

---

# Prototyping Automotive Smart Ecosystems

M.Sc. Emilia Cioroica, Dr.-Ing Thomas Kuhn, Dr.-Ing Thomas Bauer  
{Emilia.Cioroica, Thomas.Kuhn,Thomas.Bauer}@iese.fraunhofer.de

---

Luxembourg, 25 June 2018

# Motivation

## Context

- From isolated systems to open systems to digital ecosystems
  - SECO vs SES vs. SoS
  - Testing of Smart Ecosystems

DIFFERENCES BETWEEN ECOSYSTEMS AND SYSTEMS OF SYSTEMS

<b>Analyzed criteria</b>	<b>Ecosystems</b>	<b>Systems of Systems</b>
Form of Organization	Aggregated Entity	Composed Entity
Relationships between components	Collaborating, Competitive	Collaborating
Component Types	Organizations + computational systems	Computational systems
Source of the Scope/Goal	Emerging	Defined by humans
Functional Driver	Motivation, incentives	Predefined Rules
Origin	Decision of systems at Runtime	Composed by Humans

2

# Motivation



# Motivation (1/4)



# Motivation

## What is needed to test smart ecosystems?

- Virtual Engineering
  - Simulation method are more precise
  - Automation
  - Lack of expert knowledge
  - ISO 26262 explicitly recommends simulation as quality assurance
  - more..
- Extend the number of test scenarios
  - Integrate Virtual and Real world
- Flexible platform

**Solution**

# The ANKI Demonstrator

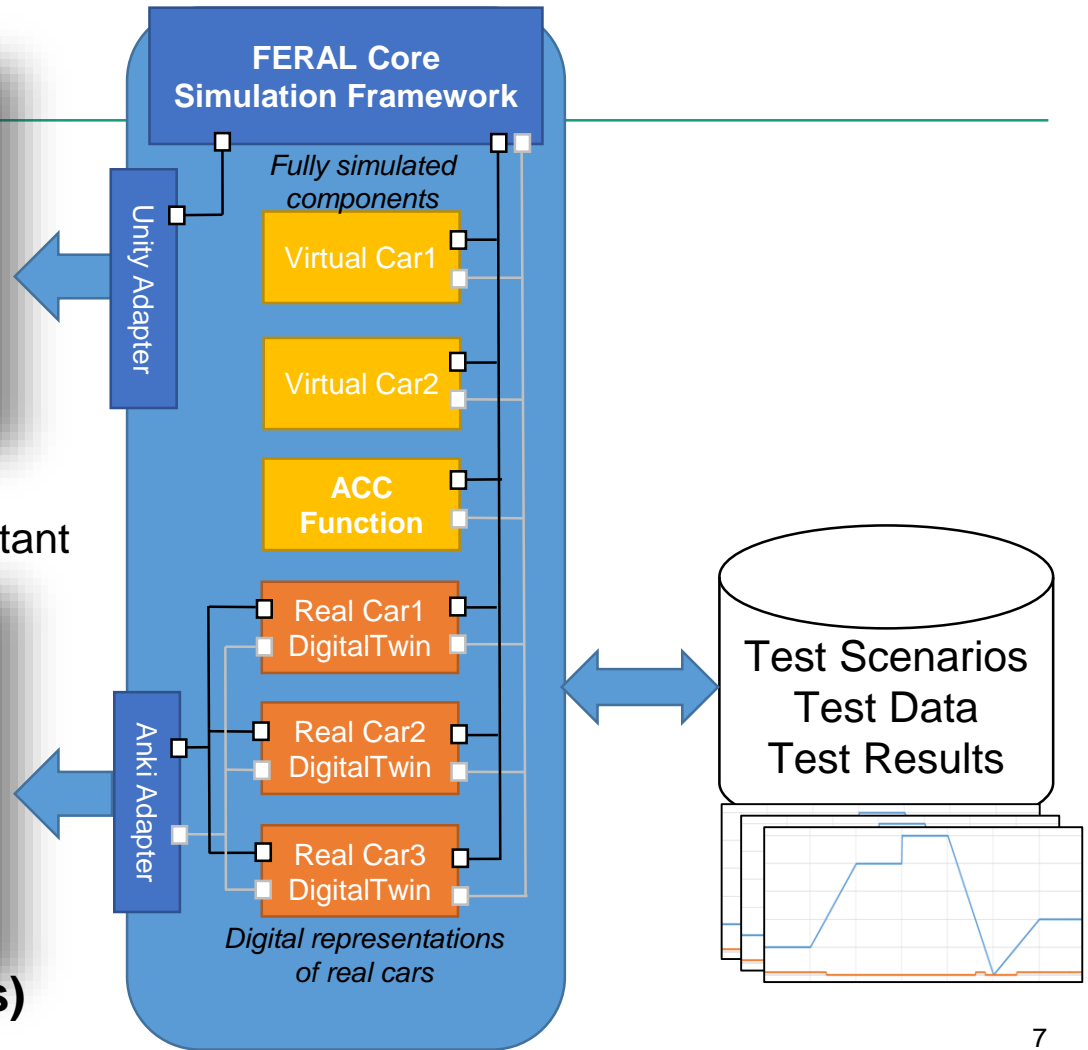
## Virtual Representation (Unity)



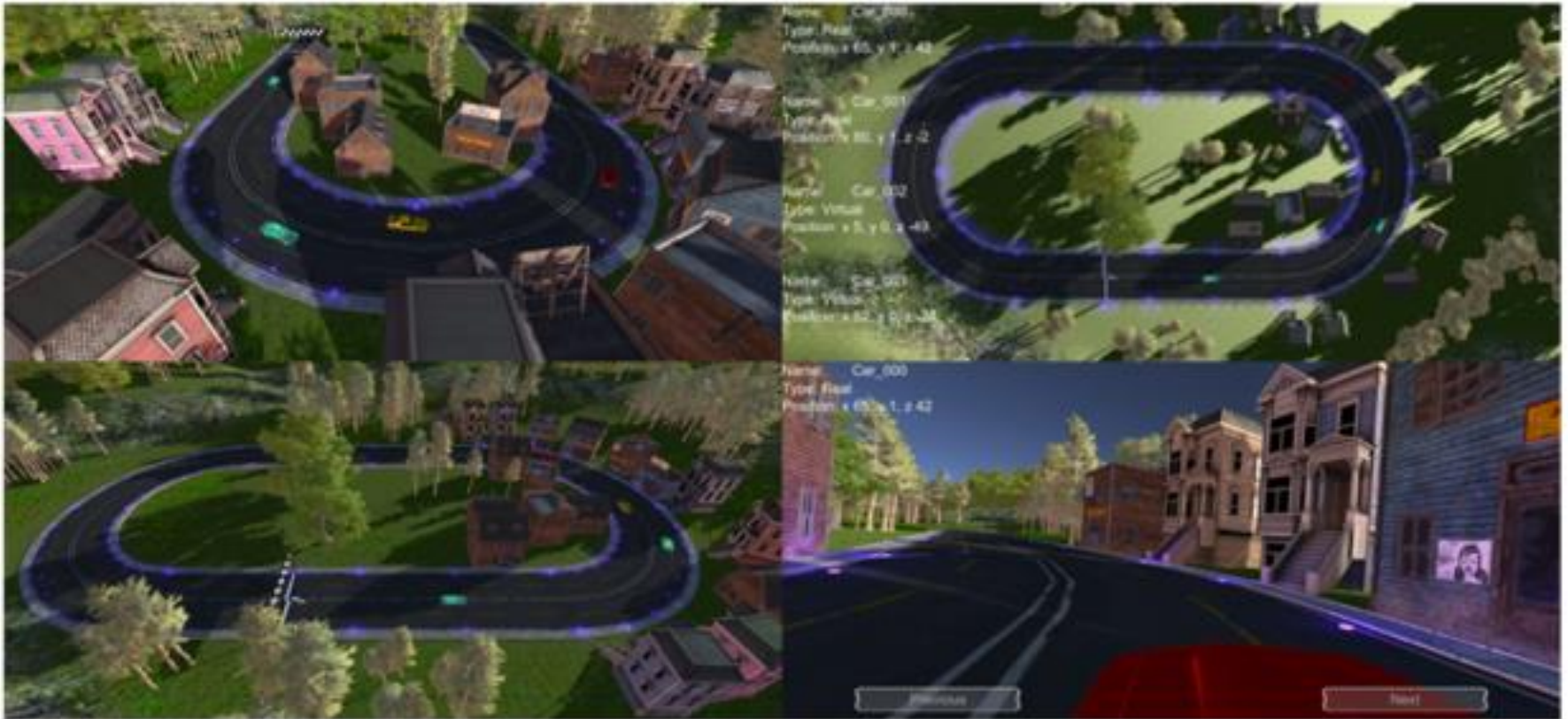
Example „System under Test“  
ACC of a convoy; intersection assistant



„Real World“ (Anki Track+Cars)  
Platform: ANKI Overdrive



# Unity for Visualization of Scenario and Car Data





# Blockly Interface for Supporting End-user

## Workspace information

Server status: **online**

Currently **60** block(s) are in use.

You can use following car(s):

**Anki-Cars (0):**

**Virtual Cars (2):** virtual\_car\_1 | virtual\_car\_2

The screenshot displays the Blockly workspace with a left sidebar containing categories: Logic, Loops, Math, Text, Lists, Variables, Functions, TCC Control, TCC ID's, TCC Variables, and TCC Functions. The workspace contains two code snippets:

- Initial Setup (Green block):**
  - start speed for all cars: 300
  - min distance: 100
  - max distance: 200
  - global min speed: 300
  - global max speed: 400
- End User Programming (Blue block):**
  - Car ID: Virtual ID's virtual\_car\_1
  - initial car speed: 300
  - start lane: lane 2
  - User Program Code:**
    - if (current distance to car ahead < min distance) do: change speed to (current speed - (current speed of car ahead - 1))
- End User Programming (Blue block):**
  - Car ID: Virtual ID's virtual\_car\_2
  - initial car speed: start speed
  - start lane: lane 2
  - User Program Code:**
    - if (current distance to car ahead > max distance) do: increase speed by 13%
    - if (current lane = lane 1) do: switch lane to lane 2, increase speed by 1
    - else: switch lane to lane 1

# Platform



# Conclusions

# Conclusions

## Prototype Platform

- Virtual Engineering
- Enhance test scenarios
  - Real agents
  - Virtual agents
- Extensible

# Question 😊

# Thank you !