltaModel	?	?	apply-to/derive	*instance of*	?

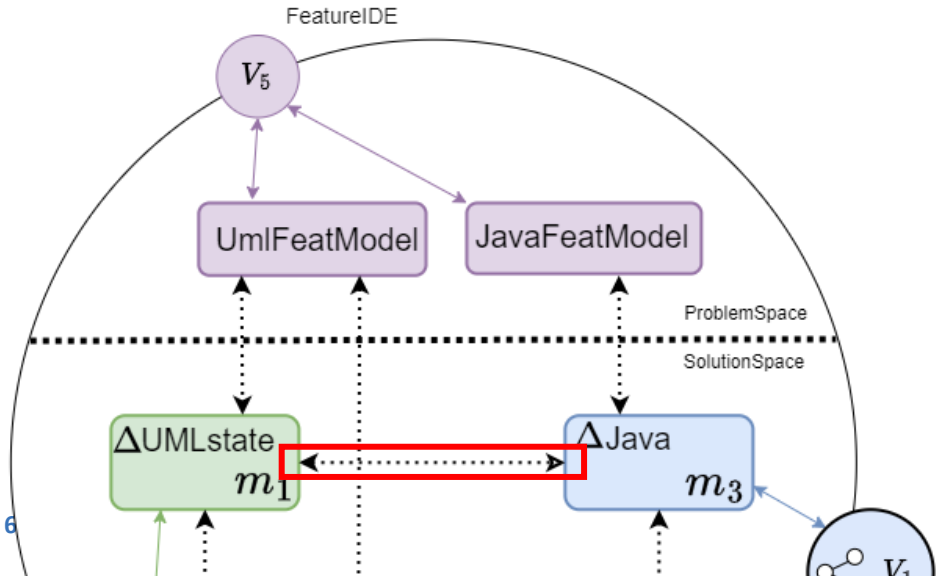




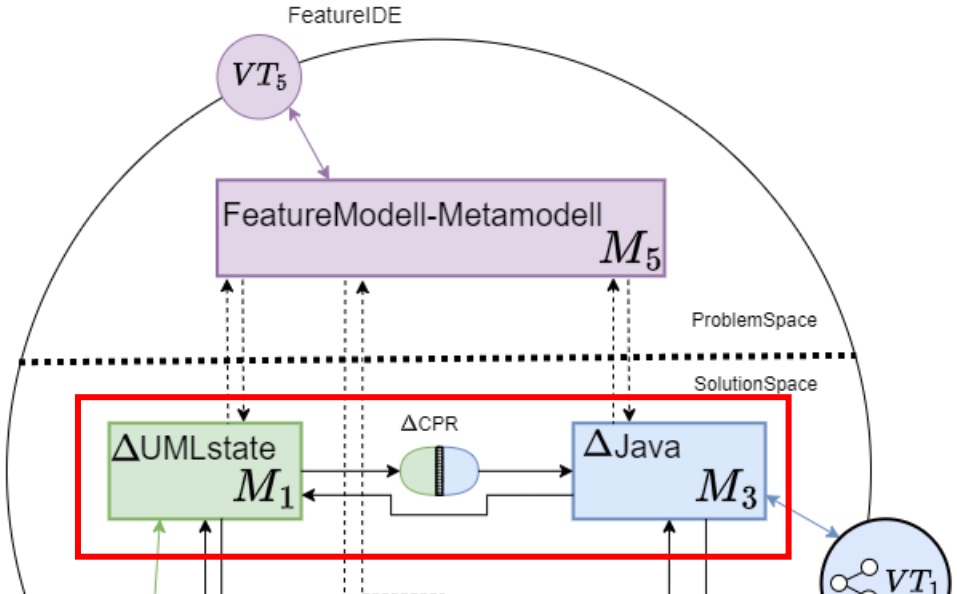
Consistency Categories

	FeatureModel	Metamodel	Model	DeltaLanguage	DeltaModel
FeatureModel	?				
Metamodel	?	Product-CPR			
Model	?	*instance of*	TraceLinks		
DeltaLanguage	?	?	?	Delta-CPR	
DeltaModel	?	?	apply-to/derive	*instance of*	TraceLinks, Order

Var-V-SUM



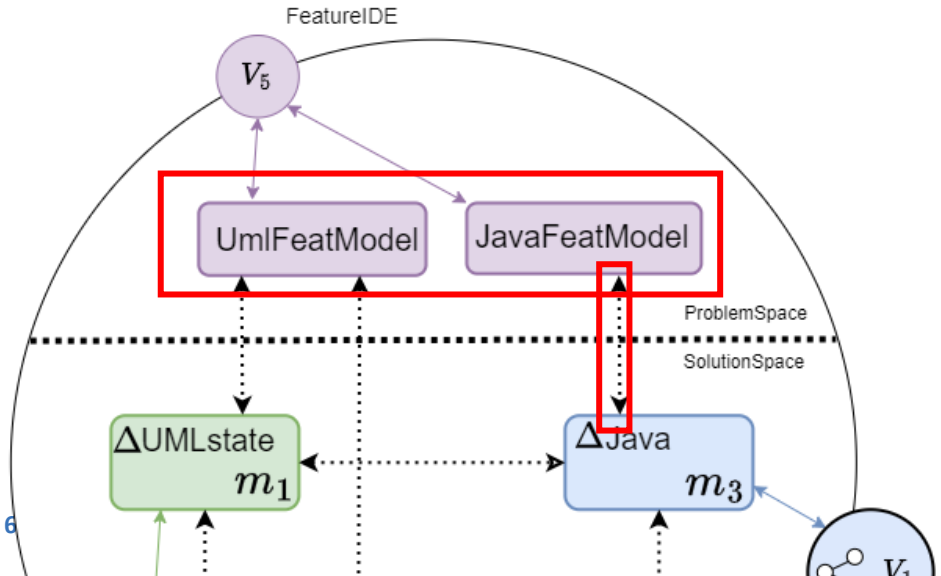
Var-V-SUMM



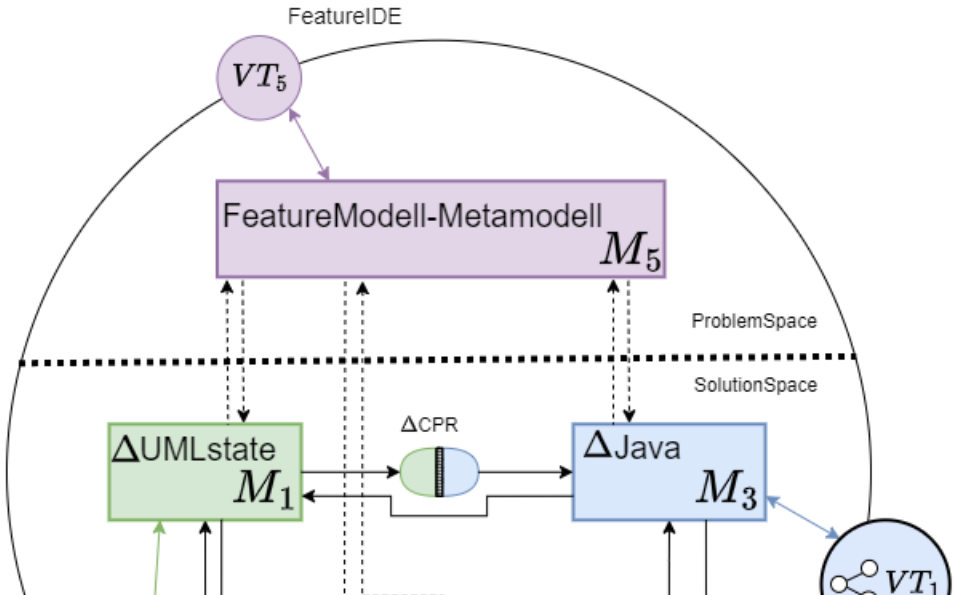
Consistency Categories

	FeatureModel	Metamodel	Model	DeltaLanguage	DeltaModel
FeatureModel	Implicit Constraints				
Metamodel	?	Product-CPR			
Model	?	*instance of*	TraceLinks		
DeltaLanguage	?	?	?	Delta-CPR	
DeltaModel	FeatureMapping	?	apply-to/derive	*instance of*	TraceLinks, Order

Var-V-SUM

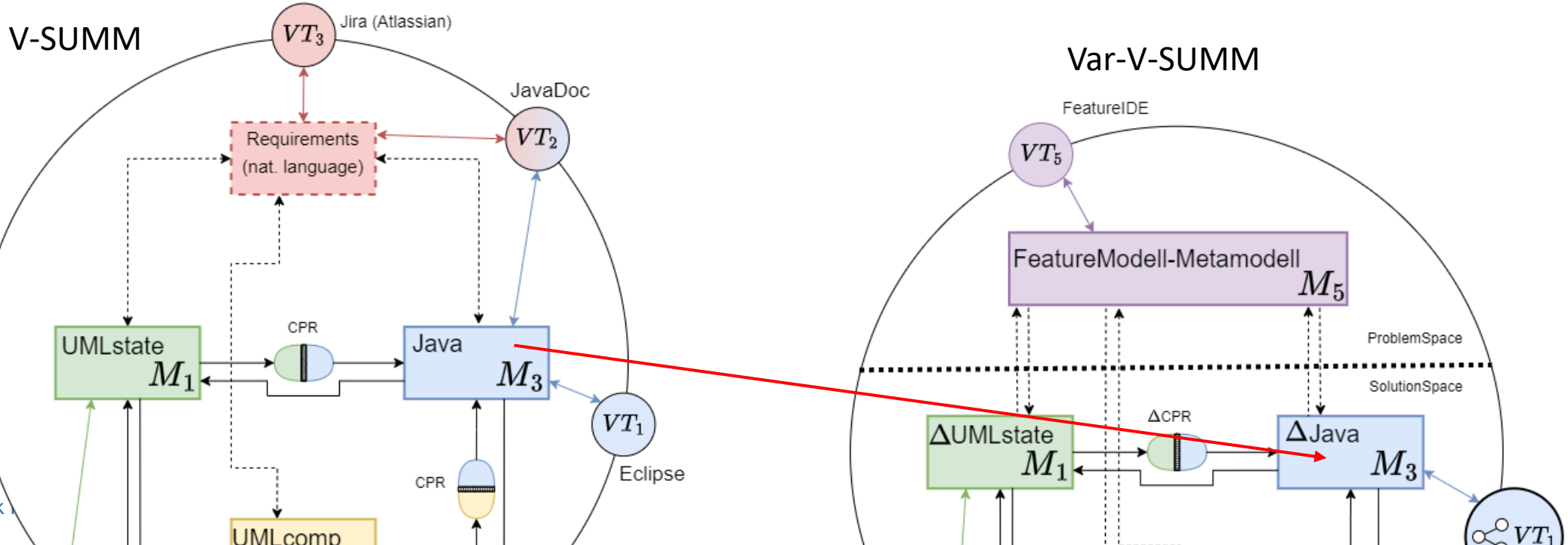


Var-V-SUMM



Consistency Categories

	FeatureModel	Metamodel	Model	DeltaLanguage	DeltaModel
FeatureModel	Implicit Constraints				
Metamodel	?	Product-CPR			
Model	?	*instance of*	TraceLinks		
DeltaLanguage	?	Synthesis	?	Delta-CPR	
DeltaModel	FeatureMapping	?	apply-to/derive	*instance of*	TraceLinks, Order



„The 3 pillars of my research“

How to combine variability and consistency? (Var-V-SUM)

How to work with a Var-V-SUM…

- product-oriented?
- plattform-oriented?

How can consistency be categorized? (Table)

„The 3 pillars of my research“

How to combine variability and consistency? (Var-V-SUM)

How to work with a Var-V-SUM…
- product-oriented?
- plattform-oriented?

How can consistency in variable systems be categorized? (Table)

My Questions for you

Which aspects are interesting?

What similar approaches or solutions do you know from the problem space of SPLE that could be applied here?

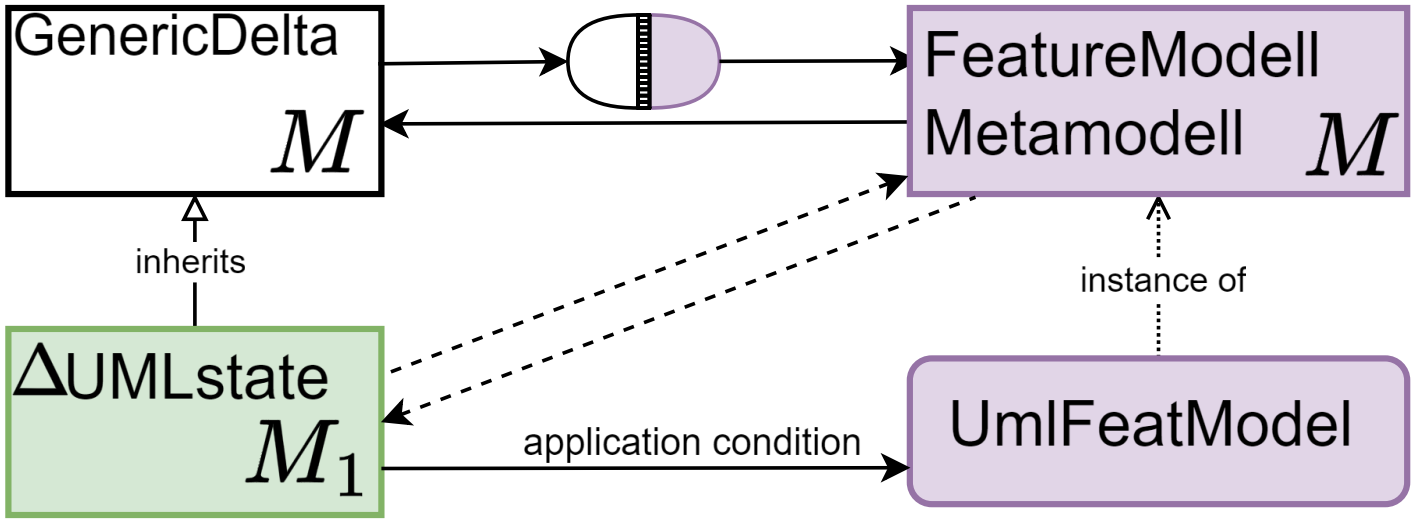
Was there something that you consider impossible or more complex than shown?

GEFÖRDERT DURCH DIE DFG Deutsche Forschungsgemeinschaft – SFB-1608 – 501798263

CONVIDE CRC 1608

Thank you! - Appendix

Connection FeatureModel & Deltas



Example CPR


```
import "statemachine_metamodel.ecore" as STMA;
import "component_metamodel.ecore" as COMP;
reaction TransitionAdded {
    after adding to STMA::Region[transitions]
    call addPort(affectedObject, newValue)
}
routine addPort(STMA::Region region, STMA::Transition trans){









}
```

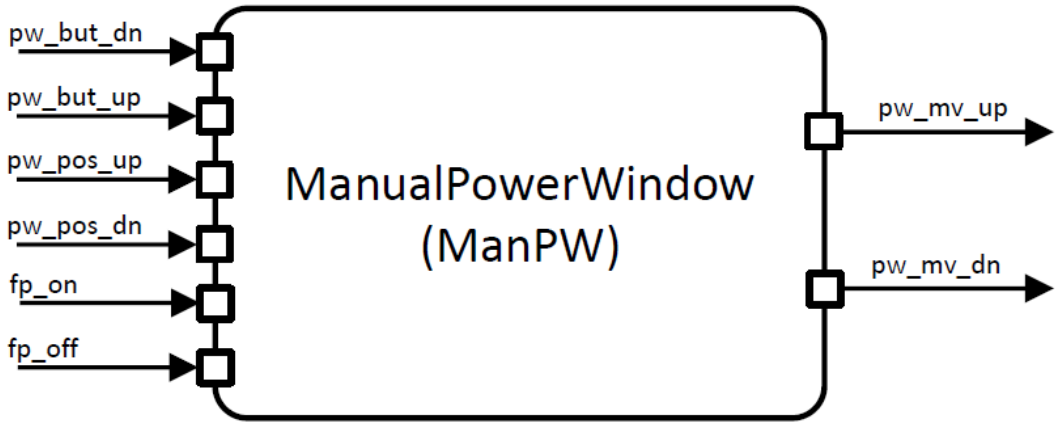
Example CPR


```
import "statemachine_metamodel.ecore" as STMA;
import "component_metamodel.ecore" as COMP;
reaction TransitionAdded {
    after adding to STMA::Region[transitions]
    call addPort(affectedObject, newValue)
}
routine addPort(STMA::Region region, STMA::Transition trans){
    match {
        assert(trans.trigger != null);
        val comp = retrieve COMP::Component corresponding to region;
       assert(comp.hasNoInput(trans.trigger));
    }



}
```

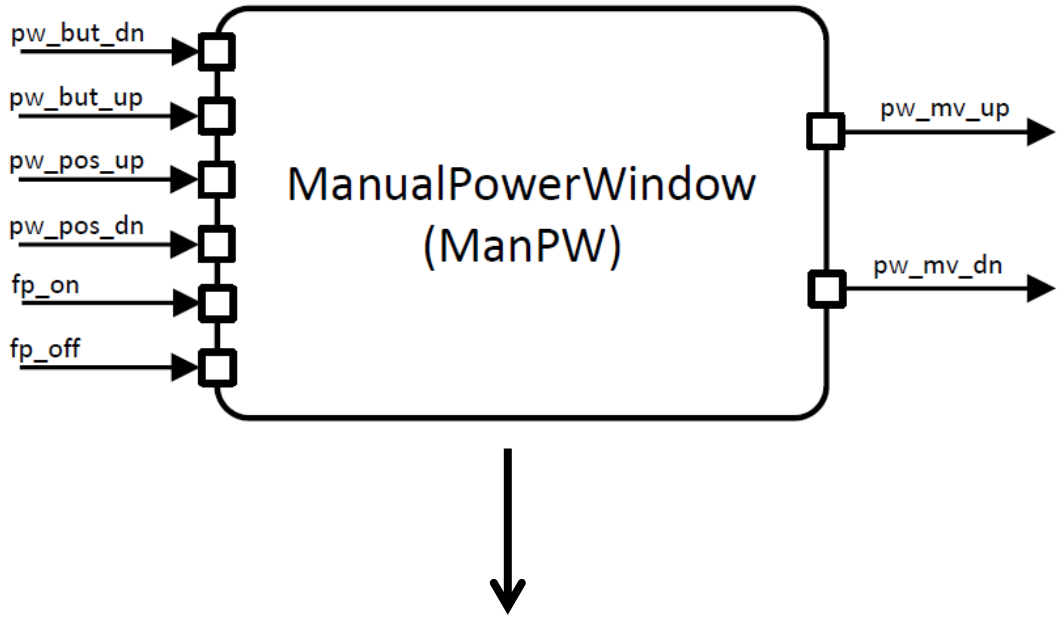
Example CPR


```
import "statemachine_metamodel.ecore" as STMA;
import "component_metamodel.ecore" as COMP;
reaction TransitionAdded {
    after adding to STMA::Region[transitions]
    call addPort(affectedObject, newValue)
}
routine addPort(STMA::Region region, STMA::Transition trans){
    match {
        assert(trans.trigger != null);
        val comp = retrieve COMP::Component corresponding to region;
       assert(comp.hasNoInput(trans.trigger));
    }
    update {
        val newPort = new COMP::Port(trans.signal);
        comp.inputPorts.add(newPort);
}   }
```

Example Delta

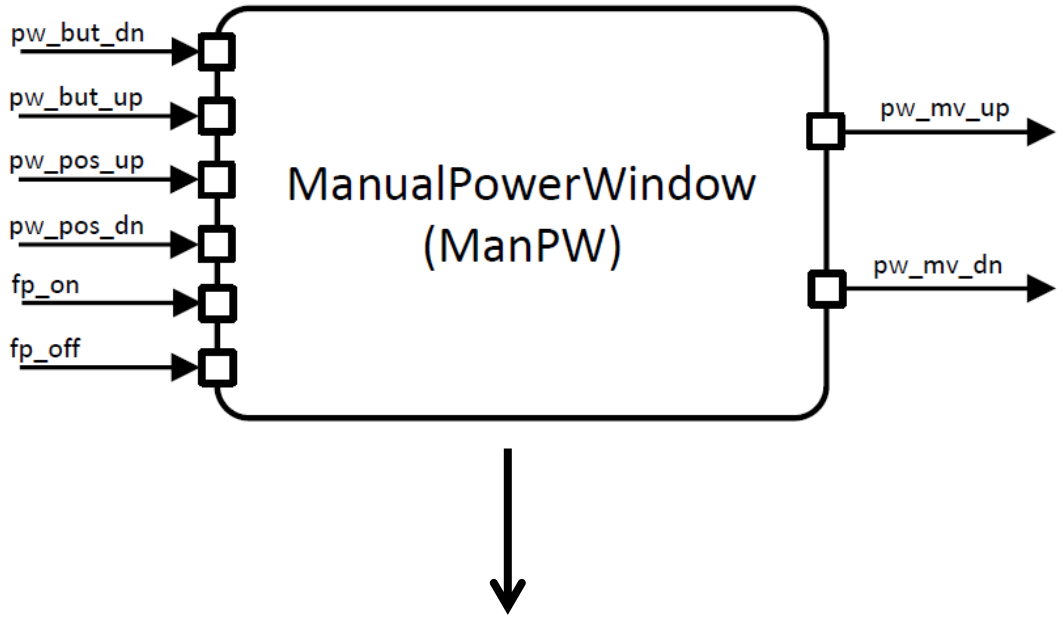


Example Delta



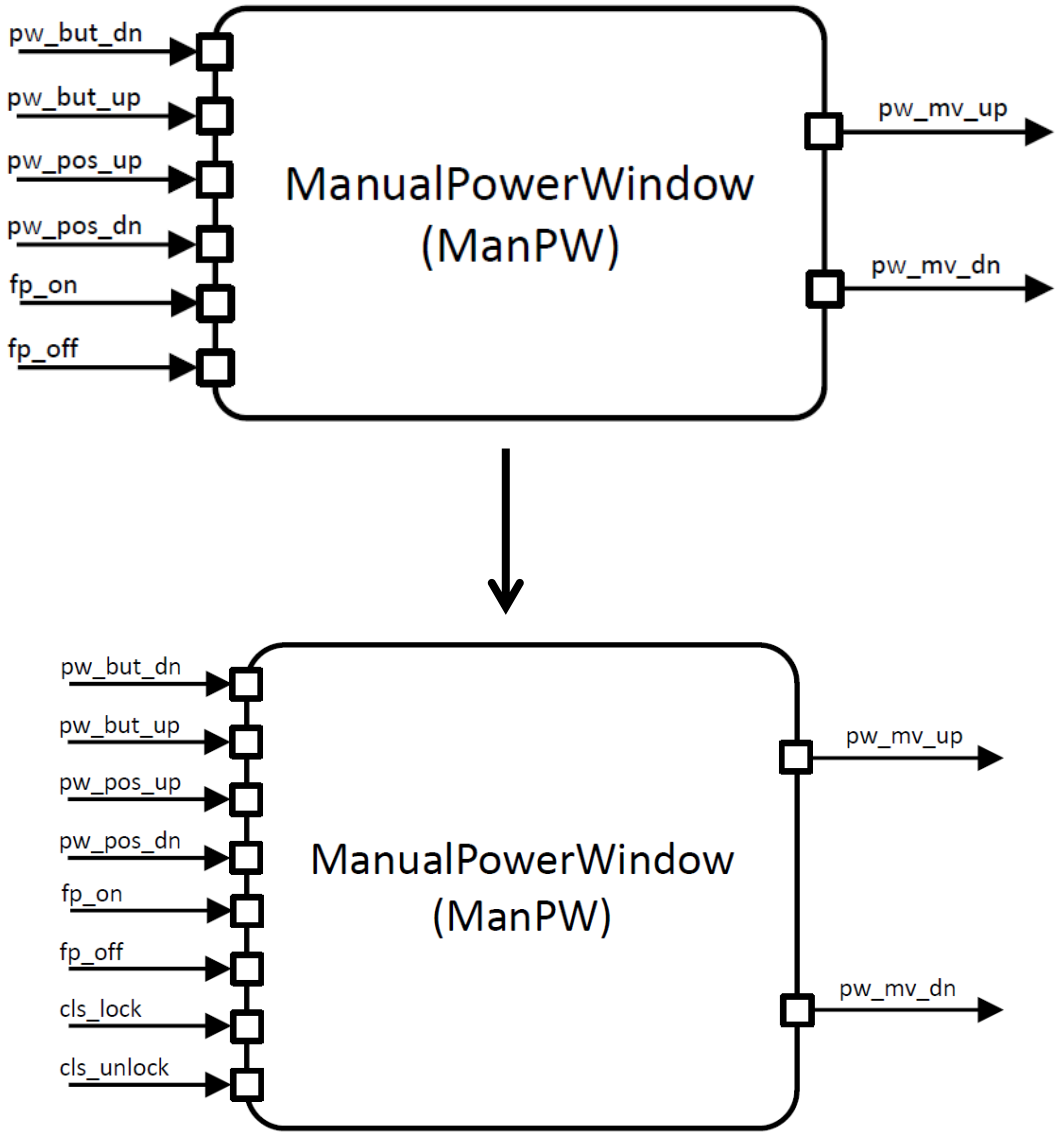

```
ComponentDelta: Component_PW_Man2Auto {
    rem_connector(cnt1)
    rem_connector(cnt2)
    rem_port(p1)
    rem_port(p2)
    rem_signal(pw_mv_up)
    rem_signal(pw_mv_dn)

}
```

Example Delta




```
ComponentDelta: Component_PW_Man2Auto {
    rem_connector(cnt1)
    rem_connector(cnt2)
    rem_port(p1)
    rem_port(p2)
    rem_signal(pw_mv_up)
    rem_signal(pw_mv_dn)

    add_signal(pw_auto_mv_up)
    add_signal(pw_auto_mv_dn)
    add_signal(pw_auto_mv_stop)
    add_port(p3,comp1,pw_auto_mv_up,out)
    add_port(p4,comp1,pw_auto_mv_dn,out)
    add_port(p5,comp1,pw_auto_mv_stop,out)
    add_connector(cnt3,p3,ENV)
    add_connector(cnt4,p4,ENV)
    add_connector(cnt5,p5,ENV)



}
```

Example Delta




```
ComponentDelta: Component_PW_Man2Auto {
    rem_connector(cnt1)
    rem_connector(cnt2)
    rem_port(p1)
    rem_port(p2)
    rem_signal(pw_mv_up)
    rem_signal(pw_mv_dn)

    add_signal(pw_auto_mv_up)
    add_signal(pw_auto_mv_dn)
    add_signal(pw_auto_mv_stop)
    add_port(p3,comp1,pw_auto_mv_up,out)
    add_port(p4,comp1,pw_auto_mv_dn,out)
    add_port(p5,comp1,pw_auto_mv_stop,out)
    add_connector(cnt3,p3,ENV)
    add_connector(cnt4,p4,ENV)
    add_connector(cnt5,p5,ENV)

    mod_comp(comp1,name,"AutoPW")
}
```

Whole Approach in one figure