

HiPEAC 2020

20-22 January, 2020, Bologna, Italy



CERBERO H2020 Project – Tutorial: Design Flow for Heterogeneous Embedded Computing Infrastructures

Tiziana Fanni², Daniel Madroñal¹, Maxime Pelcat³, Alfonso Rodriguez¹, Carlo Sau⁴, Claudio Rubattu^{2,3},
Leonardo Suriano¹

¹Universidad Politécnica de Madrid, ²Università degli Studi di Sassari, ³IETR INSA-Rennes, ⁴Università degli
Studi di Cagliari



Horizon 2020
European Union funding
for Research & Innovation

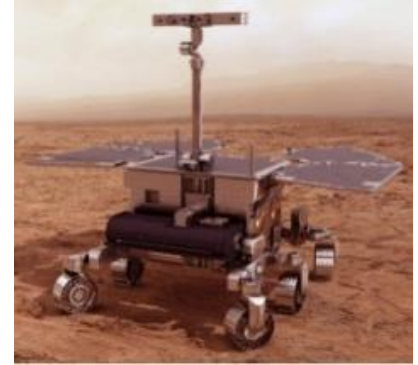
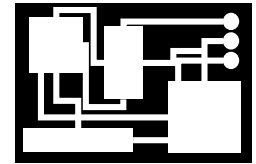
Adaptivity and Heterogeneity Requirements

- Here is a **Martian robot**



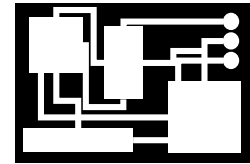
Adaptivity and Heterogeneity Requirements

- Here is a **Martian robot**
 - it has **wheels**, a **robotic arm**, many **sensors**, and a **complex control and computing system**



Adaptivity and Heterogeneity Requirements

- Here is a **Martian robot**
 - it has **wheels**, a **robotic arm**, many **sensors**, and a **complex control and computing system**
- Here is **what can happen** to this robot



Adaptivity and Heterogeneity Requirements

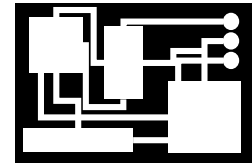
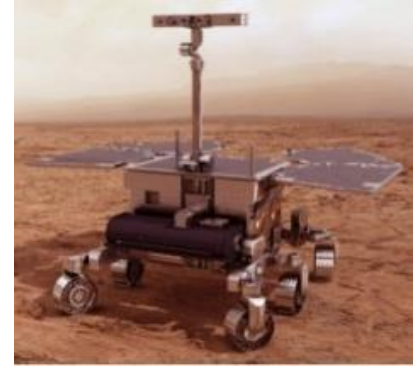
- Here is a **Martian robot**

- it has **wheels**, a **robotic arm**, many **sensors**, and a **complex control and computing system**



- Here is **what can happen** to this robot

- one of its **cameras** can be occluded



Adaptivity and Heterogeneity Requirements

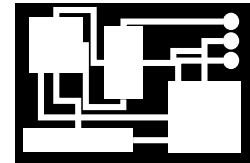
- Here is a **Martian robot**

- it has **wheels**, a **robotic arm**, many **sensors**, and a **complex control and computing system**



- Here is **what can happen** to this robot

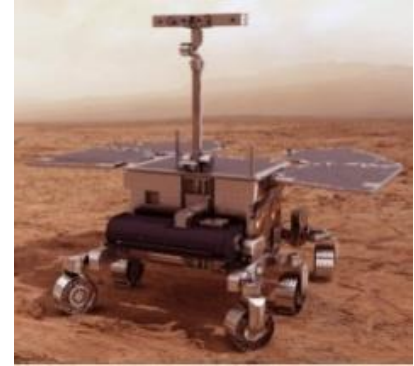
- one of its **cameras** can be occluded
- one of its **wheels** can be blocked



Adaptivity and Heterogeneity Requirements

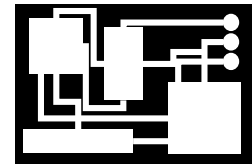
- Here is a **Martian robot**

- it has **wheels**, a **robotic arm**, many **sensors**, and a **complex control and computing system**



- Here is **what can happen** to this robot

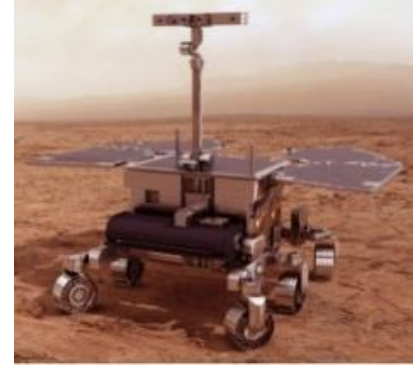
- one of its **cameras** can be occluded
- one of its **wheels** can be blocked
- one **arm** segment can be stuck



Adaptivity and Heterogeneity Requirements

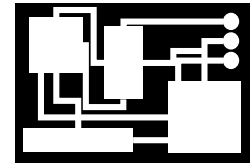
- Here is a **Martian robot**

- it has **wheels**, a **robotic arm**, many **sensors**, and a **complex control and computing system**



- Here is **what can happen** to this robot

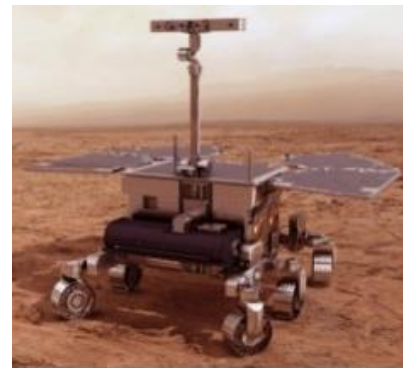
- one of its **cameras** can be occluded
- one of its **wheels** can be blocked
- one **arm** segment can be stuck
- one **motor** can overheat



Adaptivity and Heterogeneity Requirements

- Here is a **Martian robot**

- it has **wheels**, a **robotic arm**, many **sensors**, and a **complex control and computing system**



- Here is **what can happen** to this robot



- one of its **cameras** can be occluded
- one of its **wheels** can be blocked
- one **arm** segment can be stuck
- one **motor** can overheat
- on **processing** itself, a region of hardware can be temporarily or permanently unavailable
 - in particular because of ionizing particles



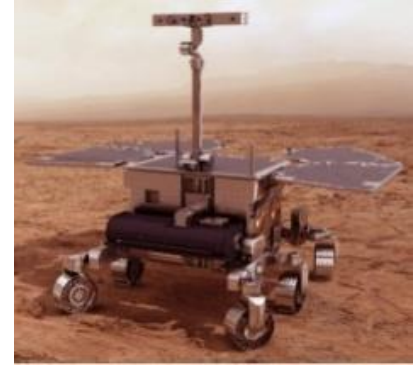
Adaptivity and Heterogeneity Requirements

- How do we **resume the mission** in case of **problems**?
 - or at least use **remaining robot capacities** for future operations?



Adaptivity and Heterogeneity Requirements

- How do we **resume the mission** in case of **problems**?
 - or at least use **remaining robot capacities** for future operations?
- By the way, **radio-controlling** is **not feasible**
 - **light** return trip to earth is **9 minutes** so...



Adaptivity and Heterogeneity Requirements

- How do we **resume the mission** in case of **problems**?
 - or at least use **remaining robot capacities** for future operations?
- By the way, **radio-controlling** is **not feasible**
 - **light** return trip to earth is **9 minutes** so...
- Proposition
 - **Self-adaptation** support and **CERBERO** tooling
 - **HW/SW reconfiguration** for reliability and performance



Ultimate Goal: Self-Adaptation Support



Self-adaptation: *runtime* action *changing structure, functionality and/or parameters of a system*, according to environment, user or self-sensing info.

[F.D. Macías-Escrivá, et al. “*Self-adaptive systems: A survey of current approaches, research challenges and applications*” In Expert Systems with Applications, 2013]

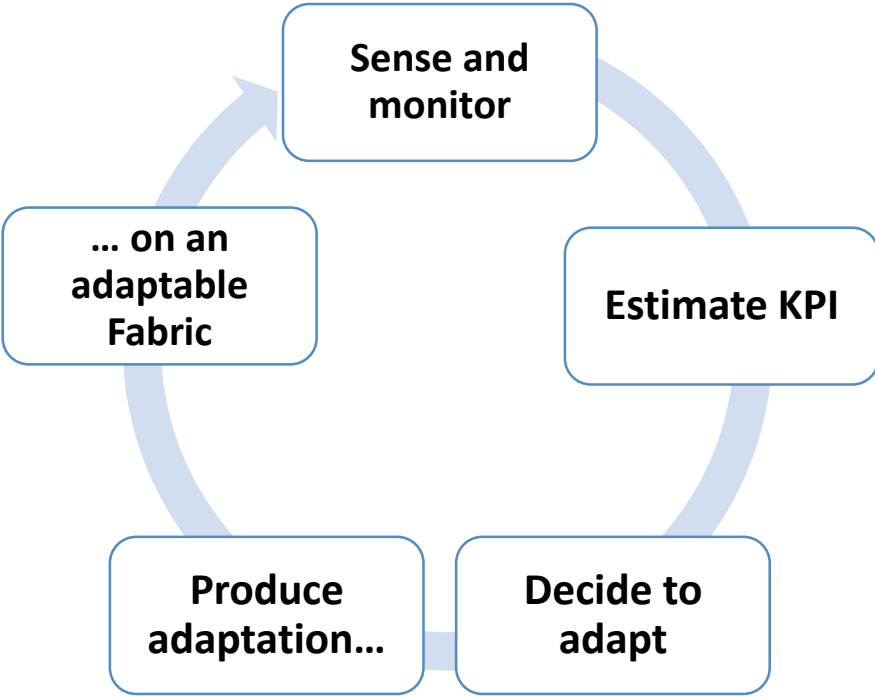


System self-adaptation: *awareness + reconfiguration.*

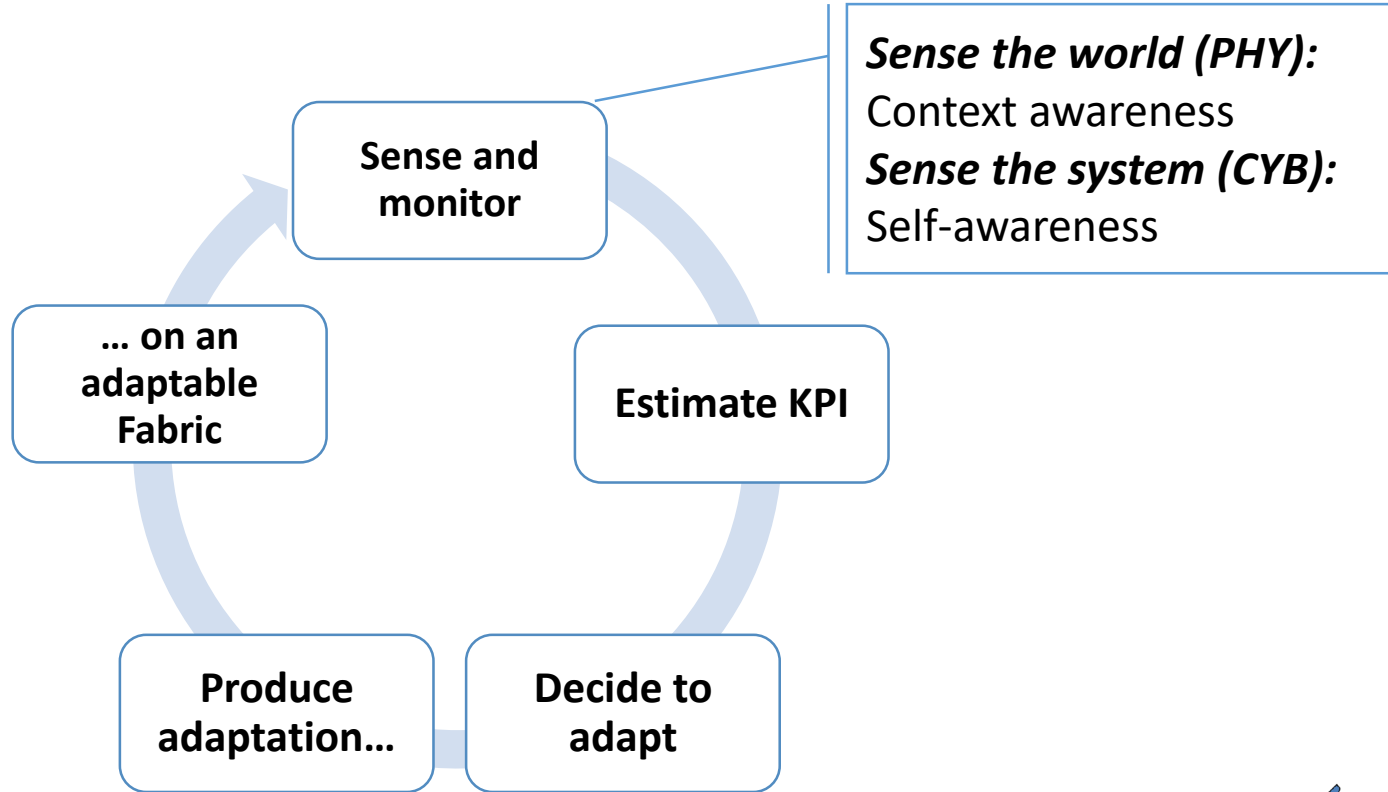
- Reconfiguration → decided *inside the CPS* by a smart *self-adaptation manager (hierarchy of managers)*.
- Decisions → based on run-time **Key Performance Indicators (KPIs)** measurements and estimations.



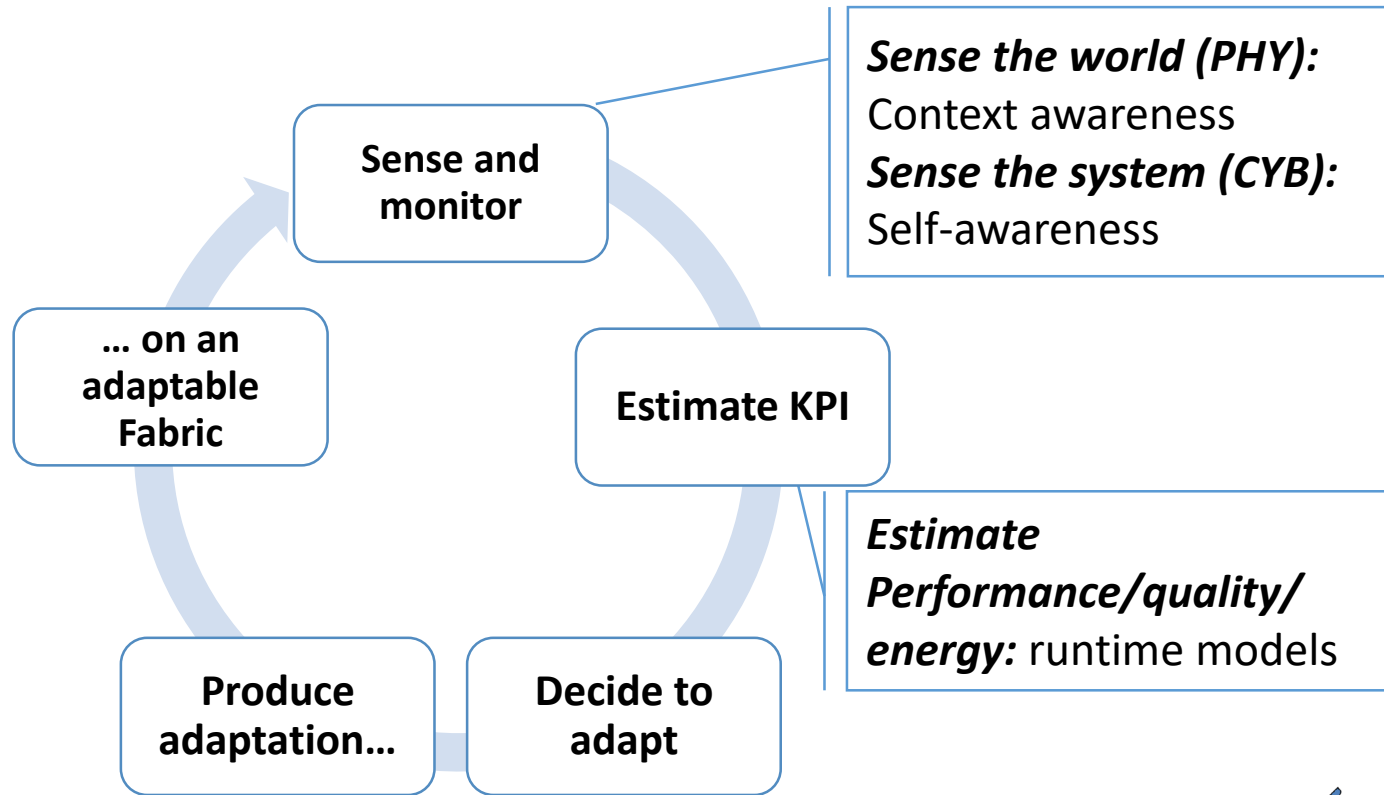
CERBERO Adaptation Loop Formalization



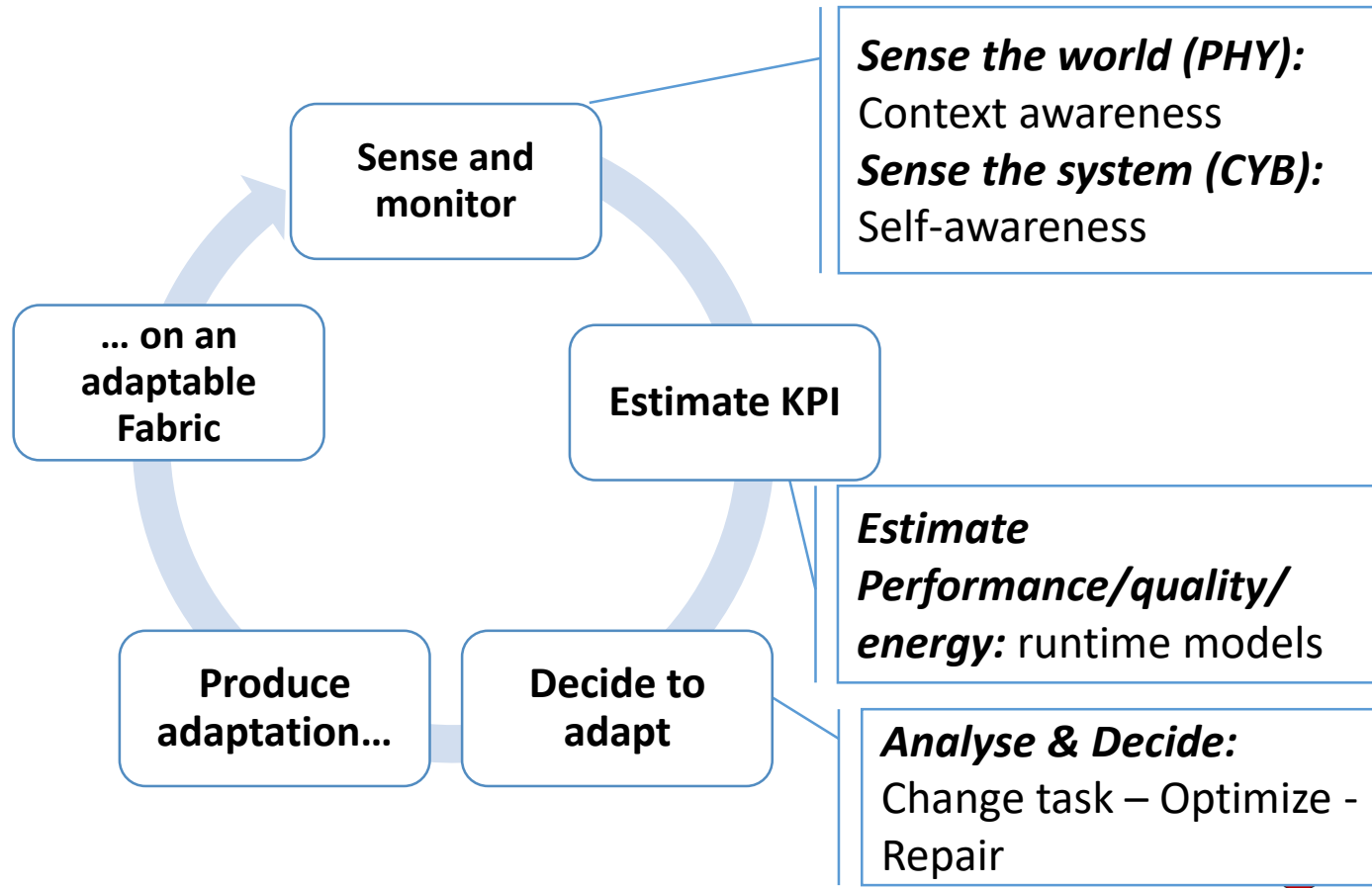
CERBERO Adaptation Loop Formalization



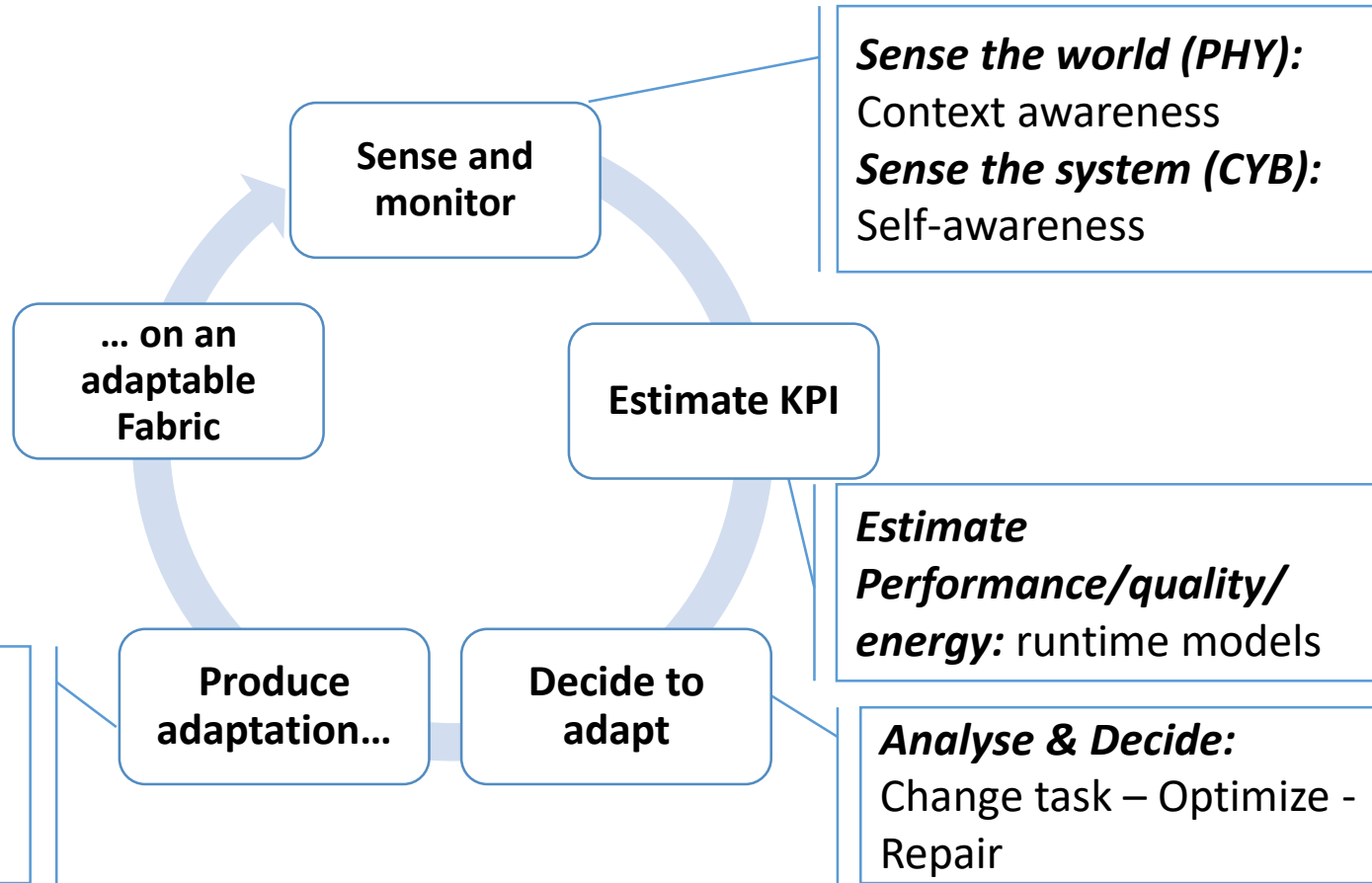
CERBERO Adaptation Loop Formalization



CERBERO Adaptation Loop Formalization



CERBERO Adaptation Loop Formalization



CERBERO Adaptation Loop Formalization

Adapt:

Reconfigure the heterogenous and multi-level computing infrastructure. Multiple fabrics.

Sense the world (PHY):

Context awareness

Sense the system (CYB):

Self-awareness

Sense and monitor

Estimate KPI

Estimate

Performance/quality/energy: runtime models

Analyse & Decide:

Change task – Optimize - Repair

Produce adaptation...

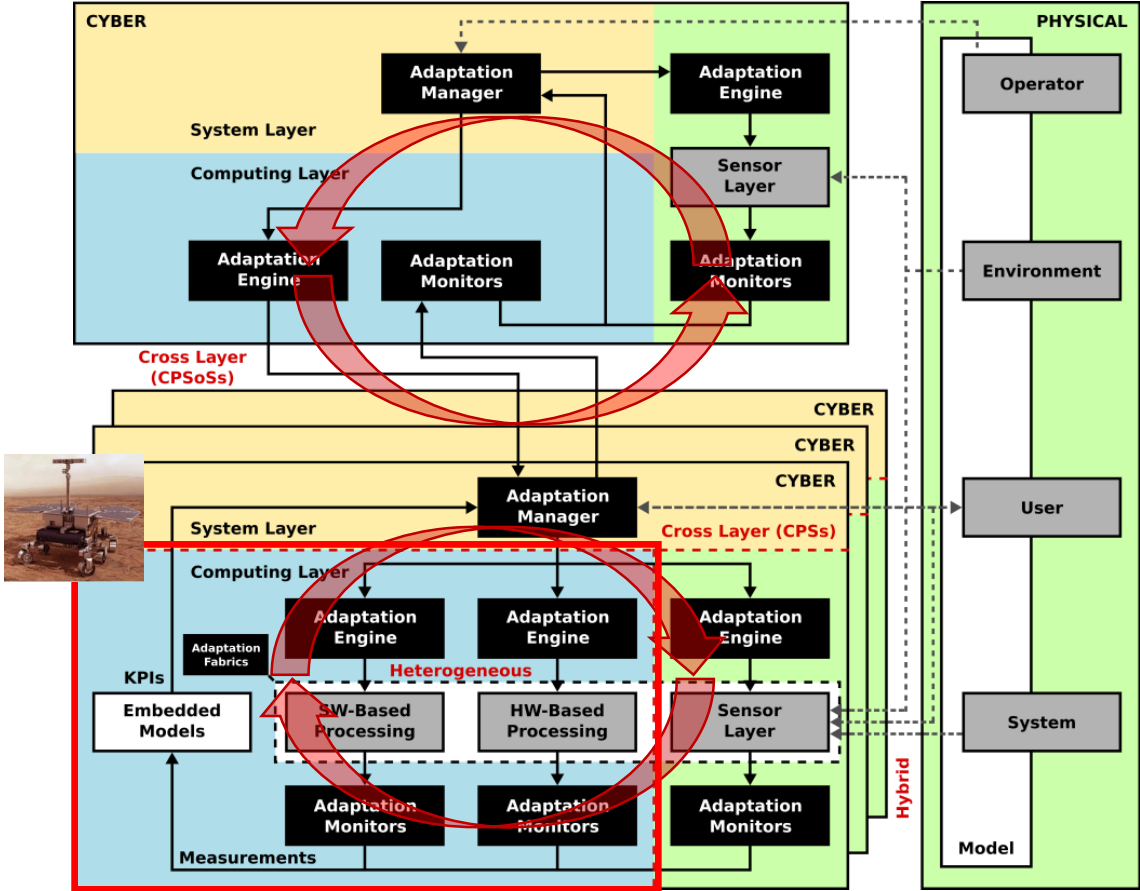
Decide to adapt

... on an adaptable Fabric

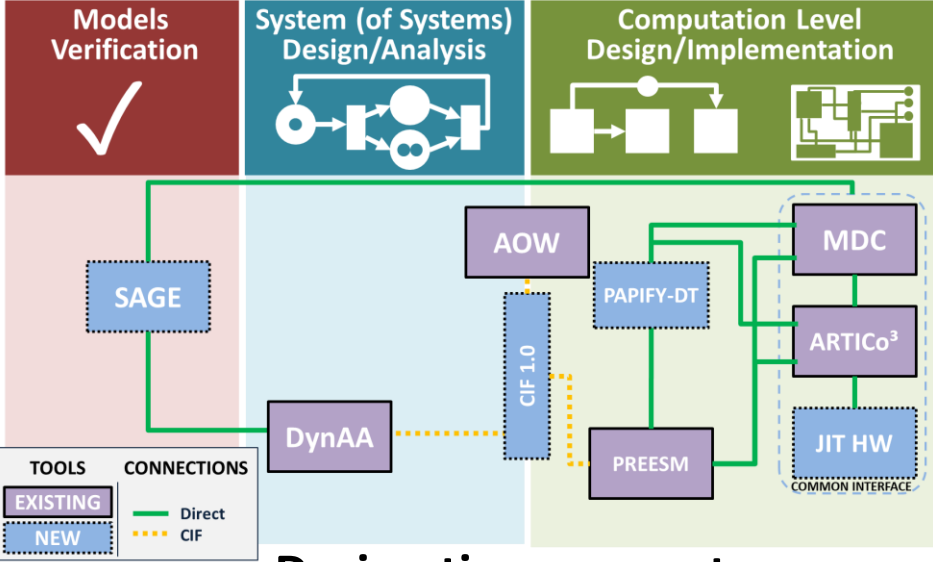
Command Adaptation:

Put in place the actions for the required adaptation

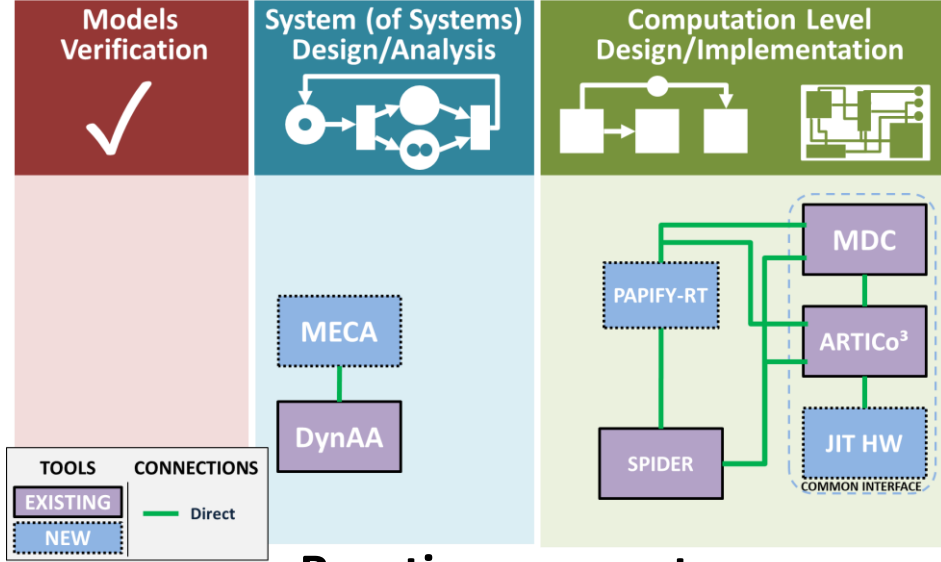
CERBERO Self-Adaptive Multi-Level Infrastructure



CERBERO Framework



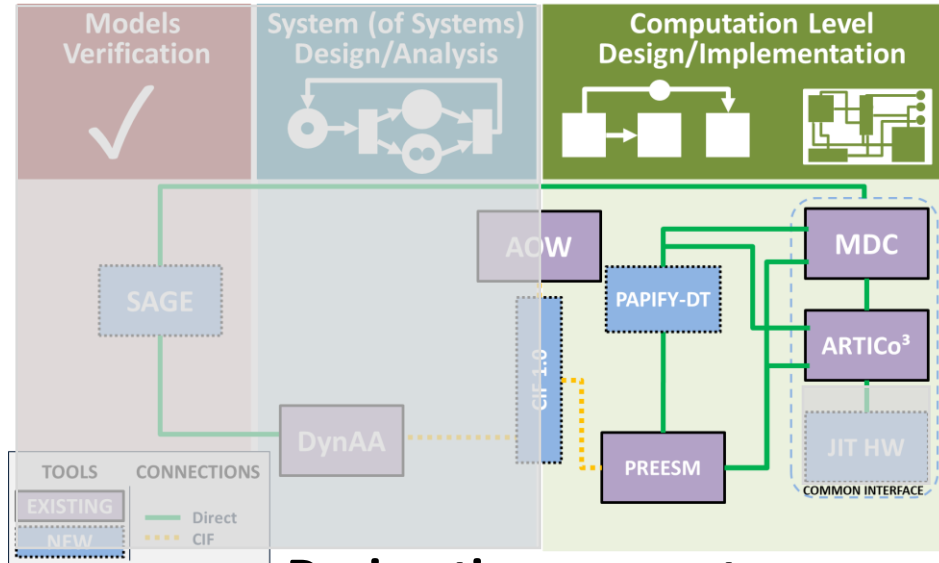
Design-time support



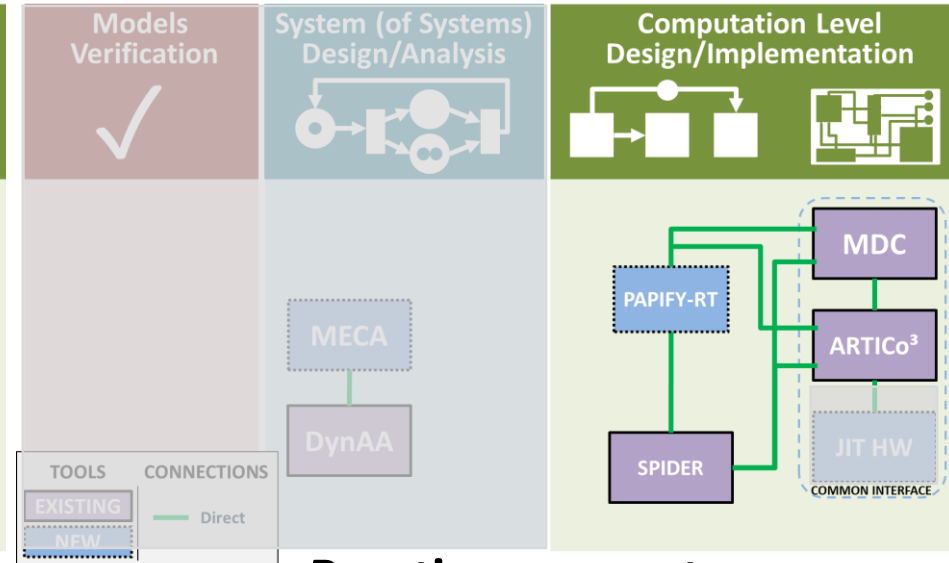
Run-time support



CERBERO Framework

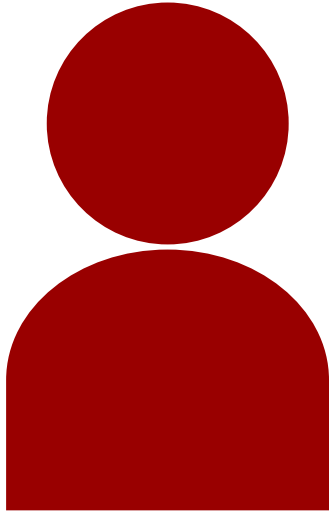


Design-time support



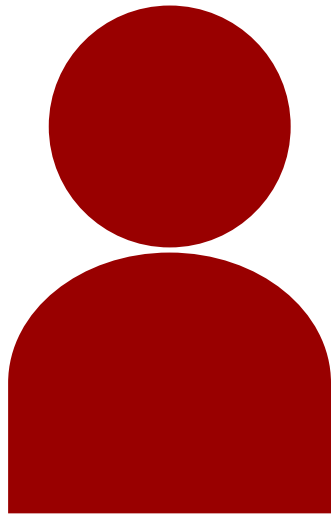
Run-time support

Adaptivity and Heterogeneity for Developers



Developer

Adaptivity and Heterogeneity for Developers



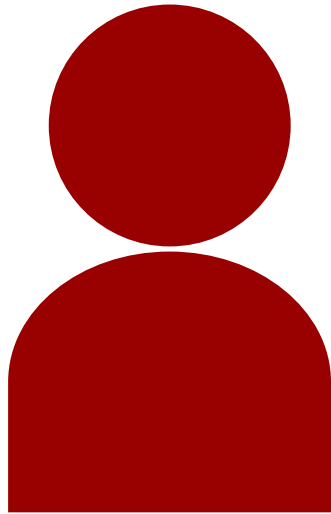
Developer

- SW skills



Application

Adaptivity and Heterogeneity for Developers

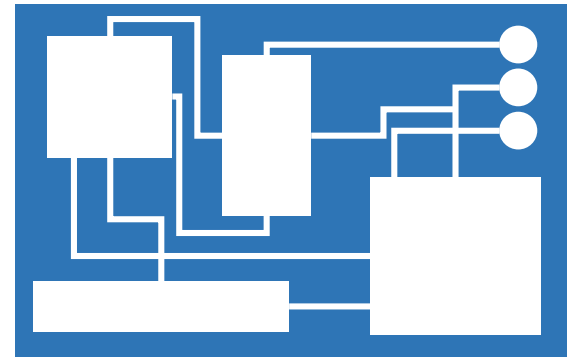


Developer

- SW skills
- HW notions

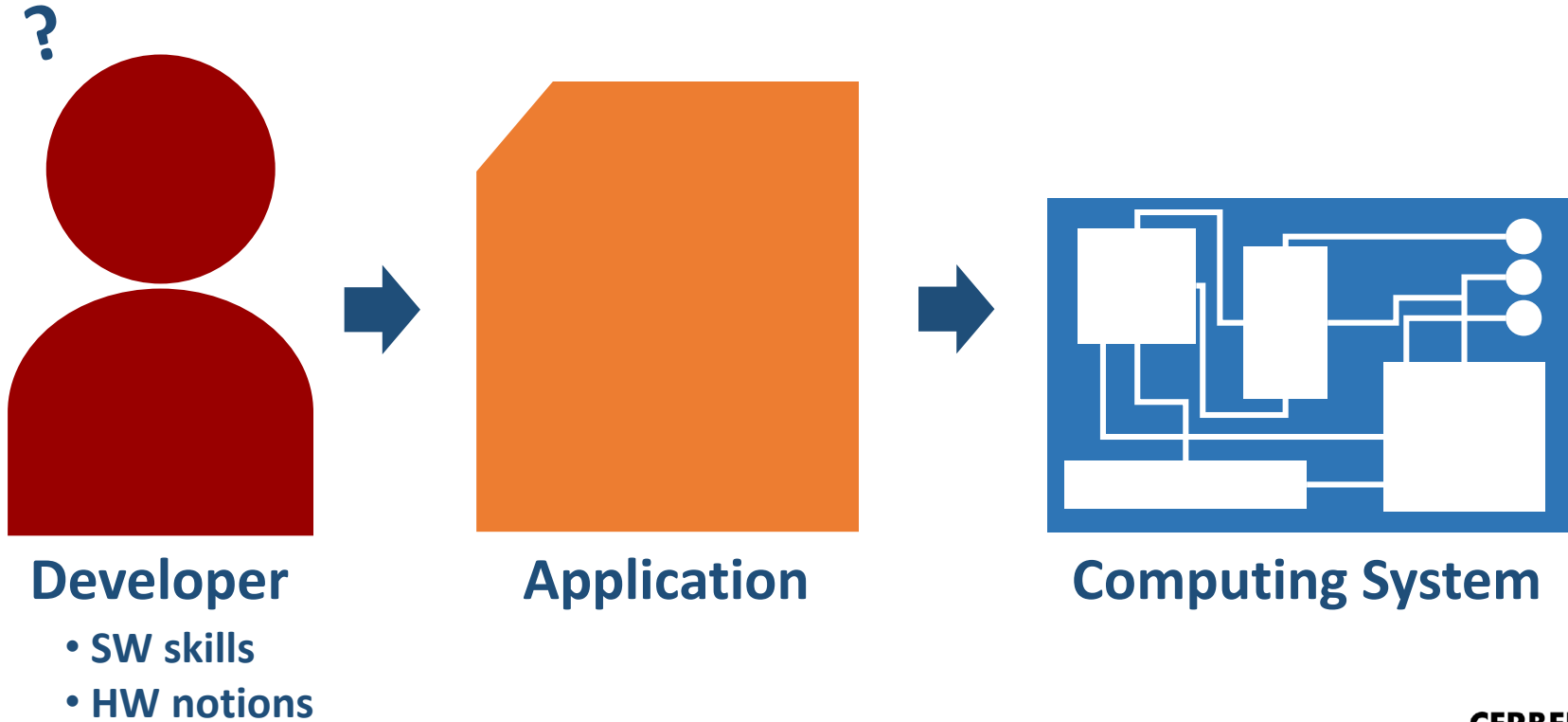


Application

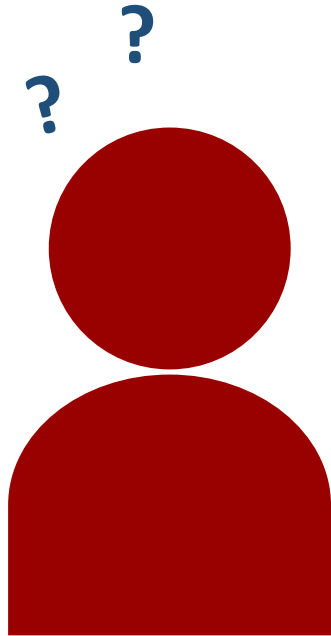


Computing System

Adaptivity and Heterogeneity for Developers



Adaptivity and Heterogeneity for Developers

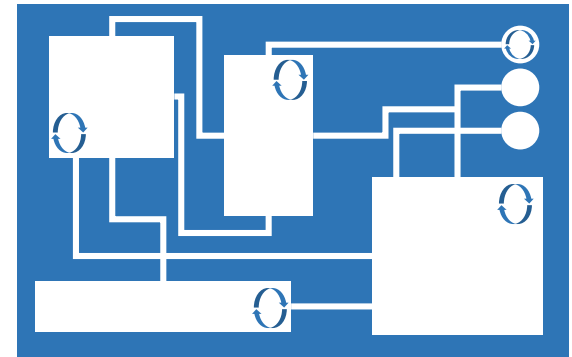


Developer

- SW skills
- HW notions



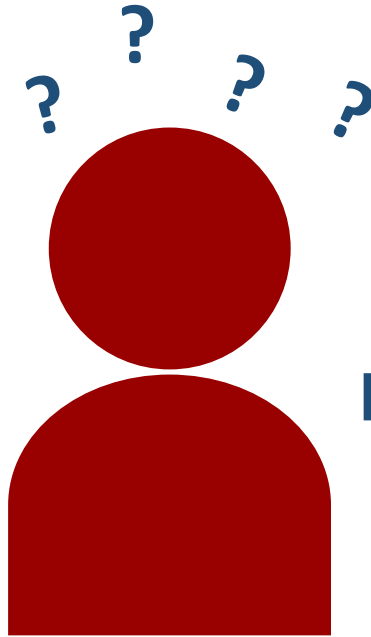
Application



Computing System

- Adaptivity

Adaptivity and Heterogeneity for Developers

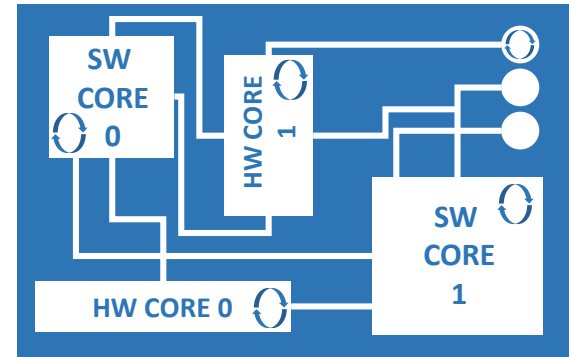


Developer

- SW skills
- HW notions



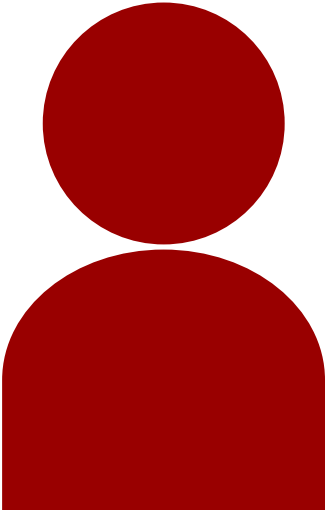
Application



- Adaptivity
- Heterogeneity

Computing System

CERBERO Toolchain Solution



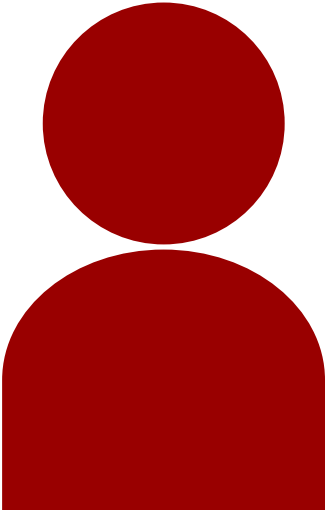
Developer

- SW skills
- HW notions



Application

CERBERO Toolchain Solution



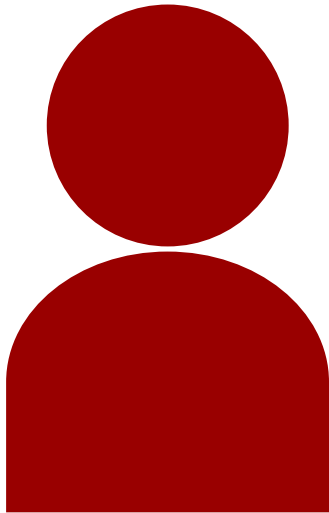
Developer

- SW skills
- HW notions



Application

CERBERO Toolchain Solution

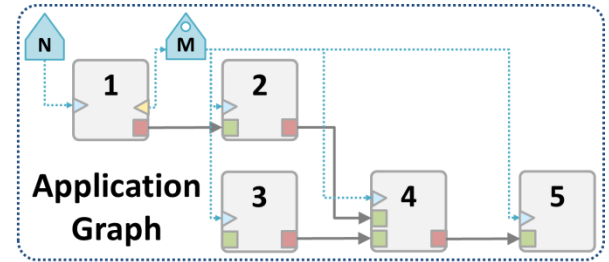


Developer

- SW skills
- HW notions

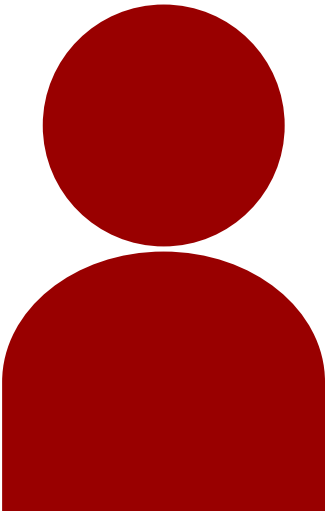


Application



**Dataflow
Application**

CERBERO Toolchain Solution

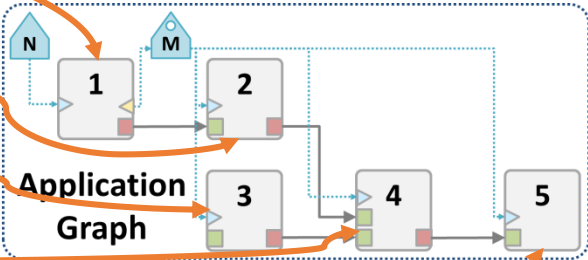


Developer

- SW skills
- HW notions



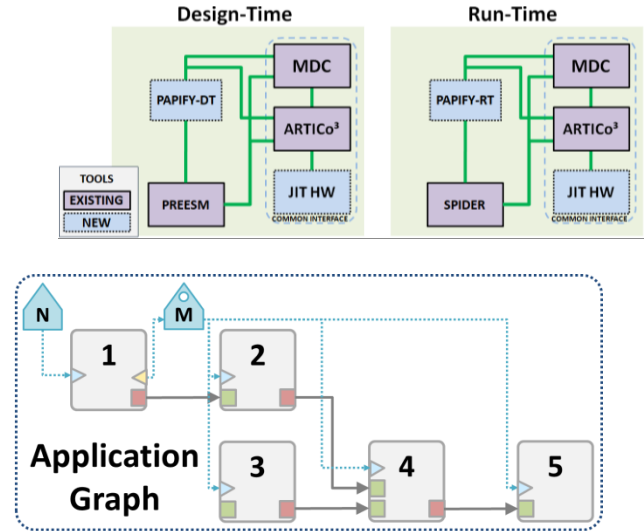
Application



Dataflow Application



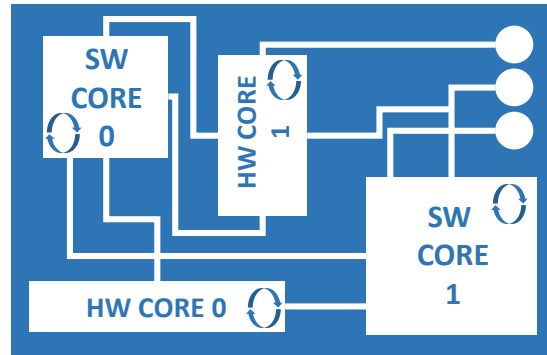
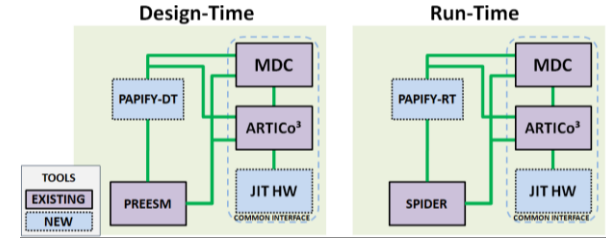
CERBERO Toolchain Solution



Dataflow
Application

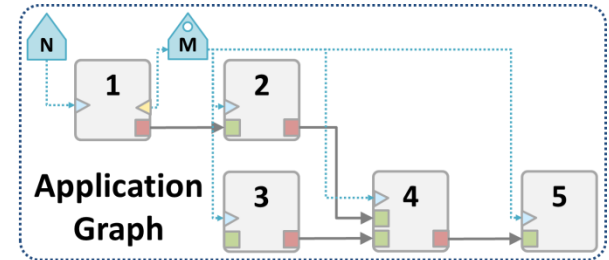


CERBERO Toolchain Solution



Computing System

- Heterogeneity
- Adaptivity

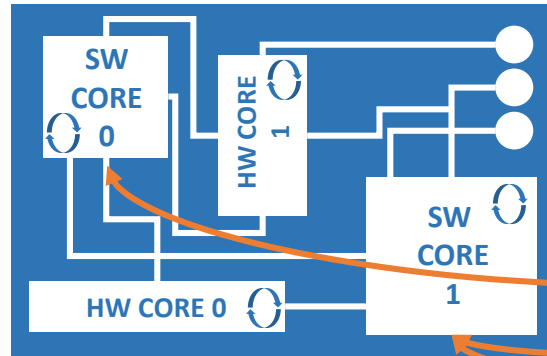
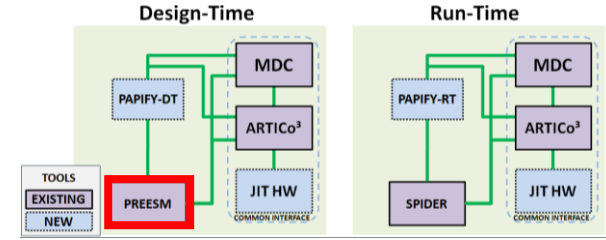


Dataflow Application



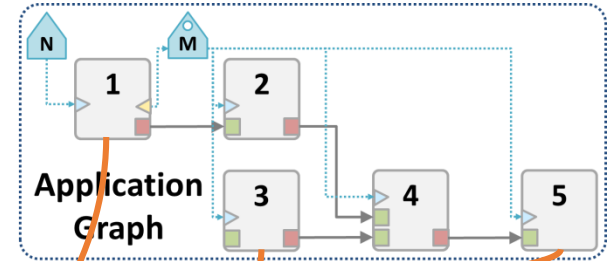
CERBERO Toolchain Solution

- Parallel SW Code Generation and Mapping



Computing System

- Heterogeneity
- Adaptivity

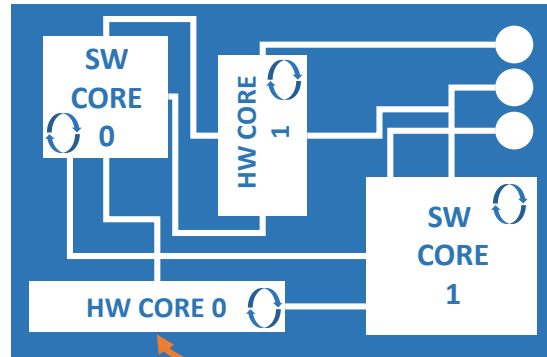
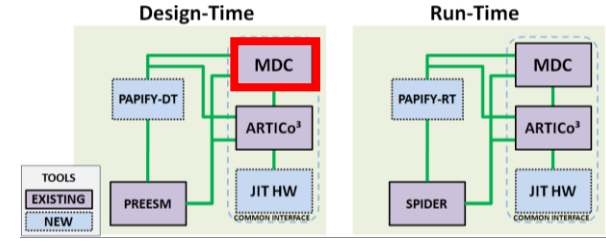


Dataflow Application



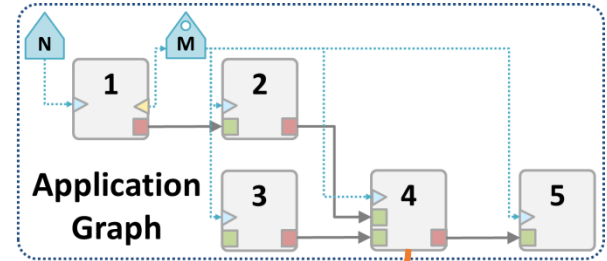
CERBERO Toolchain Solution

- Parallel SW Code Generation and Mapping
- Deployment of HW CGR Accelerator (fast reconf.)



Computing System

- Heterogeneity
- Adaptivity

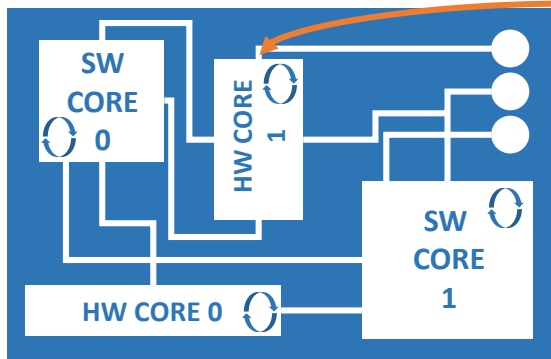


Dataflow Application



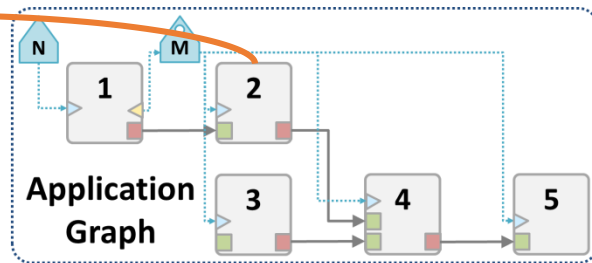
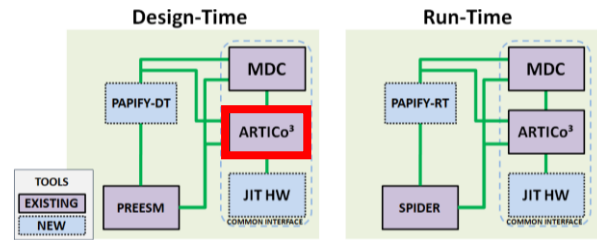
CERBERO Toolchain Solution

- Parallel SW Code Generation and Mapping
- Deployment of HW CGR Accelerator (fast reconf.)
- Deployment of HW DPR Multi-Accelerator (slow reconf., HW reuse)



Computing System

- Heterogeneity
- Adaptivity



Dataflow Application

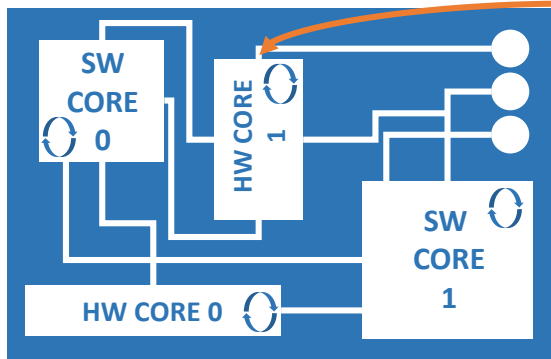


CERBERO Toolchain Solution

- Parallel SW Code Generation and Mapping
- Deployment of HW CGR Accelerator (fast reconf.)
- Deployment of HW DPR Multi-Accelerator (slow reconf., HW reuse)

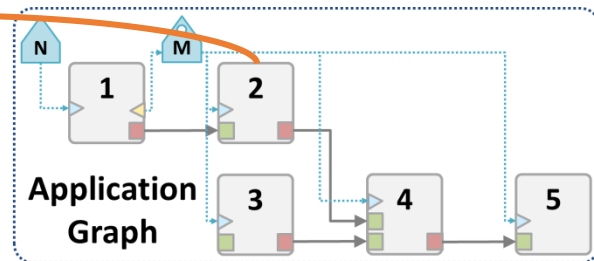
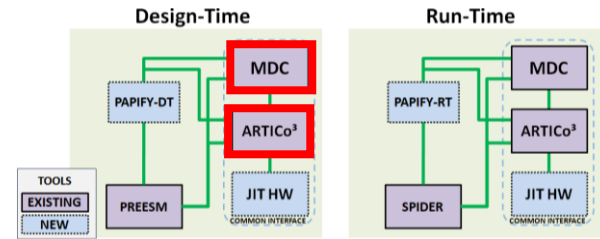


- Optionally with CGR inside (slow or fast reconf for adaptation)



Computing System

- Heterogeneity
- Adaptivity



Dataflow Application



CERBERO Toolchain Solution

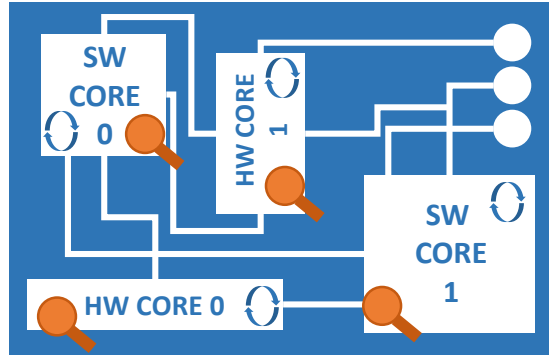
- Parallel SW Code Generation and Mapping
- Deployment of HW CGR Accelerator (fast reconf.)
- Deployment of HW DPR Multi-Accelerator (slow reconf., HW reuse)



- Optionally with CGR inside (slow or fast reconf for adaptation)

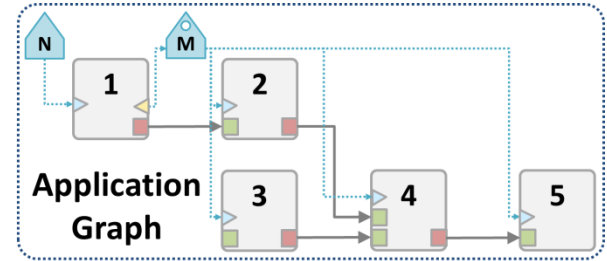
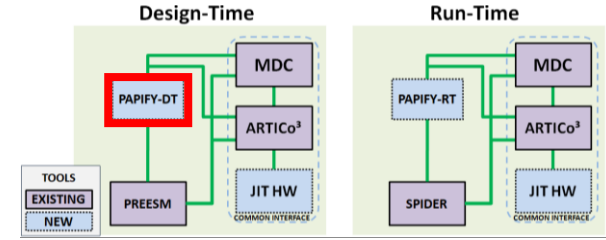


- Instrumentation for monitoring



Computing System

- Heterogeneity
- Adaptivity



Dataflow Application

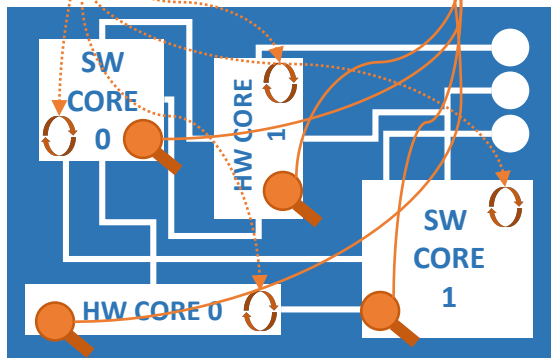


CERBERO Toolchain Solution

- Parallel SW Code Generation and Mapping
- Deployment of HW CGR Accelerator (fast reconf.)
- Deployment of HW DPR Multi-Accelerator (slow reconf., HW reuse)
 - Optionally with CGR inside (slow or fast reconf for adaptation)
- Instrumentation for monitoring
- Run-time management and self-adaptation (not covered in the tutorial)

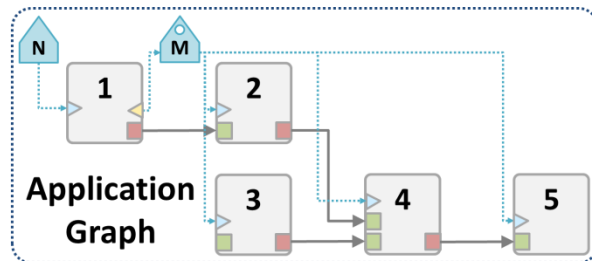
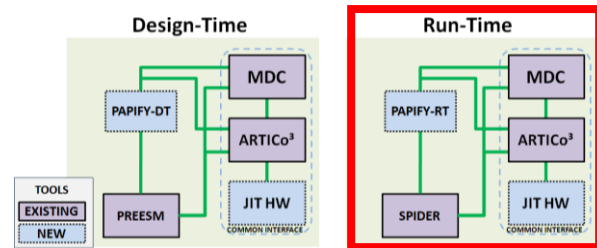


other internal/external requirements



Computing System

- Heterogeneity
- Adaptivity

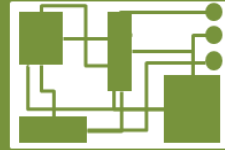


Dataflow Application



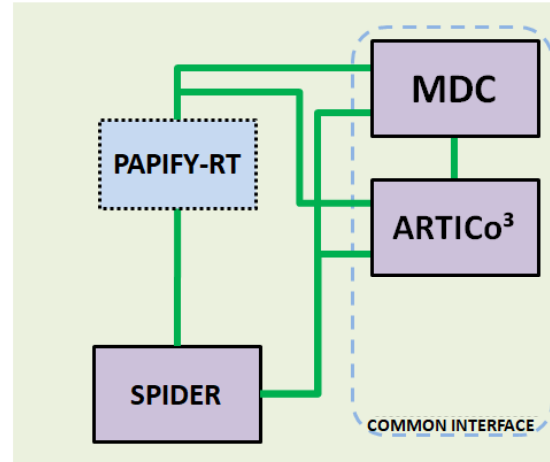
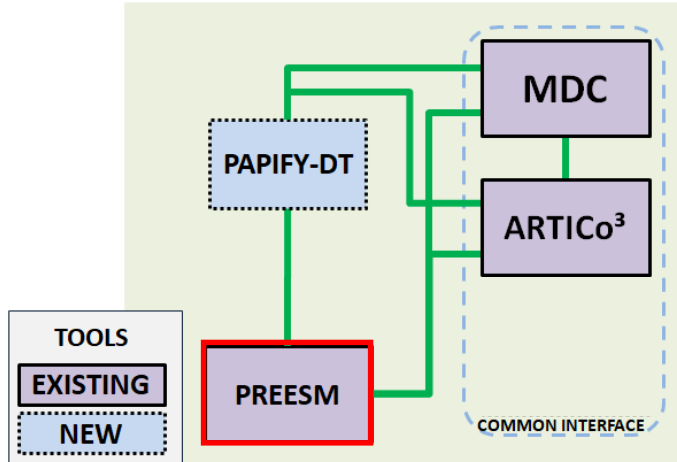
CERBERO Framework - Computation Level

Computation Level Design/Implementation



Design-Time

Run-Time



PREESM

Design-time

Parallel-
application
development.
Code generation
and re-use
capabilities.

INSA



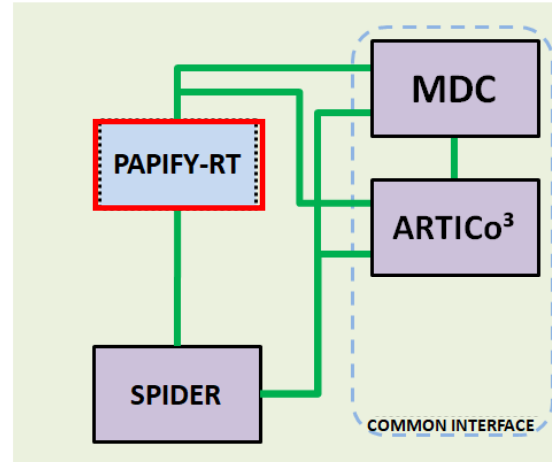
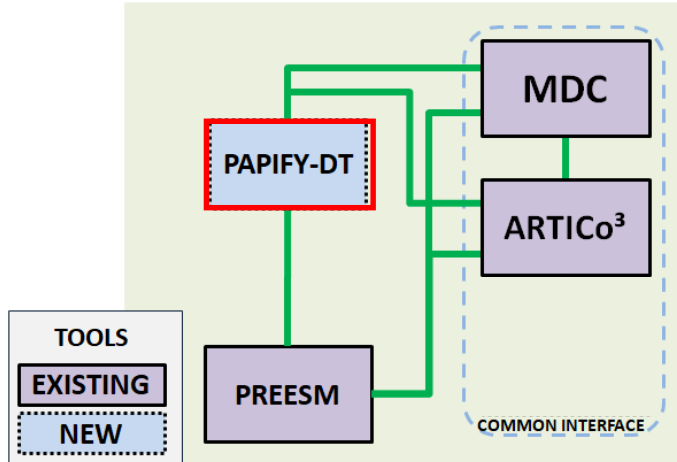
CERBERO Framework - Computation Level

Computation Level
Design/Implementation



Design-Time

Run-Time



PAPIFY

Design-time

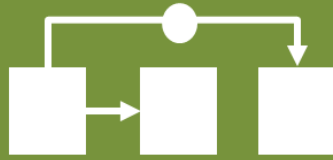
Run-time

Automatic code instrumentation for performance monitoring
Provides a large set of run-time execution information to SPIDER



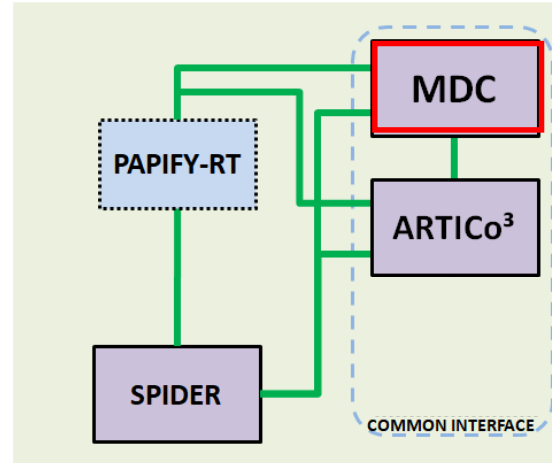
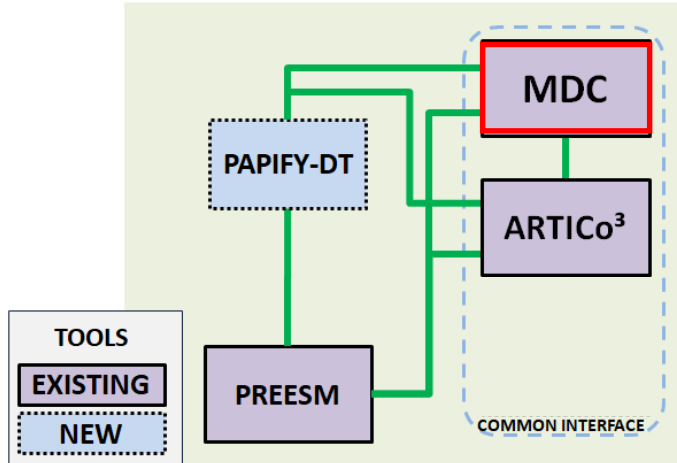
CERBERO Framework - Computation Level

Computation Level
Design/Implementation



Design-Time

Run-Time



MDC

Design-time

Run-time

Dataflow-to-
hardware

framework for
generation of CGR
accelerators

Fast HW

reconfiguration



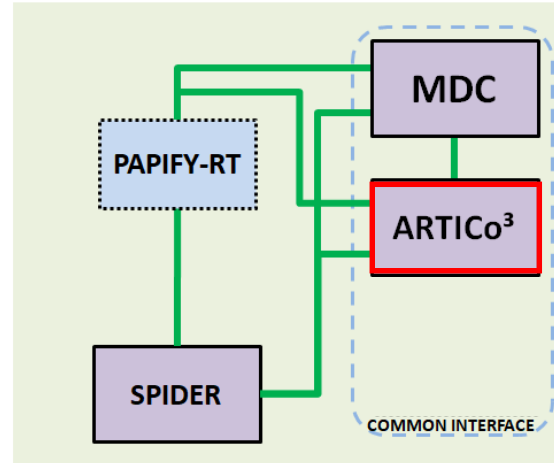
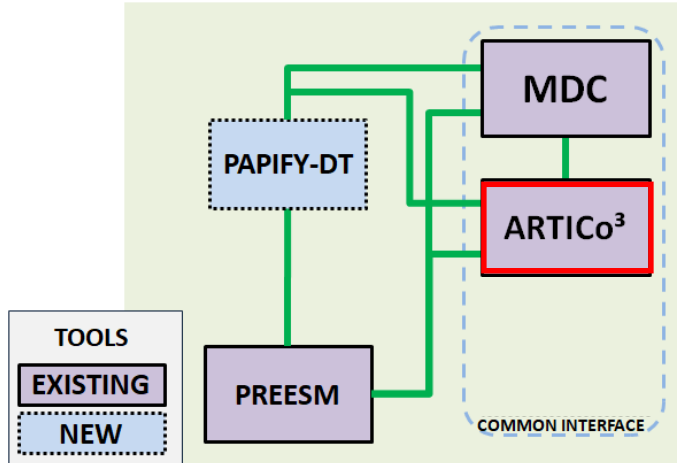
CERBERO Framework - Computation Level

Computation Level
Design/Implementation



Design-Time

Run-Time



ARTICo³

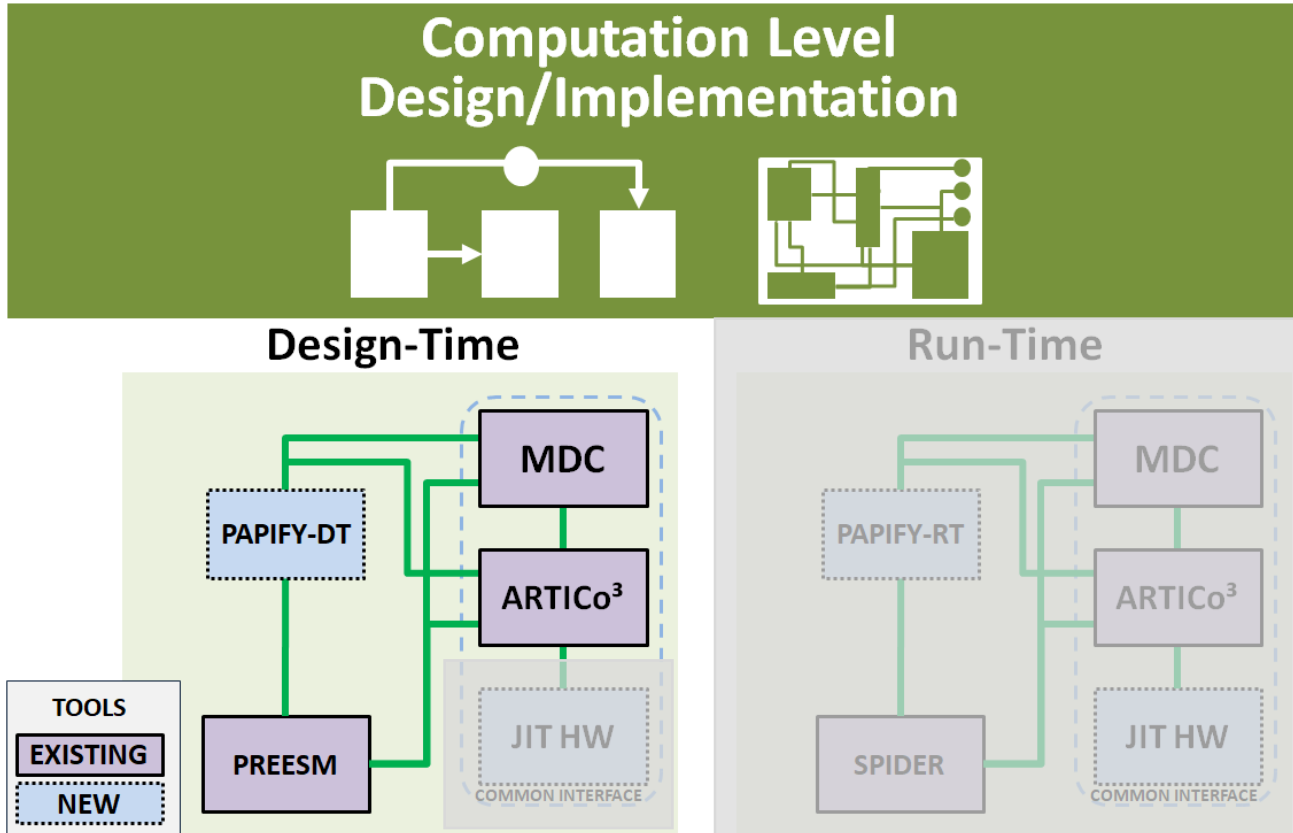
Design-time

Run-time

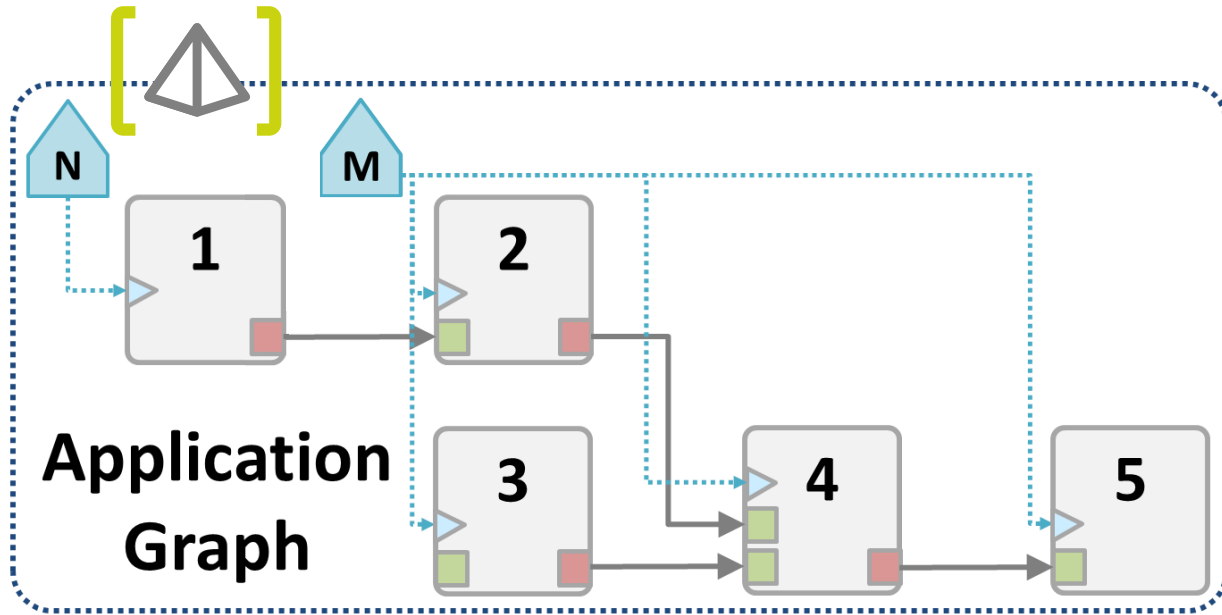
DPR-enabled
multi-accelerator
Scalable
parallelism and
fault tolerance



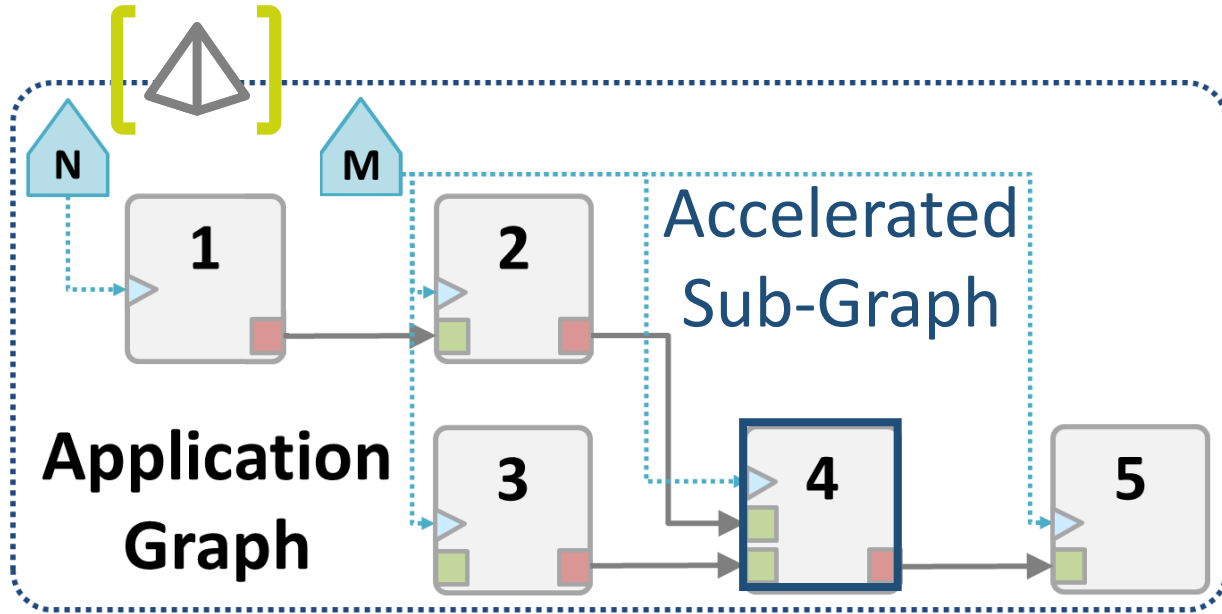
CERBERO Framework – Self-Adaptive System Design



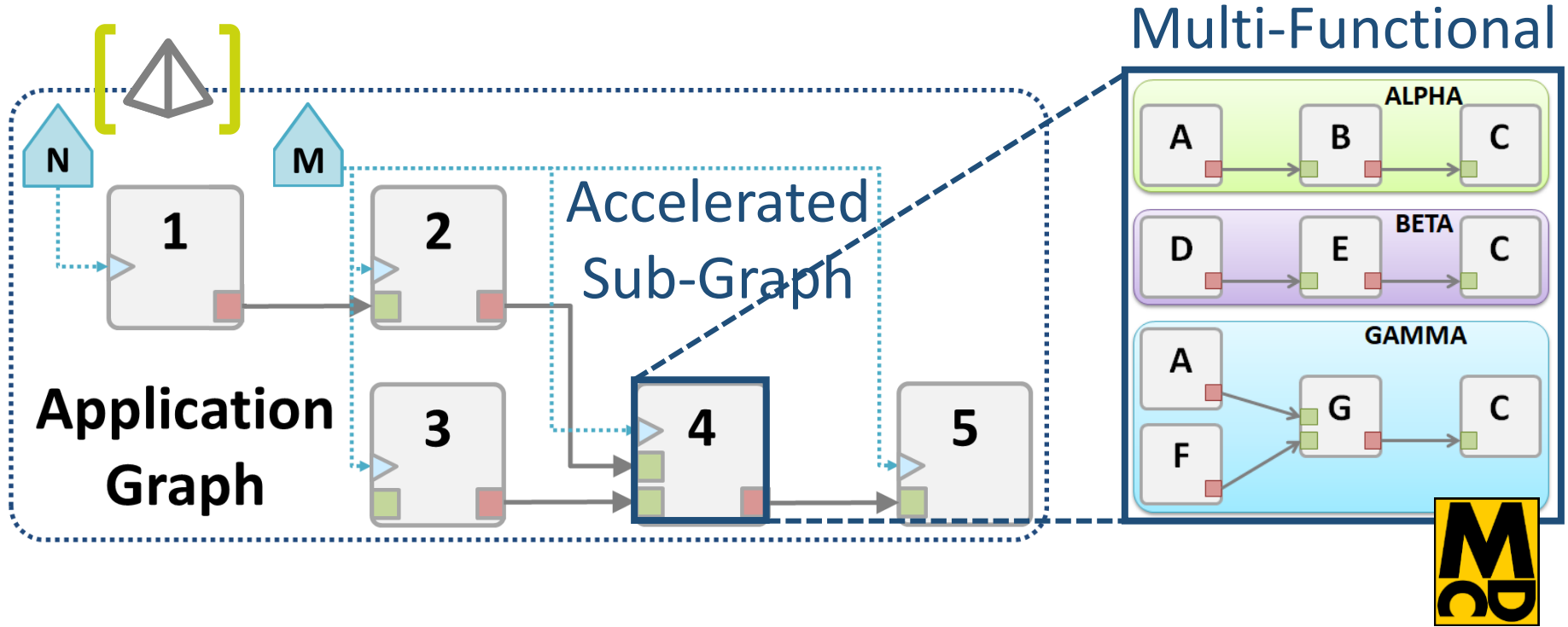
Design-Time Support: for HW Run-Time Adaptivity



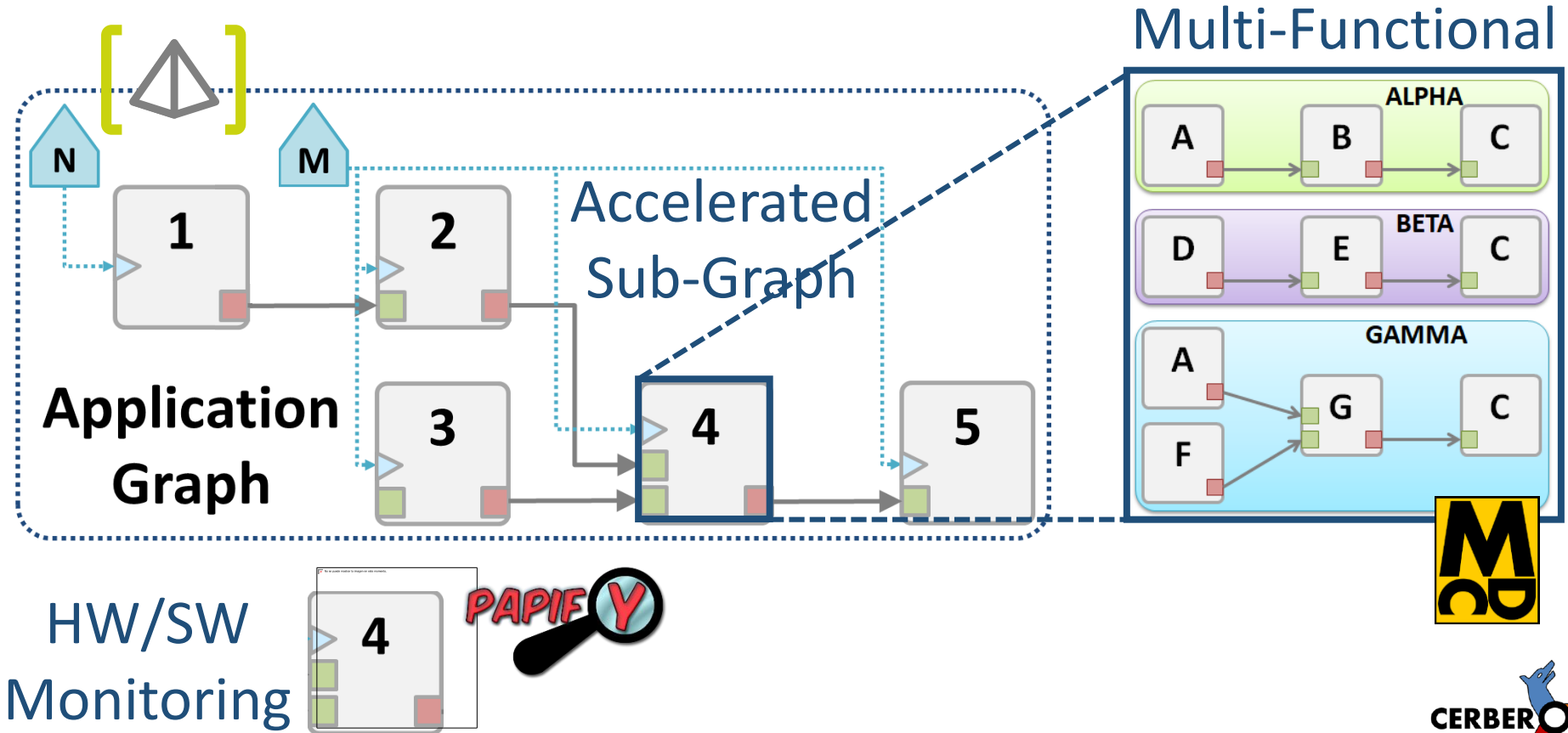
Design-Time Support: for HW Run-Time Adaptivity



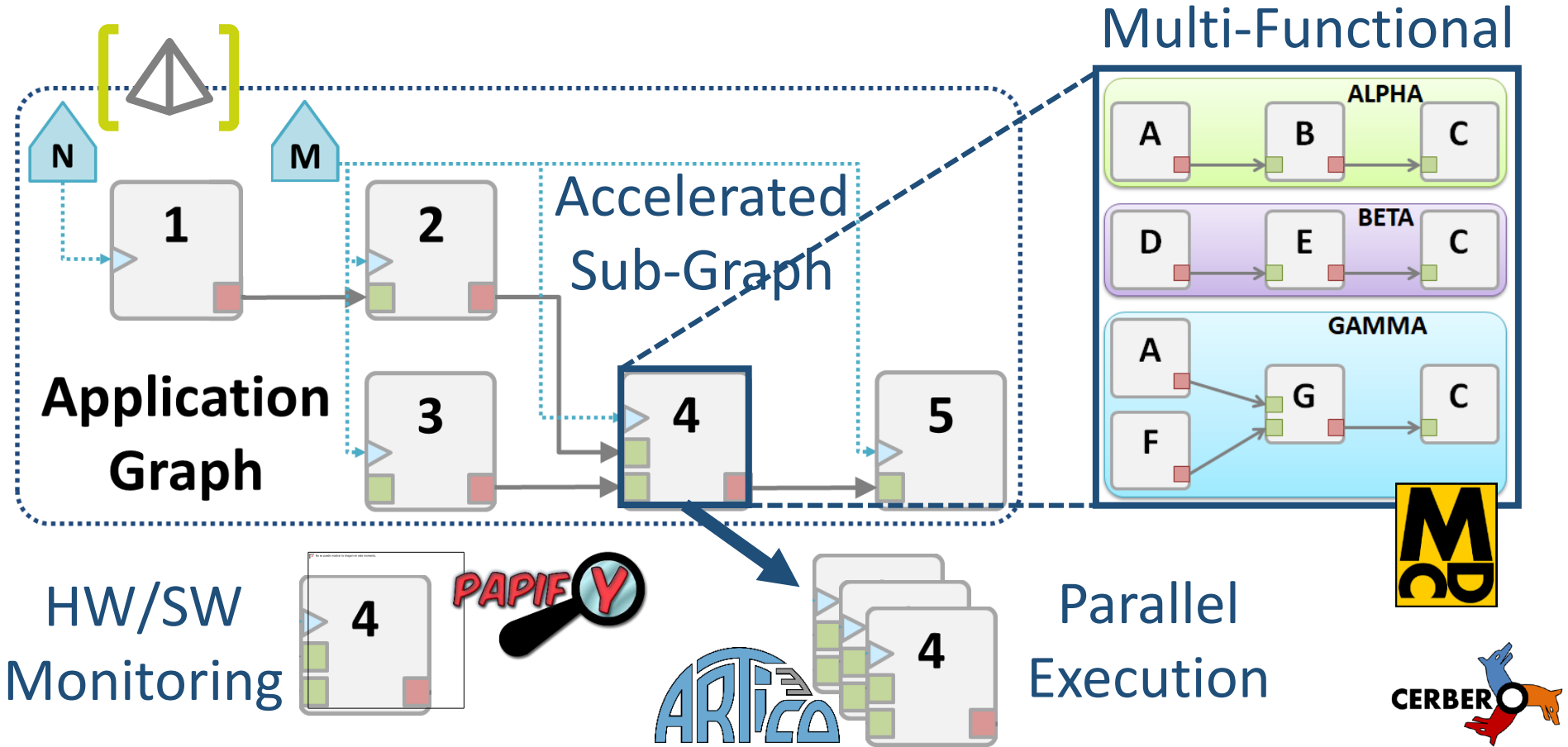
Design-Time Support: for HW Run-Time Adaptivity



Design-Time Support: for HW Run-Time Adaptivity



Design-Time Support: for HW Run-Time Adaptivity



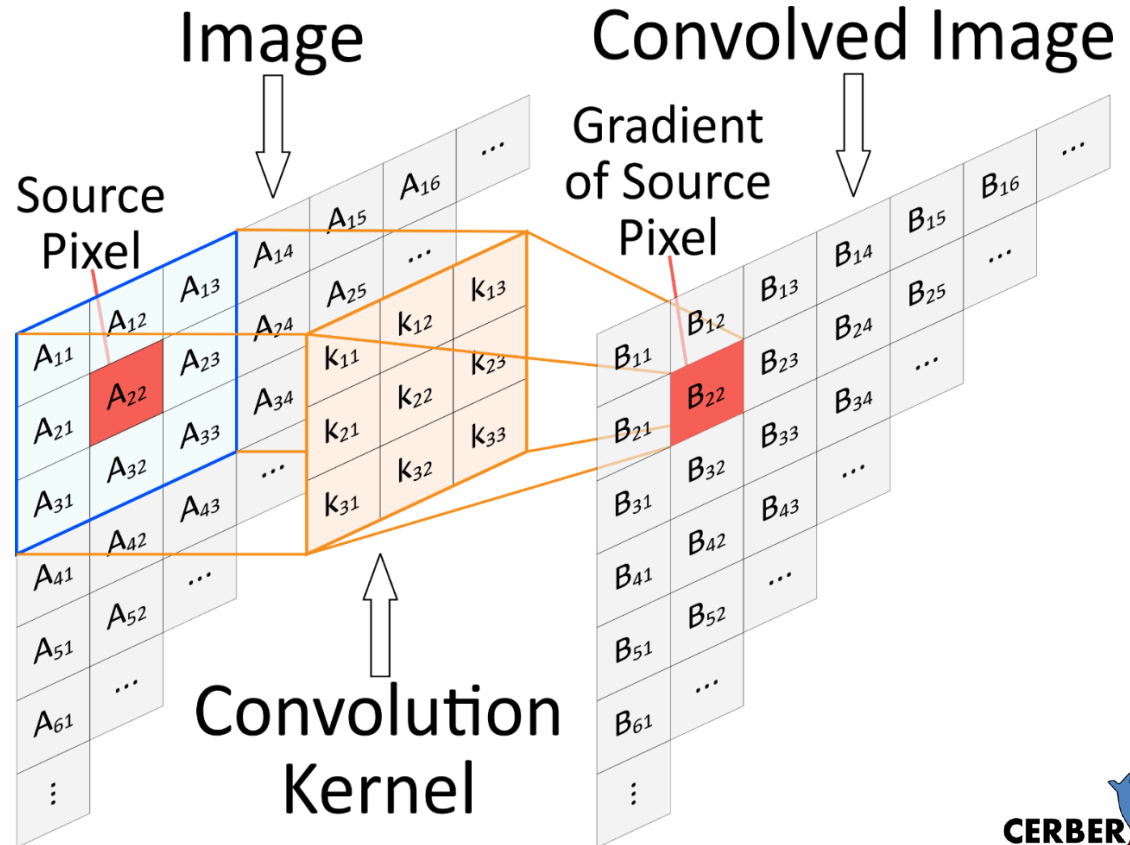
Tutorial Use Case

Sobel Operator

$$G = \sqrt{G_x^2 + G_y^2}$$

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$



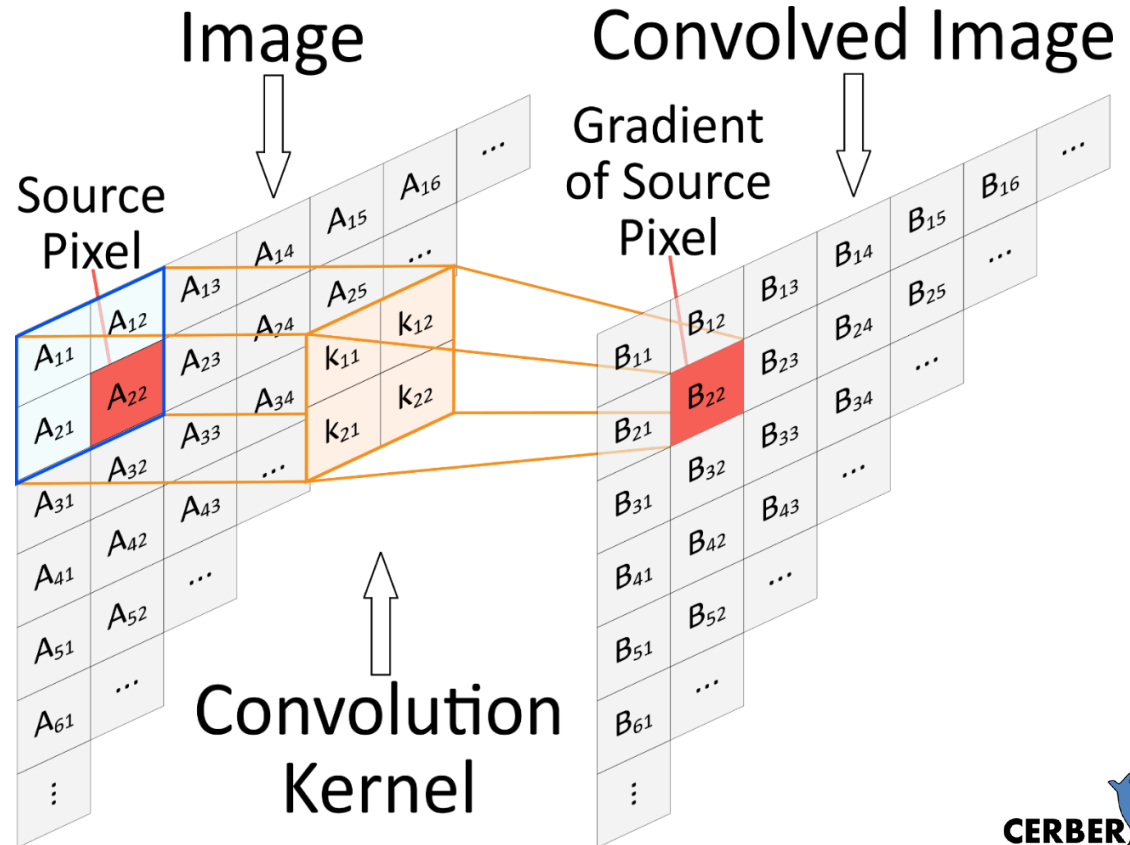
Tutorial Use Case

Roberts Operator

$$G = \sqrt{G_x^2 + G_y^2}$$

$$G_x = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$

$$G_y = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}$$



HiPEAC 2020

20-22 January, 2020, Bologna, Italy



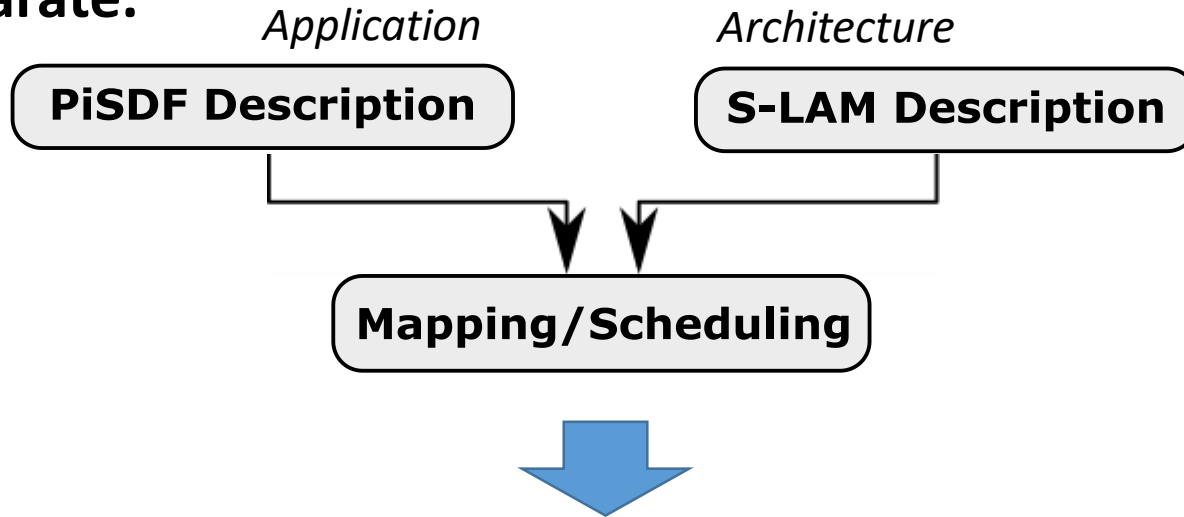
Hardware/Software Model-Based Design for Design Space Exploration and Code Generation



Horizon 2020
European Union funding
for Research & Innovation

PREESM

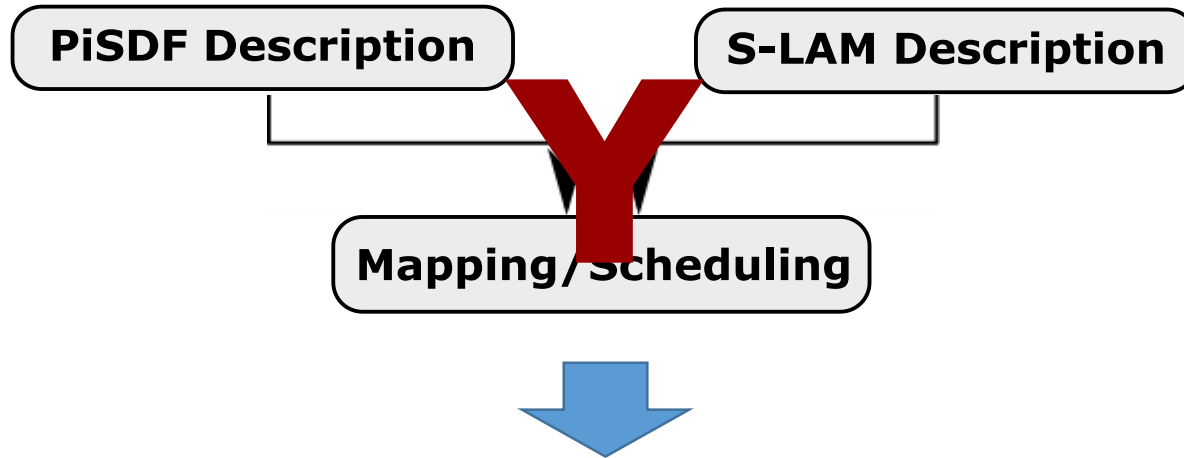
- Y co-design approach: keep architecture and application separate.



AUTOMATIC CODE GENERATION for RAPID PROTOTYPING

PREESM

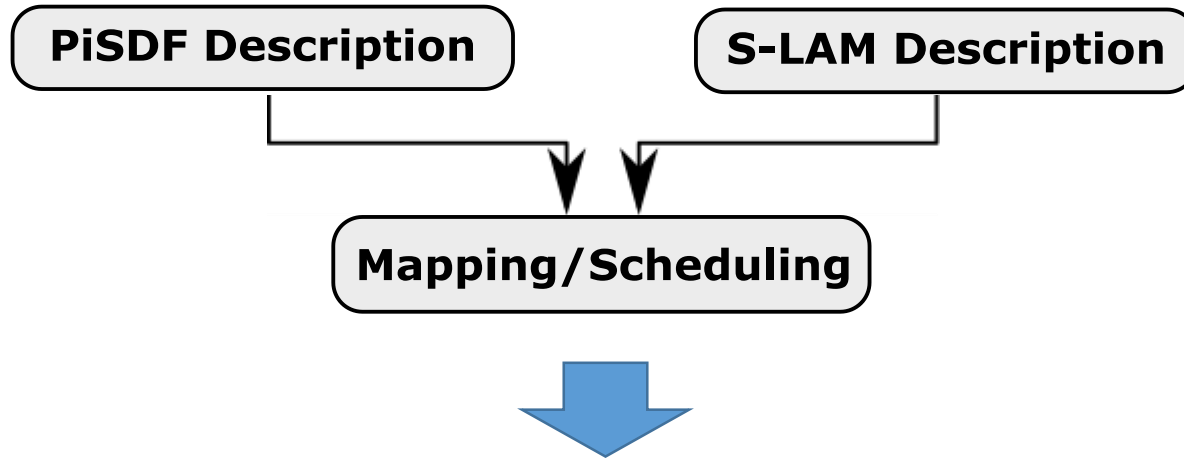
- Y co-design approach: keep architecture and application separate.



AUTOMATIC CODE GENERATION for RAPID PROTOTYPING

PREESM

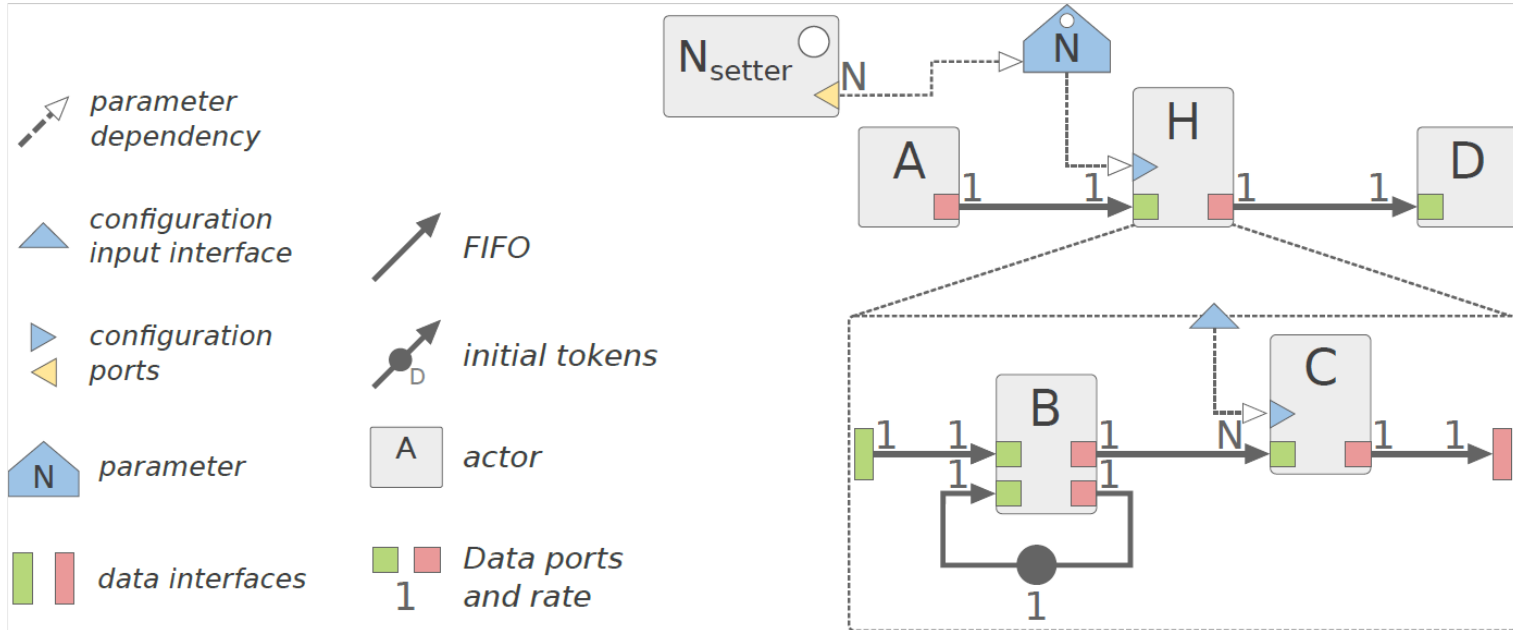
- Y co-design approach: keep architecture and application separate.



**AUTOMATIC CODE GENERATION for RAPID PROTOTYPING
for MPSoCs (CPUs + FPGA)**

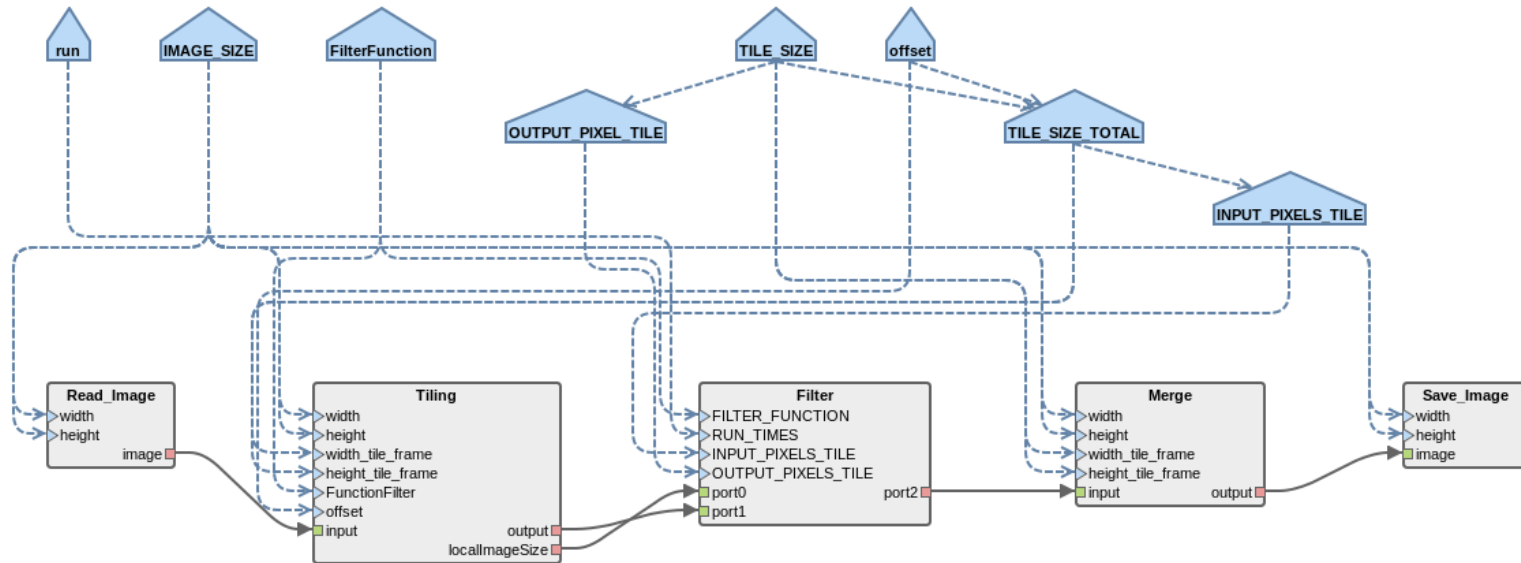
PiSDF: Parameterized and Interfaced SDF

- Semantics:

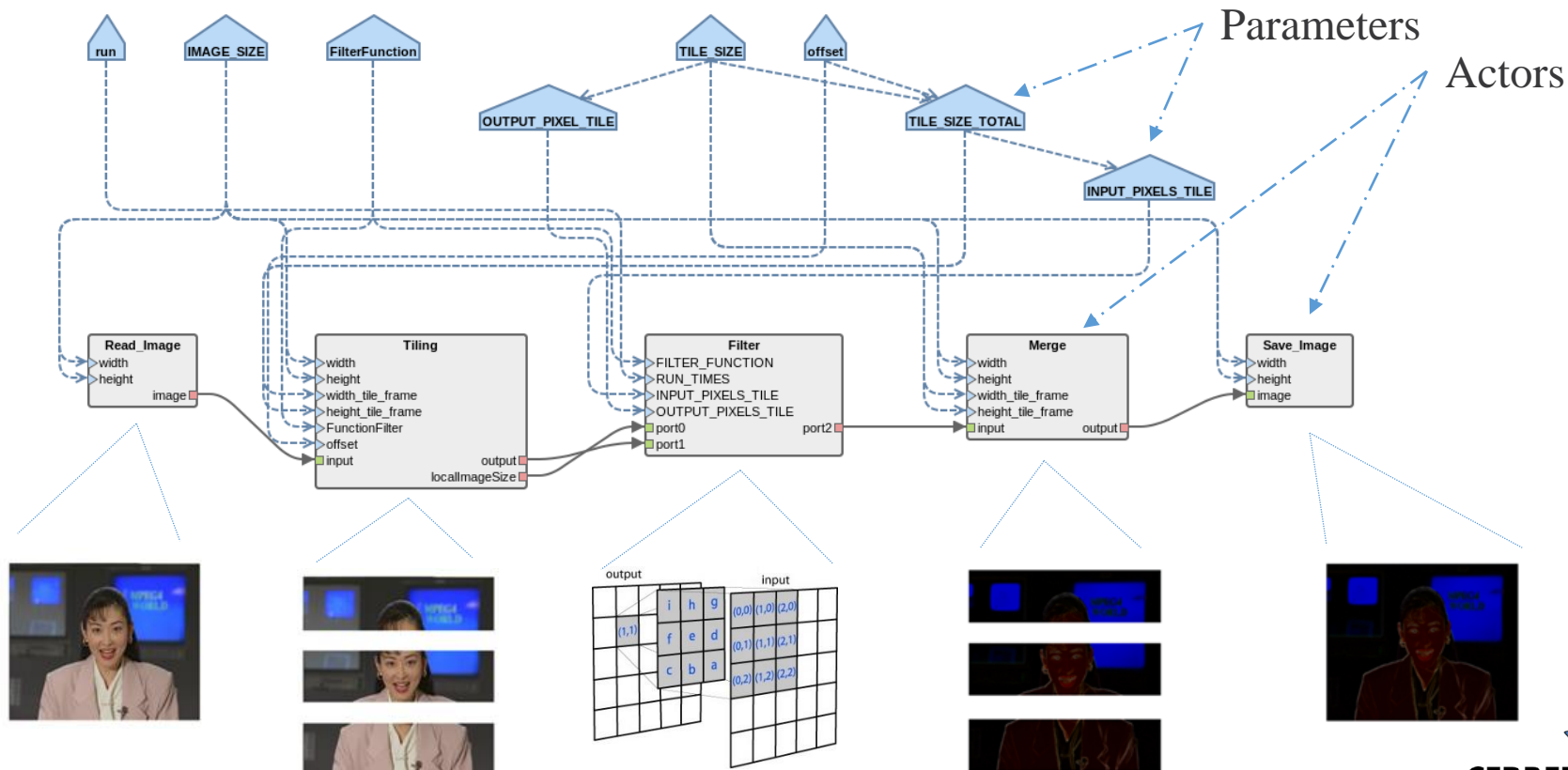


PiSDF: Parameterized and Interfaced SDF

- Application :

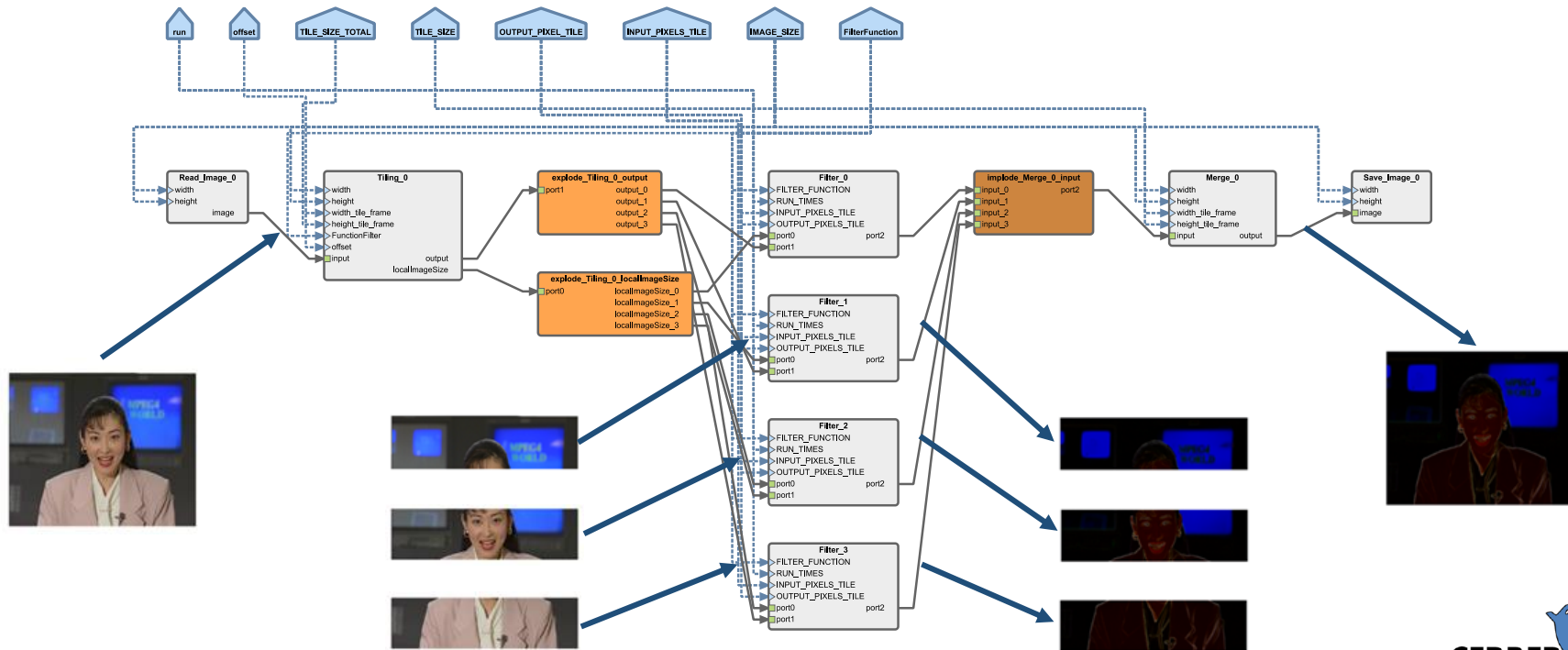


Actor Specification

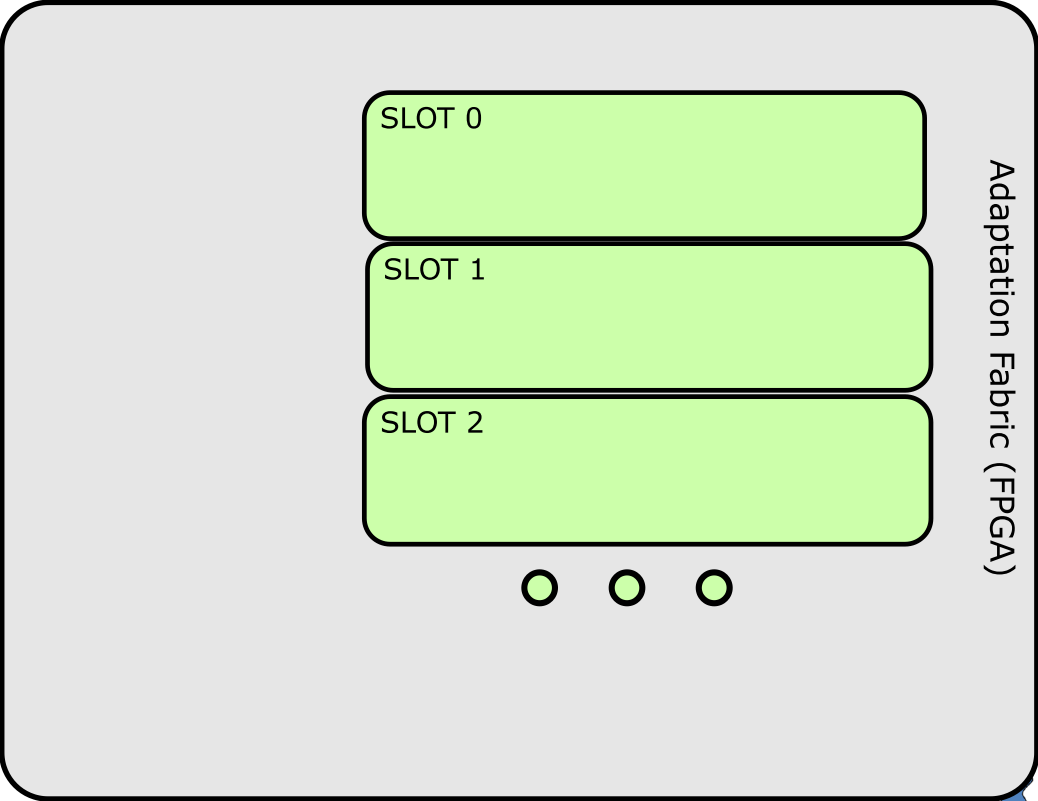
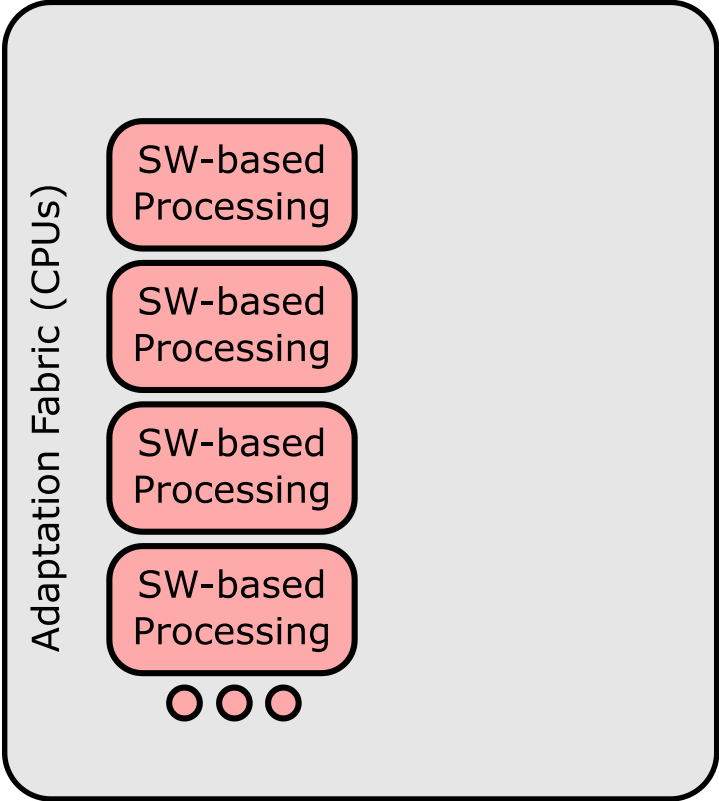


Exposed Parallelism

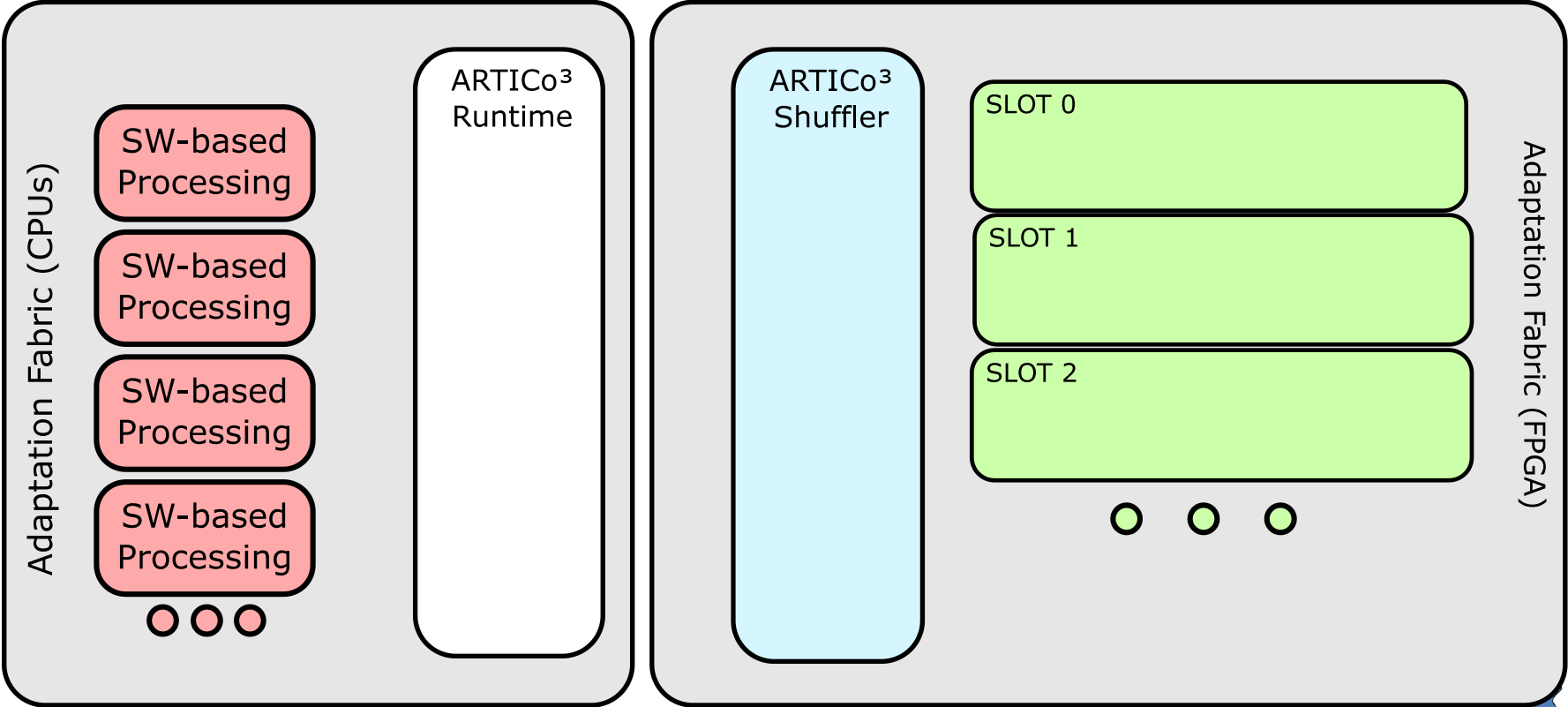
Equivalent single-rate graph where each edge has equal production and consumption rates of token.



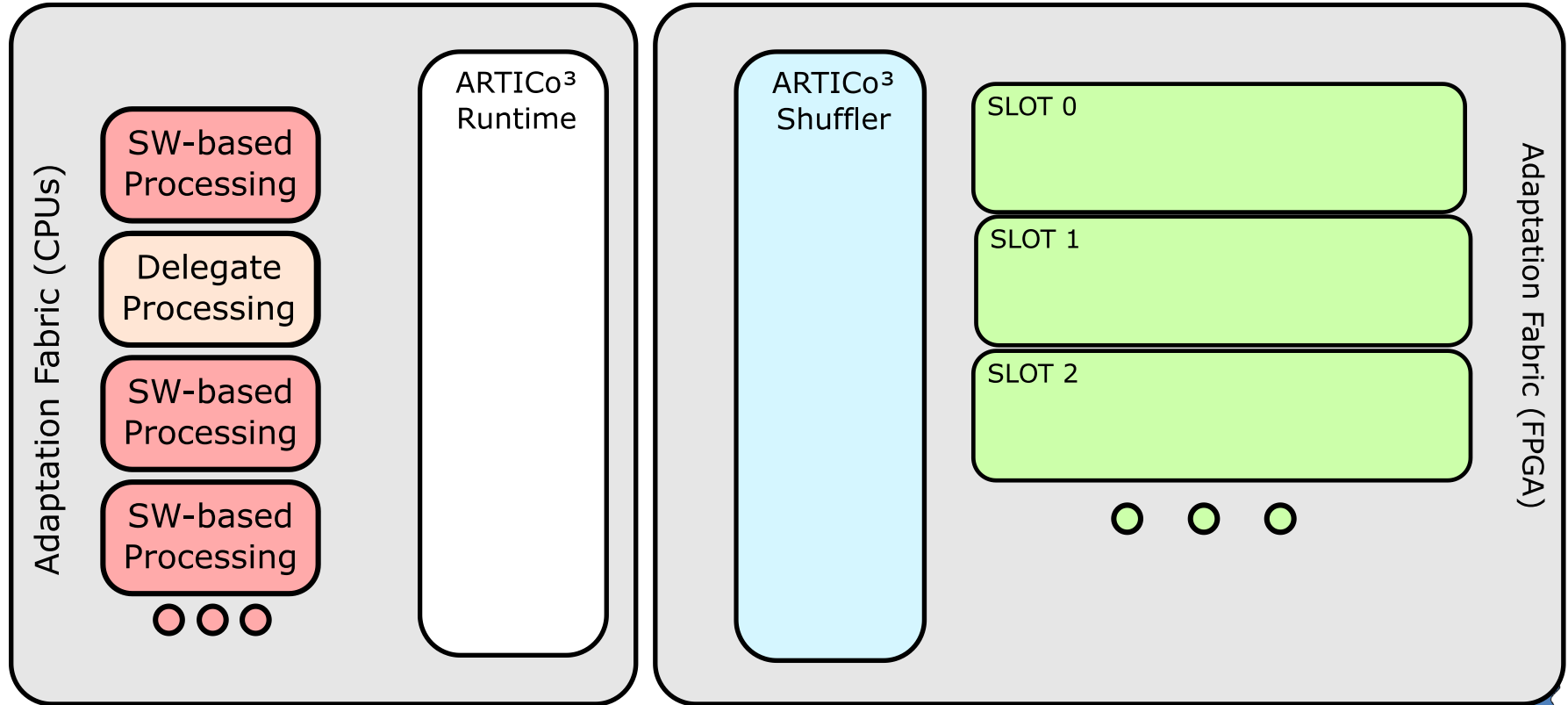
Heterogeneous Device



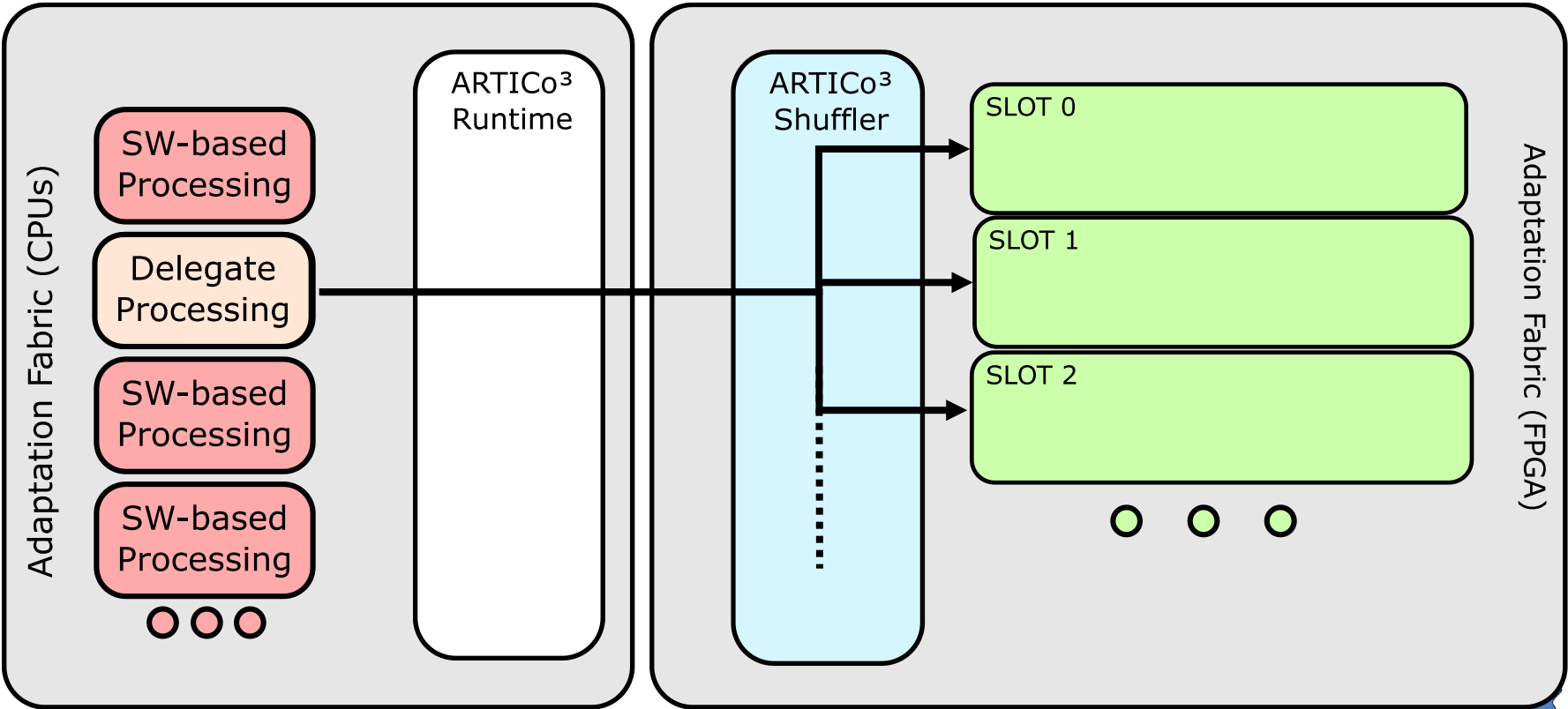
Heterogeneous Device



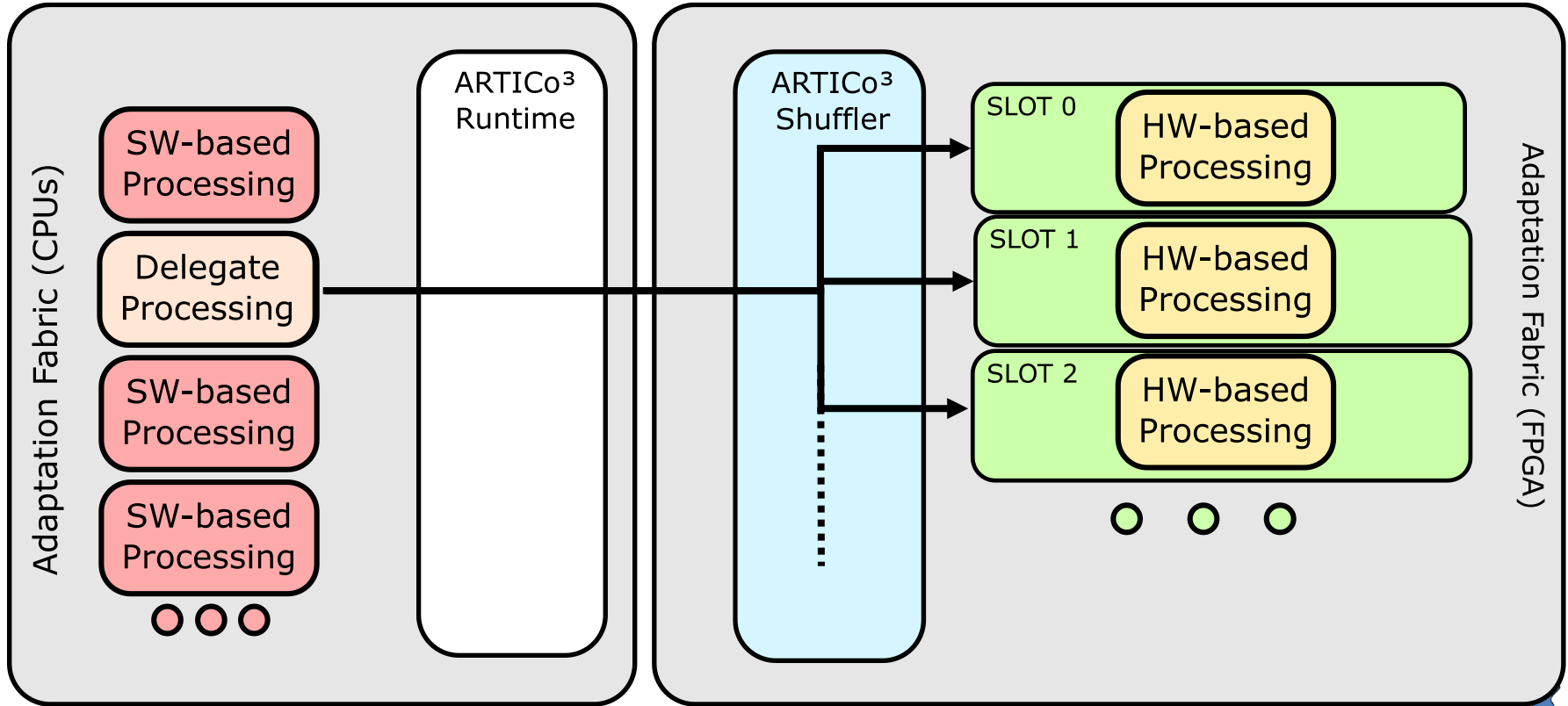
Heterogeneous Device



Heterogeneous Device

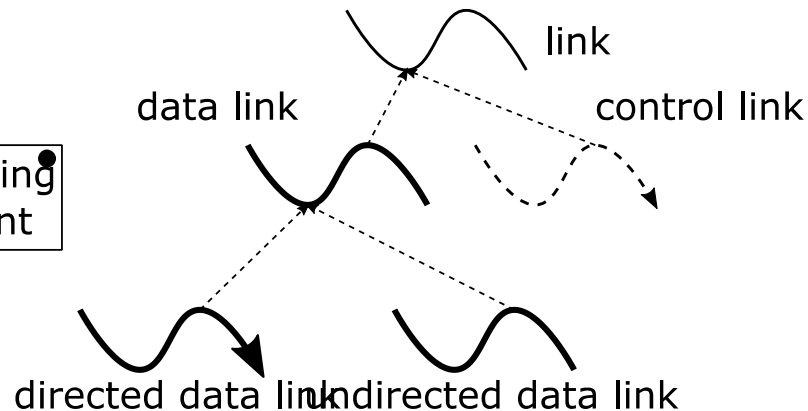
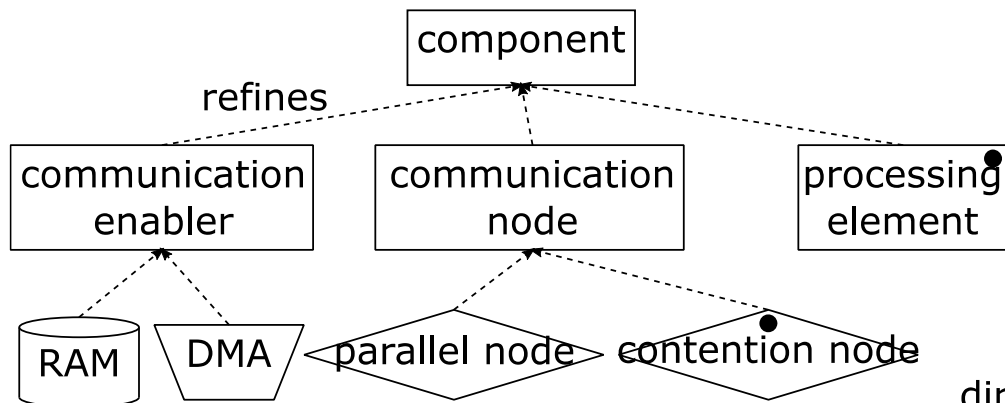


Heterogeneous Device



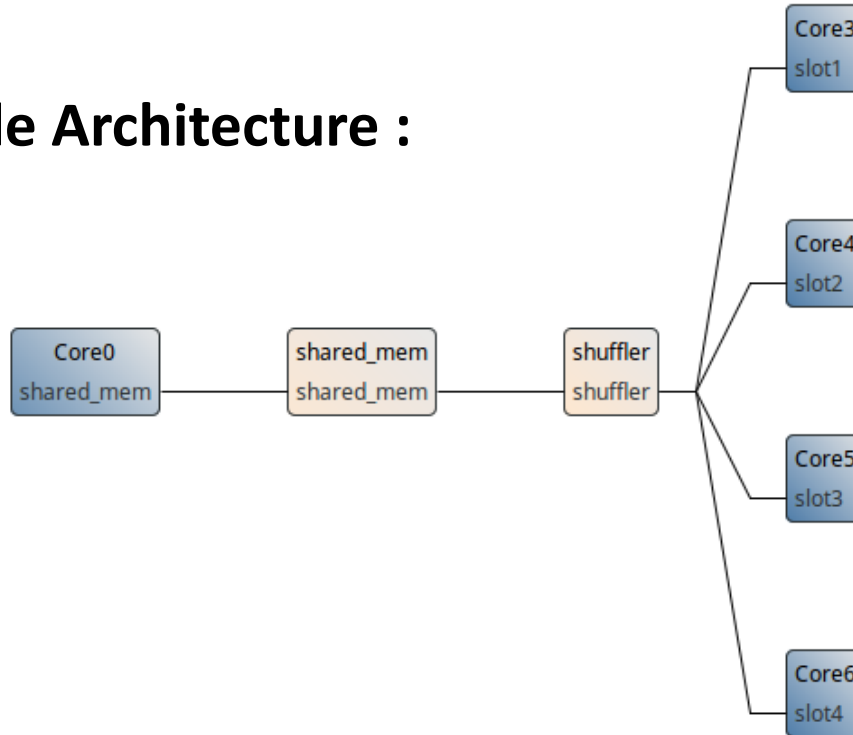
S-LAM: System-Level Architecture Model

- **S-LAM typology:**



S-LAM: System-Level Architecture Model

- **S-LAM Example Architecture :**



PREESM Mapping and Scheduling

The screenshot displays the PREESM IDE interface within an Oracle VM VirtualBox window titled "Xubuntu_1 [Running]". The IDE window is titled "workspace - Implementation - Preesm" and features a menu bar (File, Machine, View, Input, Devices, Help) and a toolbar with various icons. The Project Explorer on the left shows a project structure for "tutorialSummerSchoolFixedTile" with folders for "Algo", "Archi", "Code", "Gantt", "Scenarios", "SDF", and "Workflows". The "Codegen.workflow" file is selected under the "Workflows" folder. The Outline window at the bottom left indicates "An outline is not available."

The main workspace displays a "Solution Gantt" chart. The x-axis represents "Time" in cycles, ranging from 0 to 1750 with major ticks every 250 cycles. The y-axis lists "Operators" for "Core0", "Slot1", "Slot2", "Slot3", and "Slot4". The Gantt chart shows the execution of tasks: "Read_Tiling" on Core0 from 0 to 250 cycles, followed by "Filter" tasks on all slots from 250 to 1500 cycles, and "Merge Save_" on Core0 from 1500 to 1750 cycles. Below the Gantt chart, there are tabs for "Gantt", "Loads", "Work, Span and Achieved Speedup", "Properties", "Tasks", "Problems", and "Console".

The Console window shows the following output:

```
DFTools Workflow console
20:26:32 NOTICE: [HARDWARE] copying buffers and subbuffers and definitions.
20:26:32 NOTICE: [HARDWARE] number of FunctionCallImpl 51
20:26:32 NOTICE: [HARDWARE] number of OutputDataTransfer inserted is 51
20:26:32 NOTICE: [HARDWARE] PE_id set up to 4
20:26:32 NOTICE: [HARDWARE] End of the Hardware preProcessing.
20:26:32 NOTICE: Workflow execution finished: /tutorialSummerSchoolFixedTile/Workflows/Codegen.workflow.
```



HiPEAC 2020

20-22 January, 2020, Bologna, Italy

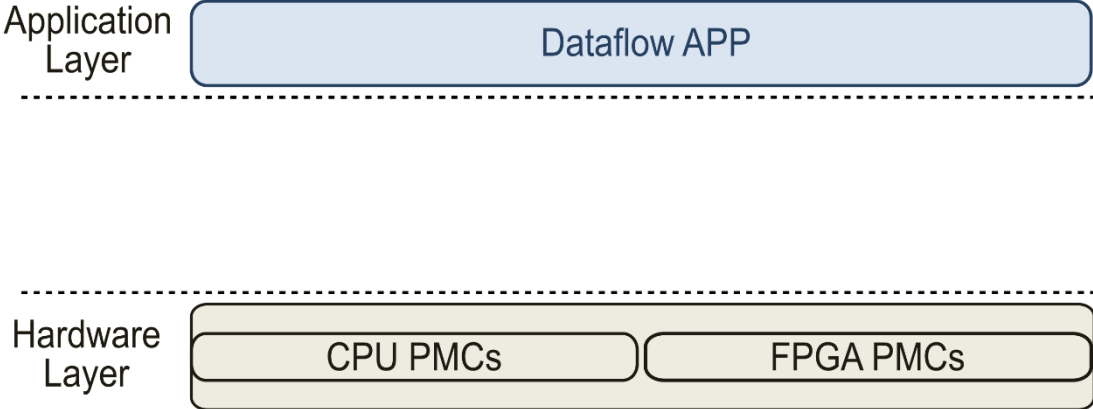


System Performance Monitoring

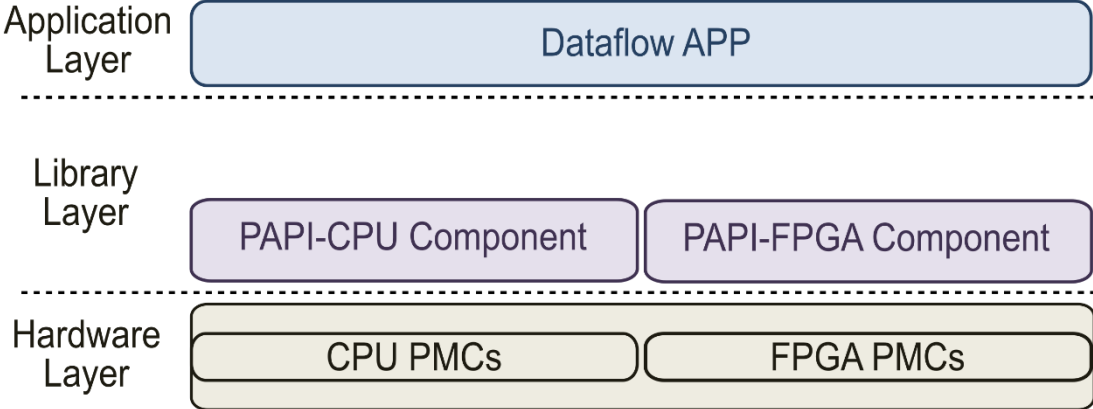


Horizon 2020
European Union funding
for Research & Innovation

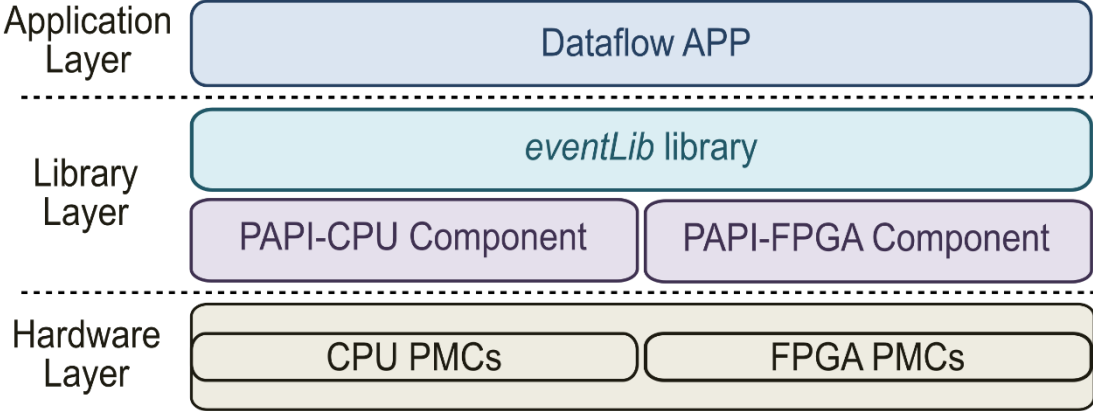
CERBERO Monitoring: PAPIFY



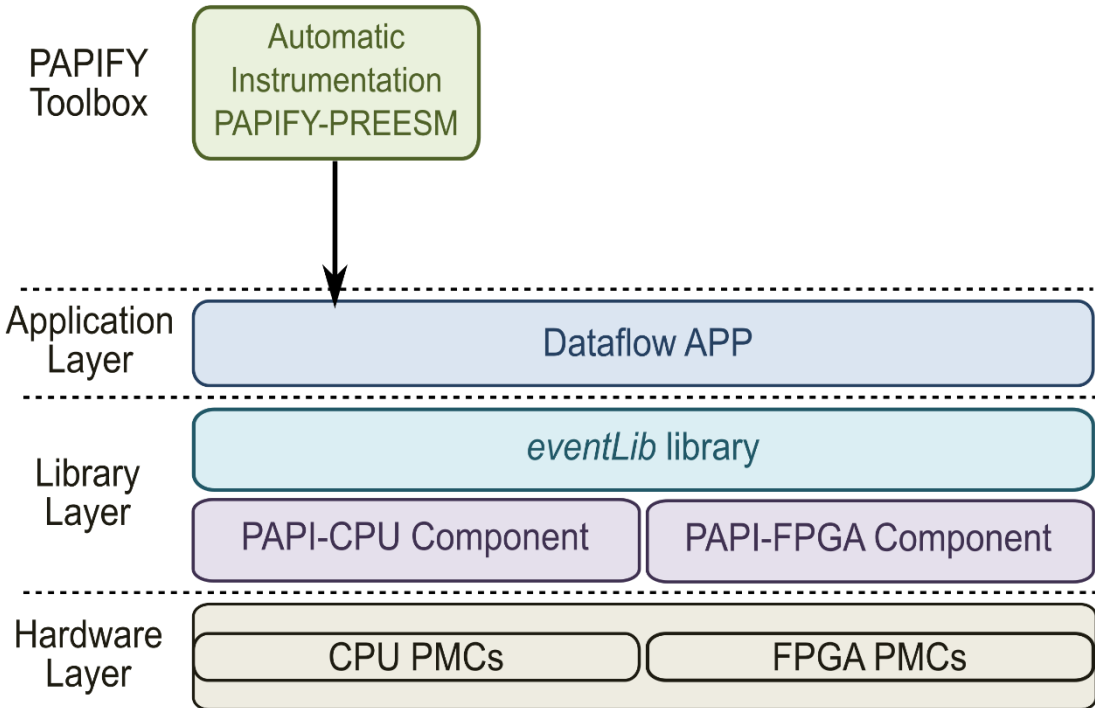
CERBERO Monitoring: PAPIFY



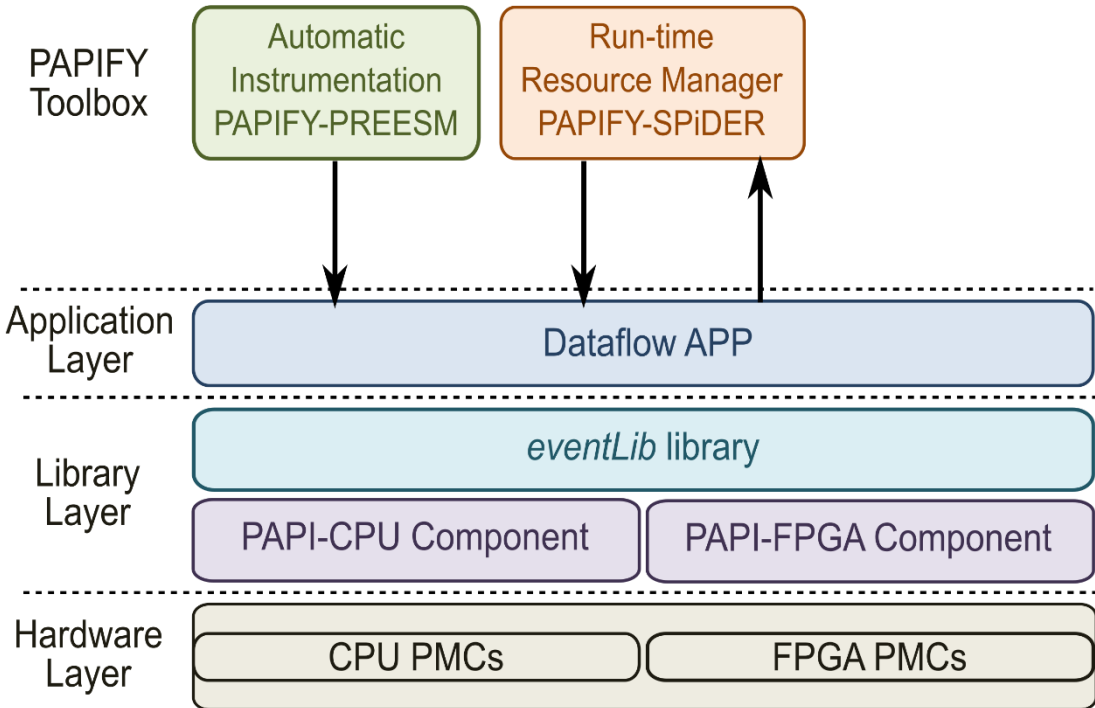
CERBERO Monitoring: PAPIFY



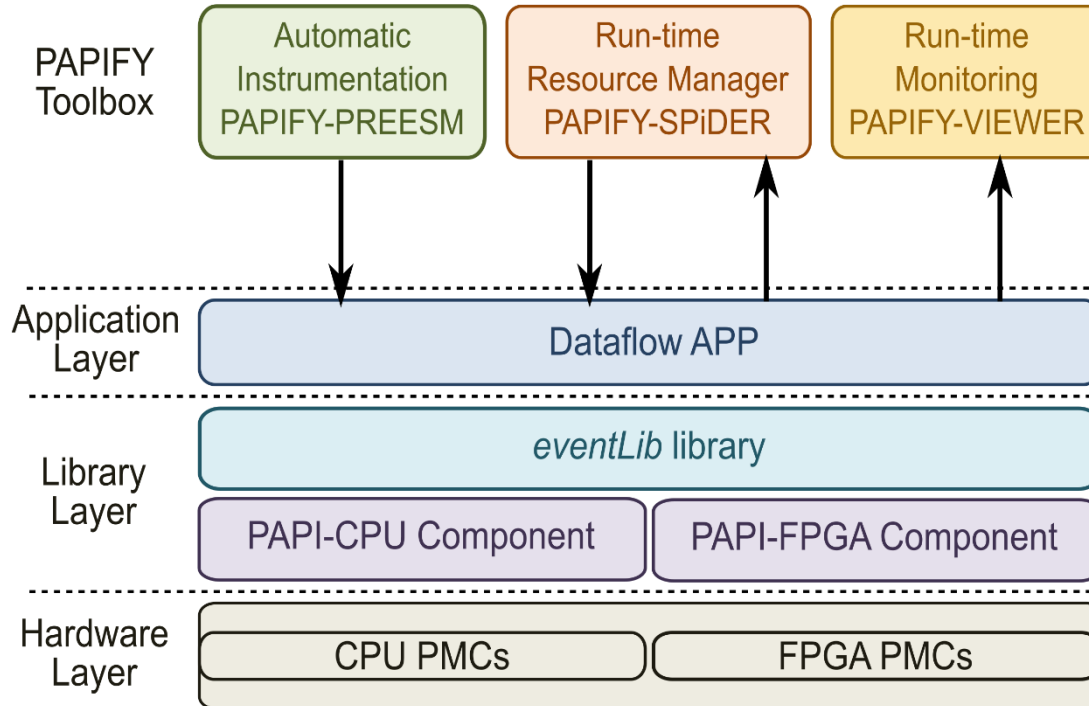
CERBERO Monitoring: PAPIFY



CERBERO Monitoring: PAPIFY



CERBERO Monitoring: PAPIFY



CERBERO Monitoring: PAPIFY

```
configure_papify_actor(...);    // Configure monitoring of each actor
configure_papify_PE(...);      // Configure each PE
```



CERBERO Monitoring: PAPIFY

```
configure_papify_actor(...); // Configure monitoring of each actor
configure_papify_PE(...); // Configure each PE
...
...

actor_execution(); // Execute the actor
```



CERBERO Monitoring: PAPIFY

```
configure_papify_actor(...); // Configure monitoring of each actor
configure_papify_PE(...); // Configure each PE
...
...
event_start(...); // Start monitoring PAPI events
event_start_papify_timing(...); // Start monitoring timing

actor_execution(); // Execute the actor

event_events(...); // Stop monitoring PAPI events
event_stop_papify_timing(...); // Stop monitoring timing
event_write_file(...); // Dump monitoring info
```



HiPEAC 2020

20-22 January, 2020, Bologna, Italy

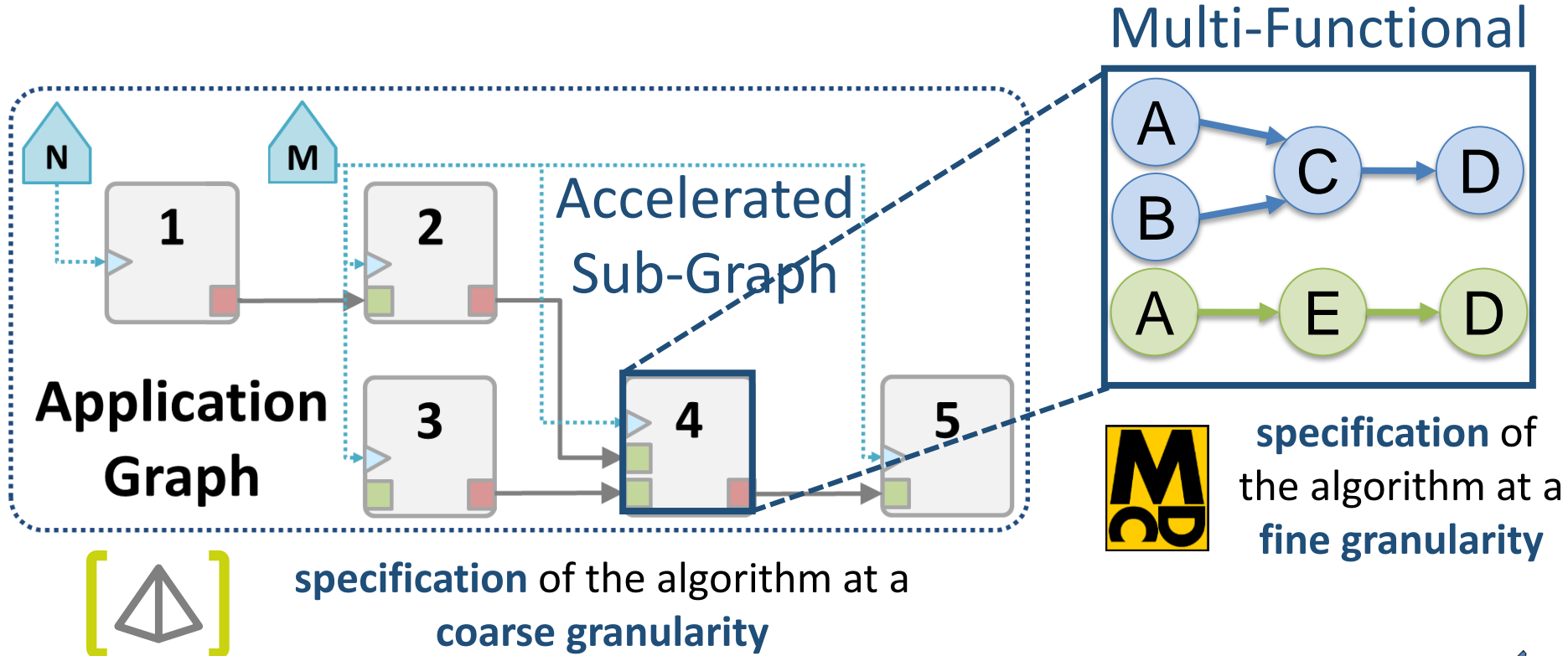


Creating Hardware Accelerators with DPR and CGR



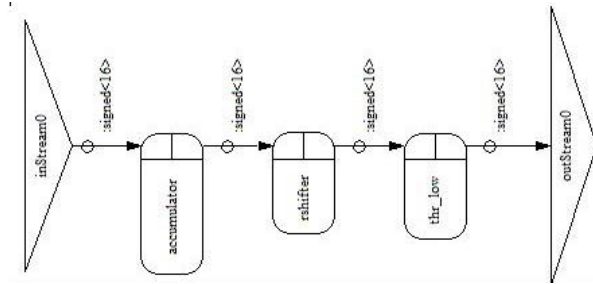
Horizon 2020
European Union funding
for Research & Innovation

MDC Suite



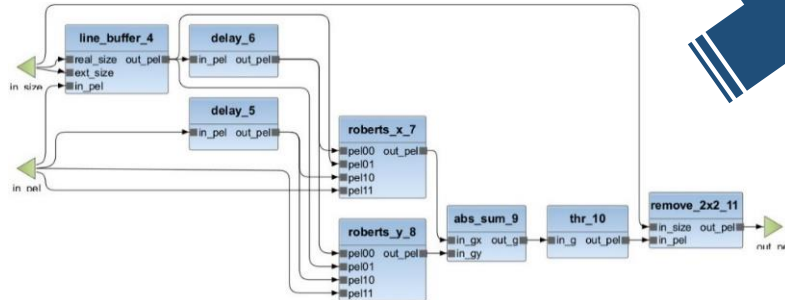
MDC Suite

CAPH



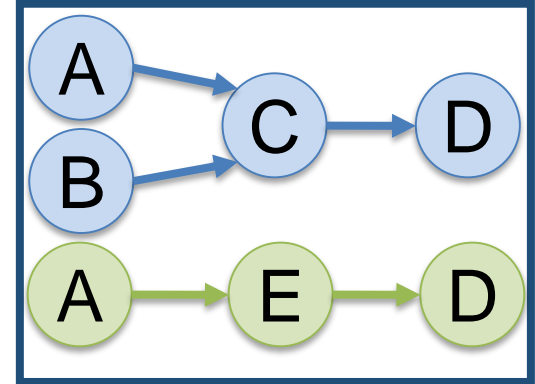
fully integrated flow, from **dataflow** to **hardware**

ORCC + Vivado HLS



two steps flow: **ORCC** for dataflow **network**, **Vivado HLS** deals for **actors** hardware code generation

Multi-Functional



specification of the algorithm at a **fine granularity**



MDC Suite



**Multi Dataflow
Composer Tool**

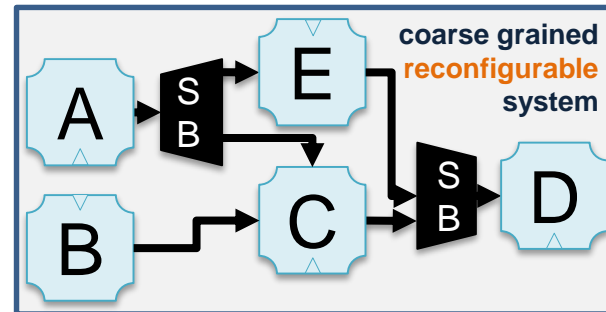
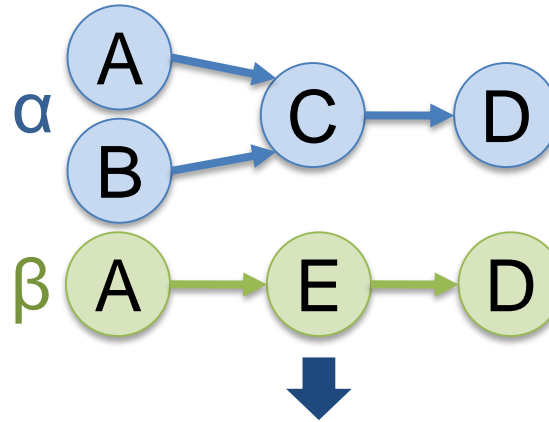
Structural Profiler

Power Manager

**System
Generator**

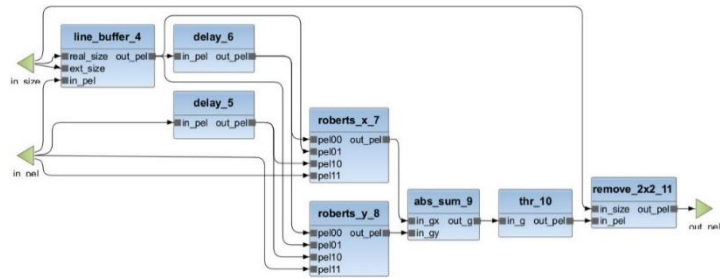
MDC design suite
<http://sites.unica.it/rpct/>

dataflow
specifications

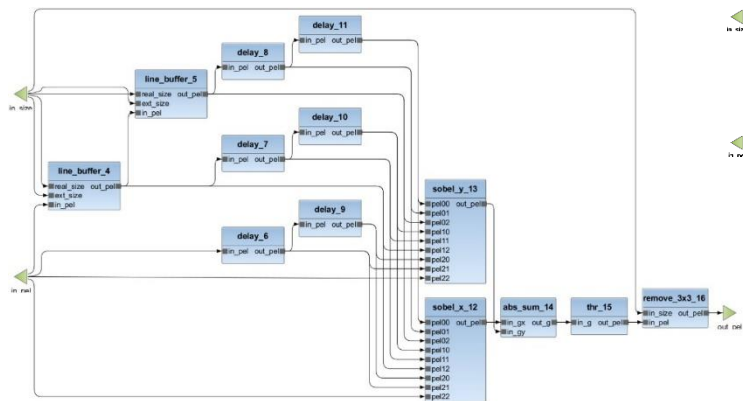


MDC: Merging

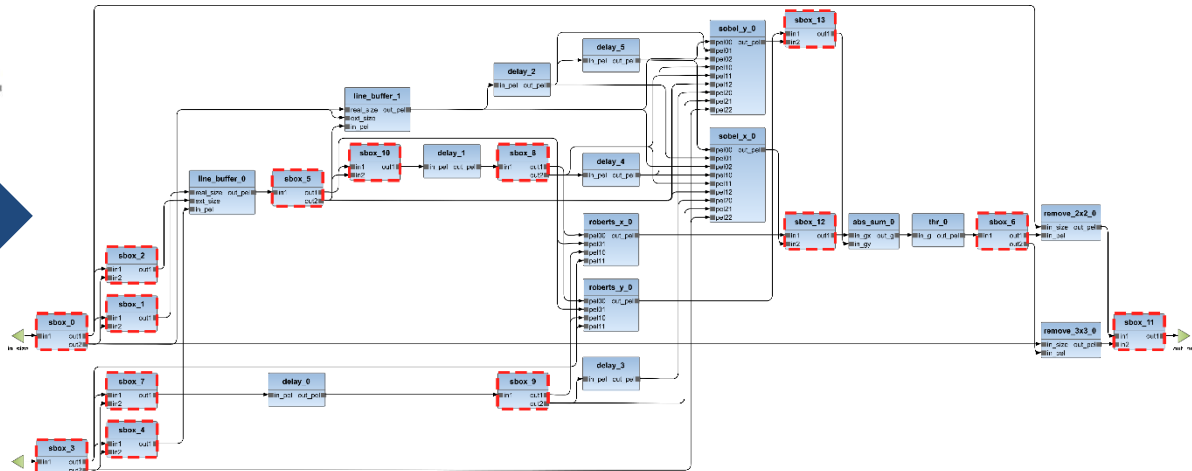
Roberts network



Sobel network



Multi-dataflow network



MDC Suite



**Multi Dataflow
Composer Tool**

Structural Profiler

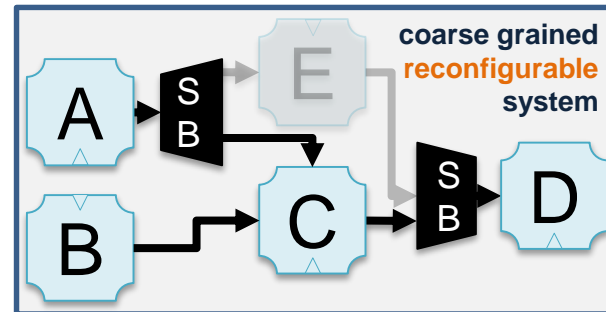
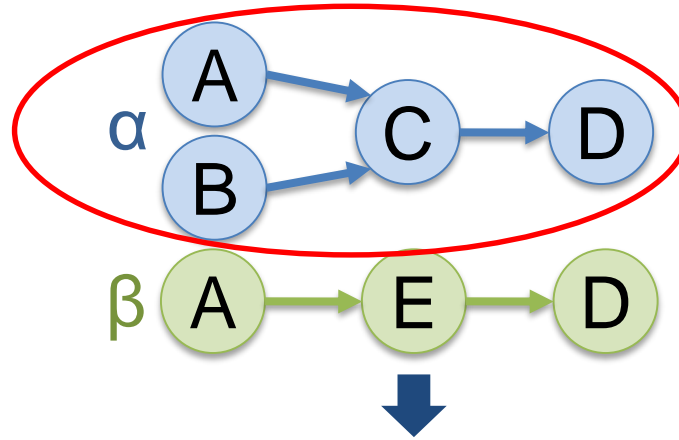
Power Manager

**System
Generator**

MDC design suite

<http://sites.unica.it/rpct/>

dataflow
specifications



MDC Suite



**Multi Dataflow
Composer Tool**

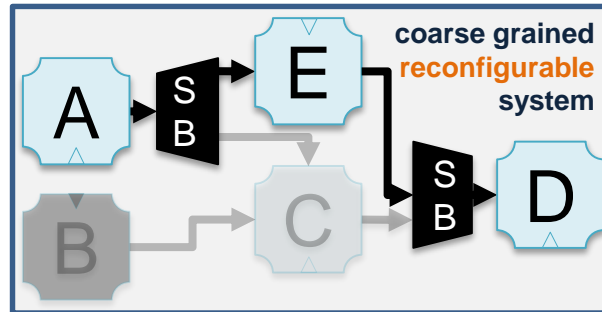
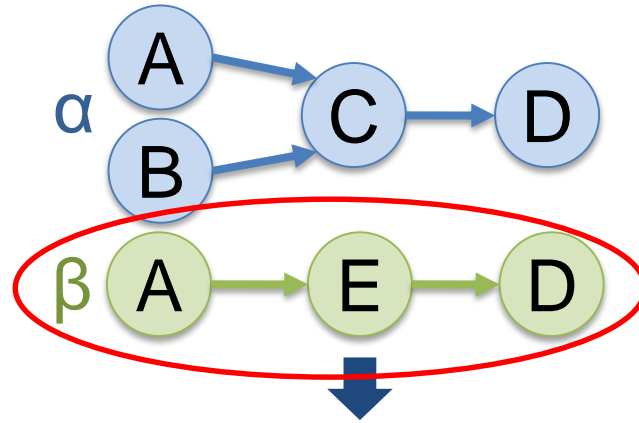
Structural Profiler

Power Manager

*System
Generator*

MDC design suite
<http://sites.unica.it/rpct/>

dataflow
specifications



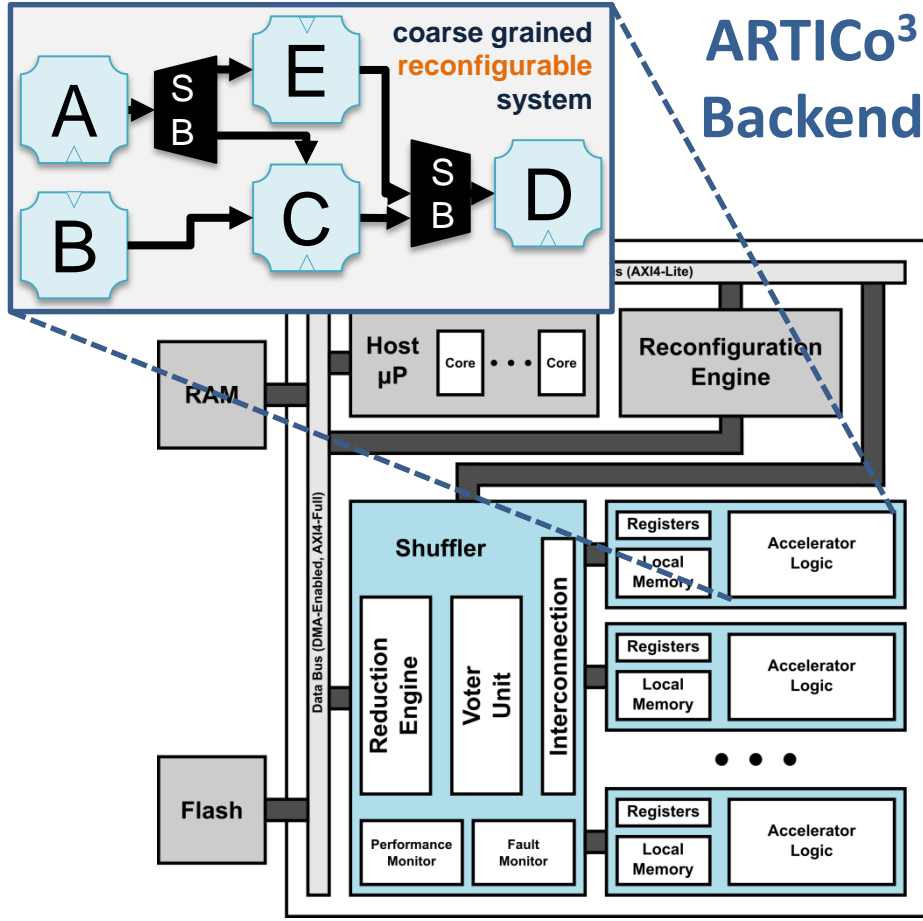
MDC Suite



Multi Dataflow
Composer Tool
Structural Profiler

Power Manager

System
Generator

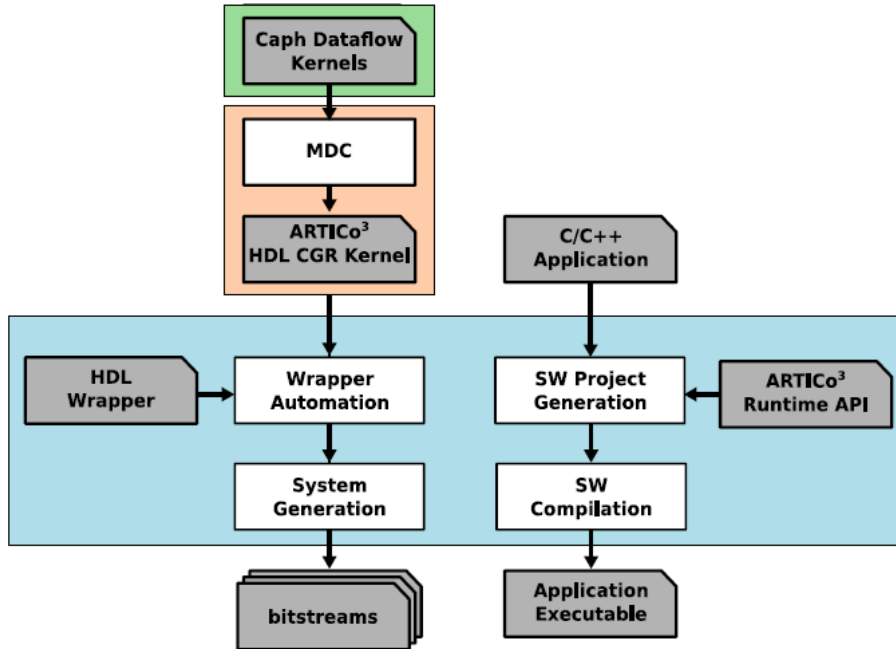


ARTICo³
Backend

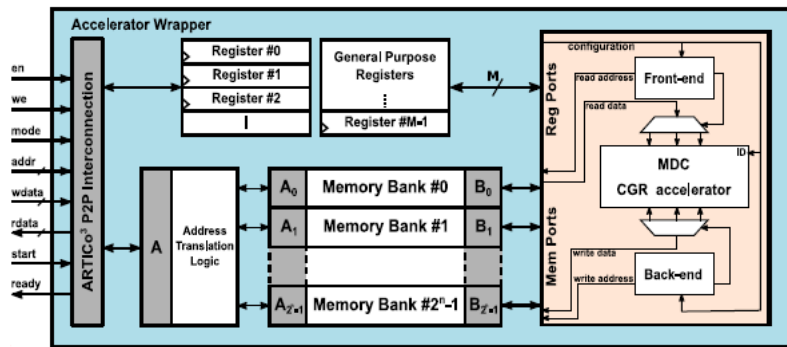
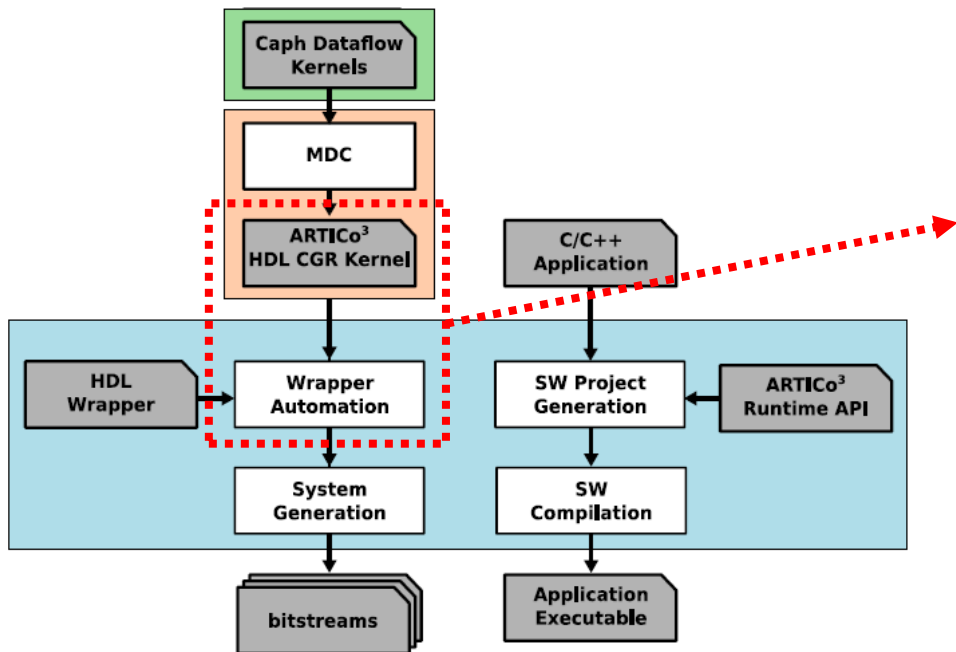
MDC design suite
<http://sites.unica.it/rpct/>



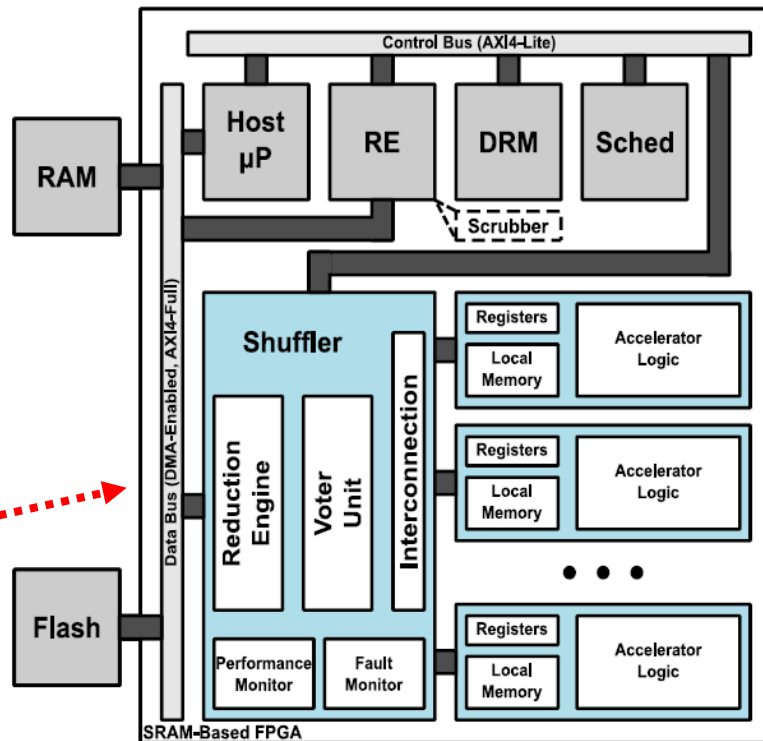
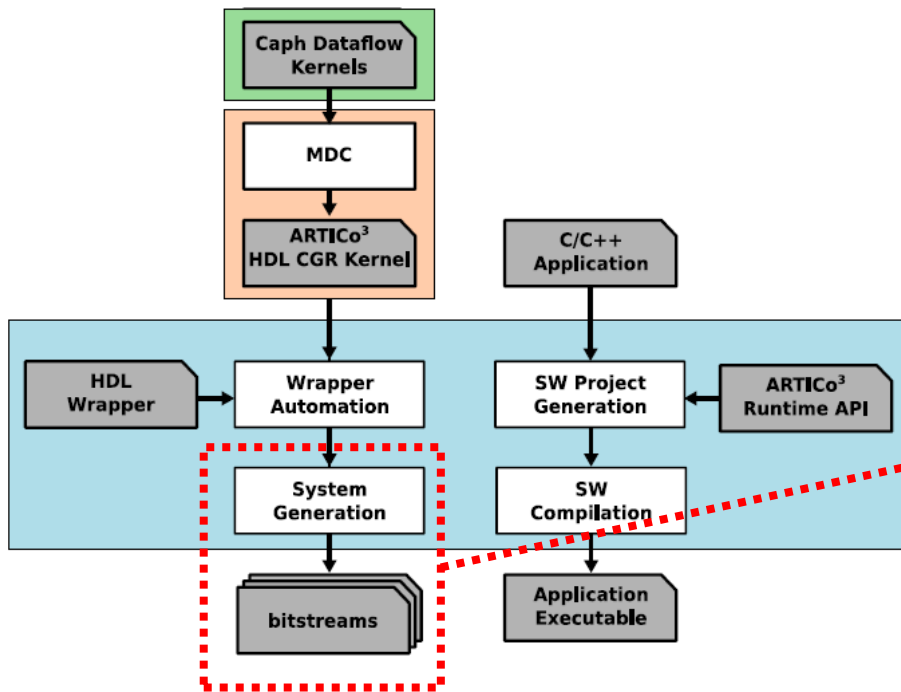
ARTICo³ Design Flow



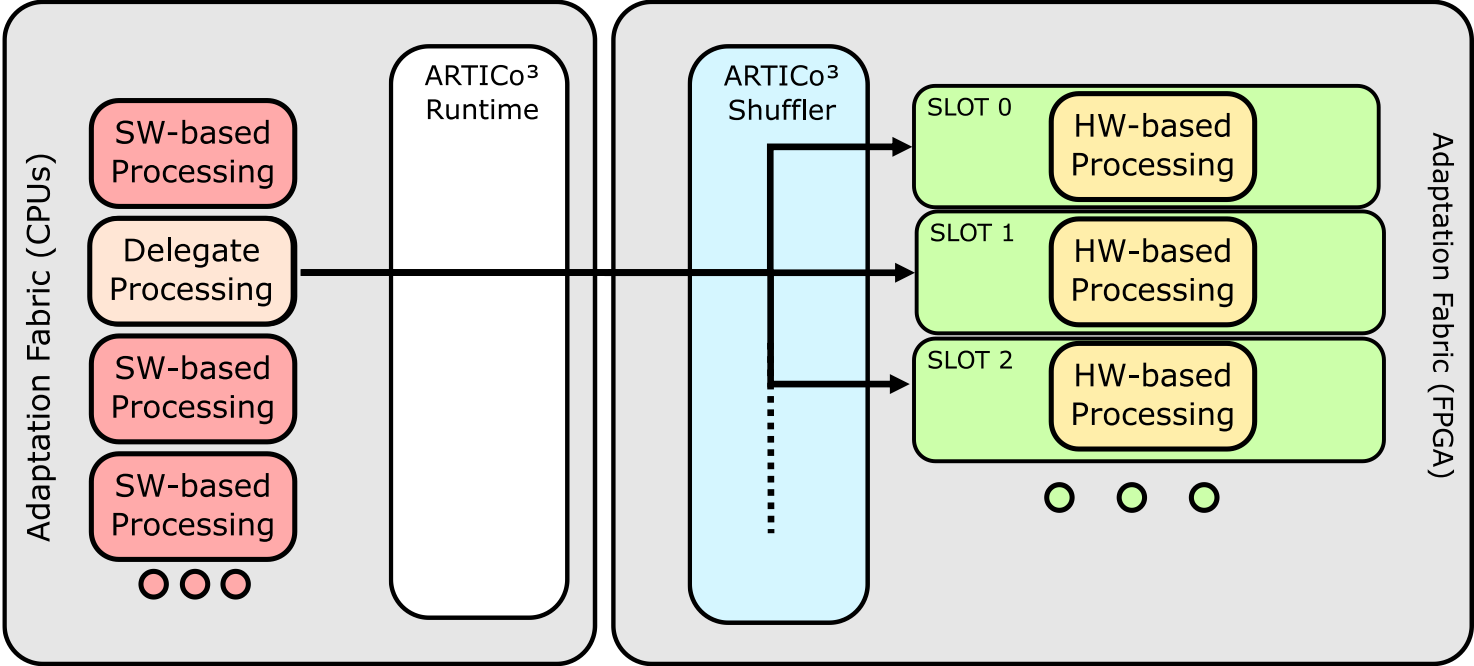
ARTICo³ Design Flow



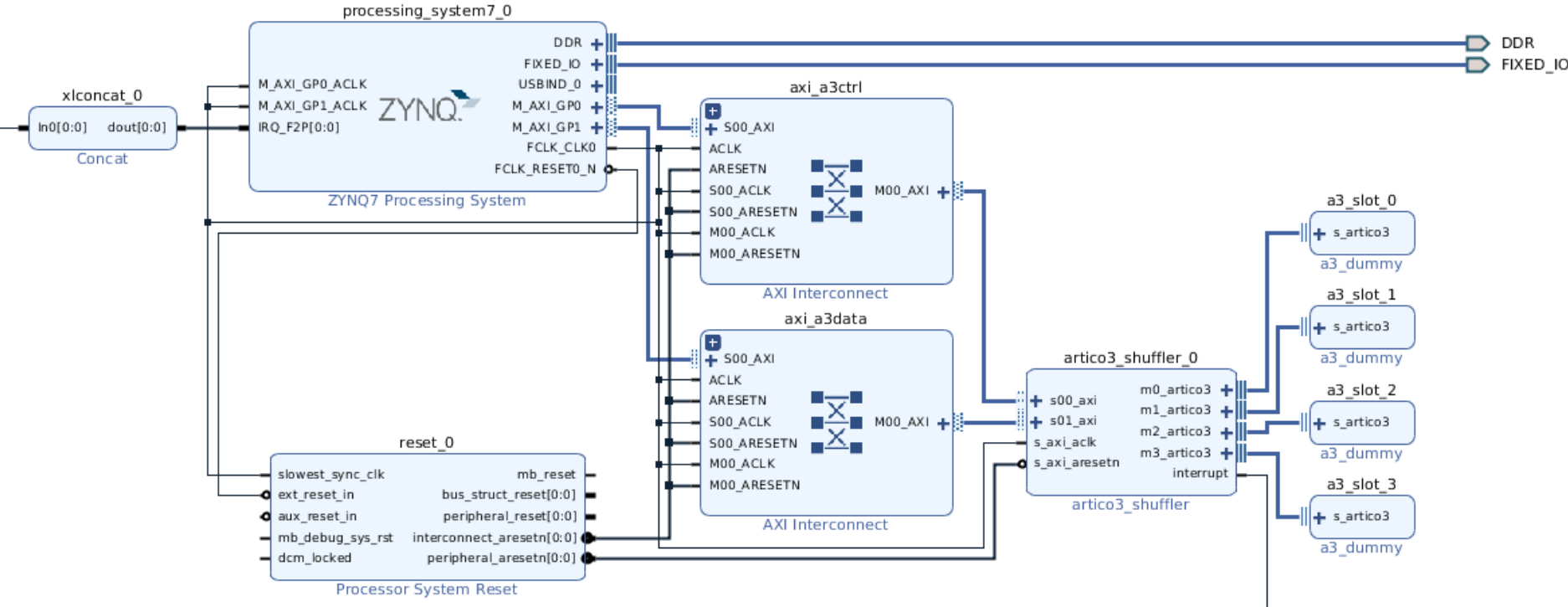
ARTICo³ Design Flow



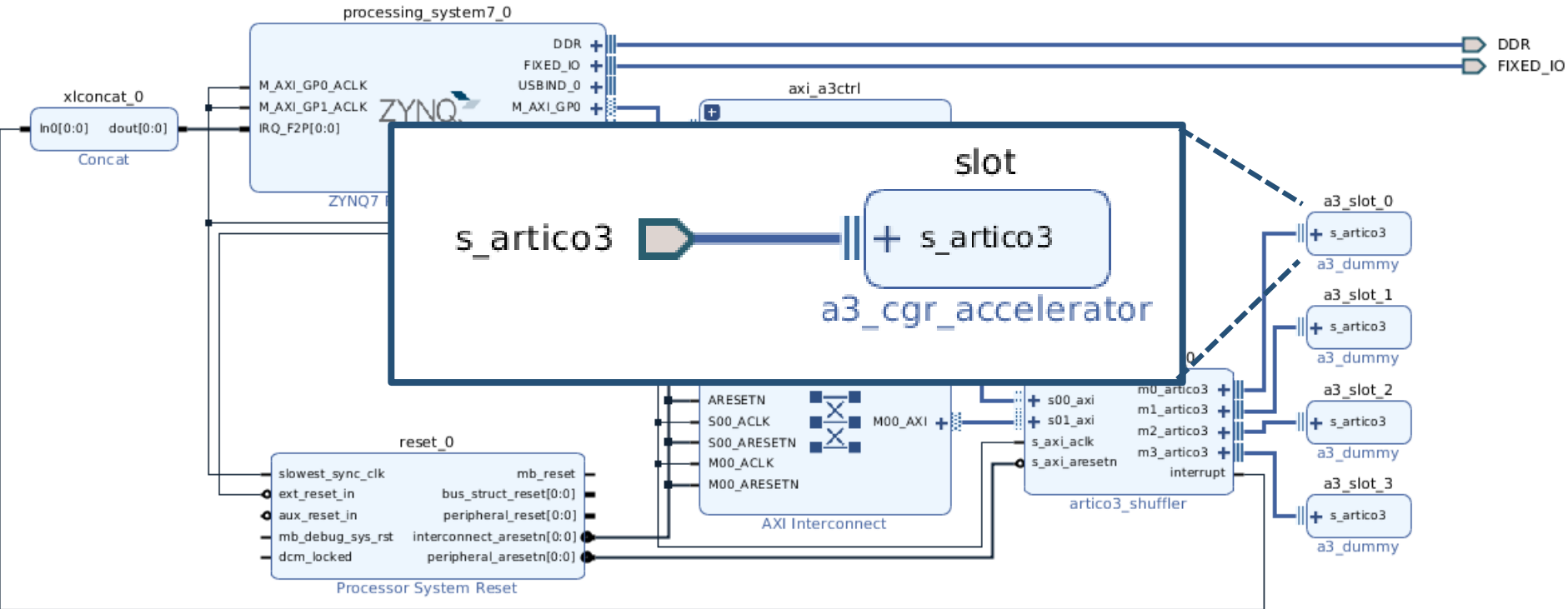
Heterogeneous Device



Architecture Overview in Vivado



Architecture Overview in Vivado



HiPEAC 2020

20-22 January, 2020, Bologna, Italy



CERBERO H2020 Project – Tutorial: Design Flow for Heterogeneous Embedded Computing Infrastructures



Horizon 2020
European Union funding
for Research & Innovation

Additional Slides

