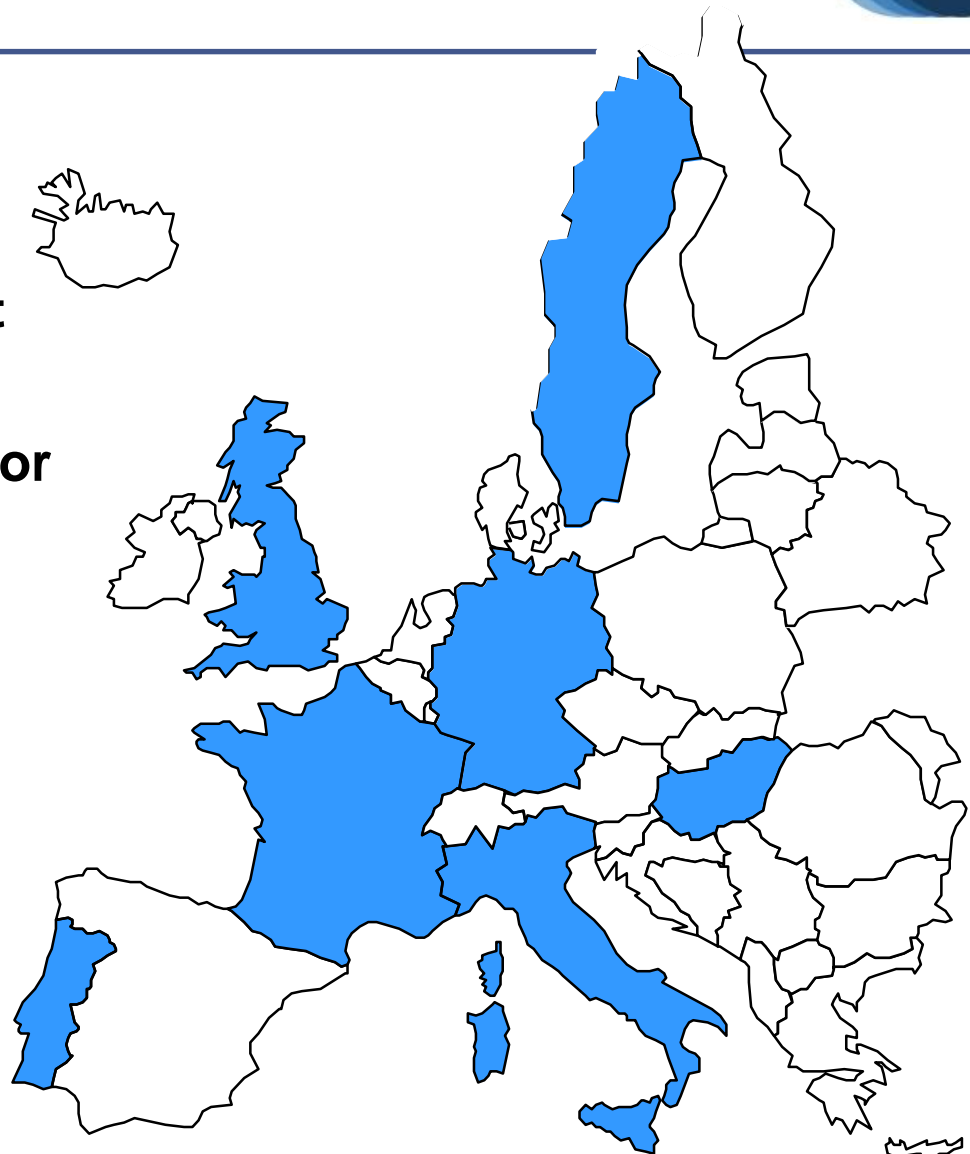# Fitting the CONCERTO component model approach to AUTOSAR development flow

Stefano Puri
Intecs S.p.A.
Pisa, Italy
stefano.puri@intecs.it

- CONCERTO methodology and component model
- CONCERTO and the AUTOSAR process
- CONCERTO tool support for AUTOSAR

ARTEMIS JU project

Call 2012

Technical Coordinator

Intecs

Partners 15

Countries 8
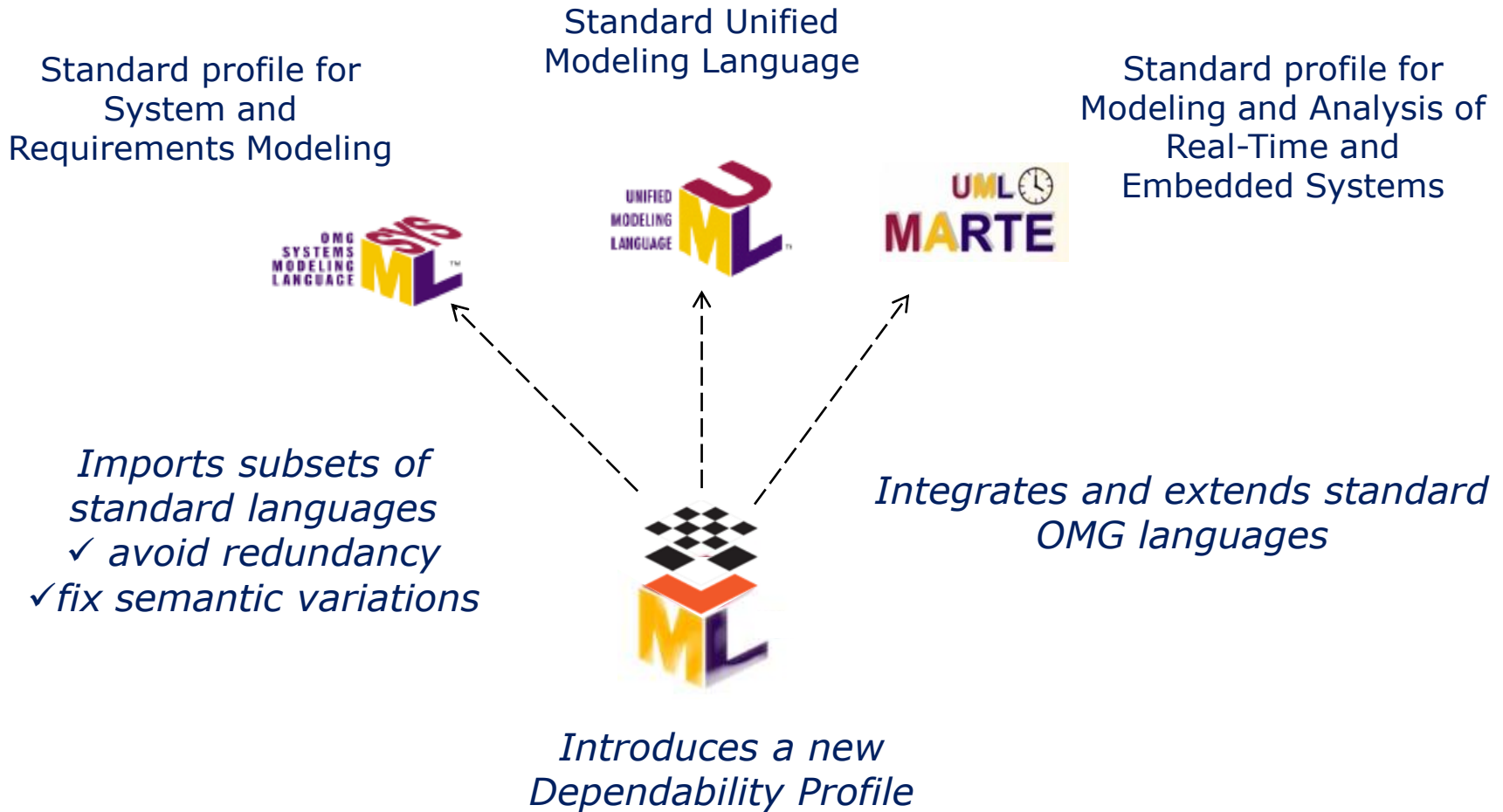
Start May 2013

End April 2016

Total cost 9,6 M €

- "Guaranteed Component Assembly with Round Trip Analysis for Energy Efficient High-integrity Multi-core Systems"- ARTEMIS JU Call 2012
  - Correctness-by-construction for multicore systems through model-driven engineering
  - Enhanced hierarchical, multi-domain component model
  - Support for separation of concerns into the multi-domain, multicore environment
  - Coverage of industrial domains
- Building on the results of the CHESS project (ARTEMIS-2008-1-100022)
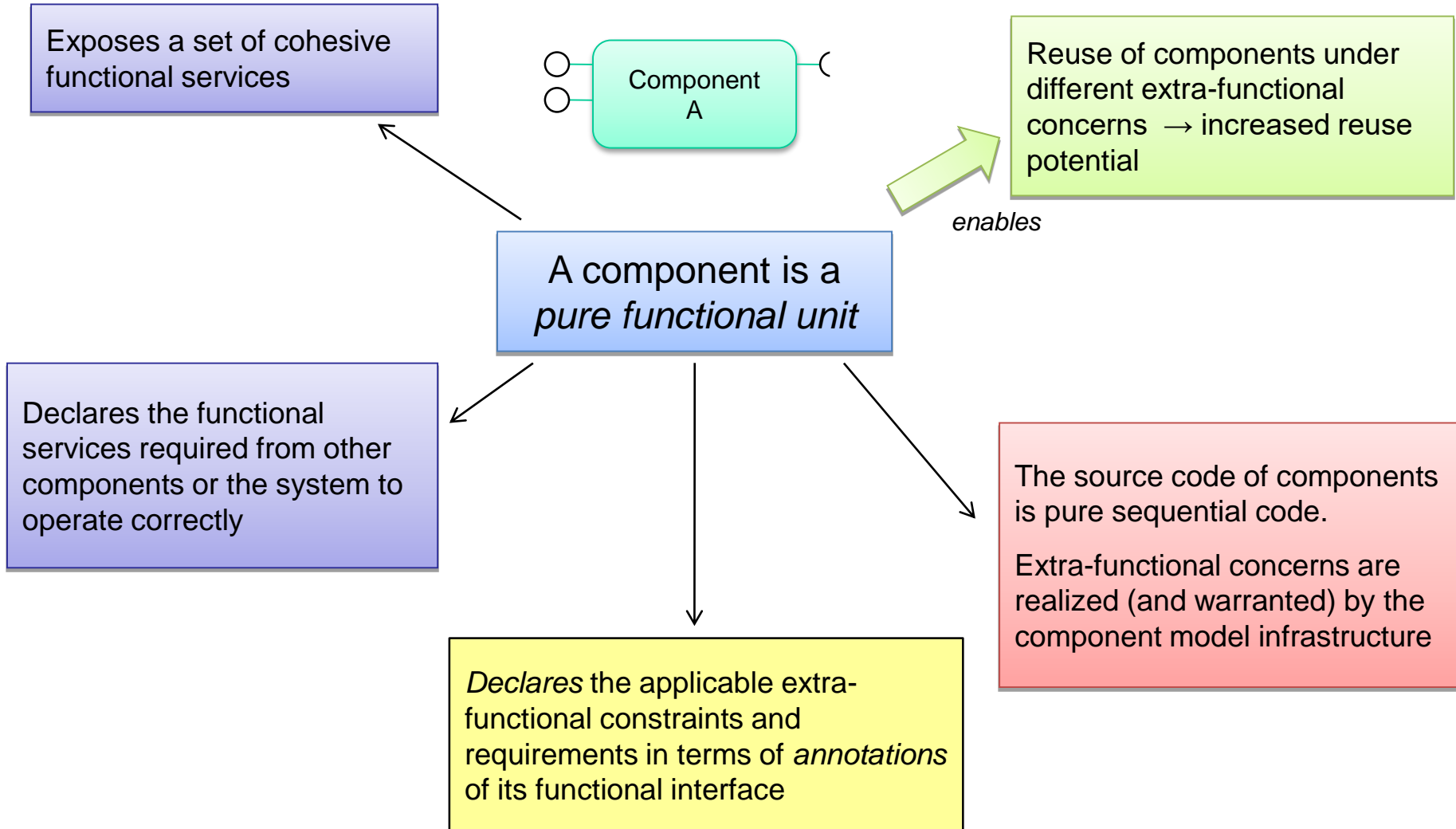 - CHESS is now a Project under the Eclipse Polarsys Working Group (https://www.polarsys.org/chess/)

Hereafter we will refer to CHESS as component model, methodology and tool, as used and extended by the CONCERTO project

CHESS goals

- ## Model-driven engineering
  - ### Models as the central development artifacts
    - CHESS Modellig Language
  - ### Tool assisted automated development
    - Based upon Eclipse, UMLPapyrus
- ## Component based development
  - ### Specialized to capture the extra-functional properties of components
    - Real Time
    - Dependability
- ## Multi-View / Separation of concerns
  - ### The architectural description of the system is organized into one or more views of the system where each view addresses one or more concerns of the system's stakeholders
    - ISO/IEC/IEEE42010:2011Systems and software engineering

The CHESS Modelling Language

Standard Unified
Modeling Language

Standard profile for
System and
Requirements Modeling

Standard profile for
Modeling and Analysis of
Real-Time and
Embedded Systems

*Imports subsets of
standard languages*
✓ *avoid redundancy*
✓ *fix semantic variations*

*Integrates and extends standard
OMG languages*

*Introduces a new
Dependability Profile*

The CHESS Component Model

Exposes a set of cohesive functional services

Component A

Reuse of components under different extra-functional concerns → increased reuse potential

enables

A component is a *pure functional unit*

Declares the functional services required from other components or the system to operate correctly

The source code of components is pure sequential code.

Extra-functional concerns are realized (and warranted) by the component model infrastructure

Declares the applicable extra-functional constraints and requirements in terms of *annotations* of its functional interface

CHESS Container and Connector

Container

Container A

-Wrapper responsible for the declared extra-functional attributes (e.g. the realization of tasking, timing behavior, fault containment and fault tolerance)

- Provides the component with a mediated connection with the execution platform and the system in general
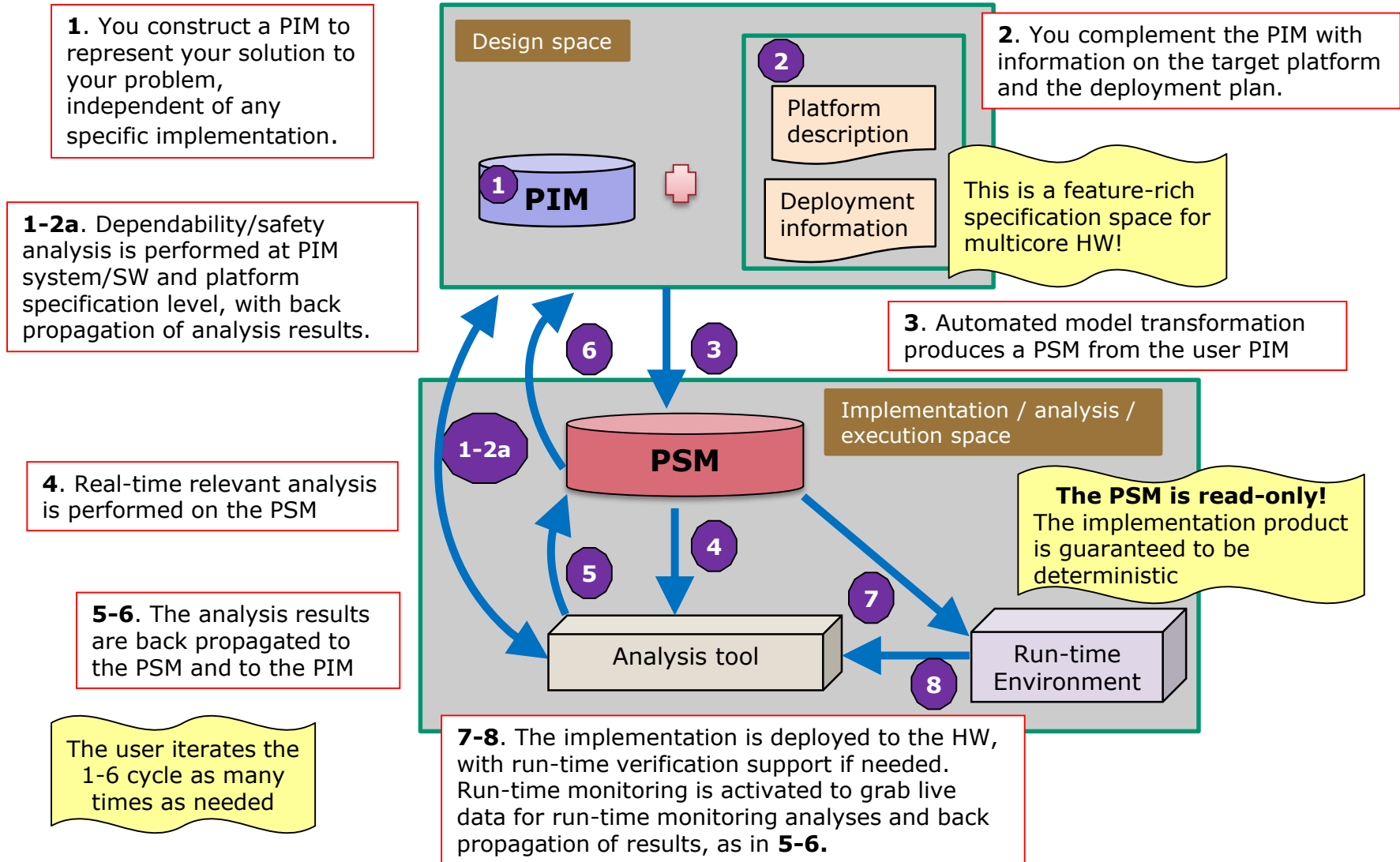
Connector

Component A

Container A

Connector AB

Component B

Container B
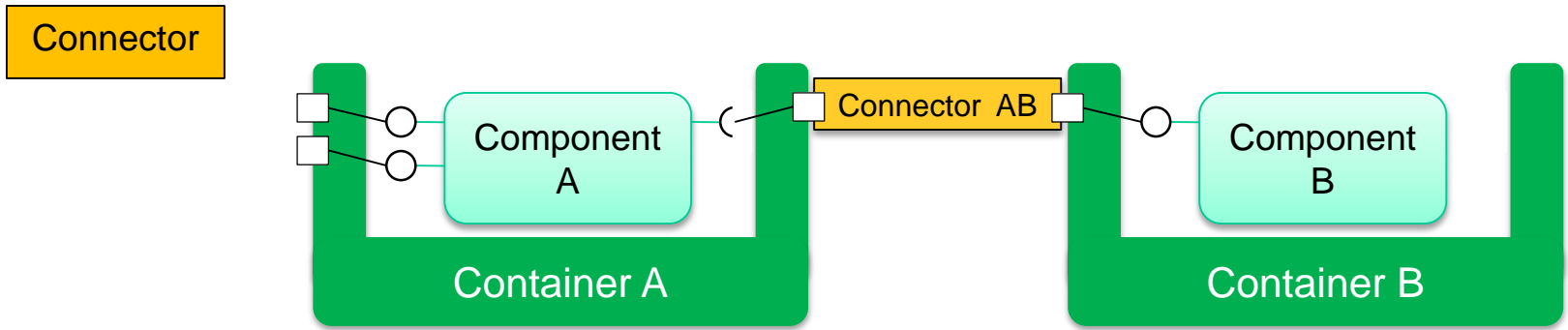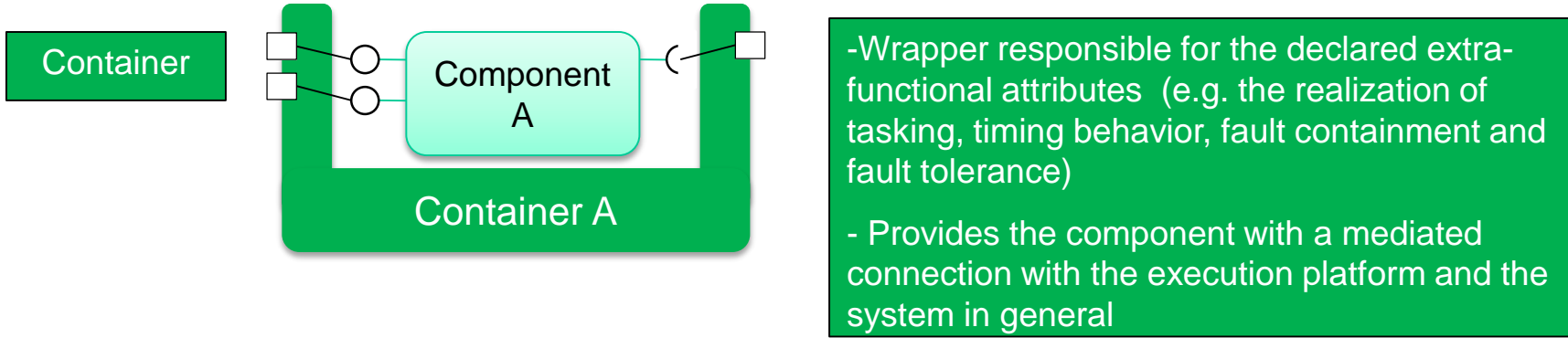
- Addresses interaction concerns

- Decouples the component from the other end-point(s) of a communication

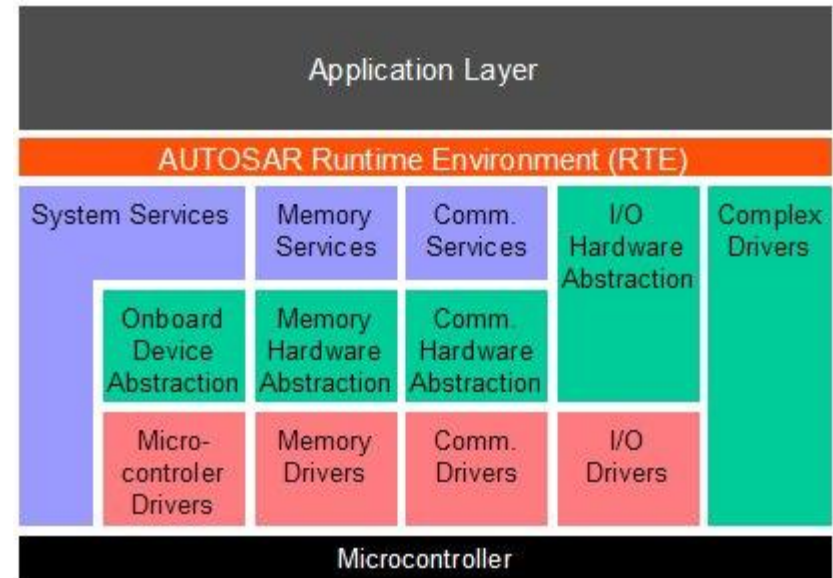- Realizes connection properties , e.g. procedure/function call, remote message passing …

CHESS Design Flow

1. You construct a PIM to represent your solution to your problem, independent of any specific implementation.

1-2a. Dependability/safety analysis is performed at PIM system/SW and platform specification level, with back propagation of analysis results.

4. Real-time relevant analysis is performed on the PSM

5-6. The analysis results are back propagated to the PSM and to the PIM

The user iterates the 1-6 cycle as many times as needed

Design space

Platform description

Deployment information

PIM

2. You complement the PIM with information on the target platform and the deployment plan.

This is a feature-rich specification space for multicore HW!

3. Automated model transformation produces a PSM from the user PIM

Implementation / analysis / execution space

PSM

The PSM is read-only! The implementation product is guaranteed to be deterministic

Analysis tool

Run-time Environment

7-8. The implementation is deployed to the HW, with run-time verification support if needed. Run-time monitoring is activated to grab live data for run-time monitoring analyses and back propagation of results, as in **5-6.**

- Cross domain
    - Scheduling and timing analysis for multicore
    - Dependability profile and analysis
        - Failure logic analysis, state based analysis
    - Run-time monitoring
    - Matlab/Simulink Synchronous Block Diagrams (SBDs) support
- Domain specific
    - Avionics
    - Telecare
    - Automotive…

AUT⊙SAR

- (AUTomotive Open System ARchitecture)
  - Open and standardized software architecture for automotive, jointly developed by automobile manufacturers, suppliers and tool developers
- Goals:
  - Scalability to different vehicle and platform variants
  - Integration of functional modules from multiple suppliers
  - Maintainability

Application Layer

AUTOSAR Runtime Environment (RTE)

System Services	Memory Services	Comm. Services	I/O Hardware Abstraction	Complex Drivers
	Onboard Device Abstraction	Memory Hardware Abstraction	Comm. Hardware Abstraction	
	Micro-controler Drivers	Memory Drivers	Comm. Drivers	I/O Drivers	

Microcontroller

Legend: Basic Software (BSW) Layers

| Services | ECU abstraction and Complex Drivers | Micro-controller Abstraction |

AUTOSAR methodology steps

- The AUTOSAR methodology comprises the following steps
  - Application Configuration
  - System Configuration
  - ECU specific information extraction
  - ECU Configuration
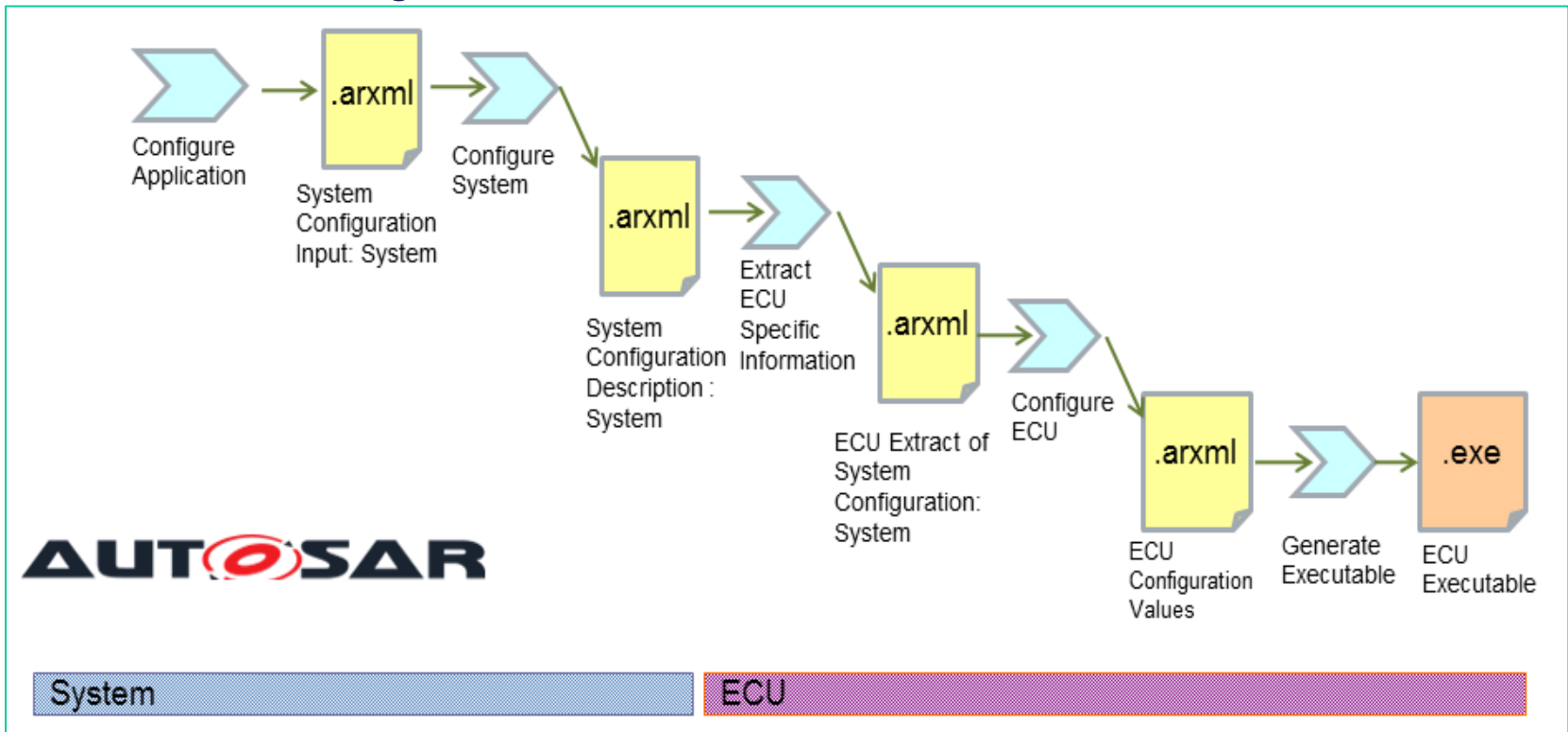
CHESS support for AUTOSAR steps

- The CHESS methodology supports the following AUTOSAR steps
  - Application Configuration
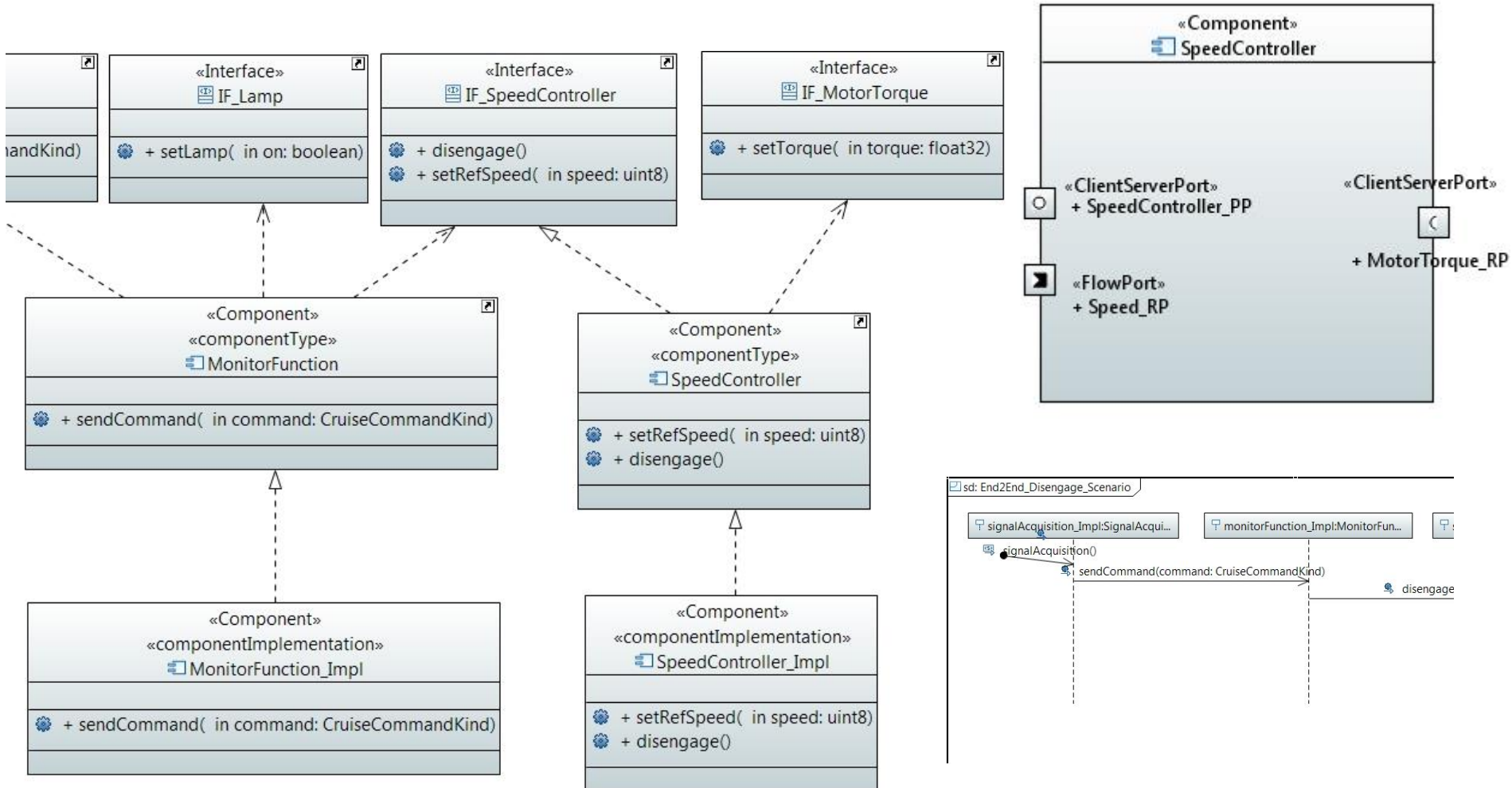 - System Configuration
  - ECU Configuration
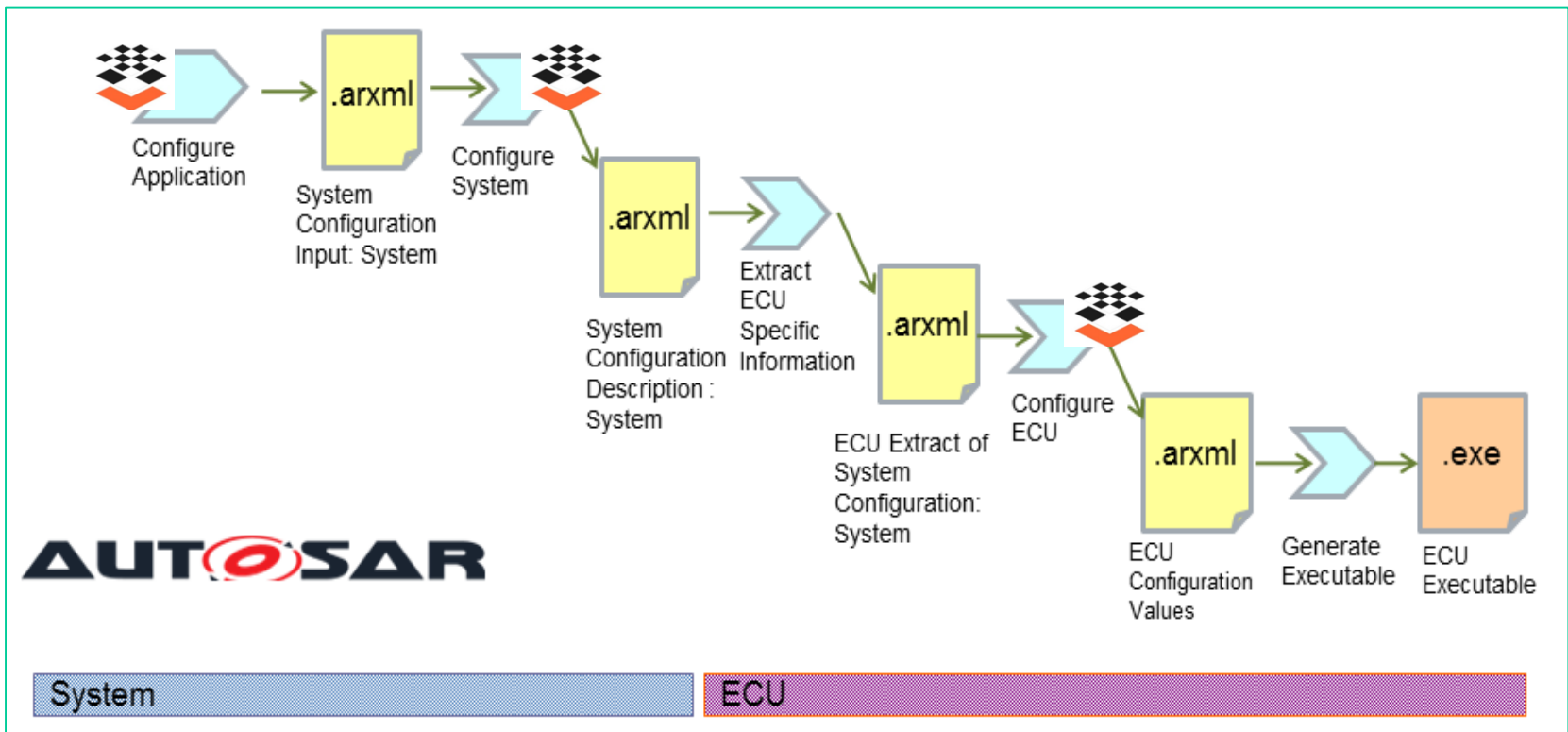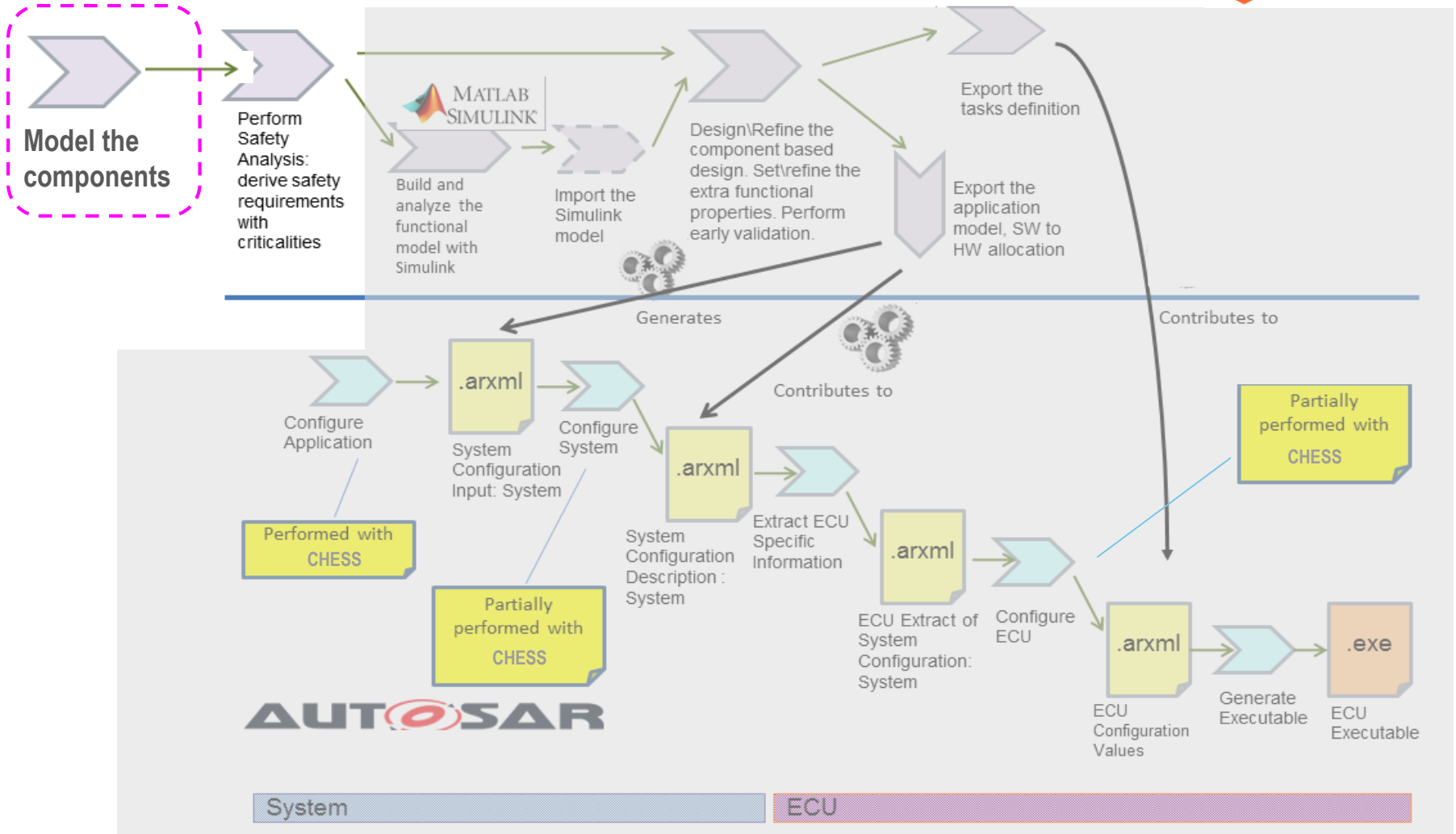
- Modelling Interfaces, Component Types and Component Implementations

- CHESS and AUTOSAR share the definition of software component based design
- Similar fundamentals concepts are available:
  - hierarchical structure
  - ports and interfaces
  - connectors
  - internal behaviour
- Extra concepts availalble in AUTOSAR (more on this later in the presentation)

Modelling extra-functional properties with CHESS

Extra-functional attributes

SendCommand must be executed **sporadically**.

MIAT=80ms

Deadline = 80ms

WCET=25ms
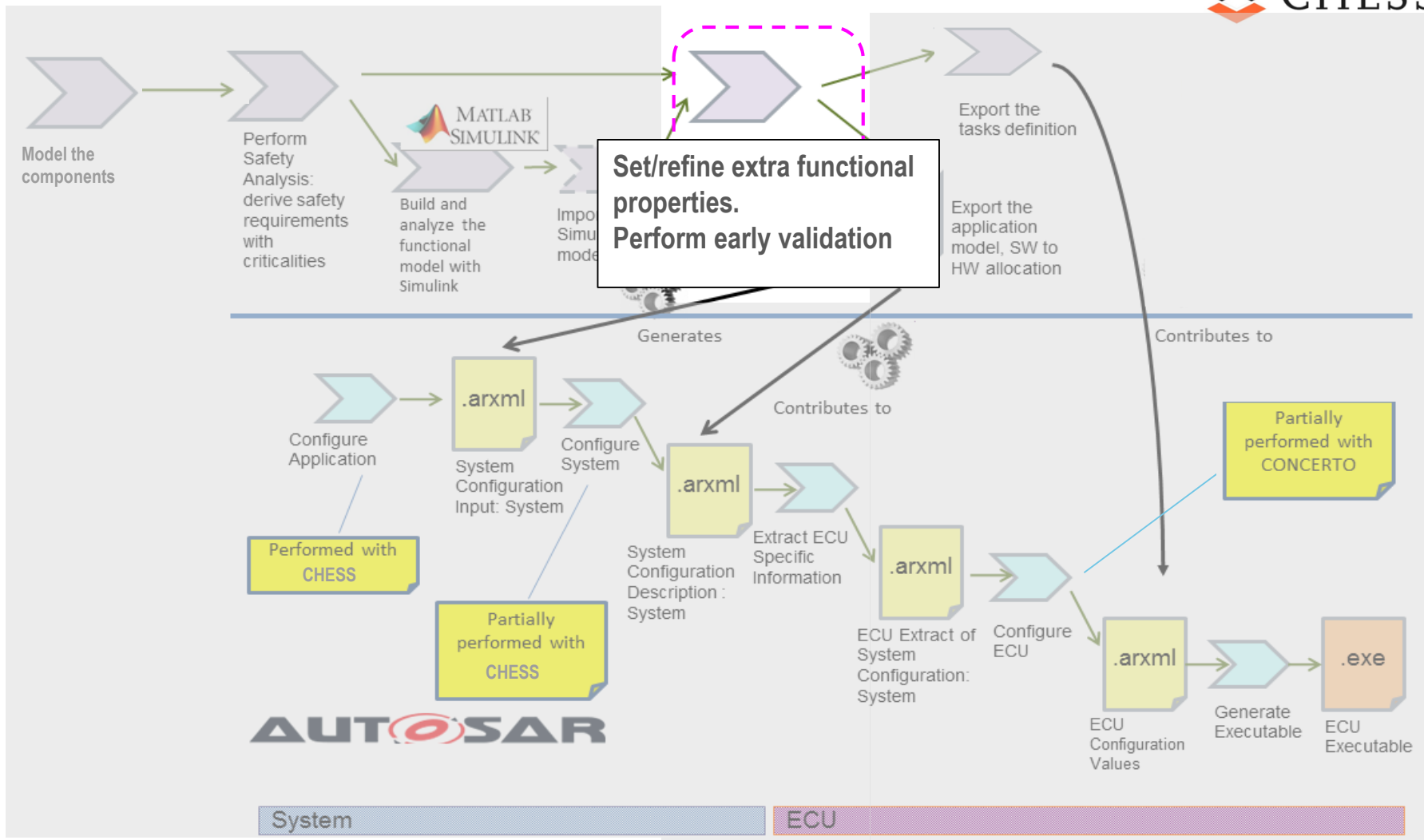
- The CHESS component model allows to perform timing analysis on the model taking into account the timing properties that decorate the components interfaces
  - An extended version of the MAST tool (http://mast.unican.es/) is used for timing analysis execution
- This way the modeller can perform early validation on the model and adjust the model until analysis results are satisfactory

Performing timing analysis with CHESS (2/2)

- Once the extra-functional properties are modelled and the analysis context is set, the CHESS user can perform timing analysis to verify if the system is schedulable
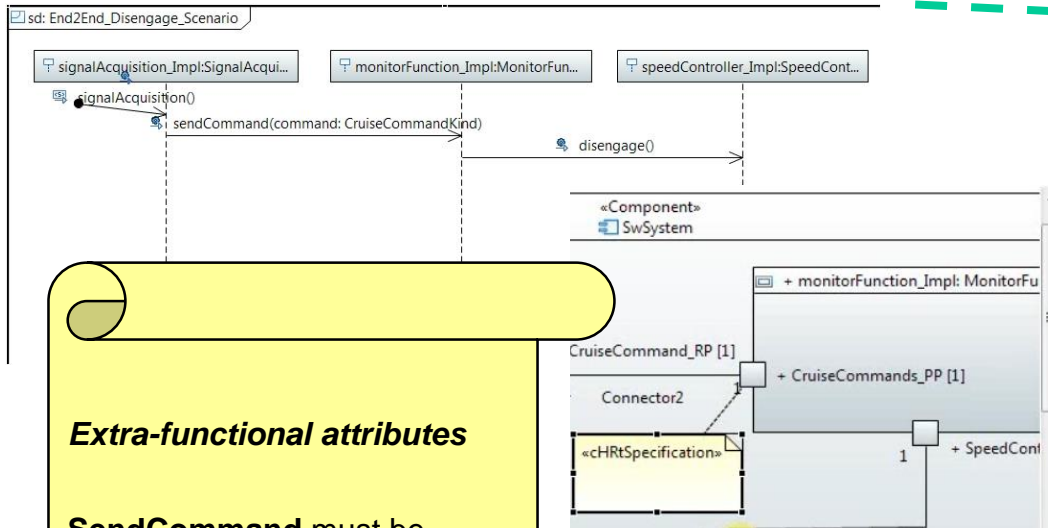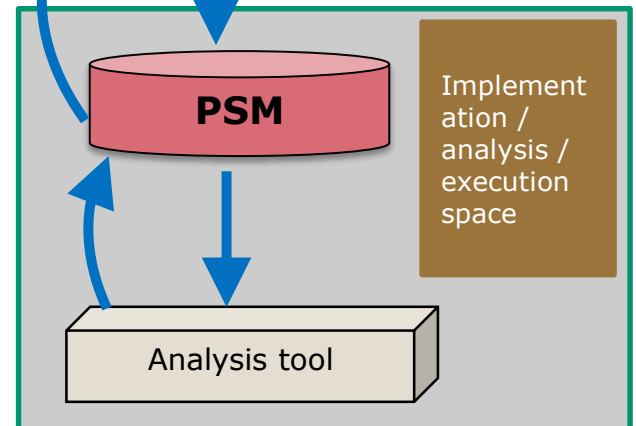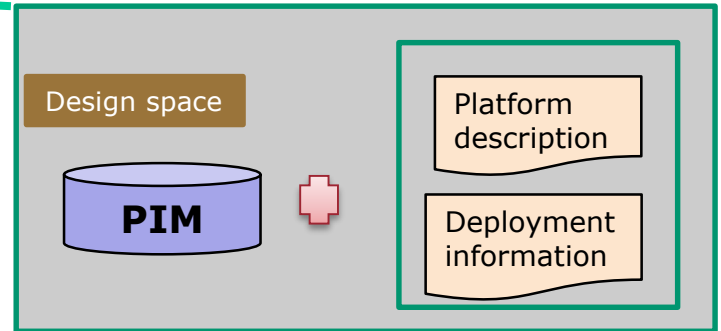

Extra-functional attributes

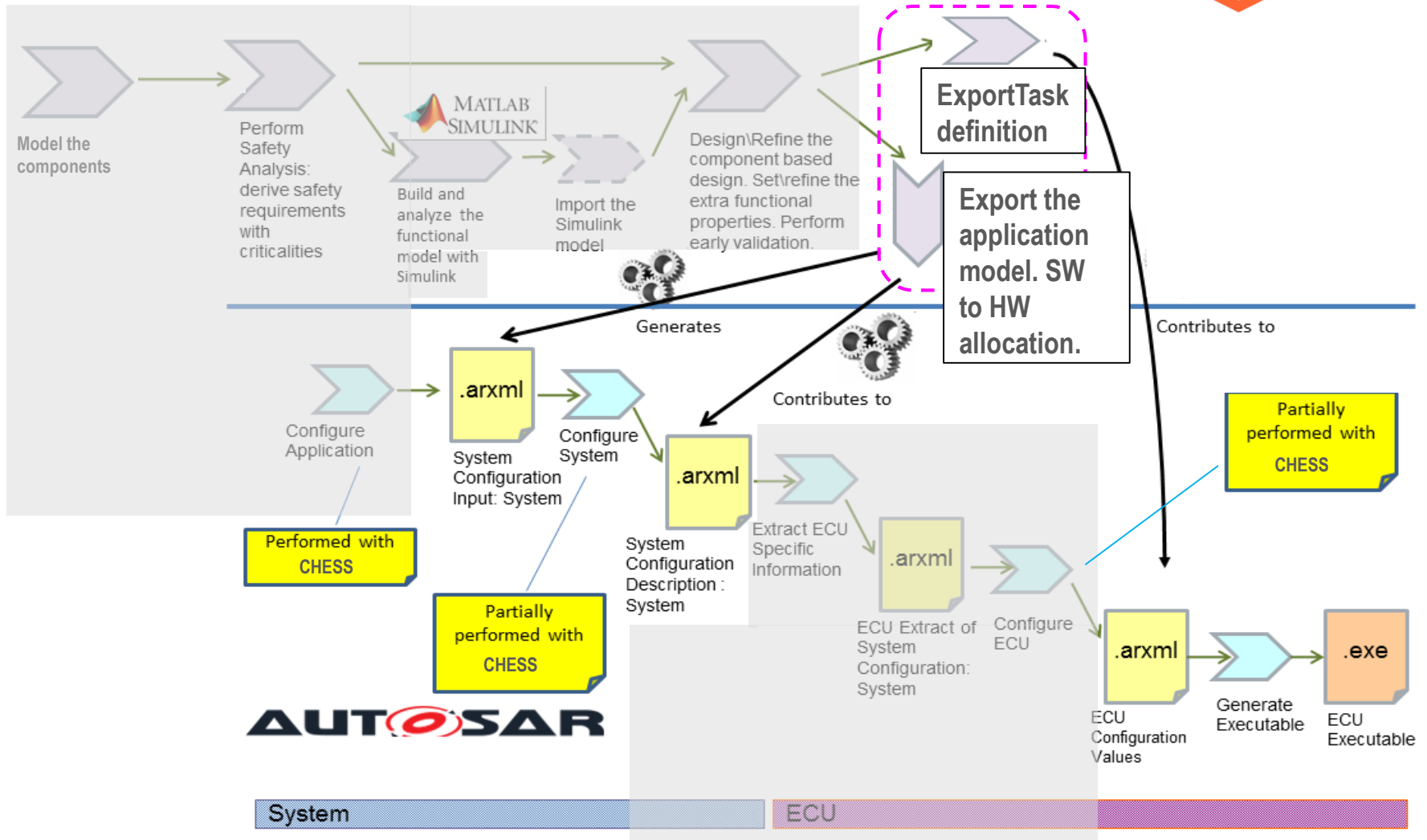**SendCommand** must be executed *sporadically*.

MIAT=80ms

WCET=25ms

Deadline = 80ms

Response Time = 70ms

- The CHESS methodology and toolset allow to sufficiently cover the initial steps of the AUTOSAR methodology

- The automatic generation of the AUTOSAR model starting from the CHESS one allows easy integration with external AUTOSAR tools

- The extraction of the information about the tasks has to be taken into account to make the added value of CHESS with respect to the AUTOSAR process and commercial tools stronger

- The AUTOSAR development can benefit from the analysis features of extra-functional properties provided by CHESS

- Evaluate CHESS component model extensions
  - Inter-runnable variable: AUTOSAR allows to model variables which are shared by runnables (i.e. functional code) defined in the same component. In particular AUTOSAR allows specifying how runnables can access this shared state such that safe (in the multithreading sense) exchange of data is possible.
    - In CHESS access to the shared data could be realized through get/set provided operations, decorated with protected or unprotected access
  - CHESS does not allow having data flow between components realized through flow ports; data flow ports are allowed for a given component only to represent the sending/receiving of events to/from the run-time environment. This is however a minor point, given that inter-components data flow can be realized through operation calls
- Improve modelling support for AUTOSAR RTE
  - E.g. to consider the impact of the basic software on the processor utilization and the system scheduling

Thank you for your attention! Questions?