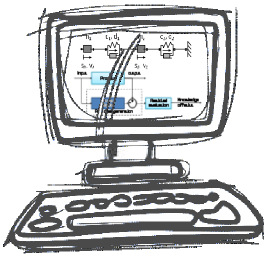
Standardizing eFMI for Embedded Systems with Physical Models in the Production Code Software

ITEA 3

Jubilee Symposium: **Future Directions of System Modeling and Simulation**

Sept. 30, 2019, Medicon Village, Lund, Sweden

EMPHYSIS



efmi

Oliver Lenord, Robert Bosch GmbH – Corporate Research
with contributions from all partners

VINNOVA AGENTSCHAP INNOVEREN & ONDERNEMEN

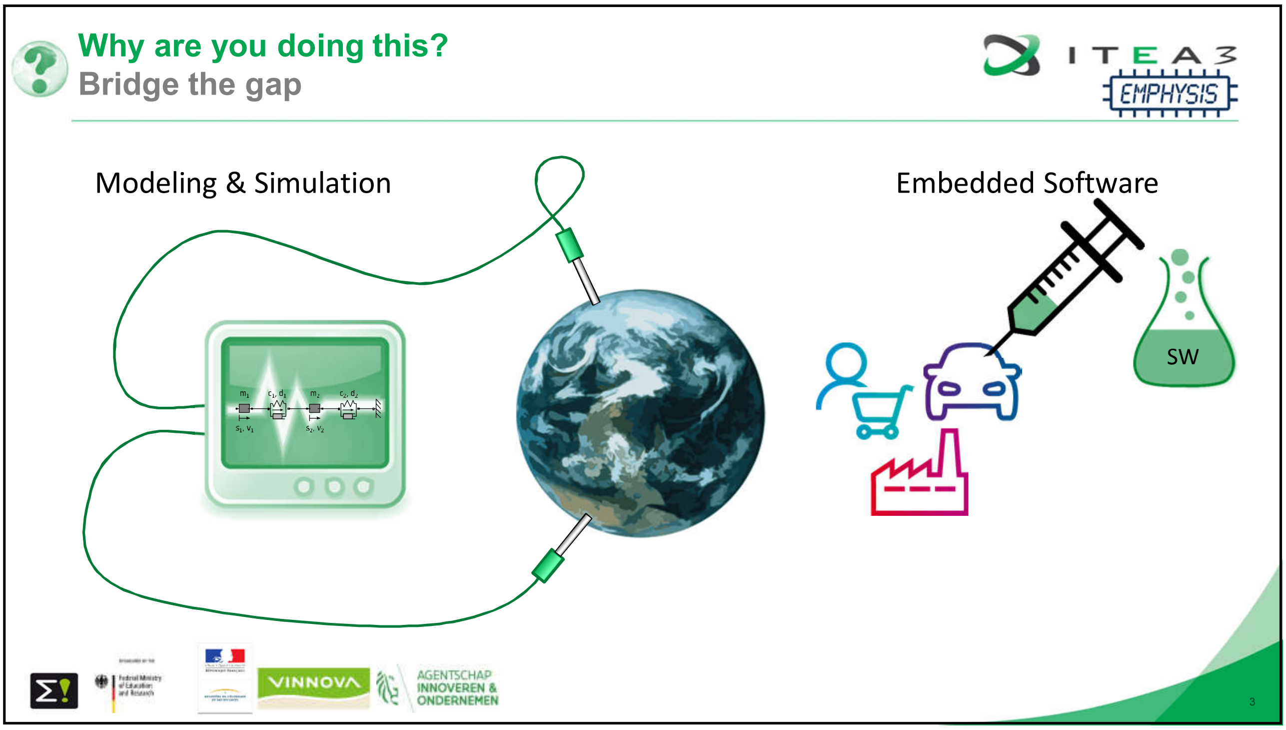
What is this all about?
The new eFMI standard

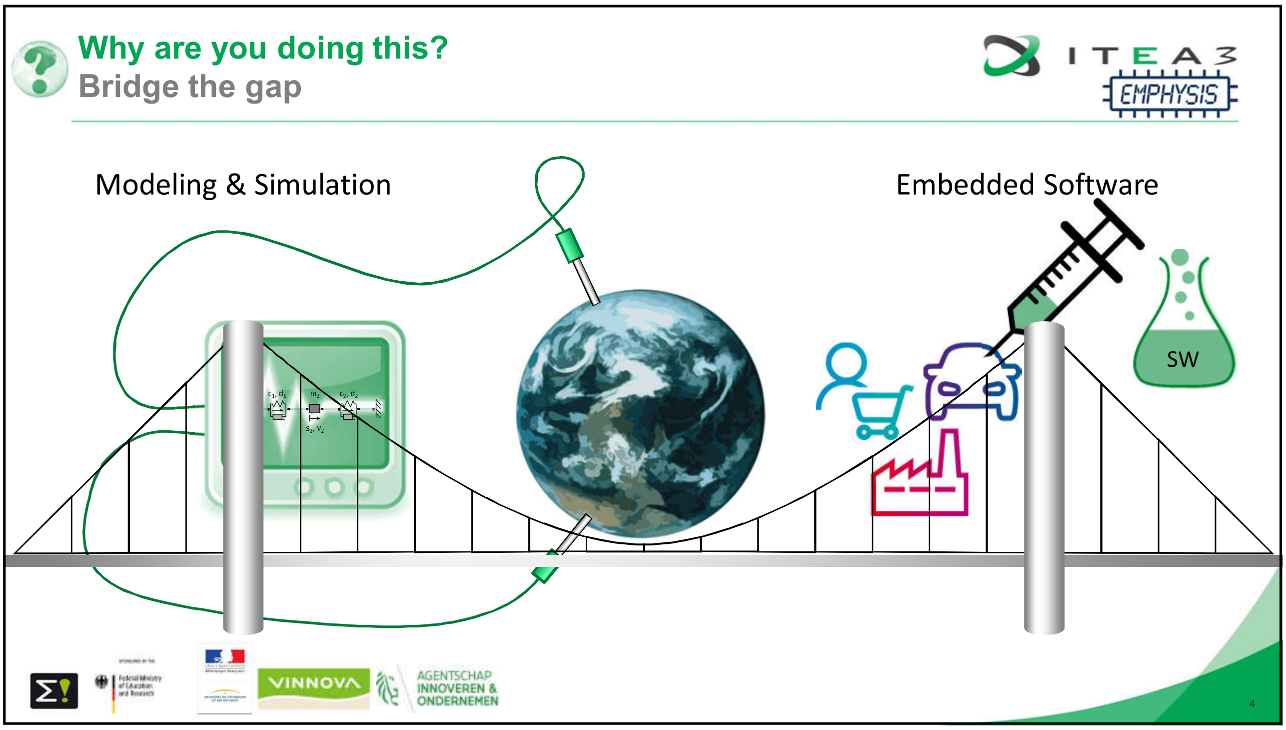
ITEA 3 EMPHYSIS

- Why are you doing this? – Purpose/Motivation
- What is new? – Problem Statement/Benefit
- How does it work? – Conceptual Idea
- How does it work in practice? – Demonstrator
- How good does it work? – Performance Metrics
- Who will use it? – Usage Scenarios
- Who supports it? – Tool Prototypes

- When can I have it? – Project Schedule
- Who is doing all this? - Acknowledgements

VINNOVA AGENTSCHAP INNOVEREN & ONDERNEMEN


Why are you doing this?
Bridge the gap
ITEA 3
EMPHYSIS
Modeling & Simulation
Embedded Software
SW
VINNOVA
AGENTSCHAP INNOVEREN & ONDERNEMEN
3

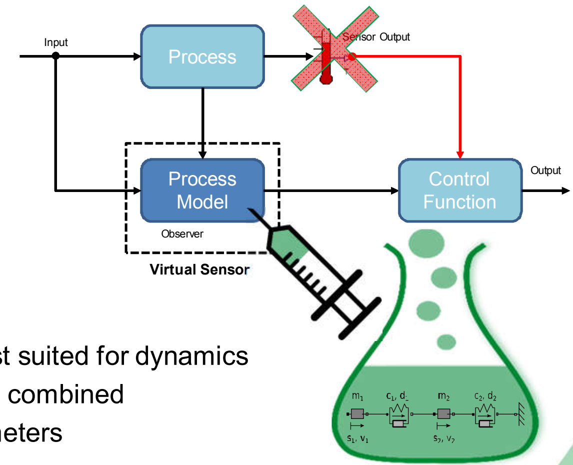

Why are you doing this?
Bridge the gap
ITEA 3
EMPHYSIS
Modeling & Simulation
Embedded Software
SW
VINNOVA
AGENTSCHAP INNOVEREN & ONDERNEMEN
4

Why are you doing this?
Physical models for embedded software

ITEA 3 EMPHYSIS

Online physical models key technology for advanced engine control software:

- virtual sensors, i.e., observers,
- model-based diagnosis,
- inverse physical models as feed forward part of control structures, and
- model predictive control.



Physical models:

- Typically described by differential equations, best suited for dynamics
- Complementary to data-based modeling, can be combined
- Reduced calibration effort due to physical parameters

Physical Model

VINNOVA AGENTSCHAP INNOVEREN & ONDERNEMEN

5

What is new?
State-of-the-art

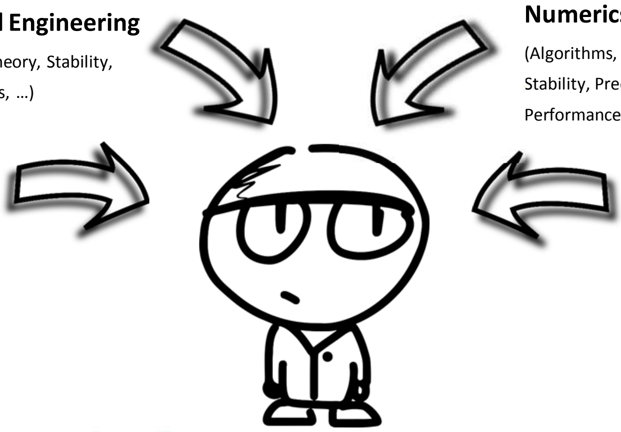
ITEA 3 EMPHYSIS

Control Engineering
(System Theory, Stability, Robustness, …)

Numerics
(Algorithms, Complexity, Stability, Precision, Realtime Performance…)

Physical Modeling
(Domain Knowledge, Physical Principles & Phenomena, System Dynamics, Model Validation, …)

ECU Software
(MISRA, ASIL, MSR, AUTOSAR, …)



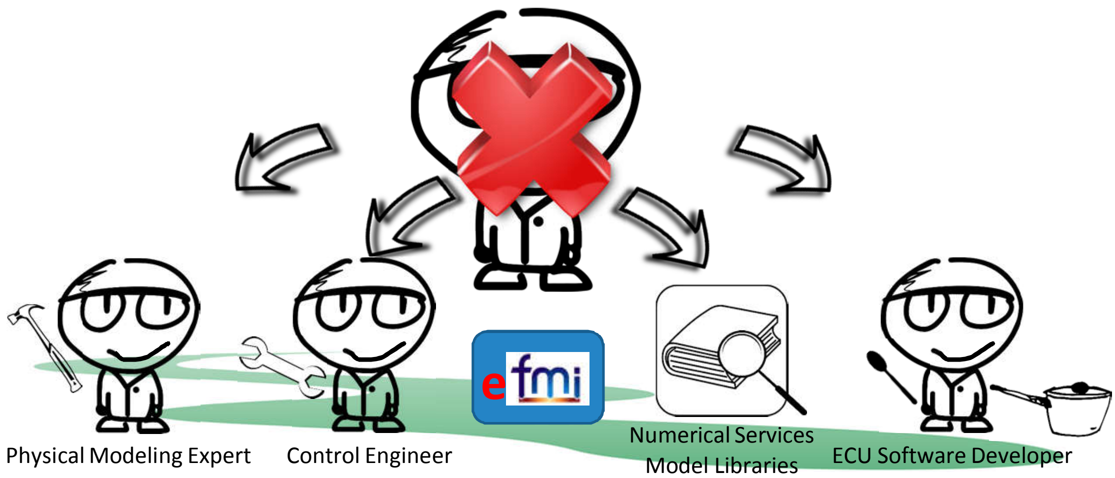
Super Hero **Function Developer**

VINNOVA AGENTSCHAP INNOVEREN & ONDERNEMEN

6

What is new?
New standard, new tool chains, new ways of collaboration

ITEA 3 EMPHYSIS



VINNOVA AGENTSCHAP INNOVEREN & ONDERNEMEN

7

What is new?
Special requirements of automotive embedded systems

ITEA 3 EMPHYSIS

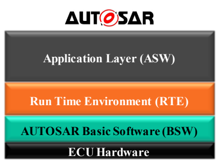
- Specialized hardware: µController with specialized cores, limitations in memory and data types (fixed-point, float)
- High safety requirements on the software
- Special coding guidelines, e.g., MISRA rules
- Special realtime operating systems (AUTOSAR-OS)
- Specialized tools and tool chains (compilers etc.)
- AUTOSAR standard defining the structure and interface of software modules, replacing proprietary solutions; support for some basic numerical functions

Today ECU software requirements are not satisfied by the FMI standard.

Bosch MDG1 ECU: current multi-core ECU

MISRA

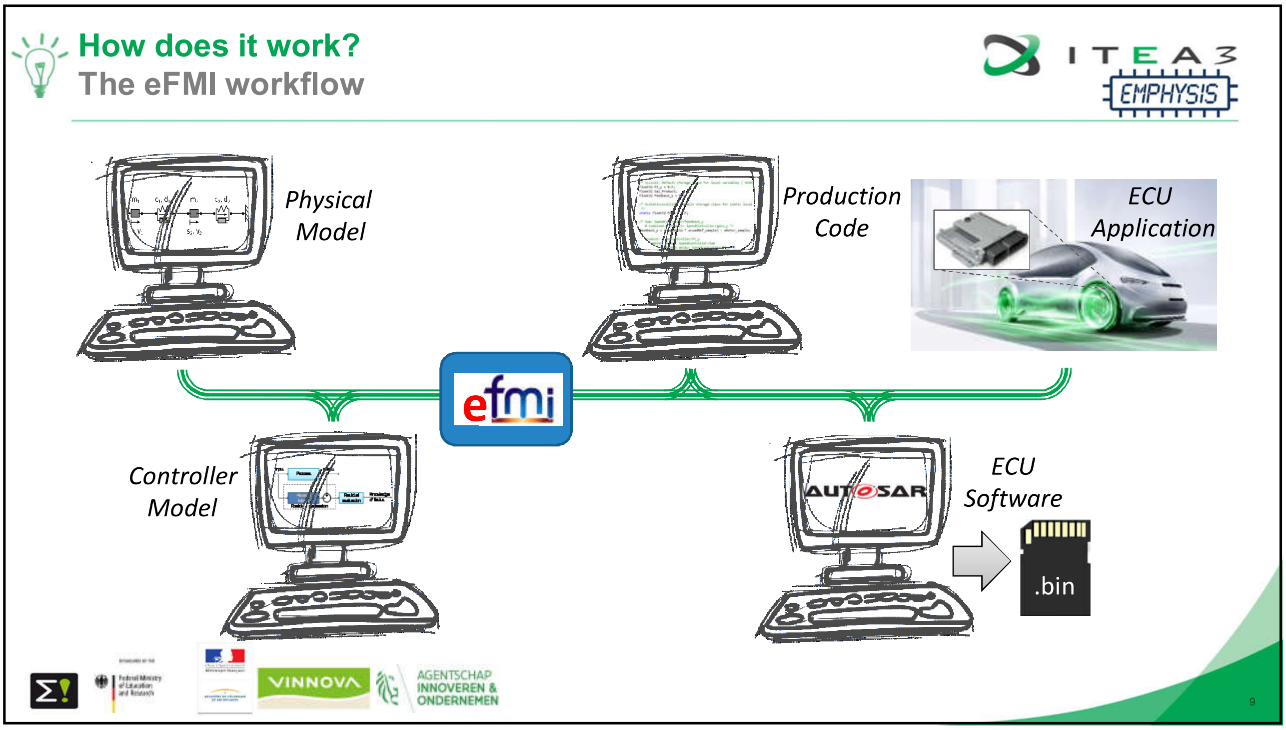
Motor Industry Software Reliability Association

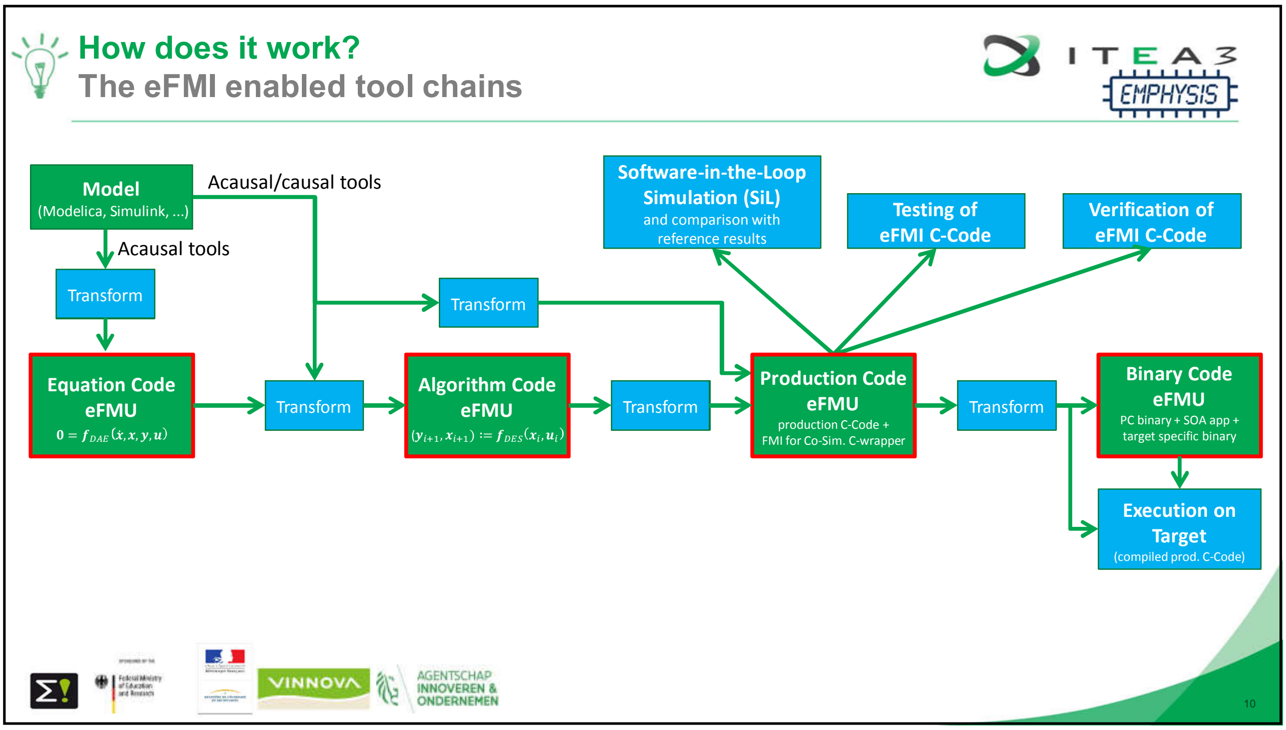


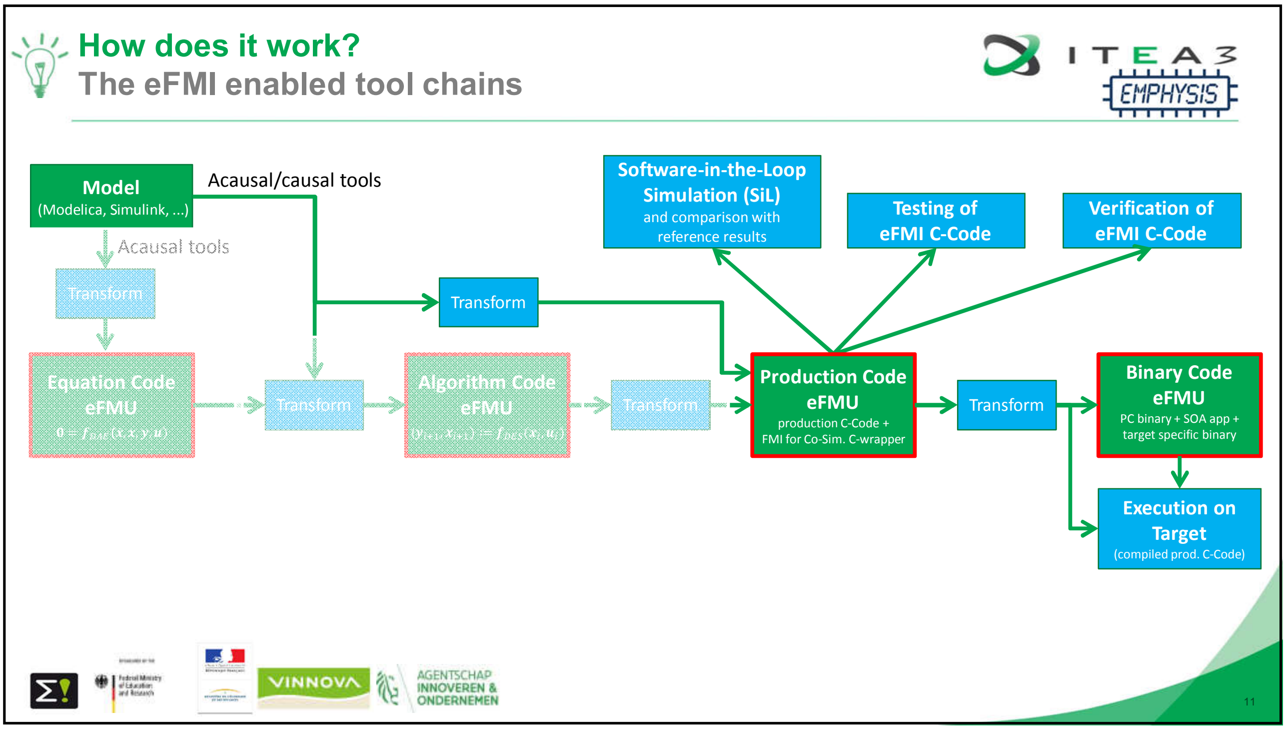
AUTOSAR architecture

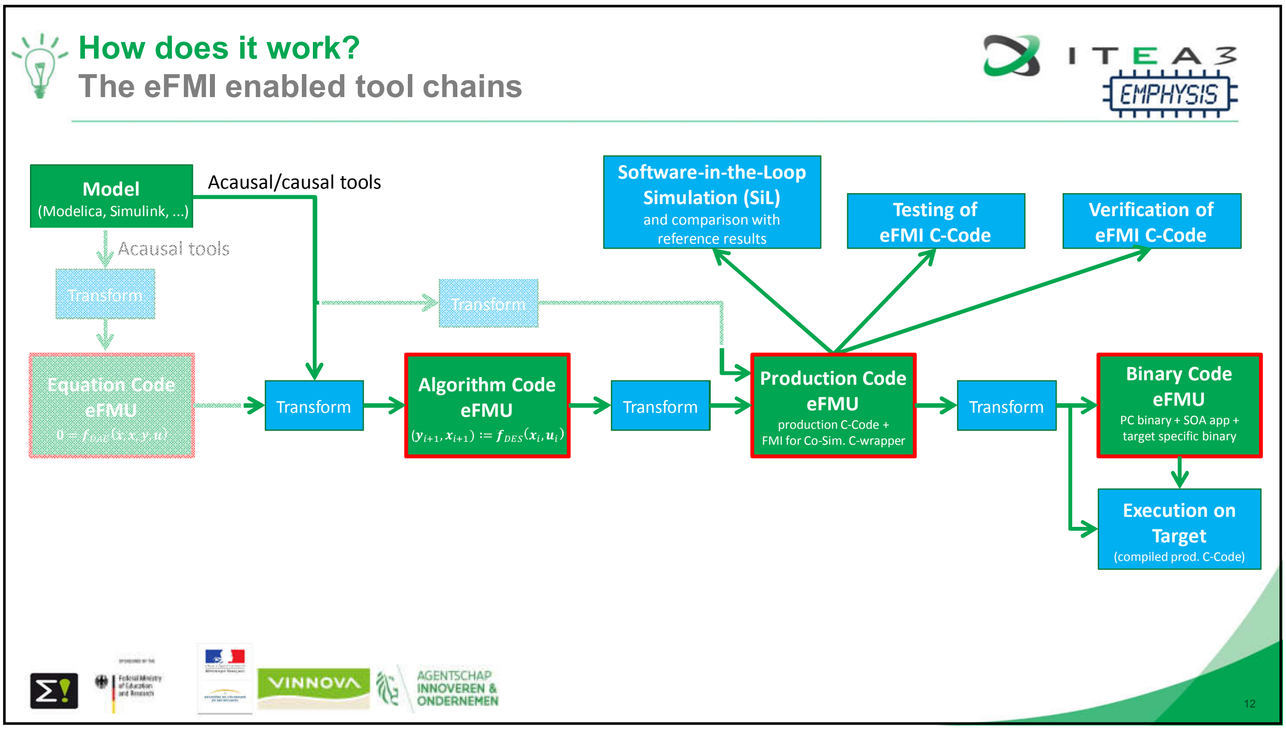
VINNOVA AGENTSCHAP INNOVEREN & ONDERNEMEN

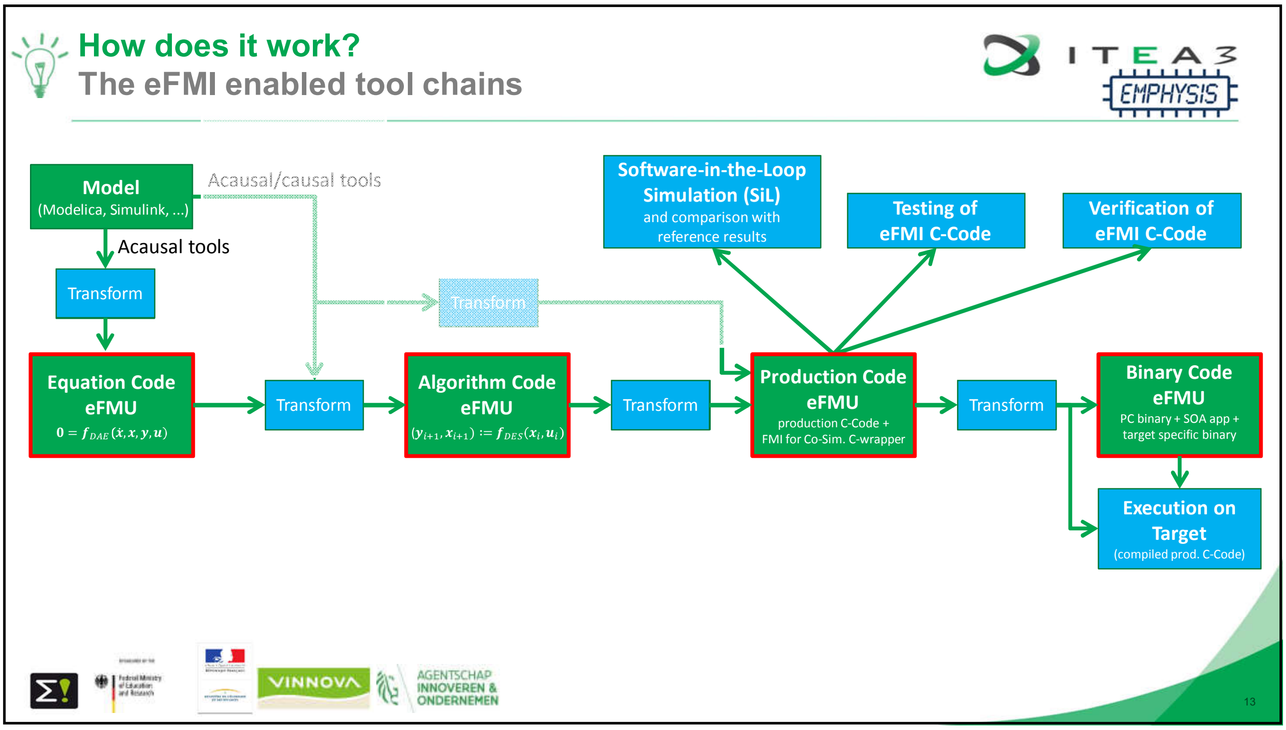
8

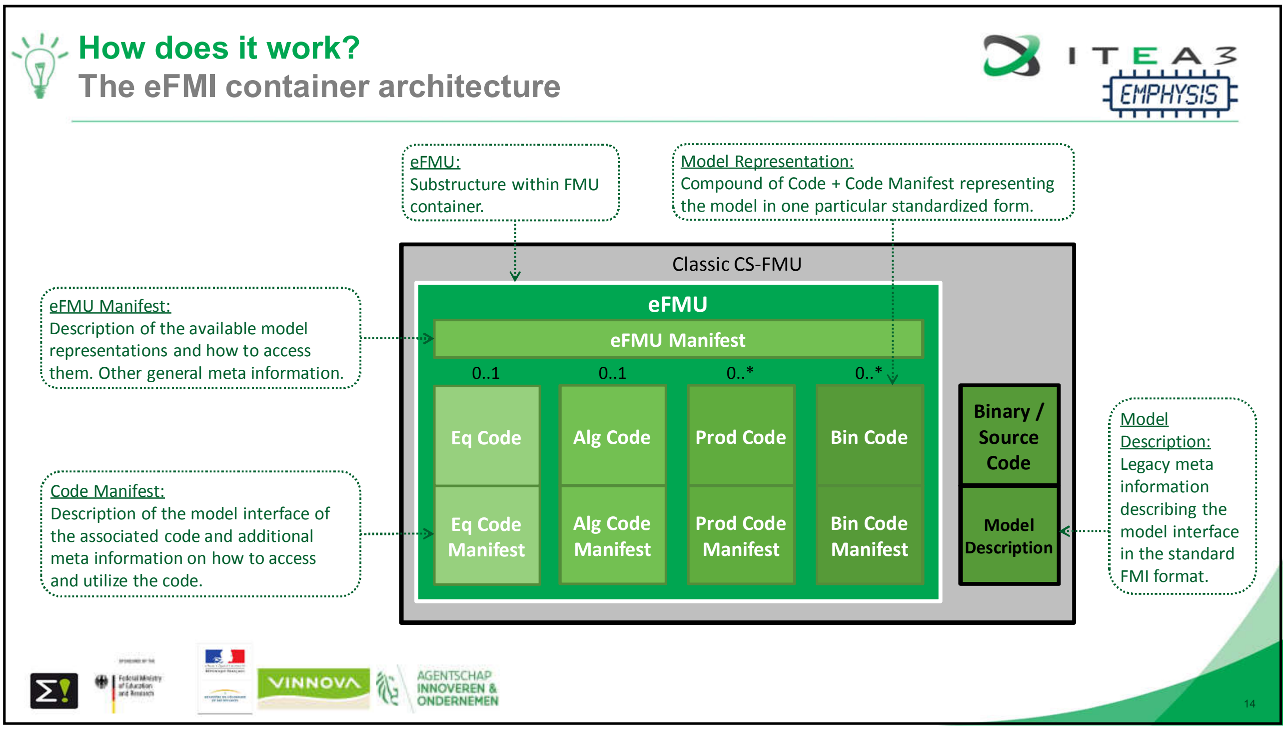

How does it work?
The eFMI workflow
ITEA 3
EMPHYSIS
Physical Model
Production Code
ECU Application
efmi
Controller Model
ECU Software
AUTOSAR
.bin
VINNOVA
AGENTSCHAP INNOVEREN & ONDERNEMEN
9


How does it work?
The eFMI enabled tool chains
ITEA 3
EMPHYSIS
Model
(Modelica, Simulink, ...)
Acausal/causal tools
Acausal tools
Transform
Equation Code eFMU
$0 = f_{DAE}(\dot{x}, x, y, u)$
Transform
Algorithm Code eFMU
$(y_{i+1}, x_{i+1}) := f_{DES}(x_i, u_i)$
Transform
Transform
Production Code eFMU
production C-Code + FMI for Co-Sim. C-wrapper
Software-in-the-Loop Simulation (SiL)
and comparison with reference results
Testing of eFMI C-Code
Verification of eFMI C-Code
Transform
Binary Code eFMU
PC binary + SOA app + target specific binary
Execution on Target
(compiled prod. C-Code)
VINNOVA
AGENTSCHAP INNOVEREN & ONDERNEMEN
10


How does it work?
The eFMI enabled tool chains
ITEA 3
EMPHYSIS
Model
(Modelica, Simulink, ...)
Acausal/causal tools
Acausal tools
Transform
Equation Code
eFMU
Transform
Algorithm Code
eFMU
Transform
Transform
Software-in-the-Loop
Simulation (SiL)
and comparison with
reference results
Testing of
eFMI C-Code
Verification of
eFMI C-Code
Production Code
eFMU
production C-Code +
FMI for Co-Sim. C-wrapper
Transform
Binary Code
eFMU
PC binary + SOA app +
target specific binary
Execution on
Target
(compiled prod. C-Code)
VINNOVA
AGENTSCHAP
INNOVEREN &
ONDERNEMEN
11


How does it work?
The eFMI enabled tool chains
ITEA 3
EMPHYSIS
Model
(Modelica, Simulink, ...)
Acausal/causal tools
Acausal tools
Transform
Equation Code
eFMU
Transform
Transform
Algorithm Code
eFMU
Transform
Software-in-the-Loop
Simulation (SiL)
and comparison with
reference results
Testing of
eFMI C-Code
Verification of
eFMI C-Code
Production Code
eFMU
production C-Code +
FMI for Co-Sim. C-wrapper
Transform
Binary Code
eFMU
PC binary + SOA app +
target specific binary
Execution on
Target
(compiled prod. C-Code)
VINNOVA
AGENTSCHAP
INNOVEREN &
ONDERNEMEN
12


How does it work?
The eFMI enabled tool chains
ITEA 3
EMPHYSIS
Model
(Modelica, Simulink, ...)
Acausal/causal tools
Acausal tools
Transform
Equation Code eFMU
$0 = f_{DAE}(\dot{x}, x, y, u)$
Transform
Transform
Algorithm Code eFMU
$(y_{i+1}, x_{i+1}) := f_{DES}(x_i, u_i)$
Transform
Production Code eFMU
production C-Code + FMI for Co-Sim. C-wrapper
Transform
Binary Code eFMU
PC binary + SOA app + target specific binary
Execution on Target
(compiled prod. C-Code)
Software-in-the-Loop Simulation (SiL)
and comparison with reference results
Testing of eFMI C-Code
Verification of eFMI C-Code
VINNOVA
AGENTSCHAP INNOVEREN & ONDERNEMEN
13


How does it work?
The eFMI container architecture
ITEA 3
EMPHYSIS
eFMU:
Substructure within FMU container.
Model Representation:
Compound of Code + Code Manifest representing the model in one particular standardized form.
eFMU Manifest:
Description of the available model representations and how to access them. Other general meta information.
Code Manifest:
Description of the model interface of the associated code and additional meta information on how to access and utilize the code.
Model Description:
Legacy meta information describing the model interface in the standard FMI format.
Classic CS-FMU
eFMU
eFMU Manifest
0..1
0..1
0..*
0..*
Eq Code
Alg Code
Prod Code
Bin Code
Eq Code Manifest
Alg Code Manifest
Prod Code Manifest
Bin Code Manifest
Binary / Source Code
Model Description
VINNOVA
AGENTSCHAP INNOVEREN & ONDERNEMEN
14

How does it work?
The eFMI Equation Code model representation

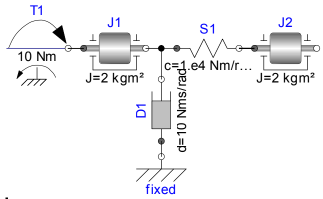
ITEA 3 EMPHYSIS

Definition:

- **Flat** unsorted set of **equations** (DAE).

Purpose:

- Capture the **mathematical essence** of a physical model.
- **Exchange** of models on equation level:
  - Enable advanced **analysis methods**, e.g. diagnosability analysis with SCODE CONGRA (ETAS).
  - Higher simulation efficiency and robustness compared to FMI based model exchange.
- Full **back-traceability** from to the original mathematical model.



Model (Modelica, Simulink, ...) → Transform → **Equation Code eFMU** $0 = f_{DAE}(\dot{x}, x, y, u)$ →

```
equation
  J1.phi=D1.phi_rel+fixed.phi0;
  S1.phi_rel=J2.phi-J1.phi;
  S1.b.tau=S1.c*S1.phi_rel-S1.c*S1.phi_rel0;
  J2.J*der(J2.w)=-S1.b.tau;
  D1.b.tau=D1.d*der(D1.phi_rel);
  J1.b.tau=S1.b.tau-D1. b.tau;
  J1.J*der(D1.w_rel)=J1. b.tau+T1.tau;
```


How does it work?
The eFMI Equation Code model representation

ITEA 3 EMPHYSIS

Specification:

- Reference to future standardized Flattened Modelica:
  - No object-orientation.
  - No algorithms.

Level of Maturity: Low

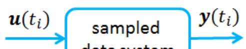
- First proposal for **Flattened Modelica** to be discussed in the Modelica Language Group at the Modelica Design Meeting (Oct. 2019).

New	flat_model
Required	constant, parameter, discrete, enumeration, record, initial, equation, der, operator, function, pure, impure, return, input, output, external, true, false, and, or, not, if, then, else, elseif, when, elsewhen, annotation, end
Dispensable	final, flow, stream, type, class, block, protected, public
Not supported	algorithm, encapsulated, expandable, for, in, loop, while, break, each

To be discussed

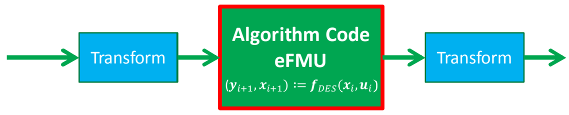
How does it work?
The eFMI Algorithm Code model representation

Definition:
- **Sampled input/output block**



$$x_{i+1} = f_x(x_i, u_i)$$
$$y_i = f_y(x_i, u_i)$$

- Sorted set of **assignments**.
- **Target independent** “logical” representation of the Production Code.




```
method DoStep
algorithm
  efmu.'D1.phi_rel' := efmu.'D1.phi_rel'+
                       efmu.T_sample* efmu.'der(D1.w_rel)';
  ...
  'J1.phi'     := 'D1.phi_rel'+'fixed.phi0';
  'S1.phi_rel' := efmu.'J2.phi' - efmu.'J1.phi';
  ...
  efmu.'der(J2.w)'     := -('S1.b.tau'/'J2.J');
  efmu.'der(D1.w_rel)' := ('J1.b.tau'+'T1.tau')/'J1.J';
end DoStep;
```

Purpose:
- Representation of simple and advanced observers, diagnosis functions, health monitoring, controllers (inverse models, model predictive control, gain scheduling, extended Kalman filter, ...)
- Reuse of the same Algorithm for different constraints, targets and applications.
- Separation of concerns: Symbolic transformation vs. embedded code generation.

How does it work?
The eFMI Algorithm Code model representation

Specification:
- Standardized small subset of Modelica:
  - No equations, No inheritance.
  - Arrays with literal dimensions + operations, no dynamic memory allocation.
  - Statically guaranteed array access.
- with extensions:
  - Methods (`DoStep`, `Initialize`,…)
  - Includes discretized integrator, such as explicit Euler or linear implicit Euler
  - Built-in functions for sin, cos, tables (1D, 2D, 3D), „solve linear equation system“, …
  - Error handling.
  - …

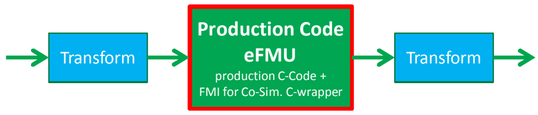
Level of Maturity: Medium-High
- Released first draft being used for on-going tool prototypes.
- Some simplifying assumptions to speed-up prototype development.

How does it work?

The eFMI Production Code model representation

Definition:

- C code, compliant with coding standards (e.g. MISRA)
- **Co-existing** generic or optimized code for **specific** architectures (e.g. AUTOSAR) or targets.




```
void DoStep(ctx* eFMI){
  eFMI->D1_phi_rel=eFMI->D1_phi_rel+eFMI->...
  ...
  _J1_phi=_D1_phi_rel+_fixed_phi0;
  _S1_phi_rel=_J2_phi-_J1_phi;
  _S1_b_tau=_S1_c*_S1_phi_rel-_S1_c*_S1_phi_rel0;
  _der__J2_w=-(_S1_b_tau/_J2_J);
  _D1_b_tau=_D1_d*_der__D1_phi_rel;
  _J1_b_tau=_S1_b_tau-_D1_b_tau;
  _der__D1_w_rel=(_J1_b_tau+_T1_tau)/_J1_J;
}
```

Purpose:

- Best performance for dedicated target.
- Optimized resource demand (memory, CPU time).
- Seamless integration in ECU environment (no wrapper).
- Integration in Software-in-the-Loop testing tools.
- Enable code verification and compliance checks.

How does it work?

The eFMI Production Code model representation

Specification:

- No standardized API.
- Existence of the methods must be guaranteed.
- Interface and structure of the functions as described in the manifest file.
- May contain target specific code (e.g. assembler code).

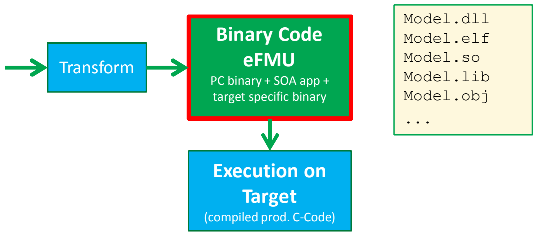
Level of Maturity: High

- Released first draft being used for on-going tool prototypes.

How does it work?
The eFMI Binary Code model representation

Definition:

- Target specific object code (e.g. PC, ECU, Service oriented Architecture (SOA) applications)



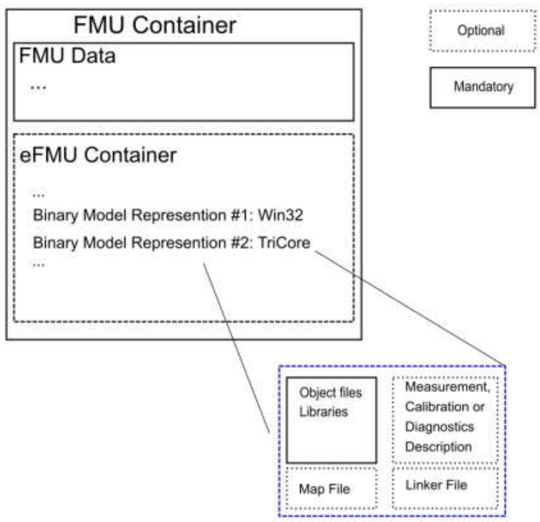
Purpose:

- Seamless integration with other ECU software to build an image for a specific ECU.
- Separation in dedicated modules.
- Protection of IP.
- Protection of integrity.
- Enable execution of the **exact same code** on a test platform.

How does it work?
The eFMI Binary Code model representation

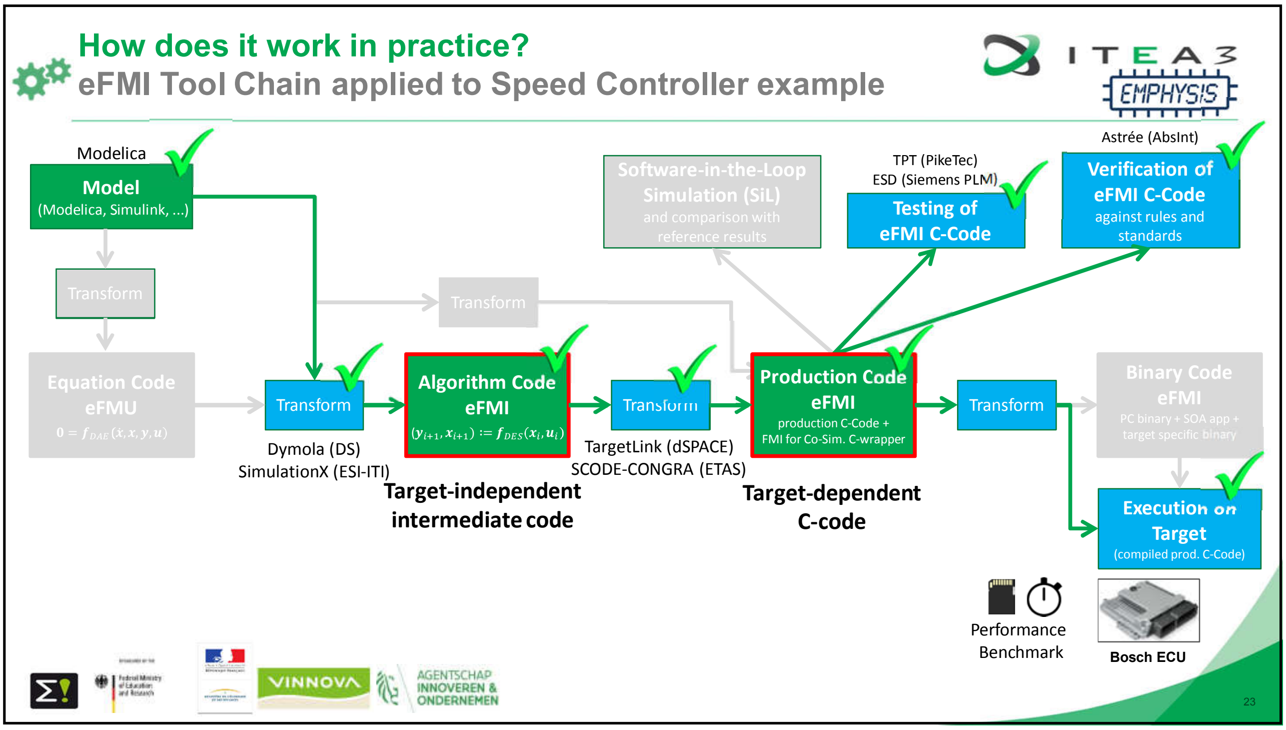
Specification:

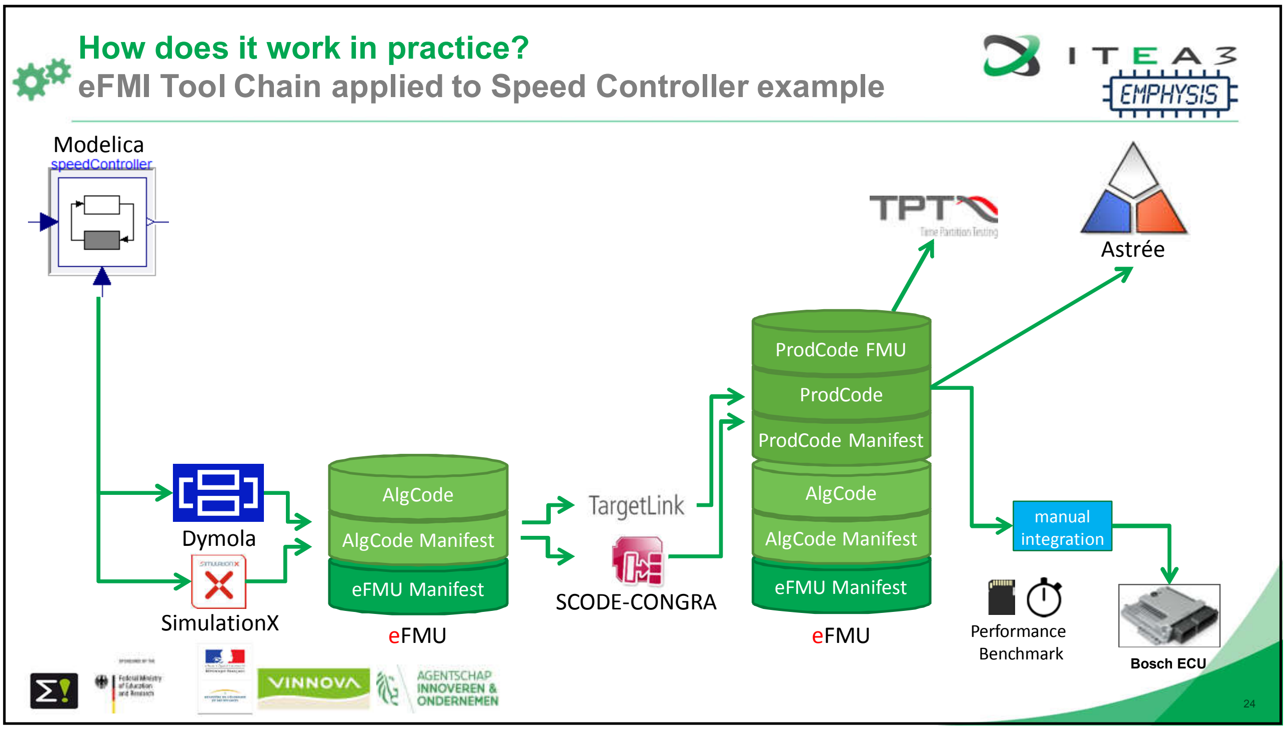
- Binary object files and libraries, e.g. ELF files.
- Measurement, Calibration or Diagnostics Description files, e.g. A2L files
- Map file
- Linker file

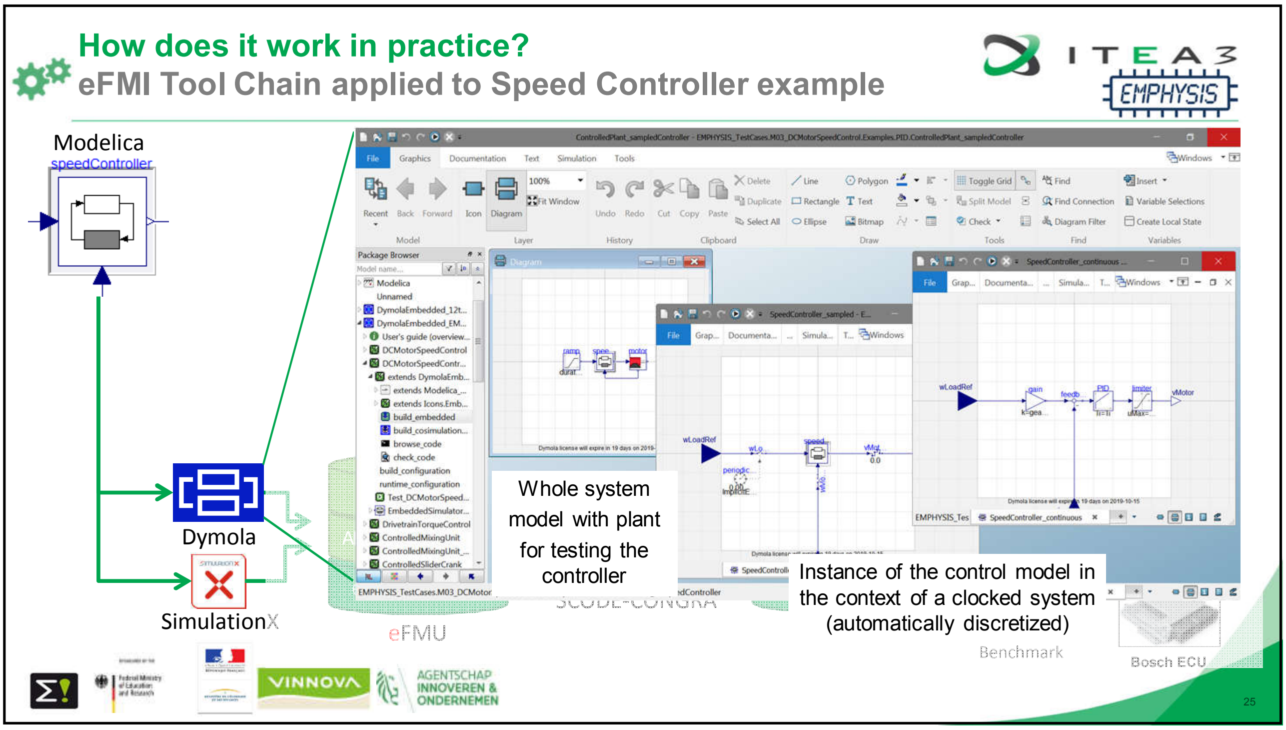


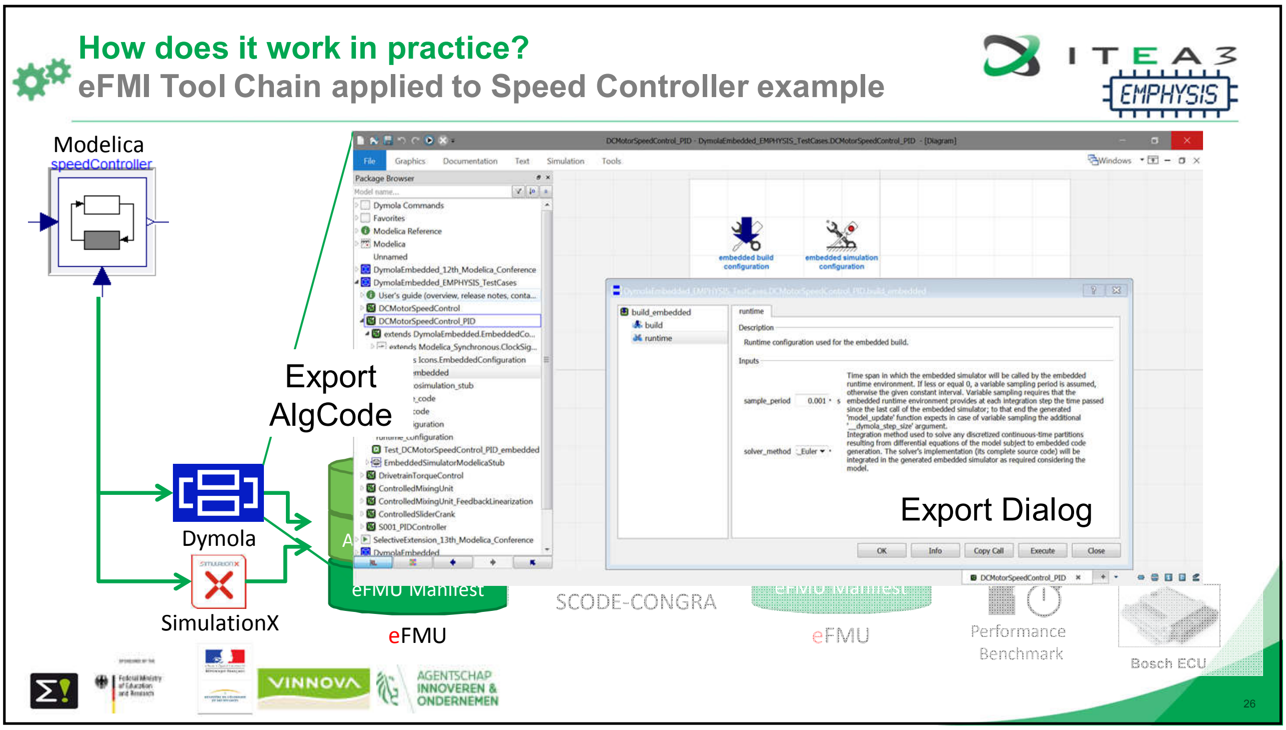
Level of Maturity: medium

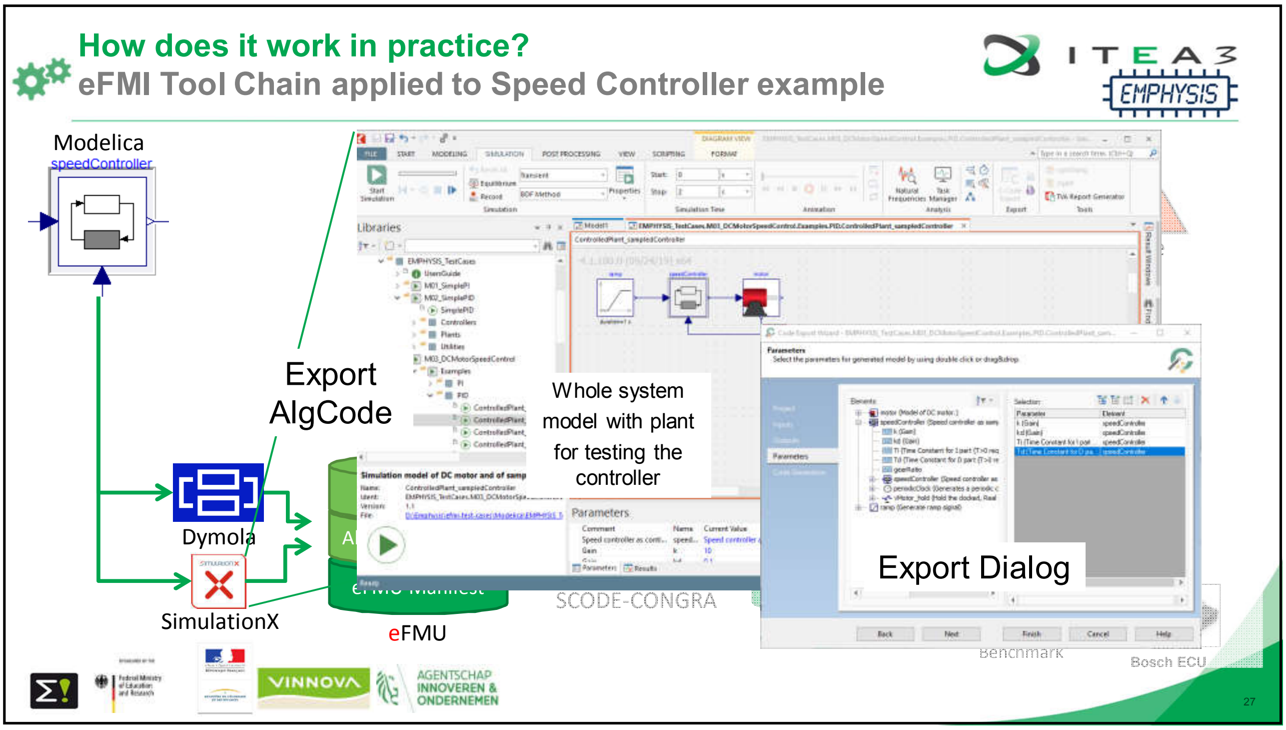
- Released first draft being used for on-going tool prototypes.

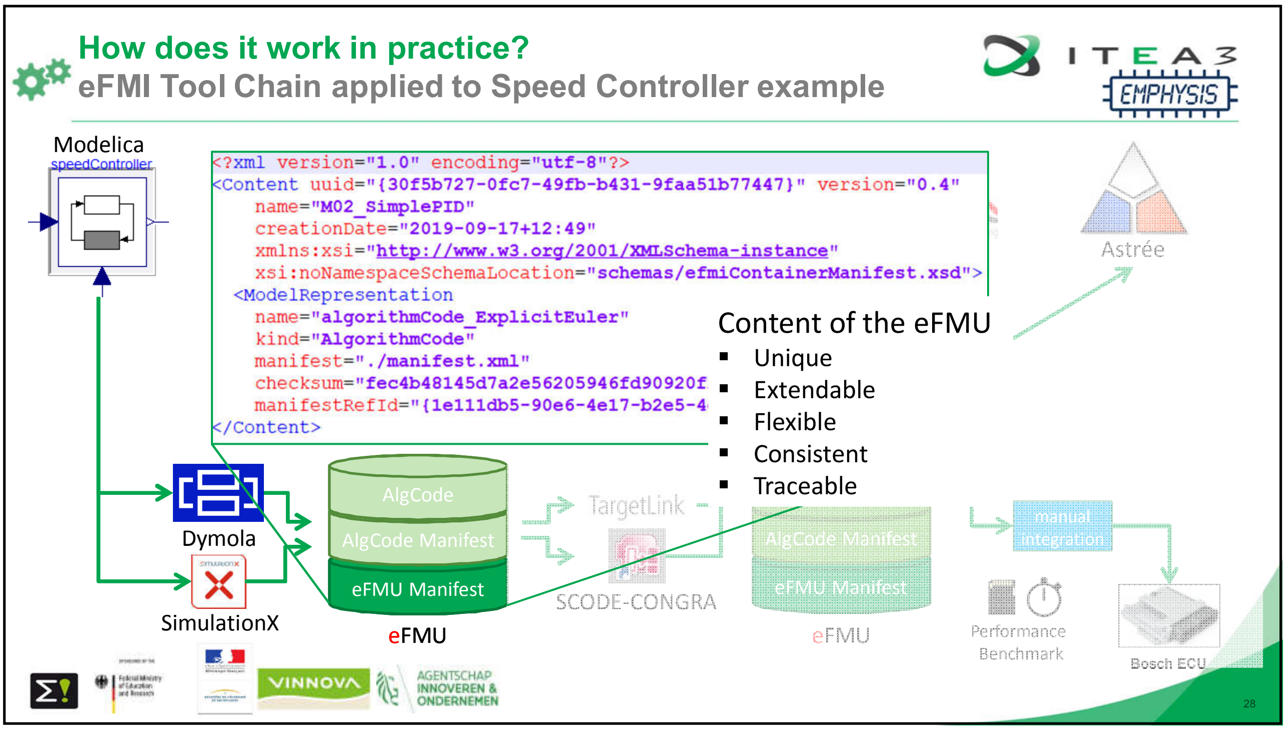

How does it work in practice?
eFMI Tool Chain applied to Speed Controller example
ITEA3
EMPHYSIS
Modelica
Model
(Modelica, Simulink, ...)
Transform
Equation Code
eFMU
$0 = f_{DAE}(\dot{x}, x, y, u)$
Transform
Dymola (DS)
SimulationX (ESI-ITI)
Transform
Algorithm Code
eFMI
$(y_{i+1}, x_{i+1}) := f_{DES}(x_i, u_i)$
Target-independent
intermediate code
Transform
TargetLink (dSPACE)
SCODE-CONGRA (ETAS)
Software-in-the-Loop
Simulation (SiL)
and comparison with
reference results
TPT (PikeTec)
ESD (Siemens PLM)
Testing of
eFMI C-Code
Astrée (AbsInt)
Verification of
eFMI C-Code
against rules and
standards
Production Code
eFMI
production C-Code +
FMI for Co-Sim. C-wrapper
Target-dependent
C-code
Transform
Binary Code
eFMI
PC binary + SOA app +
target specific binary
Execution on
Target
(compiled prod. C-Code)
Performance
Benchmark
Bosch ECU
VINNOVA
AGENTSCHAP
INNOVEREN &
ONDERNEMEN
23

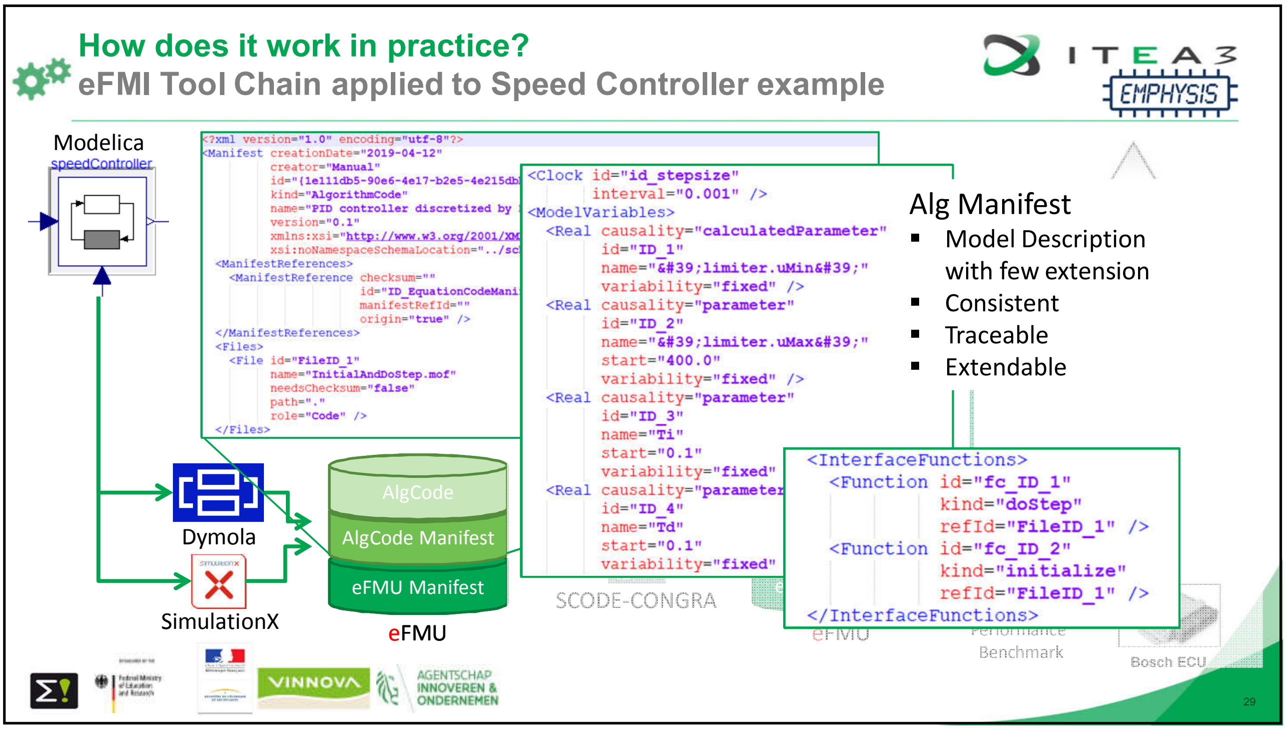

How does it work in practice?
eFMI Tool Chain applied to Speed Controller example
ITEA3
EMPHYSIS
Modelica
speedController
Dymola
SimulationX
AlgCode
AlgCode Manifest
eFMU Manifest
eFMU
TargetLink
SCODE-CONGRA
ProdCode FMU
ProdCode
ProdCode Manifest
AlgCode
AlgCode Manifest
eFMU Manifest
eFMU
TPT
Astrée
manual
integration
Performance
Benchmark
Bosch ECU
VINNOVA
AGENTSCHAP
INNOVEREN &
ONDERNEMEN
24

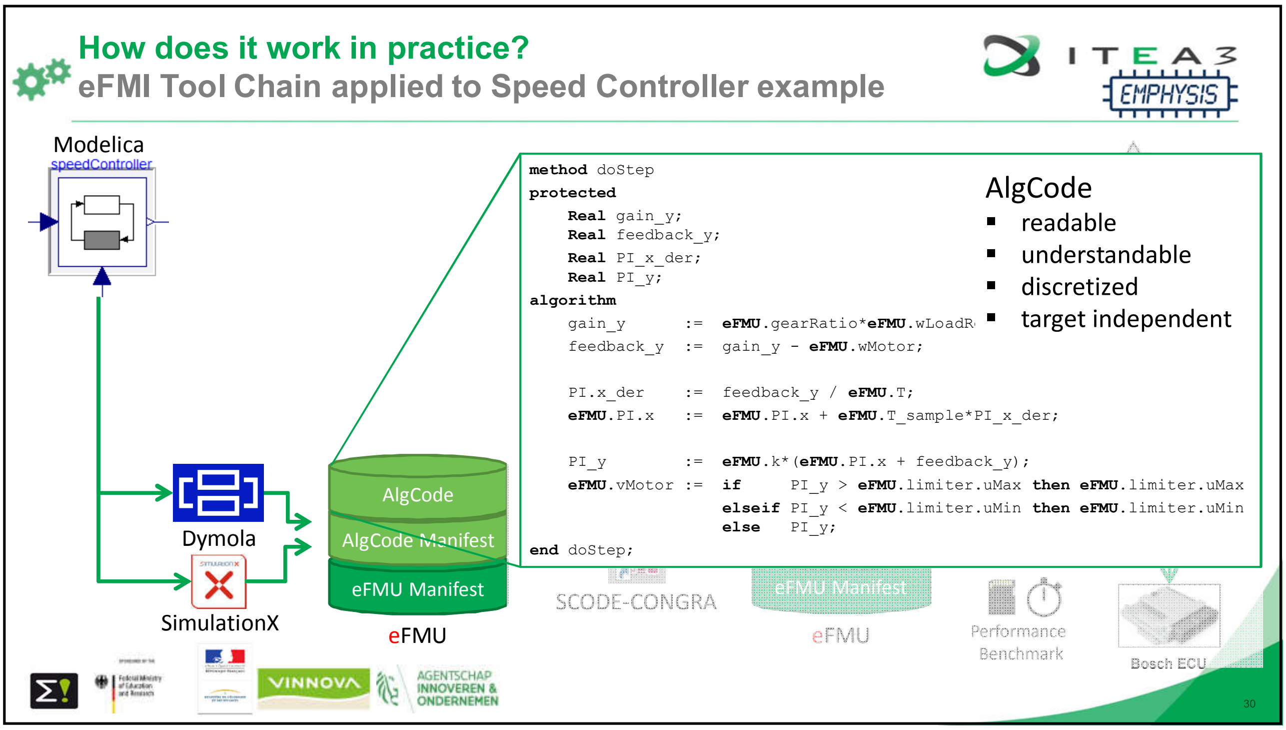

How does it work in practice?
eFMI Tool Chain applied to Speed Controller example
ITEA3
EMPHYSIS
Modelica
speedController
Dymola
SimulationX
Whole system model with plant for testing the controller
Instance of the control model in the context of a clocked system (automatically discretized)
eFMU
Benchmark
Bosch ECU
VINNOVA
AGENTSCHAP INNOVEREN & ONDERNEMEN
25

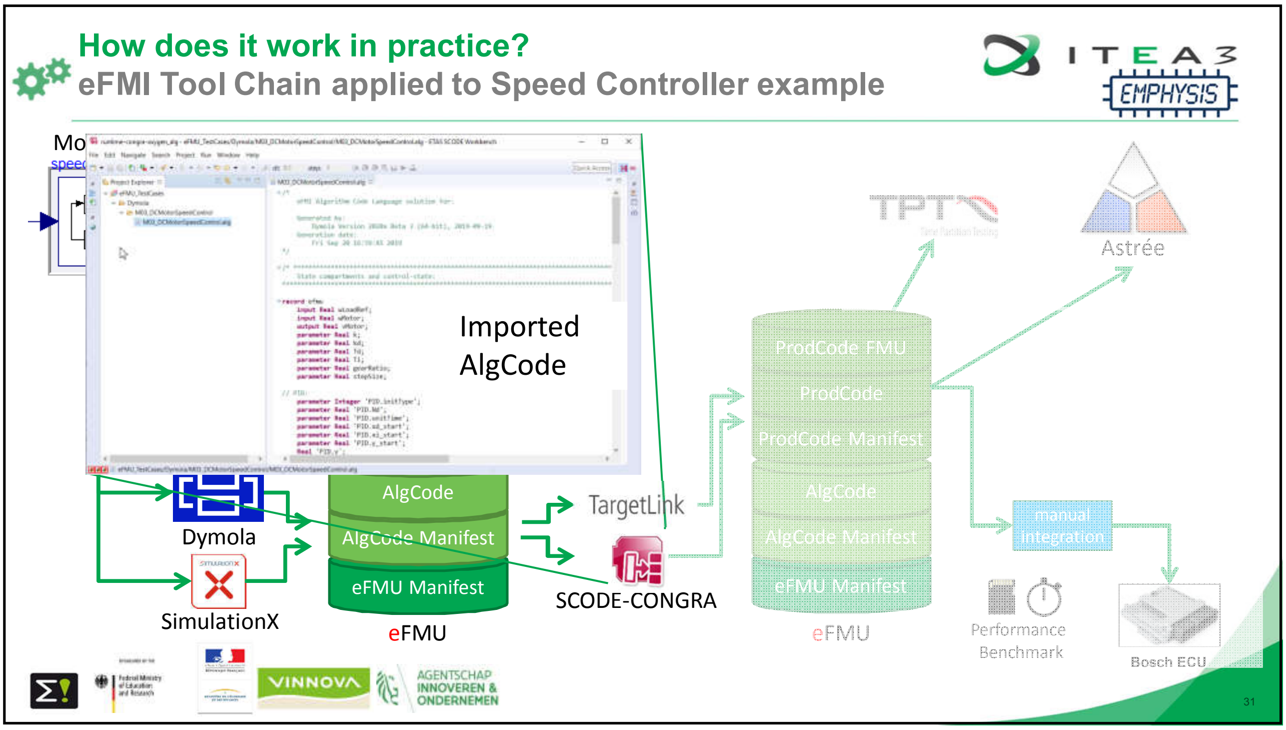

How does it work in practice?
eFMI Tool Chain applied to Speed Controller example
ITEA3
EMPHYSIS
Modelica
speedController
Export AlgCode
Dymola
SimulationX
eFMU Manifest
eFMU
SCODE-CONGRA
Export Dialog
Performance
Benchmark
Bosch ECU
VINNOVA
AGENTSCHAP INNOVEREN & ONDERNEMEN
26


How does it work in practice?
eFMI Tool Chain applied to Speed Controller example
ITEA 3
EMPHYSIS
Modelica
speedController
Export
AlgCode
Whole system
model with plant
for testing the
controller
Dymola
SimulationX
eFMU
SCODE-CONGRA
Export Dialog
Benchmark
Bosch ECU
VINNOVA
AGENTSCHAP
INNOVEREN &
ONDERNEMEN
27

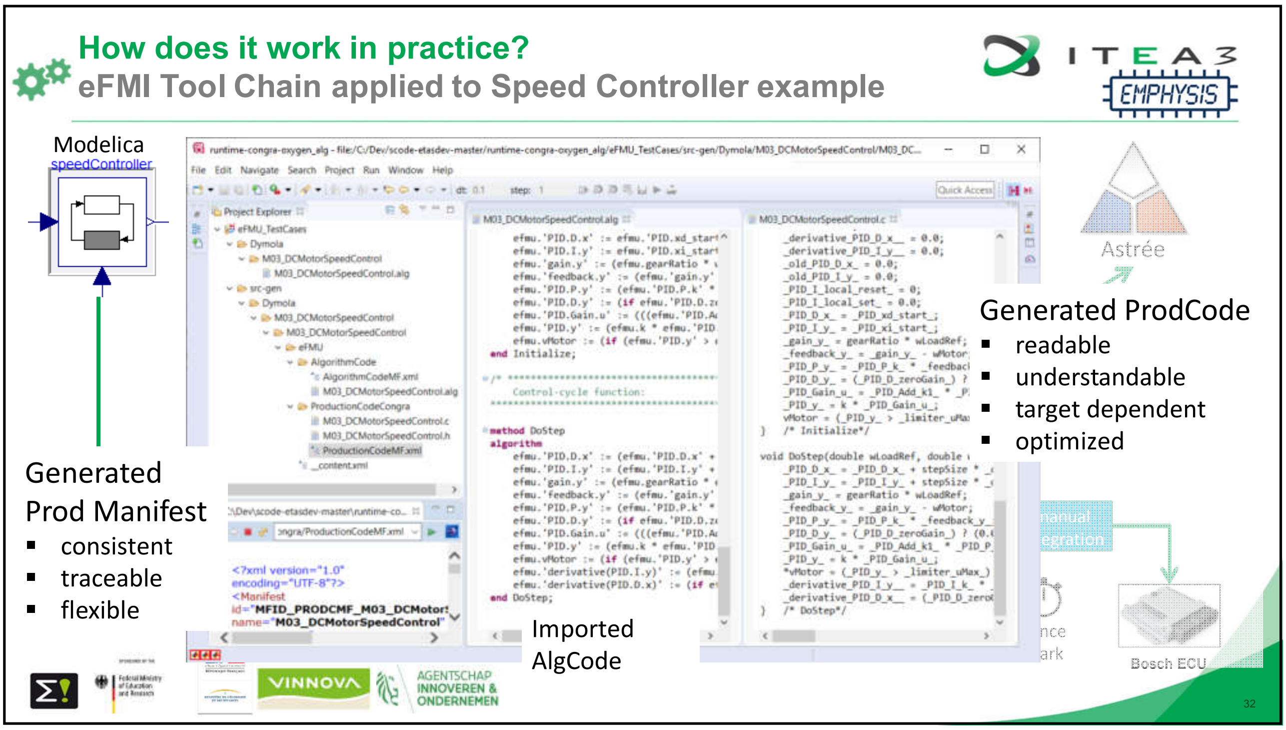

How does it work in practice?
eFMI Tool Chain applied to Speed Controller example
ITEA 3
EMPHYSIS
Modelica
speedController
<?xml version="1.0" encoding="utf-8"?>
<Content uuid="{30f5b727-0fc7-49fb-b431-9faa51b77447}" version="0.4"
name="M02_SimplePID"
creationDate="2019-09-17+12:49"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="schemas/efmiContainerManifest.xsd">
<ModelRepresentation
name="algorithmCode_ExplicitEuler"
kind="AlgorithmCode"
manifest="./manifest.xml"
checksum="fec4b48145d7a2e56205946fd90920f
manifestRefId="{1e111db5-90e6-4e17-b2e5-4
</Content>
Content of the eFMU
Unique
Extendable
Flexible
Consistent
Traceable
Astrée
AlgCode
AlgCode Manifest
eFMU Manifest
Dymola
SimulationX
eFMU
TargetLink
SCODE-CONGRA
AlgCode Manifest
eFMU Manifest
eFMU
manual
integration
Performance
Benchmark
Bosch ECU
VINNOVA
AGENTSCHAP
INNOVEREN &
ONDERNEMEN
28


How does it work in practice?
eFMI Tool Chain applied to Speed Controller example
ITEA3
EMPHYSIS
Modelica
speedController
Alg Manifest
Model Description with few extension
Consistent
Traceable
Extendable
<Clock id="id_stepsize"
interval="0.001" />
<ModelVariables>
<Real causality="calculatedParameter"
id="ID_1"
name="'limiter.uMin'"
variability="fixed" />
<Real causality="parameter"
id="ID_2"
name="'limiter.uMax'"
start="400.0"
variability="fixed" />
<Real causality="parameter"
id="ID_3"
name="Ti"
start="0.1"
variability="fixed"
<Real causality="parameter"
id="ID_4"
name="Td"
start="0.1"
variability="fixed"
<InterfaceFunctions>
<Function id="fc_ID_1"
kind="doStep"
refId="FileID_1" />
<Function id="fc_ID_2"
kind="initialize"
refId="FileID_1" />
</InterfaceFunctions>
Dymola
SimulationX
AlgCode
AlgCode Manifest
eFMU Manifest
eFMU
SCODE-CONGRA
Benchmark
Bosch ECU
VINNOVA
AGENTSCHAP INNOVEREN & ONDERNEMEN
29

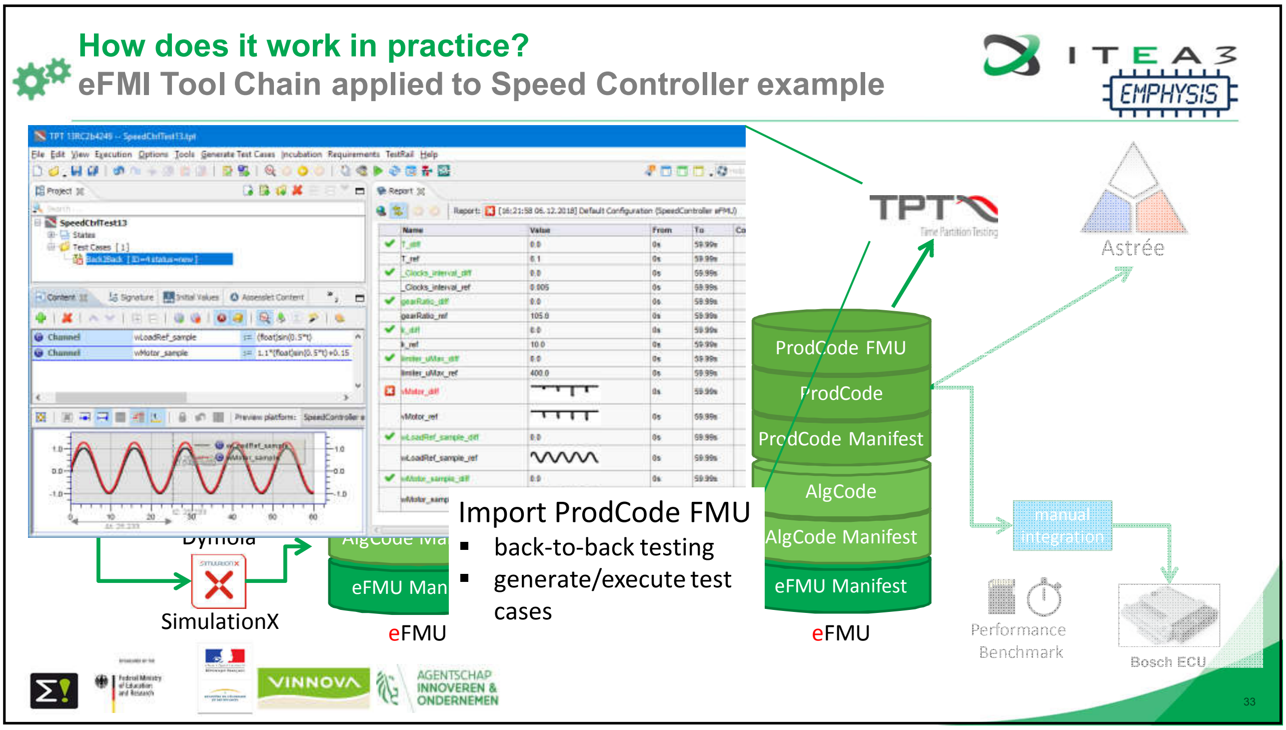

How does it work in practice?
eFMI Tool Chain applied to Speed Controller example
ITEA3
EMPHYSIS
Modelica
speedController
method doStep
protected
Real gain_y;
Real feedback_y;
Real PI_x_der;
Real PI_y;
algorithm
gain_y := eFMU.gearRatio*eFMU.wLoadR
feedback_y := gain_y - eFMU.wMotor;
PI.x_der := feedback_y / eFMU.T;
eFMU.PI.x := eFMU.PI.x + eFMU.T_sample*PI_x_der;
PI_y := eFMU.k*(eFMU.PI.x + feedback_y);
eFMU.vMotor := if PI_y > eFMU.limiter.uMax then eFMU.limiter.uMax
elseif PI_y < eFMU.limiter.uMin then eFMU.limiter.uMin
else PI_y;
end doStep;
AlgCode
readable
understandable
discretized
target independent
Dymola
SimulationX
AlgCode
AlgCode Manifest
eFMU Manifest
eFMU
SCODE-CONGRA
Performance Benchmark
Bosch ECU
VINNOVA
AGENTSCHAP INNOVEREN & ONDERNEMEN
30

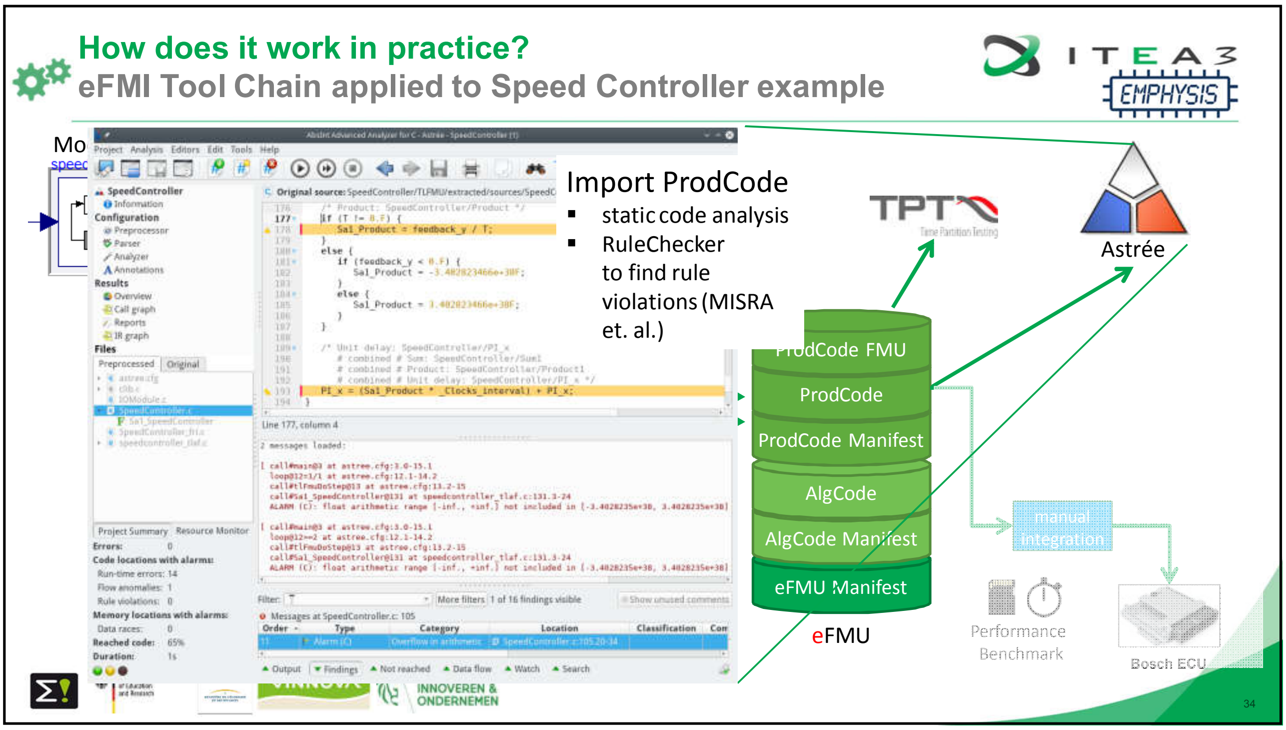


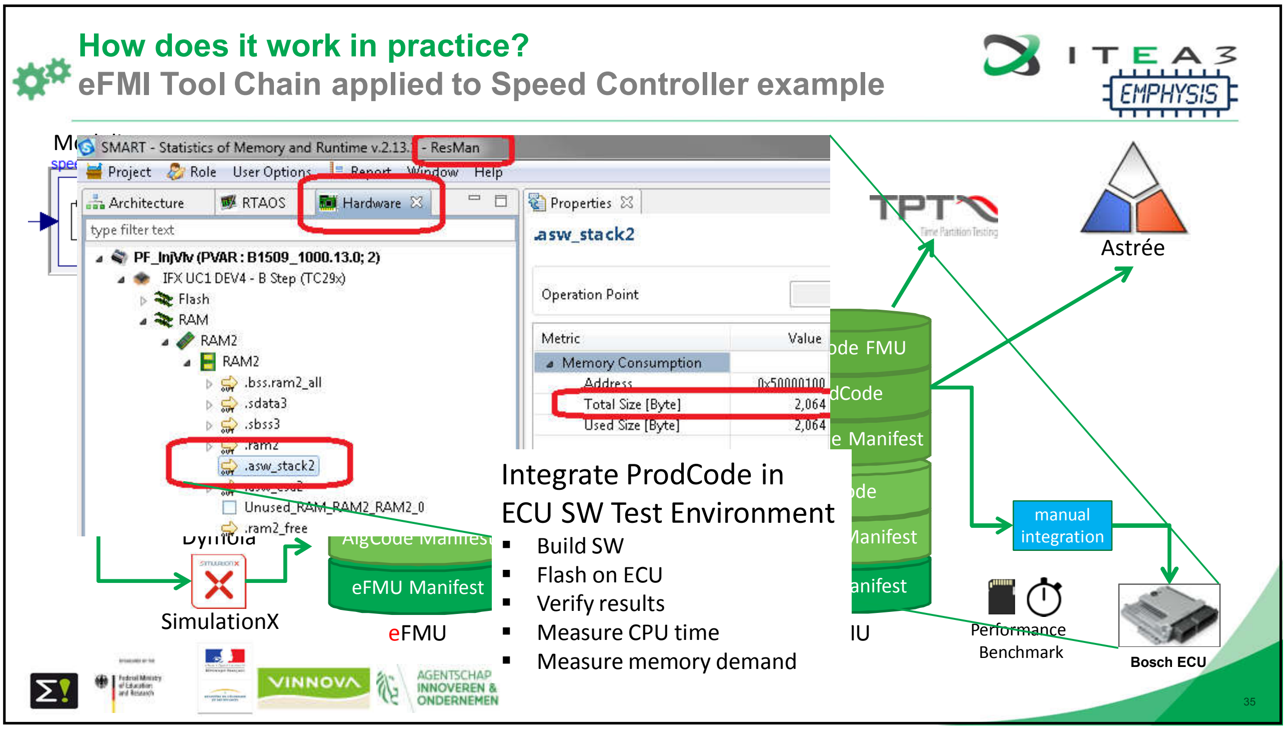
How does it work in practice?
eFMI Tool Chain applied to Speed Controller example
ITEA 3
EMPHYSIS
Imported
AlgCode
TPT
Astrée
ProdCode FMU
ProdCode
ProdCode Manifest
AlgCode
AlgCode Manifest
eFMU Manifest
eFMU
Dymola
SimulationX
AlgCode
AlgCode Manifest
eFMU Manifest
eFMU
TargetLink
SCODE-CONGRA
manual
integration
Performance
Benchmark
Bosch ECU
VINNOVA
AGENTSCHAP
INNOVEREN &
ONDERNEMEN
31



How does it work in practice?
eFMI Tool Chain applied to Speed Controller example
ITEA 3
EMPHYSIS
Modelica
speedController
Astrée
Generated ProdCode
readable
understandable
target dependent
optimized
Generated
Prod Manifest
consistent
traceable
flexible
Imported
AlgCode
Bosch ECU
VINNOVA
AGENTSCHAP
INNOVEREN &
ONDERNEMEN
32


How does it work in practice?
eFMI Tool Chain applied to Speed Controller example
ITEA3
EMPHYSIS
TPT
Time Partition Testing
Astrée
ProdCode FMU
ProdCode
ProdCode Manifest
AlgCode
AlgCode Manifest
eFMU Manifest
eFMU
Import ProdCode FMU
back-to-back testing
generate/execute test cases
manual integration
Performance Benchmark
Bosch ECU
SimulationX
Dymola
eFMU
VINNOVA
AGENTSCHAP INNOVEREN & ONDERNEMEN
33

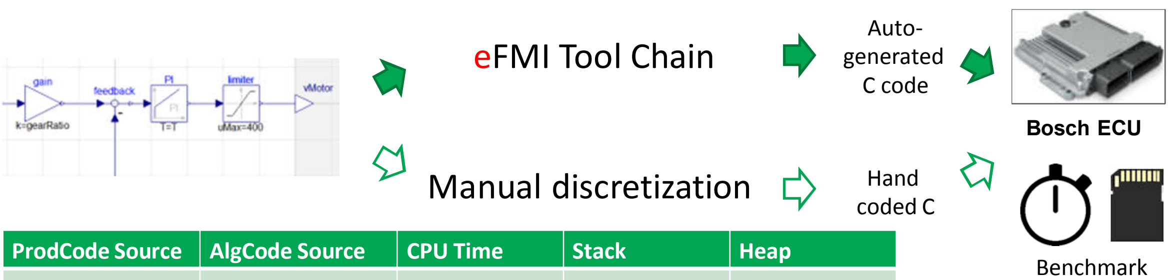

How does it work in practice?
eFMI Tool Chain applied to Speed Controller example
ITEA3
EMPHYSIS
Import ProdCode
static code analysis
RuleChecker to find rule violations (MISRA et. al.)
TPT
Time Partition Testing
Astrée
ProdCode FMU
ProdCode
ProdCode Manifest
AlgCode
AlgCode Manifest
eFMU Manifest
eFMU
manual integration
Performance Benchmark
Bosch ECU
INNOVEREN & ONDERNEMEN
34



How good does it work?

Performance benchmark

#	Test Case Name	Multi-Dim Maps	Large Maps	Large Matrices	Sparse Matrices	Nonlinear	Large Number of States	Compact Code	Stiff	DAE
M03	SpeedController	N	N	N	N	Y	N	N	N	N



ProdCode Source	AlgCode Source	CPU Time	Stack	Heap
Hand coded				
TargetLink				
SCODE-CONGRA				

First measurements available but not yet verified.

VINNOVA AGENTSCHAP INNOVEREN & ONDERNEMEN

36

Who will use it?

Usage scenarios and demonstrators

ITEA 3 EMPHYSIS

Diversity of applications

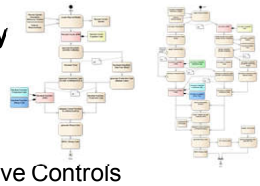
- Engine richness
- Engine vibrations
- Fault detection (thermal, cooling)
- After treatment
- Vehicle dynamics
- Energy Monitoring
- Torque vectoring
- Active damping

Tool independent format

- No S-Function constraint

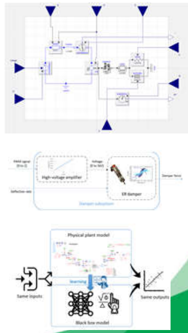
Control strategies versatility

- Feed forward
- Estimators
- Model Predictive Controls
- Non-Linear Model Predictive Controls
- Linear parameter-varying
- Kalman filters



Models types Versatility

- Non-linear models
- Inverted non-linear models
- Residuals model
- Linearized models
- Neural Networks



VINNOVA AGENTSCHAP INNOVEREN & ONDERNEMEN

Who will support it?

Planned and on-going tool development

ITEA 3 EMPHYSIS

Tool Name	Equation Code	Algorithm Code	Production Code	Binary Code
Siemens – AMEsim	?	Export, Planned	Export, Planned	
Dassault Systèmes – Dymola	Import, Export, Planned	Export, Prototype	Import, Export, Planned	
Modelon – JModelica	?	?	?	
OpenModelica	Import, Export, Planned			
ESI-ITI – SimulationX	Export, Planned	Export, Prototype		
ETAS – SCODE-CONGRA	Import, Export, Planned	Import, Export, Prototype	Export, Planned	
Dassault Systèmes – AUTOSAR Builder		Import, Planned	Import, Planned	Import, Planned
dSPACE – TargetLink		Import, Prototype	Export, Prototype	
AbsInt – Astrée			Import, Prototype	
Siemens – CSD			Import, Prototype	
PikeTec – TPT			Import, Prototype	

Import Export Planned Prototype

VINNOVA AGENTSCHAP INNOVEREN & ONDERNEMEN

When can I have it?
Schedule

eFMI Supporting Tools
- EMPHYSIS Project ends by February 2021
  - All planned tool prototypes will be finalized.
- Product readiness is expected not before mid 2021

eFMI Standard
- First draft has been finalized Mar. 2019
- After AlgCode and ProdCode have reached a stable state a preliminary version of the specification is considered to be shared under NDA.
- First official release after the end of the project after consultation of the Modelica Association.

39

Who is doing all this?
Acknowledgements

- **Germany**
  - Bosch[1,3]
  - DLR[2]
  - ETAS
  - ESI ITI
  - AbsInt
  - PikeTec
  - dSPACE
  - EFS
- **Sweden**
  - Dassault Systèmes AB[3]
  - Volvo Cars
  - Modelon
  - Linköping University
  - SICS East
- **France**
  - Siemens SAS[3]
  - Dassault Systèmes SE
  - Renault
  - CEA
  - University of Grenoble
  - FH Electronics
  - OSE
  - Soben
- **Belgium**
  - Siemens NV[3]
  - Dana
  - University of Antwerp
- **Canada***
  - Maplesoft[3]
- **OEM Advisory Board**
  - BMW
  - Daimler
  - Mazda
  - Volvo

Special thanks to the highly engaged members of the first eFMI Plug Fest, Sept. 23-25, 2019, Renningen, Germany: *Christoff Bürger, Kai Werther, Robert Reicherdt, Reinhold Heckmann, Jörg Niere, Gerd Kurzbach, Jishnu Jayaram, Yuri Durodie, Martin Otter, Andreas Pfeiffer, Christian Bertsch, Oliver Lenord* among many other contributors.

* w/o funding
1) Project Lead
2) Technical Coordination
3) National Coordination

40