FMI Industrial User Meeting: eFMI Status and Outlook

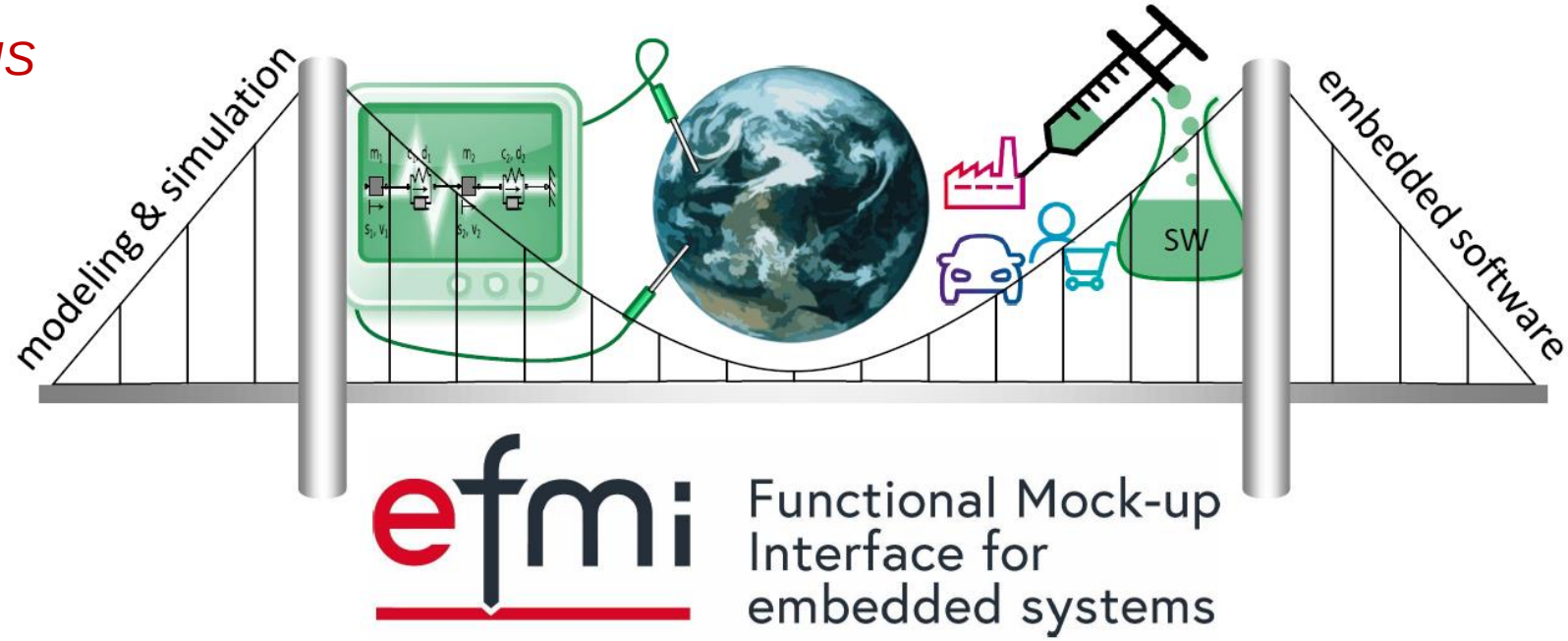
International Modelica Conference 2021, Sept. 23

Christoff Bürger
Dassault Systèmes
Christoff.Buerger@3ds.com

eFMI Standard: Mission

New standard enabling the application of (physics) models in embedded software by providing a container architecture for the step-wise refinement of a first high-level algorithmic solution to an embedded implementation on a dedicated target environment.

- Developed in the *EMPHYSIS* research project (https://emphysis.github.io/)
- Now Modelica Association Project (*MAP eFMI*) (https://efmi-standard.org/)



eFMI Standard: Status

New standard enabling the application of (physics) models in embedded software by providing a container architecture for the step-wise refinement of a first high-level algorithmic solution to an embedded implementation on a dedicated target environment.

EMPHYSIS results:

- 13 tools covering entire eFMI workflow
- Modelica library for cross-checking eFMU tooling
  - 22 applications, each with several variants
- 6 industry-driven demonstrators
- ITEA Award of Excellence winner

Detailed *paper* of standard and results:

- *"eFMI: An open standard for physical models in embedded software"*
- Monday, Sep. 20, Session 1A, 16:50-17:10

MAP eFMI status:

- 12 member organizations covering research, tool vendors and users.
- Test cases library published (https://github.com/modelica/efmi-testcases).
- Alpha draft of specification published (https://efmi-standard.org/).

efmi Functional Mock-up Interface for embedded systems

eFMI Standard: Container architecture & model representations

New standard enabling the application of (physics) models in embedded software by providing a container architecture for the step-wise refinement of a first high-level algorithmic solution to an embedded implementation on a dedicated target environment.

⇒ Standardized workspace for step-wise development of embedded solutions from models

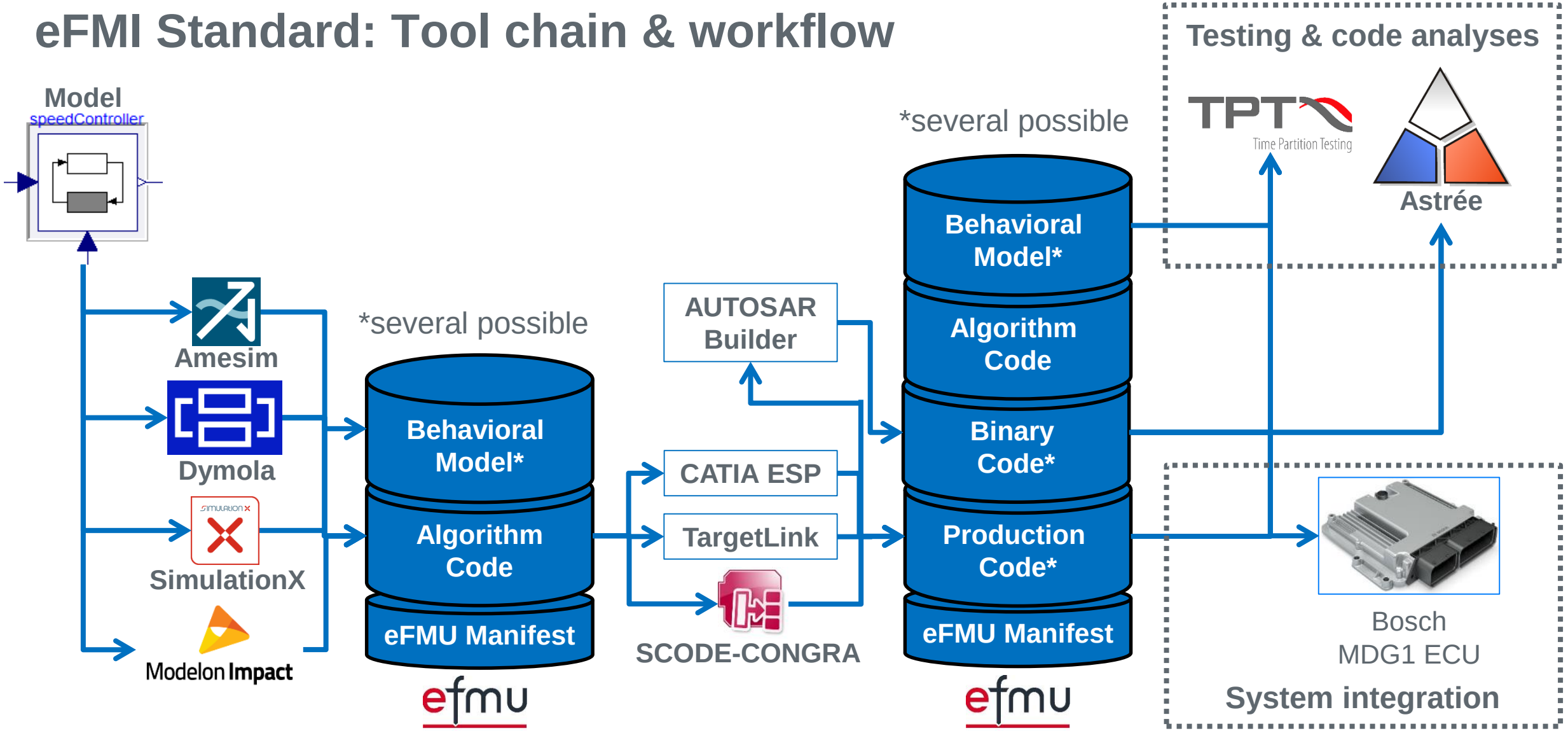
⇒ Covering most important development concerns (implementation, testing, integration)

eFMUs support:

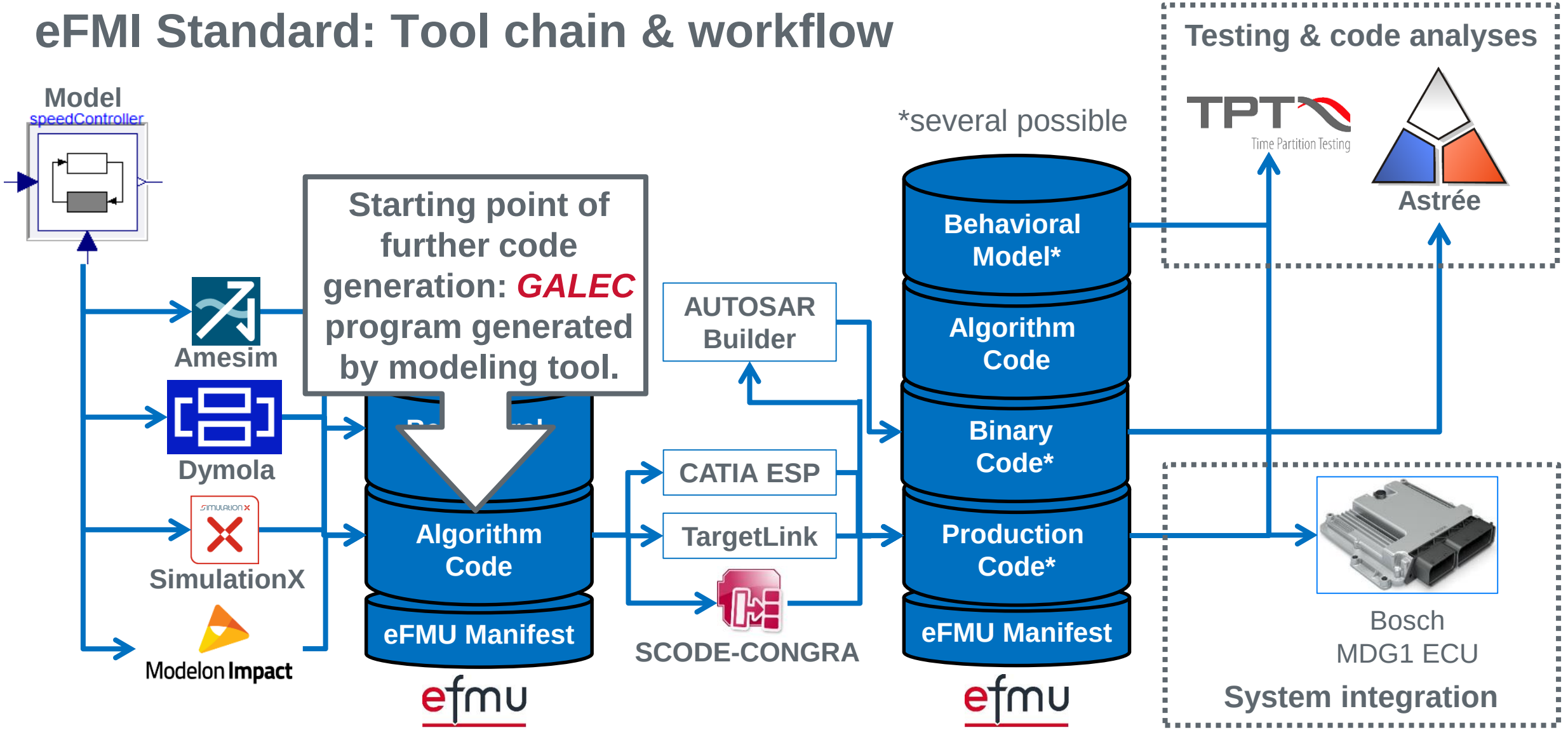
- Behavior / reference results for testing (*Behavioral Model* containers)
- Target-independent bounded algorithmic solution (*Algorithm Code* container) based on *GALEC*
- C implementations, tailored and optimized for the requirements of specific target environments (*Production Code* containers)
- Binary distributions and their „build-recipes“, ready for embedded system integration (*Binary Code* containers)

Functional Mock-up Interface for embedded systems

eFMI Standard: Tool chain & workflow



eFMI Standard: Tool chain & workflow



eFMI Standard: Algorithm Code container & GALEC language

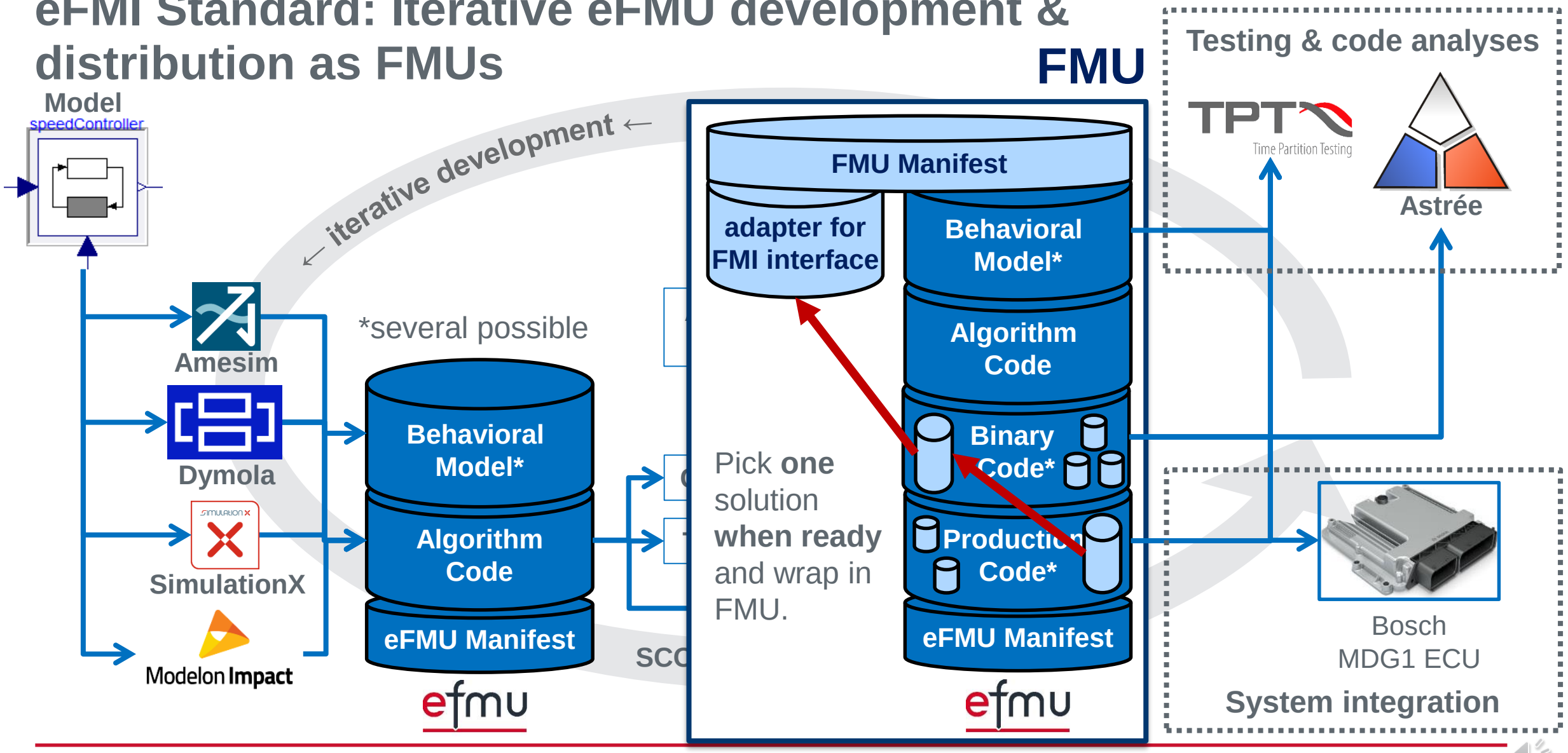
GALEC (***G***uarded ***A***lgorithmic ***L***anguage for ***E***mbedded ***C***ontrol): Intermediate representation well-suited as code generation target for modelling tools & source for embedded-code generation

- imperative / causal language of high abstraction level (e.g., multi-dimensional real arithmetic, built-in mathematical functions like sinus, cosine, interpolation 1D & 2D, solve linear equation systems etc.)
- Safe – embedded & real-time suited – semantics
  - upper bound
  - statically known sizes and safe indexing
  - well-defined & never competing side effects
- Safe floating-point numerics
  - guaranteed NaN propagation
  - saturation of ranged variables
- Ordinary control-flow integrated, strict error handling concept
  - guaranteed error signal propagation enables delayed error handling

⇒ Guards further eFMI tooling

efmi Functional Mock-up Interface for embedded systems

eFMI Standard: Iterative eFMU development & distribution as FMUs



MAP eFMI: Members

DLR

efmi Functional Mock-up Interface for embedded systems

ETAS DRIVING EMBEDDED EXCELLENCE

OpenModelica

Mercedes-Benz

BOSCH Invented for life

EFS ELEKTRONISCHE FAHRWERKSYSTEME

dSPACE

PIKETEC software engineering

Modelon

SIEMENS Ingenuity for life

DASSAULT SYSTEMES

esi

Project leader:
Christoff Bürger

DASSAULT SYSTEMES

Deputy project leader:
Hubertus Tummescheit

Modelon

https://efmi-standard.org/

MAP eFMI: Standardization processes

Apply defensive, bottom-up, prototype-based processes:

- Release of next eFMI version only when a *cross-checked tool-chain*, with tests covering all features, exists…
  - …whereas all eFMU artefacts are generated and consumed and…
  - …bad / misbehaving / to-be-rejected / evil inputs are also tested.
- Such *defensive approach* fits our domain (safety).
- The innovation/research is in the tools, not the interfaces or standard (only GALEC *might* be an exception).
- The novelty comes from actually bridging the gap from (physics) models to embedded software (making *this* really happen).

eFMI Standard version 1.0.0 release end of this year!

Alpha draft of specification already published (https://efmi-standard.org/)

MAP eFMI: Release cycle

Major new version every 2 years:

- 1st year definition of new features
  - what, why, limitations, test scenarios, prototype implementations *and*…
  - …feasibility study that a specification can achieve completeness of rules that can be reasonably implemented – ”somebody tries to put it into rules and plays the evil guy trying to break these”
- 2nd year cleanup of proposed features (no new features and ”if in doubt, feature cut”):
  - actual specification in Standard
  - cleanup and prepare release of MAP eFMI toolings/libraries
  - tool vendors cleanup their prototype implementations used in cross-checks to be ready for release
- Avoid minor standard versions
  - Such are *only* bug-fixes; we want to make sure stuff works before release → less need for bug-fixing
  - Think of versioning of formal language standards like C, C++, Scheme etc.

We are open for new members!

efmi Functional Mock-up Interface for embedded systems

DLR

ETAS
DRIVING EMBEDDED EXCELLENCE

OpenModelica

Mercedes-Benz

BOSCH
Invented for life

EFS
ELEKTRONISCHE FAHRWERKSYSTEME

dSPACE

PIKETEC
software engineering

Modelon

SIEMENS
Ingenuity for life

DASSAULT SYSTEMES

esi