



# Simulation of Model Execution for Embedded Systems

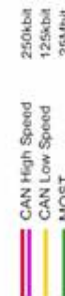
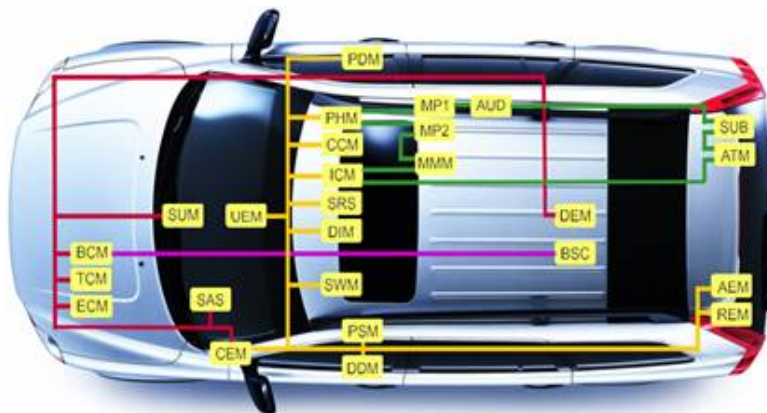
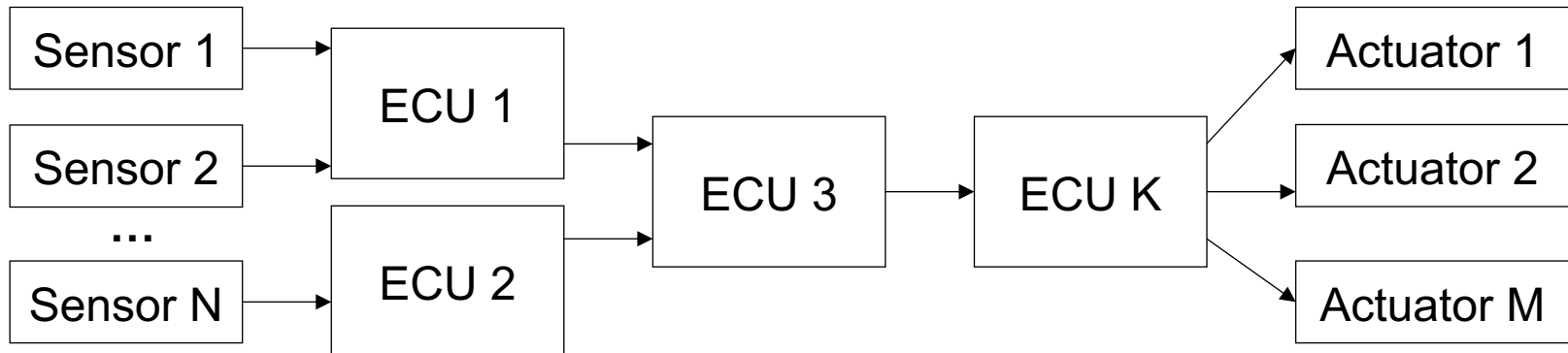
MLE 2019, Munich

Jörg Christian Kirchhof, Evgeny Kusmenko, Jean Meurice and Bernhard Rumpe  
Software Engineering  
RWTH Aachen

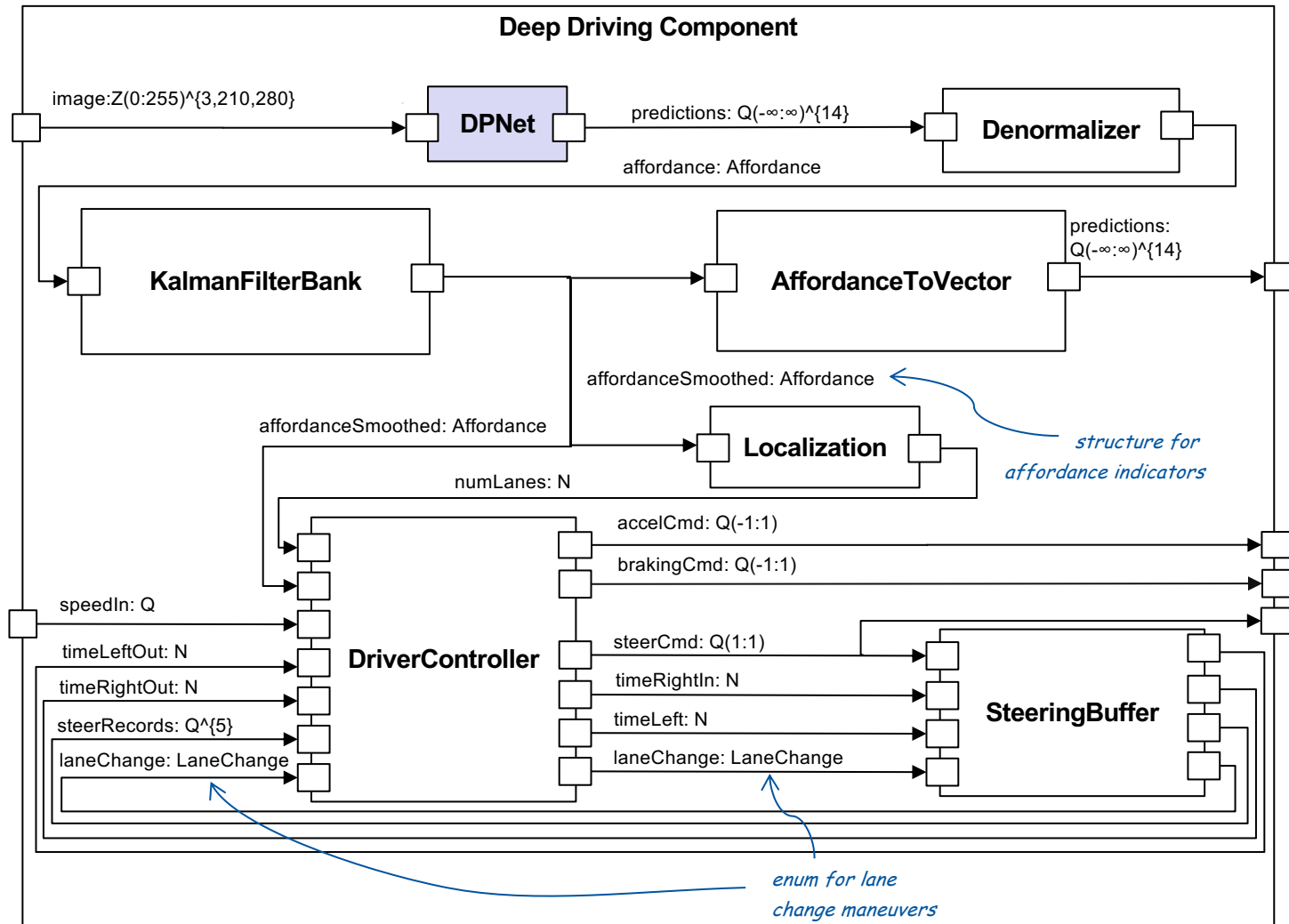
<http://www.se-rwth.de/>

# Motivation

- **Our mission:** provide **MBSE methodologies** for the **automotive** domain
- **EmbeddedMontiArc:** Modeling language for automotive systems

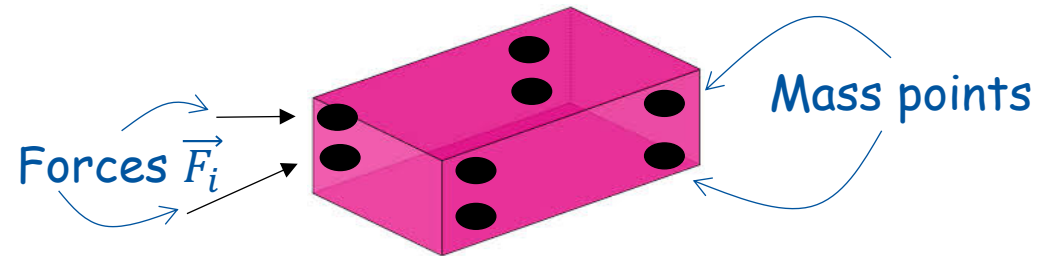
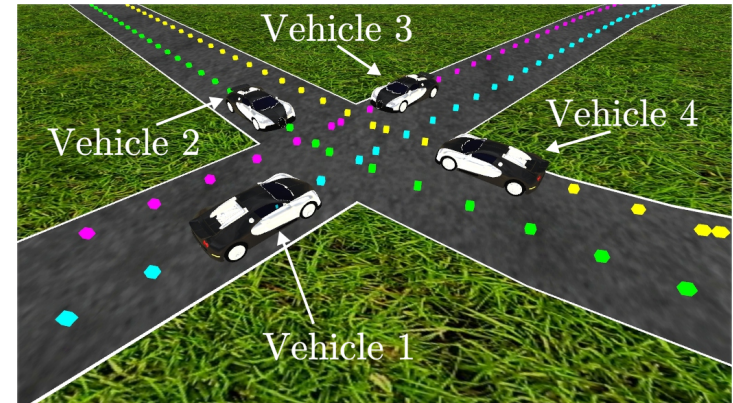


# EmbeddedMontiArc



# MontiSim

- Browser based 3D visualization
  - Simulator: Java
  - Visualization: JavaScript / ThreeJS
  - Enables CV + ML capabilities
- Environment model
  - [OpenStreetMap](#)
  - Probabilistic models for pedestrian behavior
  - Weather effects (e.g. changing the friction coefficient)
- Physics engine
  - Discrete time
  - **Rigid body** based (Euler loop)
  - Collision detection



# Requirements

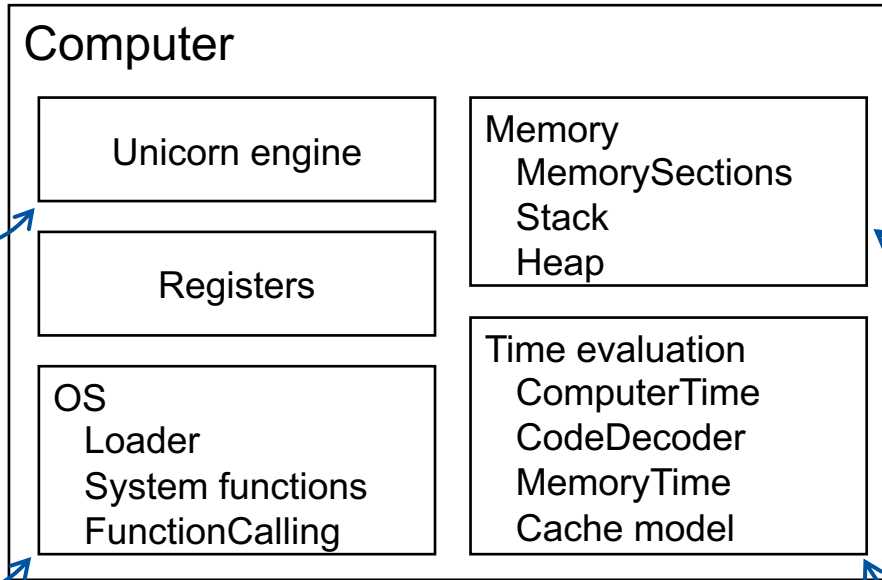
---

- **R1** The emulator must reproduce the real logic behavior of the emulated software
- **R2** Evaluate the execution time of the emulated software
- **R3** Reproducibility of the simulations (independently of the platform and hardware used)
- **R4** Allow the emulation of software compiled for other platforms
- **R5** Variability of the hardware models
- **R6** The hardware emulator can be used to emulate any program and evaluate their execution time

# Concept

*Emulate execution on hardware*

*Abstract from OS-specific functionalities*



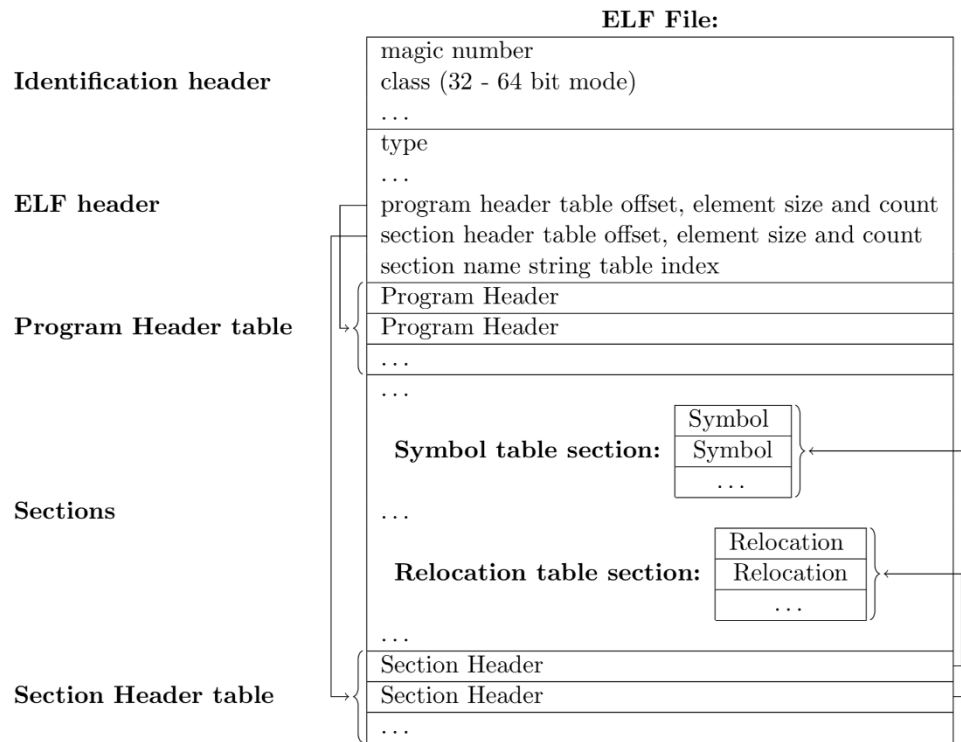
*Abstract from low-level C memory interface of the Unicorn engine*

*Connect to monitoring hooks of Unicorn engine and calculate needed time, based on hardware parameters*

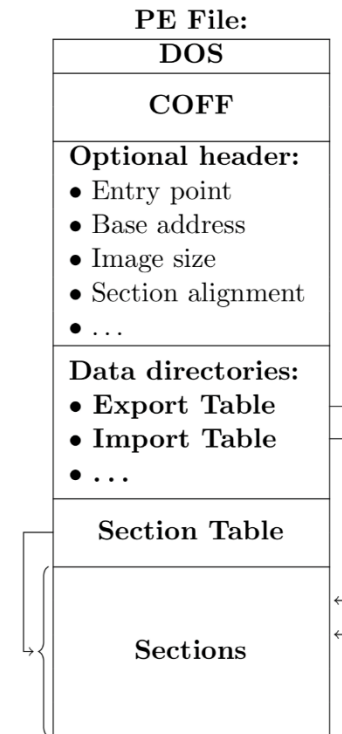
1	<code>os</code>	=	<code>windows</code>	Hardware
2	<code>cpu_frequency</code>	=	<code>1500000</code>	
3	<code>memory_frequency</code>	=	<code>150000</code>	
4	<code>cache_DL1</code>	=	<code>128, 1, 2</code>	
5	<code>cache_IL1</code>	=	<code>128, 1, 2</code>	
6	<code>cache_L2</code>	=	<code>1024, 10, 15</code>	

*Defines parameters of the hardware*

# UNIX vs. Windows



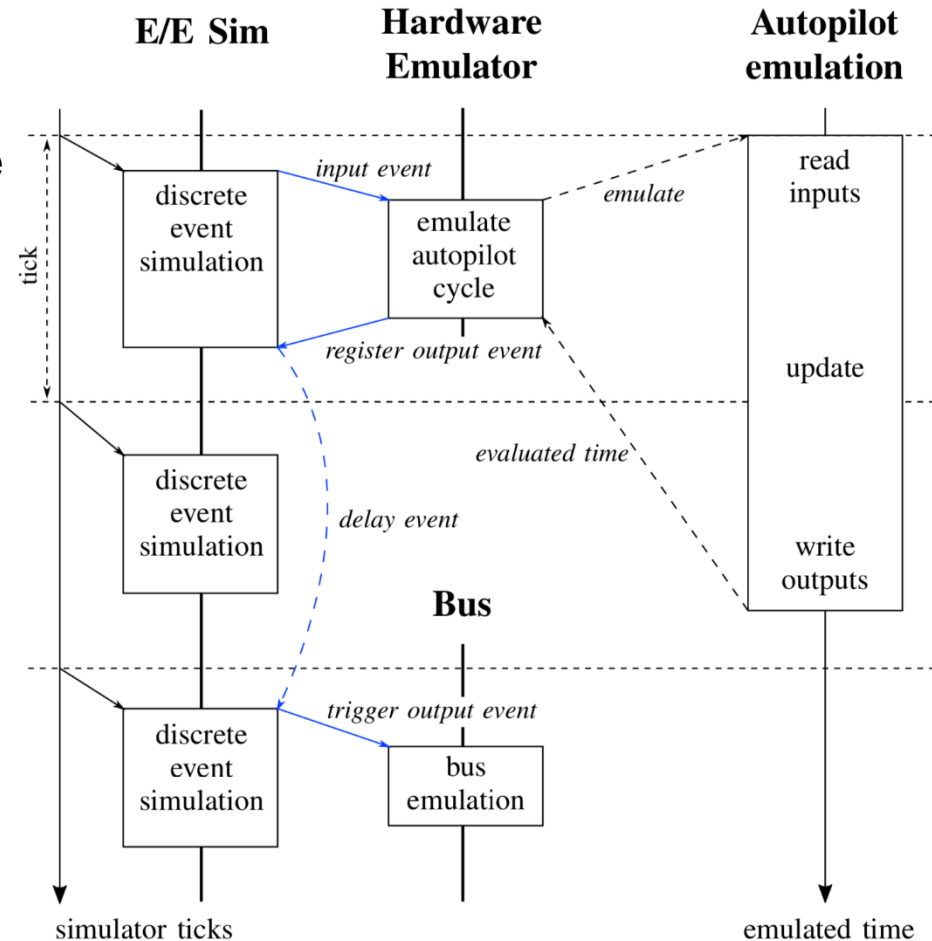
*Executable and Linking Format  
(UNIX)*



*Portable Executable  
(Windows)*

# EmbeddedMontiArc Integration

1. Discrete event simulator calls hardware emulator
2. Hardware emulator calculates outputs and time needed on real hardware
3. Discrete event simulator delays the output event according to the time estimation of the hardware emulator





# Evaluation

