Towards High-Level Models for Cloud Computing Systems

13th of December, 2023

Journées VELVET @ IMT Atlantique

Simon Bliudze, Inria Lille

Satellite software design

A collaboration with the EPFL Space Engineering Center

Component-based design in BIP of the control software for a nano-satellite

Control and Data Management System (CDMS)

Communication with other subsystems through an I^2C bus

A collaboration with ThalesAlenia Space (France) and Aristotle University of Thessaloniki (Greece)

"Catalogue of System and Software Properties"

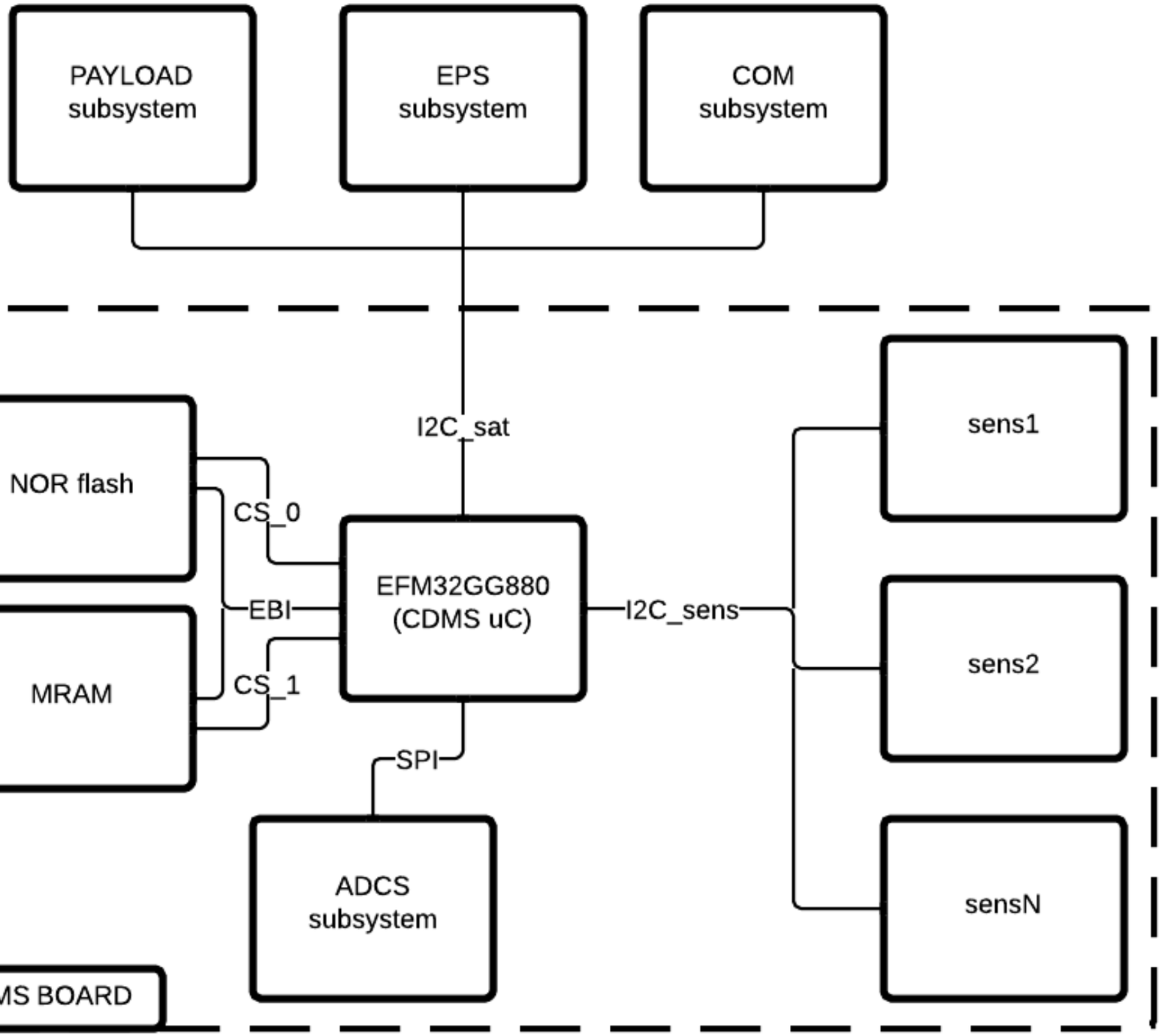
Funded by ESA

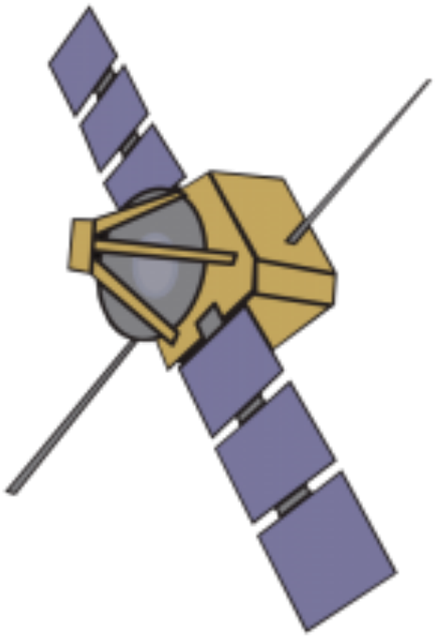
eSpace SPACE ENGINEERING CENTER

ThalesAlenia Space — A Thales / Finmeccanica Company

ARISTOTLE UNIVERSITY OF THESSALONIKI

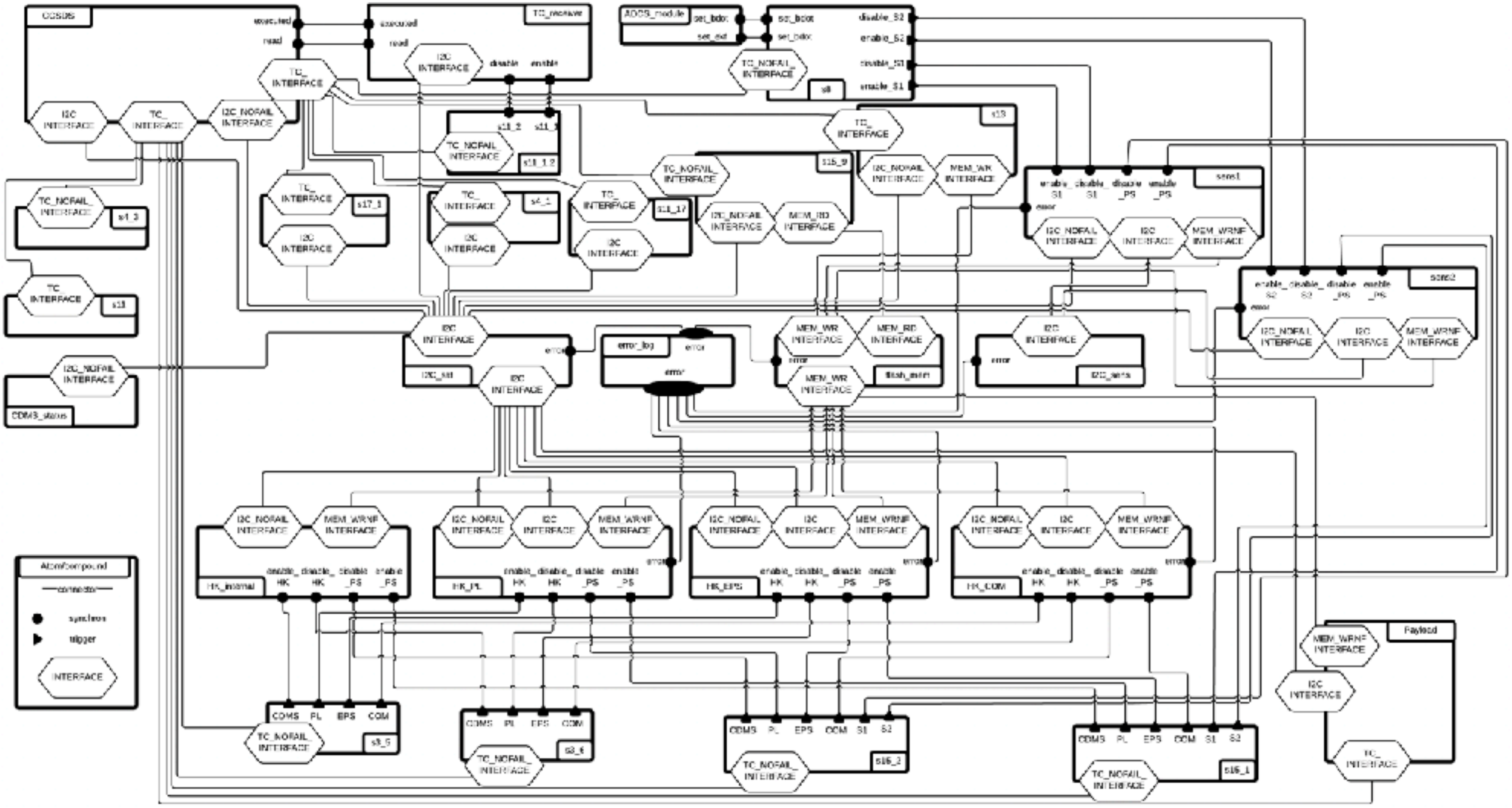
CubETH: CDMS architecture



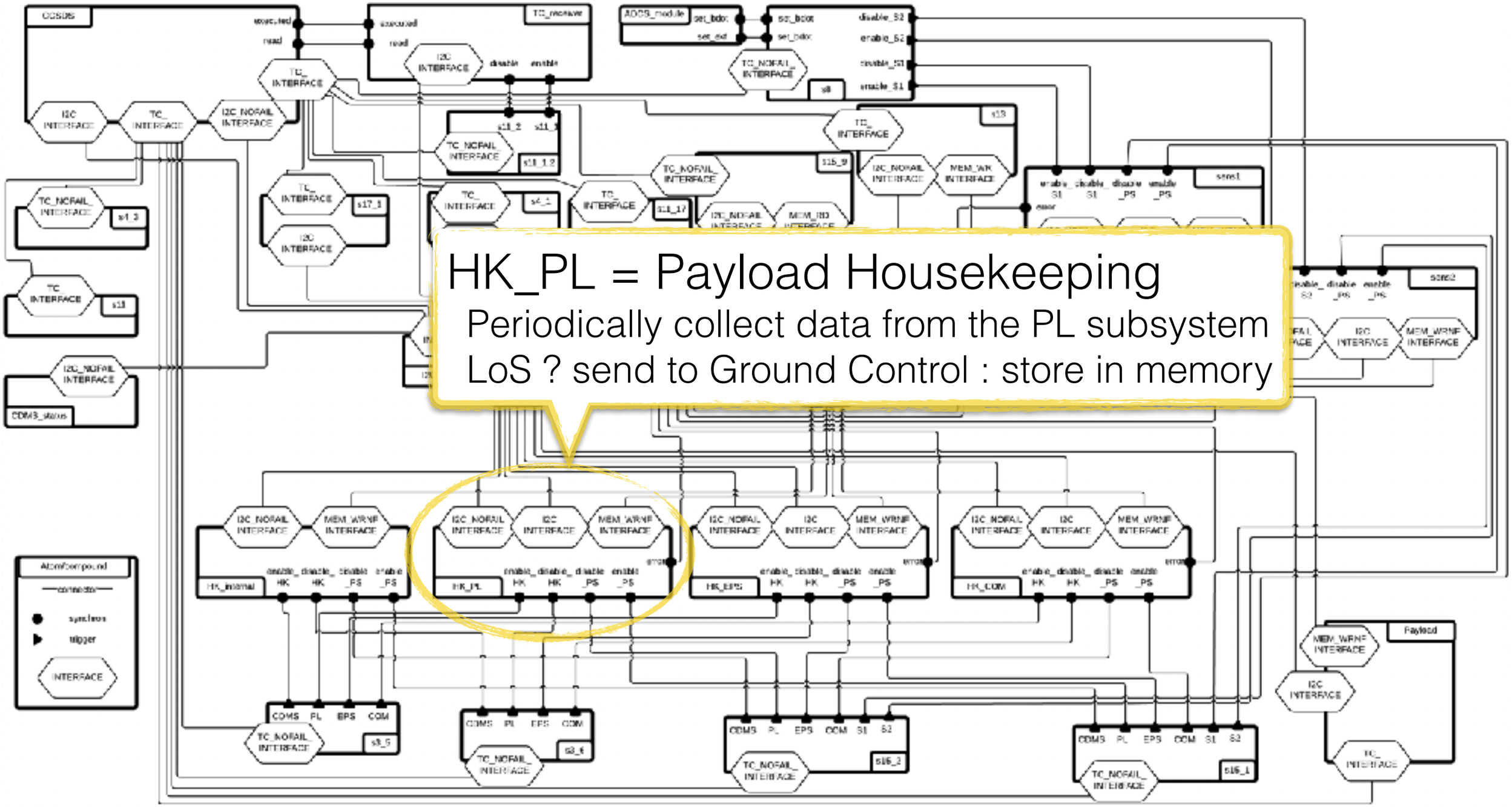


(figure courtesy of Marco Pagnamenta)

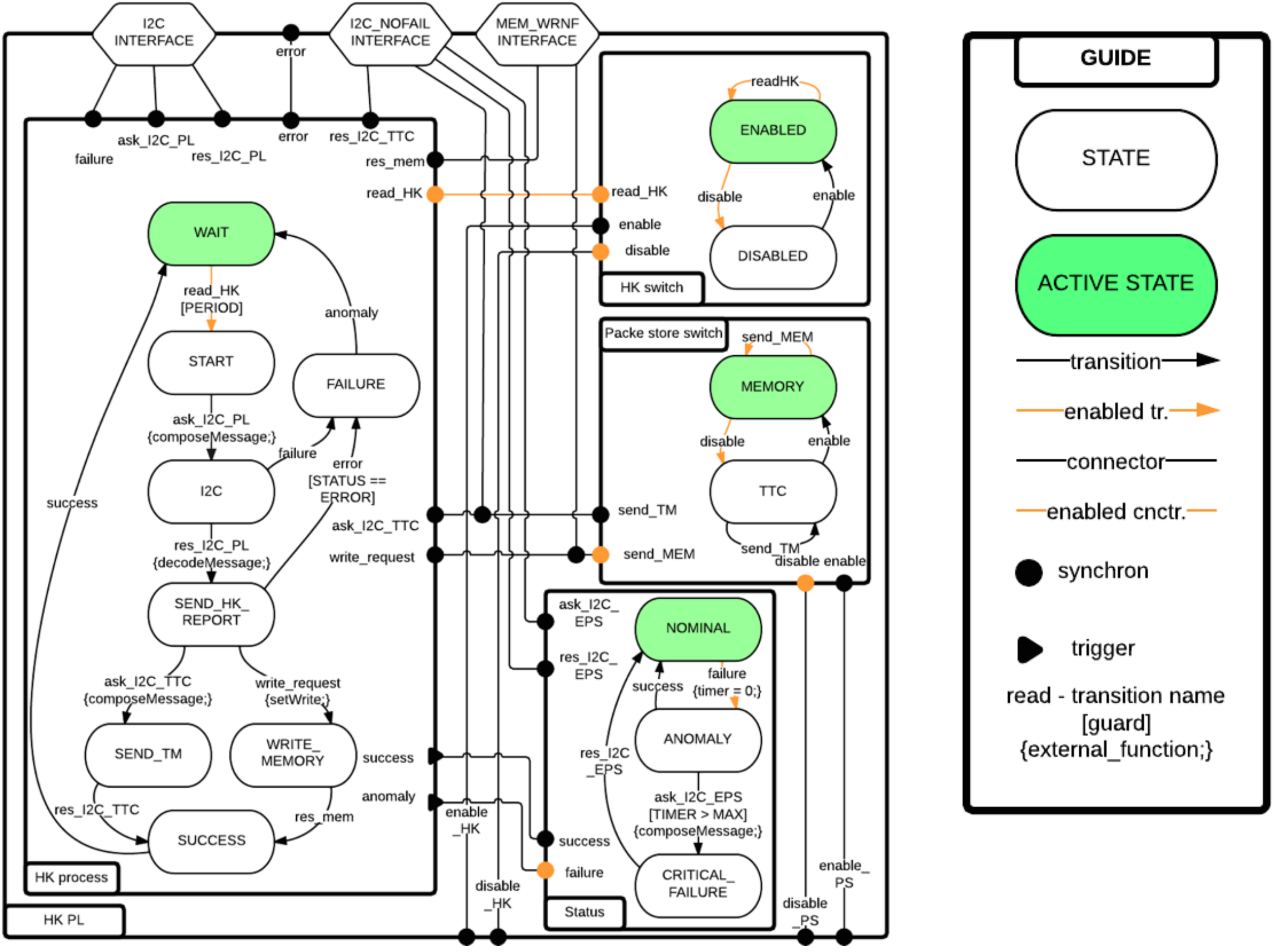
CubETH: CDMS architecture

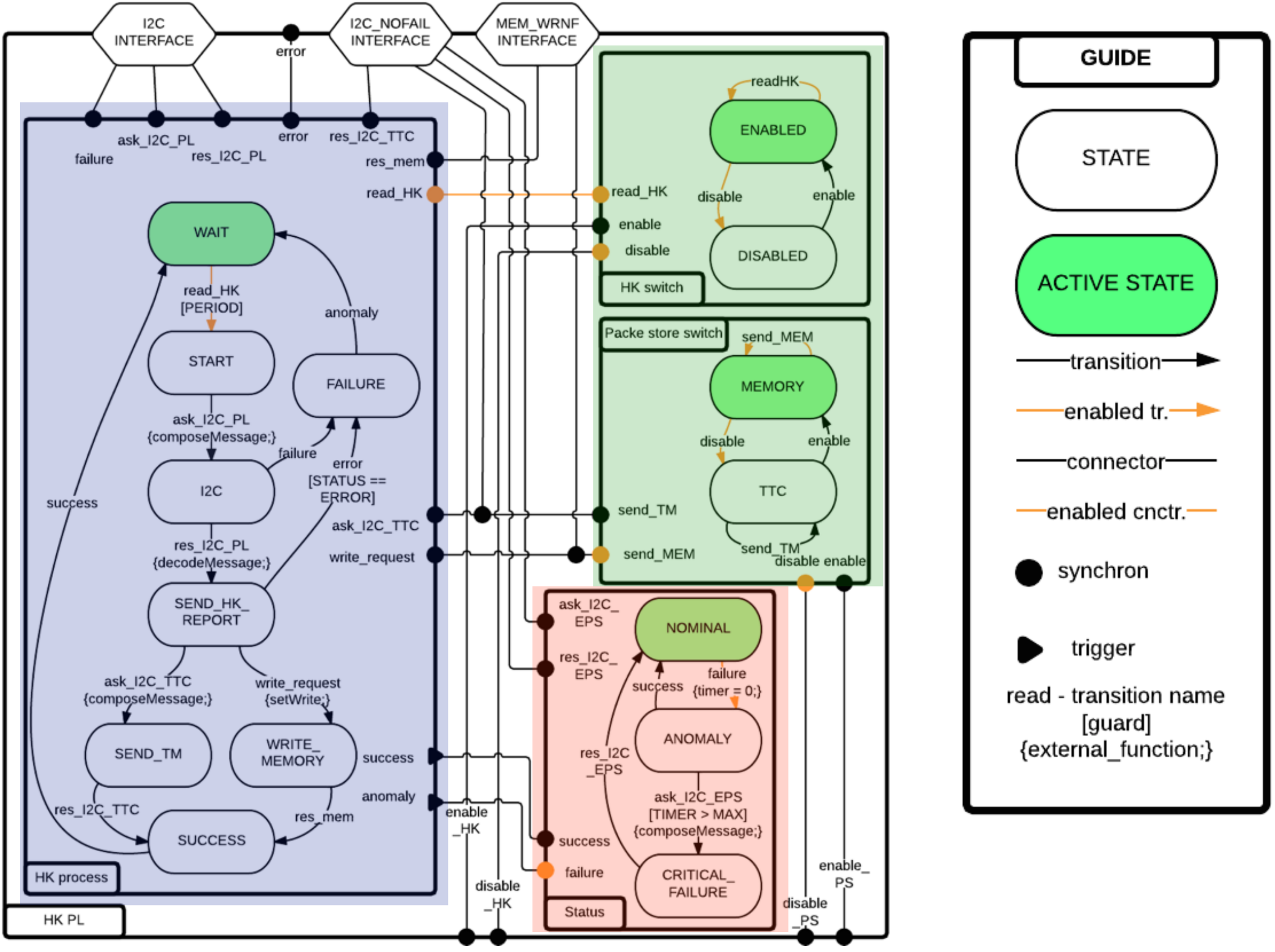


CubETH: CDMS architecture

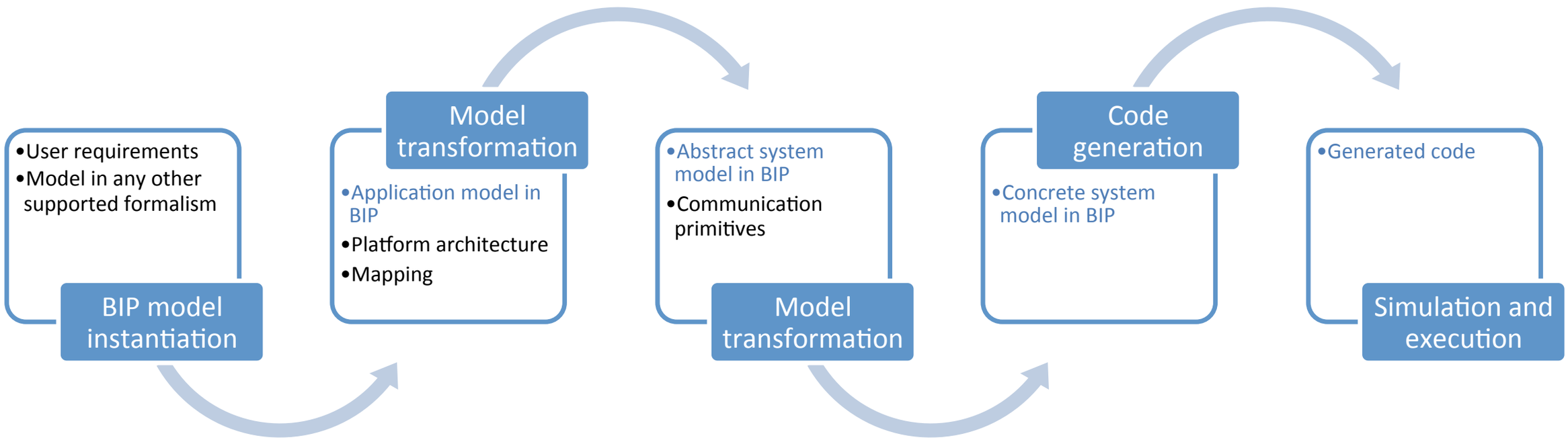


(figure courtesy of Marco Pagnamenta)


I2C INTERFACE
I2C_NOFAIL INTERFACE
MEM_WRNF INTERFACE
error
failure
ask_I2C_PL
res_I2C_PL
error
res_I2C_TTC
res_mem
read_HK
WAIT
read_HK [PERIOD]
START
ask_I2C_PL {composeMessage;}
I2C
failure
error [STATUS == ERROR]
FAILURE
anomaly
success
res_I2C_PL {decodeMessage;}
SEND_HK_ REPORT
ask_I2C_TTC {composeMessage;}
write_request {setWrite;}
SEND_TM
WRITE_ MEMORY
success
anomaly
res_I2C_TTC
res_mem
SUCCESS
HK process
HK PL
ask_I2C_TTC
write_request
enable_HK
disable_HK
read_HK
enable
disable
HK switch
readHK
ENABLED
disable
enable
DISABLED
Packe store switch
send_MEM
MEMORY
disable
enable
TTC
send_TM
send_MEM
send_TM
disable enable
ask_I2C_ EPS
res_I2C_ EPS
NOMINAL
success
failure {timer = 0;}
ANOMALY
res_I2C _EPS
ask_I2C_EPS [TIMER > MAX] {composeMessage;}
CRITICAL_ FAILURE
success
failure
Status
disable_PS
enable_ PS
GUIDE
STATE
ACTIVE STATE
transition
enabled tr.
connector
enabled cnctr.
synchron
trigger
read - transition name
[guard]
{external_function;}


I2C INTERFACE
I2C_NOFAIL INTERFACE
MEM_WRNF INTERFACE
error
ask_I2C_PL
failure
res_I2C_PL
error
res_I2C_TTC
res_mem
read_HK
WAIT
read_HK [PERIOD]
START
ask_I2C_PL {composeMessage;}
I2C
failure
error [STATUS == ERROR]
FAILURE
anomaly
success
ask_I2C_TTC
res_I2C_PL {decodeMessage;}
write_request
SEND_HK_ REPORT
ask_I2C_TTC {composeMessage;}
write_request {setWrite;}
SEND_TM
WRITE_ MEMORY
success
anomaly
res_I2C_TTC
res_mem
SUCCESS
HK process
HK PL
enable_HK
disable_HK
read_HK
enable
disable
readHK
ENABLED
disable
enable
DISABLED
HK switch
Packe store switch
send_MEM
MEMORY
disable
enable
TTC
send_TM
send_MEM
send_TM
disable enable
ask_I2C_EPS
NOMINAL
res_I2C_EPS
success
failure {timer = 0;}
ANOMALY
res_I2C_EPS
ask_I2C_EPS [TIMER > MAX] {composeMessage;}
success
failure
CRITICAL_ FAILURE
Status
enable_PS
disable_PS
GUIDE
STATE
ACTIVE STATE
transition
enabled tr.
connector
enabled cnctr.
synchron
trigger
read - transition name
[guard]
{external_function;}

Rigorous System Design flow

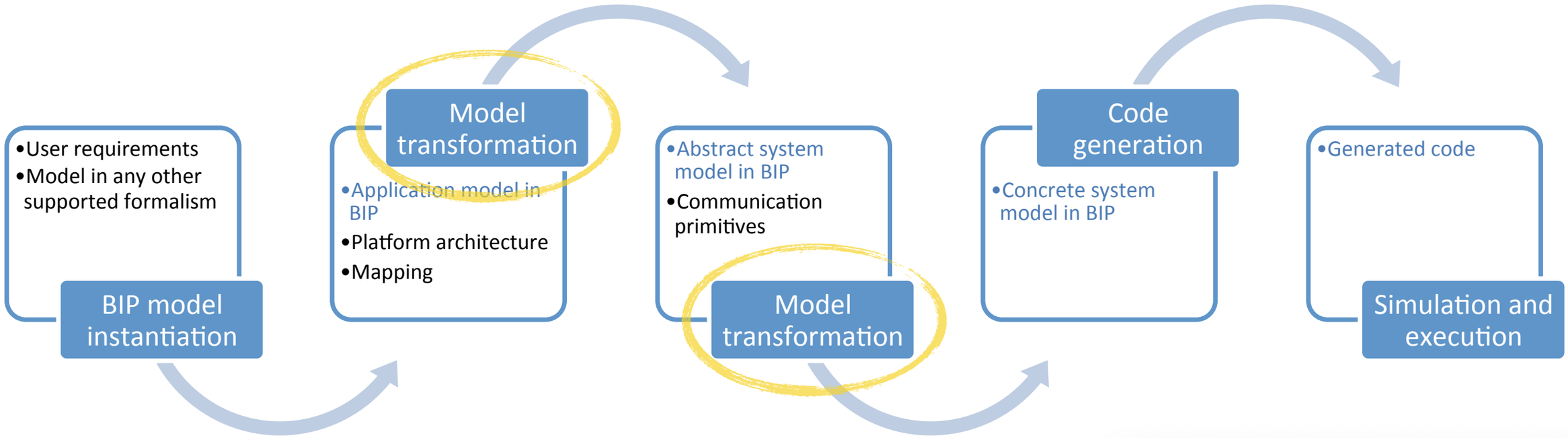


A series of semantics-preserving transformations

Correctness decomposed into
- correctness of transformations
- correctness of high-level models

Final implementation is **correct by construction**

Rigorous System Design flow



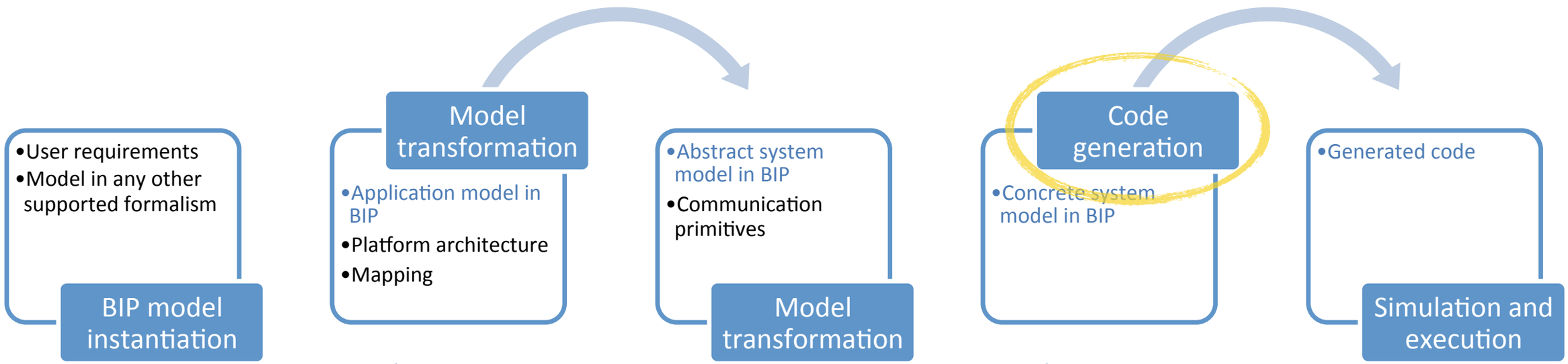
A series of semantics-preserving transformations

Correctness decomposed into
- correctness of transformations
- correctness of high-level models

Final implementation is **correct by construction**

- ☐ Unifying modelling framework

Rigorous System Design flow



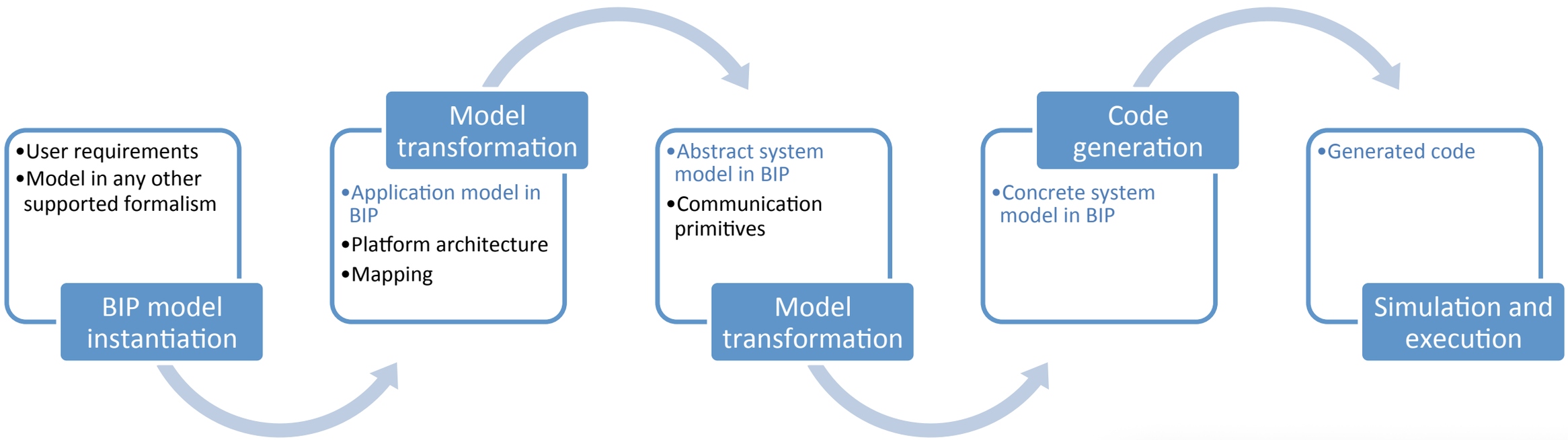
A series of semantics-preserving transformations

Correctness decomposed into
- correctness of transformations
- correctness of high-level models

Final implementation is **correct by construction**

- ☐ Unifying modelling framework
- ☐ Operational semantics

Rigorous System Design flow



A series of semantics-preserving transformations

Correctness decomposed into

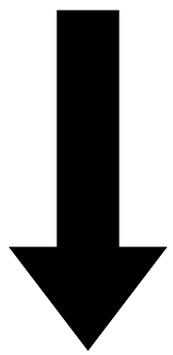
correctness of transformations

correctness of high-level models

Final implementation is **correct by construction**

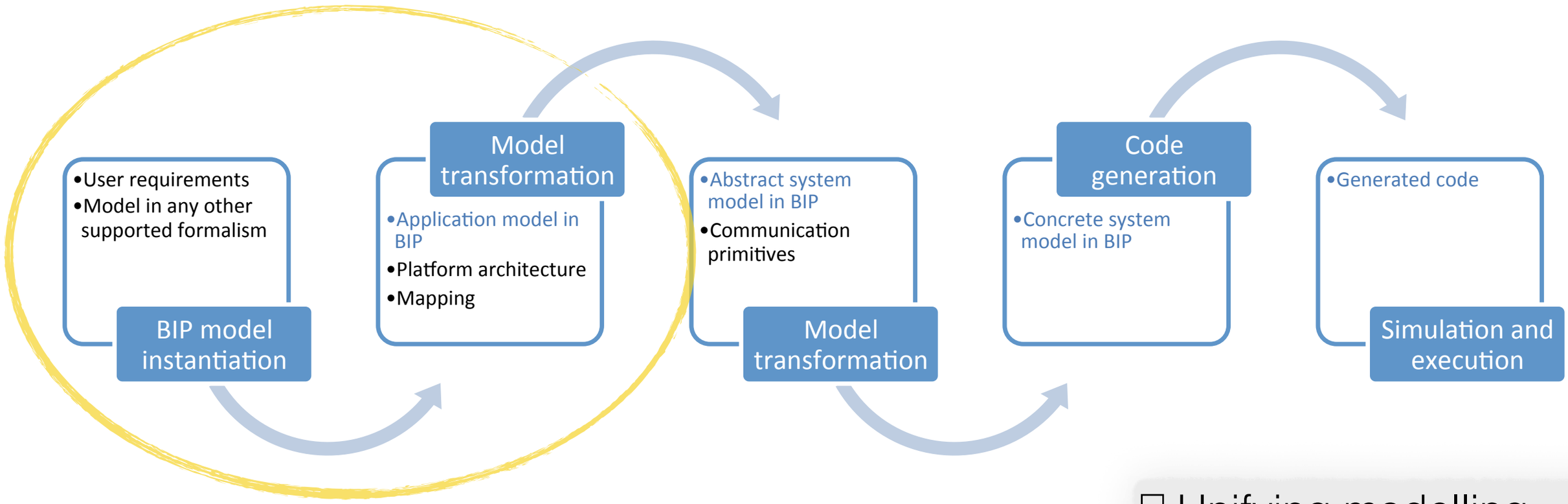
- ☐ Unifying modelling framework
- ☐ Operational semantics
- ☐ Method(s) to design correct models

Embedded systems



Cloud computing systems

Rigorous System Design flow



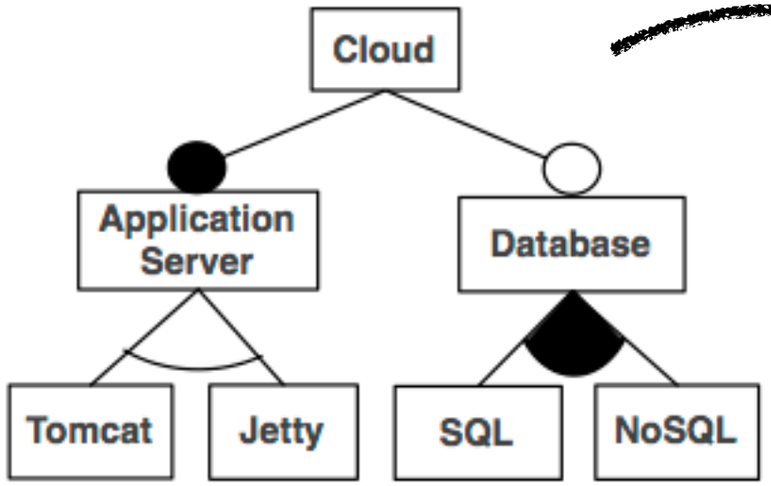
A series of semantics-preserving transformations

Correctness decomposed into
- correctness of transformations
- correctness of high-level models

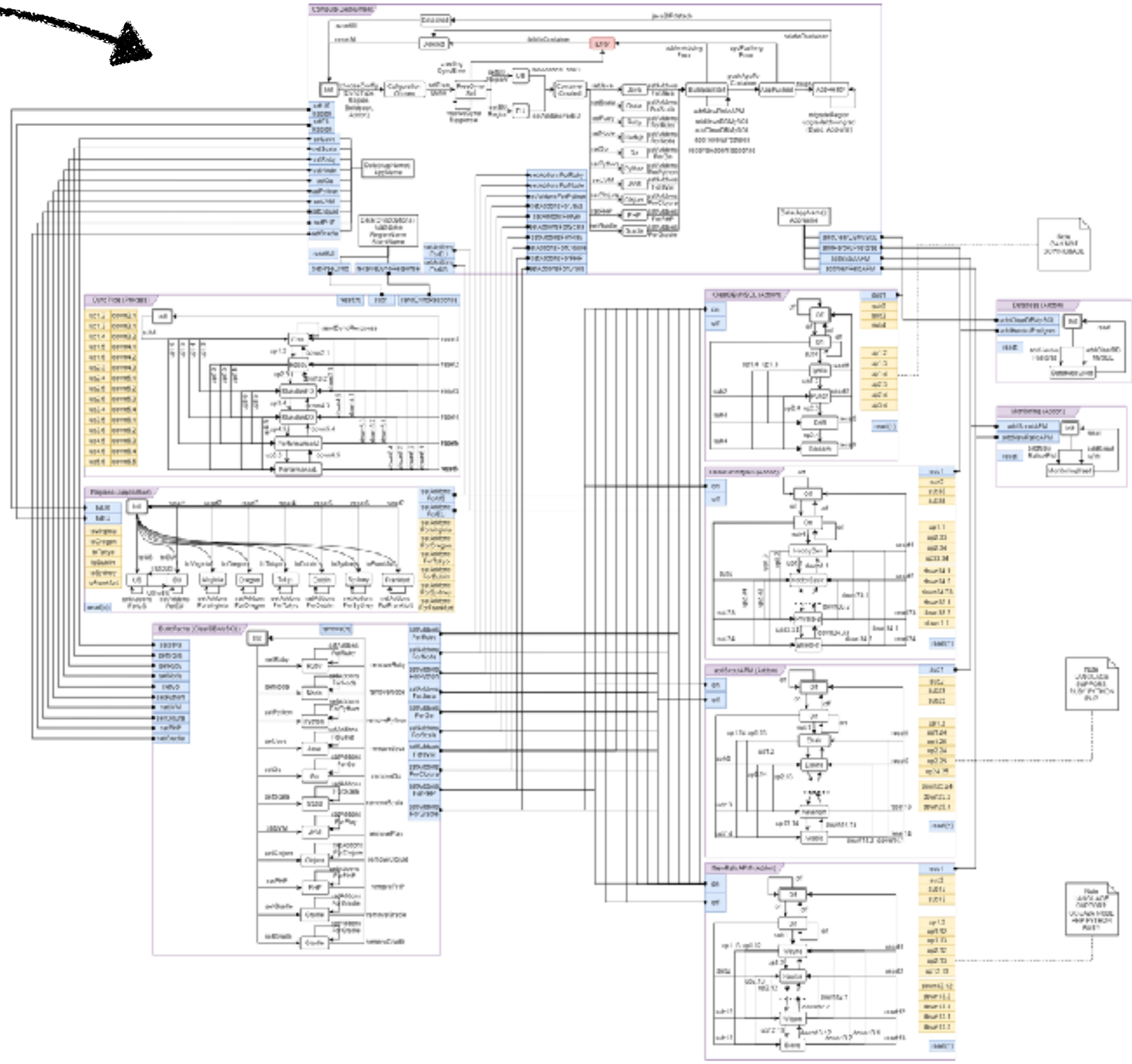
Final implementation is **correct by construction**

- ☐ Unifying modelling framework
- ☐ Operational semantics
- ☐ Method(s) to design correct models

Example: Synthesis from Feature Models



Feature Model A

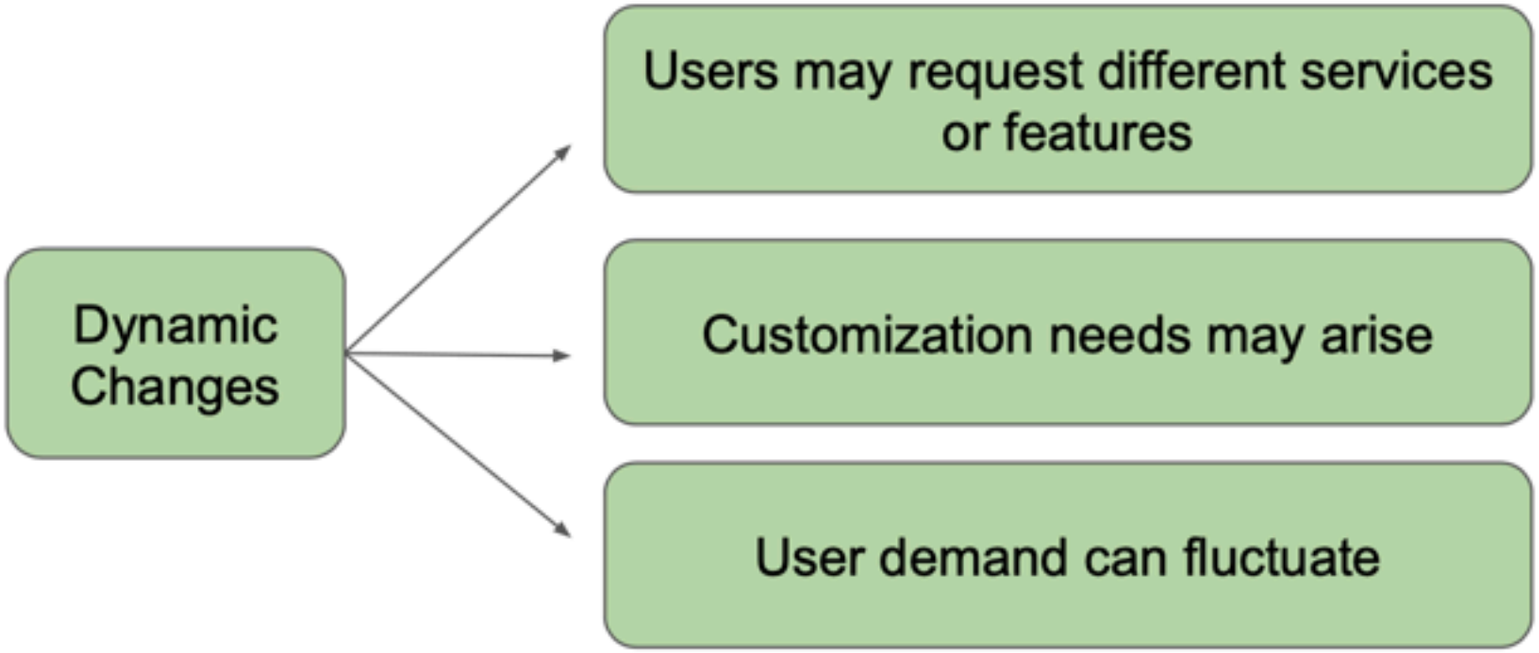


Salman Farhat
[COORDINATION 2023]

Reconfiguration

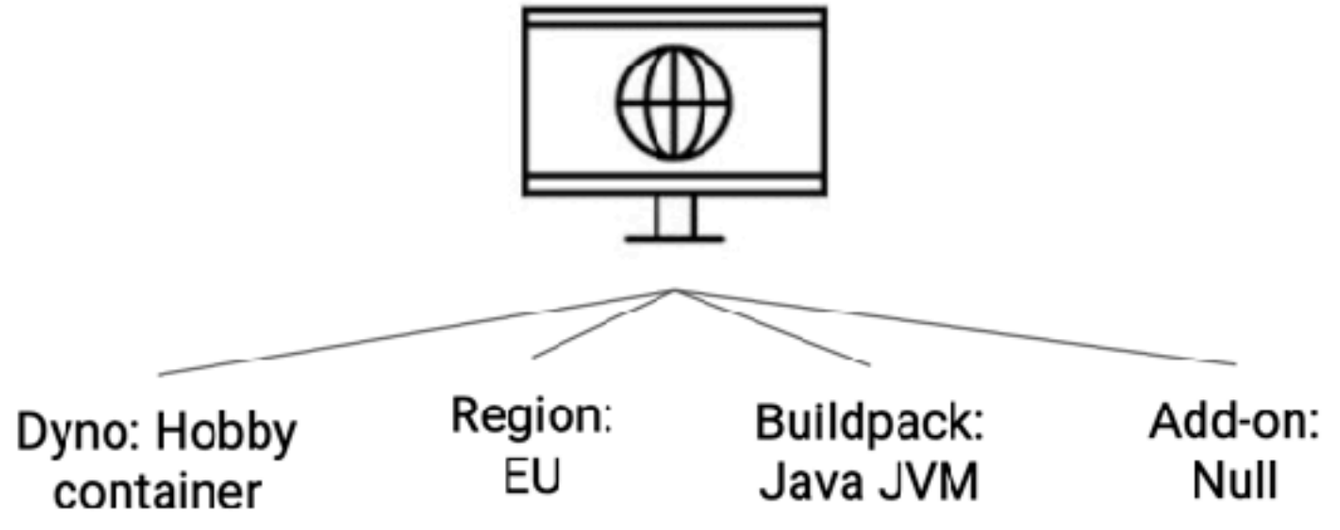
Initial configuration is meant to support initial user requirements and platform constraints

What if



Reconfiguration is necessary to change the system configuration in response to dynamic changes.

Configuration validity



Platform Constraints

- ★ A service (e.g. `Guru301`) is **only available** in the `US` Region
- ★ An Application must have at most one region

Existing approaches

Precomputing and storing the valid configurations → Memory Overhead

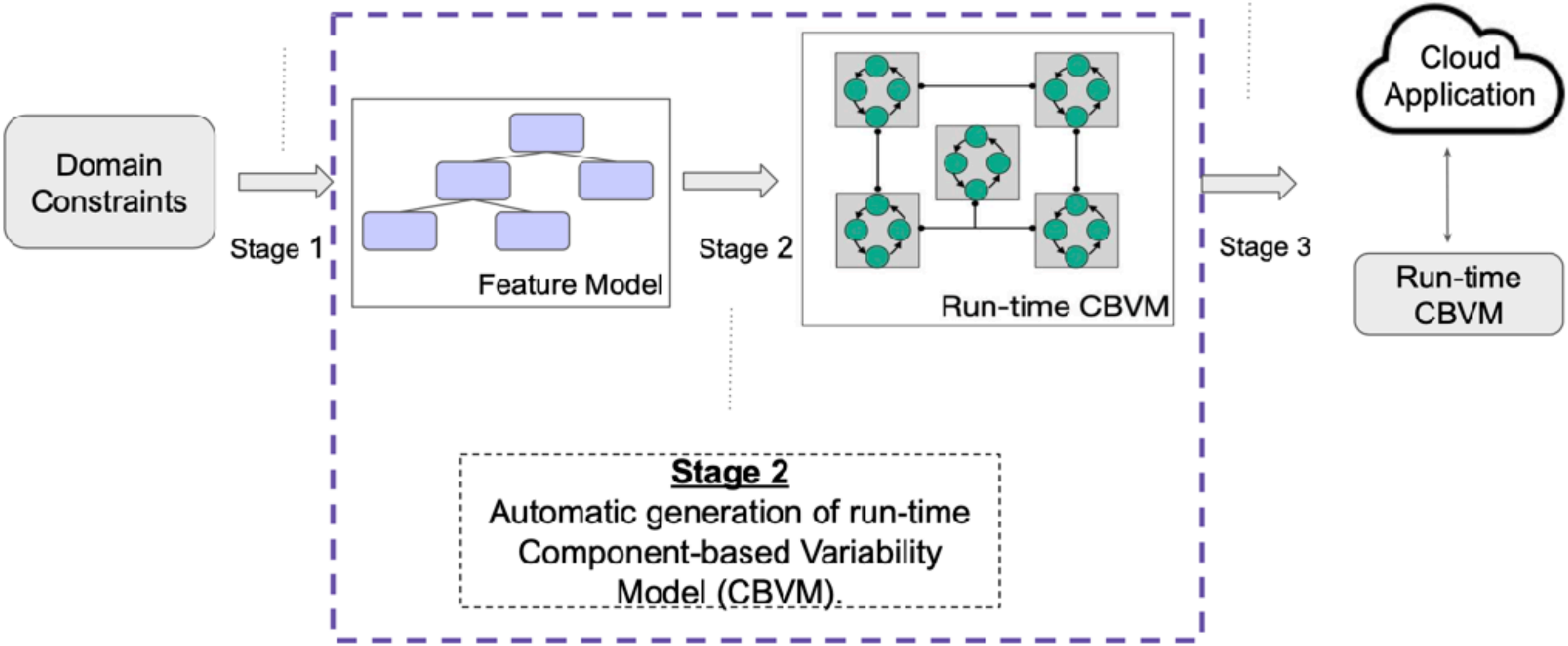
Run-time validation → Time Overhead

How can we do better?

Approach overview

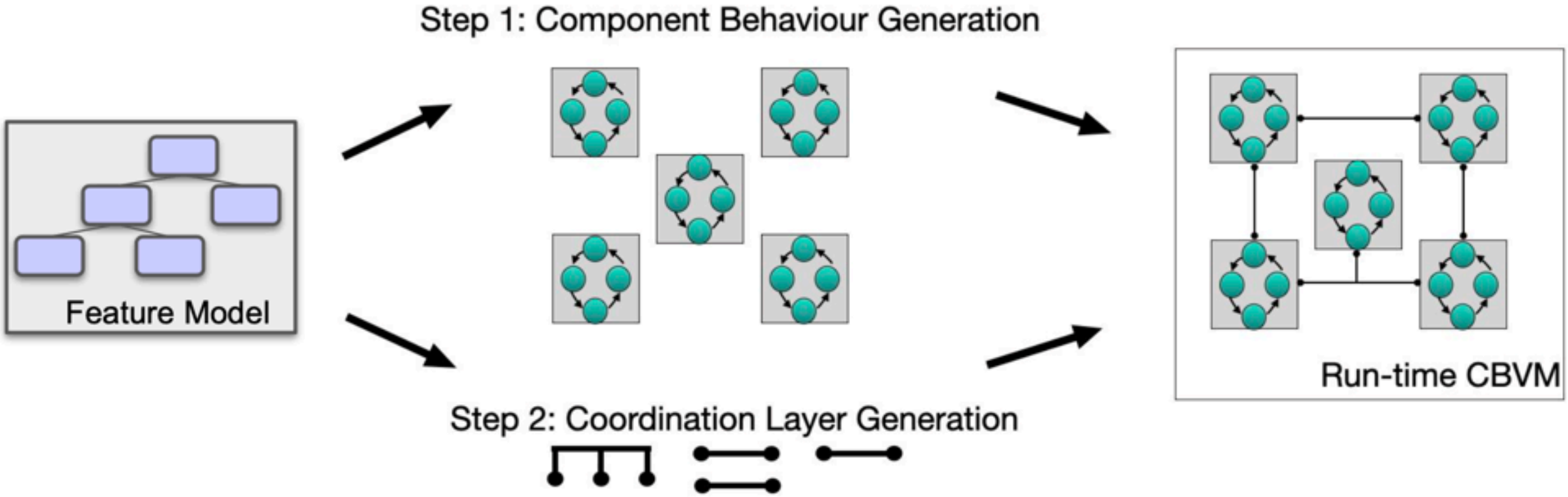
Stage 1
Domain constraints are specified as a feature model.

Stage 3
Integration of the generated model.

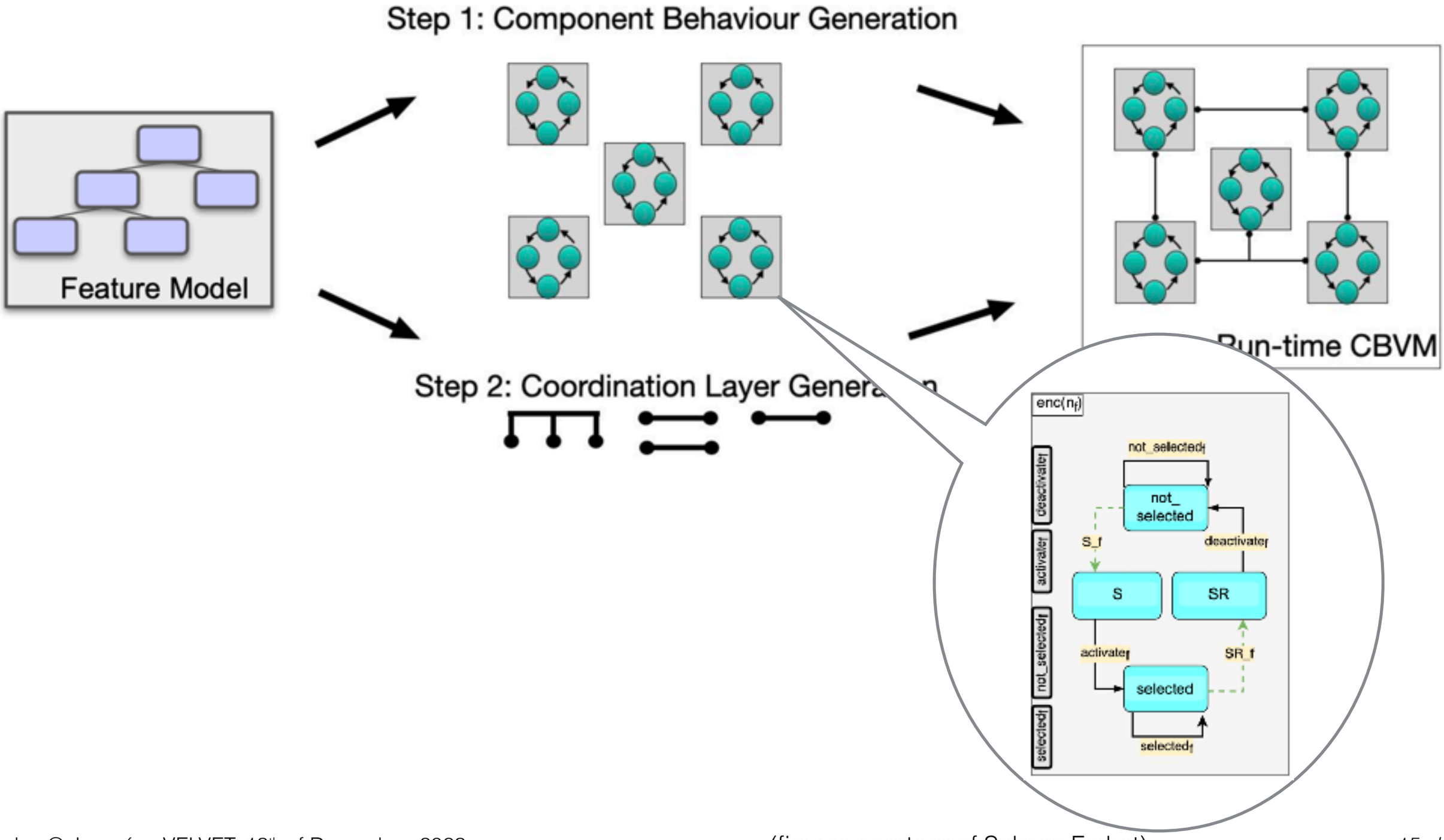


Stage 2
Automatic generation of run-time Component-based Variability Model (CBVM).

Run-time Component-Based Variability Model

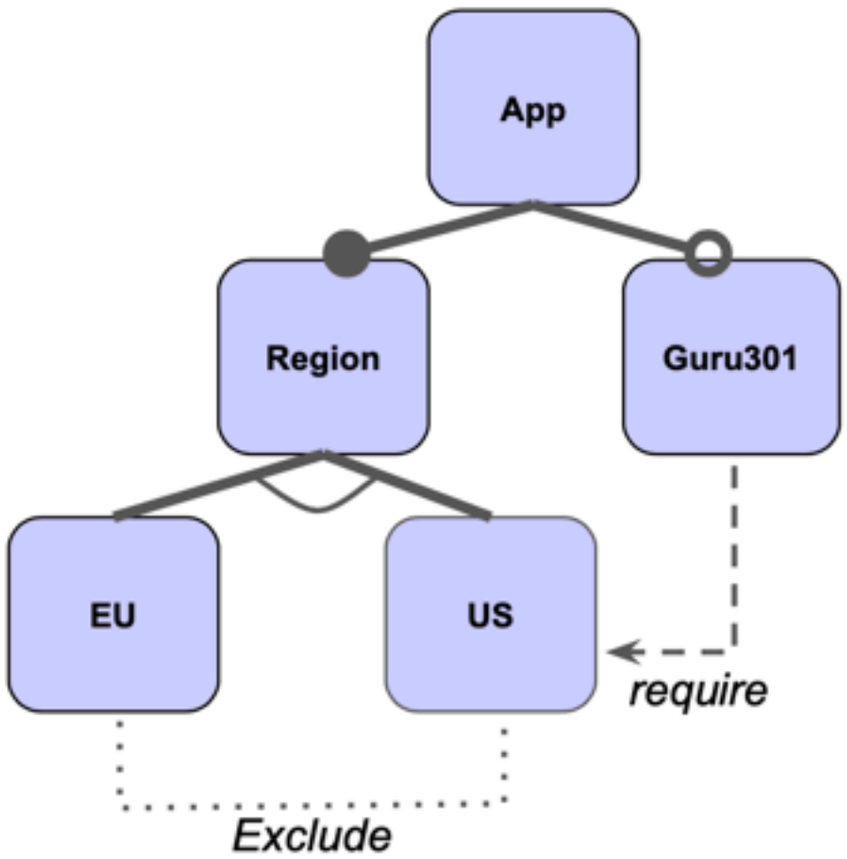


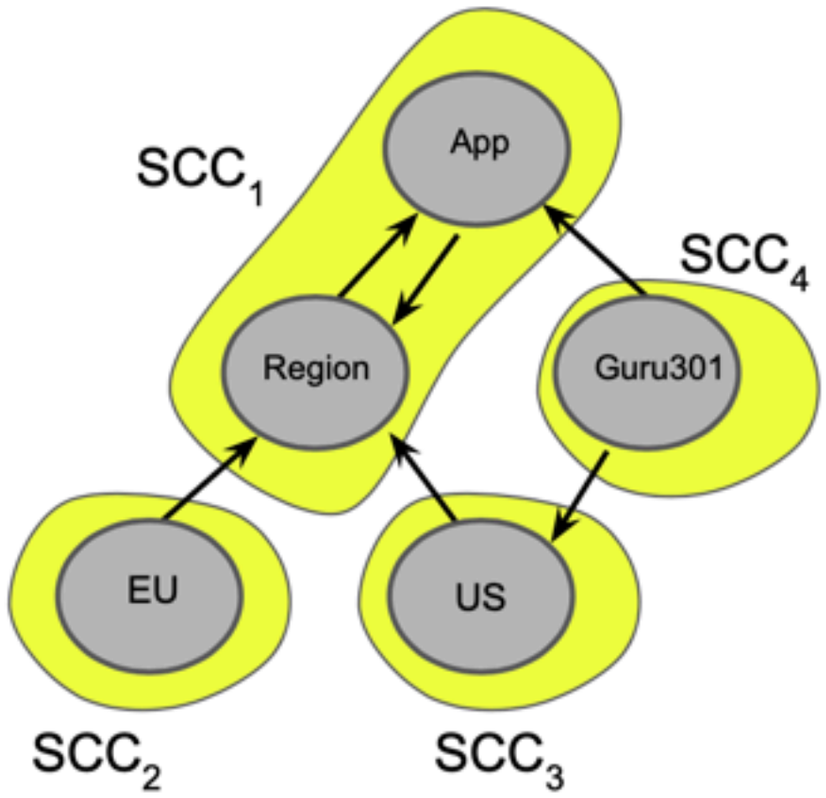
Run-time Component-Based Variability Model



(figures courtesy of Salman Farhat)

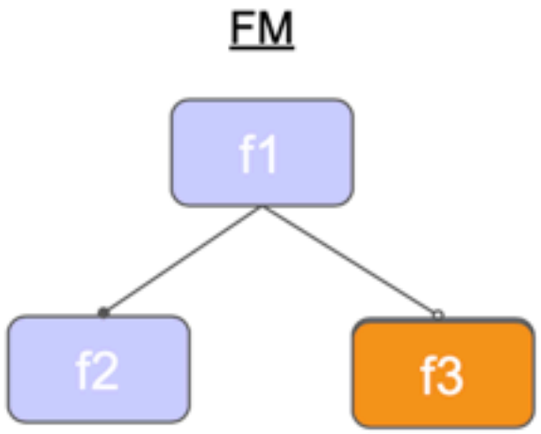
Glue generation

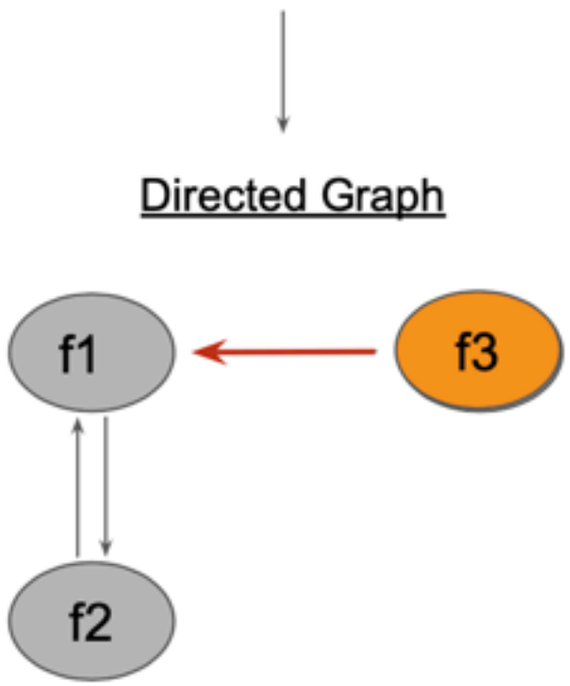


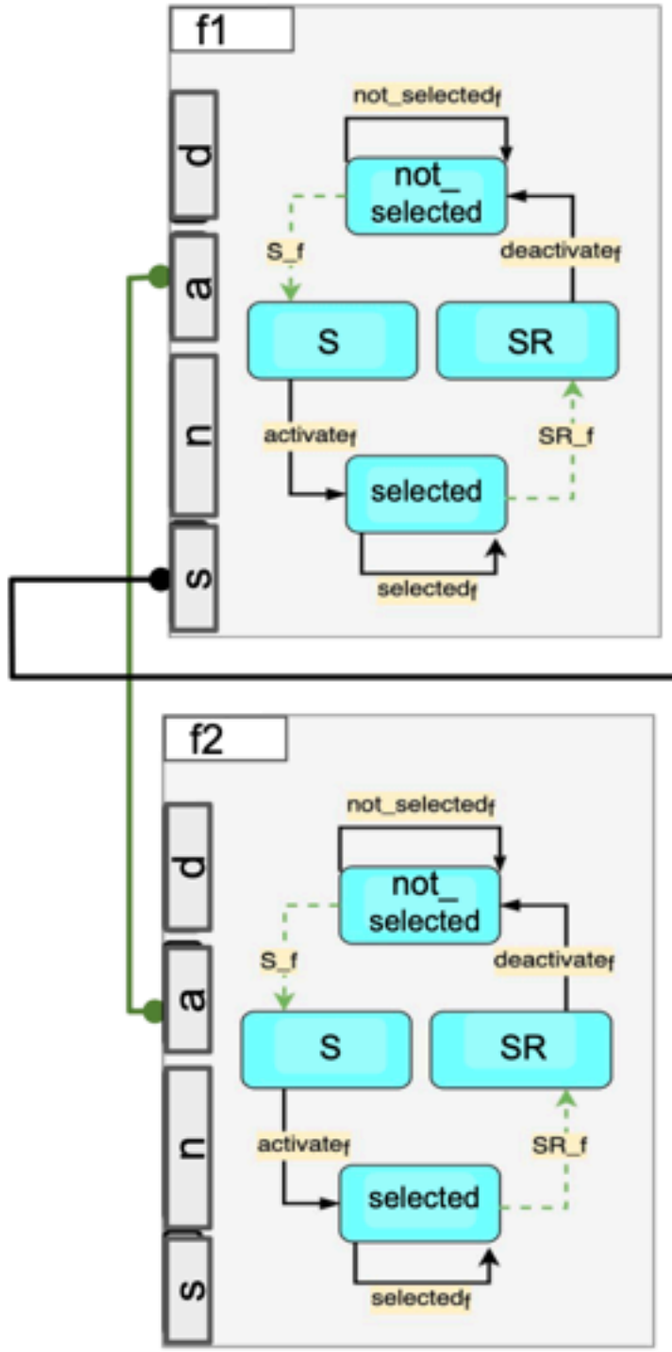


Dependency Graph (G_{FM})

Feature activation

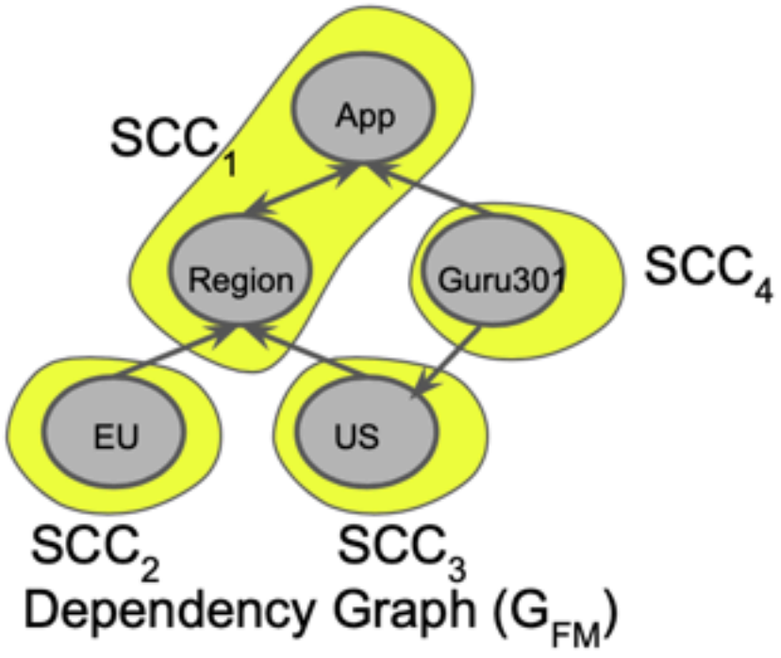






d: deactivate
a: activate
n: not_selected
s: selected

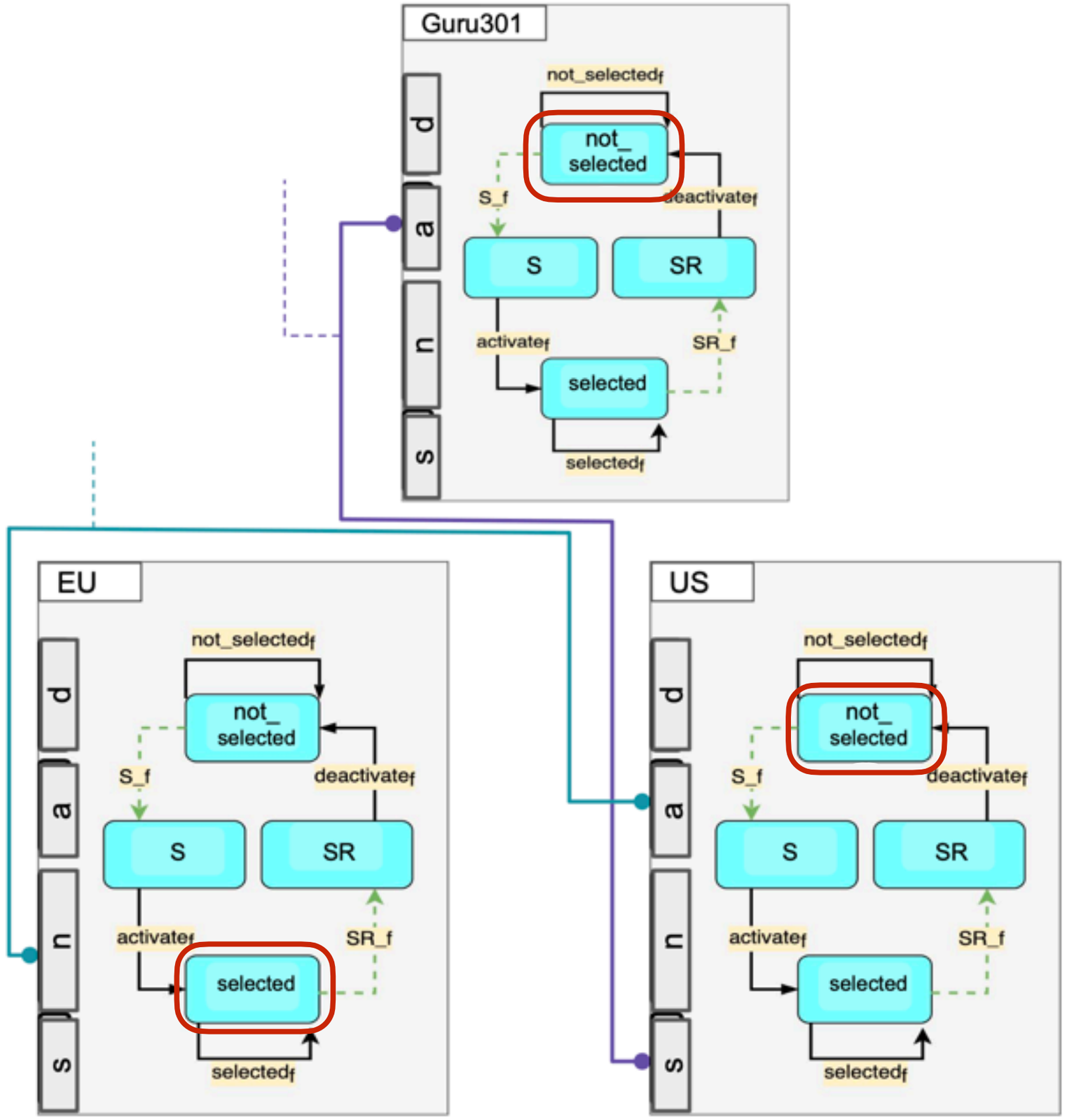
At run time



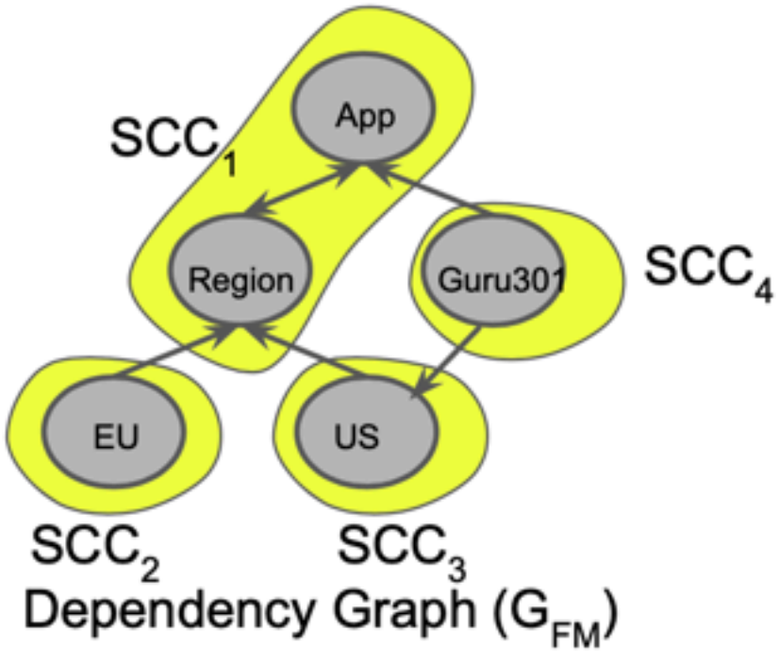
Φ: {App, Region, EU}
Φ': {App, Region, US, Guru301}

Requests:

1. **Activate Guru301**
2. **Activate US**
3. **Deactivate EU**



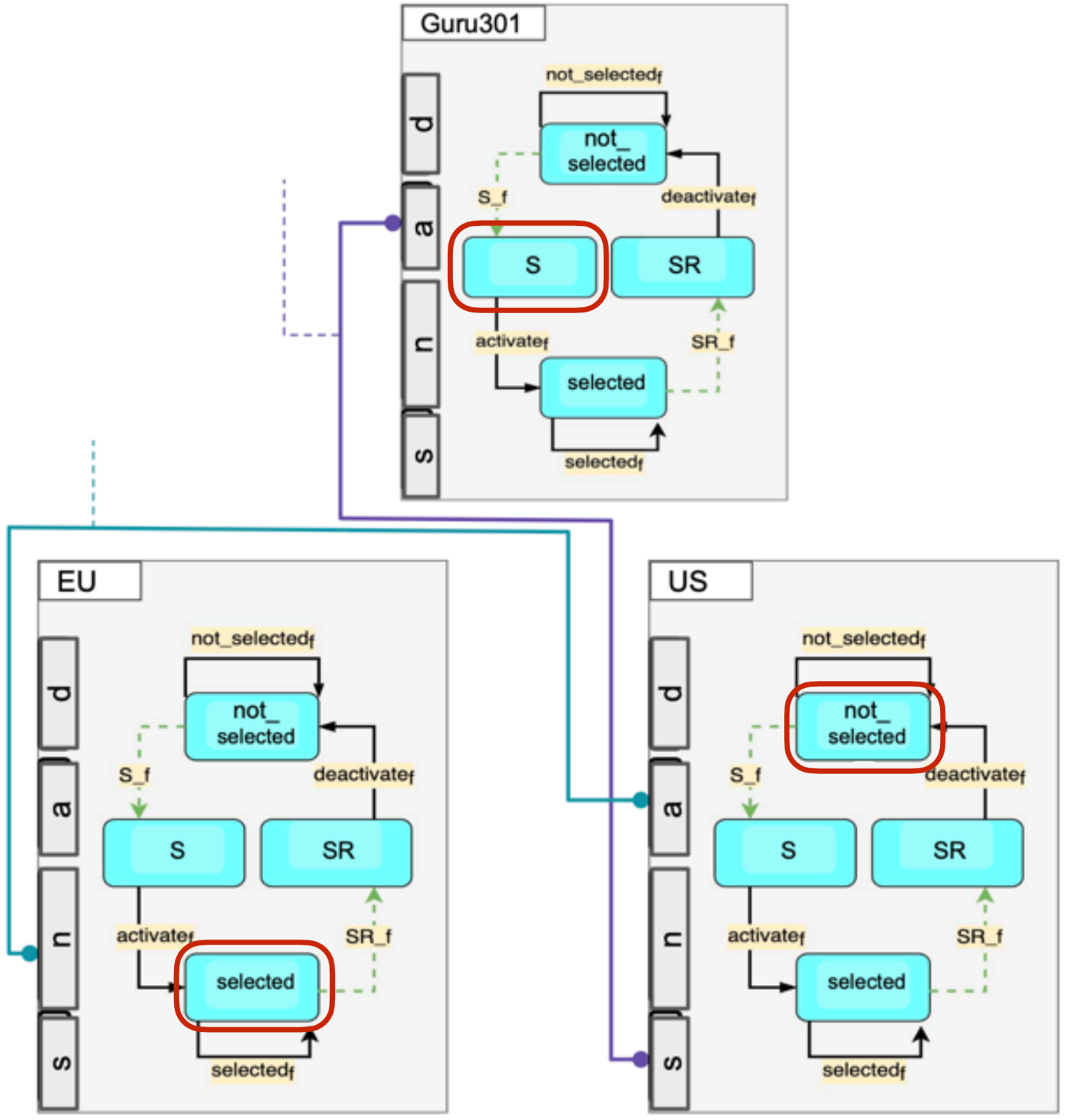
At run time



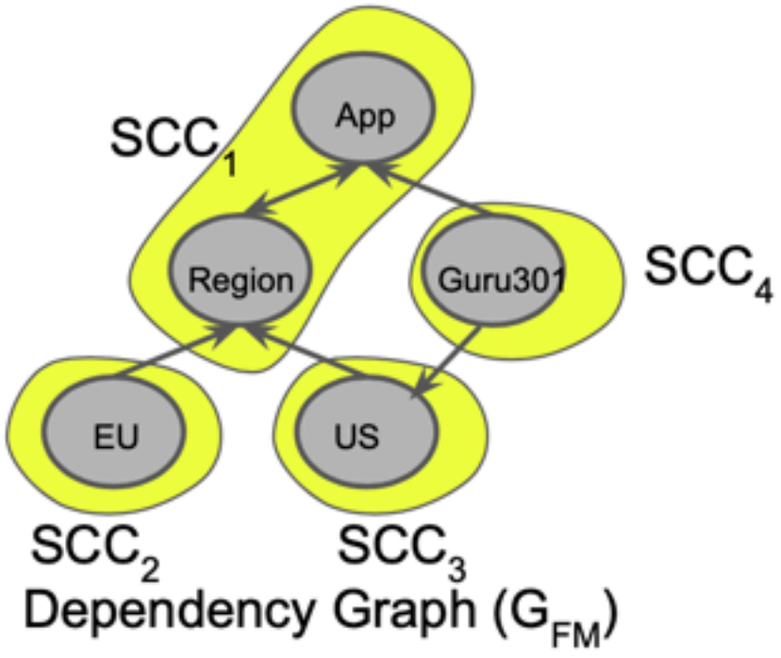
Φ: {App, Region, EU}
Φ': {App, Region, US, Guru301}

Requests:

1. **Activate Guru301**
2. **Activate US**
3. **Deactivate EU**



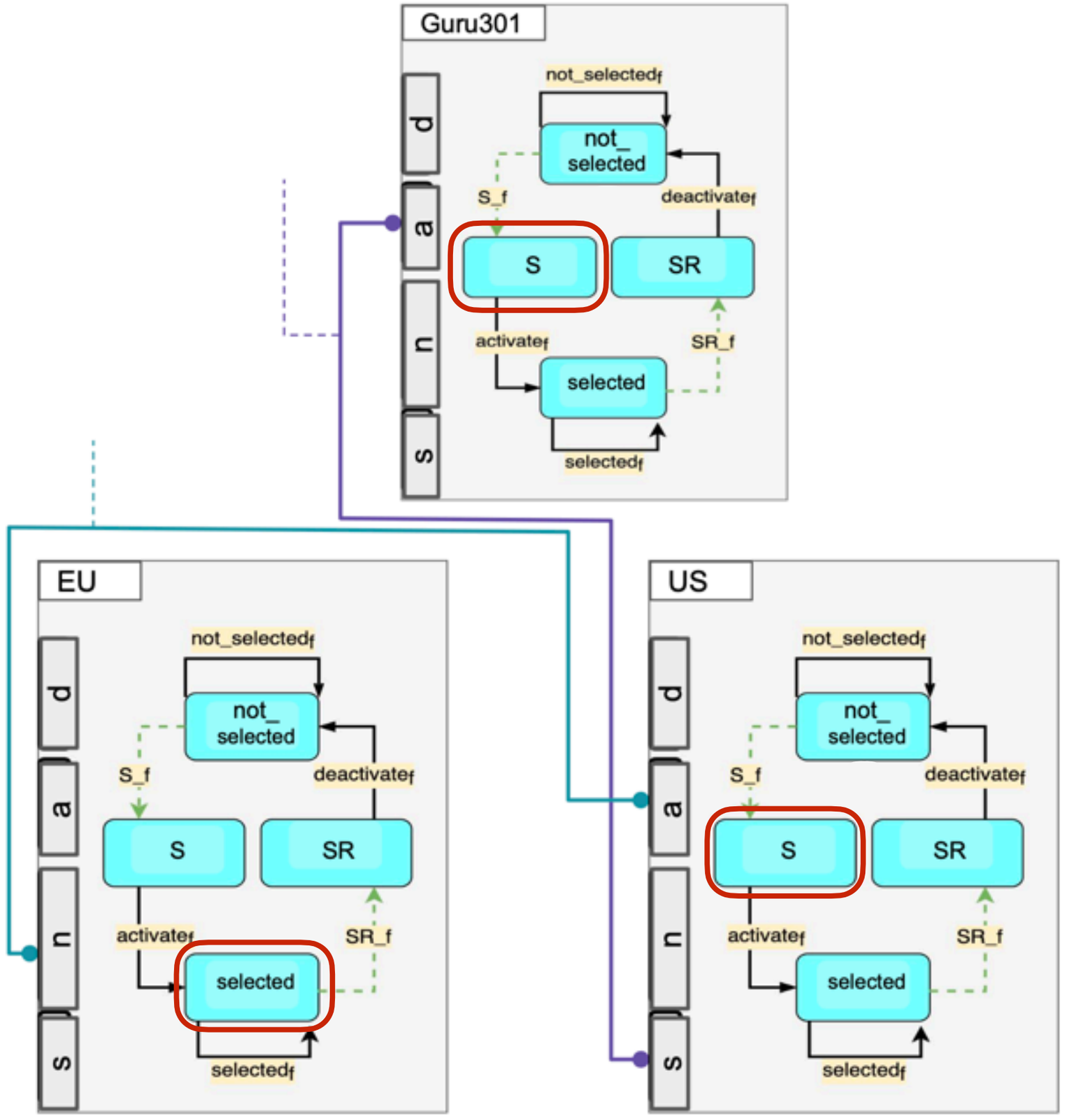
At run time



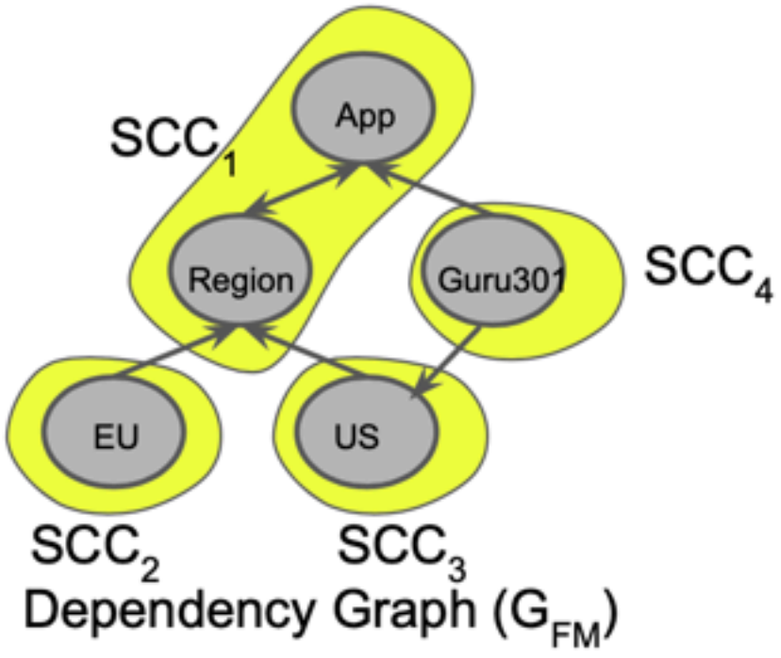
Φ: {App, Region, EU}
Φ': {App, Region, US, Guru301}

Requests:

1. **Activate Guru301**
2. **Activate US**
3. **Deactivate EU**



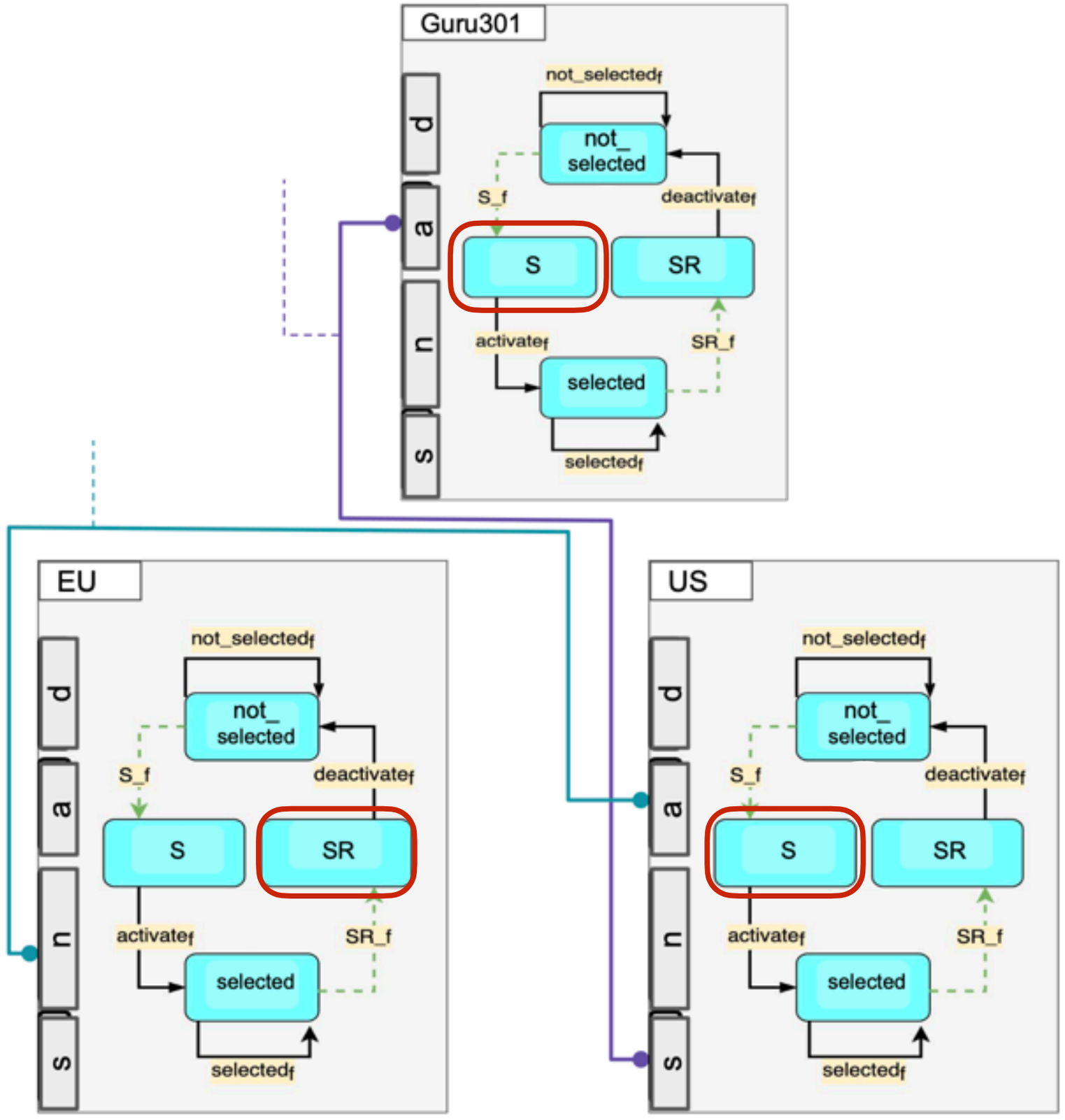
At run time



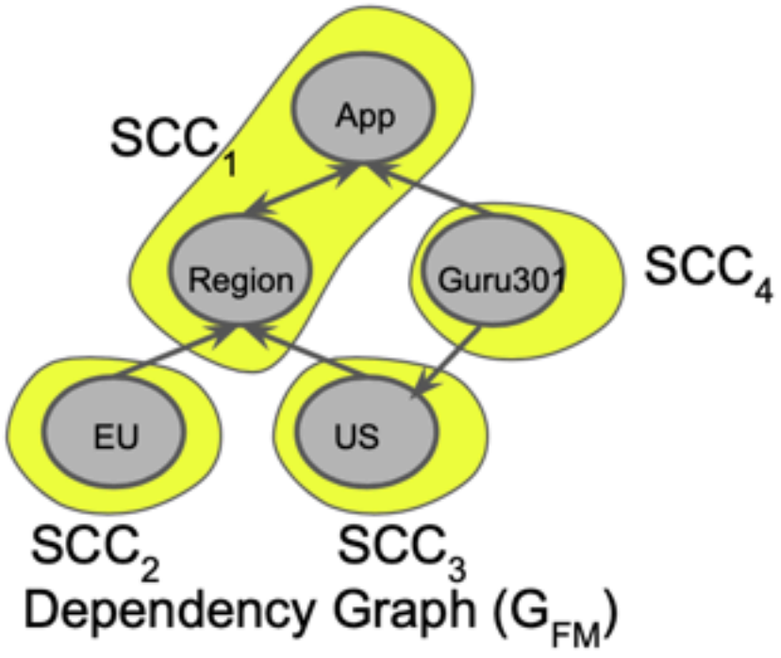
Φ: {App, Region, EU}
Φ': {App, Region, US, Guru301}

Requests:

1. **Activate Guru301**
2. **Activate US**
3. **Deactivate EU**



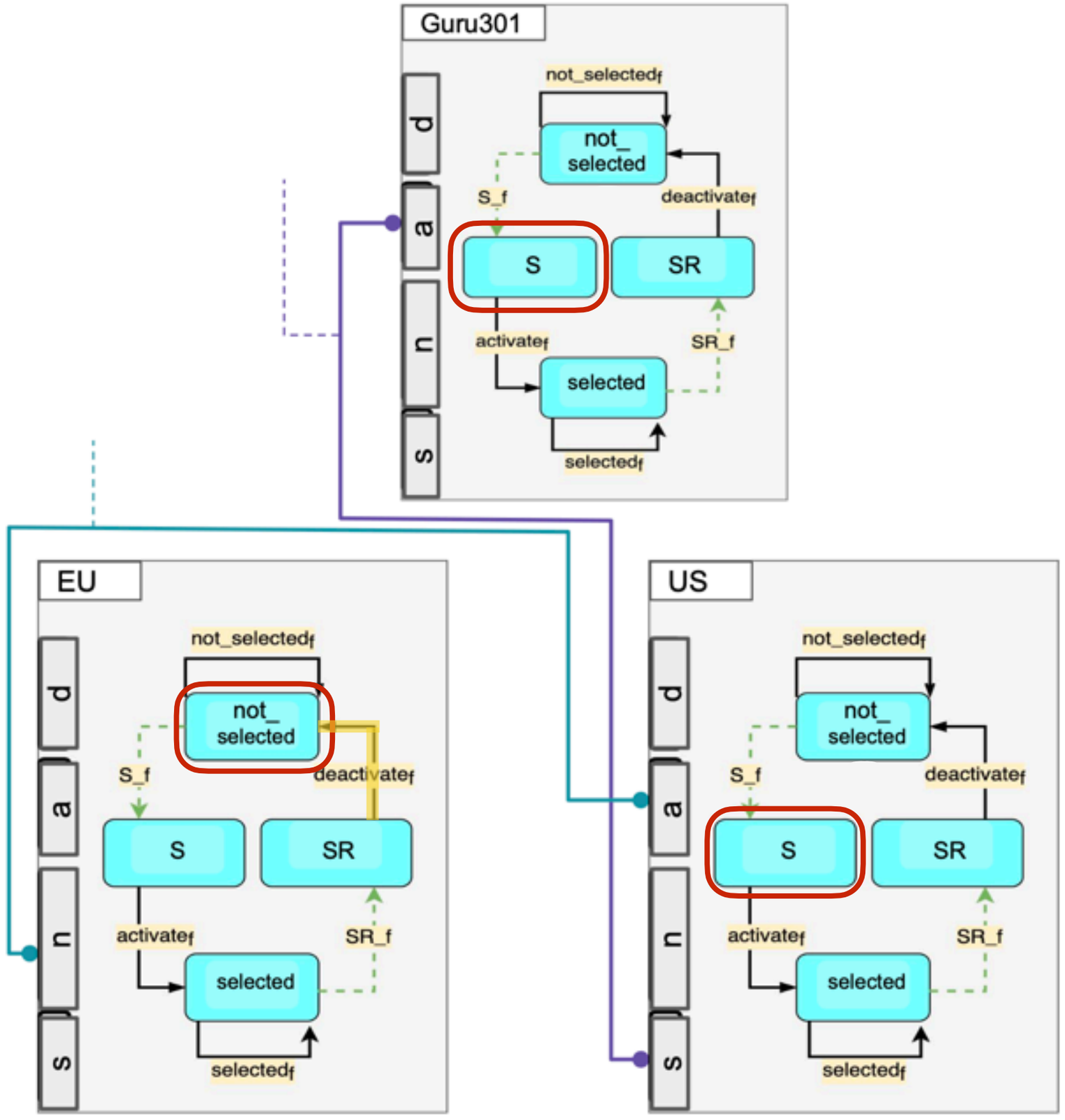
At run time



Φ: {App, Region, EU}
Φ': {App, Region, US, Guru301}

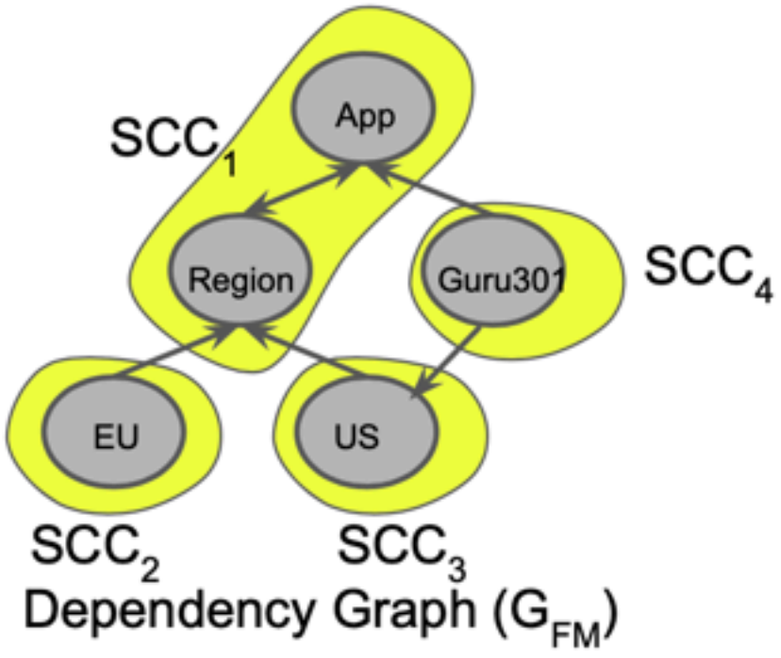
Requests:

1. **Activate Guru301**
2. **Activate US**
3. **Deactivate EU**



(slide courtesy of Salman Farhat)

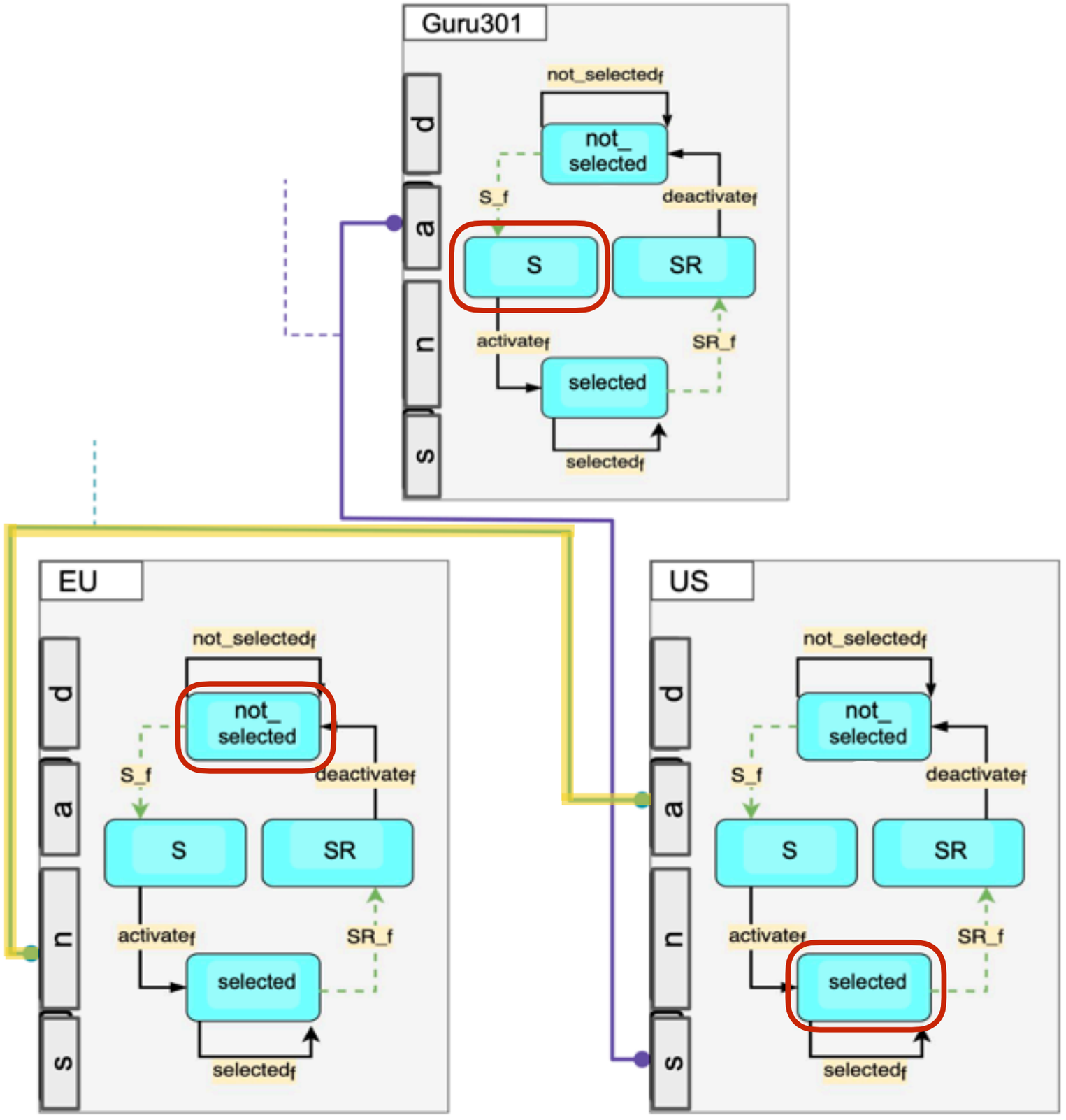
At run time



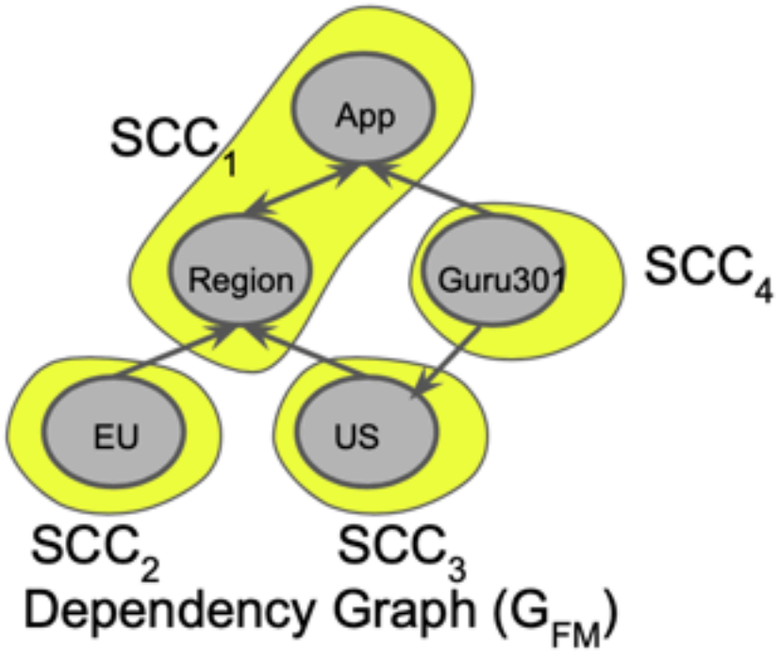
Φ: {App, Region, EU}
Φ': {App, Region, US, Guru301}

Requests:

1. **Activate Guru301**
2. **Activate US**
3. **Deactivate EU**



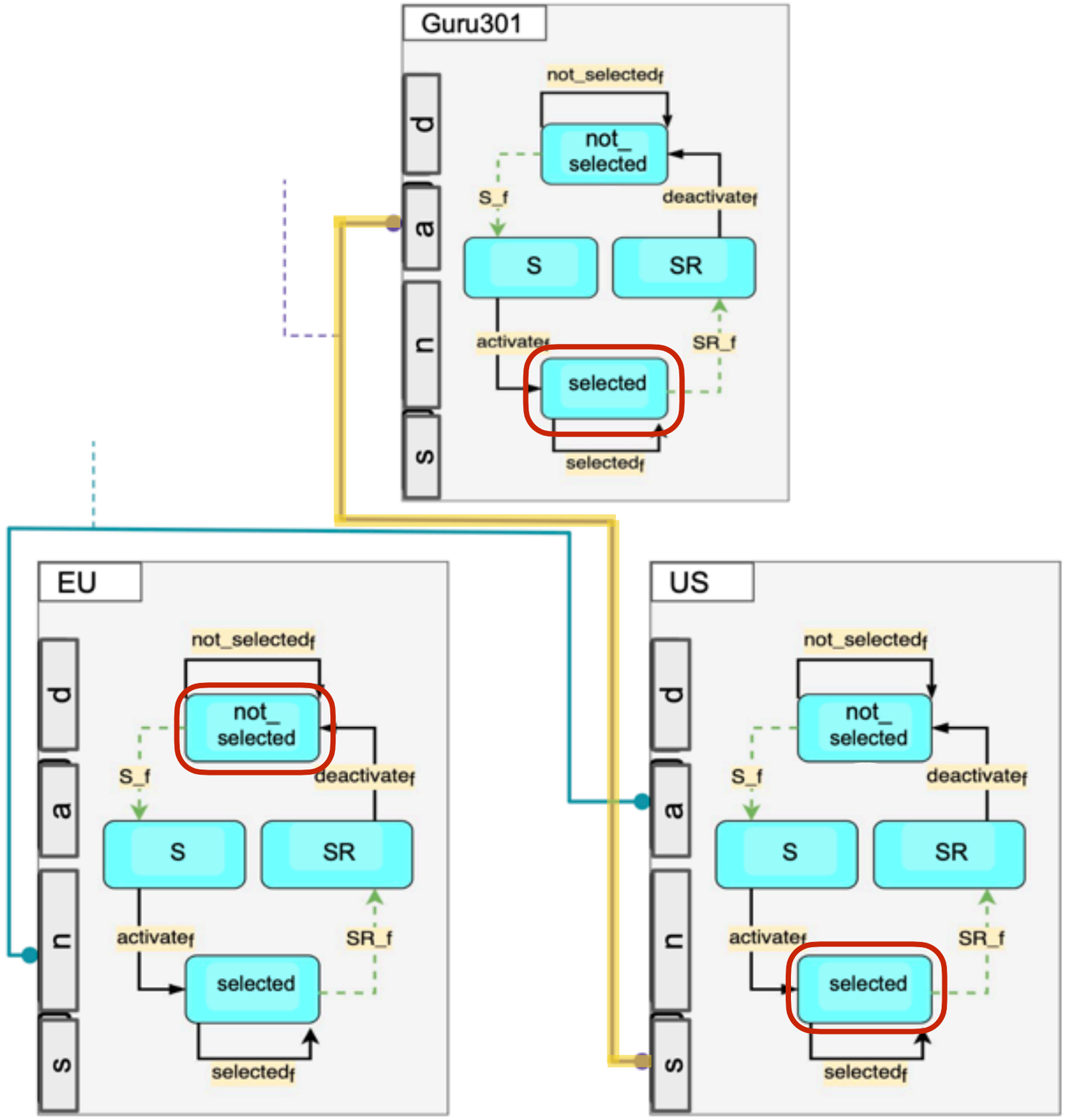
At run time



Φ: {App, Region, EU}
Φ': {App, Region, US, Guru301}

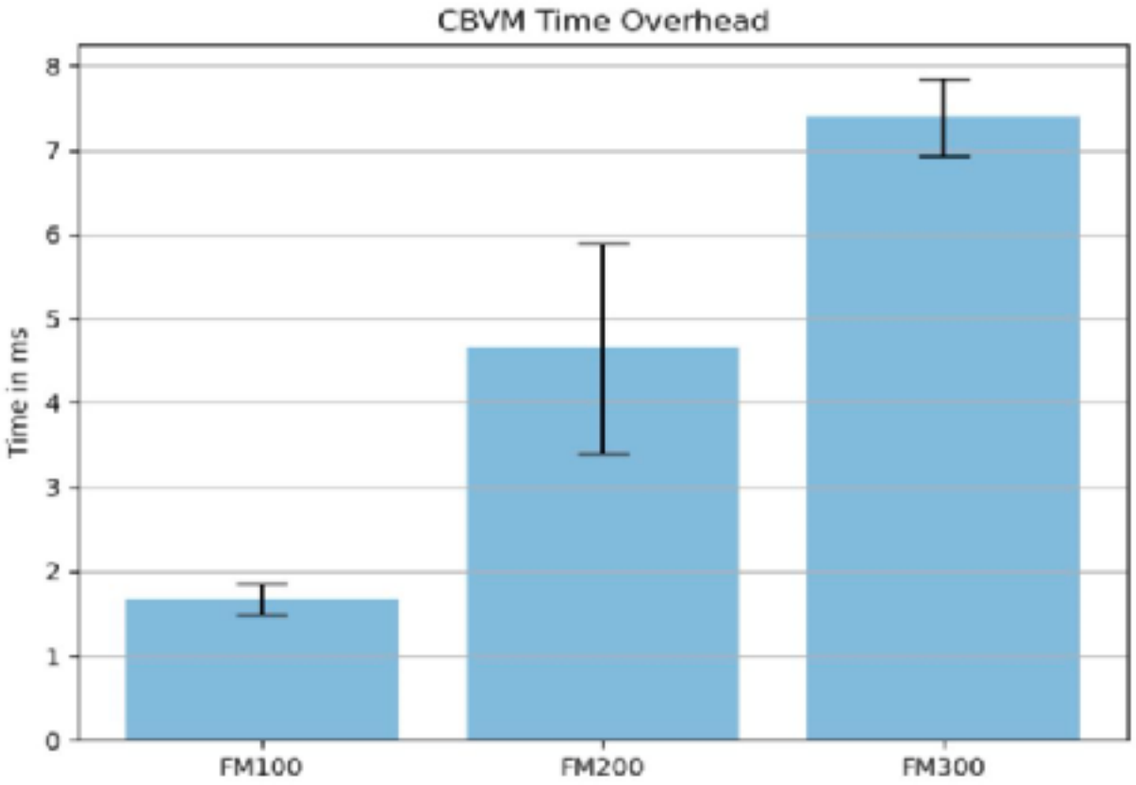
Requests:

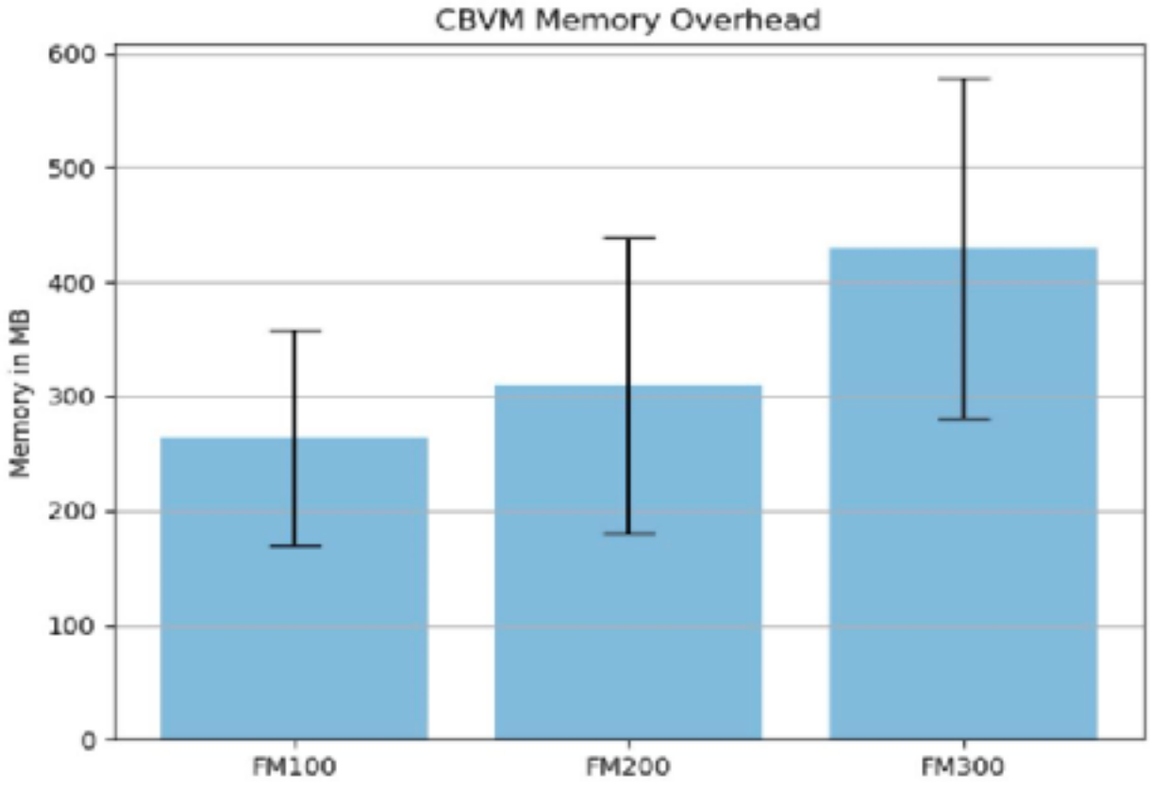
1. **Activate Guru301**
2. **Activate US**
3. **Deactivate EU**



(slide courtesy of Salman Farhat)

Experimental evaluation



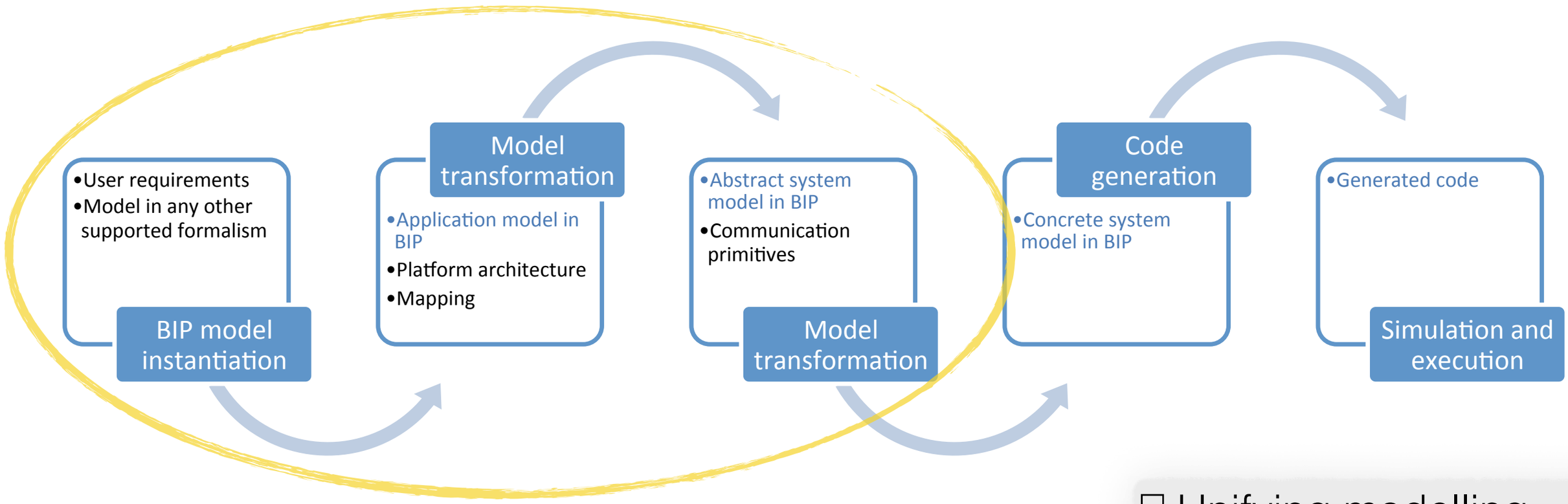


Safely manage the reconfiguration of concurrent component-based applications at runtime

Minimize computational overhead

(figures courtesy of Salman Farhat)

Rigorous System Design flow



A series of semantics-preserving transformations

Correctness decomposed into
- correctness of transformations
- correctness of high-level models

Final implementation is **correct by construction**

- ☐ Unifying modelling framework
- ☐ Operational semantics
- ☐ Method(s) to design correct models

SmartCloud

ANR PRCE project

42 months starting the 15th of January 2024

Inria Lille

Simon Bliudze, Philippe Merle

University of Bologna (Inria Sophia-Antipolis)

Gianluigi Zavattaro, Saverio Giallorenzo, Ivan Lanese

Scalair (Cloud provider and operator)

Damien Vignault et al

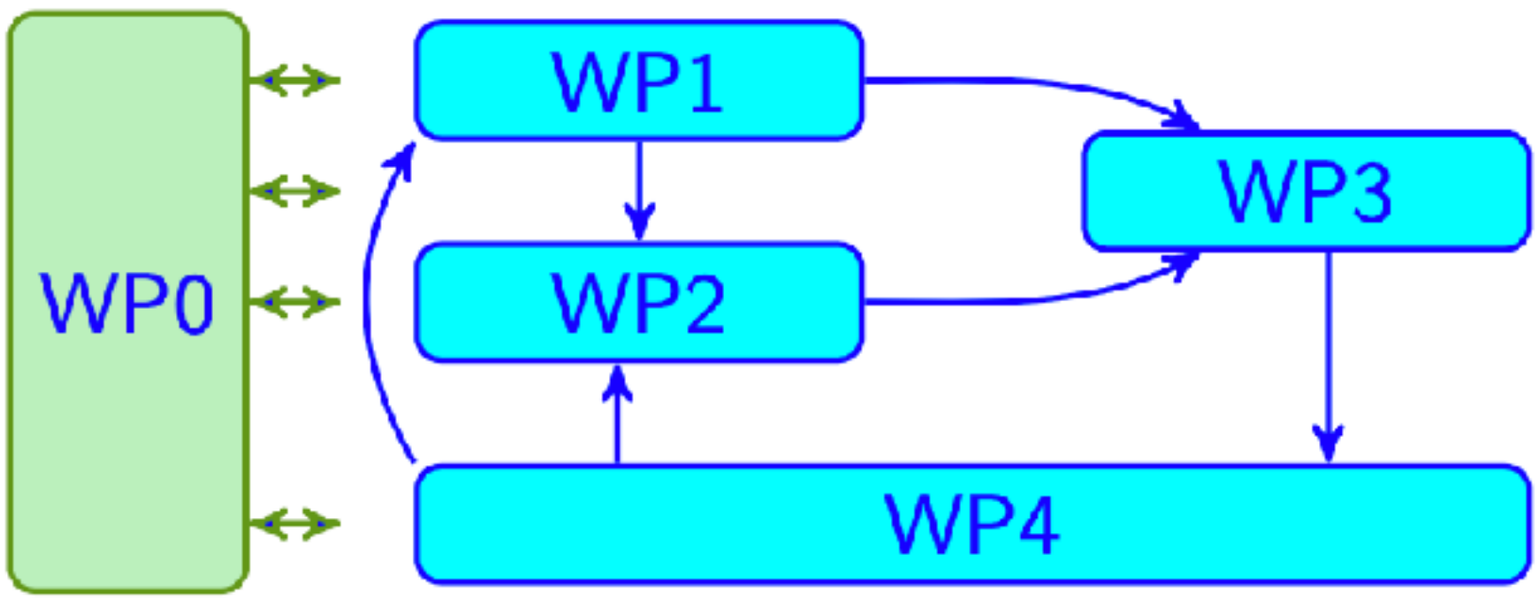
Objectives

Flexible infrastructure for the design of **smart and coordinated dynamic adaptation** frameworks for CC systems

- New formalism for joint modelling of platform and application aspects, providing mechanisms for dynamic adaptation
  - monitoring
  - coordination
  - control

New algorithms and heuristics for **distributed on-line optimisation**

SmartCloud



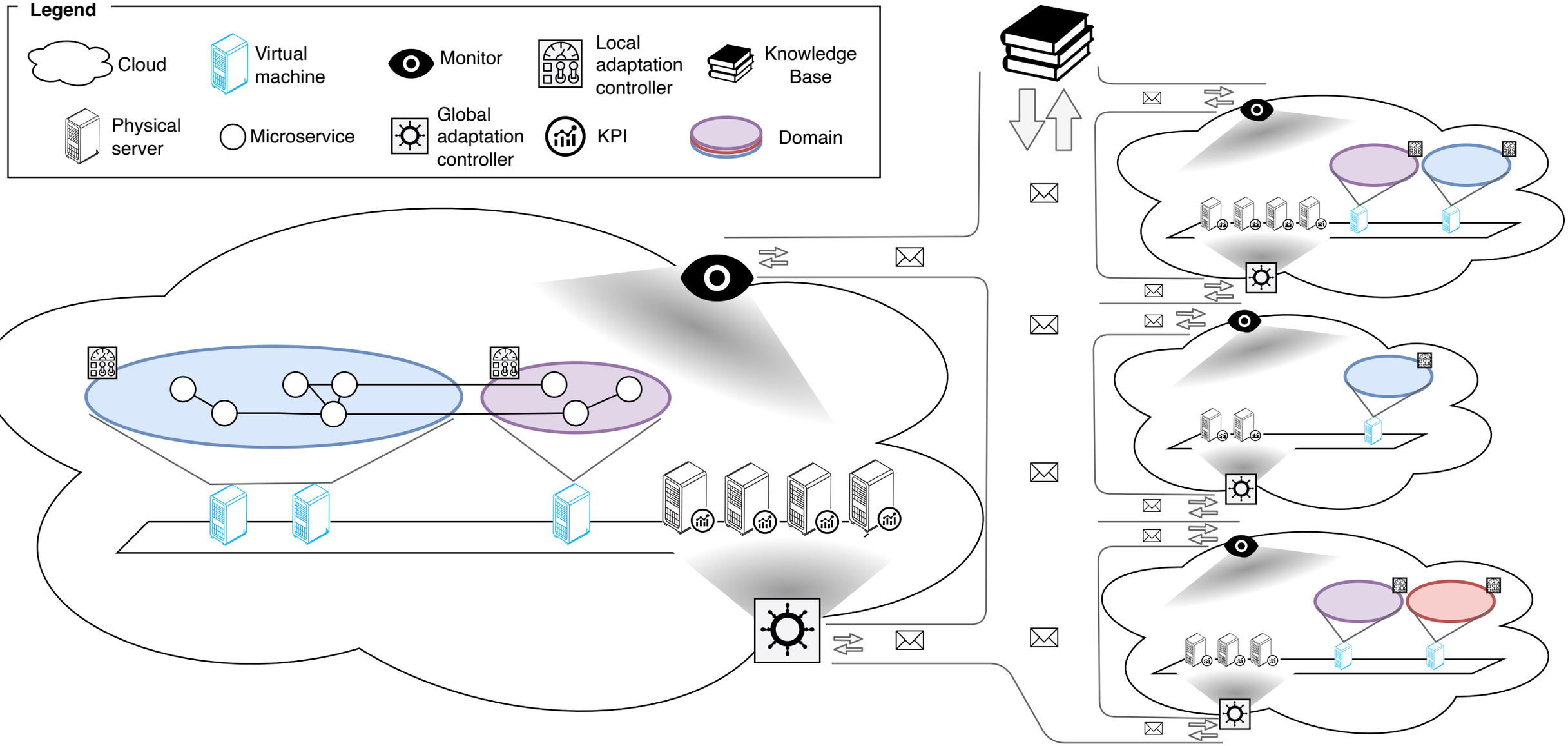
WP1: Design framework

WP2: Dynamic adaptation planning

WP3: Prototyping and integration

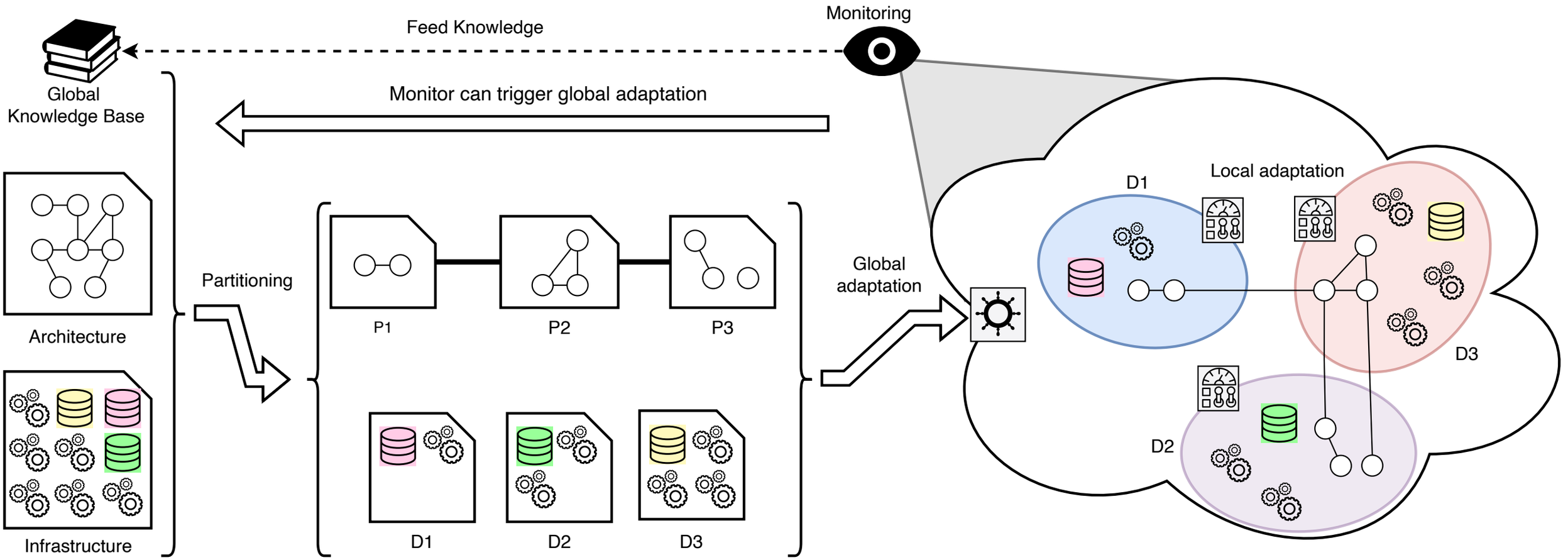
WP4: Evaluation

WP1: Design framework



Two models allowing for the representation of both the **inter-dependencies/interactions among CC system components** and the **computing resources** required by each component

WP2: Dynamic adaptation planning



Hybrid techniques for the automatic deployment of CC systems, combining optimisation with heuristics

- compute at the global level an appropriate partitioning of software entities and resources
- allocate each sub-set of resources to a local controller

We will be hiring

A post-doc in Lille

WP 1: Design framework

12 months

1-2 post-doc(s) in Bologna

WP 1,2: Dynamic adaptation planning

24 months

An engineer in Lille

WP 3: Prototyping and integration

18 months

Inria

https://www.puratura.com/archives/portfolio/items/i-want-you-for-science

Appendices

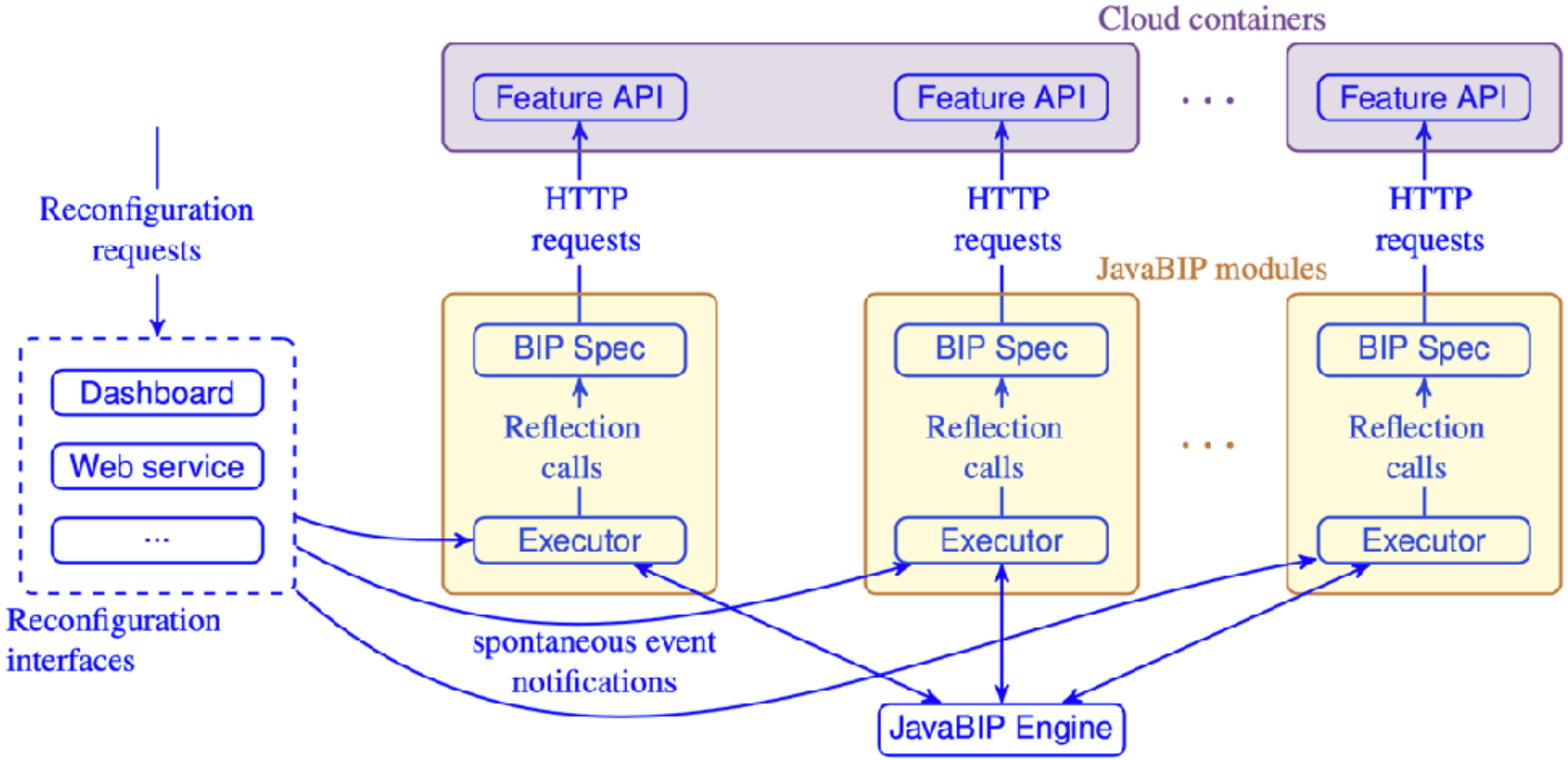
JavaBIP components

```
 1 @Ports({
 2  @Port(name = "add", type = PortType.enforceable),
 3   @Port(name = "rm", type = PortType.enforceable)
 4 })
 6 @ComponentType(initial = "on", name = "MemoryMonitor")
 7 public class MemoryMonitor {
 8
 9   final private int memoryLimit;
10   private int currentCapacity = 0;
11
12   public MemoryMonitor(int memoryLimit) {
13       this.memoryLimit = memoryLimit;
14       }
15
16   @Transition(name = "add", source = "on", target = "on",
17               guard = "hasCapacity")
18   public void addRoute(@Data("memoryUsage") Integer deltaMemory) {
19     currentCapacity += deltaMemory;
20   }
21
22   @Transition(name = "rm", source = "on", target = "on", guard = "")
23   public void removeRoute(@Data(name="memoryUsage") Integer deltaMemory) {
24     currentCapacity -= deltaMemory;
25   }
26
27   @Guard(name = "hasCapacity")
28   public boolean hasCapacity(@Data("memoryUsage") Integer memoryUsage) {
29     return currentCapacity + memoryUsage < memoryLimit;
30   }
31 }
```

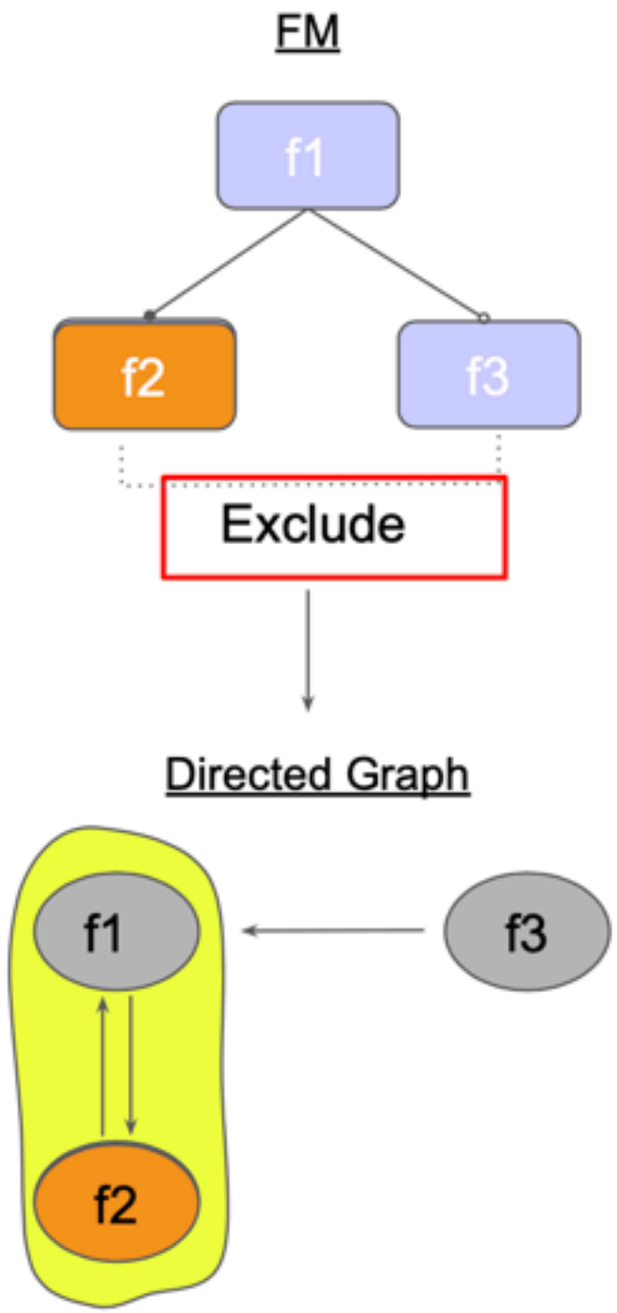
JavaBIP glue

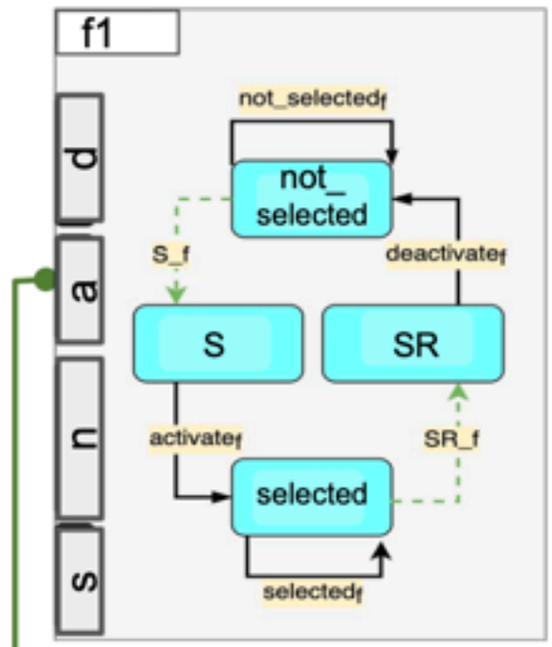
```
1 public static void main(String[] args) throws Exception {
2   ...
3   BIPGlue glue = new TwoSynchronGlueBuilder() {
4     @Override
5     public void configure() {
6       /** Synchronisation specifications **/
7       port(Operator.class, DECIDE_BET).requires(Casino.class, CASINO_WIN);
8       port(Casino.class, CASINO_WIN).requires(Operator.class, DECIDE_BET);
9
10      port(Operator.class, DECIDE_BET).requires(Casino.class, PLAYER_WIN);
11      port(Casino.class, PLAYER_WIN).requires(Operator.class, DECIDE_BET);
12      port(Player.class, RECEIVE_MONEY).requires(Casino.class, PLAYER_WIN);
13
14      port(Casino.class, CASINO_WIN).accepts(Operator.class, DECIDE_BET);
15      port(Casino.class, PLAYER_WIN)
16         .accepts(Operator.class, DECIDE_BET, Player.class, RECEIVE_MONEY);
17      port(Operator.class, DECIDE_BET)
18         .accepts(Casino.class, CASINO_WIN, Casino.class, PLAYER_WIN,
19                  Player.class, RECEIVE_MONEY);
20      port(Player.class, RECEIVE_MONEY)
21         .accepts(Casino.class, PLAYER_WIN, Operator.class, DECIDE_BET);
22
23      /** Data wire specifications **/
24      data(Operator.class, OUTGOING_FUNDS).to(Casino.class, INCOMING_FUNDS);
25      data(Operator.class, ID).to(Casino.class, OPERATOR);
26    }
27  }.build();
28
29  BIPEngine engine = engineFactory.create(ENGINE, glue);
30  ...
31 }
```

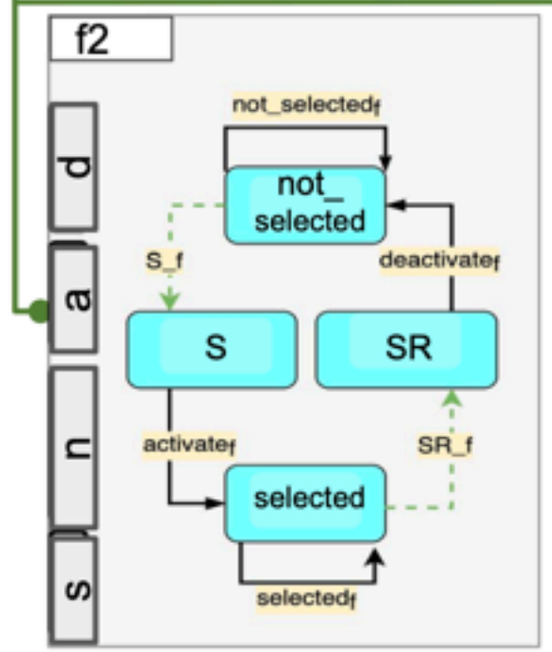
JavaBIP–Cloud integration



Exclude constraints







d: deactivate
a: activate
n: not_selected
s: selected