2. Applications for MOST - Design Tools for Complex Systems

Prof. Dr. Uwe Aßmann

Technische Universität Dresden

Institut für Software- und Multimediatechnik

Lehrstuhl Softwaretechnologie

http://st.inf.tu-dresden.de/teaching/most

WS 17-0.2, 30.09.17

1) Cyber-physical systems (CPS)
2) Two forms of CPS
    1) Experience with Cloud Robots
    2) A Killer App for CPS
3) Design Tools for Complex Software Systems
4) Design of CPS with Domain-Specific CPS tool chain
5) Why MDSD-TS?

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Obligatory Literature

- [Preevision] Vector. Modellbasierte Elektrik-/Elektronik-Entwicklung vom Architekturentwurf bis zur Serienreife. Preevision Handbuch
  - http://vector.com/portal/medien/cmc/marketing_items/web/91106.pdf
- [Reichmann] Clemens Reichmann, Daniel Gebauer, Klaus D. Müller-Glaser. Model Level Coupling of Heterogeneous Embedded Systems. Technical Report, FZI, 2008
  - http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.101.366
- [ETAS] Ulrich Lauff, Christoph Stoermer, Thomas Dollmaier, Mathias Klauda. ETAS GmbH, Stuttgart, Germany. Development Tools for Hybrids and Electric Cars.
  - http://www.etas.com/download-center-files/products_ASCET_Software_Products/1002_ATZ_elektronik_Entwicklungswerkzeuge_fuer_HEV_EV_EN.pdf

© Prof. U. Aßmann

Literature

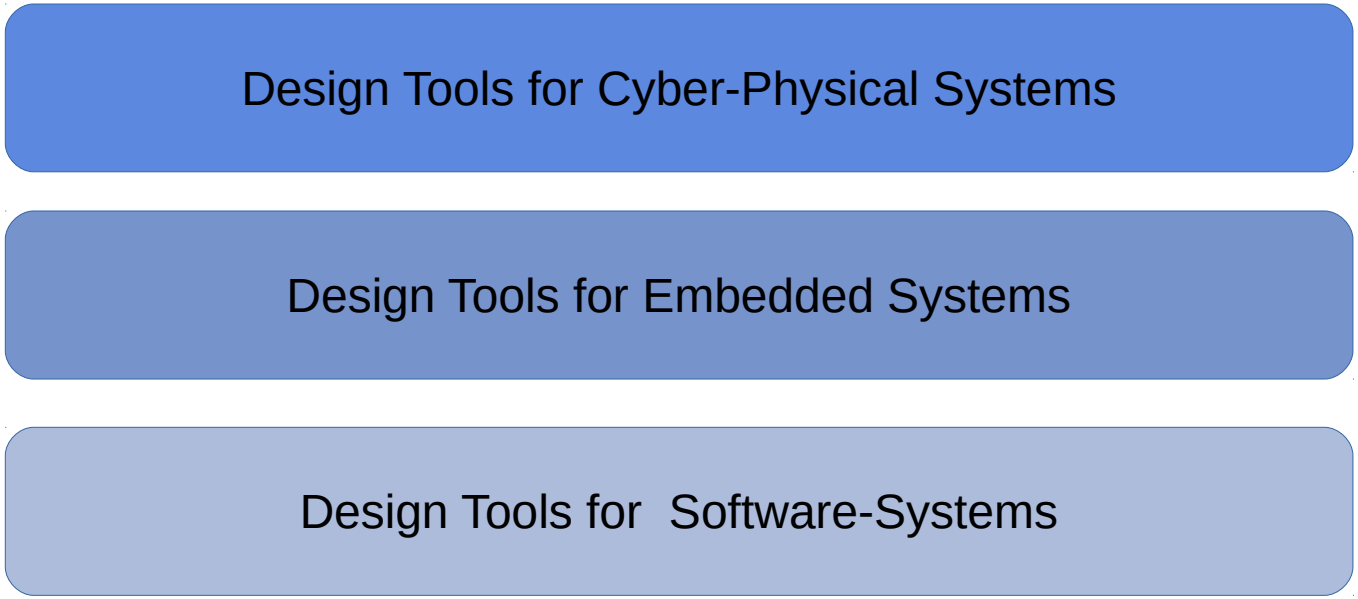
- [Zverlov] Sergey Zverlov. Comparison of two level-based Approaches for the Development of Embedded Systems. Bachelor Thesis in Computer Science. TU München, 2008.
- [Wurman] Peter R. Wurman, Raffaello D'Andrea, and Mick Mountz. Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses. AI Magazine Volume 29 Number 1 (2008) (© AAAI)
- [MüGl09] Prof. Dr.-Ing. Klaus D. Müller-Glaser. Slide set. Model-Driven Engineering for Automotive Systems. UCSD SAASE 2009
  - http://jacobsschool.ucsd.edu/GordonCenter/g_leadership/l_summer/docs/saase/symposium-presentations/KlausMuellerGlaser.pdf

© Prof. U. Aßmann

2.1. Why Do We Need Model-driven Software Development in Technical Spaces?

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Intelligent Design Tools (Integrated Development Environments)



© Prof. U. Aßmann

2.1. What is a Cyber-Physical System (CPS)?

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Smart Parking

http://commons.wikimedia.org/wiki/File:Bundesarchiv_Bild_183-H0605-0007-001,_Rostock,_Ernst-Th%C3%A4lmann-Platz,_Parkplatz,_Marienkirche.jpg#mediaviewer/File:Bundesarchiv_Bild_183-H0605-0007-001,_Rostock,_Ernst-Th%C3%A4lmann-Platz,_Parkplatz,_Marienkirche.jpg

© Prof. U. Aßmann, TU Dresden

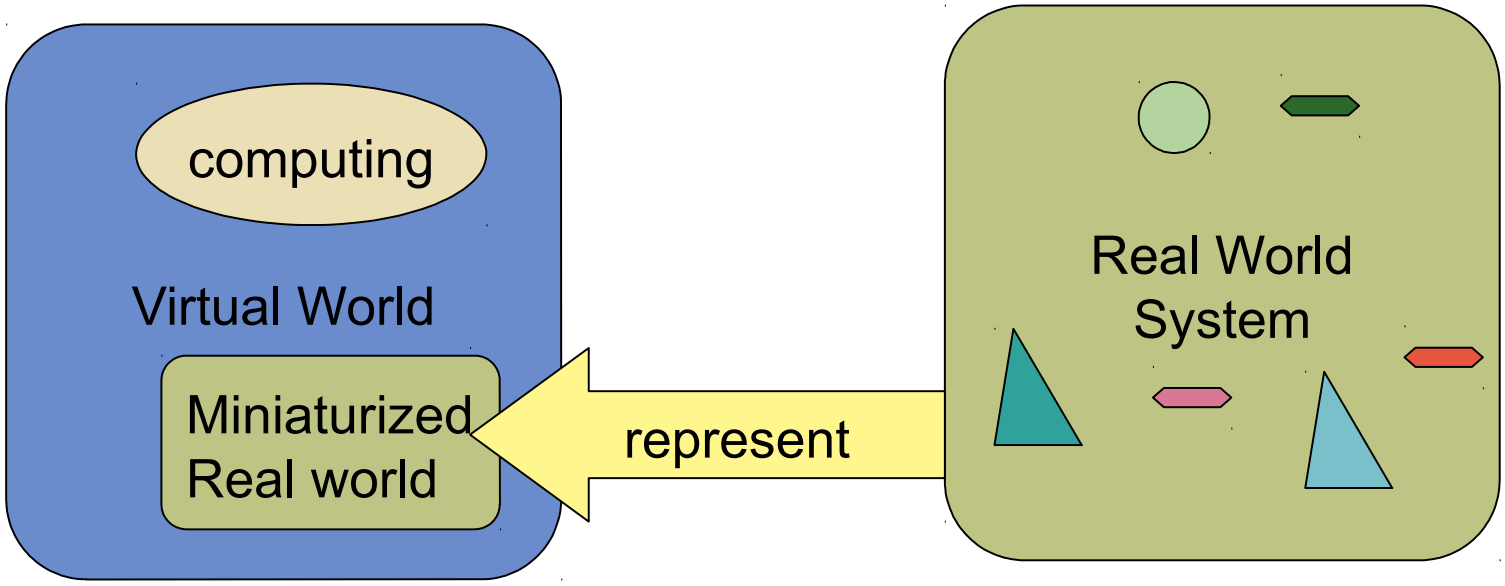
Kiva Bots for Logistics

- [Wurmer] Just search on YouTube for Kiva Systems
- https://www.youtube.com/watch?v=8gy5tYVR-28
- https://www.youtube.com/watch?v=6KRjuuEVEZs

© Prof. U. Aßmann

„Standard“ Computing

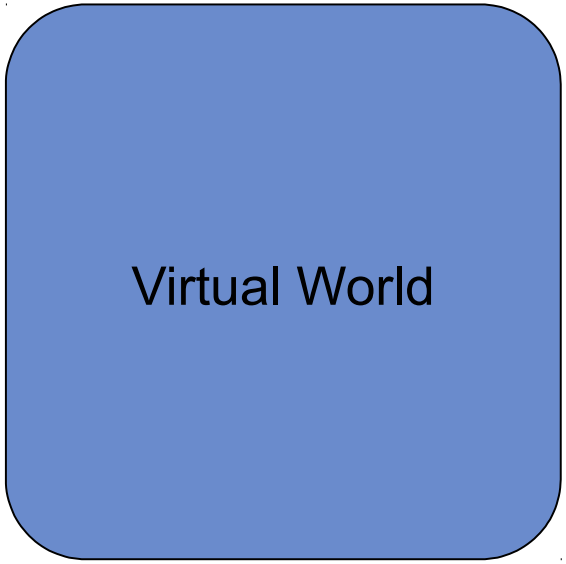
- „Standard“ Computing maps the real world into the computer and computes about it by simulation

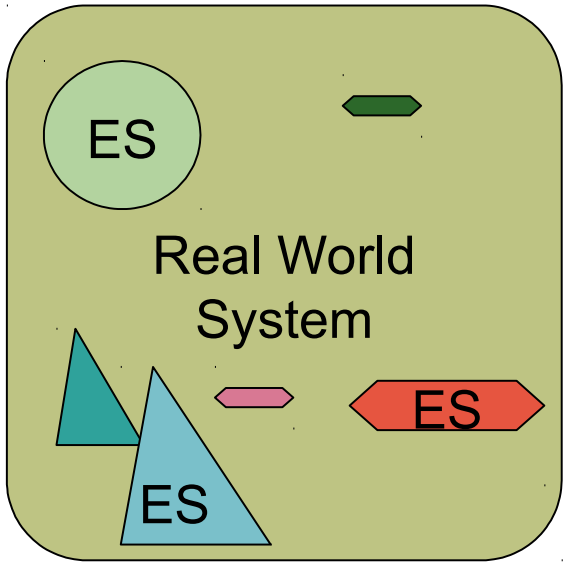


© Prof. U. Aßmann

Embedded System

- The computer is integrated into the real-life object

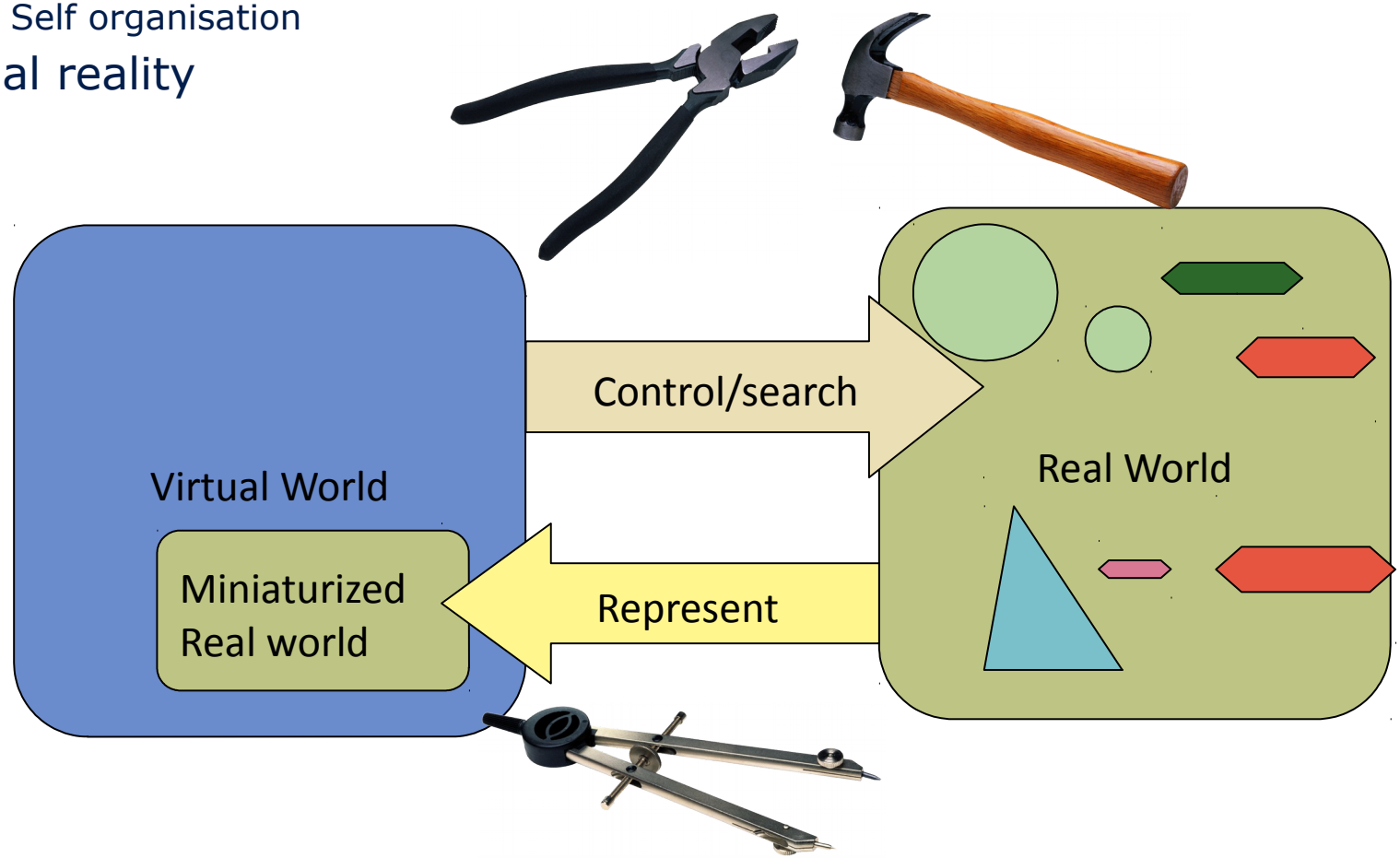




© Prof. U. Aßmann

Cyber-Physical System (CPS)

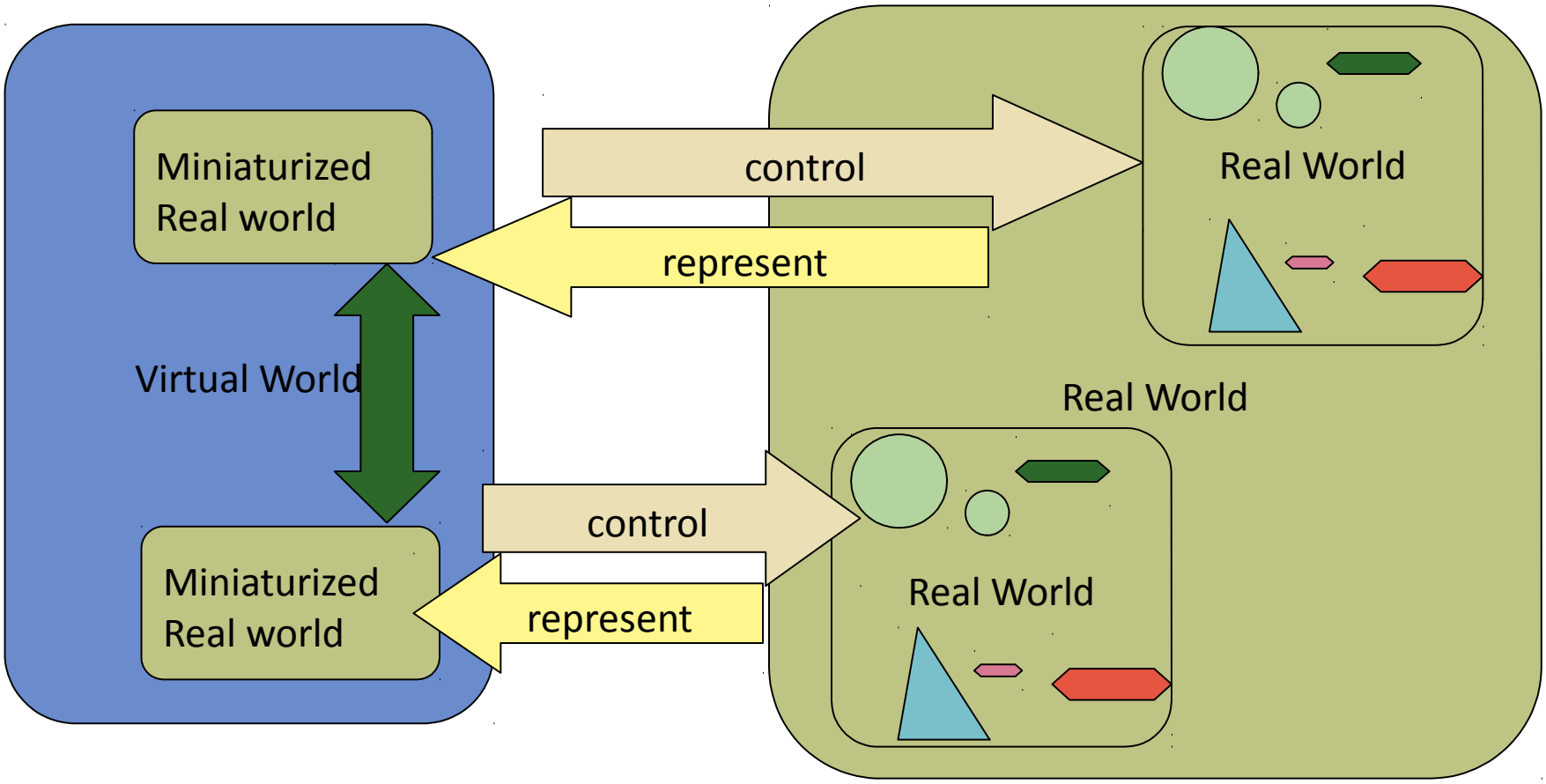
- Simulation of intelligent things in space and time
  - Search possible
- Control of the intelligent things in space and time
  - Self regulation
  - Self optimization
  - Self organisation
- Dual reality



© Prof. U. Aßmann

The Internet of Things

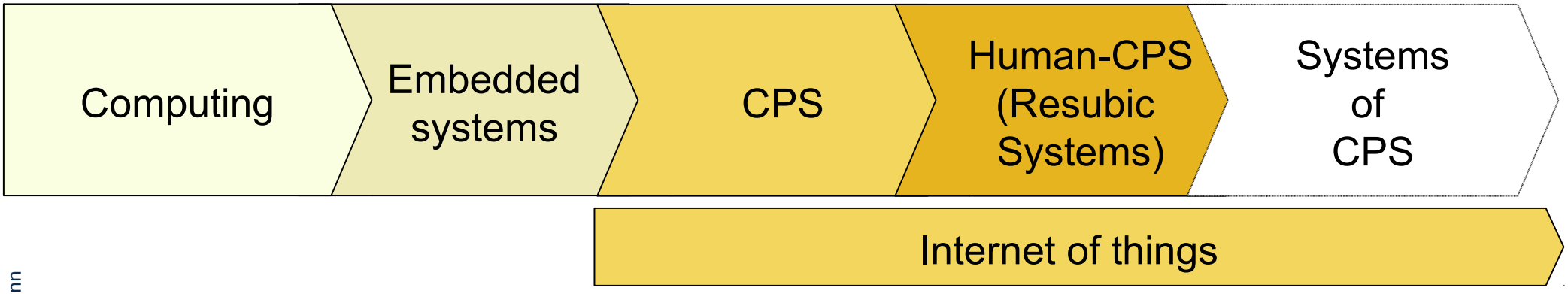
- Systems of CPS, i.e., remote tools



© Prof. U. Aßmann

Trend CPS

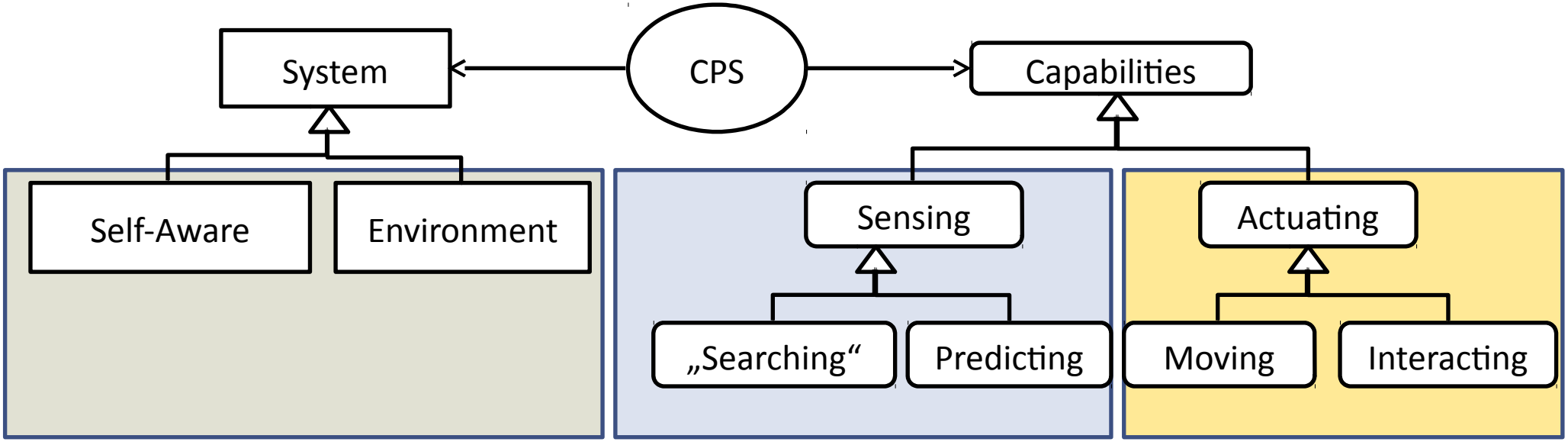
- Cyber-physical systems are the first step in the internet of things



10**9 chips 10**10 chips 10**13 chips

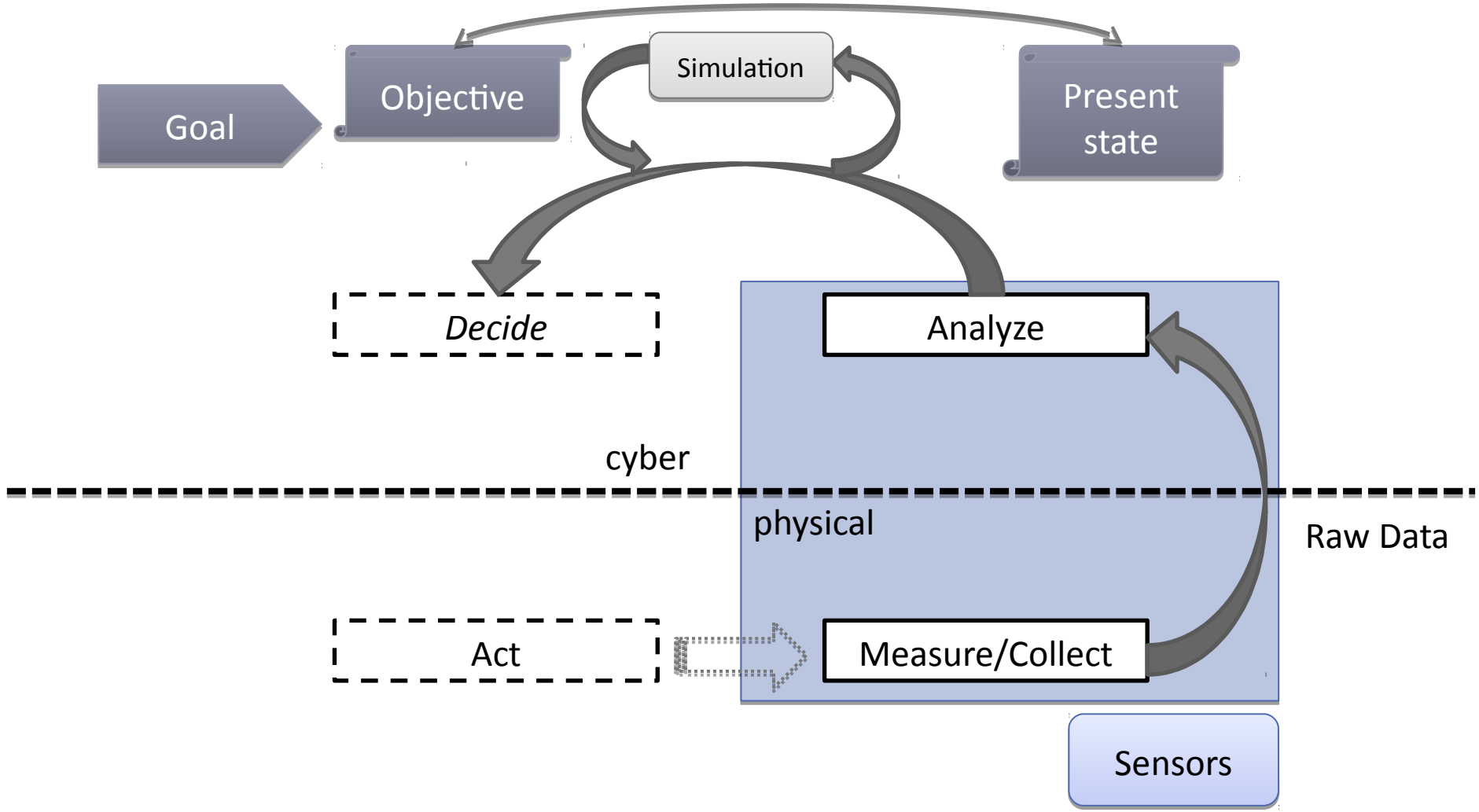
© Prof. U. Aßmann

Two Classes of Cyber-Physical Systems for Cyber-Physical **Search** and **Management**



© Prof. U. Aßmann

Cyber-Physical Database Systems = Analysis, Simulation and Prediction



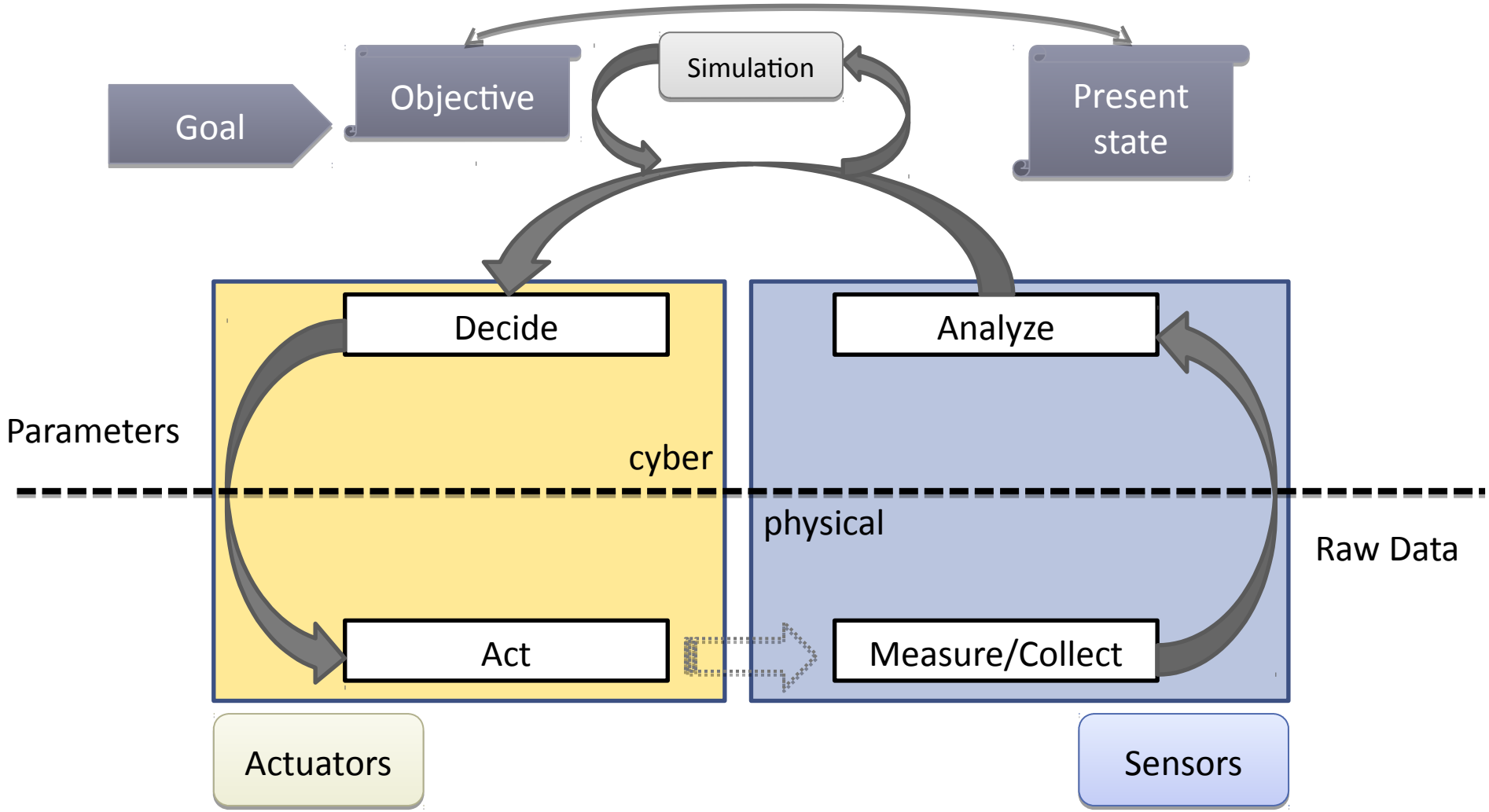
© Prof. U. Aßmann

A Cyber-Physical System

http://commons.wikimedia.org/wiki/File:Traffic_seen_from_top_of_Arc_de_Triomphe.JPG

© Prof. U. Aßmann

Cloud Robots = Cyber-Physical Management Systems



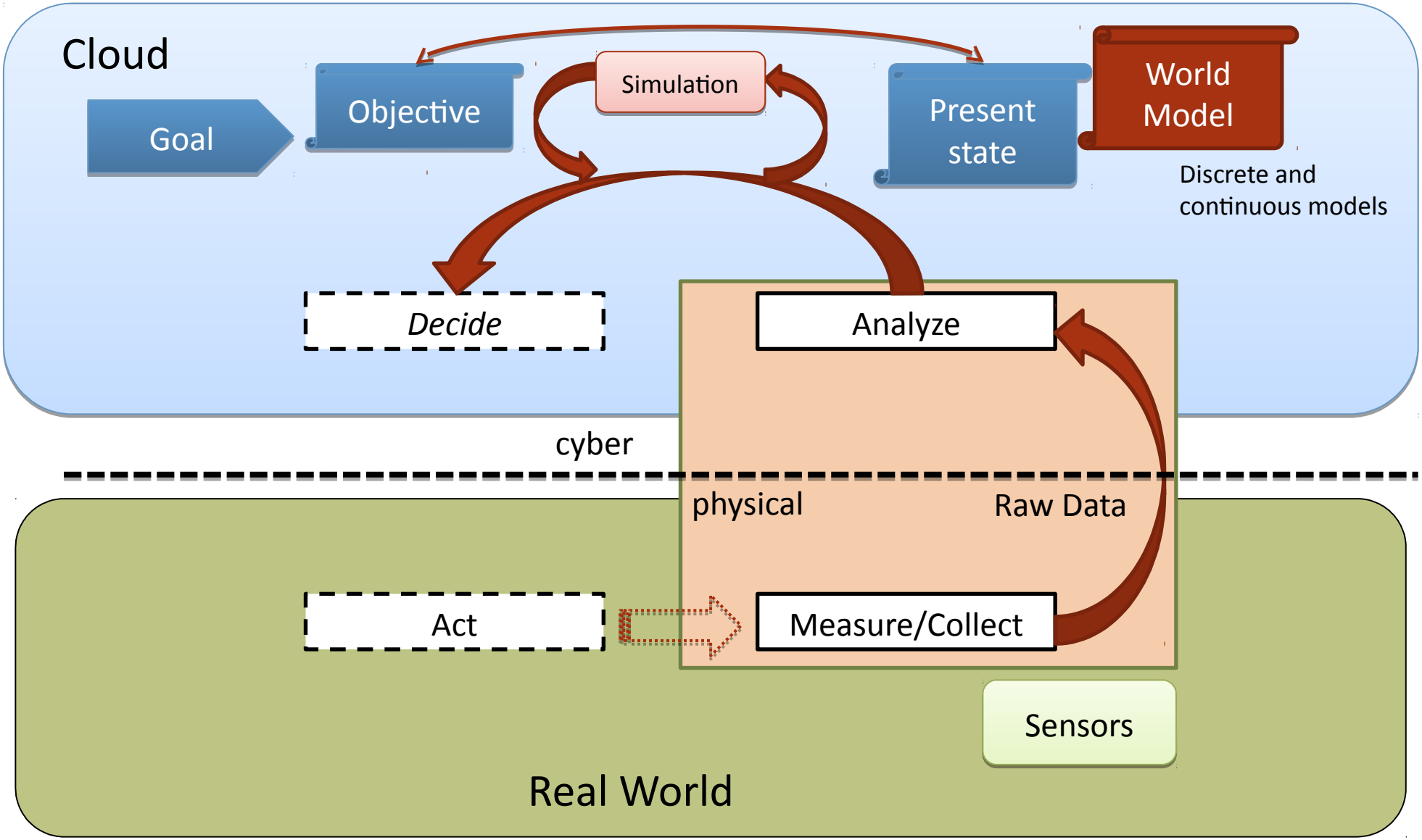
© Prof. U. Aßmann

2.1.2. Two Basic Forms of CPS

- World Databases
- Cloud Robots

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

World Database Systems are Monitoring CPS (Analysis, Simulation and Prediction)



© Prof. U. Aßmann

Ex.: The VAMOS Traffic Management System (Verkehrsleitsystem) Dresden

- Realtime data from the city‘s traffic
- http://www.vamosportal.de/
- http://wwwpub.zih.tu-dresden.de/~vamos/flyer/vamos_web.pdf

© Prof. U. Aßmann, TU Dresden

2.1.2 Important World Models of World Databases

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Physical Location of Thing in Environment

- Where is my thing in space?
  - Model of Physical Environment required
  - spatial, real-timed
  - magnetic, heat, humidity, user-defined
  - Continuous models

CPS
Need
Real-time
World Models

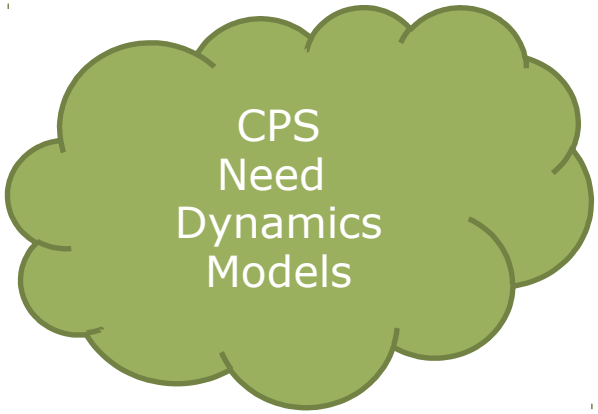
http://tf3dm.com/3d-model/the-city-39441.html

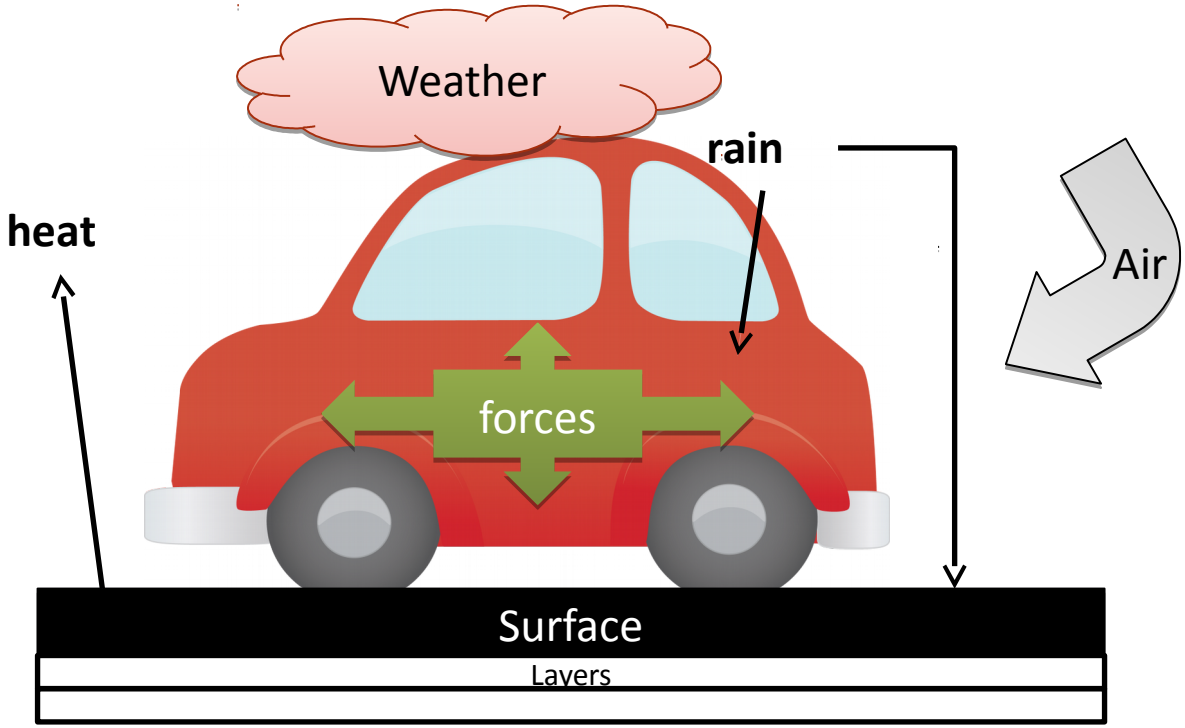
3D office models
Building models
City models
http://www.turbosquid.com

© Prof. U. Aßmann

Physical Dynamics (Movement) of Thing

- How does it move in space?
  - Continuous modeling languages (Modelica)
  - Www.modelica.org, www.openmodelica.org



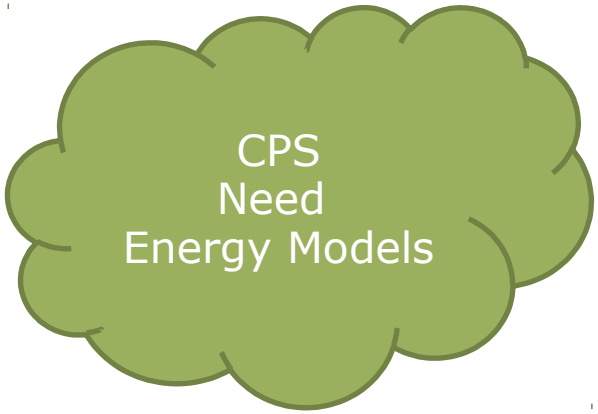


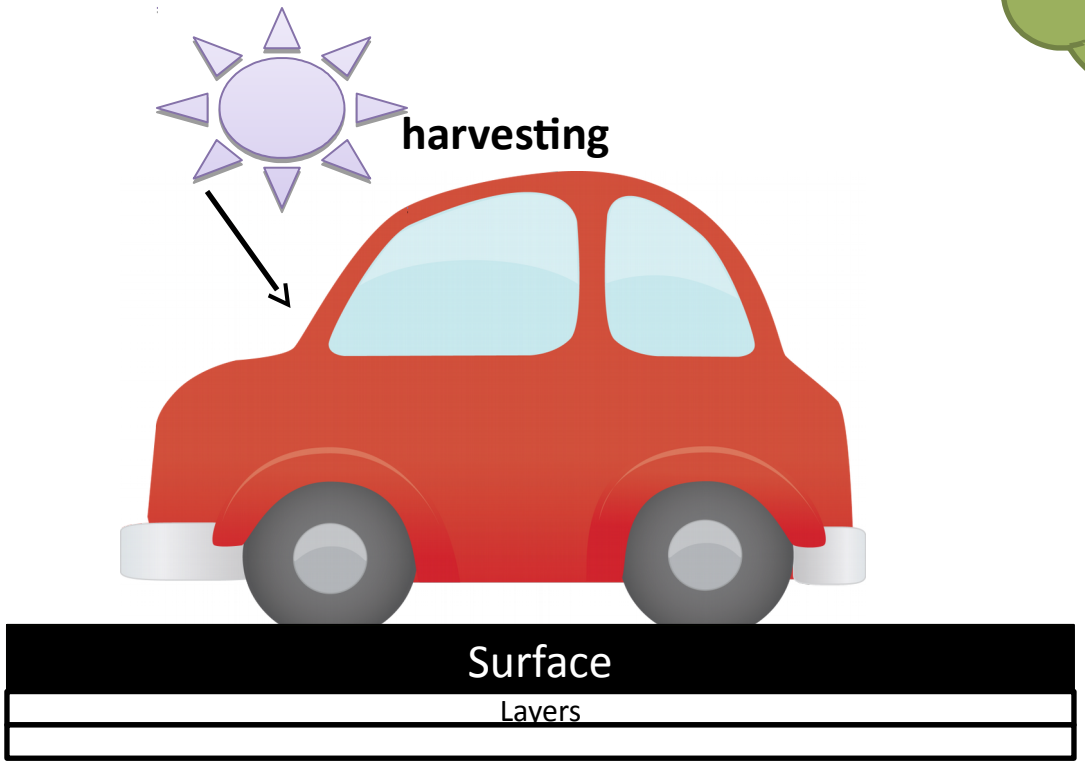
complex interplay of
- surface props
- weather: wind, rain, heat

© Prof. U. Aßmann

Energy Consumption of Thing

- How much energy is left for its tasks?

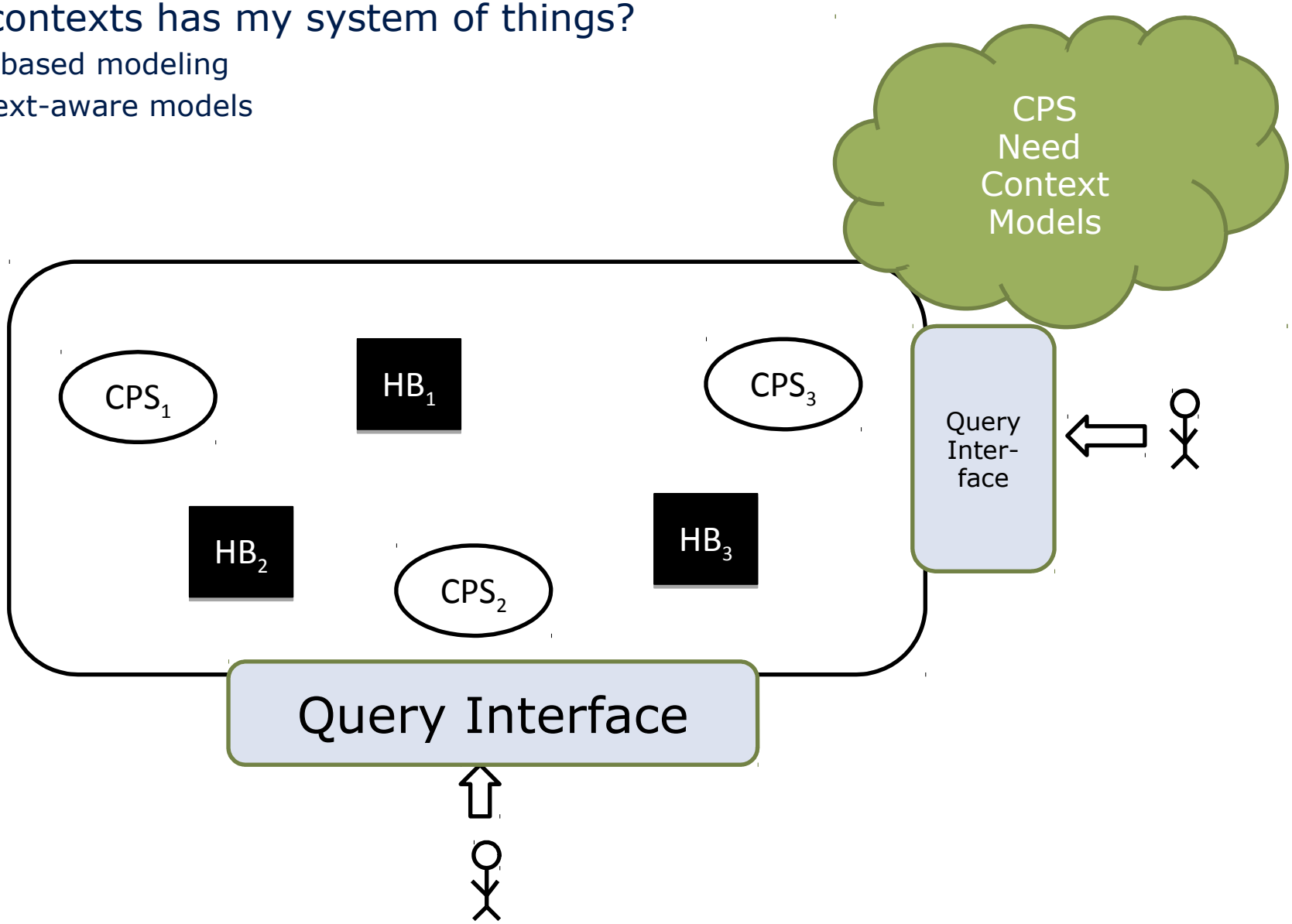




© Prof. U. Aßmann

Current Physical Composition of a Thing

- Which contexts has my system of things?
  - Role-based modeling
  - Context-aware models

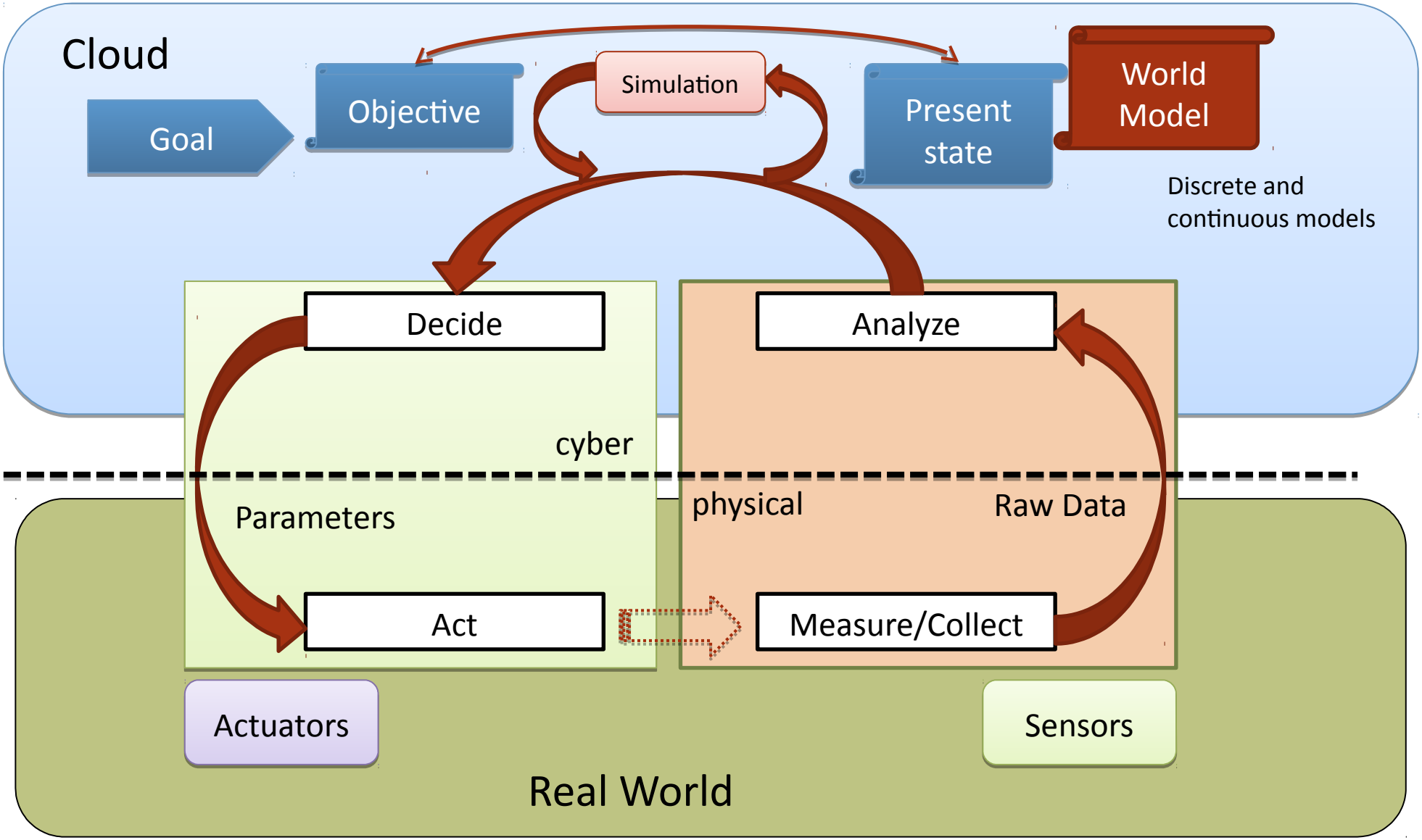


© Prof. U. Aßmann

2.1.3. What is a Cloud Robot?

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Cloud Robots are Controlling CPS

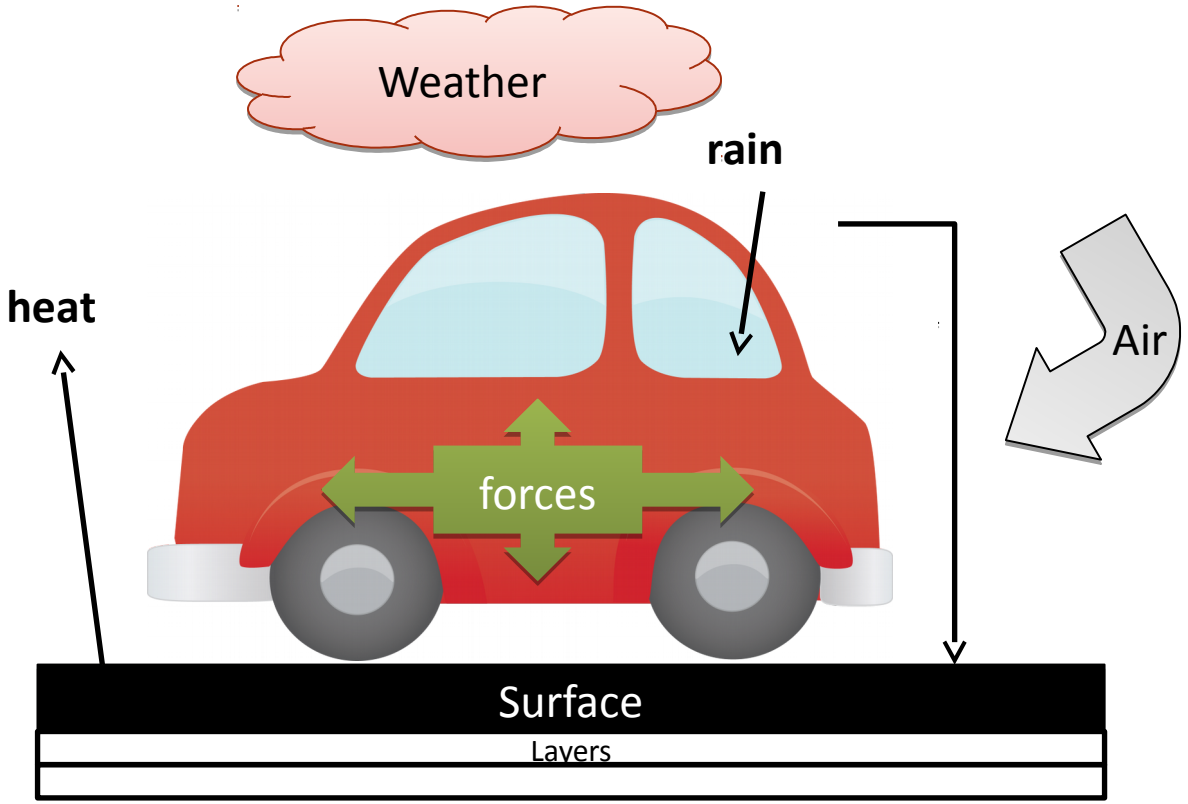


© Prof. U. Aßmann

Physical Dynamics (Movement) of Cloud Robot

- How can I **control** a cloud robot move in space?

Cloud Robots
Need
World Models



© Prof. U. Aßmann

2.1.4. Experiments with Cloud Robots

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

A Cloud Robot uses a Standard Robotic Platform
Hello, I'm NAO

Made by

- ALDEBARAN Robotics Paris, Frankreich
  [http://www.aldebaran-robotics.com/]

Application fields

- Teaching (Robot programming)
- Research
  - Robotics, AI
  - RoboCup
  - **Software Engineering**

Price

- 9.000 – 12.000 €

© Prof. U. Aßmann

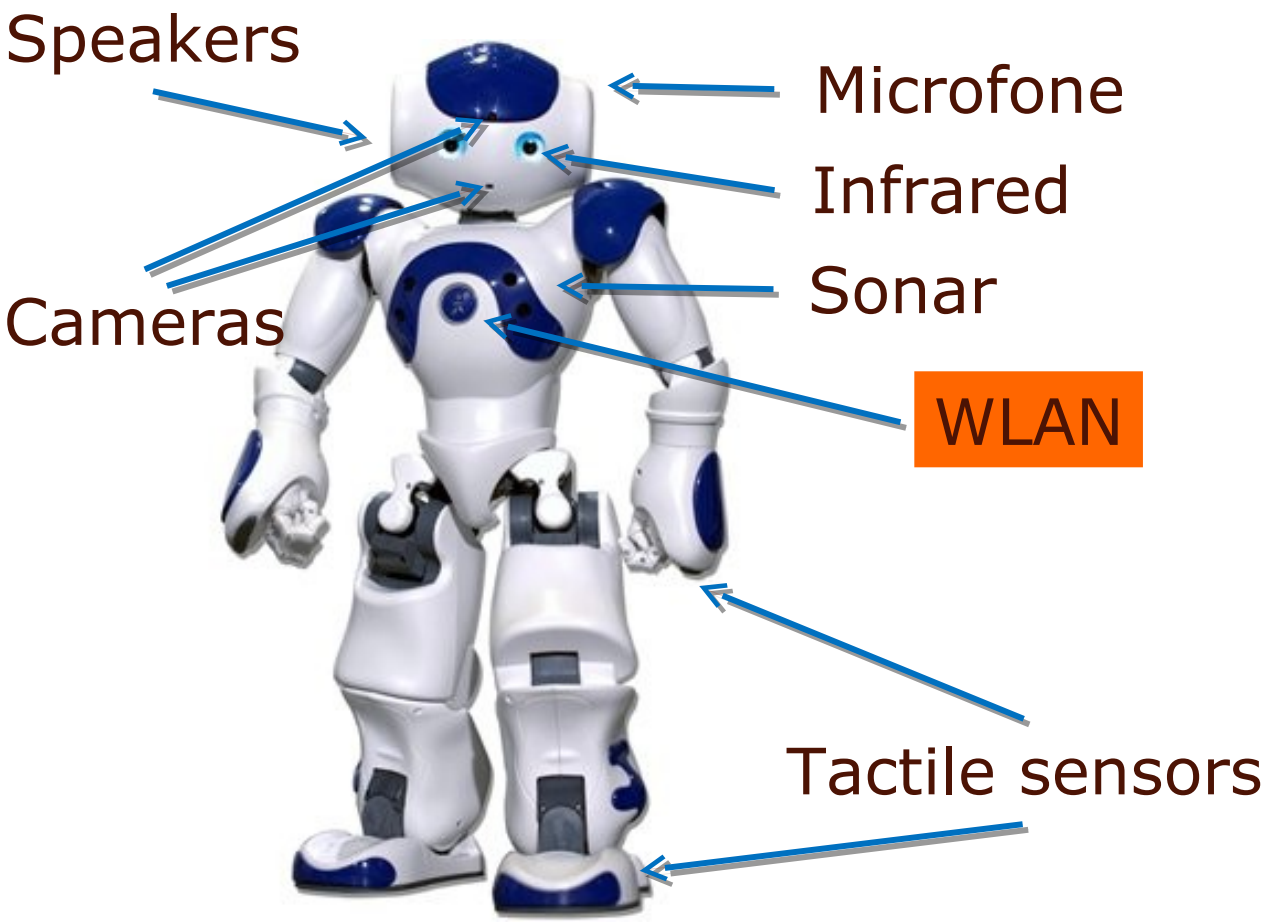
Nao Fact Sheet

Length: 58cm

Weight: 5kg

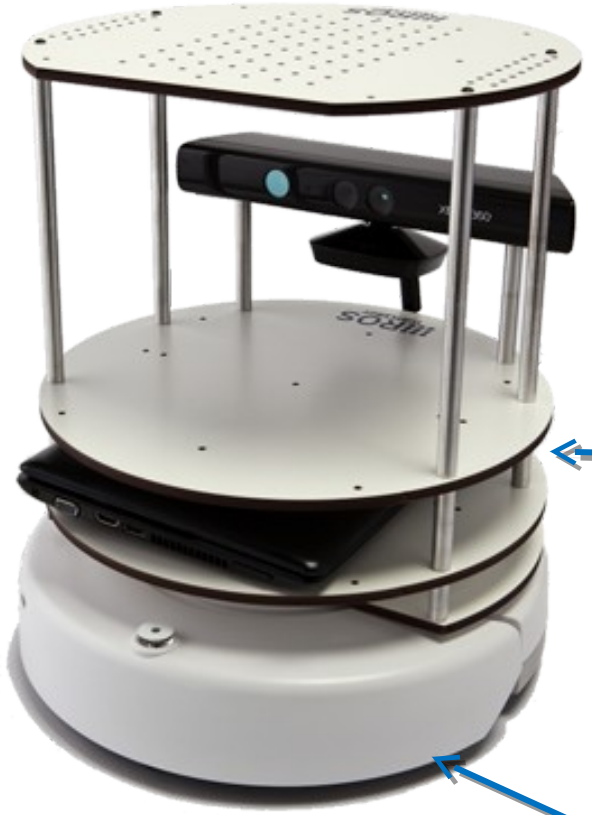
Hardware:
- x86 AMD GEODE 500MHz
- 256MB RAM
- 21 motors
- Battery 55Wh

OS:
Embedded Linux 32bit



© Prof. U. Aßmann

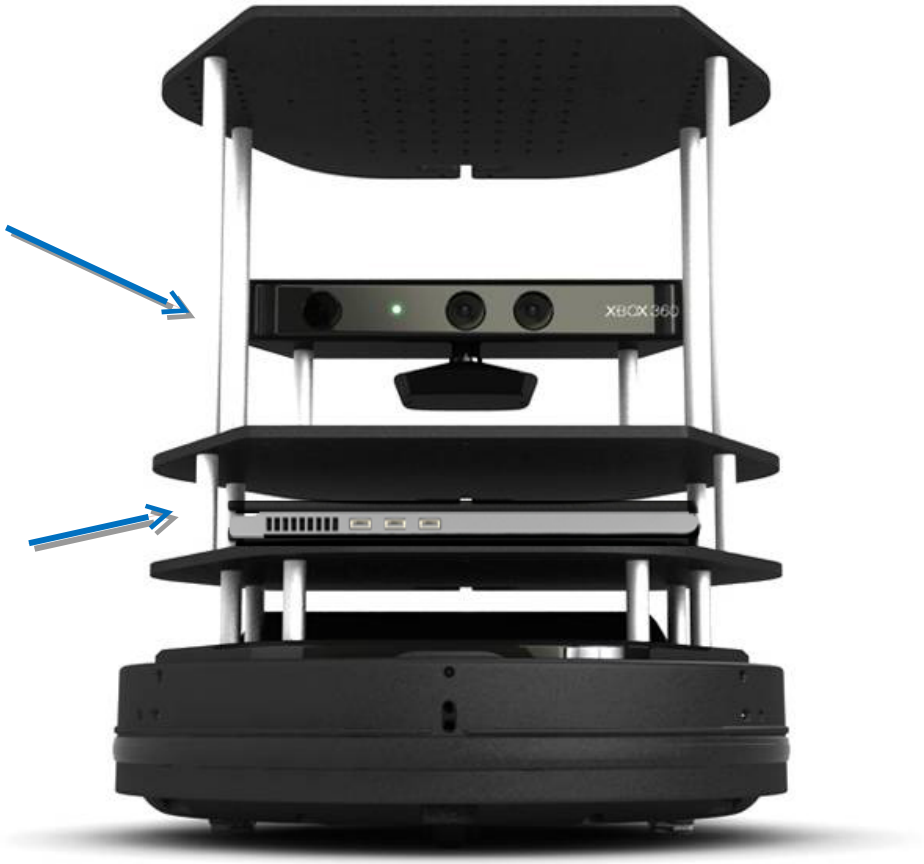
Turtle Bot



Kinect

Netbook

Roomla



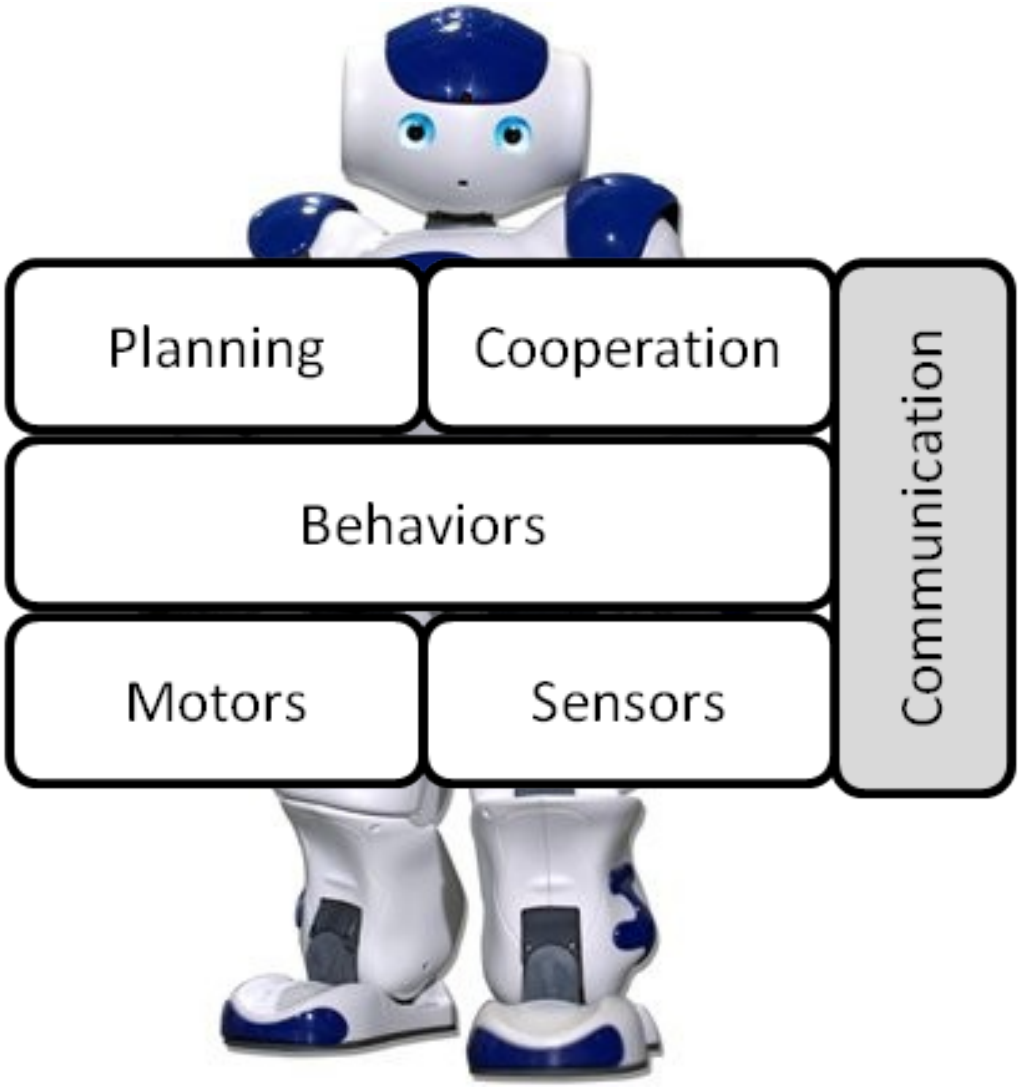
Kobuki

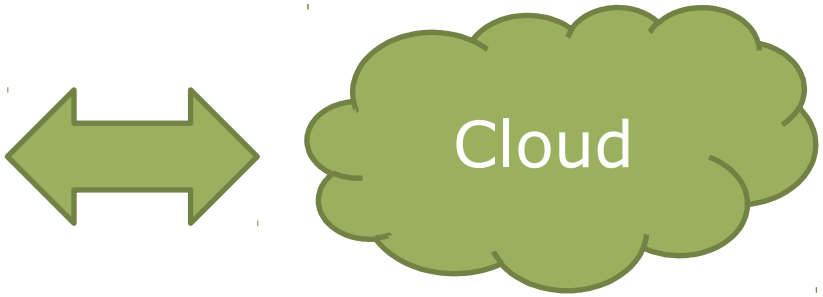
50kHz Sensor data rate

http://wiki.ros.org/Robots/TurtleBot
http://www.turtlebot.com

Prof. U. Aßmann, TU Dresden
© Prof. U. Aßmann

ResUbic Lab: NAO Web Service Architecture



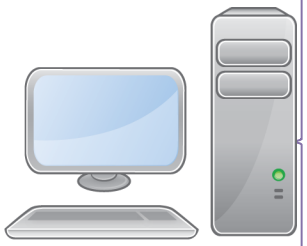


http://code.google.com/p/naoservice/

© Prof. U. Aßmann, TU Dresden

NAO Web Service and Communication Framework

Runs in Cloud



Client (Java, NAOText)

Nao Utility Classes (Java)

Java Nao Web Service Proxies (Java)

Generated

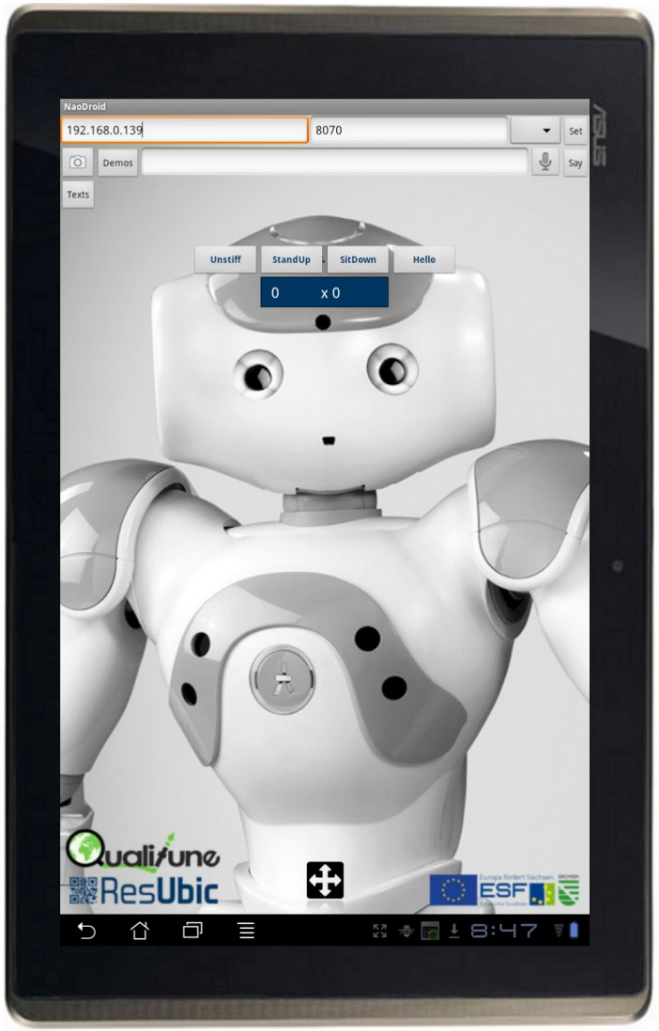
HTTP

Runs on Nao

Nao Web Service (Python, reflective)

Python Bridge for NaoQi (Python)

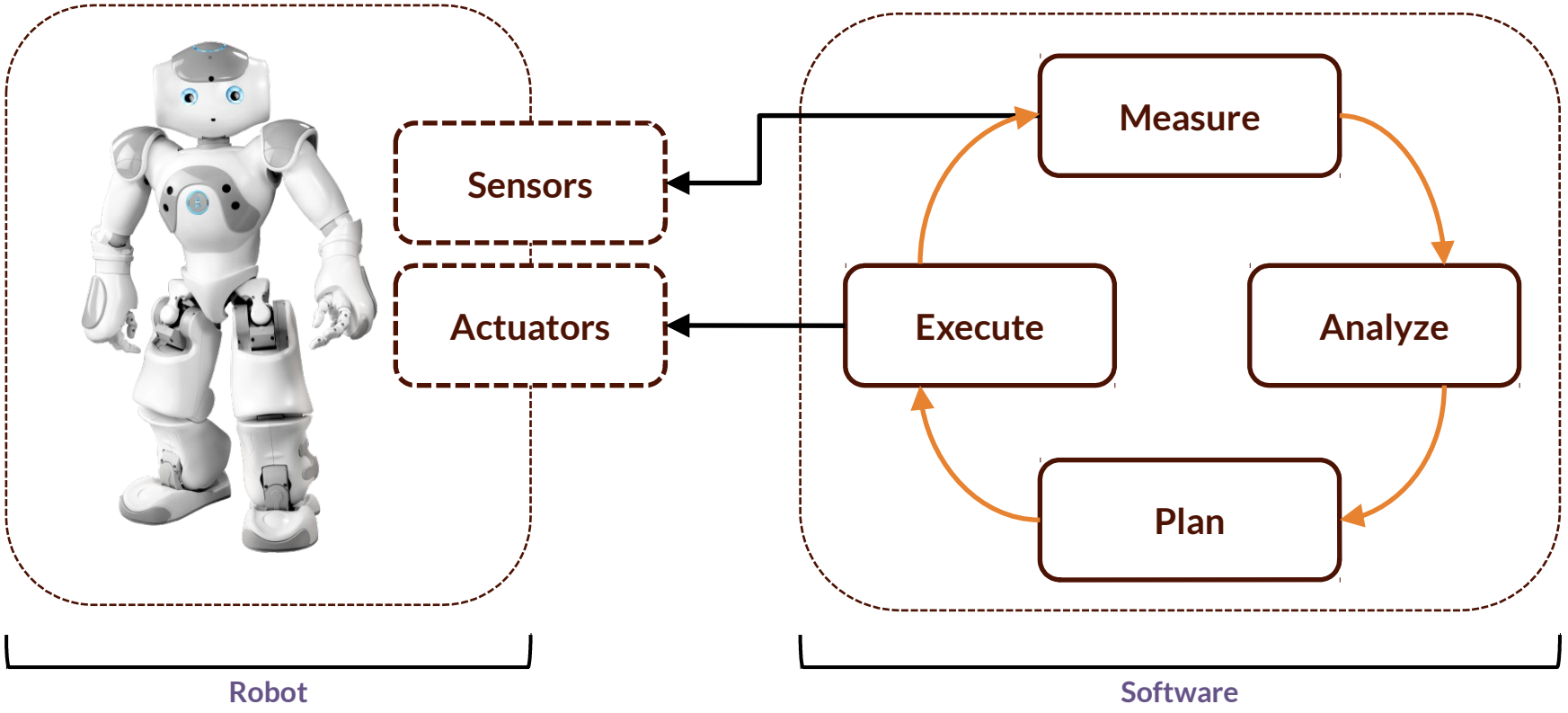
NaoQi (C++)



https://github.com/max-leuthaeuser/naoservice

© Prof. L

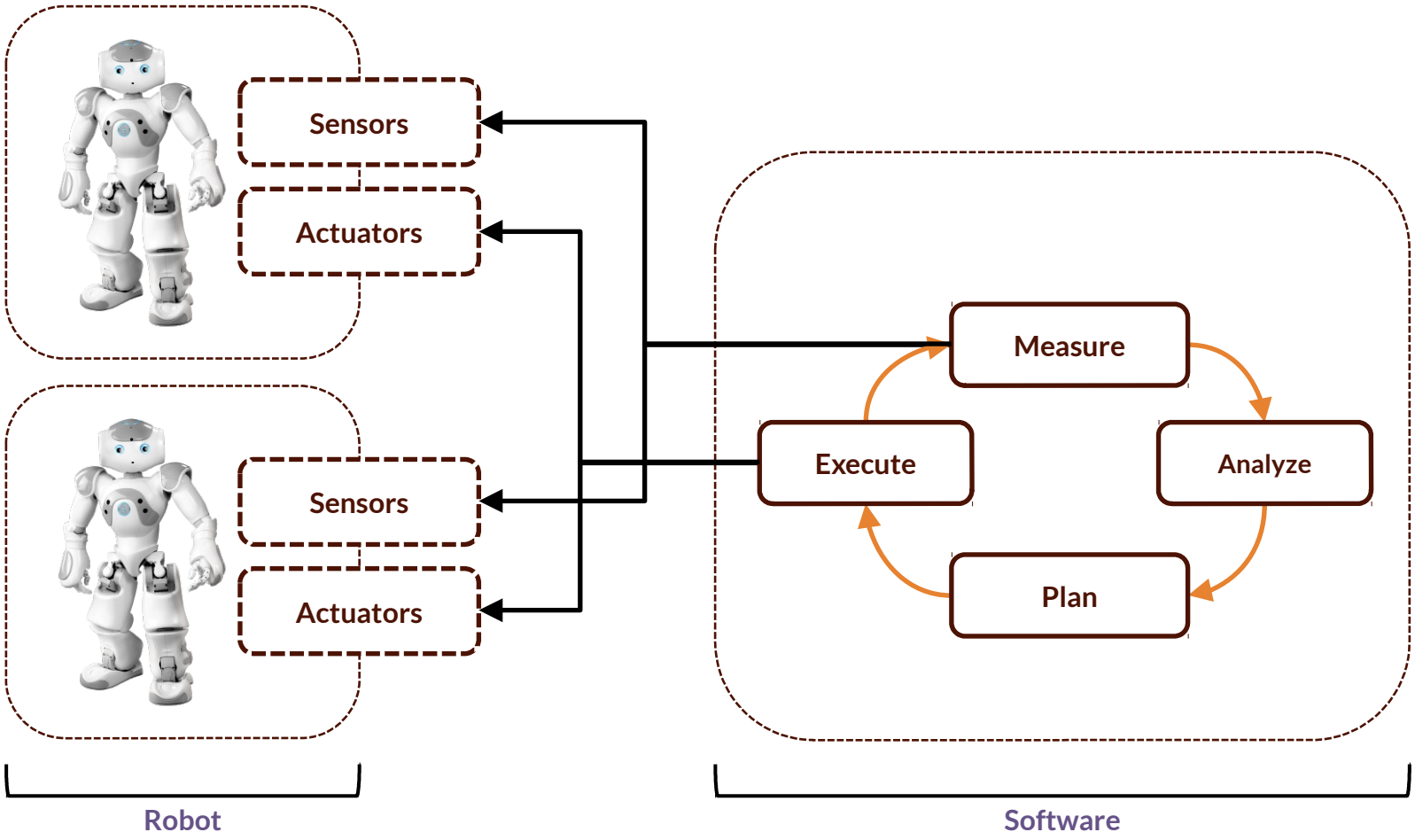
Cloud Robots are Adaptive Systems (MAPE Loop)



Elastic Architecture

© Prof. U. Aßmann

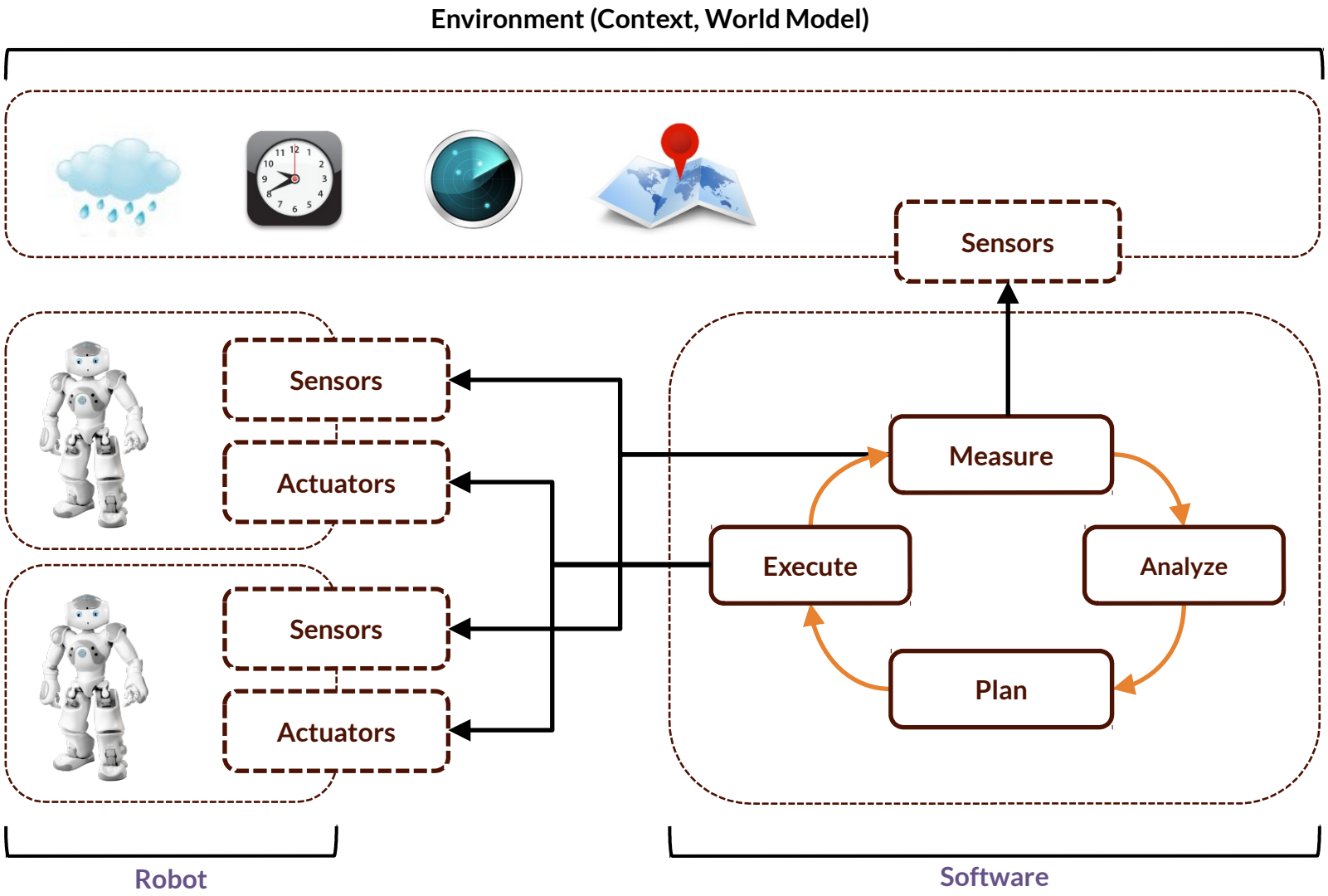
Cloud Robots are Multi-Adaptive Systems



Elastic Architecture

© Prof. U. Aßmann

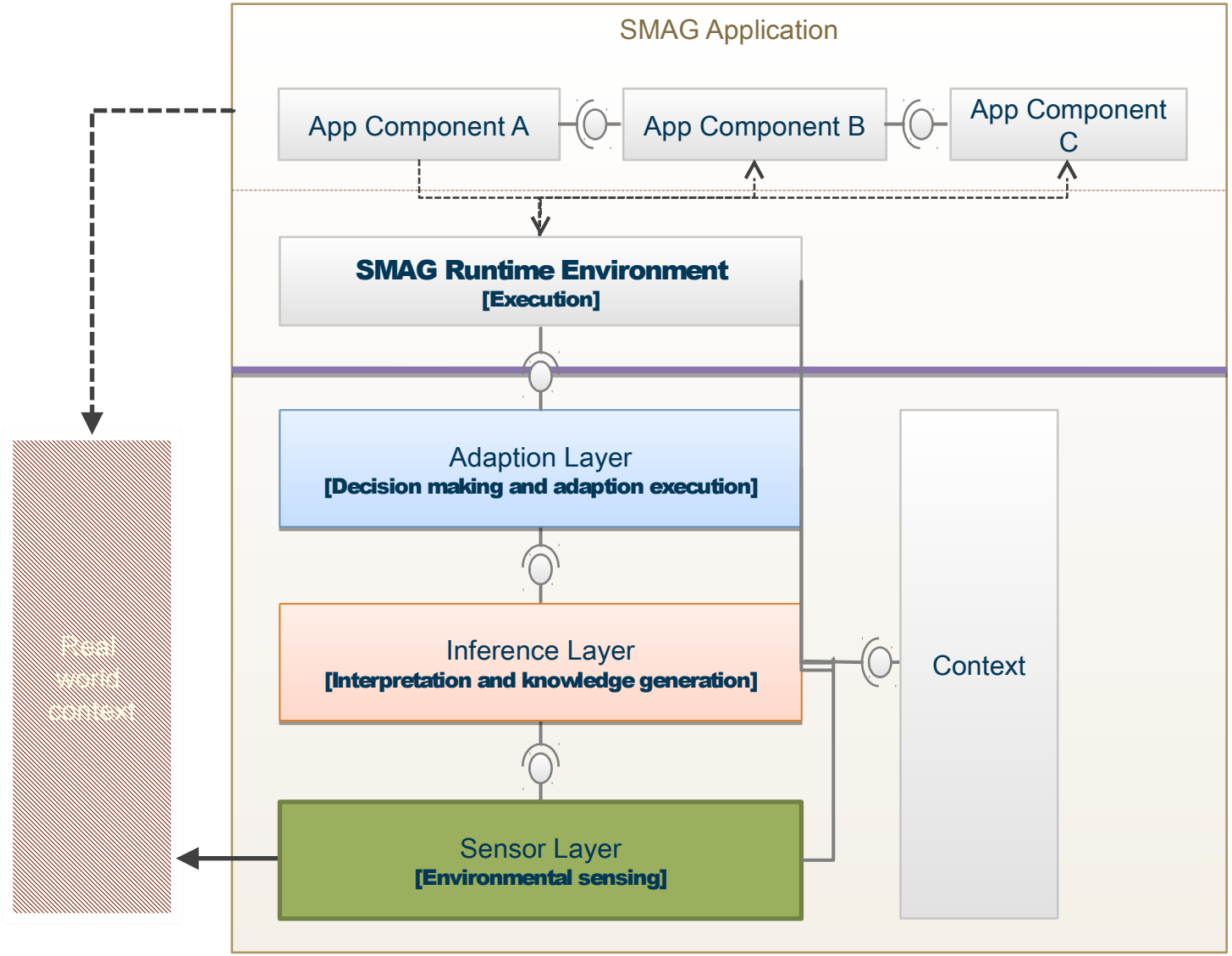
Cloud Robots are Context-Adaptive Systems



Elastic Architecture

© Prof. U. Aßmann

An Adaptive Runtime System: Smart Applications on Smart App Grid Infrastructure



© Prof. U. Aßmann

http://st.inf.tu-dresden.de/smags

2.1.5. A Killer App for Cloud Robots: Donut Production in „Nachtsprung“

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Donuts Should be Individual....

And

https://www.flickr.com/photos/amiga-commodore/10059167335/

© Prof. U. Aßmann

Situation Today

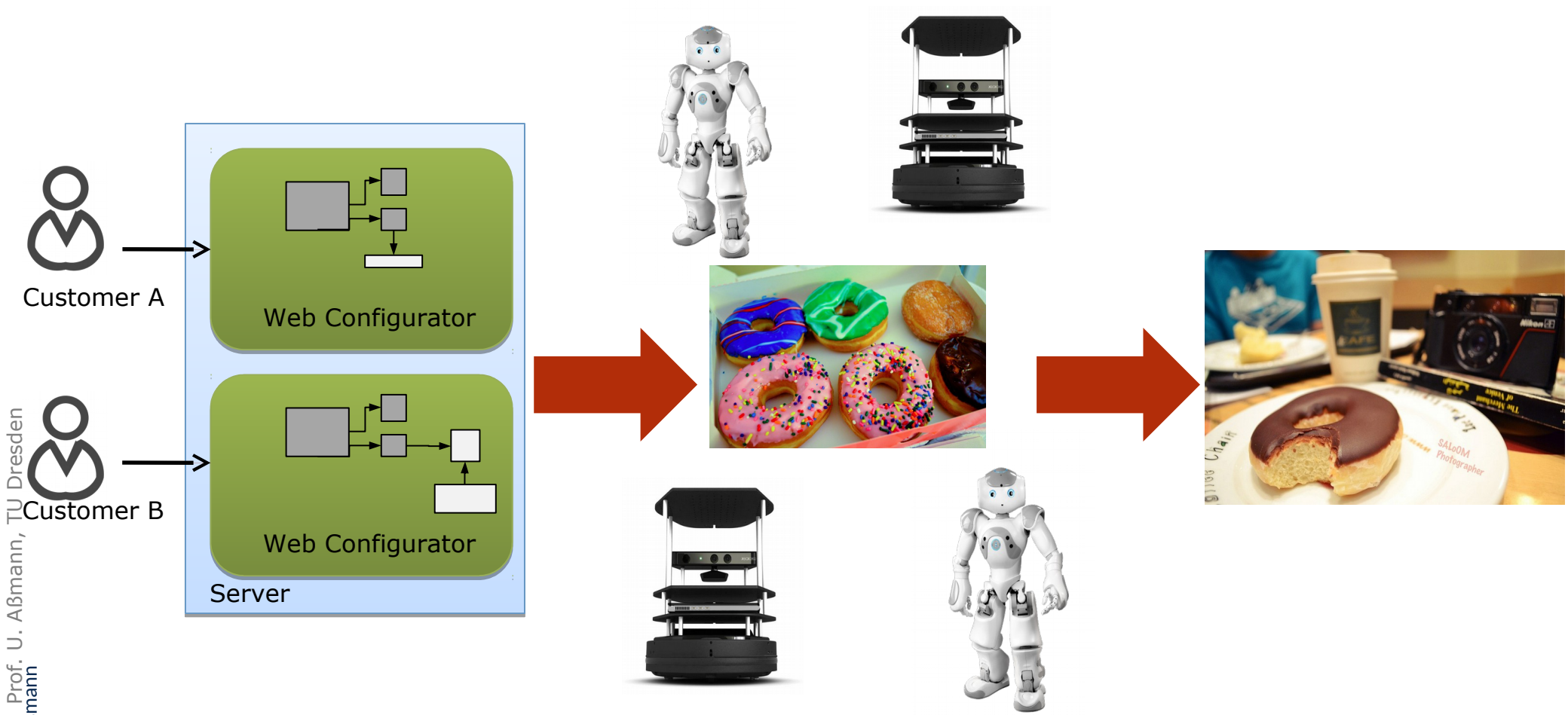
https://www.flickr.com/photos/jeades/2383525381/

- Mass production
- No individual configuration
- No fast, individualized production
- No „Nachtsprung“

© Prof. U. Aßmann

Donut Industry-4.0: Pulling Individual Donuts out in Nachtsprung

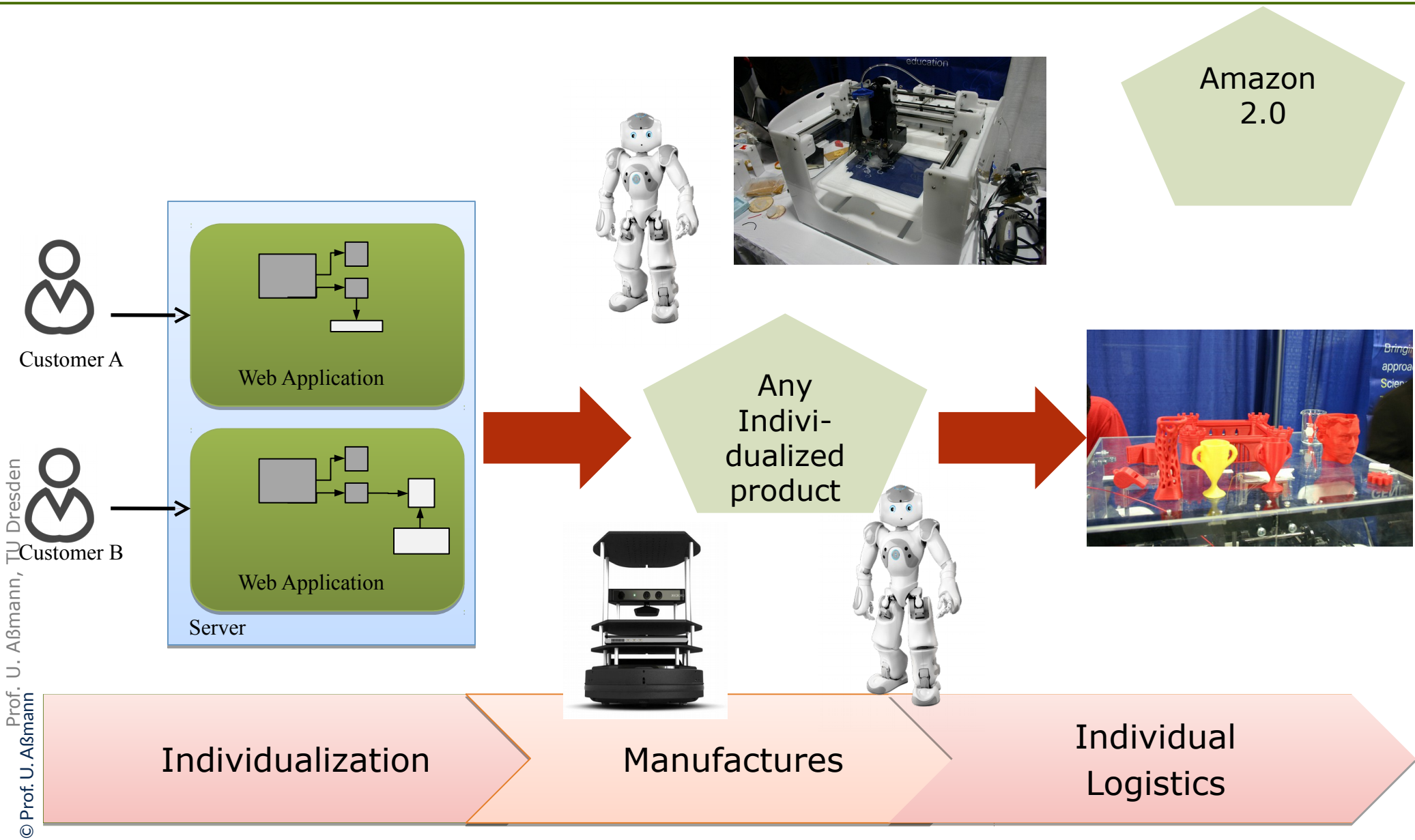
https://www.flickr.com/photos/soso__1991/7179199134/



Configuring in the evening → Producing in the night → Shipping in the early morning

© Prof. U. Aßmann, TU Dresden

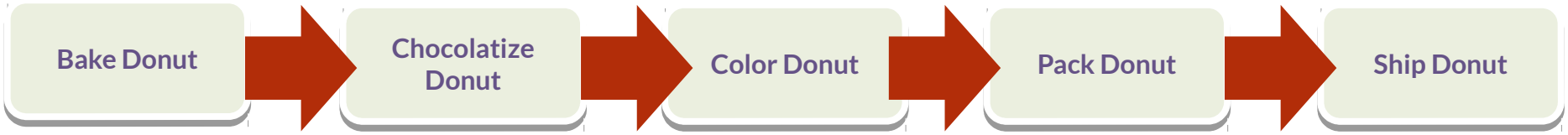
Industry-4.0: Economic Consequences

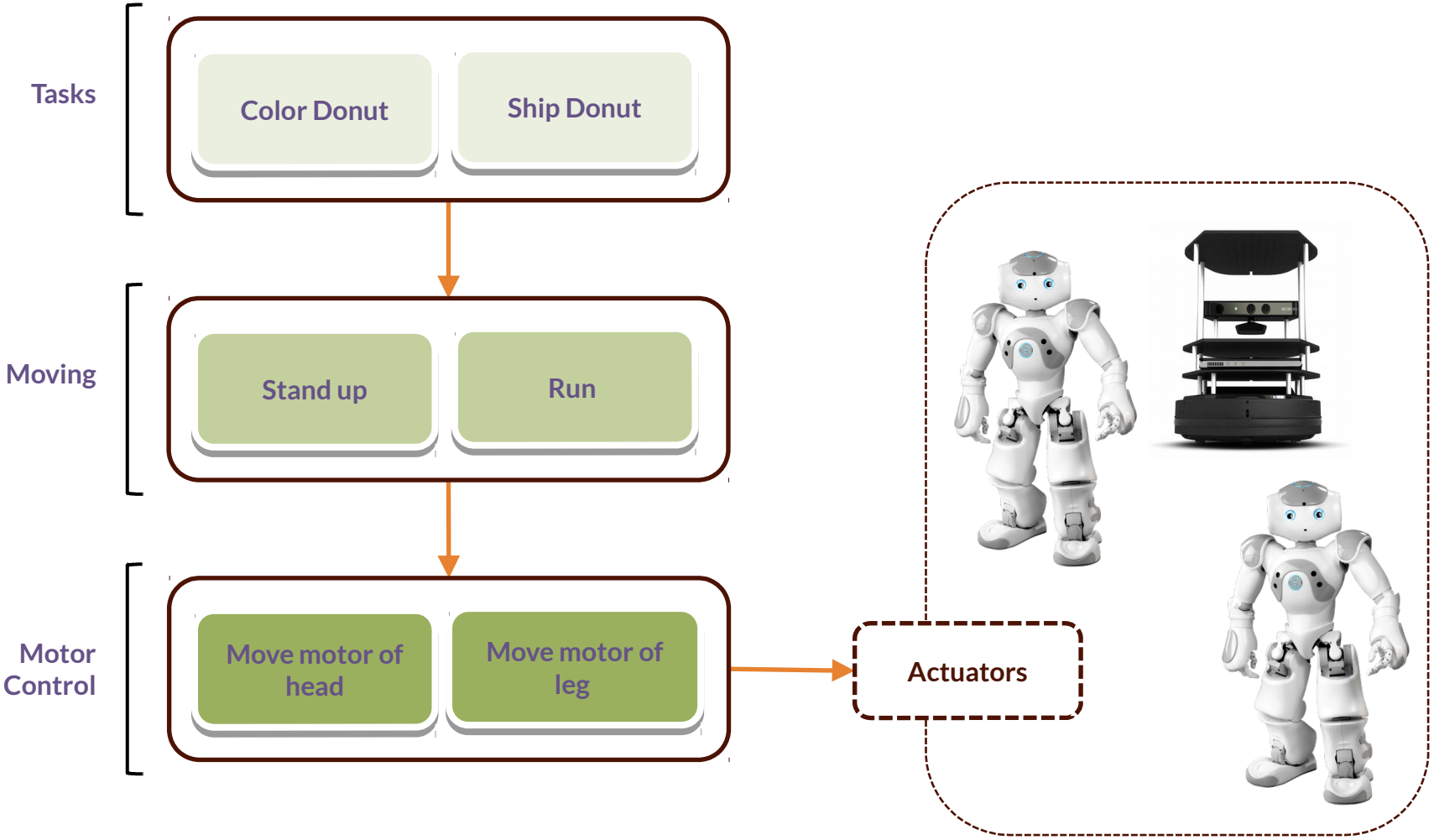


https://www.flickr.com/photos/ideonexus/7311856946/
https://www.flickr.com/photos/ideonexus/7311859510

© Prof. U. Aßmann, TU Dresden

Industry-4.0:
Cloud Robots Produce Things in Workflows





© Prof. U. Aßmann

Industrie-4.0 (Smart Factory) with CPS

- Embedded System: machines, robots, presses, transport systems
- CPS: Autonomous control of the factory
  - Self assembly of the products
  - Autonomous control of logistics
  - Pull of products instead of push

http://commons.wikimedia.org/wiki/File:Mail_sorting_assembly_line.jpg

http://commons.wikimedia.org/wiki/File:Factory_Automation_Robotics_Palettizing_Bread.jpg?uselang=de

© Prof. U. Aßmann

Smart Traffic/Transport/Logistics mit CPS

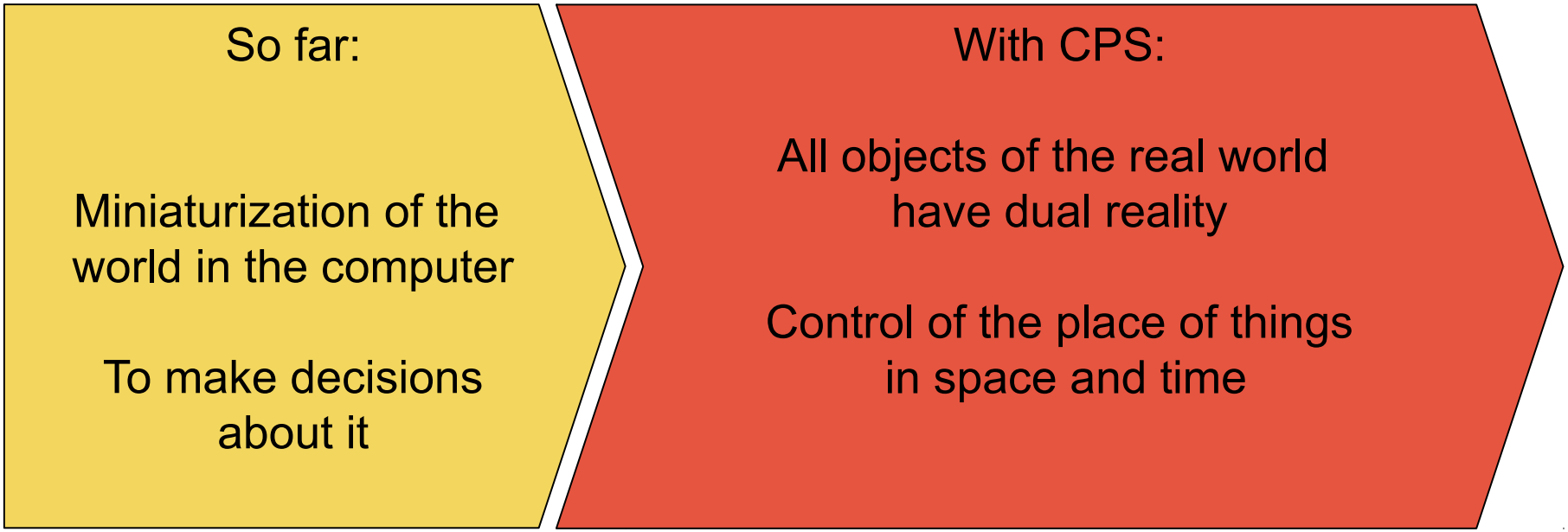
- Embedded System: Railcabs are autonomous train cars (Paderborn)
- CPS: Optimization of the German logistics

http://www.railcab.de

© Prof. U. Aßmann

The Revolution of CPS

- All domains in transport, logistics, assembly, housing, cities will change
- Nothing will stay as it is
- All engineering disciplines will change until 2020



© Prof. U. Aßmann

Questions

How can we build such complex tool suites for CPS (CPS-IDE)?

Answer: By Model-Driven Software Development (MDSD) for software **and** system, with

- Metamodels of languages (on M2)
- Models (on M1)
- Repositories (on M0)

© Prof. U. Aßmann

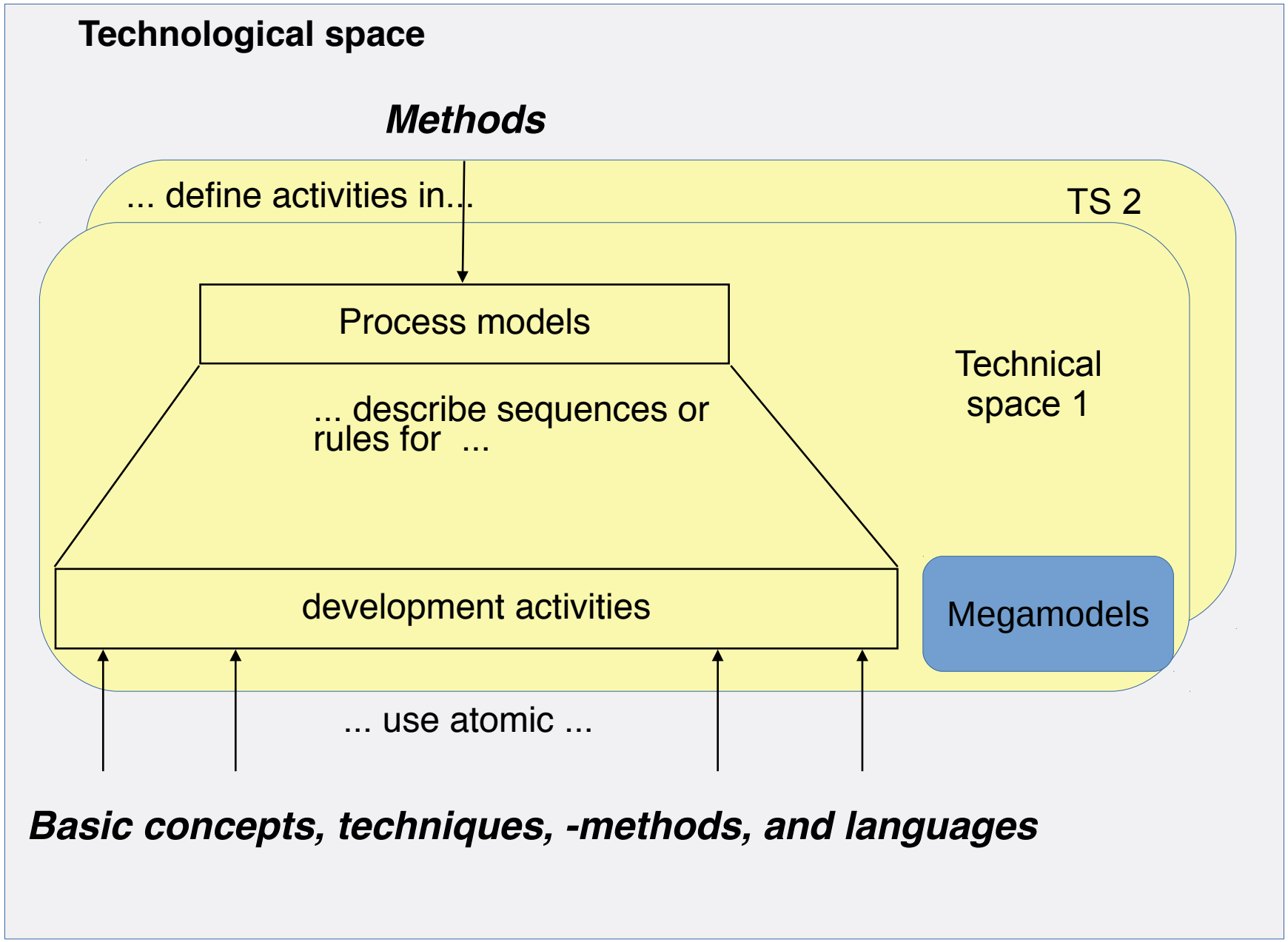
2.2 A Second Class of Big Tool Chains: Integrated Development Environments for Software (MDSD-Software-IDE)

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Software Development is Heterogeneous

[TLS]

[Raasch]



© Prof. U. Aßmann

Method Engineering (Process and Workflow Engineering)

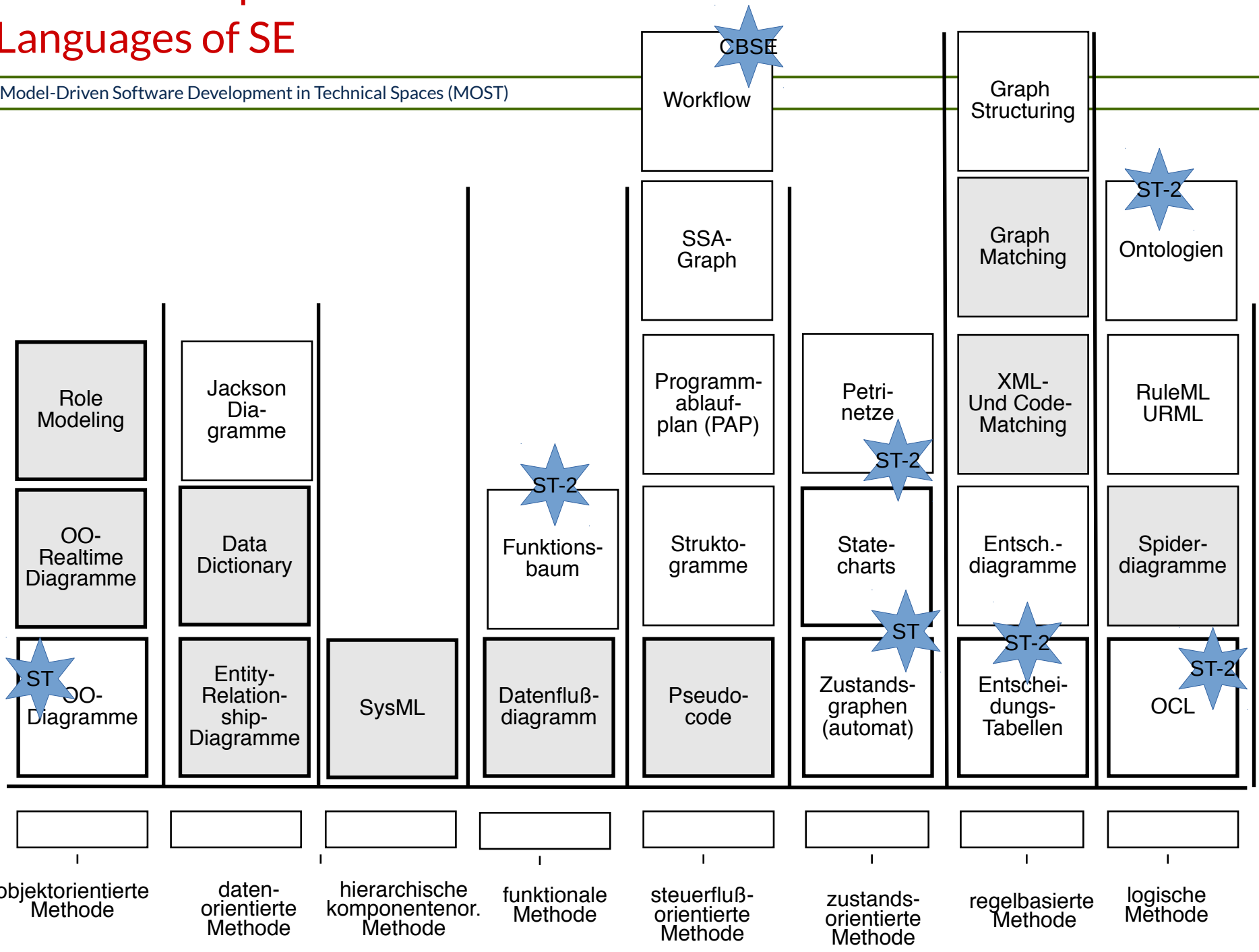
Process Engineering (Method Engineering) is the discipline of specifying and constructing methods and processes for a team of people to conduct a project.
Software Process Engineering (Software Method Engineering) focuses on software development processes.

Workflow Engineering is the discipline of running executable processes (workflows)
- For a team
- In an application

Workflow engineering uses **behavioral languages.**
Workflows are interpreted by automata (workflow engines)

© Prof. U. Aßmann

Basic Techniques and Languages of SE



Quelle: angelehnt an [BAL]

© Prof. U. Aßmann

Building Software Tools for Basic Techniques is Expensive

Tool	Person years	Cost in kEuro
Compiler	1-2	100
Optimizer	1-3	150
Back-End	0.5-1	100
Compiler component framework	20	1000
UML-IDE	5	250
Java-Refactorer	2-4	200
Energy Unit Test-Framework	1	50
Tool for Requirments management	2-4	200
Mobile Phone Test-Framework	2	100

© Prof. U. Aßmann

Questions

How can I reuse simple tools for more complex tools, to support several basic techniques?
How can I compose tools in an MDSD IDE?

Answer: By composing systematically
- Metamodels of base languages (on M2)
- Models (on M1)
- Repositories (on M0)

© Prof. U. Aßmann

2.3 How will We Design Such CPS?

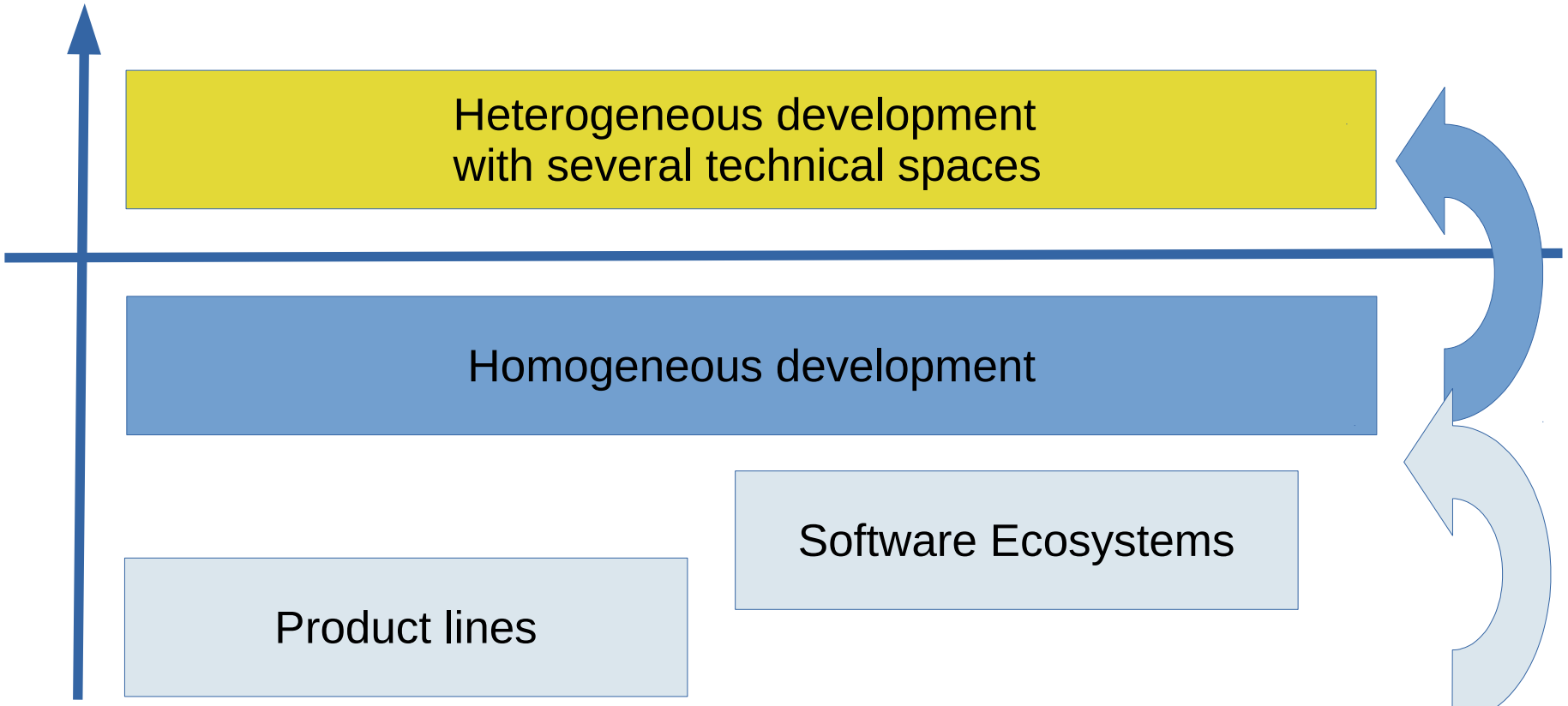
DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

2.3.1 Domain-Specific Design Tools for Design of Cyber-Physical Systems

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

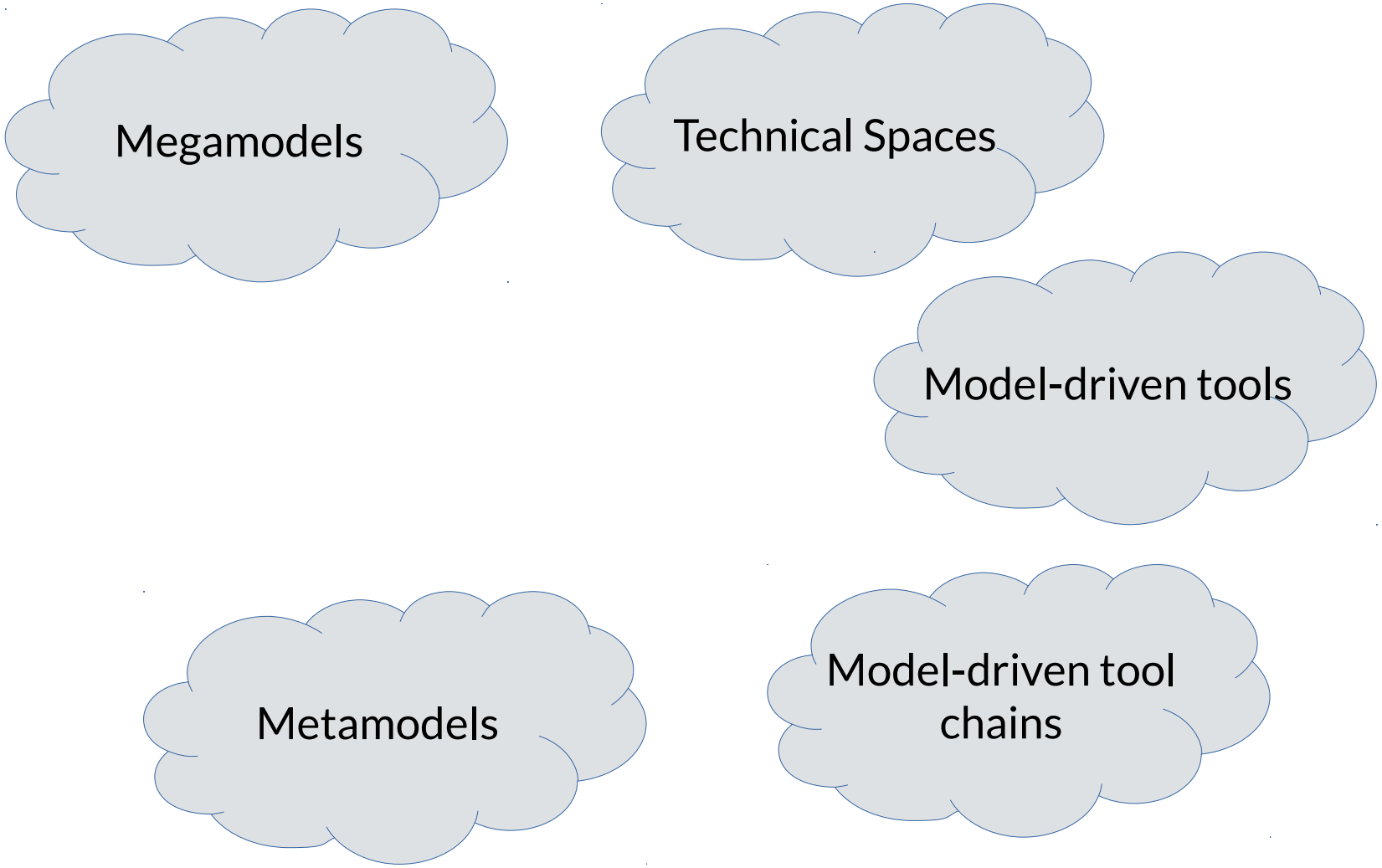
Maturity Levels of Software Companies

- Many companies work with *homogeneous software development* in *one technical space*
- Some companies master *heterogeneous software development in one technical spaces for complex software systems*. Tools are required
- Some companies master *heterogeneous software development in several technical spaces for very complex software systems*. MDSD tool chains are required



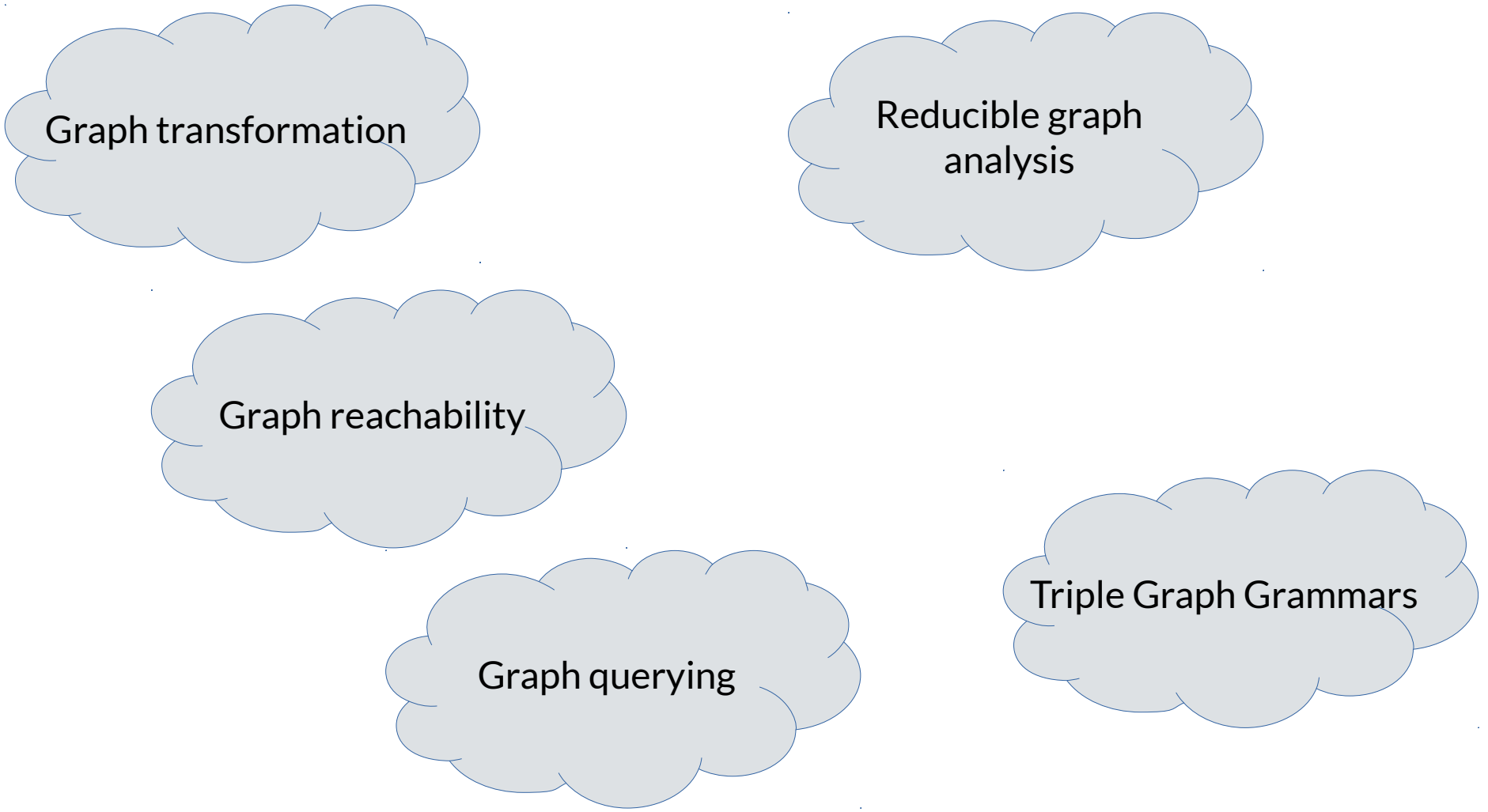
© Prof. U. Aßmann

Concepts of the Course



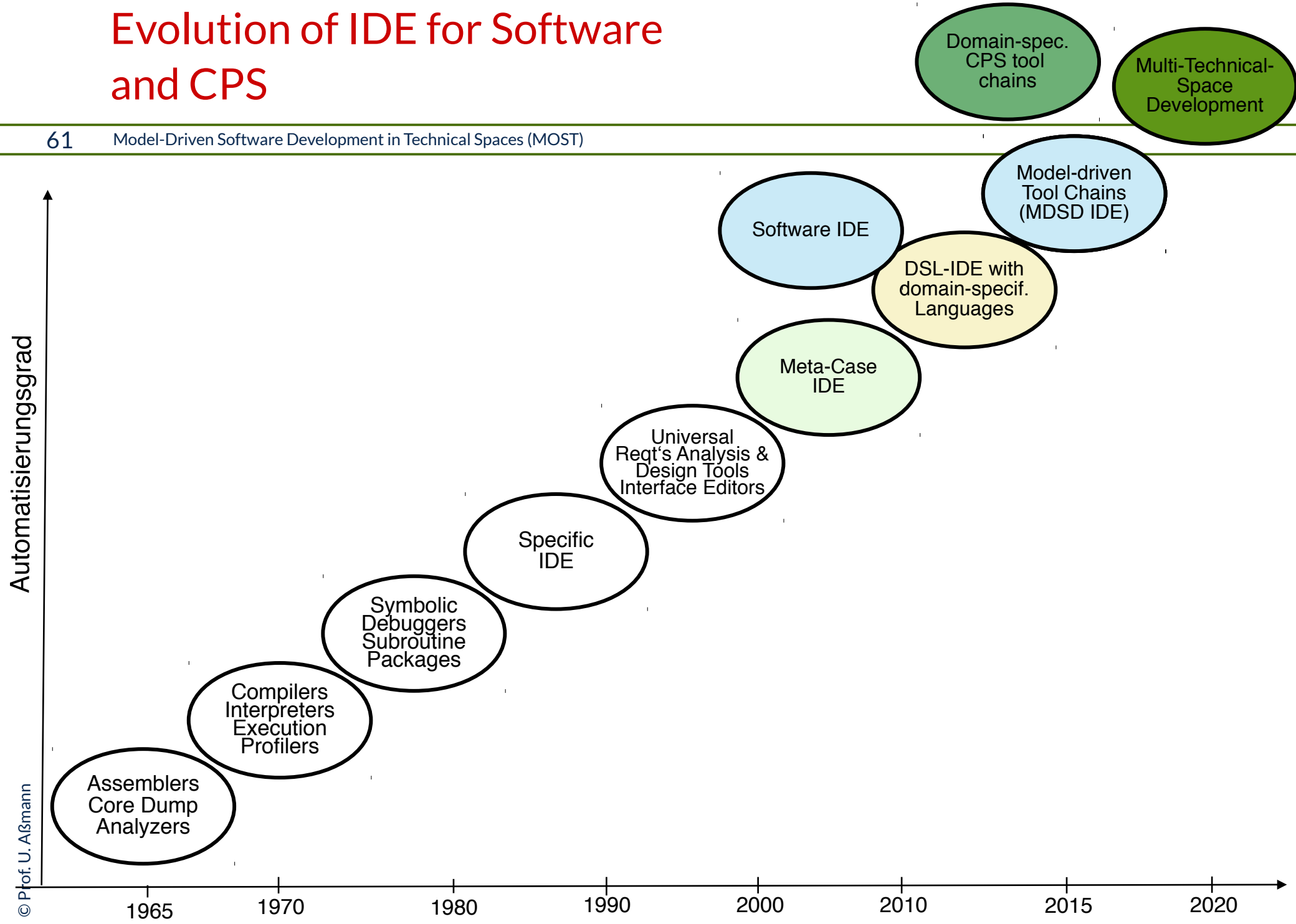
© Prof. U. Aßmann

Concepts of the Course



© Prof. U. Aßmann

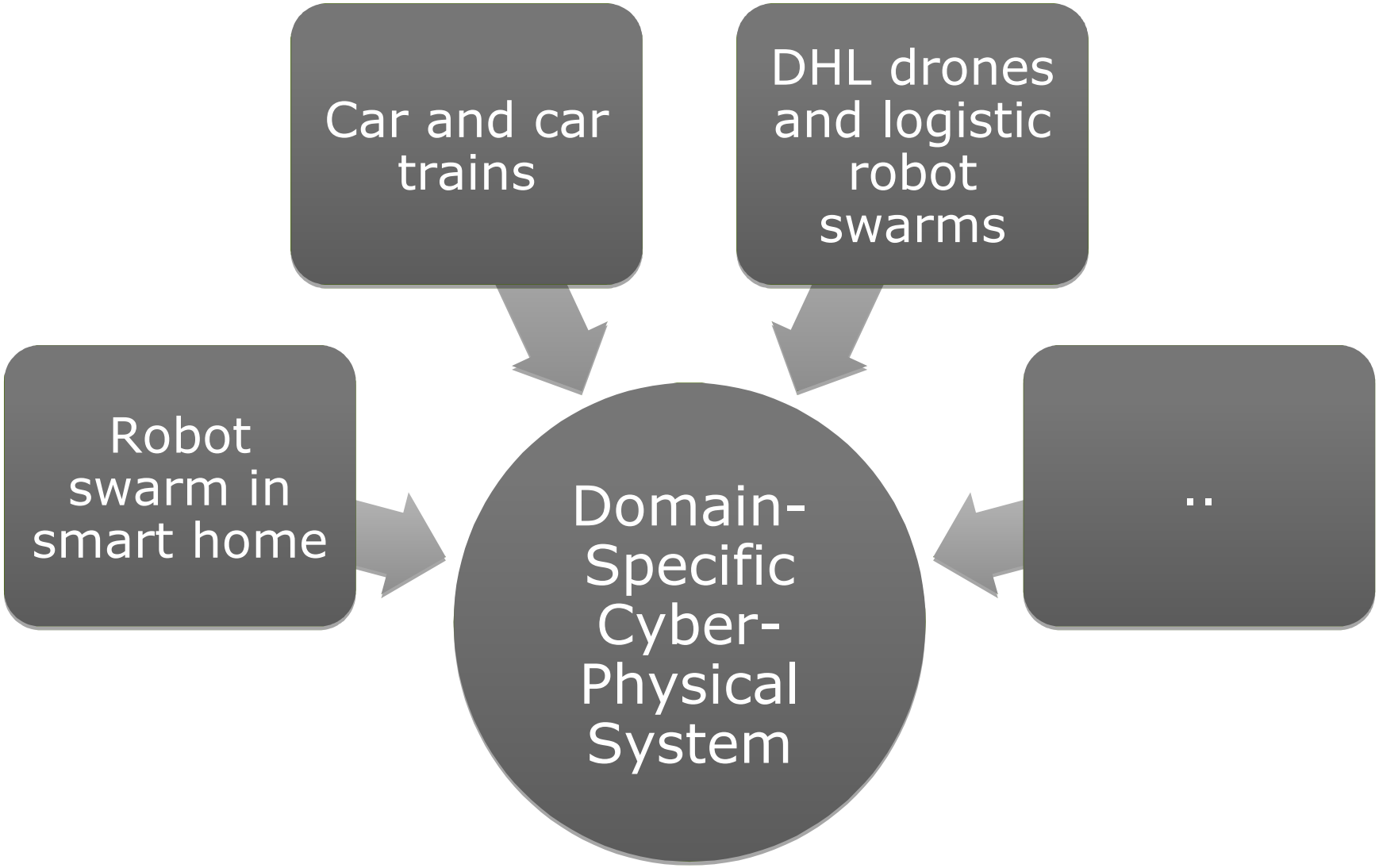
Evolution of IDE for Software and CPS



Quelle: nach [Fisher91, S.20]

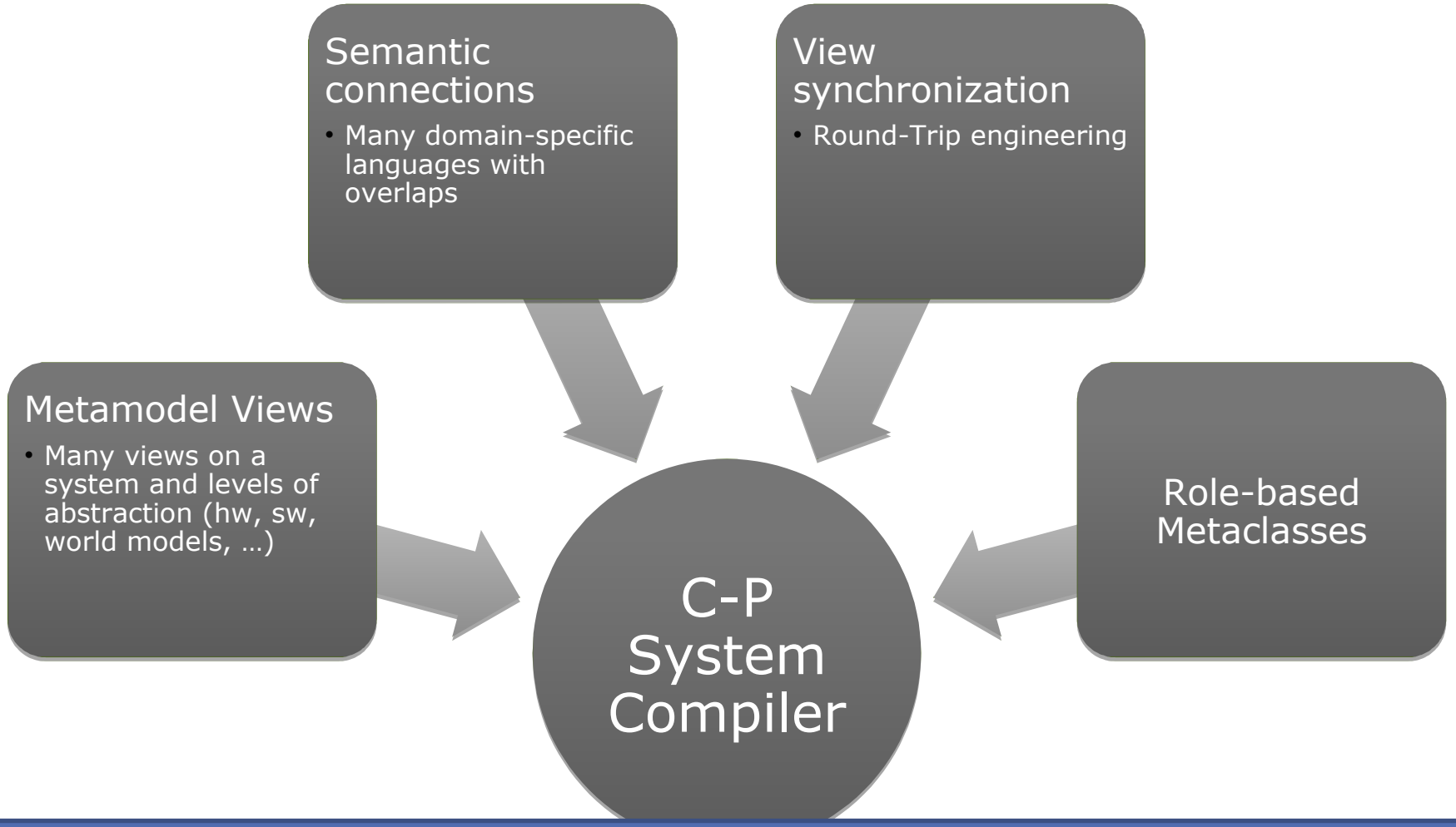
© Prof. U. Aßmann

Domain-Specific CPS



© Prof. U. Aßmann

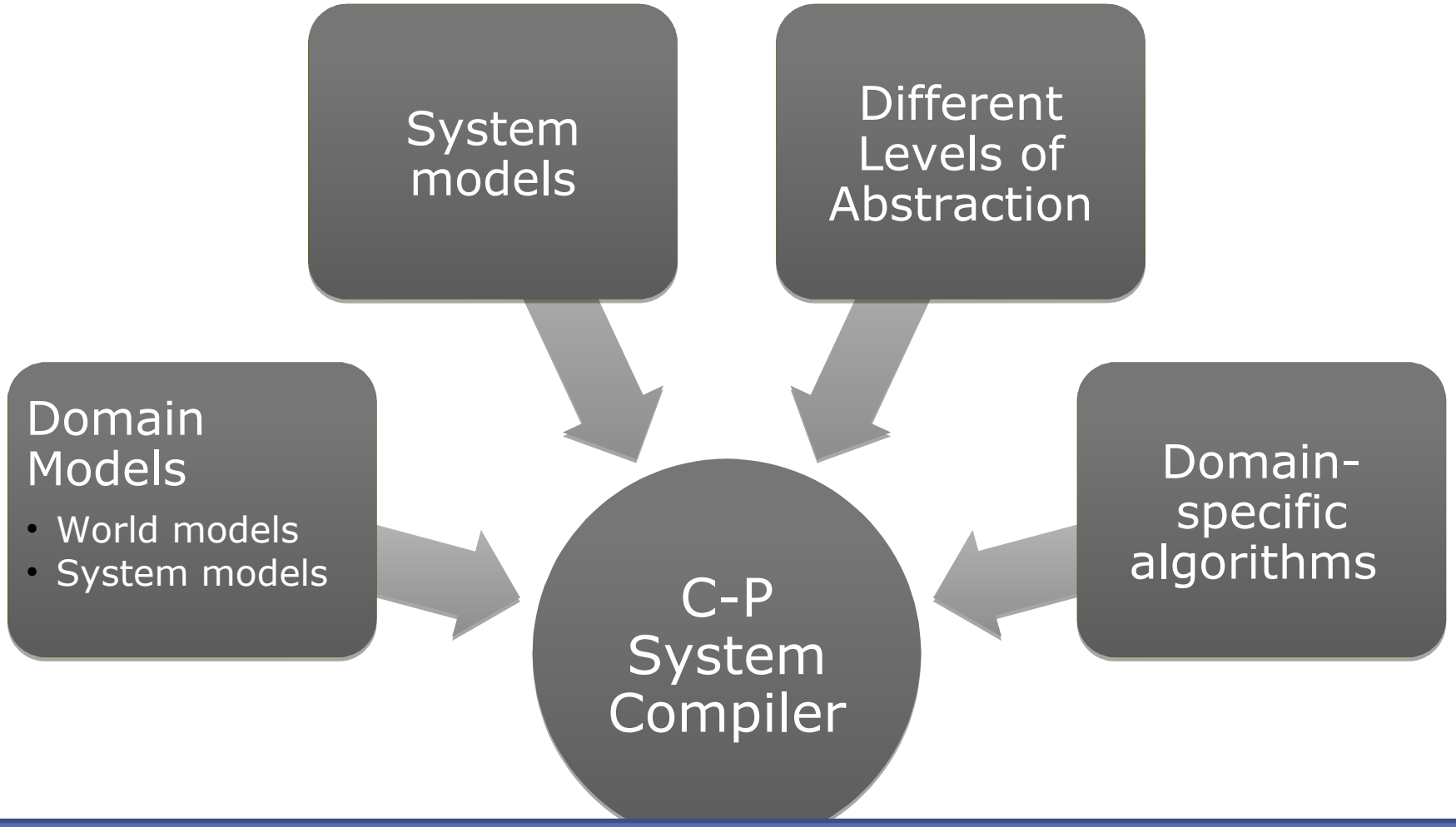
Answer: with Model-Driven CPS Tool Chains (aka "CPS Compilers")



Cyber-Physical System Compilers should be based on Metamodels

© Prof. U. Aßmann, TU Dresden
Prof. U. Aßmann

CPS Compilers are Domain-Specific



Cyber-Physical System Compilers are domain-specific

© Prof. U. Aßmann, TU Dresden
Prof. U. Aßmann

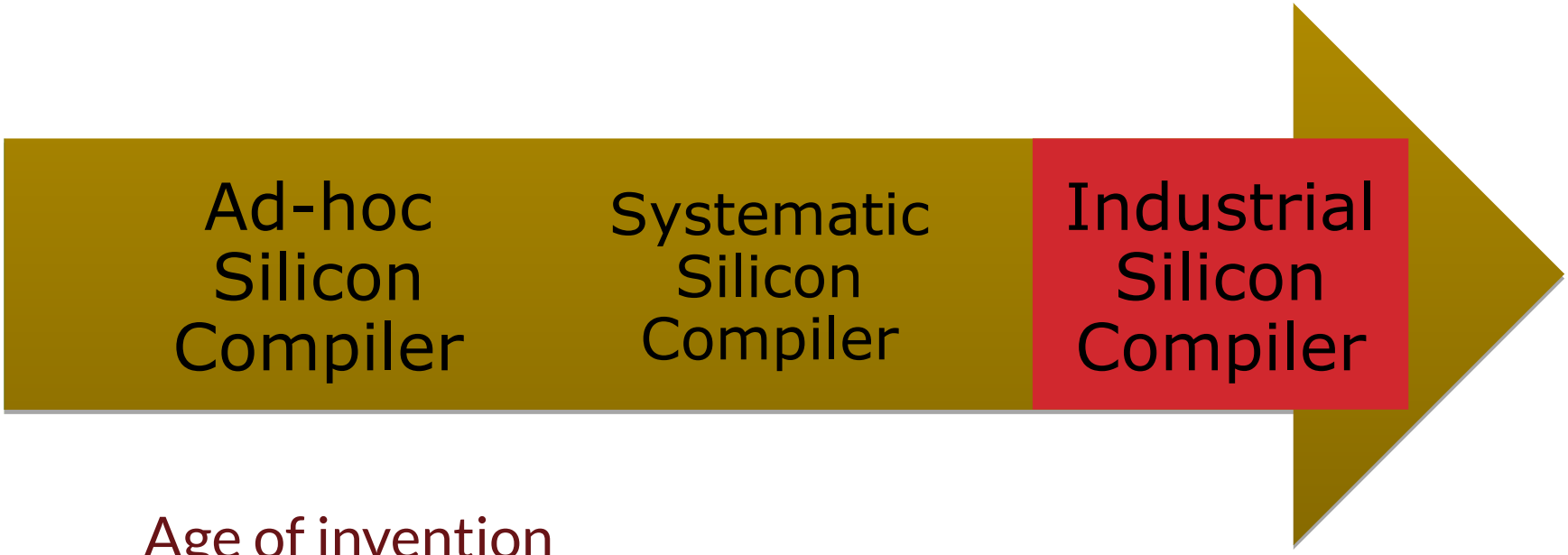
Example 1: MDSD ToolChain: Silicon Compilers

- [Wikipedia:Silicon_Compiler] A **silicon compiler** is a software system that takes a user's specifications and automatically generates an integrated circuit (IC). The process is sometimes referred to as hardware compilation.
- [Wikipedia:Design_flow_(EDA)]
- Alberto Sangiovanni-Vincentelli distinguished three periods of EDA [Tides]:
- **"The Age of Invention:** During the invention era, routing, placement, static timing analysis and logic synthesis were invented.
- **The Age of Implementation:** In the age of implementation, these steps were drastically improved by designing sophisticated data structures and advanced algorithms. This allowed the tools in each of these design steps to keep pace with the rapidly increasing design sizes. However, due to the lack of good predictive cost functions, it became impossible to execute a design flow by a set of discrete steps, no matter how efficiently each of the steps was implemented.
- **The Age of Integration:** This led to the age of integration where most of the design steps are performed in an integrated environment, driven by a set of incremental cost analyzers."

© Prof. U. Aßmann, TU Dresden
Prof. U. Aßmann

Example 1: How the Silicon Compiler Industry Matured over Time

- Sangiovanni-Vincentelli claims that other industries (e.g., for CPS) will go the same way[66]

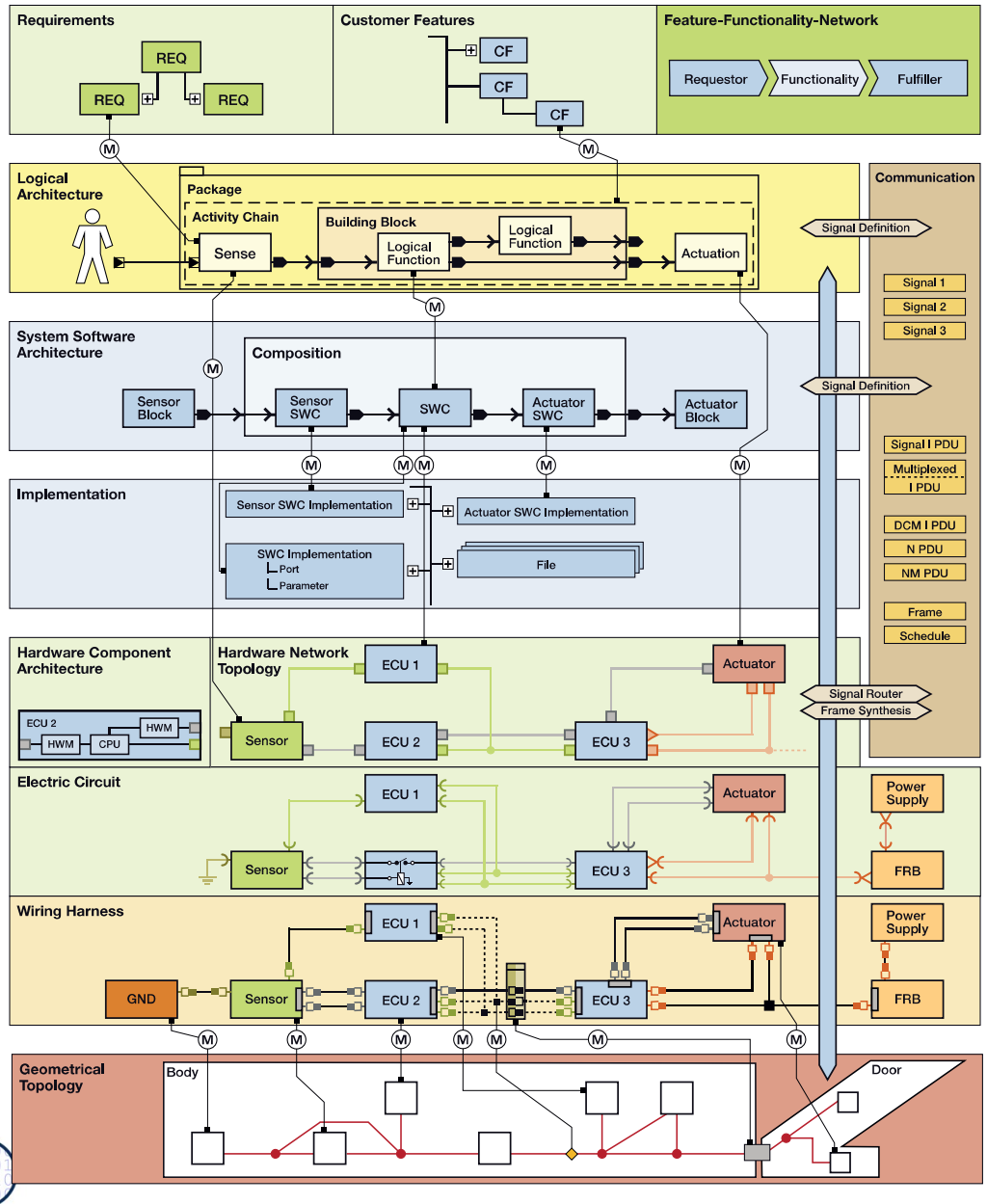


[Sangiovanni-Vincentelli Tides]

© Prof. U. Aßmann

Example 2: Car Design with PREEVision (Vector)

[Preevision]



Excel — Requirements

RIF — Requirements

Logical Architecture

AUTOSAR — System Software Architecture

Simulink, ASCET — Implementation

FIBEX, DBC — Hardware Architecture

LDF — Hardware Architecture

ELOG — Electric circuit

KBL — Wiring harness

3D-CAD Daten (KBL) — Geometric topology

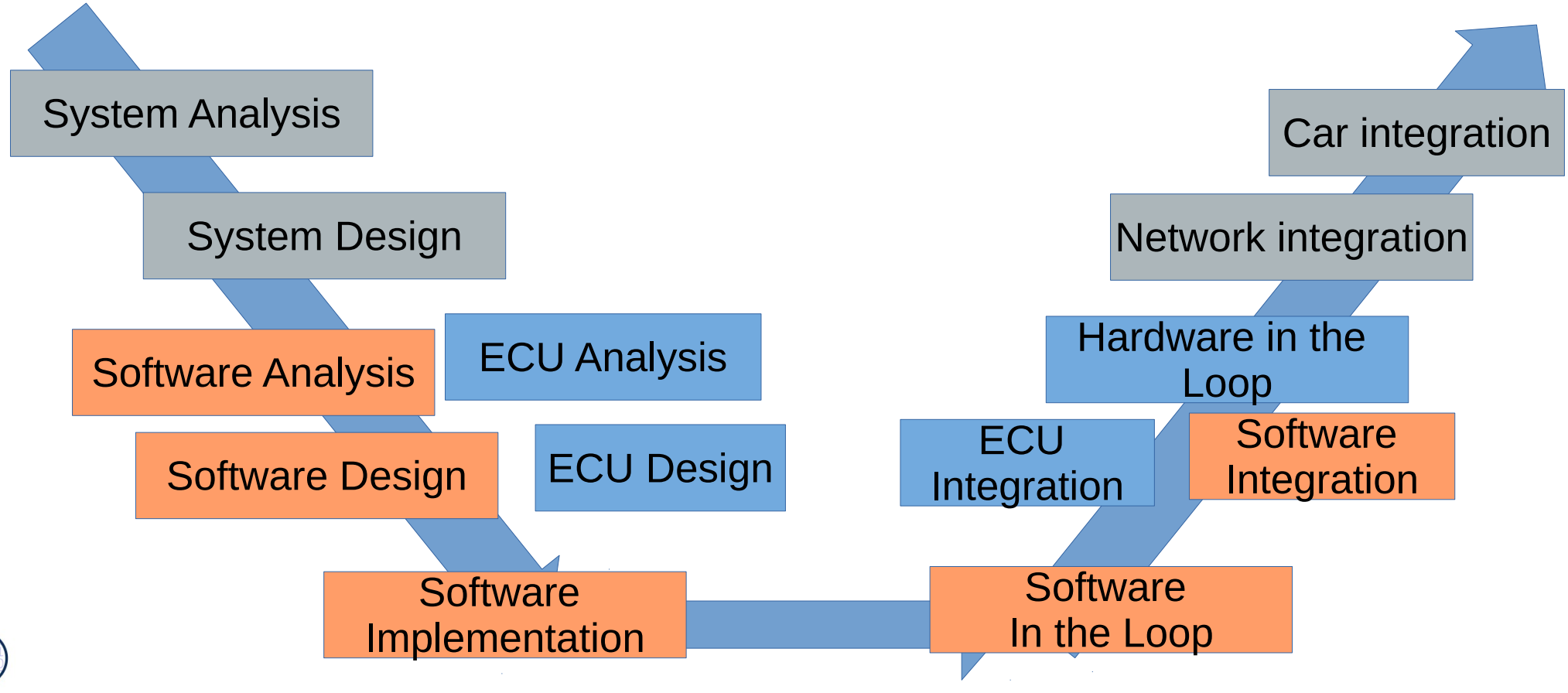
© Prof. U. Aßmann

PreeVision has 3 Tools Steered by Metamodels

[MüGl09]

- PREEvision Architect
- PREEvision Function Designer
- PREEvision Electric Designer

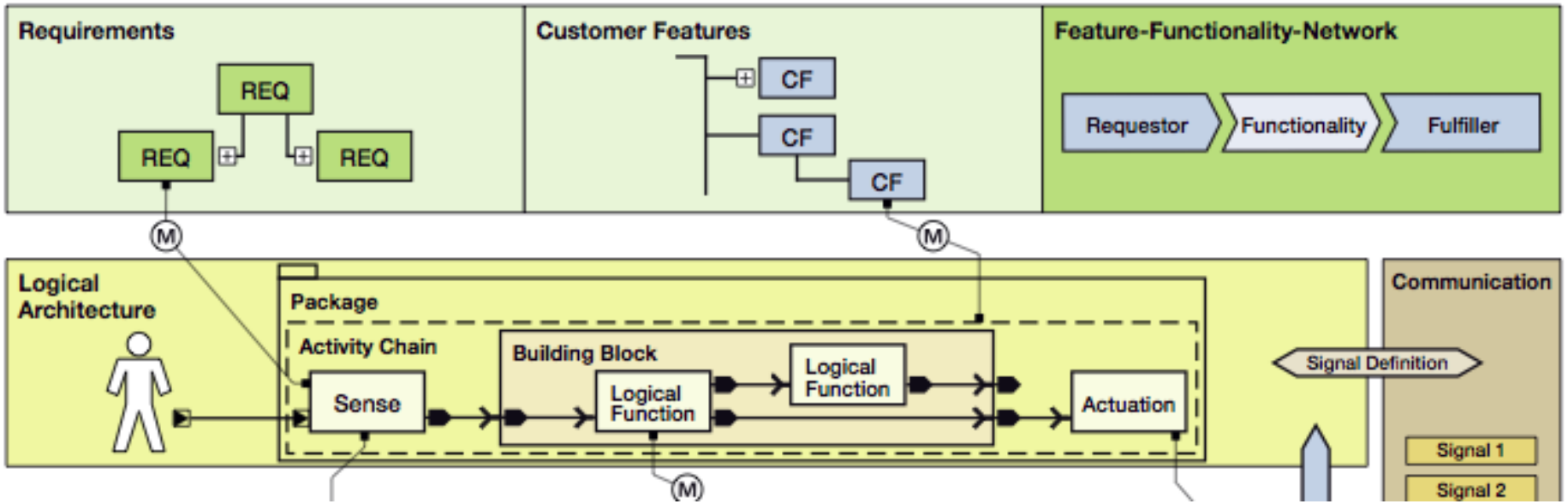
- With options:
  - vTESTcenter
  - PREEvision Collaboration Platform
- All involved models are metamodeled



© Prof. U. Aßmann

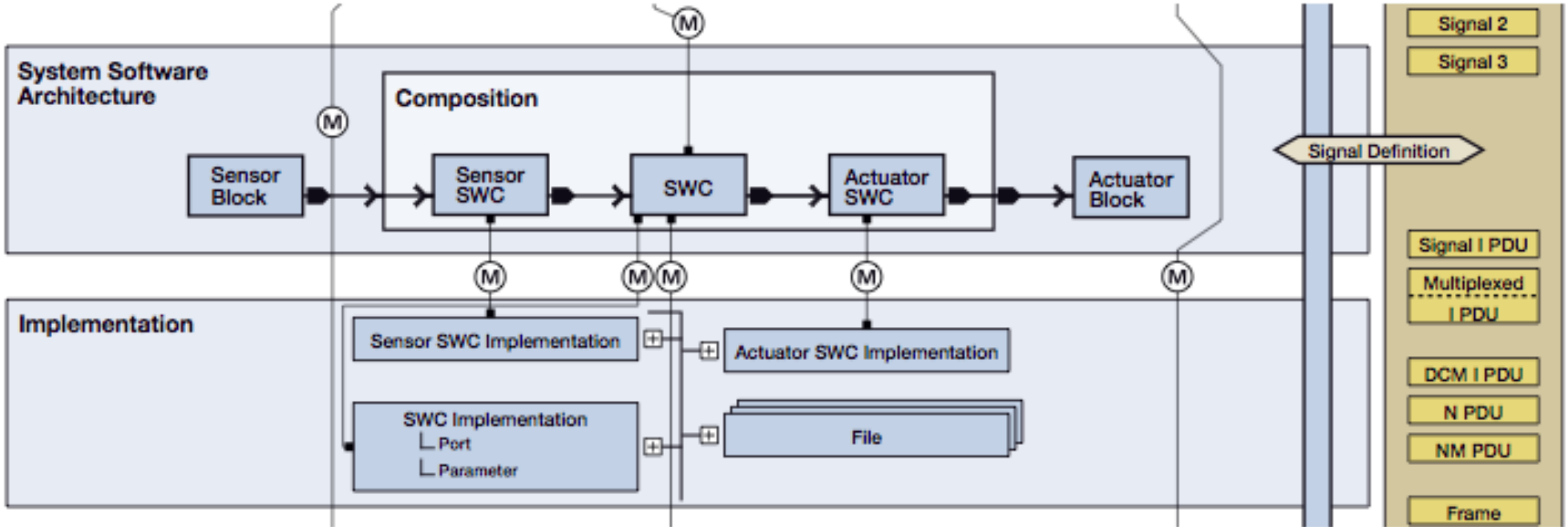
PreeVision Models in More Details

- Requirements specification with Excel and Requirements Interchange Format (RIF)
- Logical architecture with AUTOSAR components



PreeVision Models in More Details

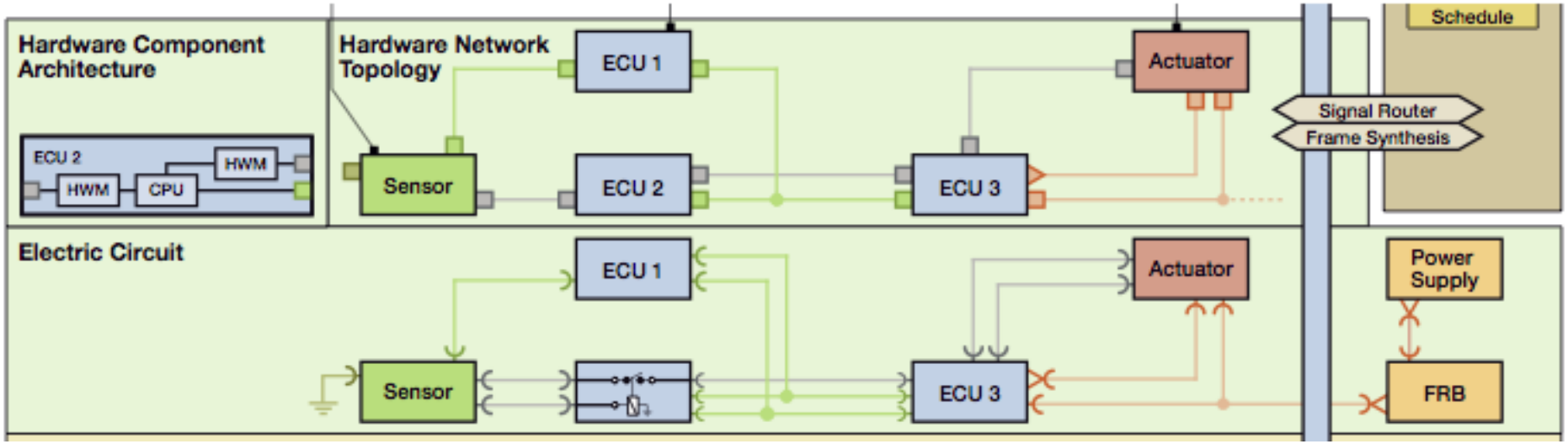
- Software Architecture with Simulink components (blocks) and ASCET model components (from ETAS)
- Implementation (generated or hand written)



© Prof. U. Aßmann

PreeVision Models in More Details

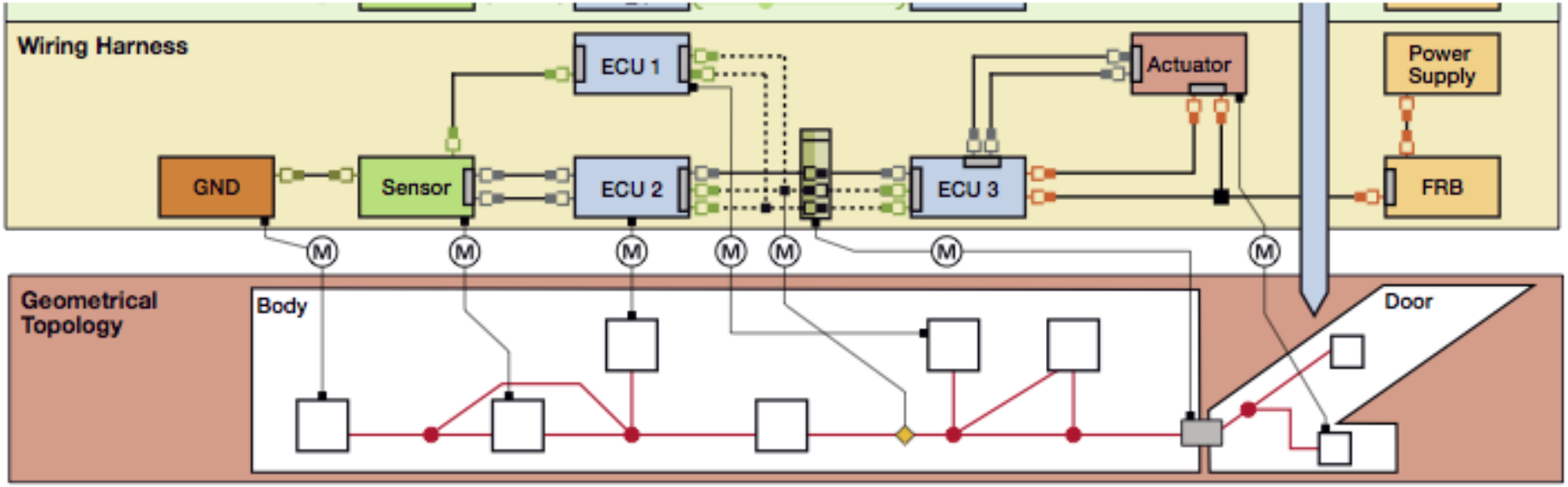
- Hardware architecture with LDF component model
- Electronic circuit design in ECU by ELOG



© Prof. U. Aßmann

PreeVision Models in More Details

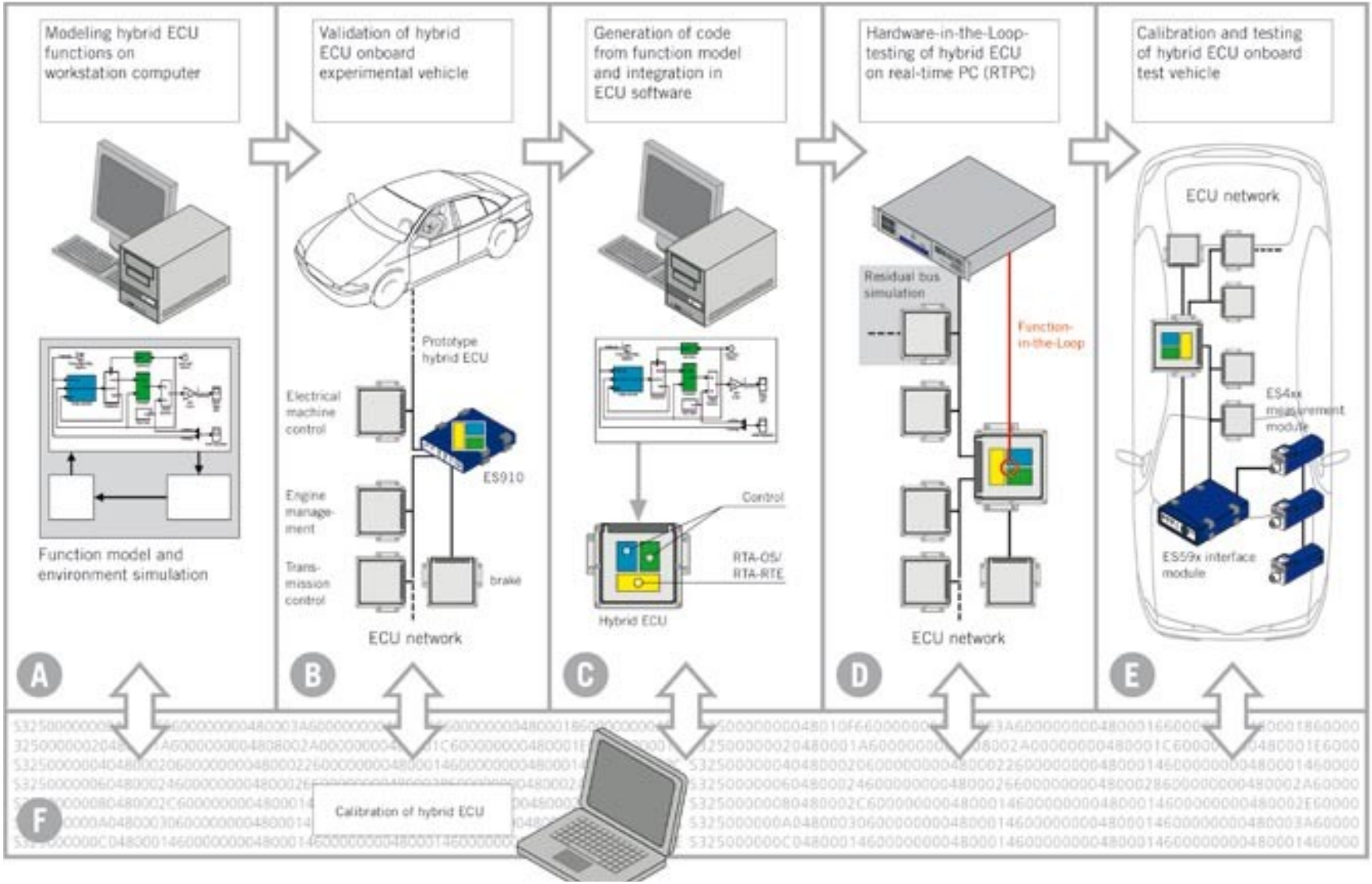
- Wiring in the car (physical network) with KBL
- 3-D CAD drawings for geometrical topology



© Prof. U. Aßmann

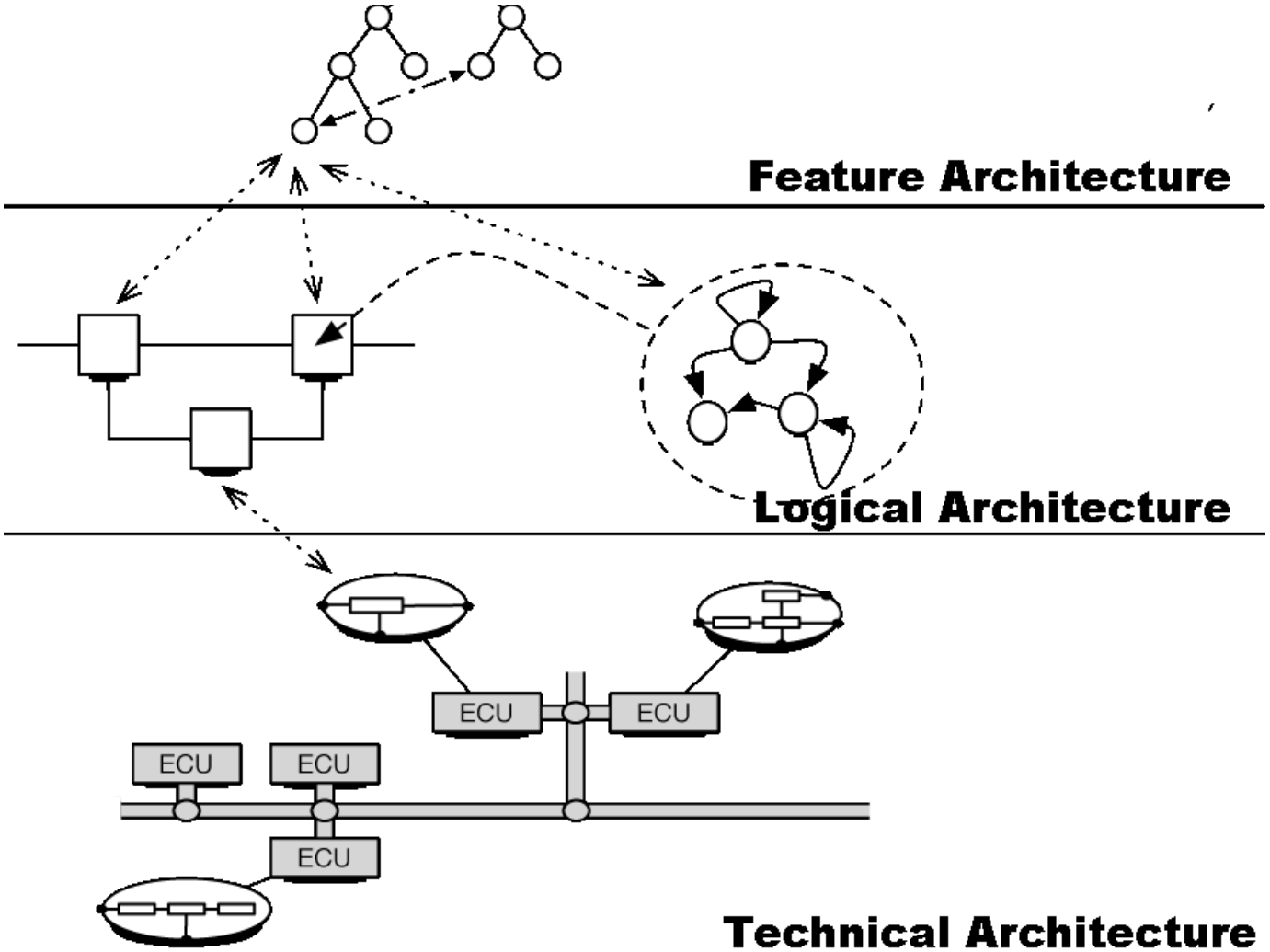
Electric Cars (ETAS)

[ETAS] http://www.etas.com/en/products/ascet_md_modeling_design.php



© Prof. U. Aßmann

COLA (TUM)



© Prof. U. Aßmann, TU Dresden

CPS IDE (CPS Tool Chains) are a Sign of a Maturing Productivity Industry

Age of invention

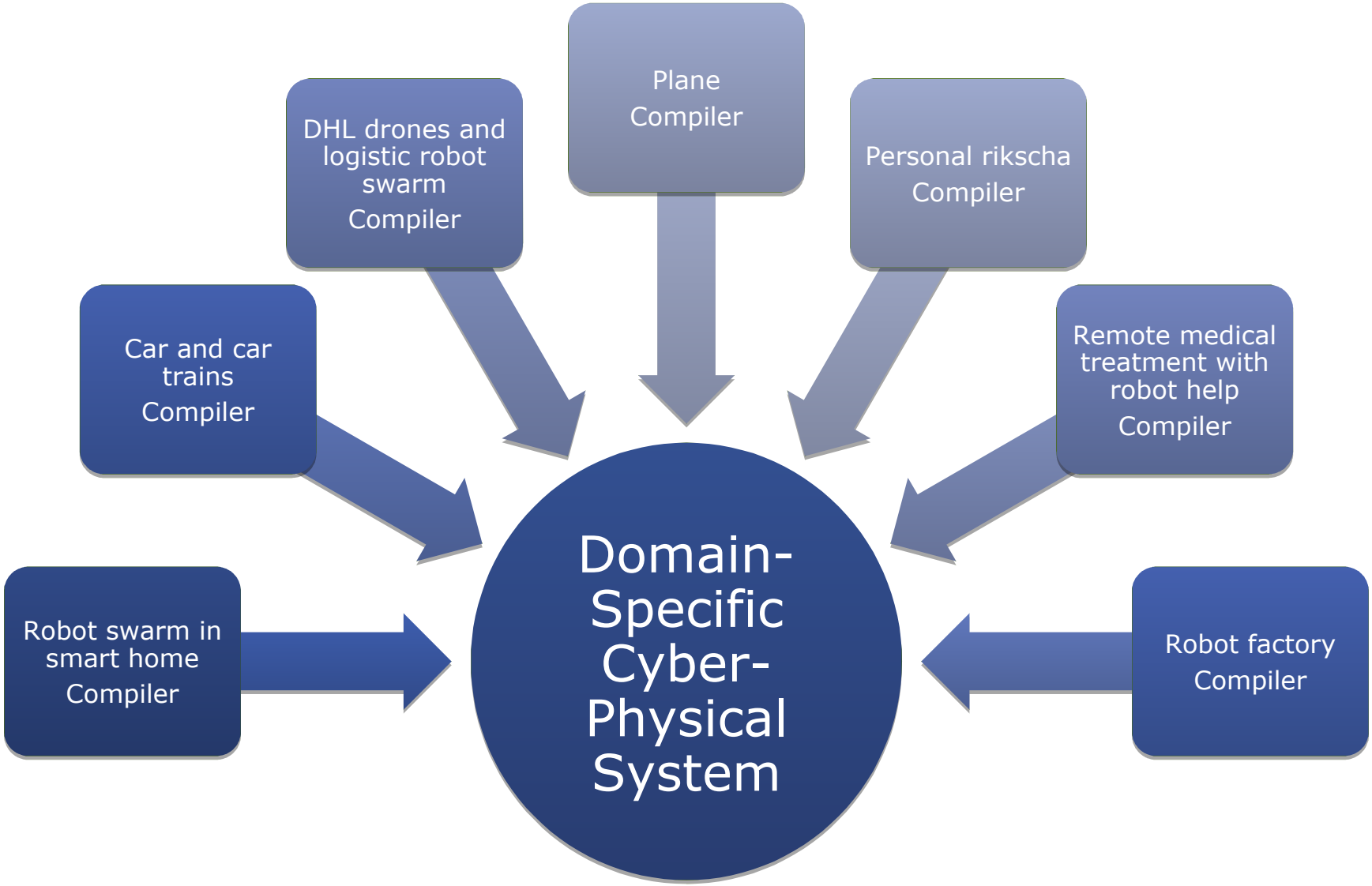
Age of implementation

Age of integration

Will hold for all domains of CPS!

© Prof. U. Aßmann, TU Dresden
Prof. U. Aßmann

We will design Domain-Specific CPS with Domain-Specific CPS-IDE



© Prof. U. Aßmann, TU Dresden

Domain-Specific CPS-IDE are Industry-Critical

CPS-IDE are strategic

- View-based Language Engineering
- Basis of all Cyber—Physical Systems

CPS IDE are domain-specific

- Domains are isolated

Business concepts for domain-specific CPS IDE

- Business for every domain

A project for CPS IDE will create future industries…

- Who will have the CPS IDE?

© Prof. U. Aßmann

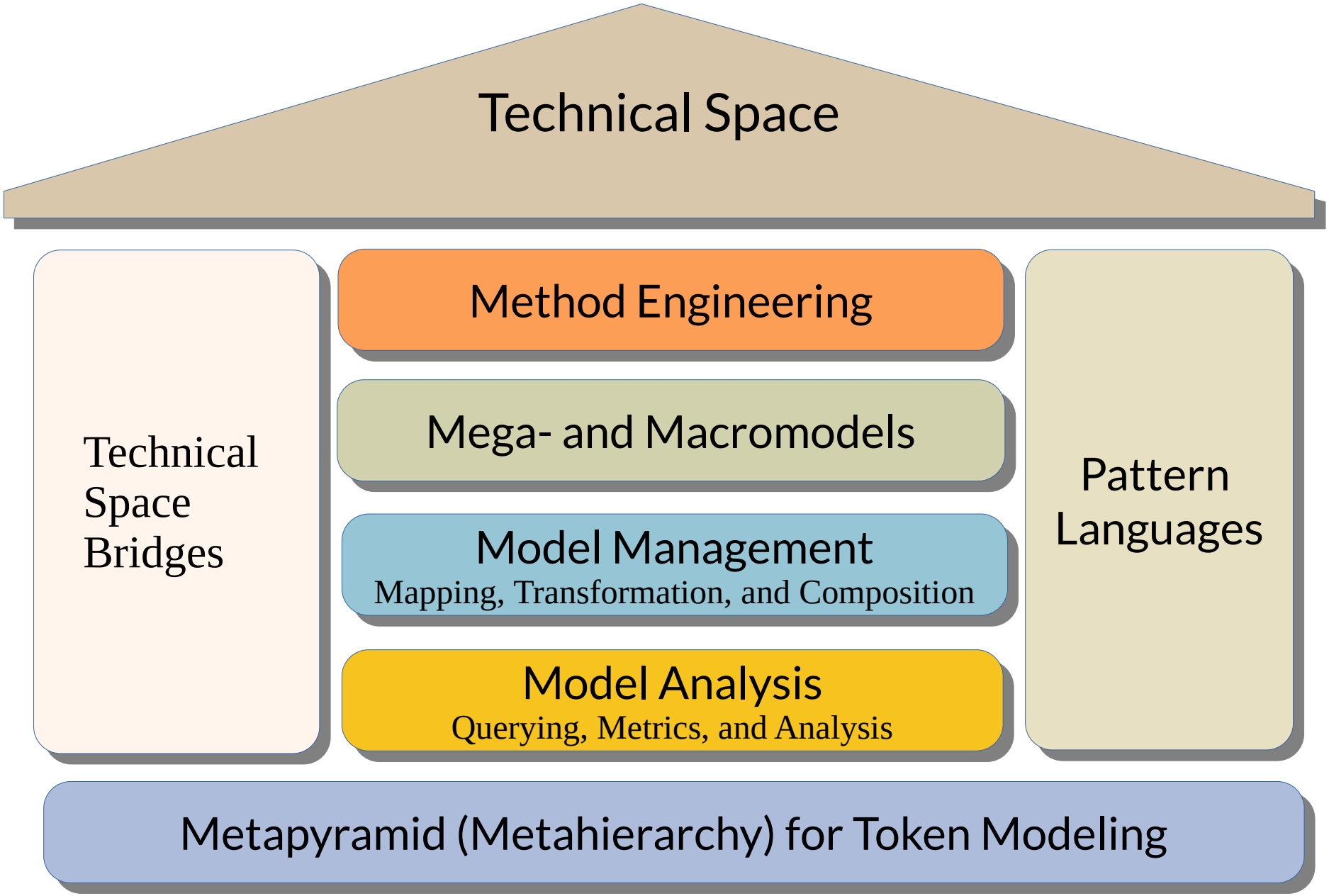
2.4 Why Do We Need MDSD in TS?

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Intelligent Design Tools

© Prof. U. Aßmann

Q10: The House of a Technical Space



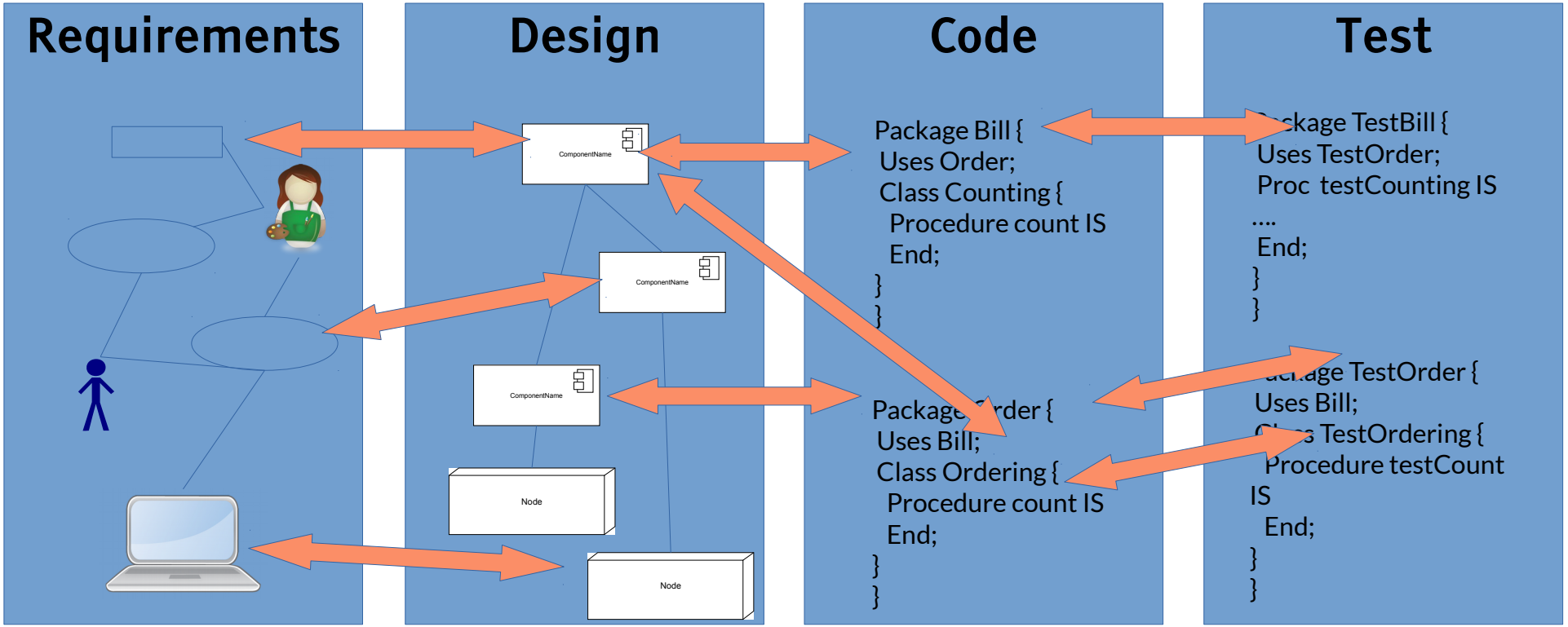
© Prof. U. Aßmann

Q11: Overview of Technical Spaces in the Classical Metahierarchy

	Grammarware (Strings)	Text-ware	Table-ware		Treeware (trees)			Graphware/Modelware			Role-Ware	Ontology-ware
	Strings	Text	Text-Table	Relational Algebra	NF2	XML	Link trees	MOF	Eclipse	CDIF	MetaEdit+	OWL-Ware
M3	EBNF	EBNF		CWM (common warehouse model)	NF2-language	XSD	JastAdd, Silver	MOF	Ecore, EMOF	ERD	GOPPR	RDFS OWL
M2	Grammar of a language	Grammar with line delimiters	csv-header	Relational Schema	NF2-Schema	XML Schema, e.g. xhtml	Specific RAG	UML-CD, -SC, OCL	UML, many others	CDIF-languages	UML, many others	HTML XML MOF UML DSL
M1	String, Program	Text in lines	csv Table	Relations	NF2-tree relation	XML-Documents	Link-Syntax-Trees	Classes, Programs	Classes, Programs	CDIF-Models	Classes, Programs	Facts (T-Box)
M0	Objects	Sequences of lines	Sequences of rows	Sets of tuples	trees	dynamic semantics in browser		Object nets	Hierarchical graphs	Object nets	Object nets	A-Box (RDF-Graphs)

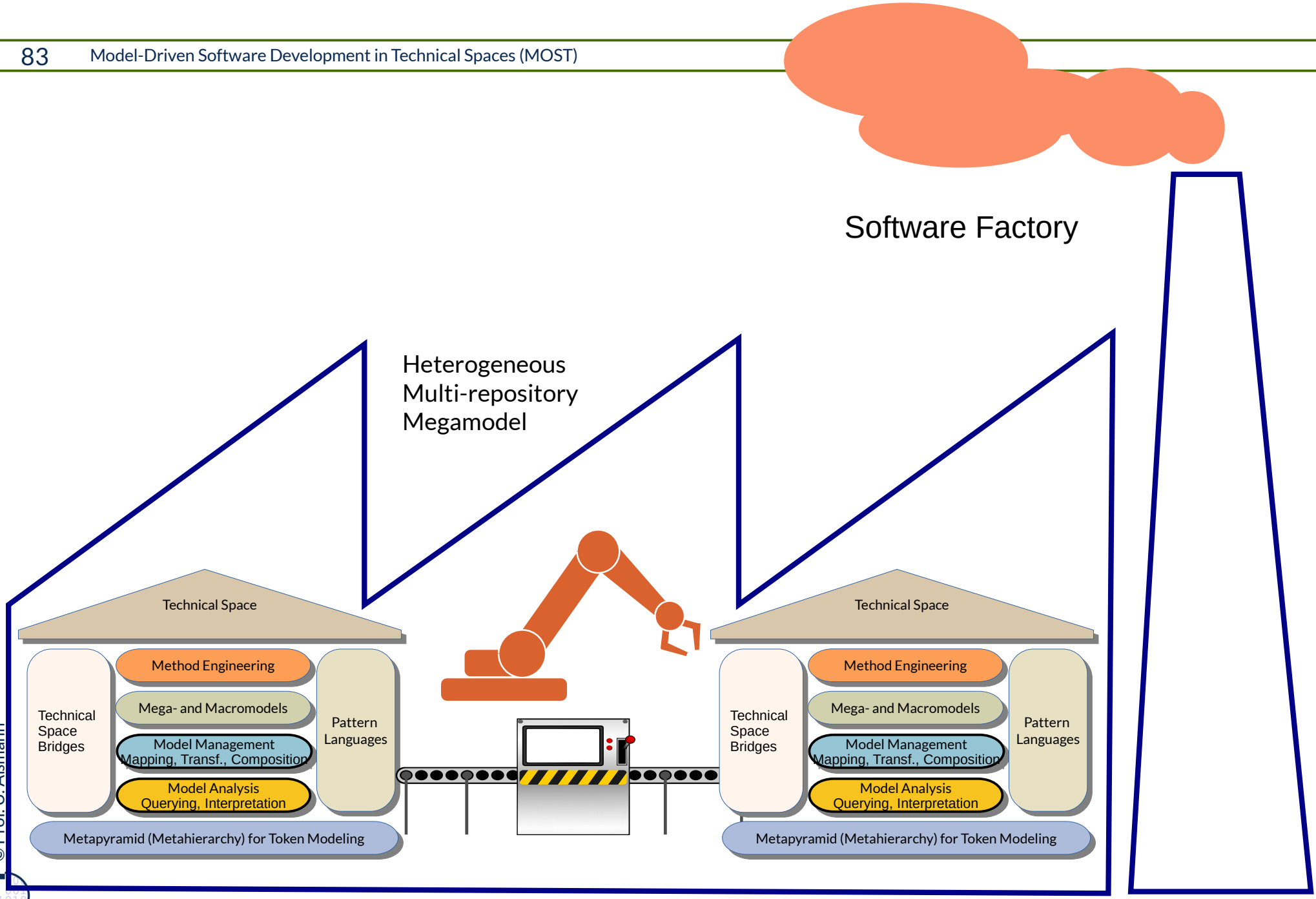
Q12: The ReDeCT Problem and its Macromodel

- The **ReDeCT problem** is the problem how requirements, design, code and tests are related (→ V model)
- Mappings between the Requirements model, Design model, Code, Test cases
- A **ReDeCT macromodel** has maintained mappings between all 4 models



© Prof. U. Aßmann

Q13: A Software Factory's Heart: the Multi-TS Megamodel



© Prof. U. Aßmann

The End

- Why are future CPS a good application area for model-driven software development?
- Explain the model-driven tool chain Preevision, which problems about heterogeneous software systems it solves
- Why are CPS based on collaboration, contexts and roles?
- Why is modeling important for CPS?

© Prof. U. Aßmann