



Architecture, Modelling, and Validation of Safety-Critical Systems – the DECOS Approach

Wolfgang Herzner
ARC Seibersdorf research GmbH
Smart Systems Division
wolfgang.herzner@arcs.ac.at

DECOS

Dependable Embedded Components and Systems

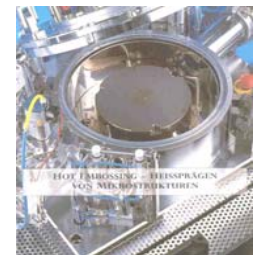
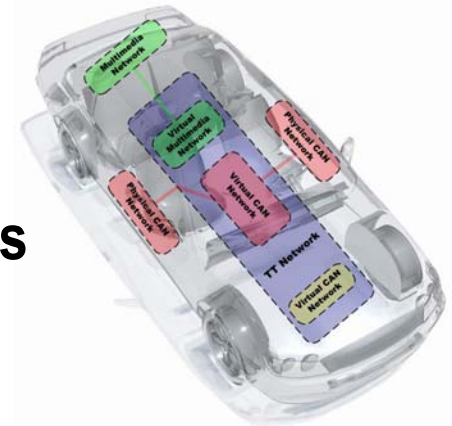
(IP-Project #511764 in EU FP6 / Priority [2] IST)

- **Co-ordination**
 - ◆ ARC Seibersdorf research
- **Partner (19)**
 - ◆ **Industry:** Airbus, AEV, EADS, Infineon, TTTech, Fiat, Profactor, Hella, Liebherr, Thales, Esterel
 - ◆ **Universities:** TU Vienna, TU Darmstadt, TU Hamburg, Uni Kassel, Uni Kiel, Budapest Uni of Techn. and Economics
 - ◆ **Research Centres:** ARCS, SP Swedish Test. & Res. Inst.
- **Budget:** 14.3 Mio €
- **EU Funding:** 9 Mio €
- **Duration:** 7/04 – 12/07



DECOS Goals

- Uniform platform for integration of embedded distributed (real-time) applications of mixed (up to highest) criticality
 - ◆ hardware reduction
 - ◆ flexibility increase
- ⇒ from **federated** to **integrated** systems
- Implication: fault-isolation of and non-interference between integrated systems has to be guaranteed
- ⇒ provision of appropriate
- ◆ architectures
 - ◆ components and services
 - ◆ development and verification tools



DECOS Subprojects

- ◆ **SP 1: Architecture Design (TU Darmstadt + TU Vienna)**
- ◆ **SP 2: Component Design and Implementation (TTTech)**
- ◆ **SP 3: Silicon Infrastructure (Infineon)**
- ◆ **SP 4: Validation and Certification (ARCS)**

- ◆ **SP 5: Application Automotive (Audi)**
- ◆ **SP 6: Application Aerospace (Airbus)**
- ◆ **SP 7: Application Control (Profactor)**

- ◆ **SP 8: Training, Dissemination and Standardization (ARCS)**
- ◆ **SP 9: IP Management and Assessment (ARCS)**

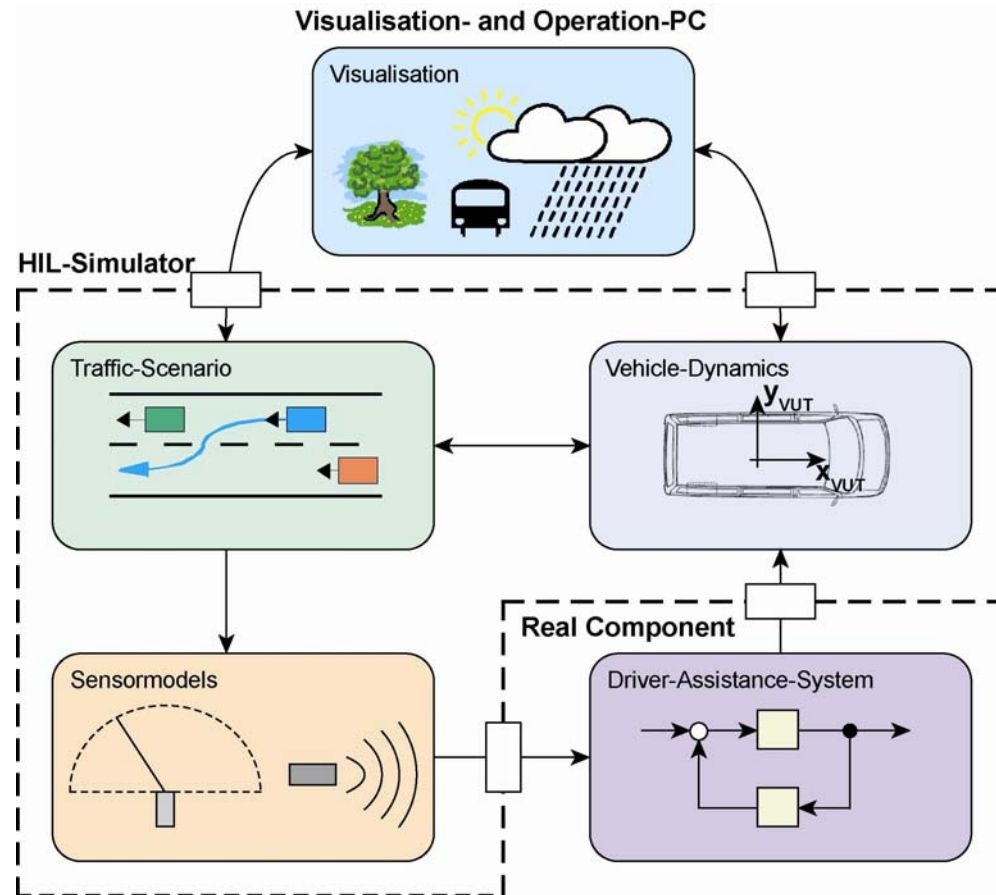
DECOS Application: Automotive

HIL- Demonstrator for Driver Assistance

- ◆ Simulation and visualisation of traffic environment
- ◆ Generation of traffic scenarios
- ◆ Generation of sensor data (for radar and laser)
- ◆ Modelling of vehicle and driver behaviour

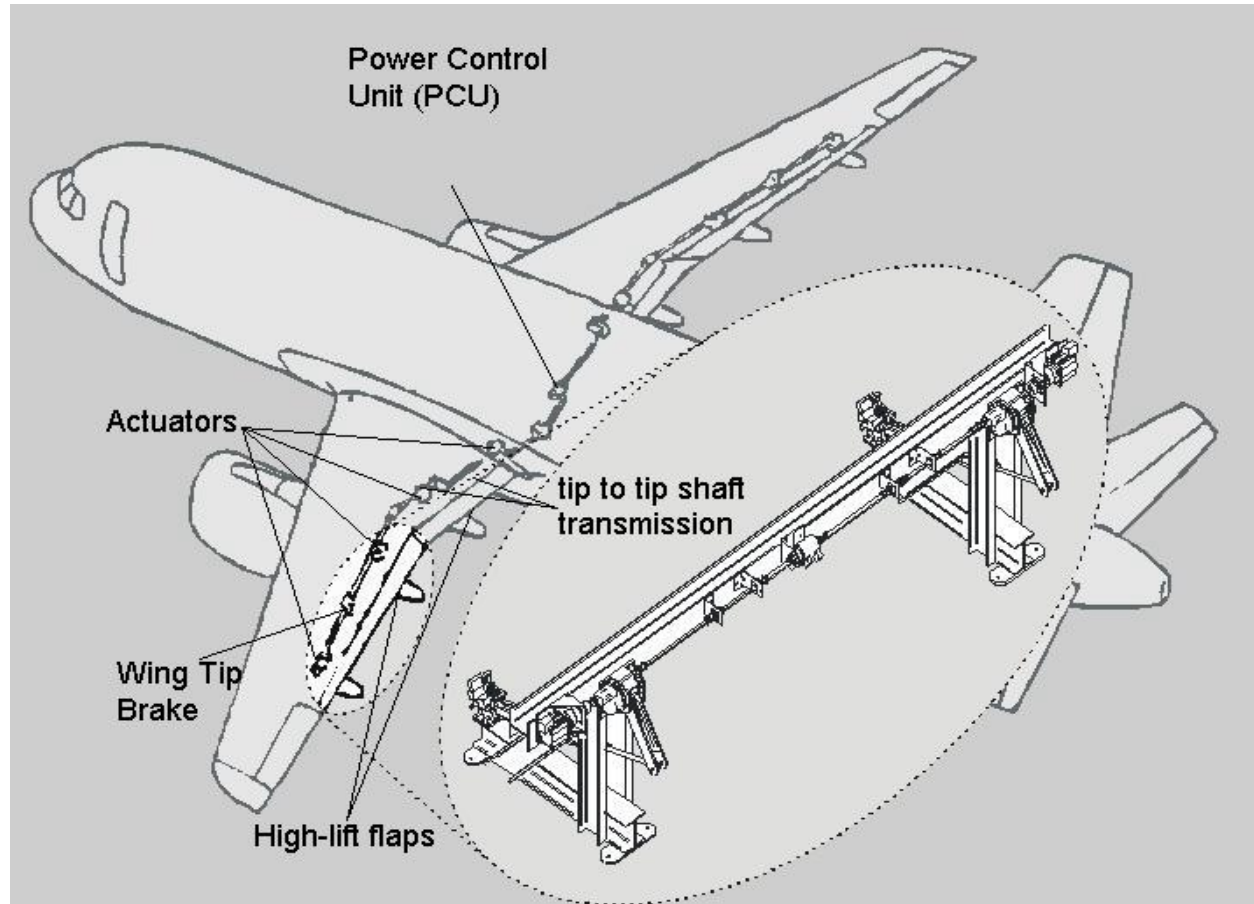
Subsystems

- ◆ Traffic Jam Assistant
- ◆ Heading Control
- ◆ Adaptive Lighting
- ◆ Door Control System



DECOS Application: Aerospace

Flap Control Demonstration System for Airbus Outer Flap System

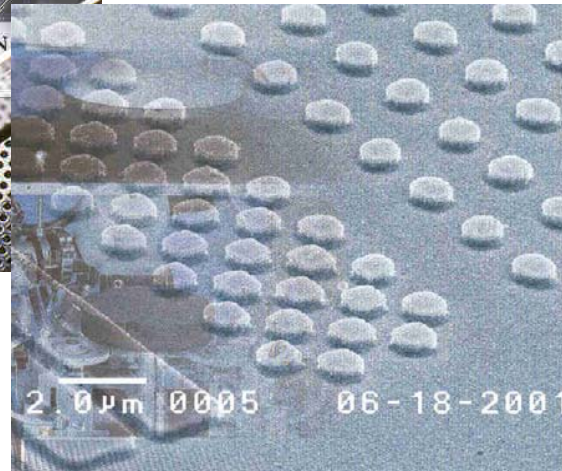


DECOS Application: Industrial Control

Vibration Control Demonstration System for Nano Imprinting Machines



Suppression of **critical vibrations**
in high-end nano-imprinting machines
for next-generation Sensors, Microoptics,
Organic Electronics, Bio- and Nanotechnology.



Other application areas:
machinery, automotive.
construction engineering

DECOS "Wasteline" Architecture Model

■ DECOS high-level services

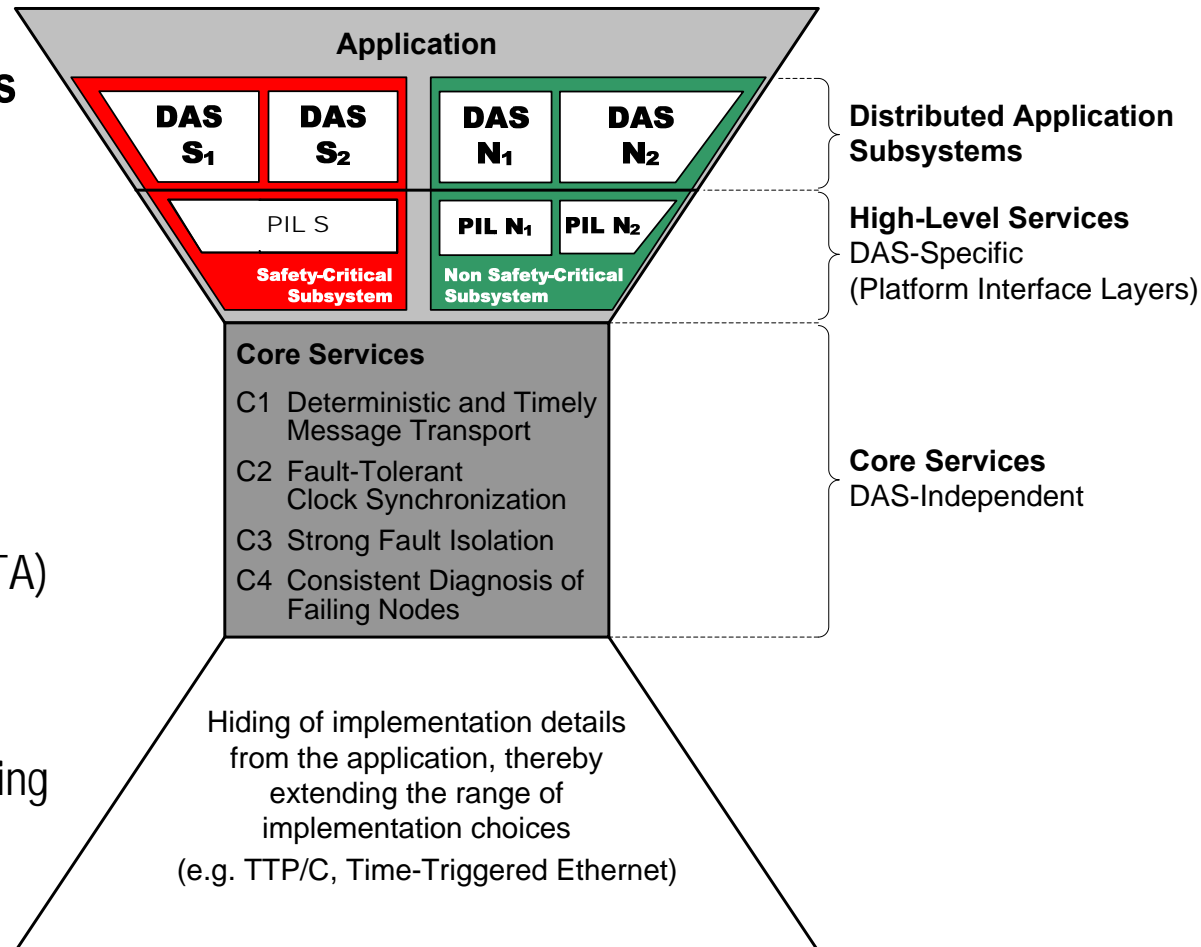
- ◆ Encapsulated Execution Environment
- ◆ Virtual networks
- ◆ Gateways
- ◆ Diagnosis service
- ◆ Fault Tolerance Layer

■ DECOS core services

- ◆ Prevalidated (FIT, NEXT TTA)

■ Domain and Platform Independence:

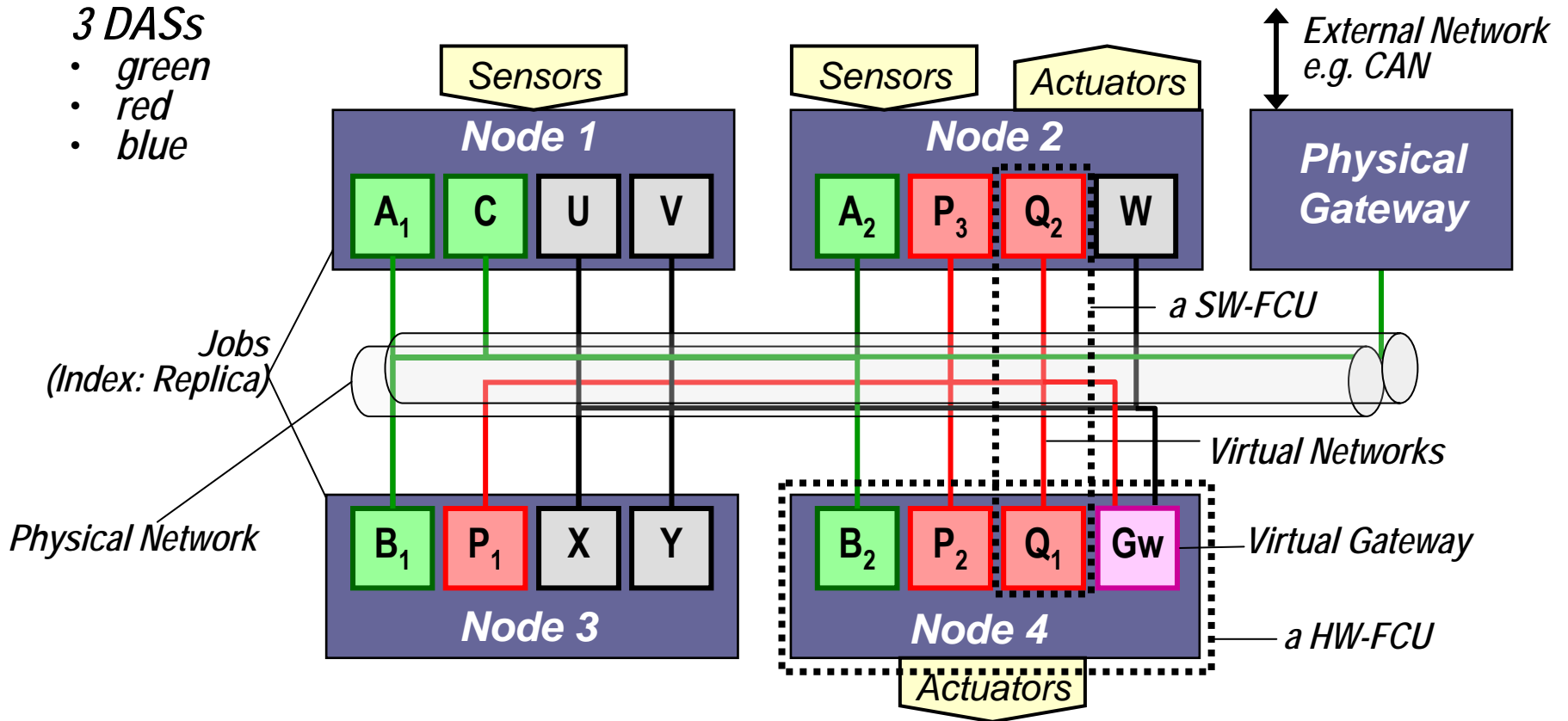
- ◆ Any core technology providing core services suffices
- ◆ (TTP/C, FlexRay, TT-Ethernet, ...)



DECOS High Level Services

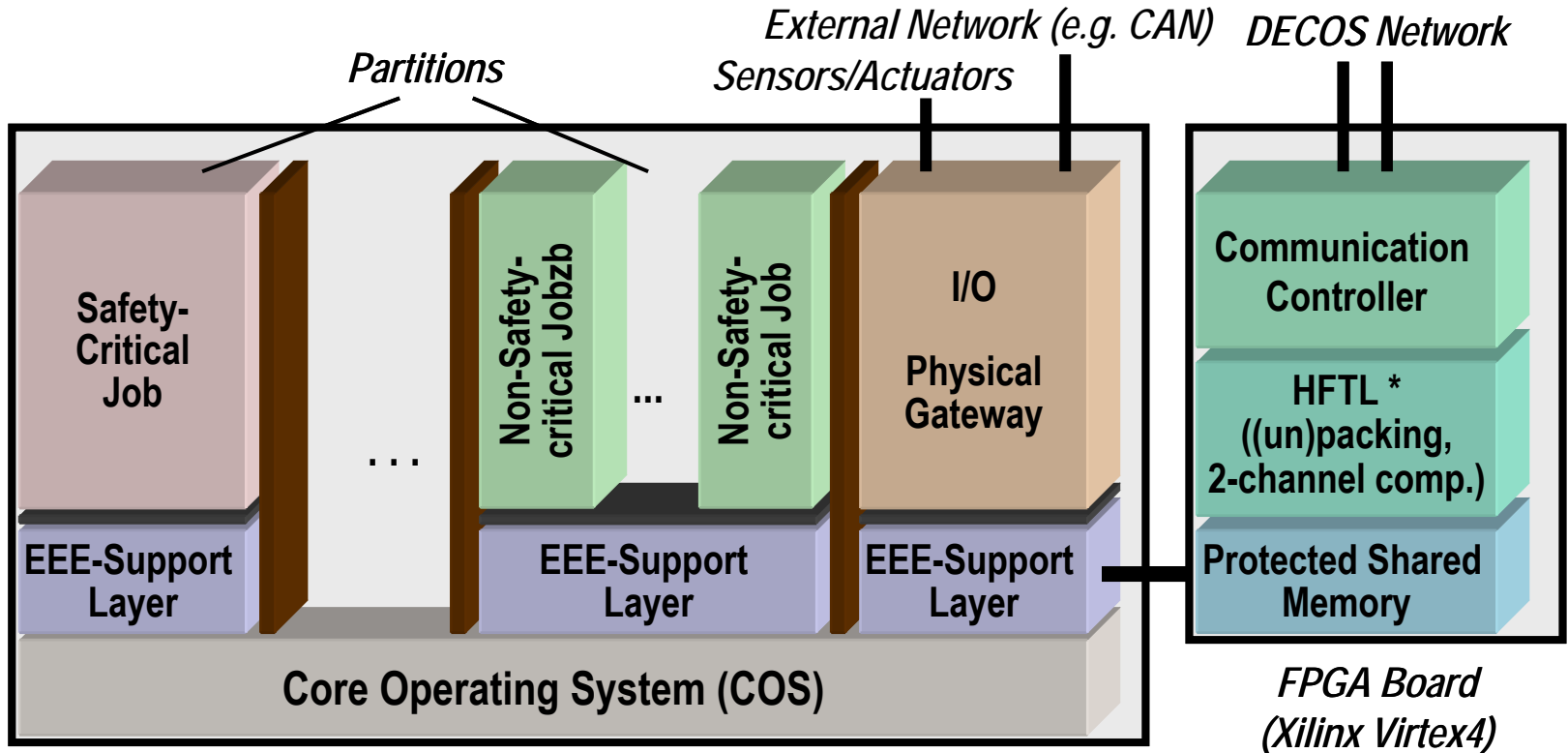
- **Virtual Networks**
 - ◆ the original "federated" networks
- **Encapsulation**
 - ◆ suppression of fault propagation among (jobs of) different DASs
- **Fault Tolerance Layer (FTL)**
 - ◆ compensation of hardware (and software) faults
- **Diagnosis Service**
 - ◆ fault categorisation (transient, permanent), distribution analysis
e.g. for smart wear-out warning
- **Virtual Gateways**
 - ◆ enable communication among DASs
- **Physical Gateways**
 - ◆ enable communication with non-DECOS networks

DECOS Cluster Architecture (Example)



Fault-Containment Units (FCU): Hardware – Node, Software – Job (all replicas)

Implementation on DECOS Platform



Encapsulated Execution Environment 'EEE' (TC 1796)

EEE-Support Layer: oFTL + SIL
(optimized FTL + System Interface Layer)

Per partition: - memory protection
- execution time slot

* Hardware FTL

Time-Triggered (TT) / Event-Triggered (ET)

■ Computation

- ◆ Time-triggered jobs (safety-critical): periodic execution
- ◆ Event-triggered jobs (non s-c): "on demand" execution

■ Communication

- ◆ by means of messages (with predefined type and (max) length)
- ◆ Time-triggered messages:
 - periodic transmission (known latency, minimal jitter)
 - *state semantics*: new message instance overwrites previous ⇒ "State messages"
- ◆ Event-triggered messages:
 - mean transmission rate (predefined distribution)
 - *event semantics*: messages are queued ⇒ "Event messages"

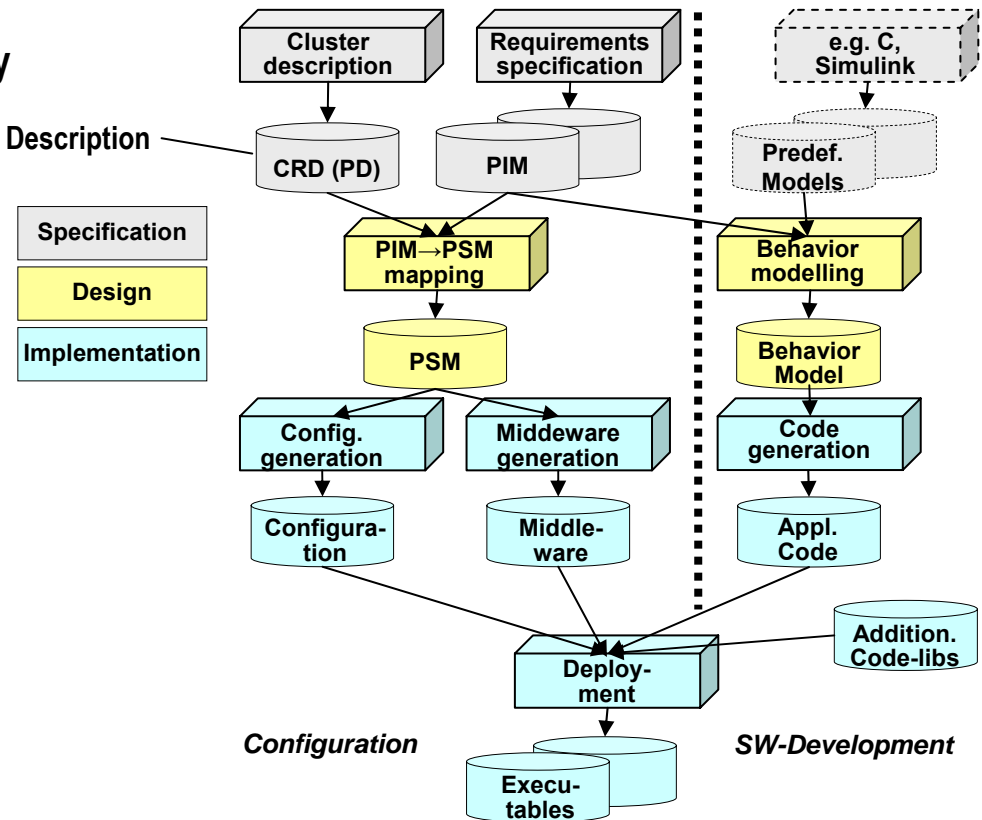
■ Constraints

- ◆ Safety-critical ⇒ Time-triggered
- ◆ Event-triggered ⇒ Non safety-critical

Tool Chain: Integrated Development Support

"From Requirements To Deployment"

1. Requirements
 - ◆ functional, performance, dependability
2. Cluster modelling
 - ◆ nodes, network
3. Behaviour modelling
 - ◆ of jobs
4. Configuration
 - ◆ allocation and scheduling
5. Middleware generation
 - ◆ APIs, fault-tolerance
6. Deployment
 - ◆ compile, link, download
7. Verification & Validation (V&V)
 - ◆ accompanying (Test Bench)



Requirements / Application Modelling – PIM

- **1 PIM per DAS (Distributed Application Subsystem)**
- **Functional (structural) Requirements**
 - ◆ *Jobs*: time- or event-triggered
 - ◆ *Messages*: state ~ or event ~
 - ◆ *State variables*: types
 - values transmitted in messages
 - ◆ *Interfaces*: which job sends/receives which messages
 - ◆ *I/O*: required sensors, actuators
 - ◆ ...
- **Performance Requirements**
 - ◆ *Jobs*: periods, deadlines, WCETs, ...
 - ◆ *Time-triggered messages*: periods, phase shifts, validity spans, ...
 - ◆ *Event-triggered messages*: min/mean/max production rates, ...
- **Dependability Requirements**
 - ◆ *Jobs*: SIL, redundancy degree, failure mode, ...
 - ◆ *Messages*: idempotency, redundancy degree, ...
 - ◆ *Resources*: latency
 - ◆ ...

Meta-model defined in UML (incl. OCL-constraints)

PIM Generation

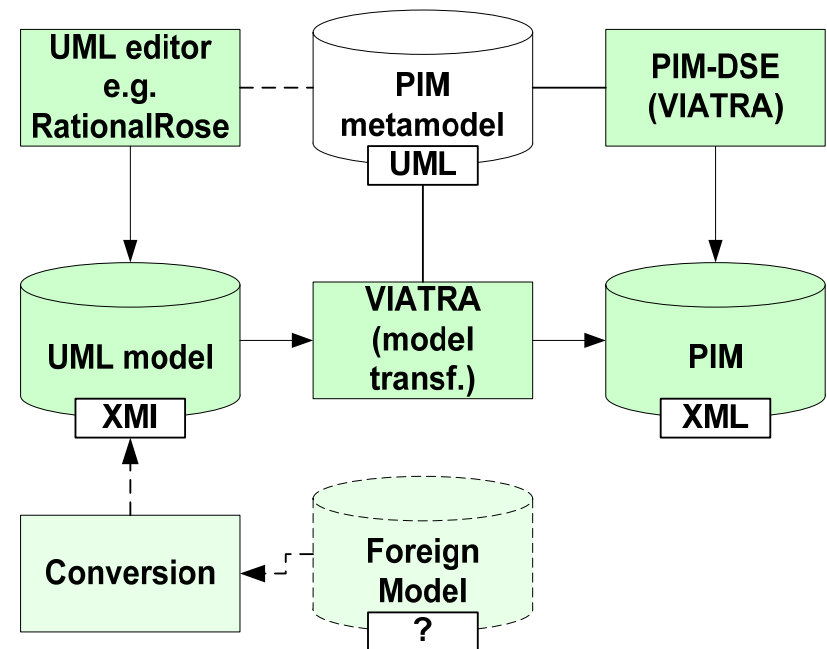
Either

- With UML-tool
- Directly in XML (XMI)
- With some other tool
 - ◆ e.g. Spreadsheet
 - ⇒ Conversion needed

proof of conformance
with meta-model needed

Or

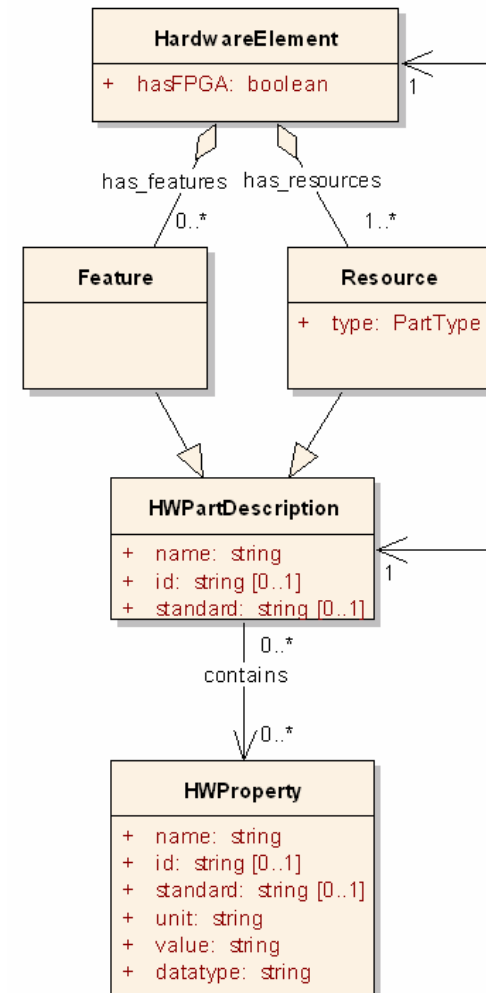
- Domain-Specific Editor (PIM-DSE)
 - ◆ implemented in VIATRA*
 - ◆ only meta-model conform PIMs



* "Visual (Model) TRAnsformations",
by Budapest University of Technology and Economics,
Eclipse plug-in

Cluster Modelling

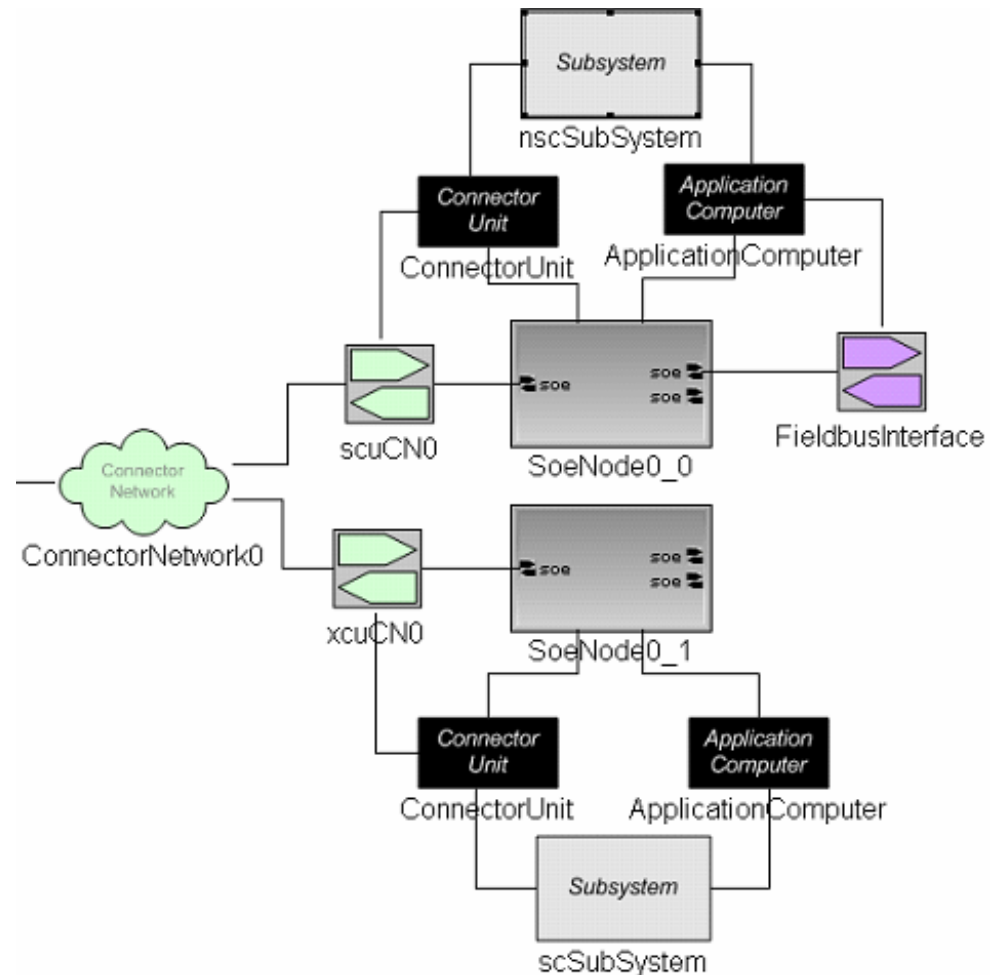
- **Description of available resources**
 - ◆ *Computational resources*
(i.e. processors and memory elements)
 - ◆ *Communication resources*
(i.e. communication interfaces, controllers and connectors)
 - ◆ *Special purpose HW*
(e.g. sensors, actuators, physical gateways)
- **Models**
 - ◆ **Meta-model ("Hardware Specification Model" HSM)**
 - UML-oriented, analogue to PIM meta-model
 - ◆ **Specific models ("Cluster Resource Description" CRD)**
 - finally XML



CRD Generation

(Similar to PIM)

- **Either with foreign tools**
- **Or with "CRD-DSE"**
 - ◆ **implemented with GME***
 - ◆ **only meta-model conform CRDs**



* "Generic Modelling Environment ", by Vanderbilt University

PSM generation

■ PIM marking

(adding information missing in PIMs and CRD), e.g.

- ◆ *PIM-PSM data type mapping*: assigning platform specific types to abstract type names in PIM (e.g. "real" → "float")
- ◆ *Special message protocols*: e.g. for "tunneling" CAN messages in DECOS messages
- ◆ *Job code information*: memory, CPU, and other resource requirements

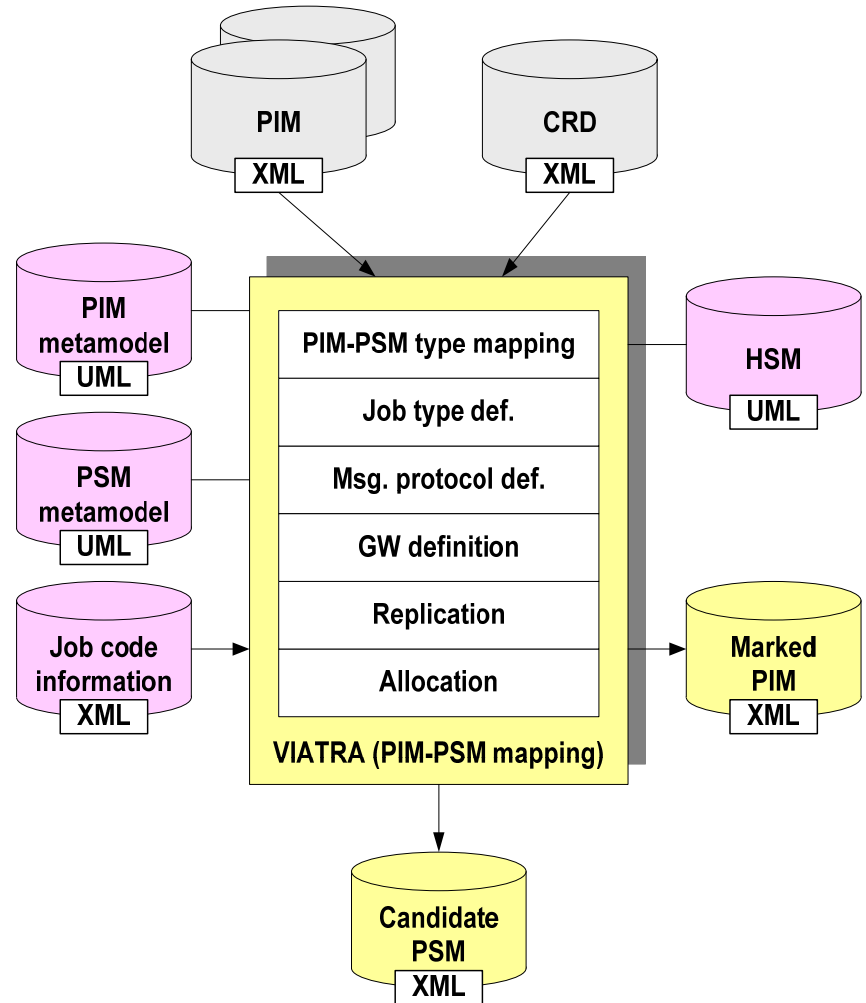
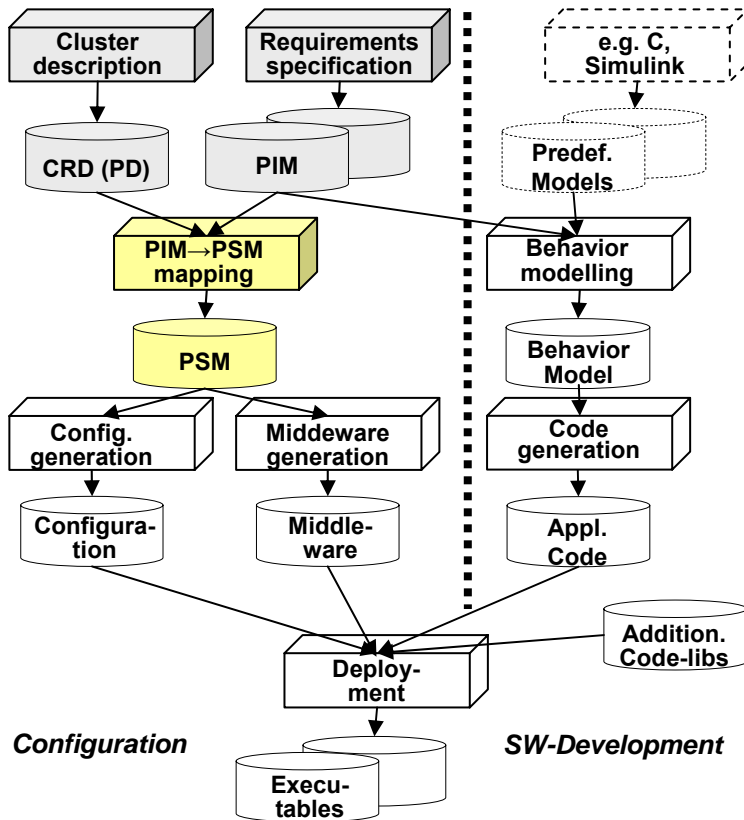
■ Allocation

- ◆ Jobs to nodes
- ◆ Sensors/actuators to jobs

under consideration of available data and constraints, e.g.
"replicas of same job never on same node"

- ◆ Either manually– needs explicit constraints check
- ◆ Or automated

Tool-Chain / Part 1

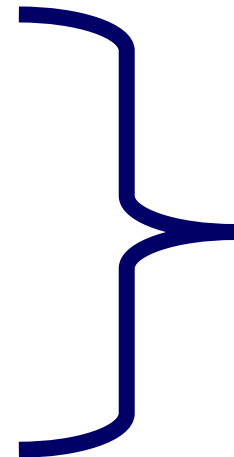


Configuration and Middleware Generation

Note: PSM is still a model, platform-independent with respect to its representation

■ Goals

- ◆ Message scheduling
- ◆ Job execution scheduling
- ◆ Memory layout
- ◆ Fault Tolerance Layer (FTL)
- ◆ Platform Interface Layer (PIL)



*Code and
control data
generation*

■ for given platform

- ◆ DECOS primary platform: EEE on TC1796 / Virtex 4 with „Layered TTP“ or „Layered FlexRay“ as comm. protocol
- ◆ but others should be possible

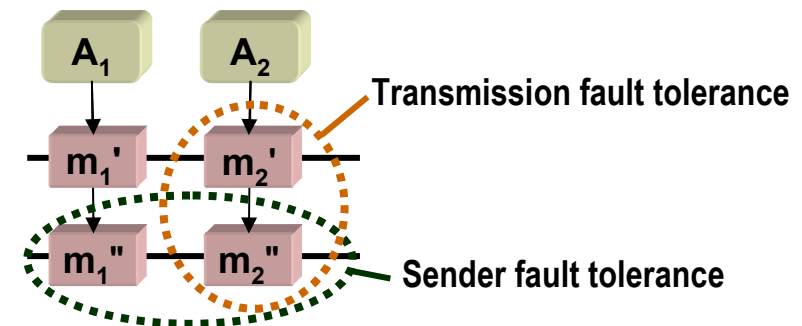
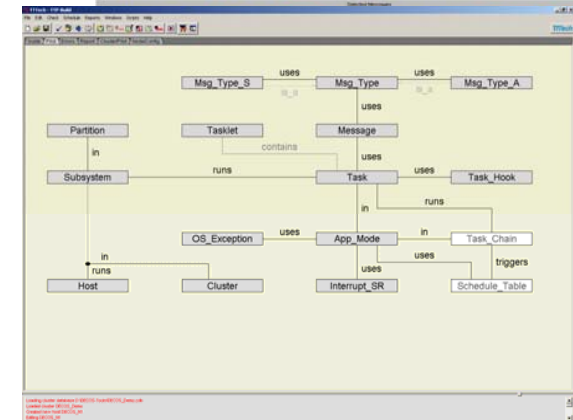
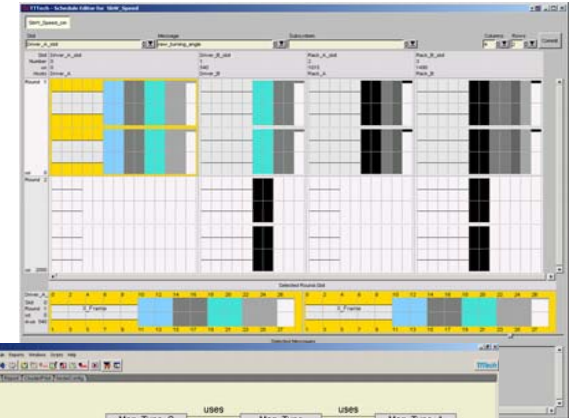
Scheduling and Fault Tolerance

■ Scheduling

- ◆ Extension of existing tools from TTTech
 - DECOS-TTPplan: message scheduling
 - DECOS-TTPbuild: partition scheduling

■ Fault tolerance

- ◆ Transmission faults:
 - redundant message transmission
 - bit-wise comparison of copies (in HW)
 ⇒ „HFTL“ – on FPGA
- ◆ Sender faults:
 - sender job replicas
 - voting over message replicas (in SW)
 ⇒ „oFTL“ (optimised FTL) – on TriCore



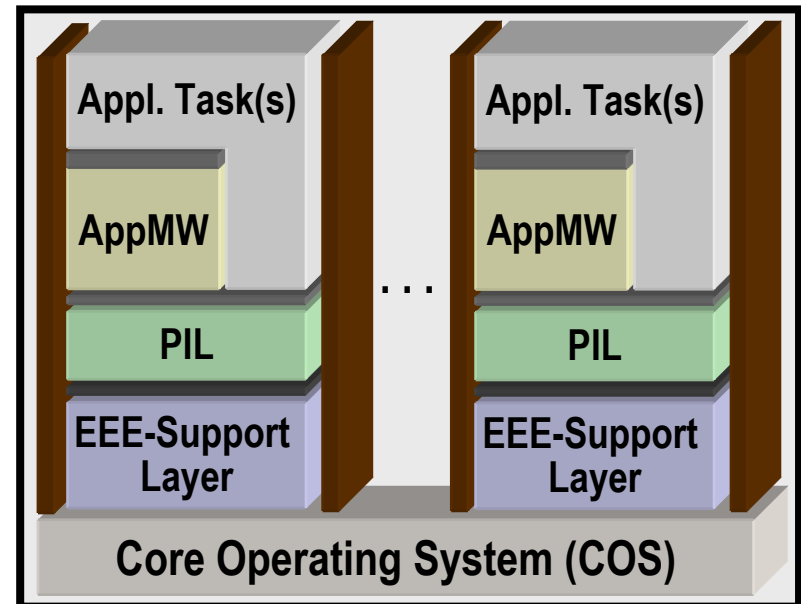
Platform Interface

PIL (Platform Interface Layer)

- Platform-independent boundary between DECOS applications and underlying technology platform
- Basic PIL services
 - ◆ TT and ET virtual network services
 - ◆ cluster time and membership services
- PIL-API
 - ◆ DAS-specific – generated from PIM
 - ◆ Programming-language specific
 - ◆ "Safe" (help to avoid coding mistakes)
 - ◆ e.g. in C: use message and variable names/types rather than char* or void*

AppMW (Application Middleware):

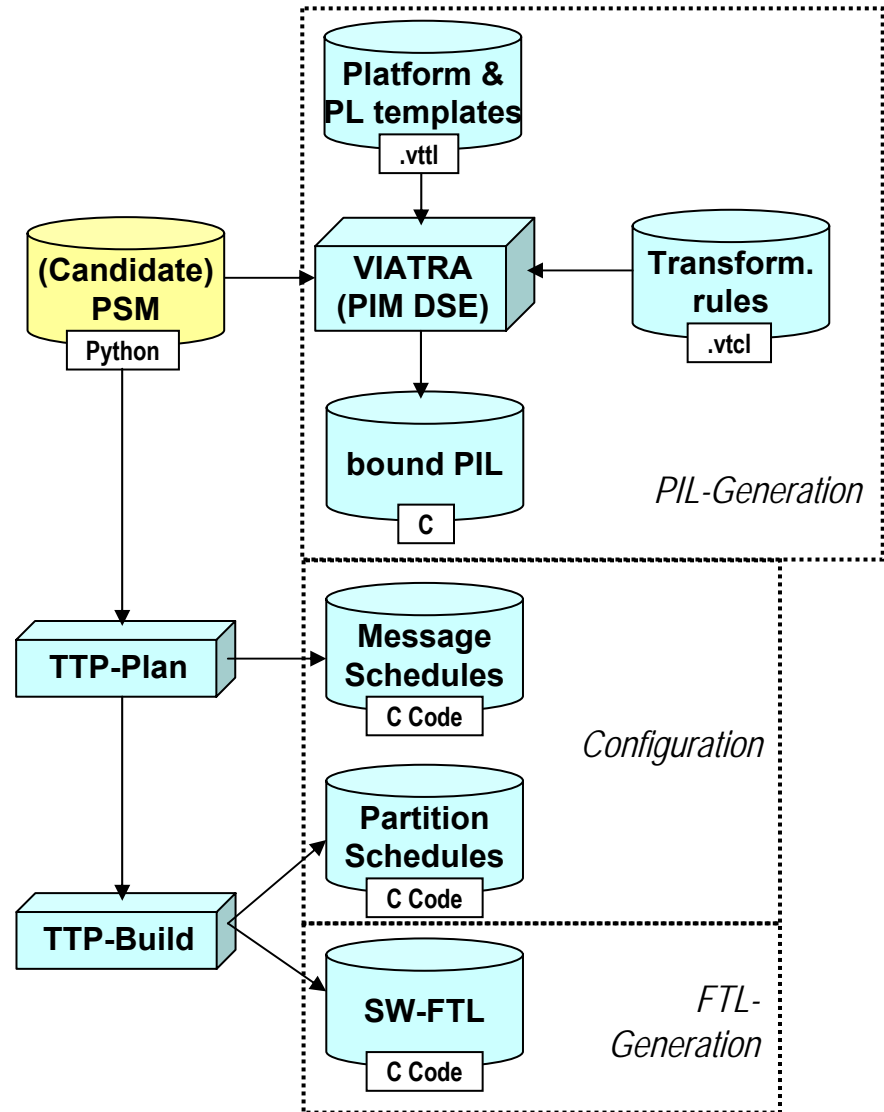
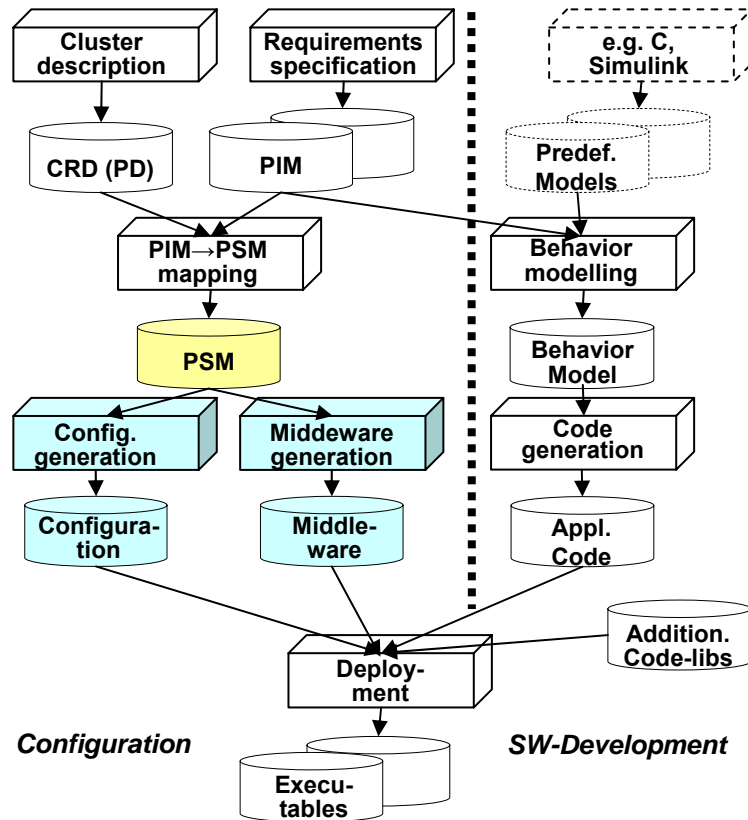
- optional domain-oriented services (e.g. for CAN)



- e.g., event messages not supported by EEE-support layer
- ⇒ queueing realised in PIL

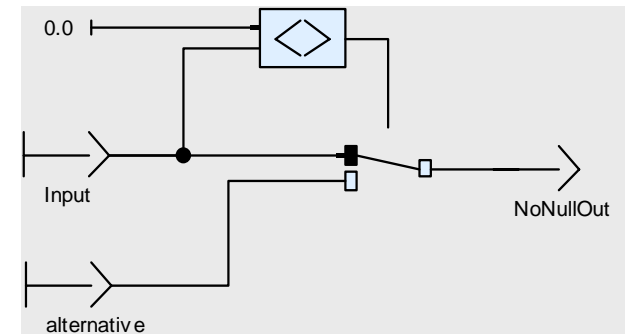
Job = Task(s) + PIL (+ AppMW)

Tool-Chain / Part 2



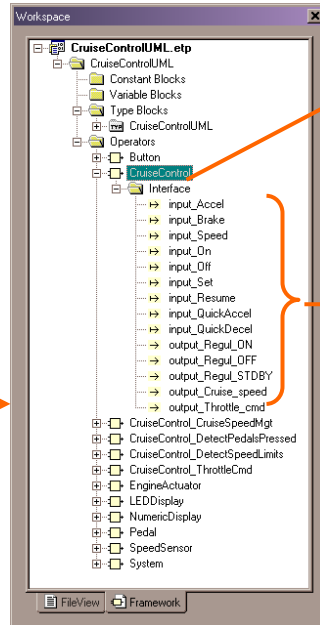
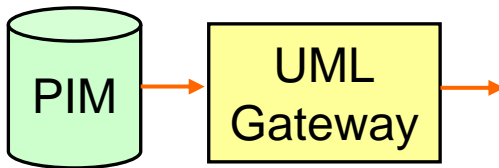
Behaviour Modelling

- **Considerations**
 - ◆ UML (including Action Semantics) not well suited for precise behaviour modelling
 - ◆ Could be done in C, but error prone
 - ◆ Data-flow languages increasingly exploited for embedded systems
- **Implication: using SCADE* for behaviour modelling**
 - ◆ **Based on a formally defined data-flow notation**
 - strong typing, explicit initialization
 - explicit time management (delays, clocks, etc)
 - simple expression of concurrency (data dependencies)
 - ◆ **Graphical data-flow editor**
 - ◆ **Immediately executable**
 - Simulation (MIL)
 - ◆ **Verification**
 - model checking, test coverage analysis
 - ◆ **Qualified code generation**
 - KCG code generator, certified against DO178B level A and IEC 61508 at any SIL level

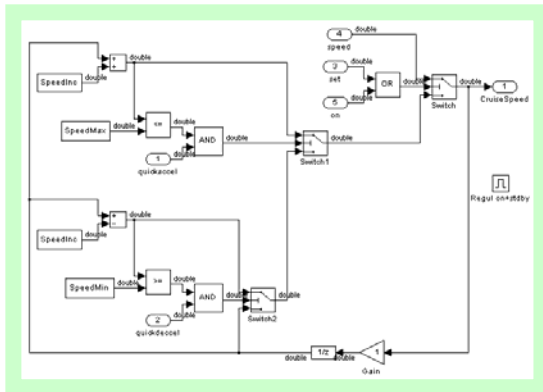
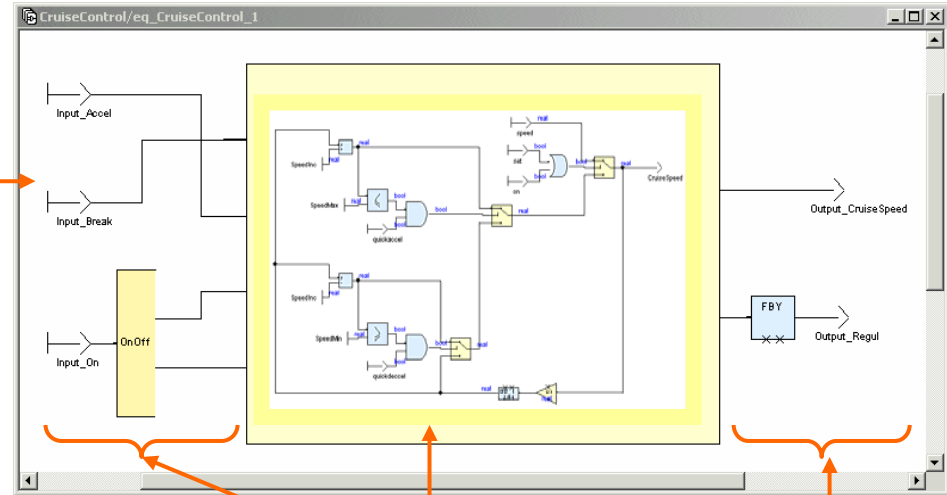


* " Safety-Critical Application Development Environment ", by Esterel Technologies

SCADE Gateways



1 job → 1 SCADE node



optional

Interface adaptation

SCADE Simulink Gateway

Code Generation

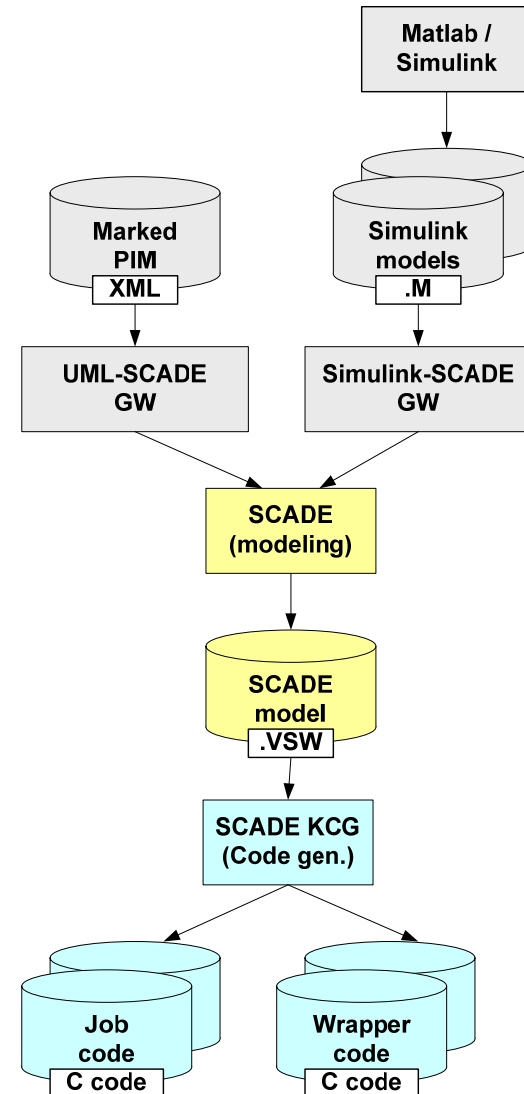
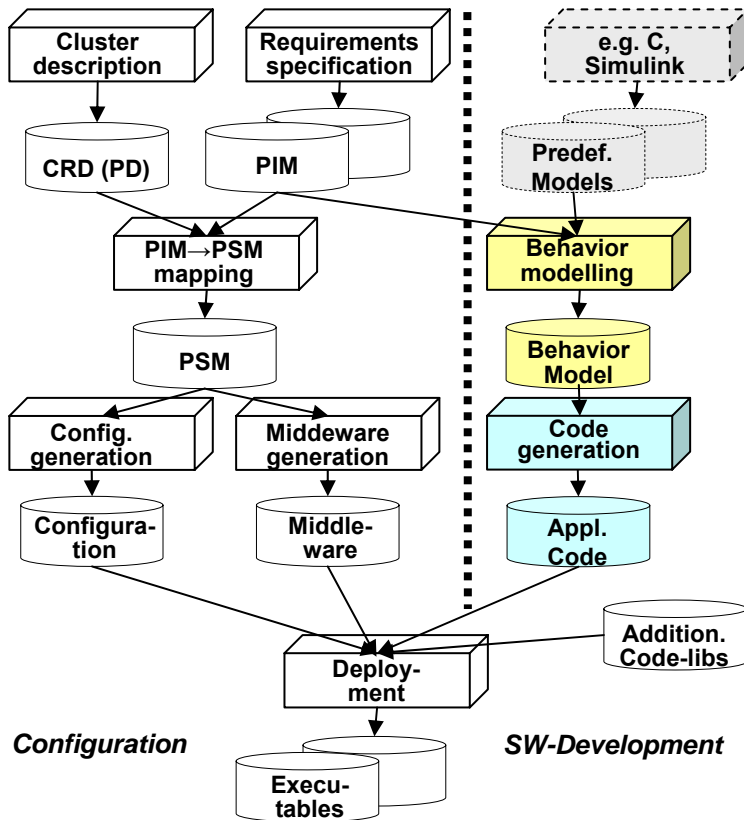
Standard (KCG):

- Per node, 1 C-function of same name
- Data transfer between nodes:
 - ◆ context structures, containing input and output parameter

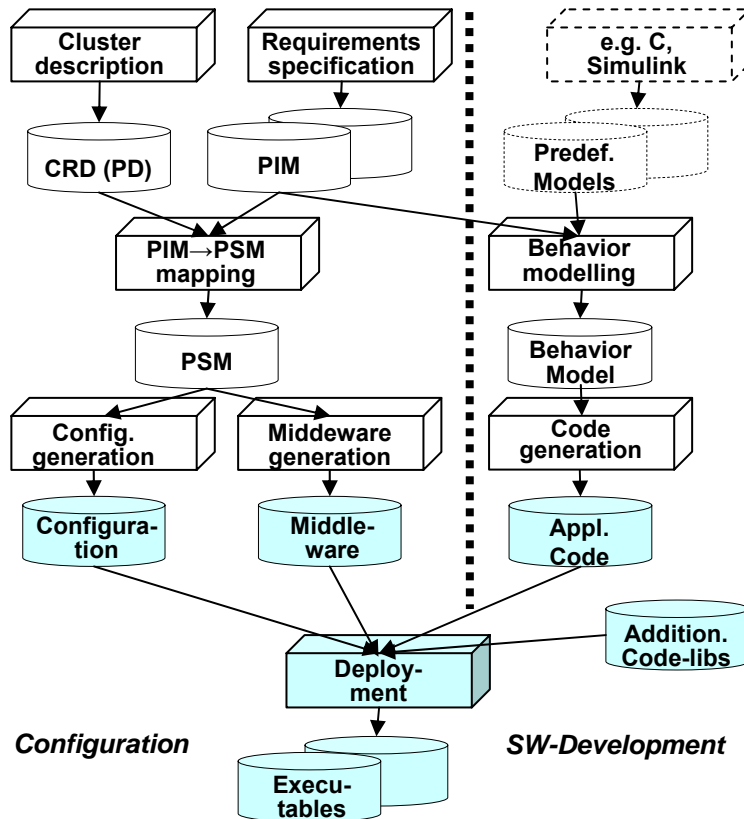
DECOS specific:

- "Wrapper": access to PIL
 1. read input messages from PIL into context structures (considering availability)
 2. execute node
 3. write output messages from context structures into PIL (considering availability)

Tool-Chain / Part 3



Deployment (Tool-Chain / Part 4)

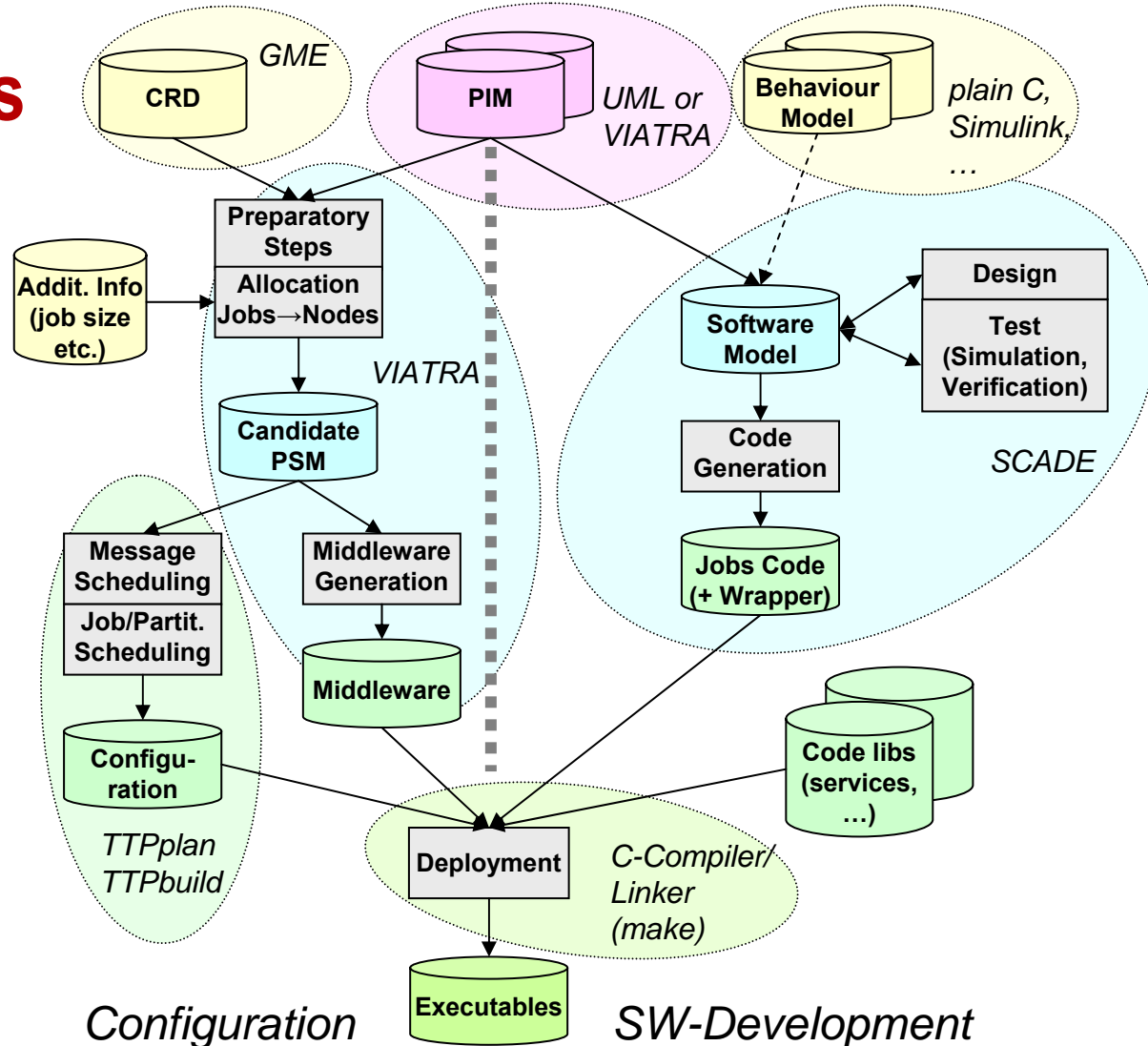


- **Activities**
 - ◆ **Compile**
 - ◆ **Link**
 - ◆ **Download**
- **Commercial tools used**

Tool-Chain – Tools

- CRD – Cluster Resource Description
- DAS – Distributed Application Subsystem
- PIM – Platform Independent Model
- PSM – Platform Specific Model
- PIL – Platform Interface Layer

- GME – Generic Modelling Environment
(Vanderbilt University)
- VIATRA – Visual Automated (Model)
Transformations
(Budapest University of Technology
and Economics - BUTE)
- SCADE – Safety-Critical Application
Development Environment
(Esterel Technologies)
- TTP – Time-Triggered Protocol
(TTTech)

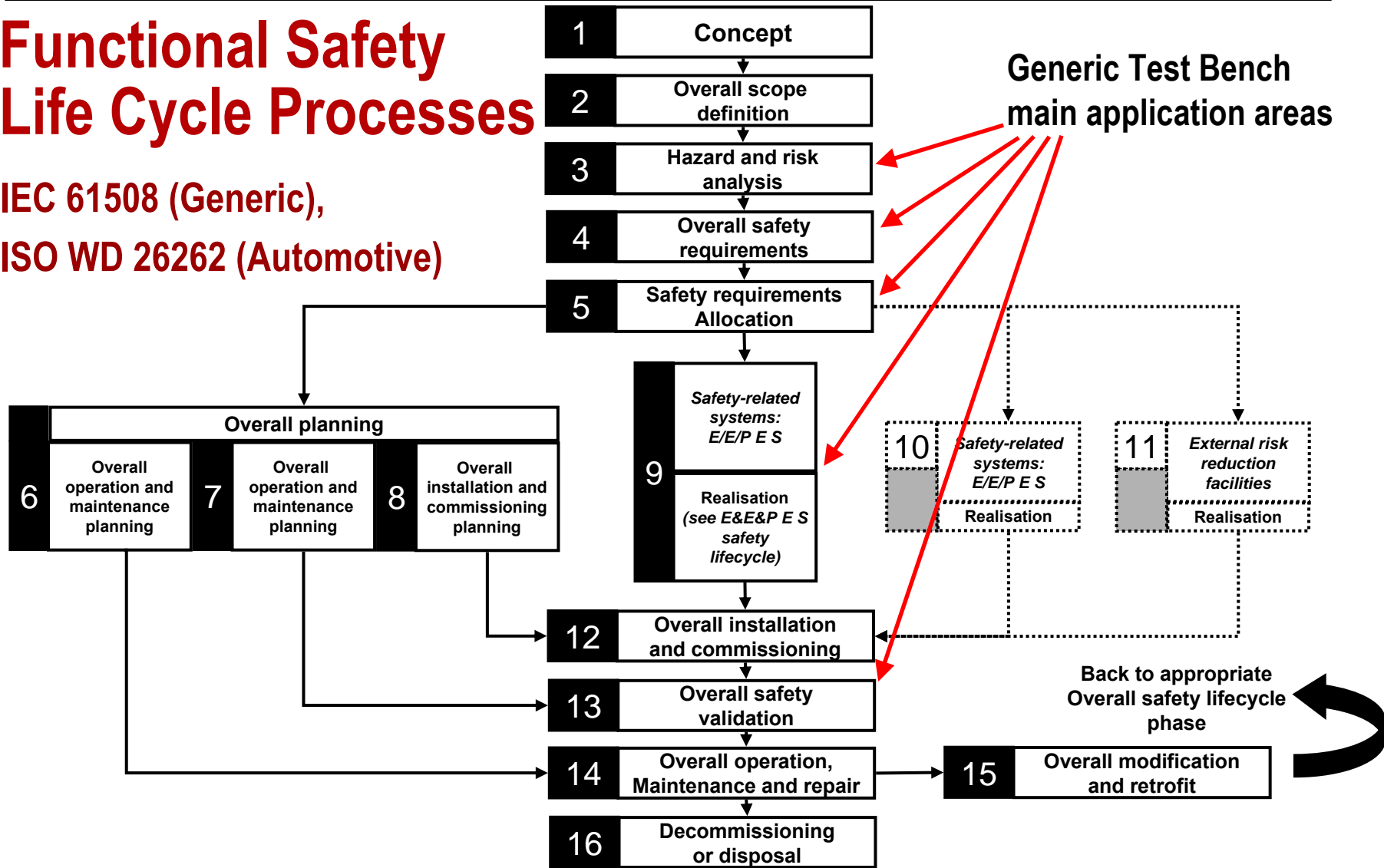


Generic Test Bench – Objectives

- **Guides designers through the verification and validation (V&V) process**
 - ◆ Follows functional **safety standards** (e.g. IEC 61508 and related sectoral standards, e.g. EN 50129, ISO 26262, ...)
 - ◆ Defines a **(requirements driven) workflow** for generation of modular (component-based) *safety cases*
- **Helps in identifying and carrying out V&V activities**
 - ◆ Constitutes a **framework** for integration of V&V capabilities (methods, tools)
 - ◆ for the various DECOS artefacts (architecture model, core/high level services, hardware, tools, applications)
 - ◆ Provides **validation plans** (v-plans) to control V&V process (traceability, completeness)
- **Collects documented results of V&V activities in repository for establishing safety case (and for reuse)**

Functional Safety Life Cycle Processes

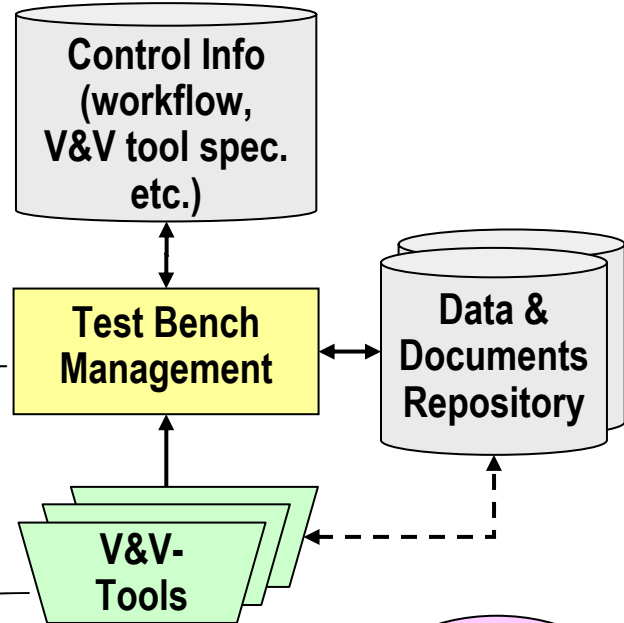
IEC 61508 (Generic),
ISO WD 26262 (Automotive)



Generic Test Bench – Ingredients

ID	VVStatus	Type	Phase	V&V-Activity
1 V-Plan Components and Middleware				
VPCM58	Not ready			
VPCM1	Not ready	Compound		
VPCM2	Completed	Compound		
VPCM3	Completed	Elementary		
VPCM4	Completed	Elementary		
VPCM5	Completed	Compound		
VPCM6	Completed	Elementary		
VPCM7	Completed	Elementary		
VPCM8	Completed	Elementary		
VPCM9	Completed	Compound		
VPCM10	Completed	Elementary		
VPCM11	Not ready	Elementary		
VPCM12	Not ready	Elementary		
VPCM13	Not ready	Elementary		
VPCM14	Not ready	Elementary		
VPCM15	Not ready	Elementary		

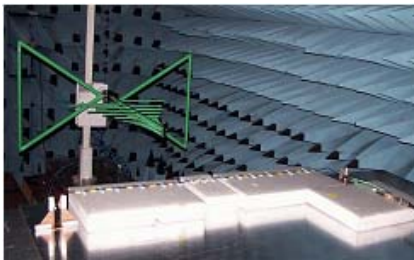
Implemented in DOORS™



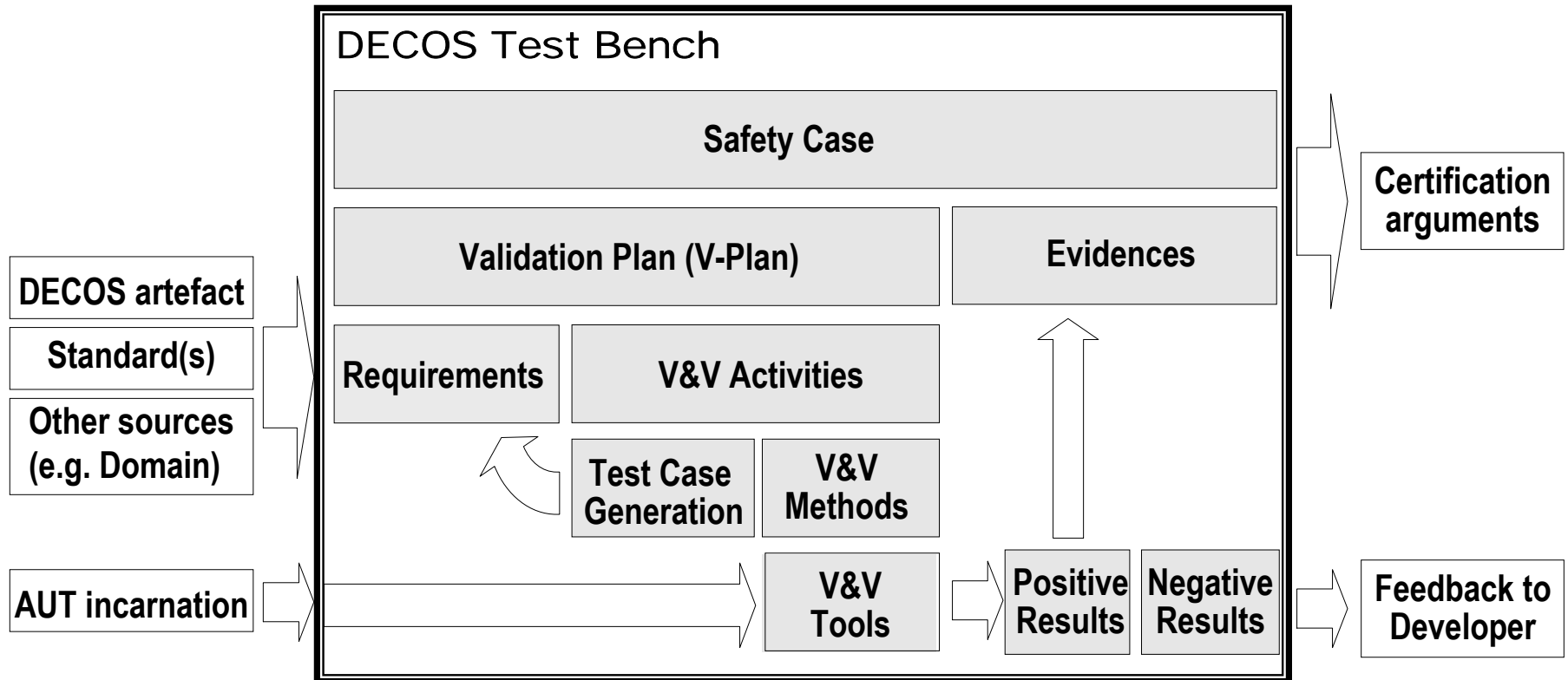
e.g.

- ◆ ITEM (Hazard and Risk Analysis)
- ◆ RACER (Formal Verification)
- ◆ SCADE MTC (Simulation)
- ◆ LDRA (Testing)
- ◆ PROPANE (Fault Injection)
- ◆ EMV Test Bench

- ◆ Models
- ◆ Tools
- ◆ Middleware
- ◆ Hardware
- ◆ Applications



Generic Test Bench – Conceptual Framework



'AUT': Artefact Under Test

Generic Test Bench: distributed implementation

Test Bench Framework – V-Plan example

Formal module '/DECOS_TestBench/V-Plans/Components and Middleware' current 0.0 - DOORS

File Edit View Insert Link Analysis Table Tools User Decos Help

Overview All levels

ID		VVStatus	Type	Phase	V&V-Activity	Input
VPCM53	1 V-Plan Components and Middleware	Not ready				
VPCM1	Arch-gen-1	Not ready	Compound			
VPCM2	Arch-gen-core-1	Completed	Compound			
VPCM3	Arch-core-predictable-transport-1	Completed	Elementary		prevalidated	.
VPCM4	Arch-core-ft-clock-sync-1	Completed	Elementary		prevalidated	.
VPCM5	Arch-core-fault-isolation-1	Completed	Compound			
VPCM6	Arch-core-fault-hypothesis-1	Completed	Elementary		prevalidated	.
VPCM7	Arch-core-never-give-up-1	Completed	Elementary		preva	
VPCM8	Arch-core-transient-faults-1	Completed	Elementary		preva	
VPCM9	Arch-core-consistent-diagnosis-1	Completed	Elementary		preva	
VPCM10	Arch-gen-core-2	Not ready	Compound			
VPCM11	Arch-DECOS-high-level-services-1	Not ready	Compound			
VPCM12	Arch-DECOS-exec-1				Functional Testing	EEE-implem.
VPCM13	Arch-DECOS-comm-1					
VPCM14	Arch-DECOS-diagnosis-1	Not ready	Compound			
VPCM15	Arch-ft-1	Not ready	Compound			
VPCM16	Arch-ft-2	Not ready	Compound			
VPCM17	Arch-ft-3	Not ready	Compound			
VPCM18	Arch-eee-temporal-1	Not ready	Compound			
VPCM19	Arch-eee-temporal-2	Not ready	Compound			
VPCM20	Arch-eee-temporal-2-1	Not ready	Elementary		Functional Testing	EEE-src, EEI
VPCM21	Arch-eee-temporal-2-2	Not ready	Compound			

CL1 Arch-DECOS-exec-1 The temporal ensured by CPU time,

VV50 Functional Testing Method

Username: Bill Young Exclusive edit mode

Test Bench Framework – Nested V-Plans

Format module: /DECOS_TestBench_test2/V-Plans/FT-COM V-Plan current 0.0 - DOORS

File Edit View Insert Link Analysis Table Tools User Decos Help

Overview All levels

ID	WVStatus	Type	Reconn	Phase	V&V-Activity	Input	Output	Pass/Fail	Errior
65	Not ready	Elementary	R	System design	Inspection of hardware		Report R-VP0-A9-<name>-<time stamp>	All activities of the v-plan executed with positive results	N/A
51	Not ready	Compound	R		Execution of validation plan VP1		Report R-VP0-A10a-<name>-<time stamp>	All activities of the v-plan executed with positive results	N/A
65	Not ready	Compound	R		validation plan VP2		Report R-VP0-A10b-<name>-<time stamp>	All activities of the v-plan executed with positive results	N/A
53	Not ready	Compound	R		Execution of validation plan VP3		Report R-VP0-A10c-<name>-<time stamp>	All activities of the v-plan executed with positive results	N/A
65	Not ready	Compound	R		Execution of validation plan VP3		Report R-VP0-A10d-<name>-<time stamp>	All activities of the v-plan executed with positive results	N/A
54	Not ready	Compound	R		Execution of validation plan VP3		Report R-VP0-A10d-<name>-<time stamp>	All activities of the v-plan executed with positive results	N/A
65	Not ready	Compound	R		Execution of validation plan VP3		Report R-VP0-A10d-<name>-<time stamp>	All activities of the v-plan executed with positive results	N/A

Format module: /DECOS_TestBench_test2/V-Plans/FT-COM V-Plan current 0.0 - DOORS

File Edit View Insert Link Analysis Table Tools User Decos Help

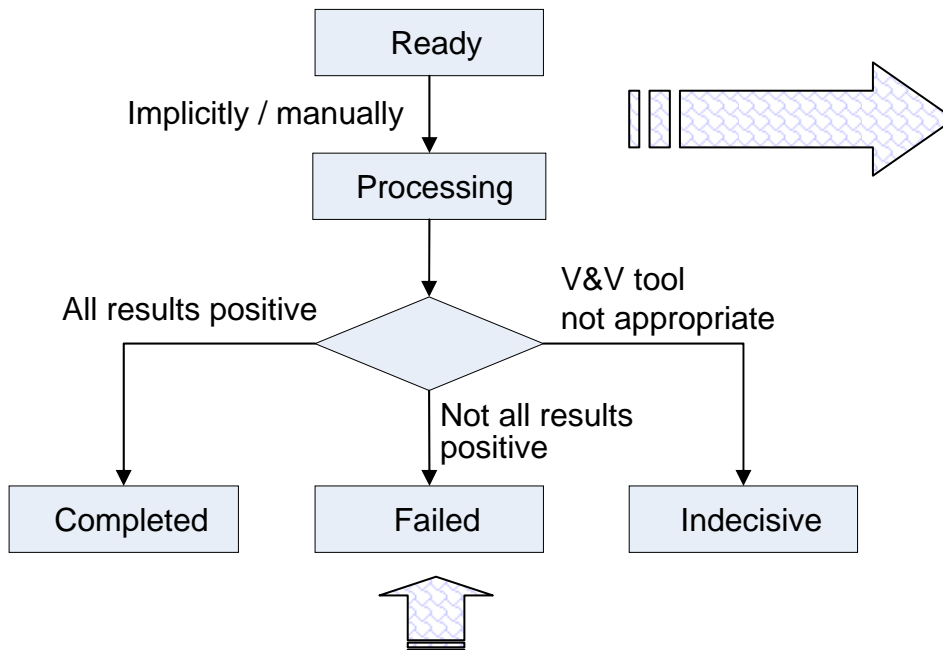
Overview All levels

ID	WVStatus	Type	Reconn	Phase	V&V-Activity	Input	Output	Pass/Fail	Errior
65	Not ready								
83	Not ready								
65	Not ready	Elementary	HR	Requirements specification	Inspection	Document "D2.4.1 Requirement Specification Optimized Fault-Tolerance Layer (FTL)"	Report R-VP2-A1-<name>-<time stamp>	All items of the checklist covered	N/A
84	Not ready	Elementary	HR	System design	Inspection	Document "D2.4.1 Requirement Specification Optimized Fault-Tolerance Layer (FTL)" Document "D2.4.2 Design Document of	Report R-VP2-A2-<name>-<time stamp>	All items of the checklist covered	N/A
65	Not ready	Elementary	HR	System design	Inspection	Document "D2.4.1 Requirement Specification Optimized Fault-Tolerance Layer (FTL)" Document "D2.4.2 Design Document of	Report R-VP2-A2-<name>-<time stamp>	All items of the checklist covered	N/A
85	Not ready	Elementary	HR	System design	Inspection	Document "D2.4.1 Requirement Specification Optimized Fault-Tolerance Layer (FTL)" Document "D2.4.2 Design Document of	Report R-VP2-A2-<name>-<time stamp>	All items of the checklist covered	N/A

Username: Bill Young Exclusive edit mode

Modular / compositional safety cases

Test Bench Framework – V&V Activity



E-Mail to Start V&V Activity - DOORS

E-Mail to Start V&V Activity:

E-Mail Address:

Subject:

Message:

<Sample text:>
Dear Erwin Schoitsch,

The V&V activity "Arch-DECOS-exec-1" (V-Plan "Components and Middleware") is ready for testing.

Please perform the required tests using the following input "EEE-implem." and return the results by November 2005 the latest.

IMPORTANT INFO: to allow the answer e-mail to be automatically processed, it should contain the test result (PASS/FAIL/INDECISIVE) as well as the links to the designated data repository.

With best regards,
N.N.

E-Mail to Communicate the Outcome FAILED of the V&V Activity - DOORS

E-Mail to Communicate the Outcome FAILED of the V&V Activity:

E-Mail Address:

Subject:

Message:

<Sample text:>
Dear Erwin Schoitsch,

The AUT "DECOS node" has to be changed due to the outcome FAILED of the V&V activity "Arch-DECOS-exec-1" (V-Plan

**V&V
Activity!**

Generic Test Bench – V&V Tool Integration

Tool integration levels

- **No external tool:** e.g. Checklist
 - ◆ Tool implemented in DOORS
- **Manually executed external tool:** e.g. PROPANE (SWIFI)
 - ◆ Start of tool in dialog (“pressing a button”)
- **Automatically executed external tool:** e.g. RACER (Ontology based consistency and completeness check)
 - ◆ Start of tool by "mailing" to corresponding server (no user interaction)
- **External test bench:** e.g. EMI Hardware Test Bench
 - ◆ Tool runs on separate hardware, feedback by email/message flow

For all levels, corresponding interaction workflows provided

Example for (remote) manual integration

EMI Hardware Test and Simulation

- test input sent to lab
via e-mail + links to data
 - ◆ detailed DUT (Device Under Test) description
 - ◆ EMC phenomena to be tested
- DUT provided by user (customer) to lab
- test equipment set up according to input
- tests executed 'manually' at lab
- test result and test report returned by e-mail

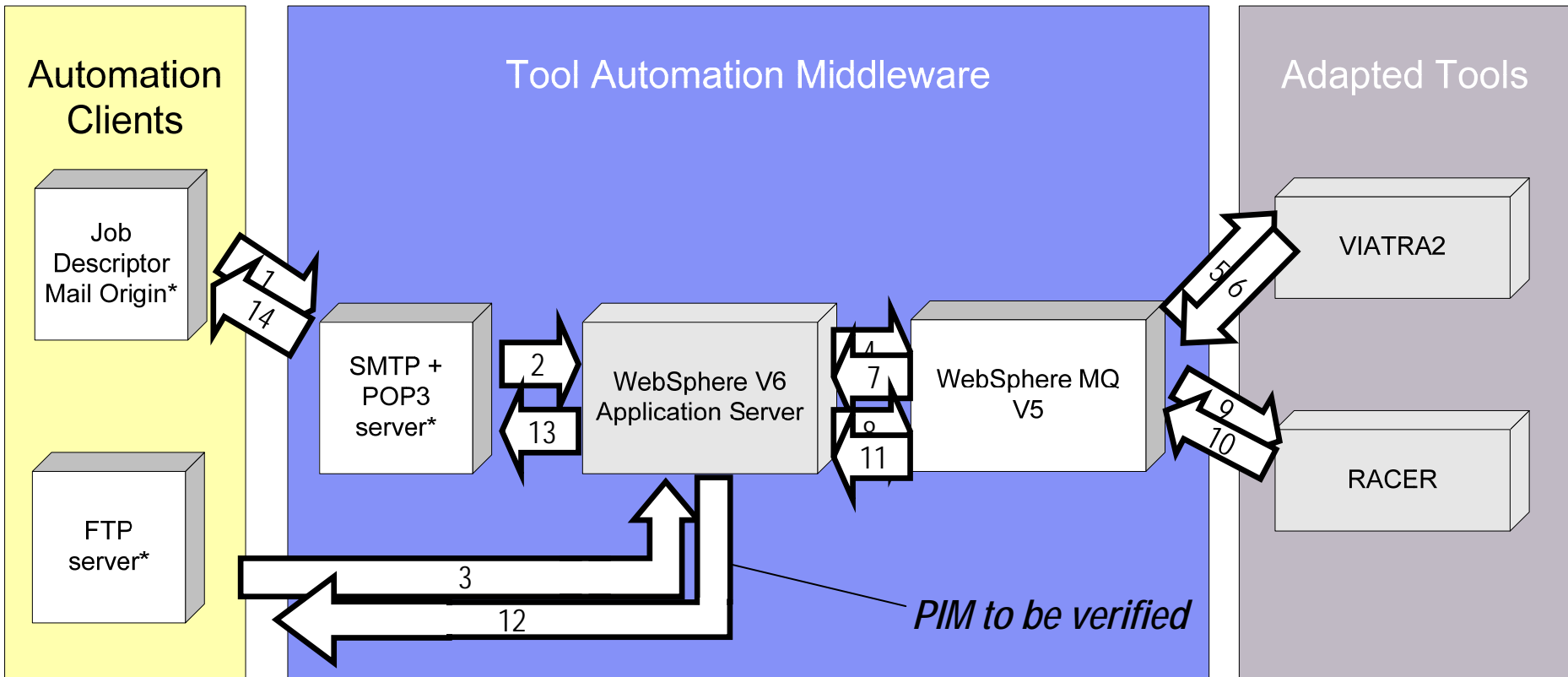


Format of input and results standardised

EXTENT OF THE TEST ↴			
→ please tick where applicable*			
Emission*		<input type="checkbox"/>	
Directive*	*		
Applied standards:*			
EN XXXXX*	<input type="checkbox"/>	EN XXXXX*	<input type="checkbox"/>
EN XXXXX*	<input type="checkbox"/>	EN XXXXX*	<input type="checkbox"/>
DOCUMENTATION OF THE TEST → Please tick the desired			
EN	Test report*		
EN	Test results only (raw data):*		Plots*
E			Disc*
			Mail*
Ad	Language: <input type="checkbox"/>		Report/Copy:*
En	German*	<input type="checkbox"/>	Original*
	English*	<input type="checkbox"/>	Duplicate*
		<input type="checkbox"/>	PDF*

Example for automatically executed external tool

PIM-validation with Racer



Danke für Ihre Aufmerksamkeit