2. Applications for MOST

Prof. Dr. Uwe Aßmann

Technische Universität Dresden

Institut für Software- und Multimediatechnik

Lehrstuhl Softwaretechnologie

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

WS 15/16-1.3, 19.10.15

1) Cyber-physical systems (CPS)
2) Design of CPS with Domain-Specific CPS tool chain
3) Experience with Cloud Robots
4) A Killer App for CPS

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

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

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

Smart Parking

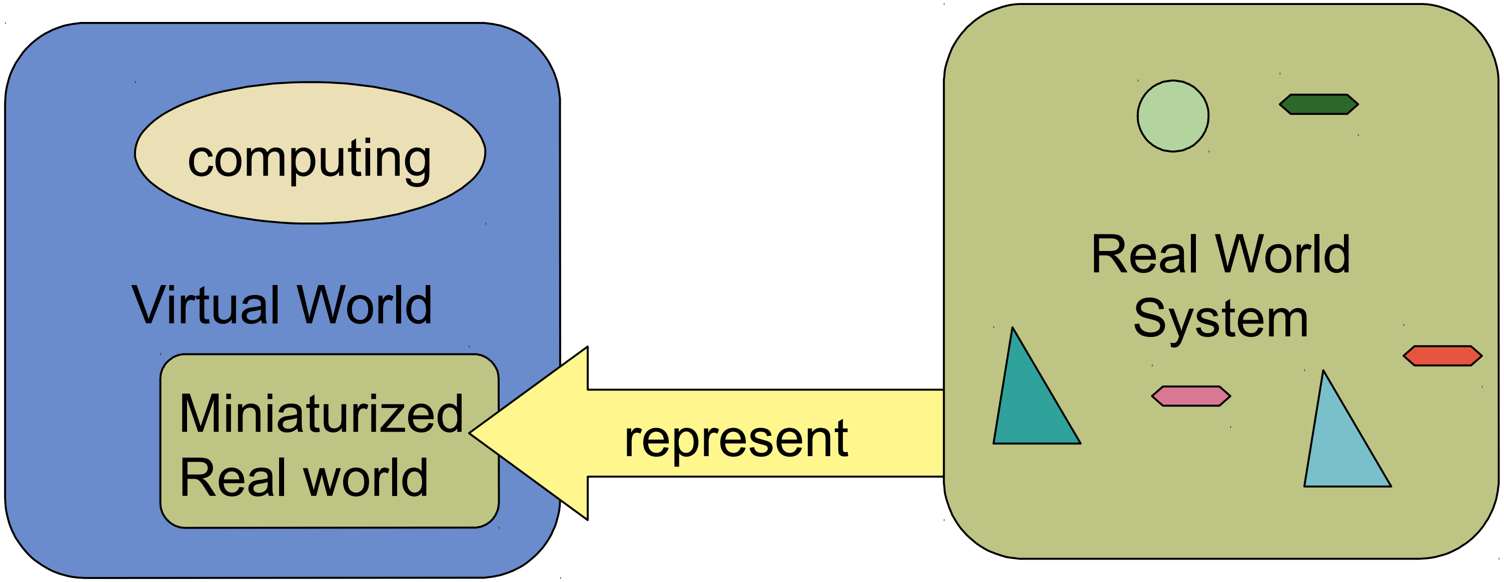
http://commons.wikimedia.org/wiki/File:Bundesarchiv_Bild_183-H0605-0007-001,_Rostock,_Ernst-Th%C3%A4lmann-Platz,_Parkplatz,_Marienkirche.jpg#mediaviewer/File:Bundesarchiv_Bild_183-H0605-0007-001,_Rostock,_Ernst-Th%C3%A4lmann-Platz,_Parkplatz,_Marienkirche.jpg

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

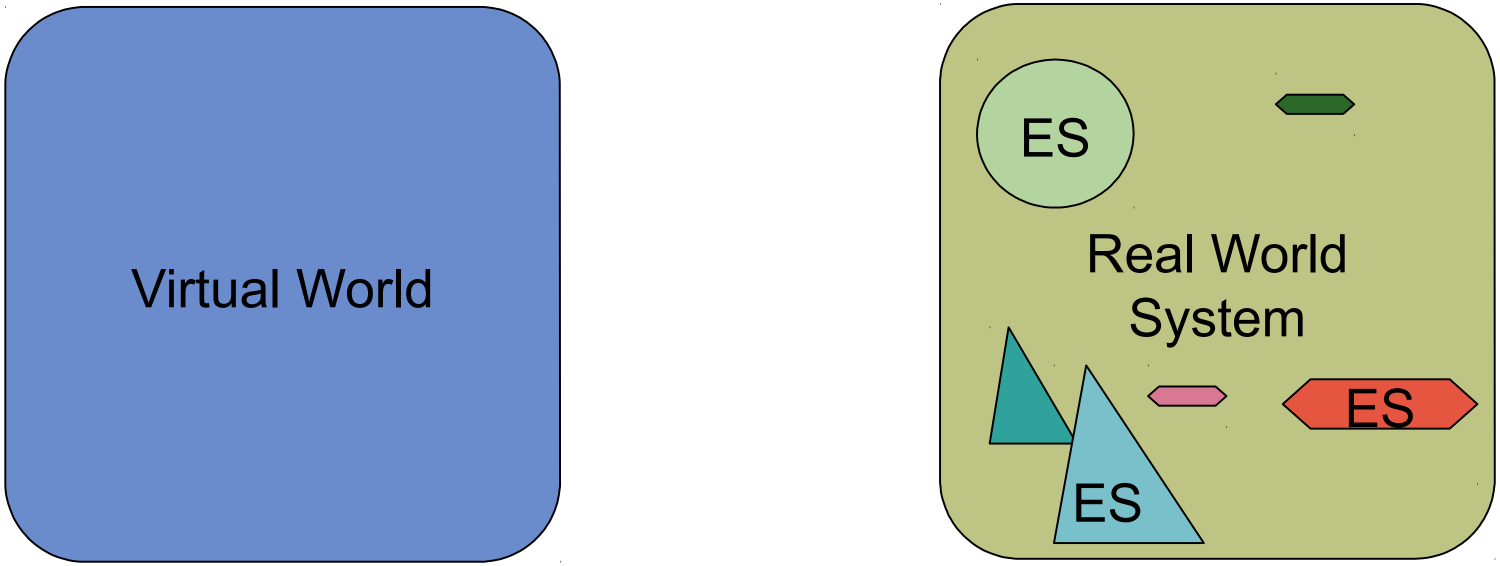
„Standard“ Computing

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



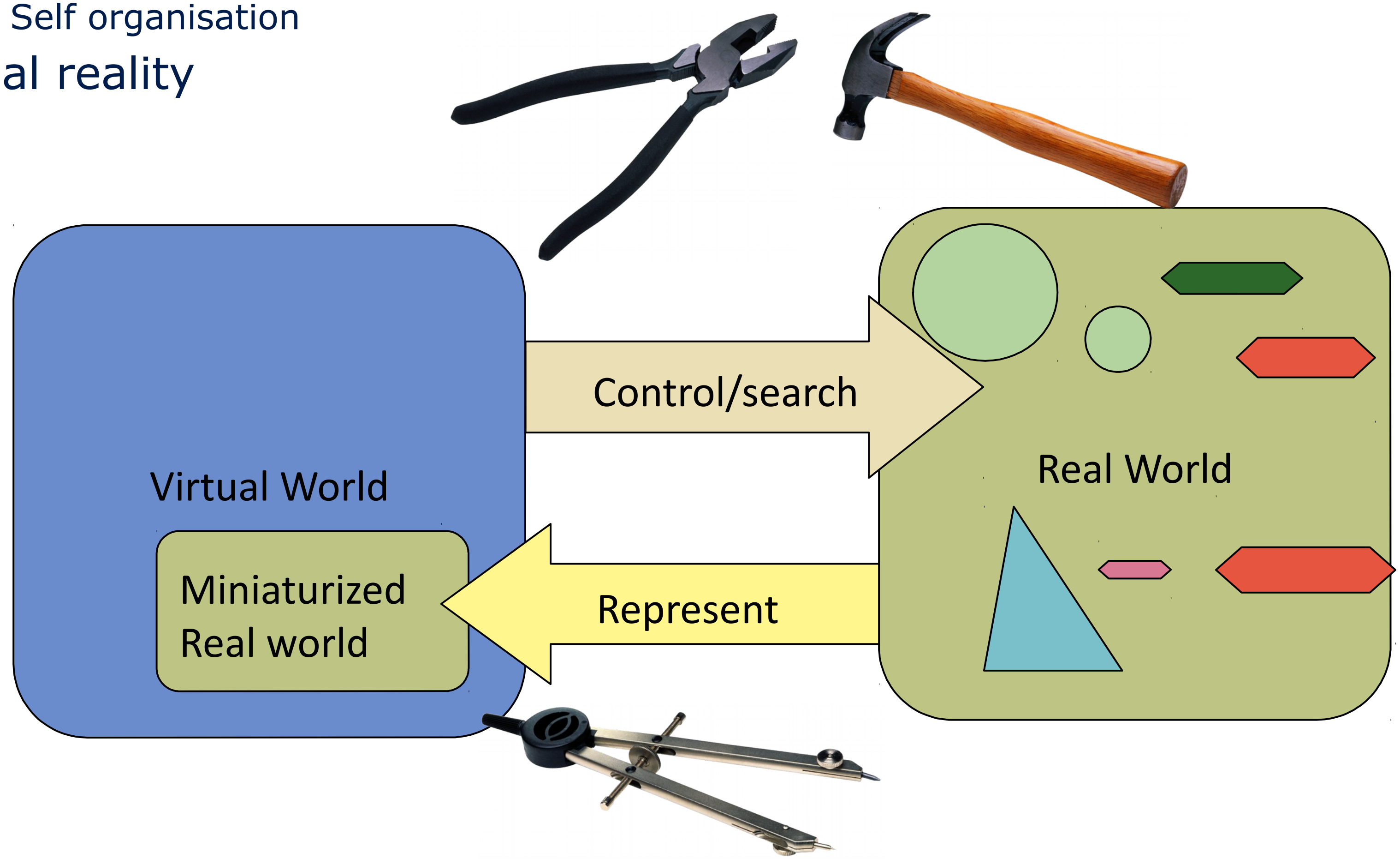
Embedded System

- The computer is integrated into the real-life object



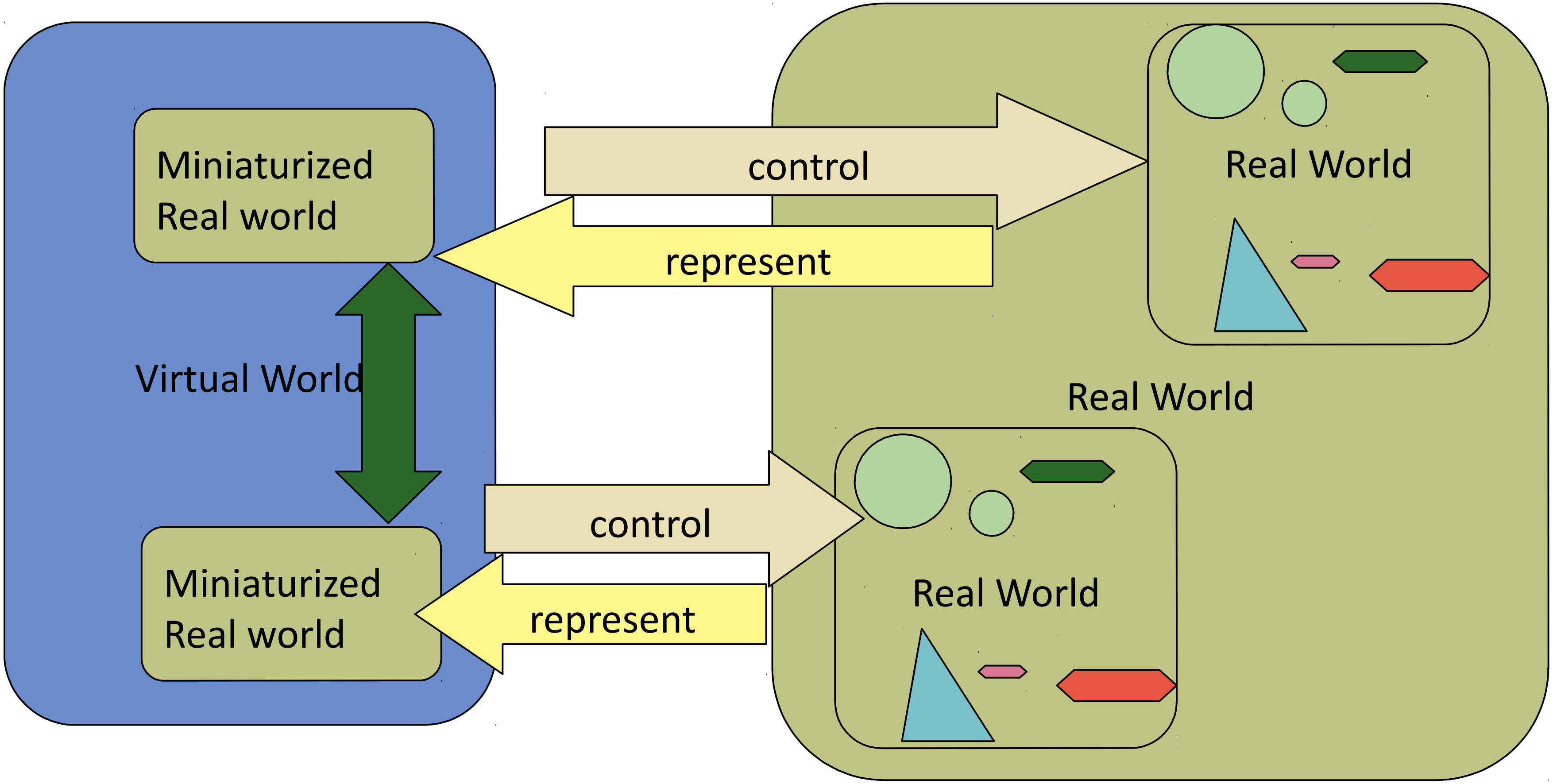
Cyber-Physical System (CPS)

- 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



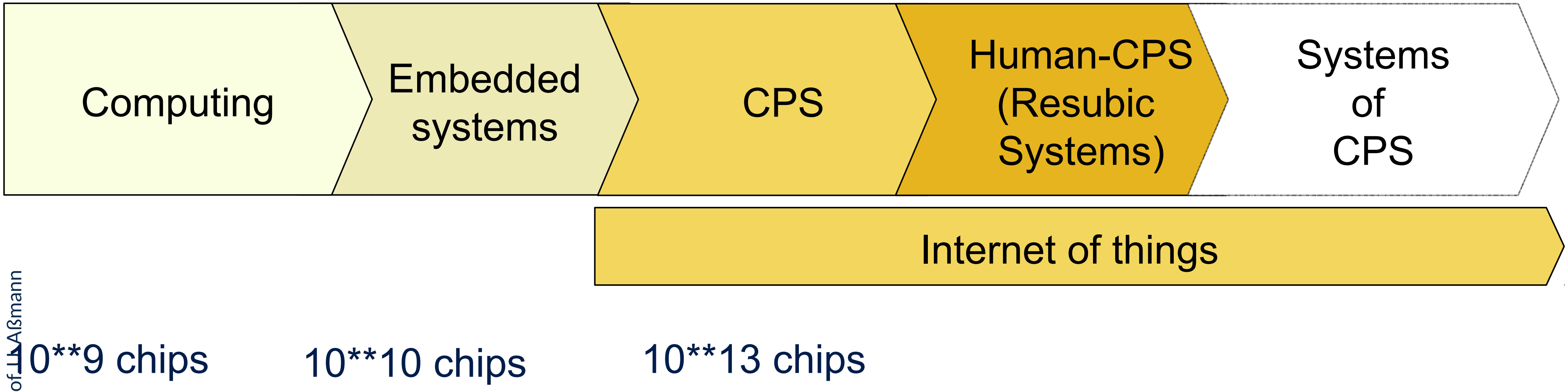
The Internet of Things

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

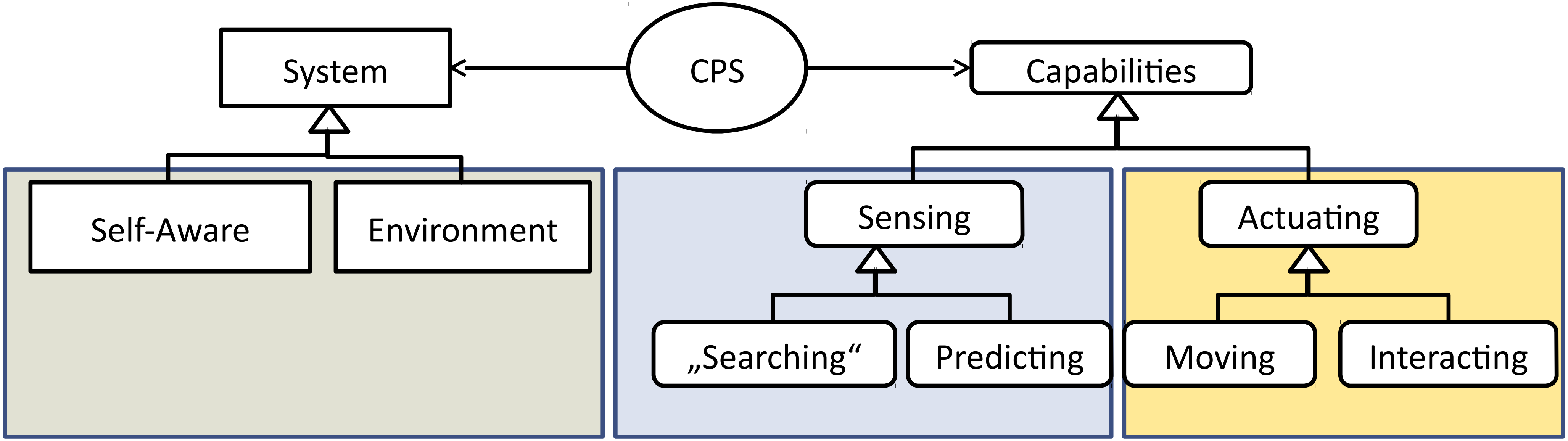


Trend CPS

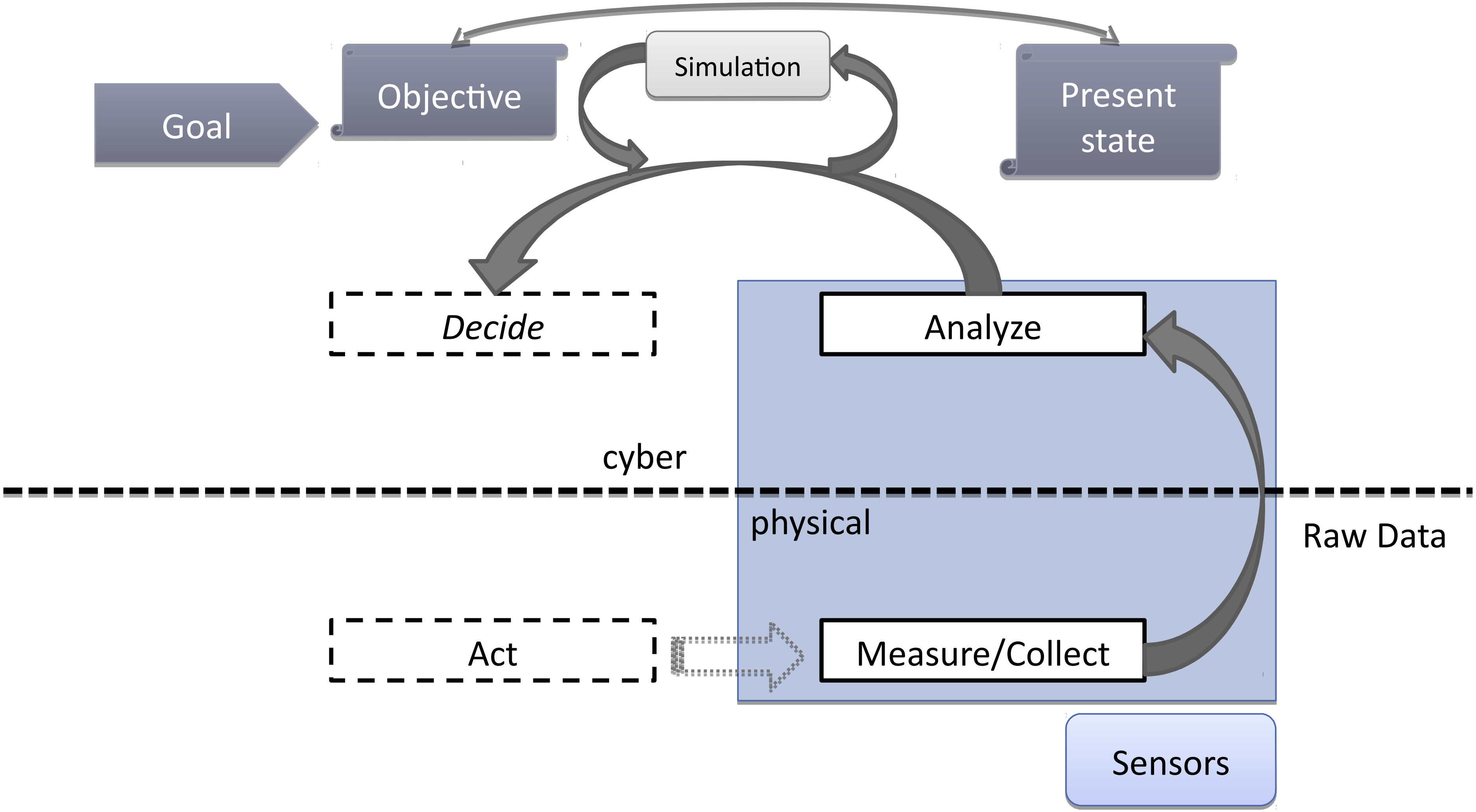
- Cyber-physical systems are the first step in the internet of things



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



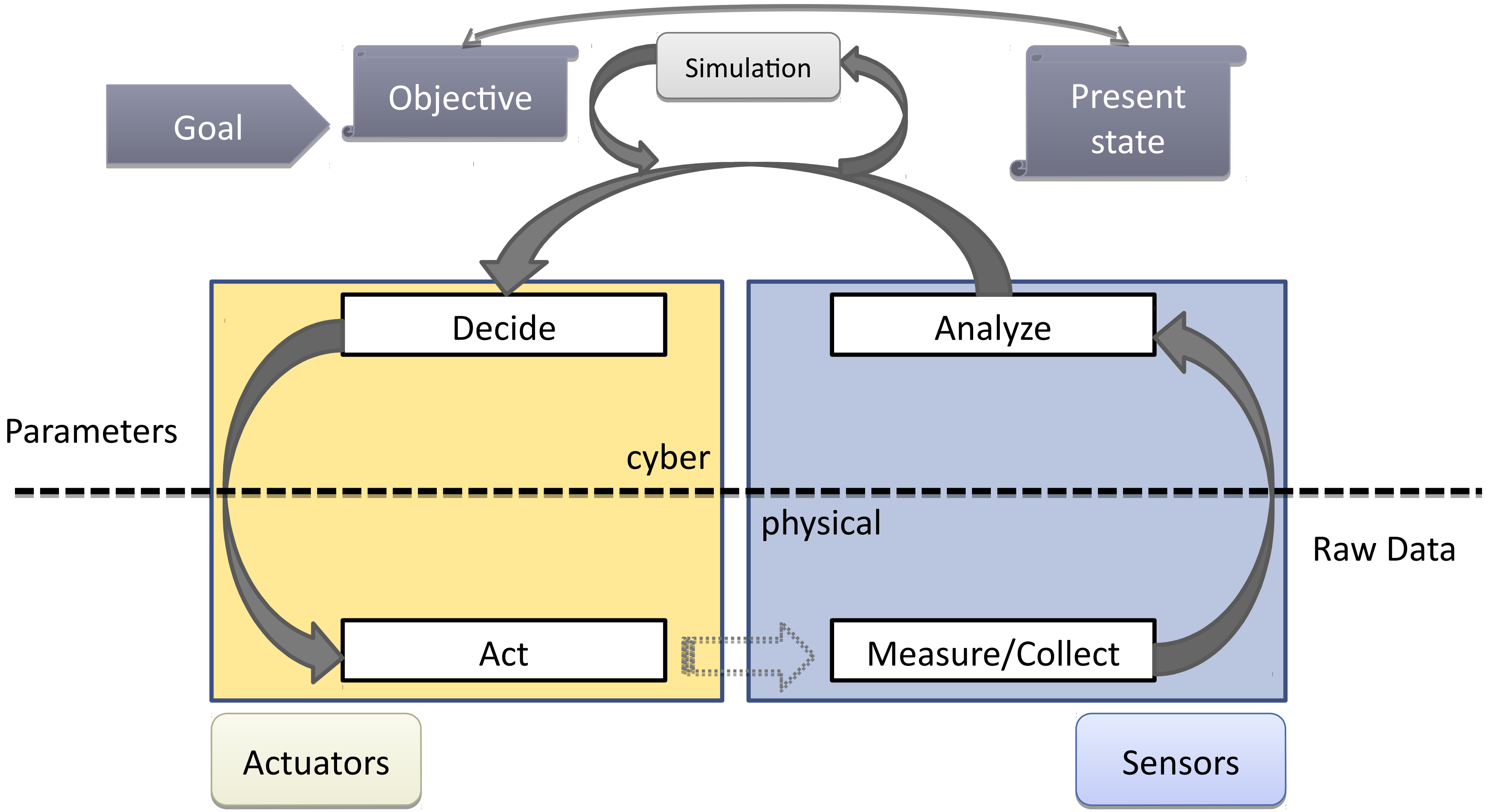
Cyber-Physical Database Systems = Analysis, Simulation and Prediction



A Cyber-Physical System

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

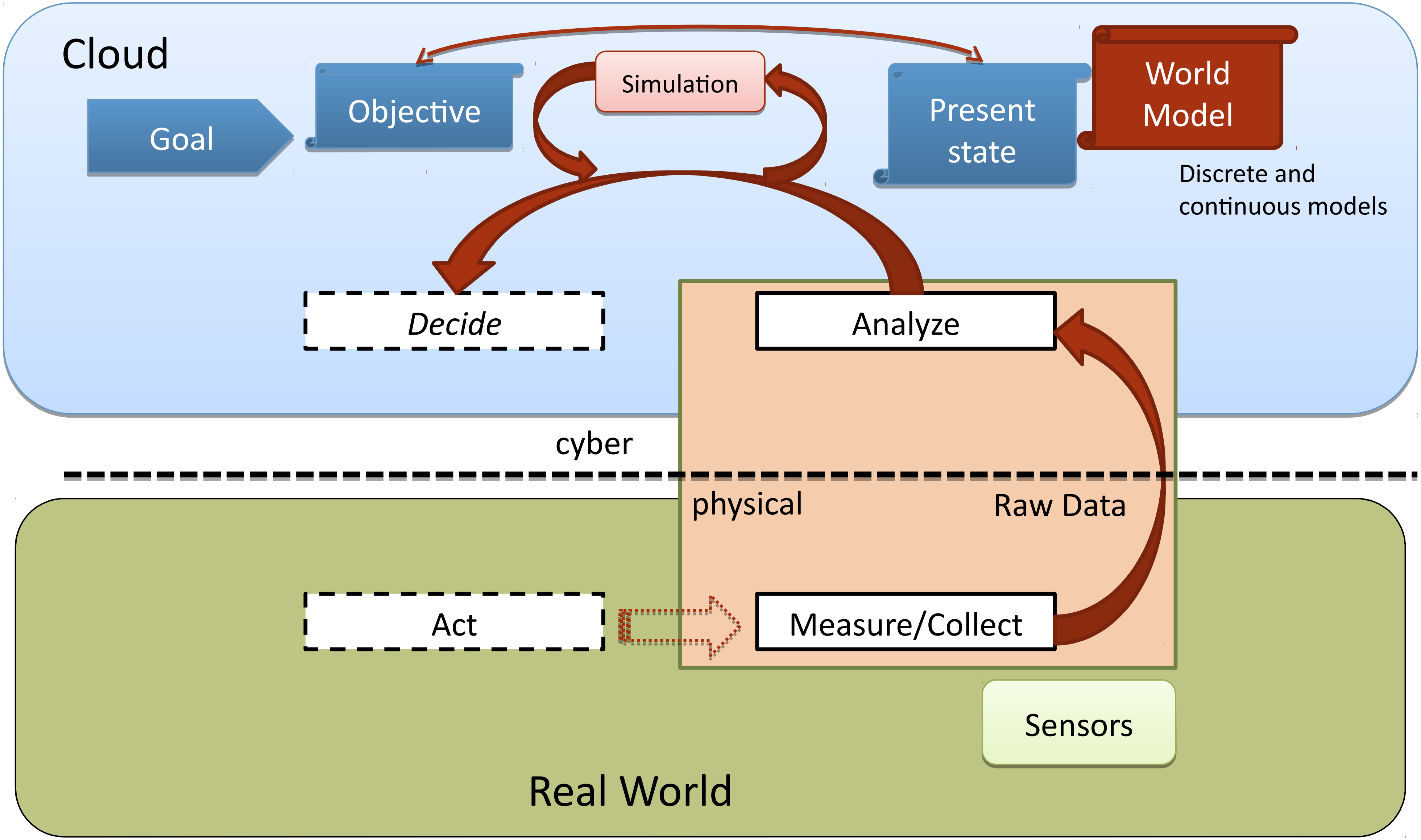
Cloud Robots =
Cyber-Physical Management Systems



2.1.2. Two Basic Forms of CPS

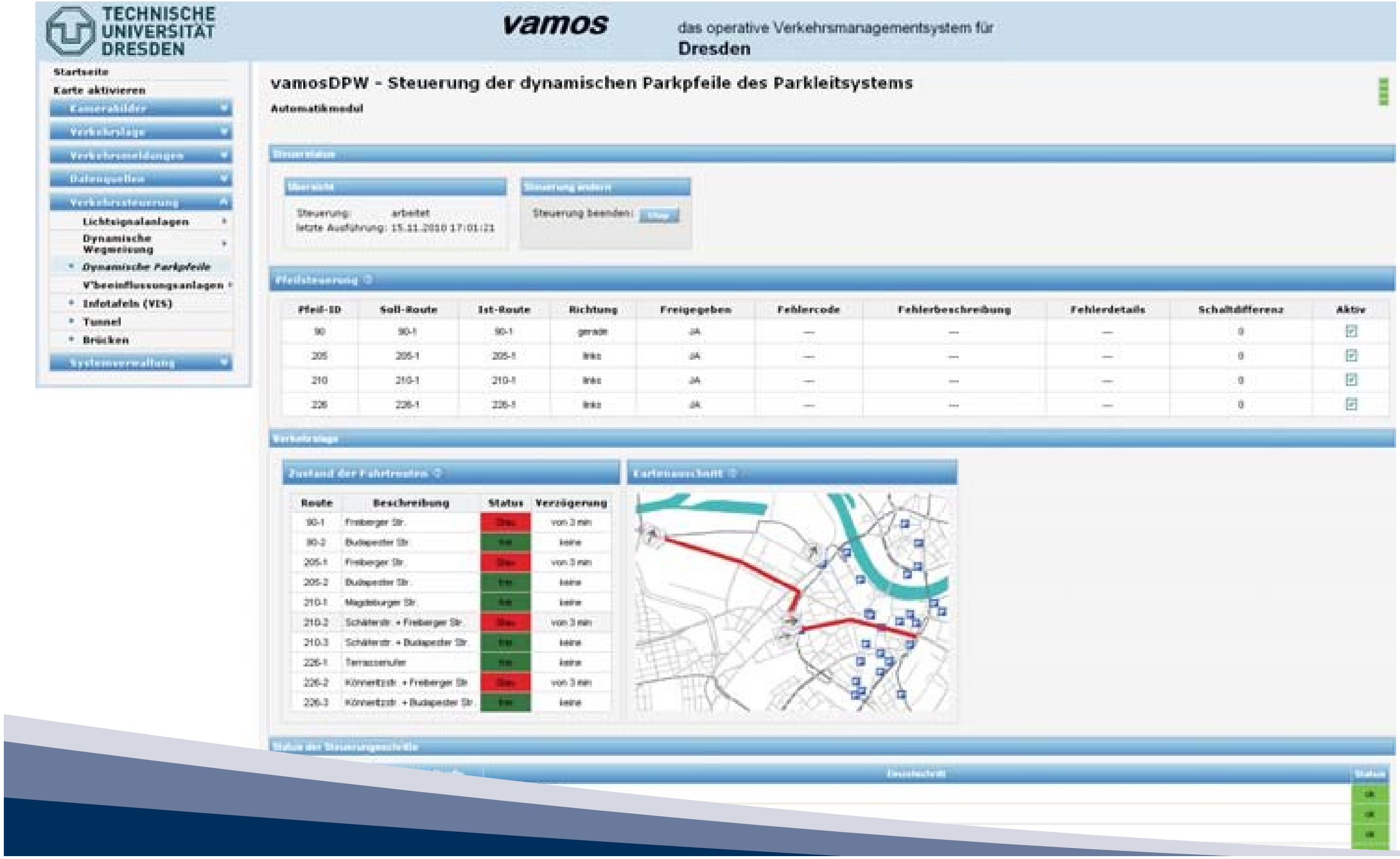
- World Databases
- Cloud Robots

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



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



TECHNISCHE UNIVERSITÄT DRESDEN

vamos

das operative Verkehrsmanagementsystem für Dresden

vamosDPW - Steuerung der dynamischen Parkpfeile des Parkleitsystems

Automatikmodul

Steuerstatus

Übersicht

Steuerung: arbeitet
letzte Ausführung: 15.11.2010 17:01:21

Steuerung ändern

Steuerung beenden: Stop

Pfeilsteuerung

Pfeil-ID	Soll-Route	Ist-Route	Richtung	Freigegeben	Fehlercode	Fehlerbeschreibung	Fehlerdetails	Schaltdifferenz	Aktiv
90	90-1	90-1	gerade	JA	---	---	---	0	☑
205	205-1	205-1	links	JA	---	---	---	0	☑
210	210-1	210-1	links	JA	---	---	---	0	☑
226	226-1	226-1	links	JA	---	---	---	0	☑

Verkehrslage

Zustand der Fahrtrouten

Route	Beschreibung	Status	Verzögerung
90-1	Freiberger Str.	Stau	von 3 min
90-2	Budapester Str.	frei	keine
205-1	Freiberger Str.	Stau	von 3 min
205-2	Budapester Str.	frei	keine
210-1	Magdeburger Str.	frei	keine
210-2	Schäferstr. + Freiberger Str.	Stau	von 3 min
210-3	Schäferstr. + Budapester Str.	frei	keine
226-1	Terrassenufer	frei	keine
226-2	Könneritzstr. + Freiberger Str.	Stau	von 3 min
226-3	Könneritzstr. + Budapester Str.	frei	keine

Kartenausschnitt

Status der Steuerungsschritte

2.1.2 Important World Models of World Databases

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

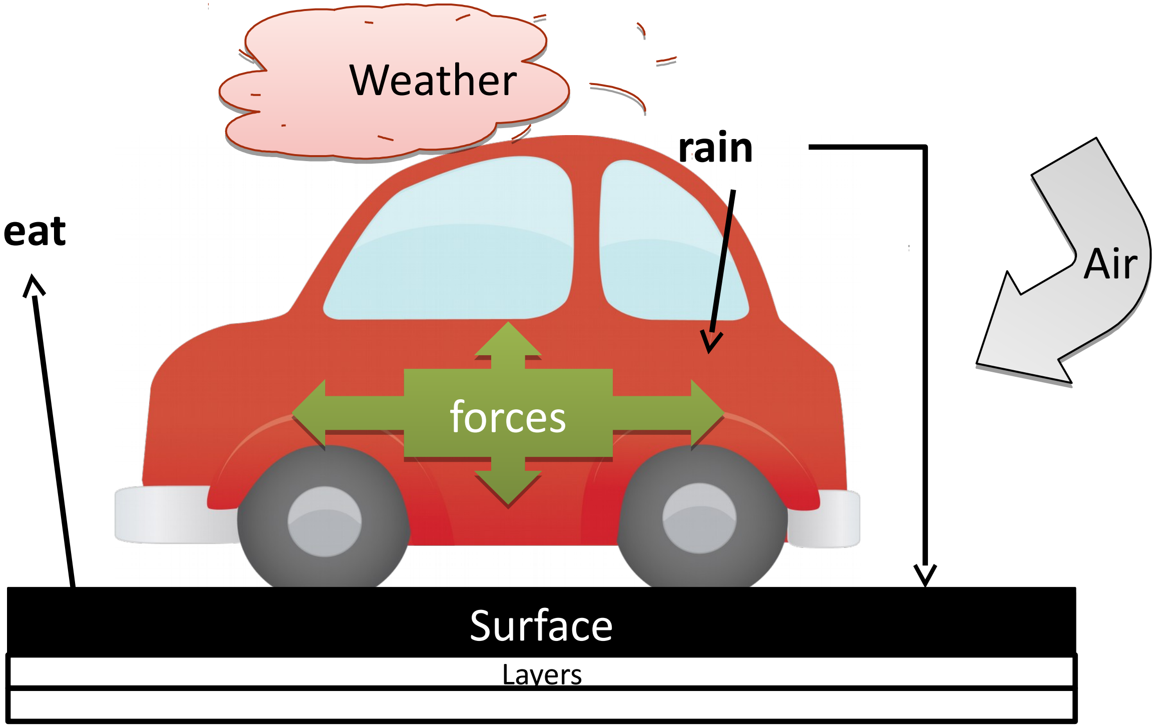
3D office models
Building models
City models
http://www.turbosquid.com

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

Physical Dynamics (Movement) of Thing

- How does it move in space?
  - Continuous modeling languages (Modelica)

CPS
Need
Dynamics
Models



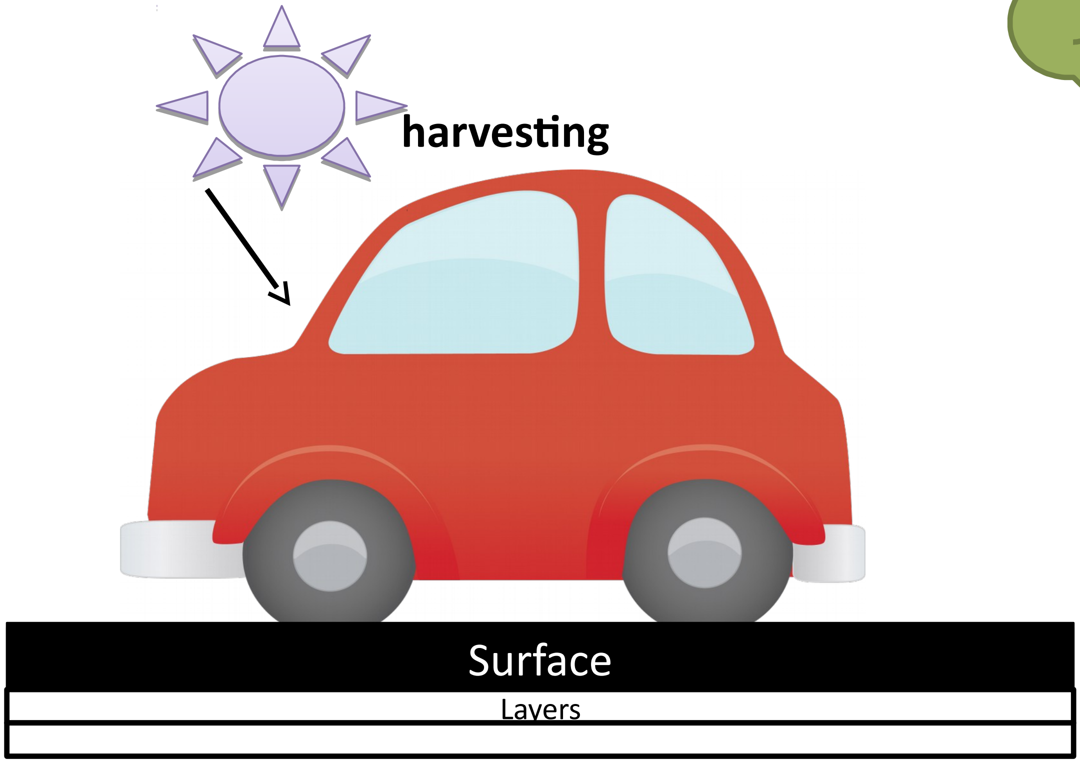
complex interplay of

- surface props
- weather: wind, rain, heat

Energy Consumption of Thing

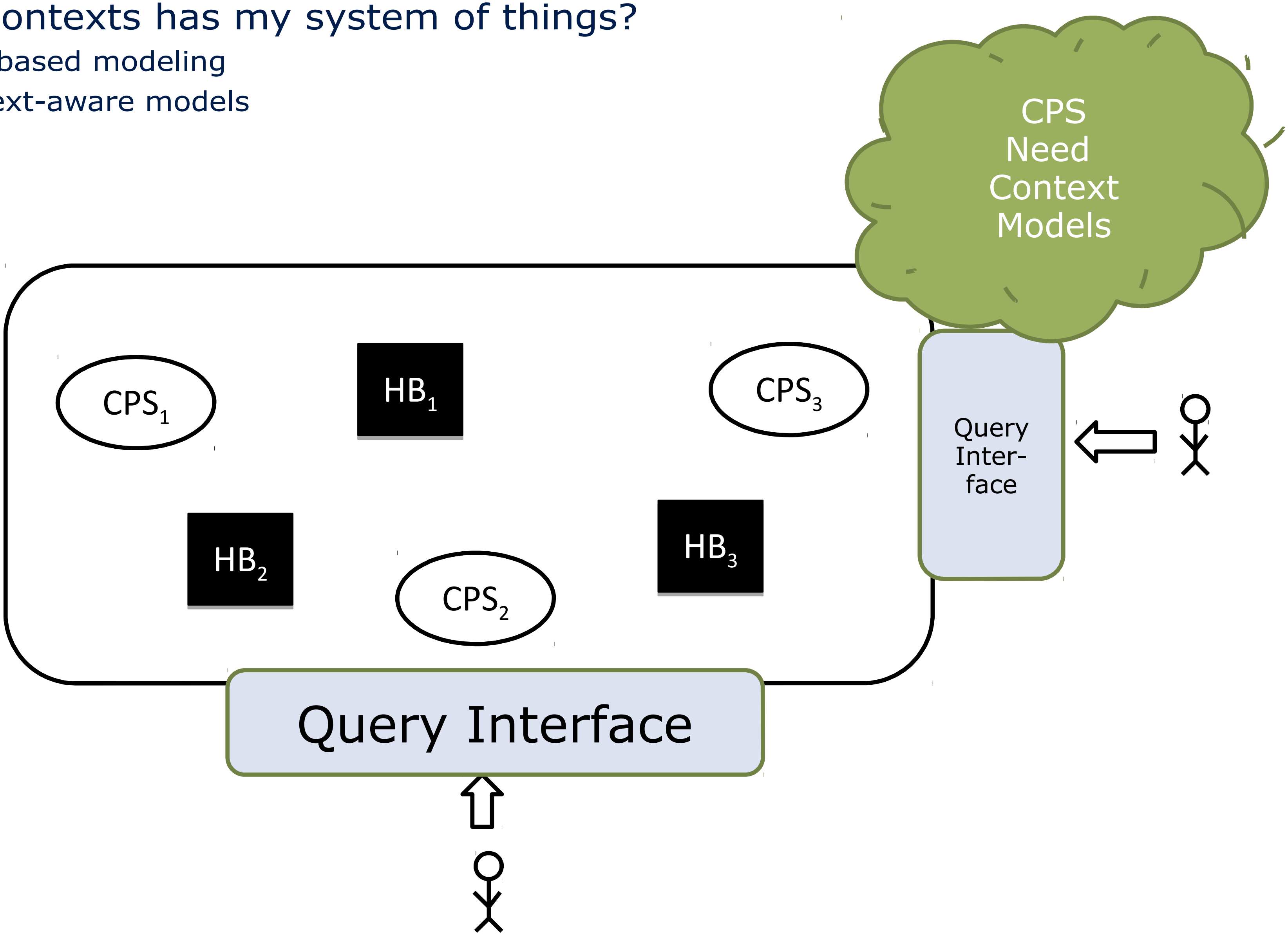
- How much energy is left for its tasks?

CPS
Need
Energy Models



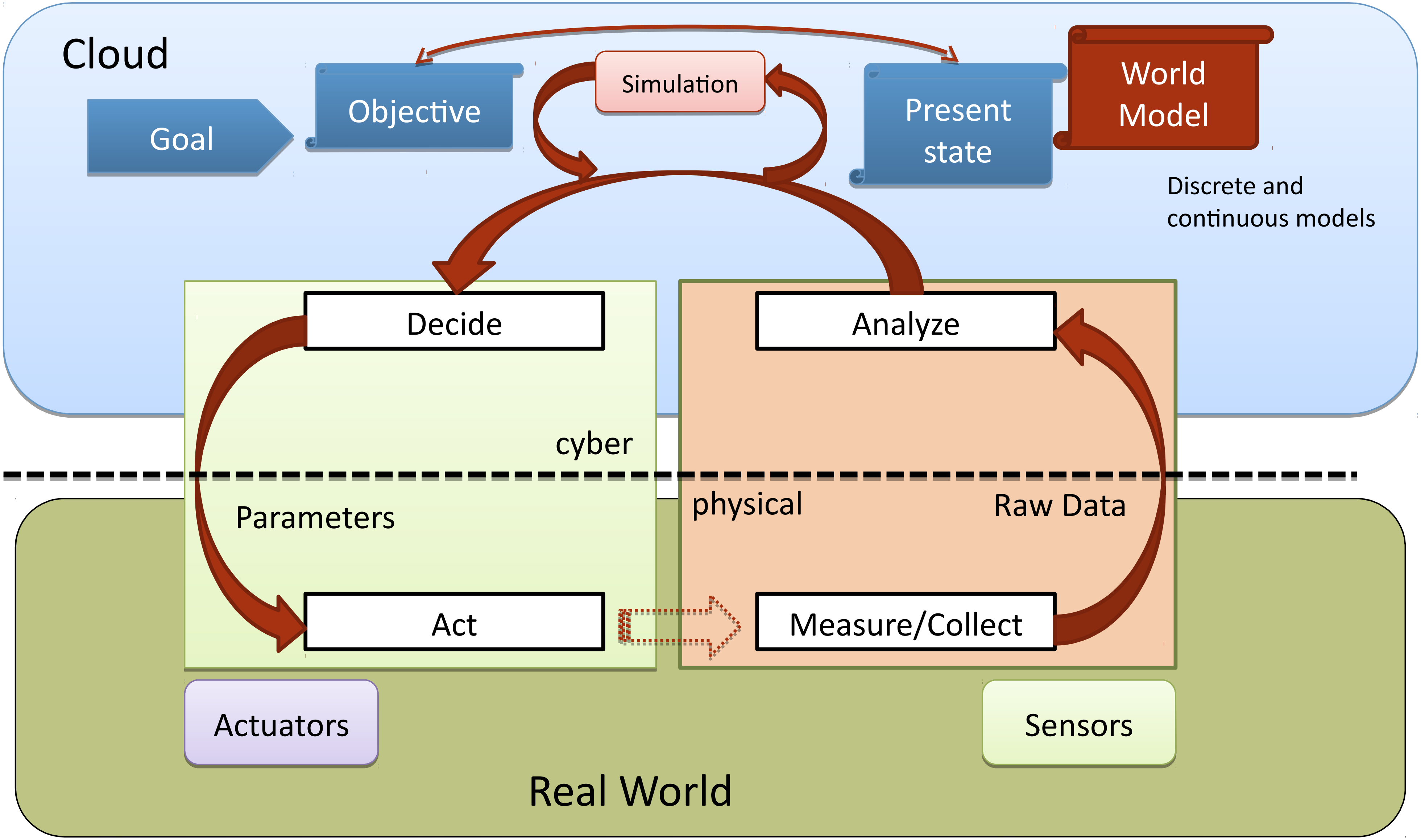
Current Physical Composition of a Thing

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



2.1.3. What is a Cloud Robot?

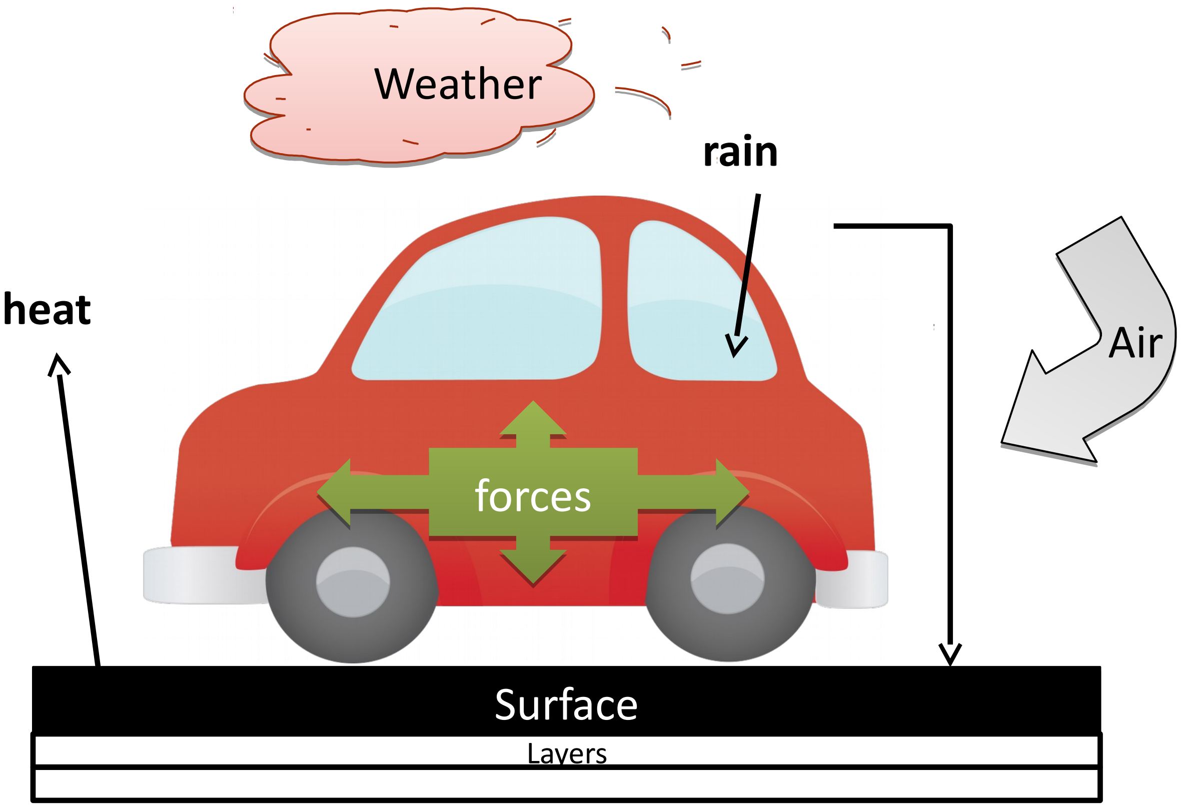
Cloud Robots are Controlling CPS



Physical Dynamics (Movement) of Cloud Robot

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

Cloud Robots Need World Models

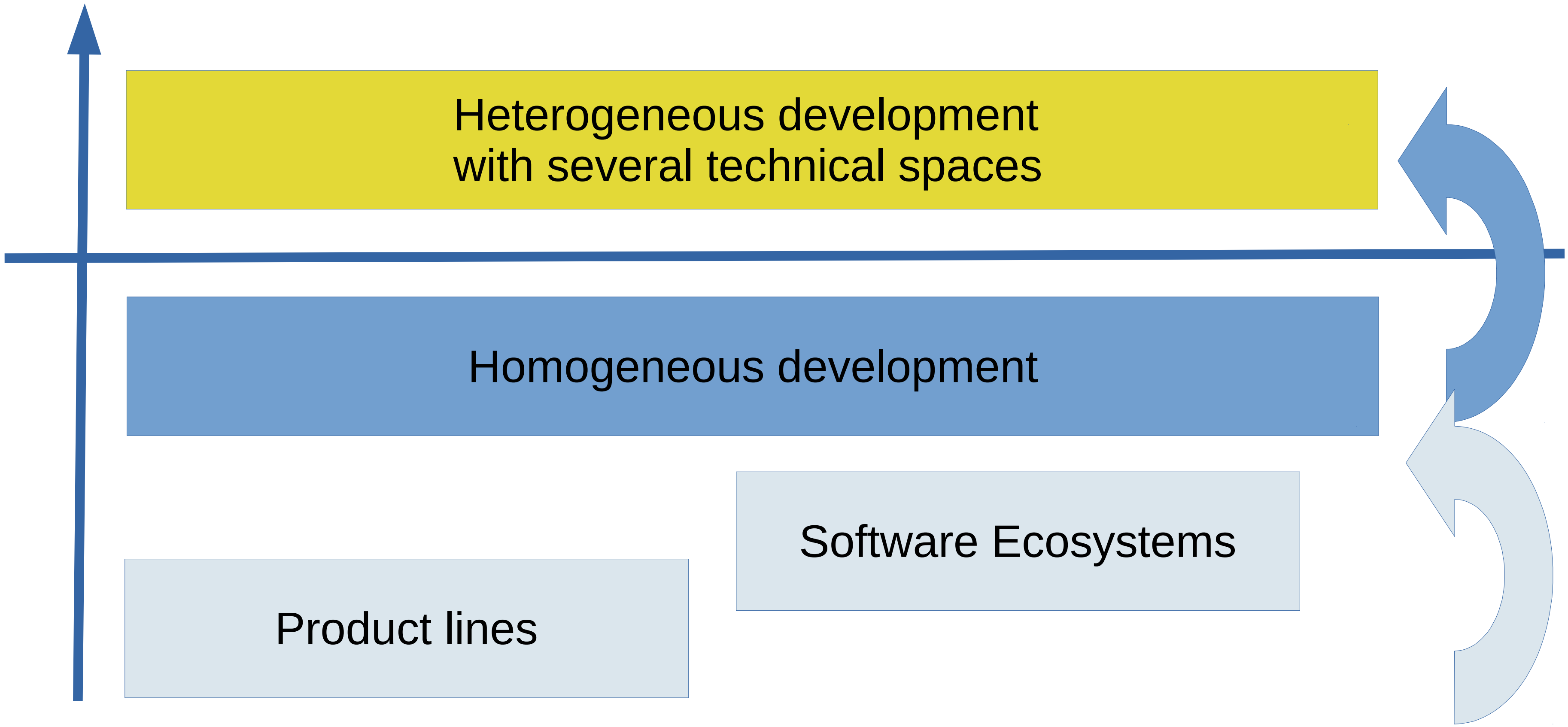


2.2 How will We Design Such CPS?

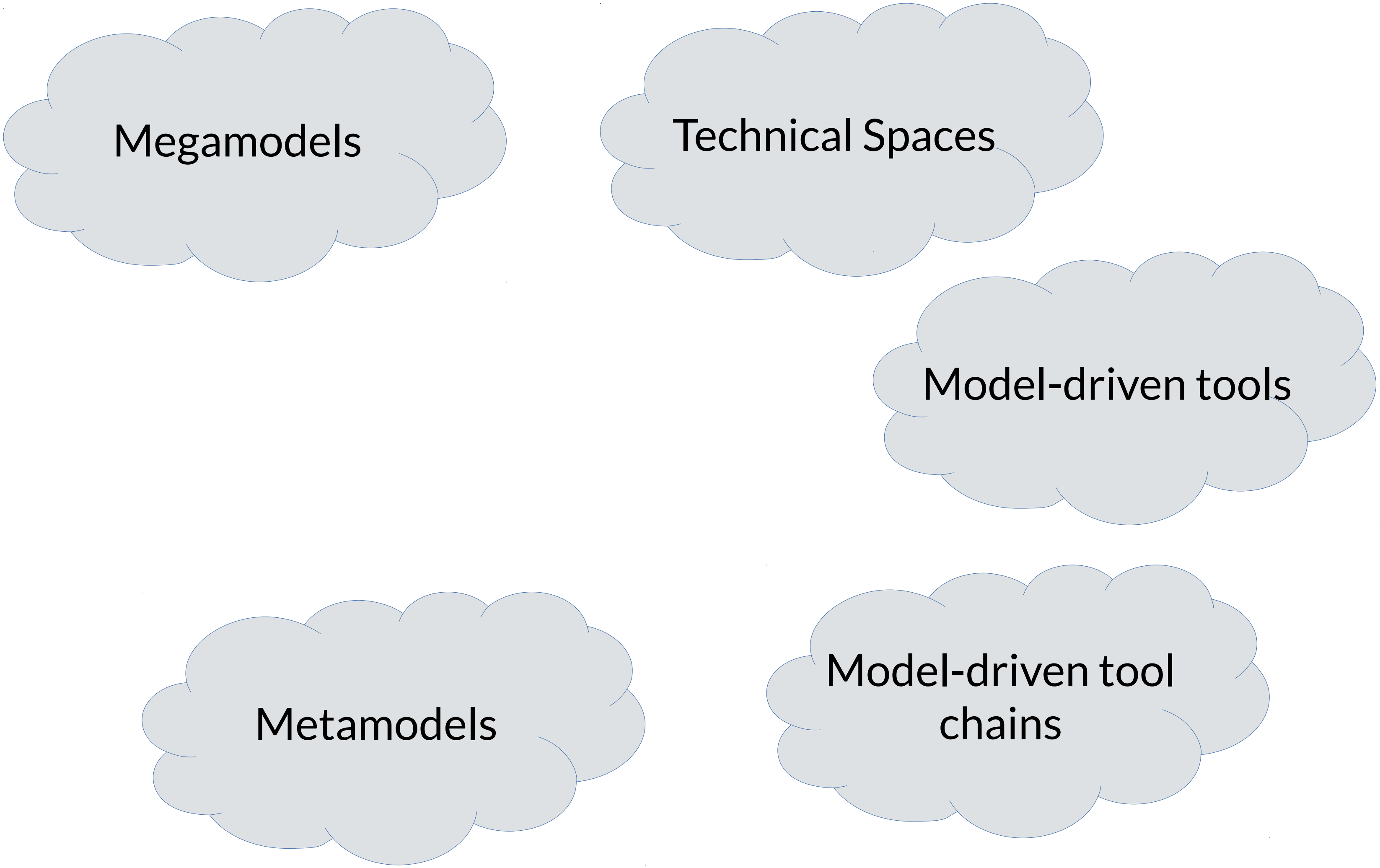
2.2.1 Domain-Specific MDSD IDE for Design of Cyber-Physical Systems

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



Concepts of the Course



Megamodels

Technical Spaces

Model-driven tools

Metamodels

Model-driven tool chains

Concepts of the Course

Graph transformation

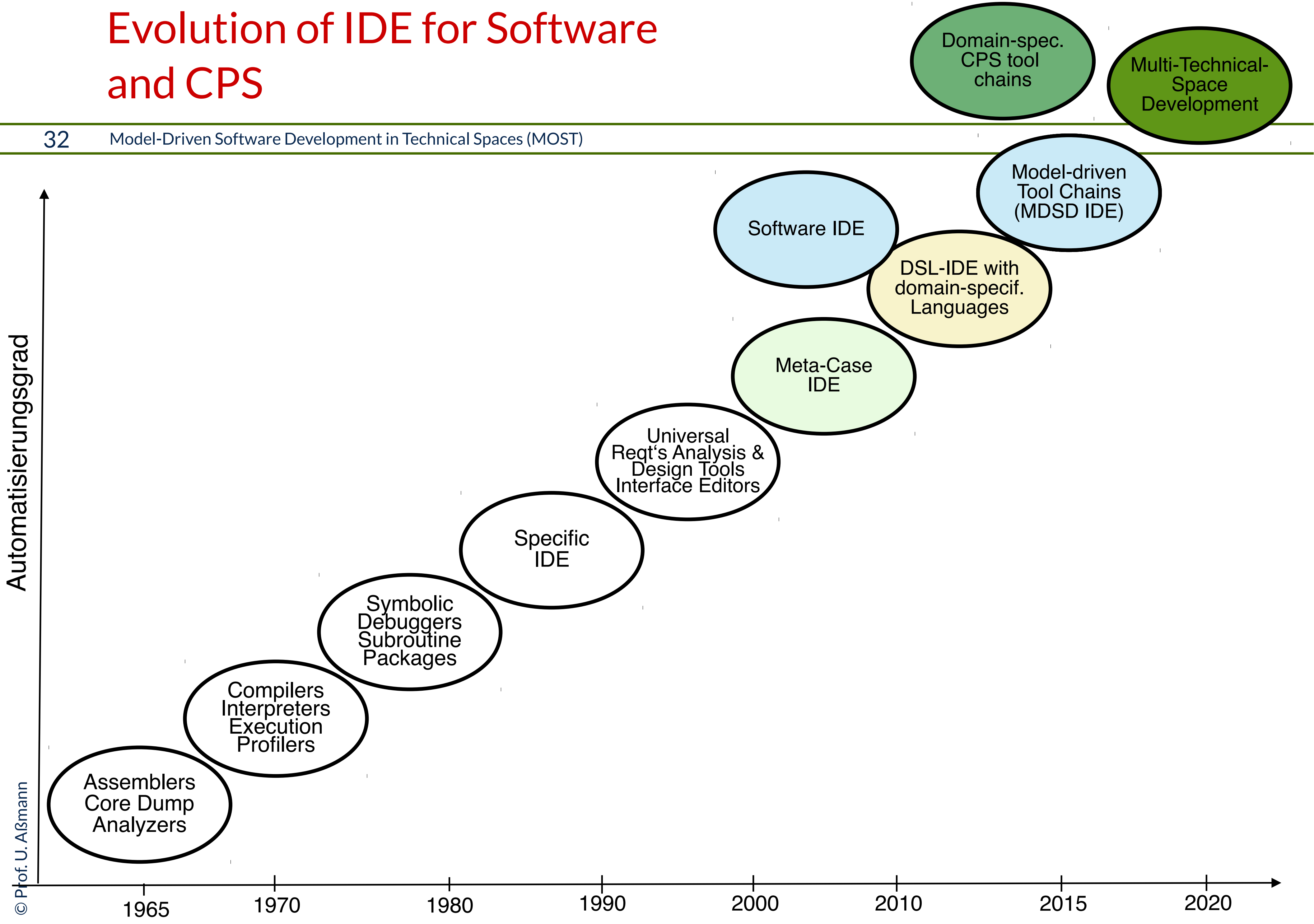
Reducible graph analysis

Graph reachability

Triple Graph Grammars

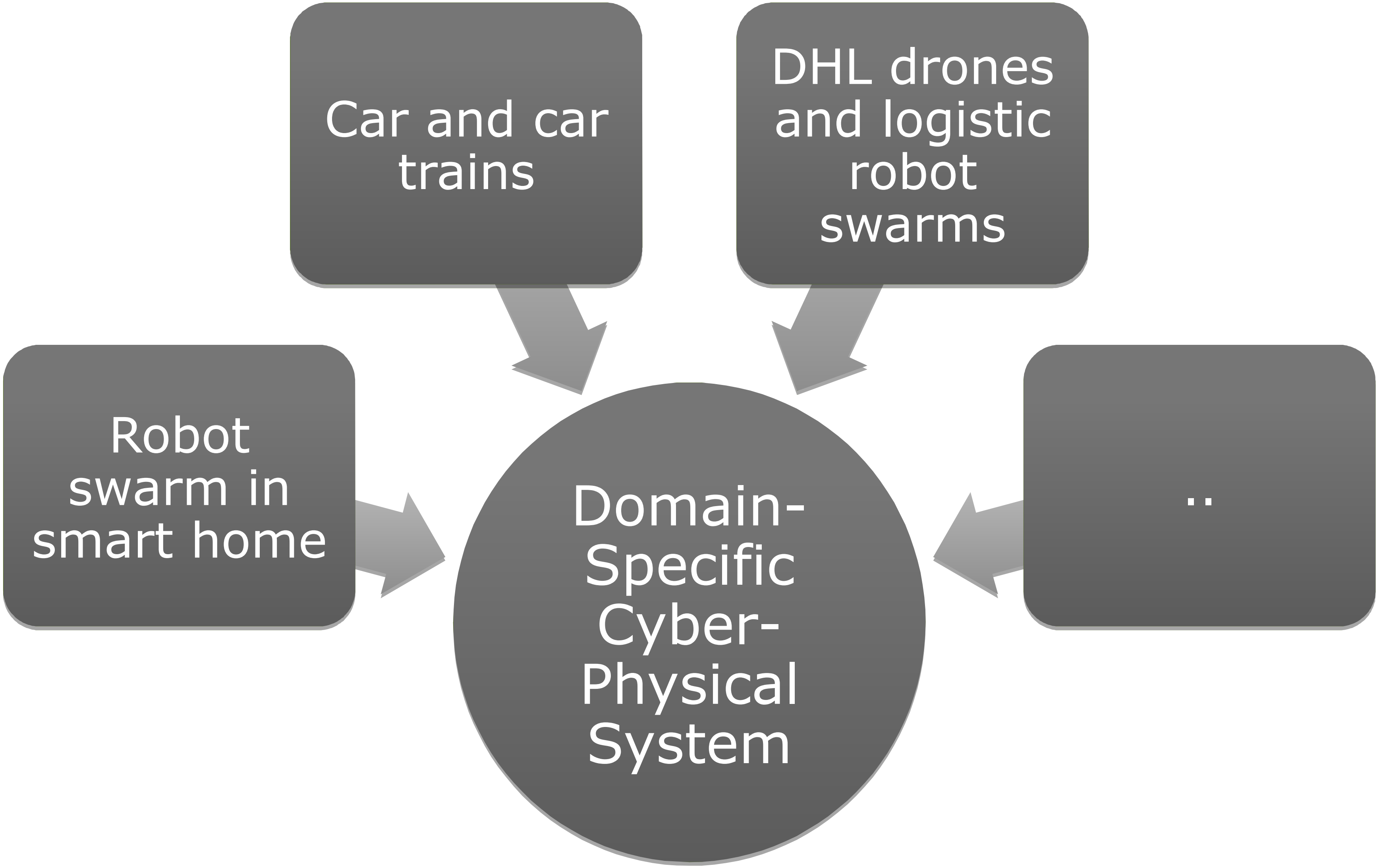
Graph querying

Evolution of IDE for Software and CPS

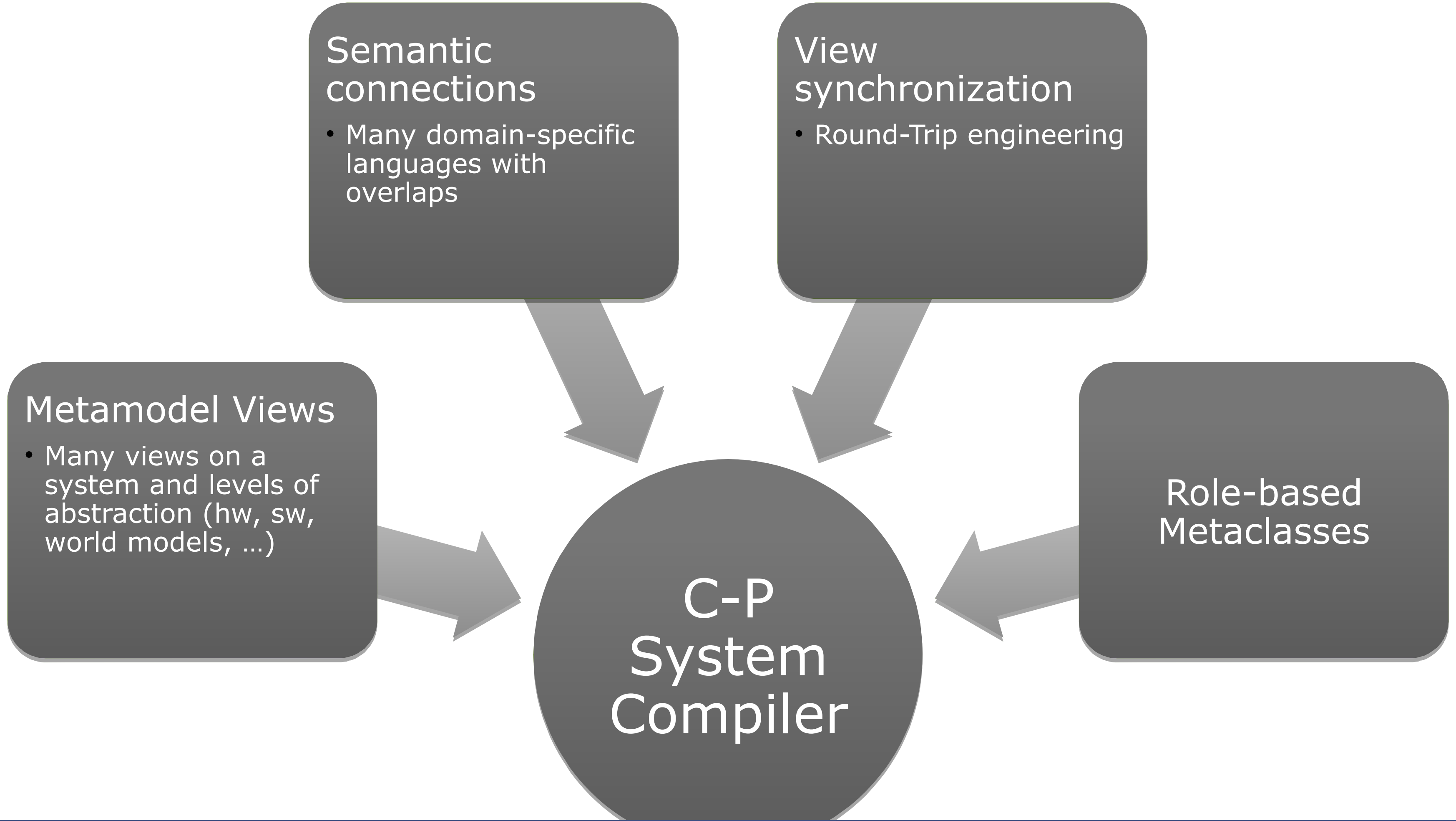

Domain-spec. CPS tool chains
Multi-Technical-Space Development
Model-driven Tool Chains (MDSD IDE)
Software IDE
DSL-IDE with domain-specif. Languages
Meta-Case IDE
Universal Reqt's Analysis & Design Tools Interface Editors
Specific IDE
Symbolic Debuggers Subroutine Packages
Compilers Interpreters Execution Profilers
Assemblers Core Dump Analyzers
Automatisierungsgrad
1965
1970
1980
1990
2000
2010
2015
2020

Quelle: nach [Fisher91, S.20]

Domain-Specific CPS


Car and car trains
DHL drones and logistic robot swarms
Robot swarm in smart home
Domain-Specific Cyber-Physical System
..

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



Semantic connections

- Many domain-specific languages with overlaps

View synchronization

- Round-Trip engineering

Metamodel Views

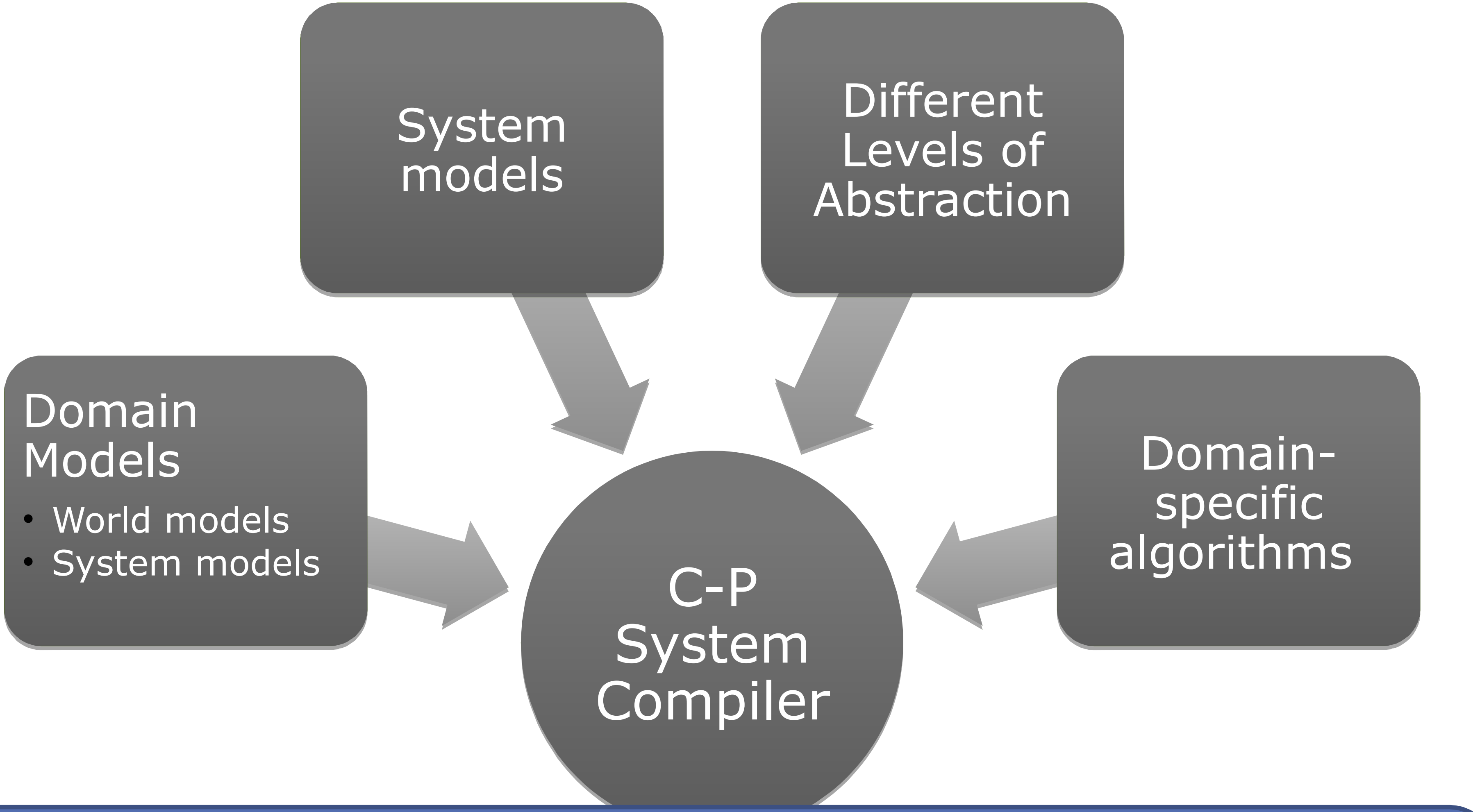
- Many views on a system and levels of abstraction (hw, sw, world models, ...)

Role-based Metaclasses

C-P System Compiler

Cyber-Physical System Compilers should be based on Metamodels

CPS Compilers are Domain-Specific

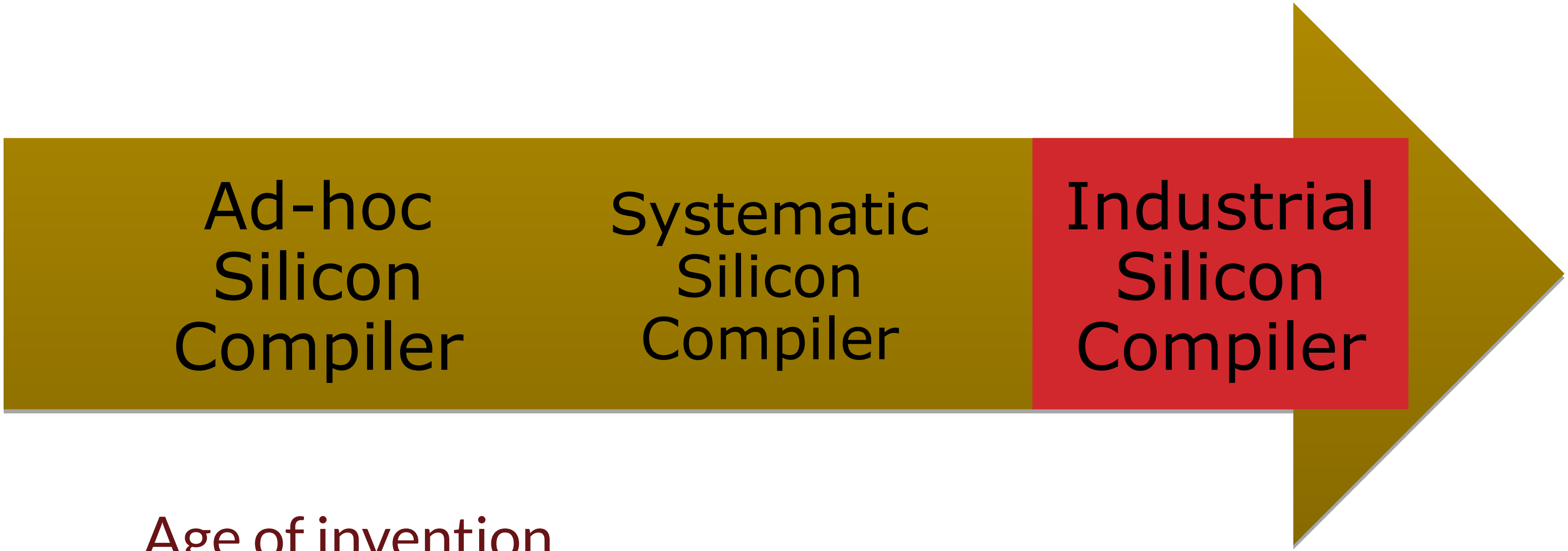


Cyber-Physical System Compilers are domain-specific

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."

How the Silicon Compiler Industry Matured over Time



Age of invention

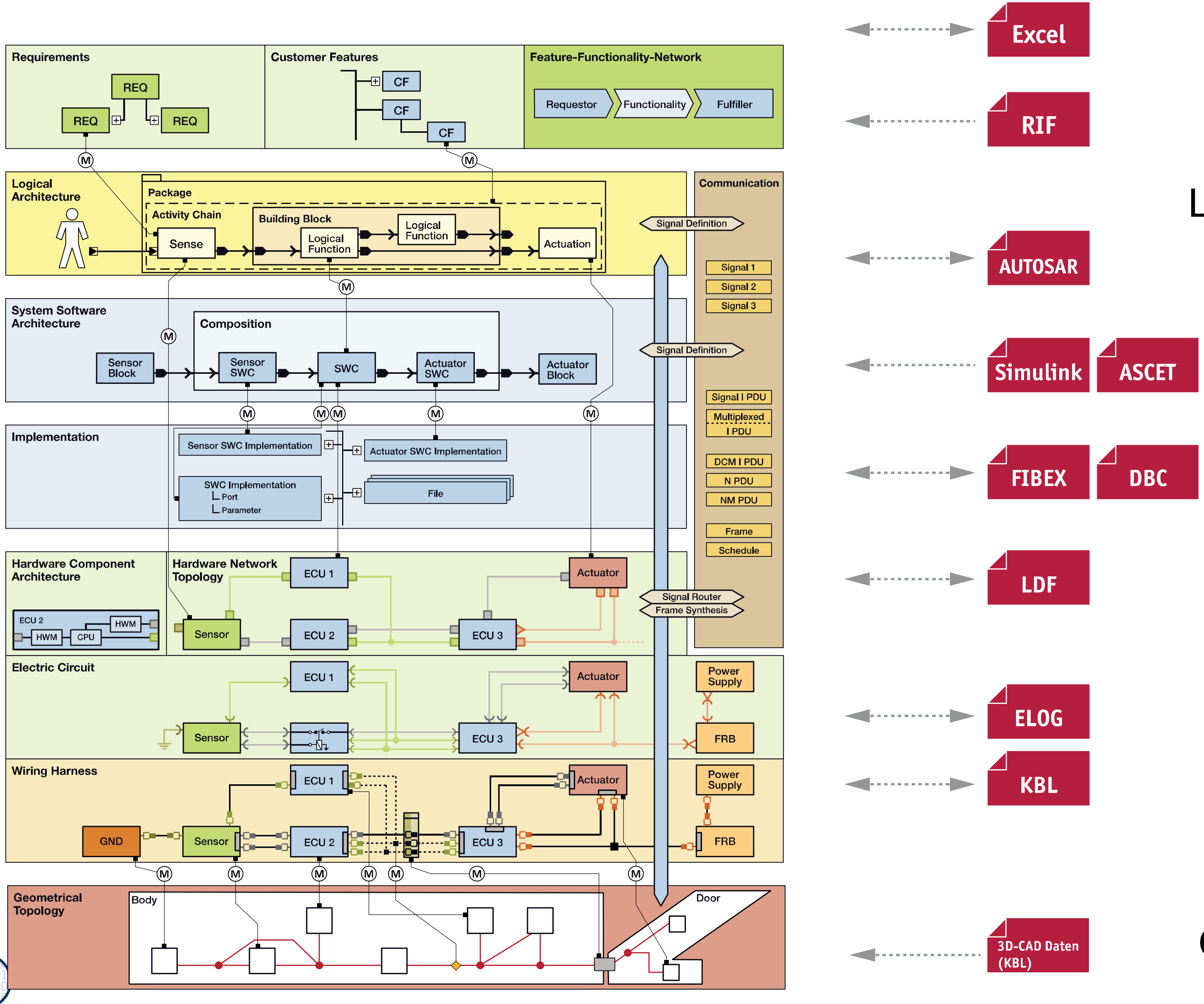
Age of implementation

Age of integration

[Sangiovanni-Vincentelli Tides]

Example 2: Car Design with PREEVision (Vector)

[Preevision]



Requirements

Logical Architecture

System Software Architecture

Implementation

Hardware Architecture

Electric circuit

Wiring harness

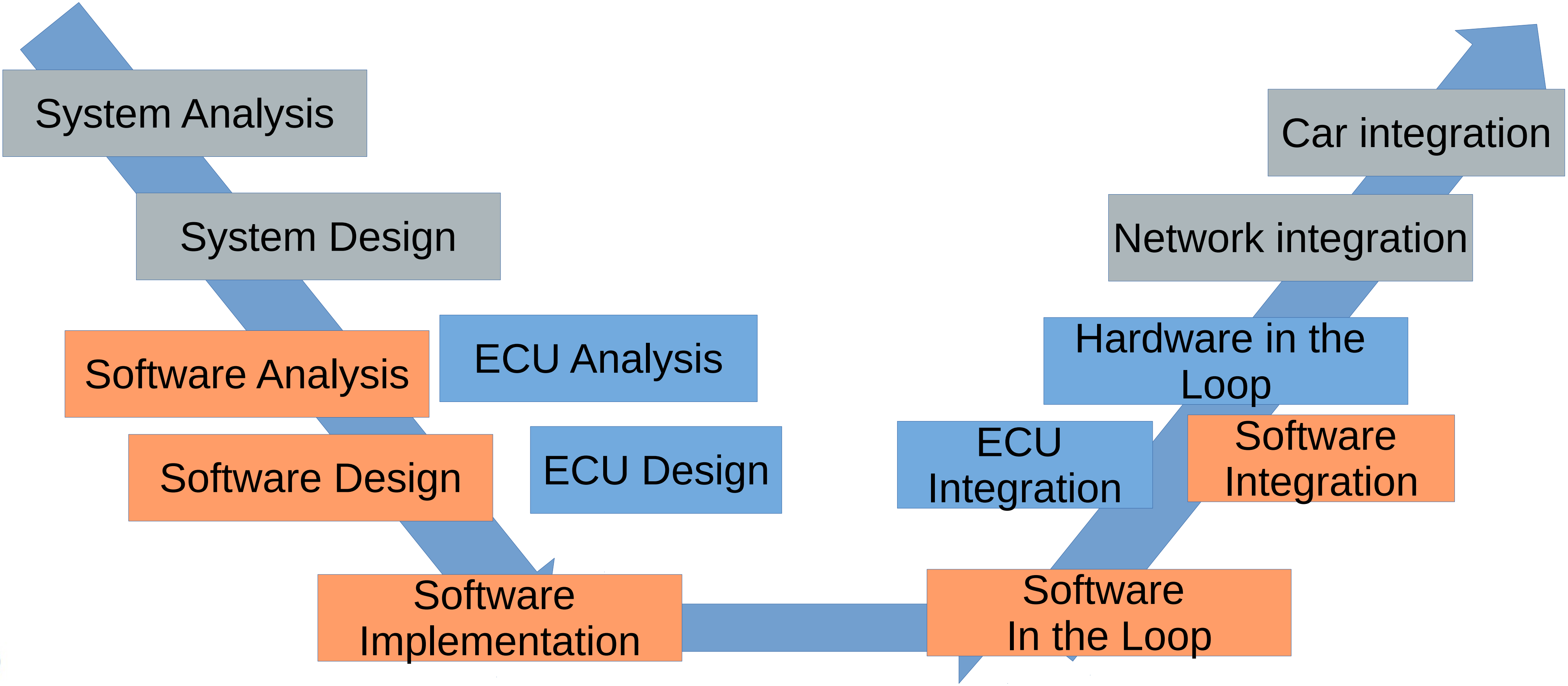
Geometric topology

PreeVision has 3 Tools Steered by Metamodels

[MüGl09]

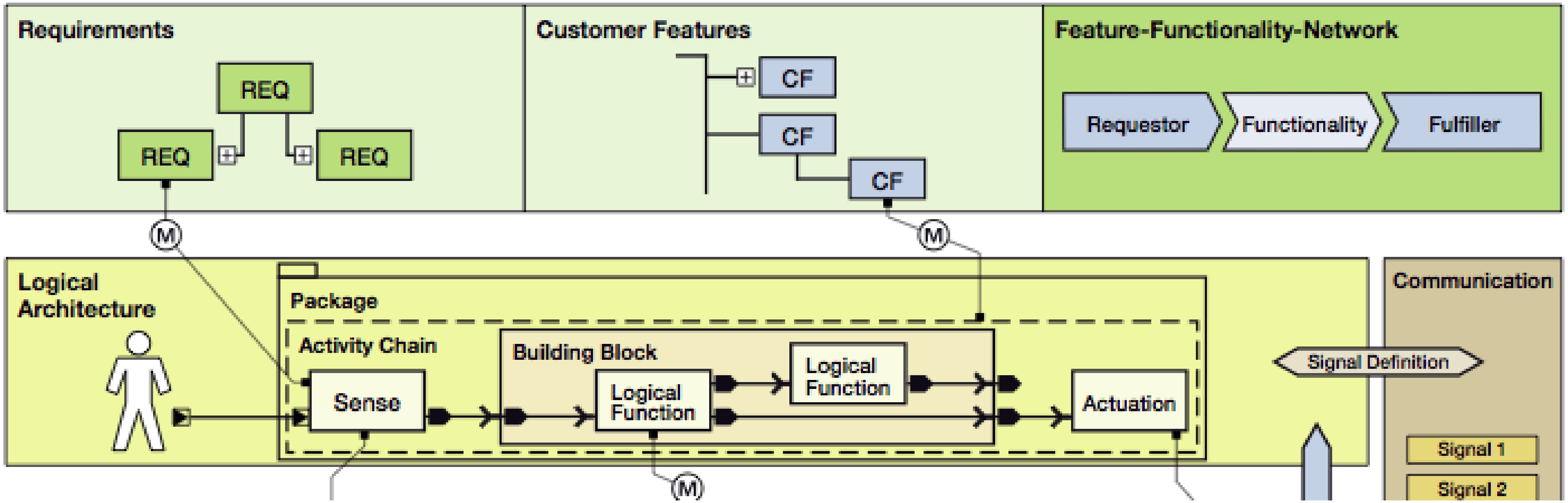
- PREEvision Architect
- PREEvision Function Designer
- PREEvision Electric Designer

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



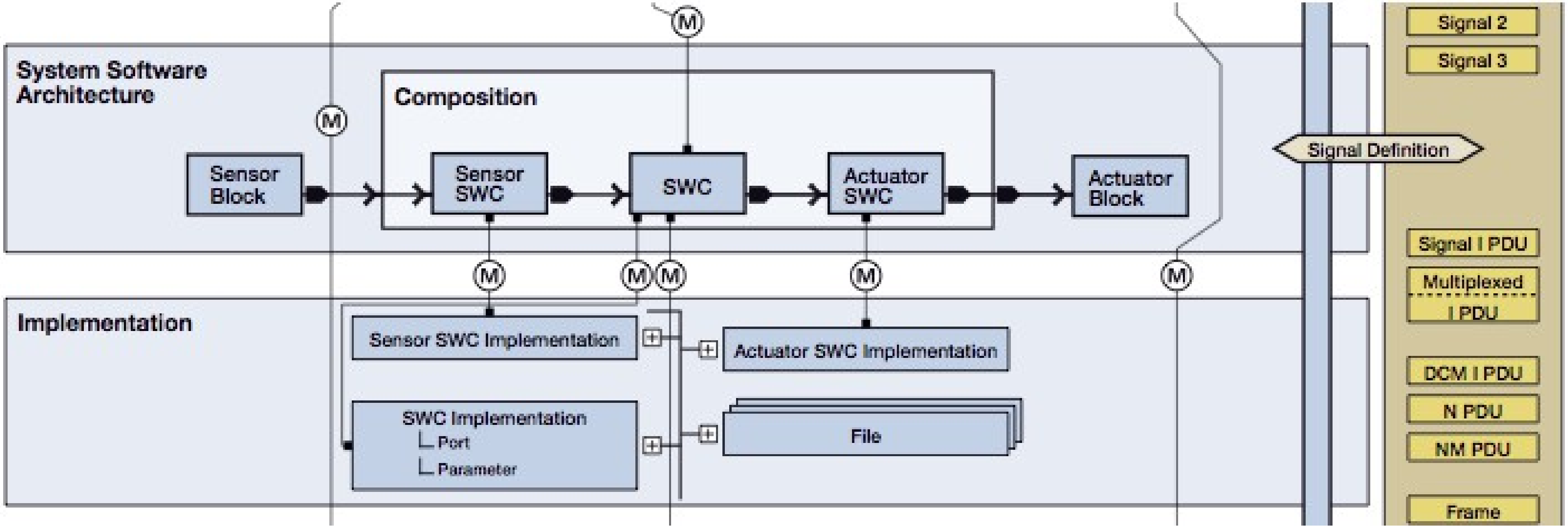
PreeVision Models in More Details

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



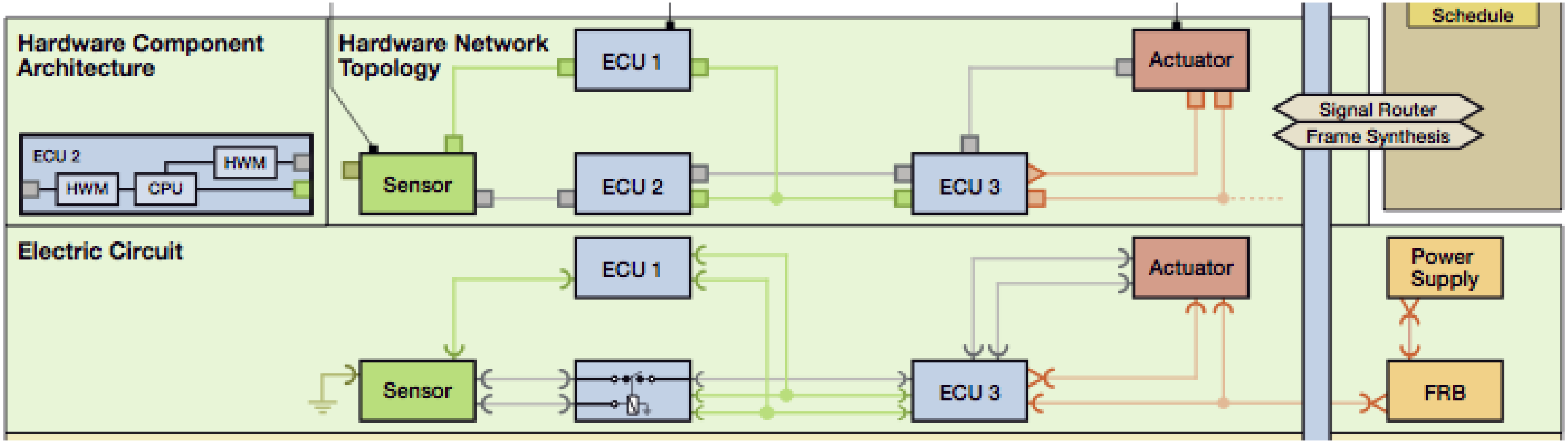
PreeVision Models in More Details

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



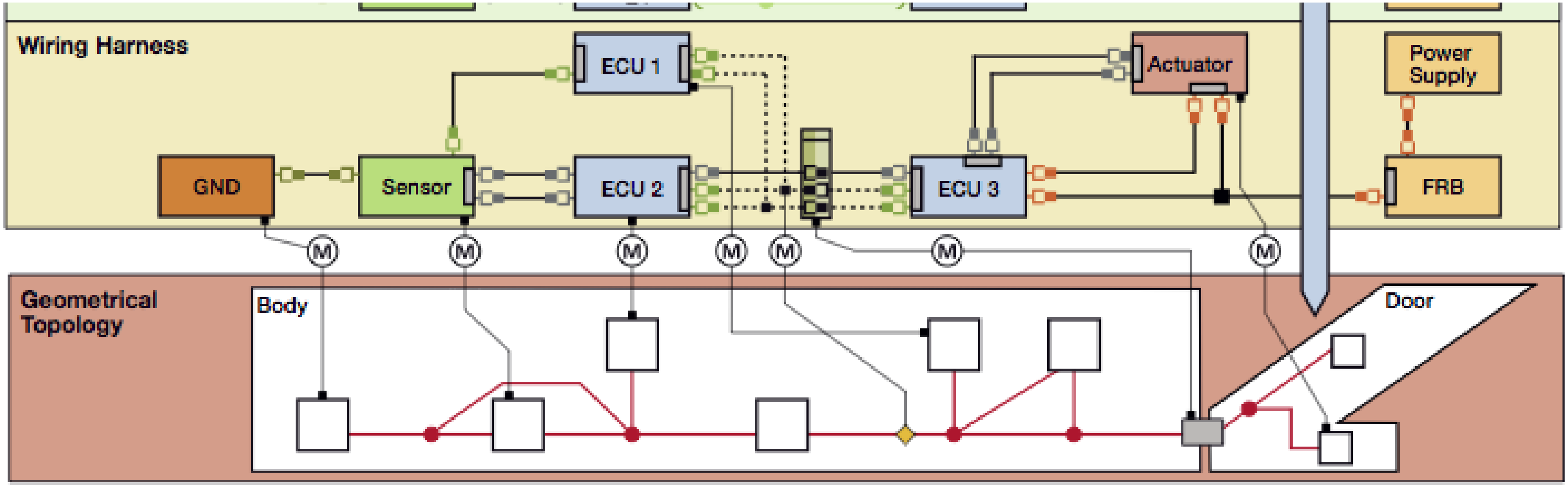
PreeVision Models in More Details

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



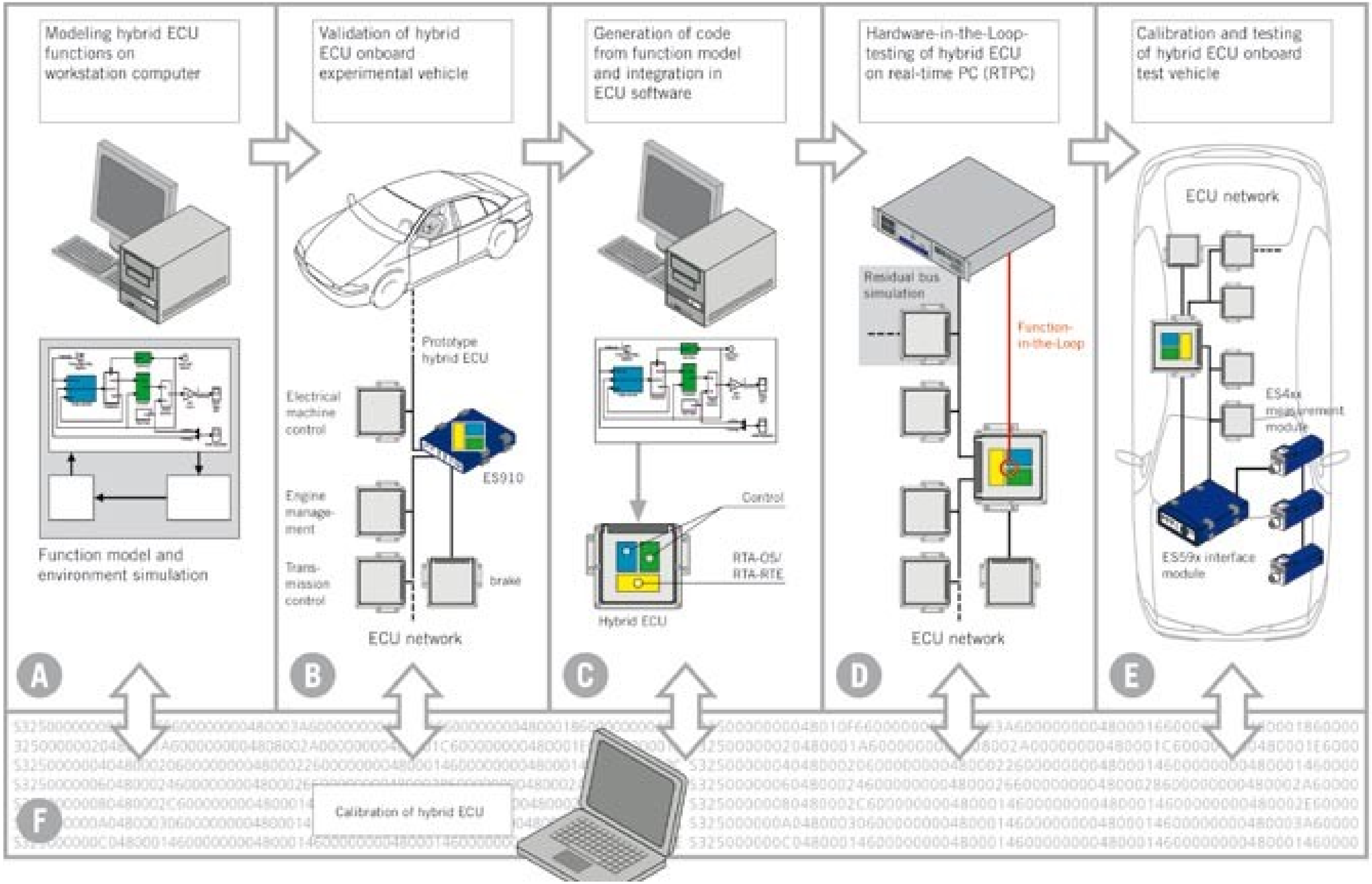
PreeVision Models in More Details

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



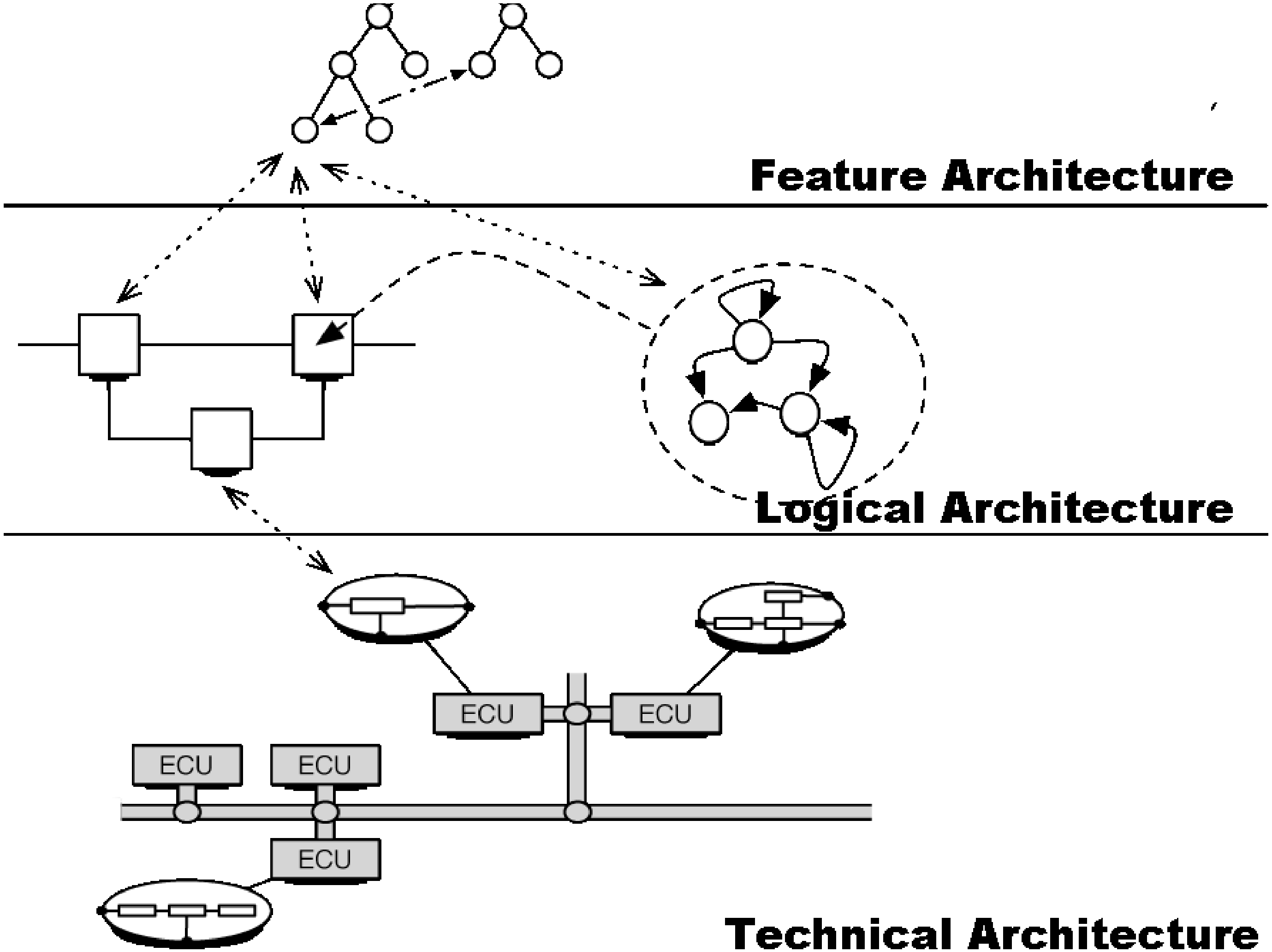
Electric Cars (ETAS)

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

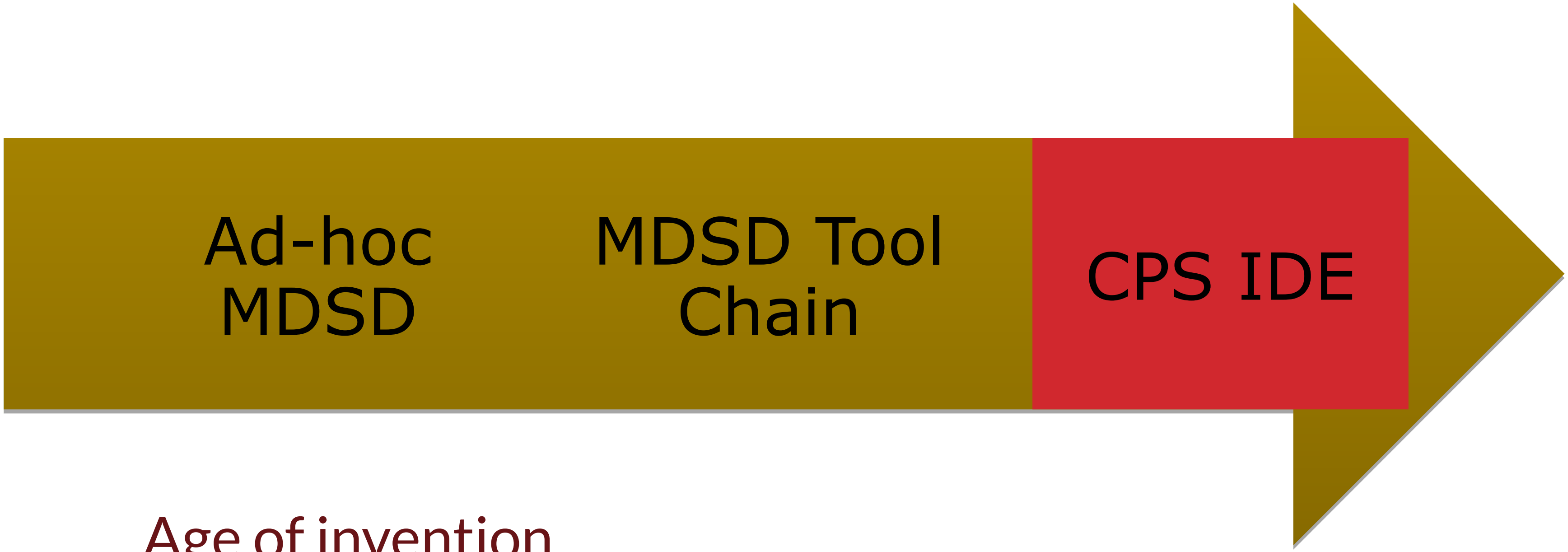

Modeling hybrid ECU functions on workstation computer
Function model and environment simulation
A
Validation of hybrid ECU onboard experimental vehicle
Prototype hybrid ECU
Electrical machine control
ES910
Engine management
Trans-mission control
brake
ECU network
B
Generation of code from function model and integration in ECU software
Control
RTA-OS/ RTA-RTE
Hybrid ECU
C
Hardware-in-the-Loop-testing of hybrid ECU on real-time PC (RTPC)
Residual bus simulation
Function-in-the-Loop
ECU network
D
Calibration and testing of hybrid ECU onboard test vehicle
ECU network
ES4xx measurement module
ES59x interface module
E
F
Calibration of hybrid ECU

COLA (TUM)

[Zverlov]



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



Ad-hoc MDSD

MDSD Tool Chain

CPS IDE

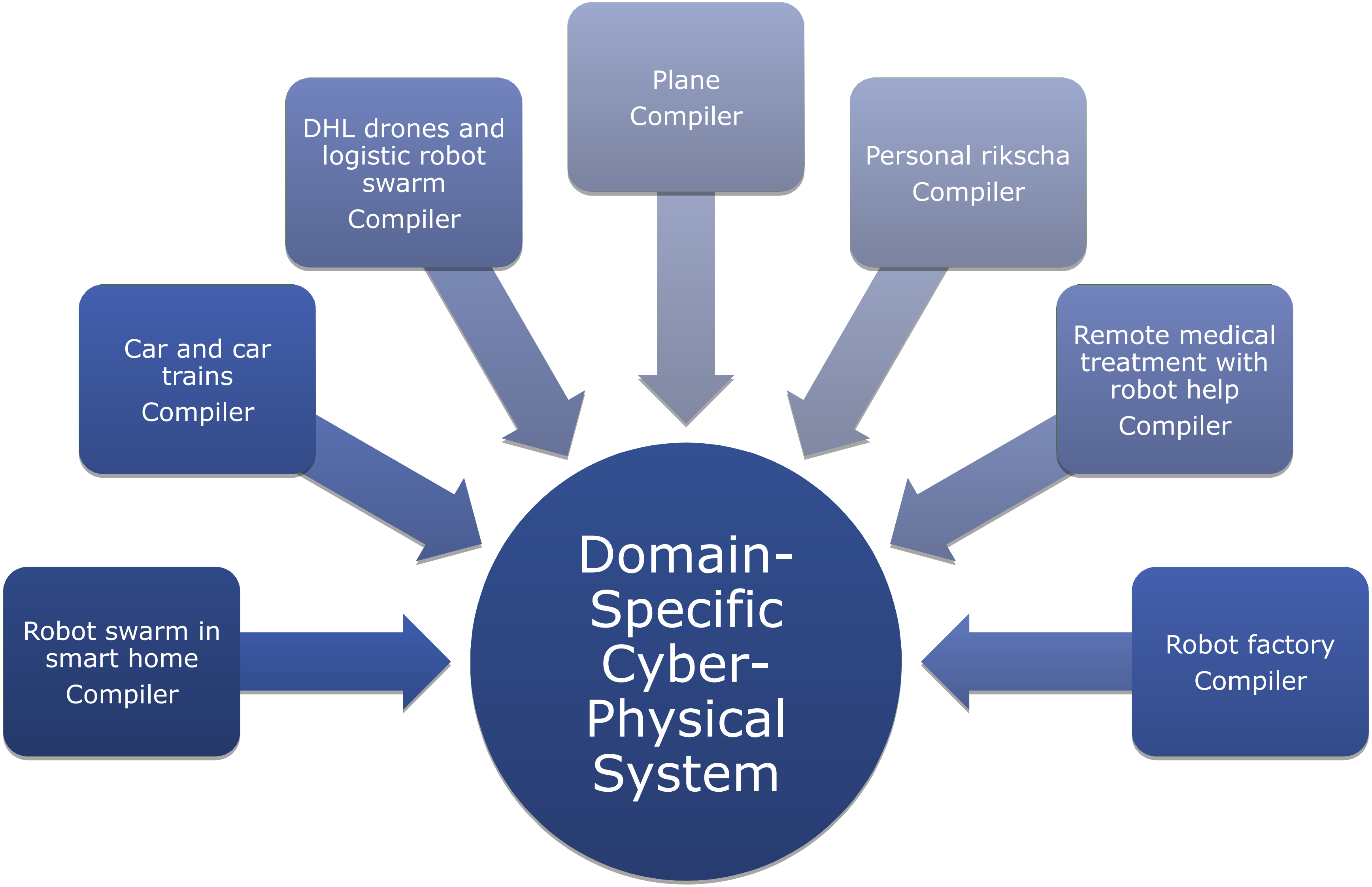
Age of invention

Age of implementation

Age of integration

Will hold for all domains of CPS!

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



DHL drones and logistic robot swarm Compiler

Plane Compiler

Personal rikscha Compiler

Car and car trains Compiler

Remote medical treatment with robot help Compiler

Robot swarm in smart home Compiler

Domain-Specific Cyber-Physical System

Robot factory Compiler

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?

2.3. Experiments with Cloud Robots

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 €

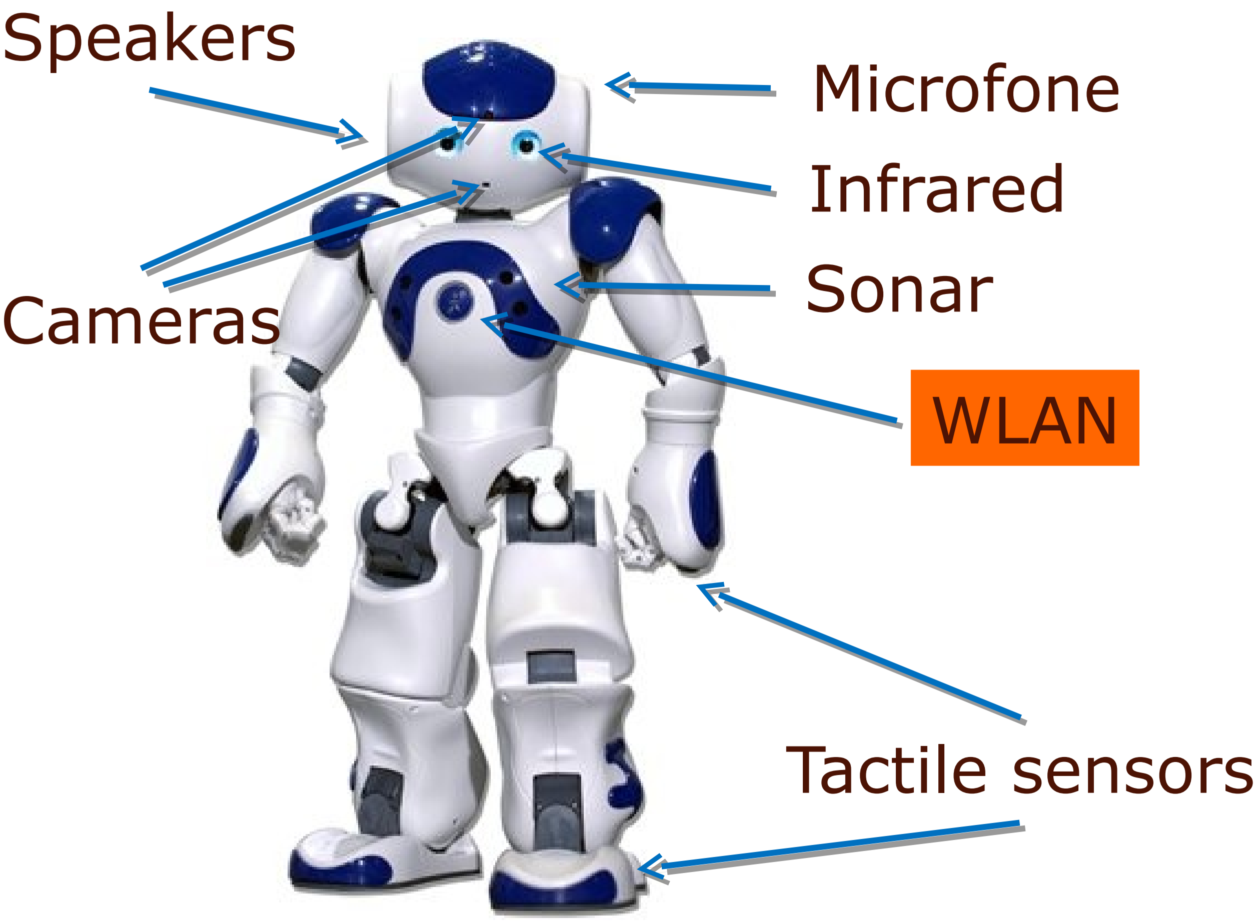
Nao Fact Sheet

Length: 58cm

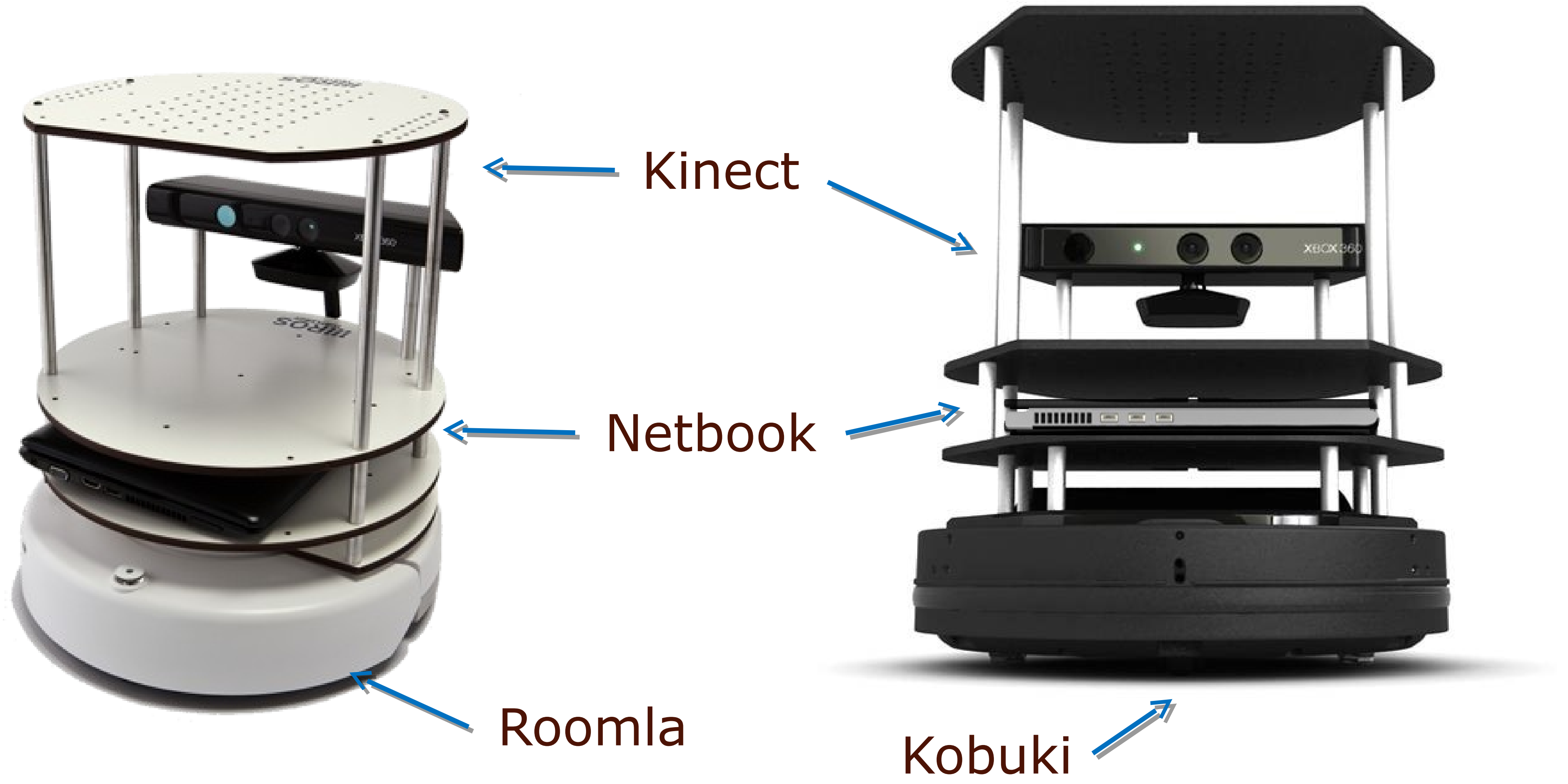
Weight: 5kg

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

OS: Embedded Linux 32bit



Turtle Bot



Kinect

Netbook

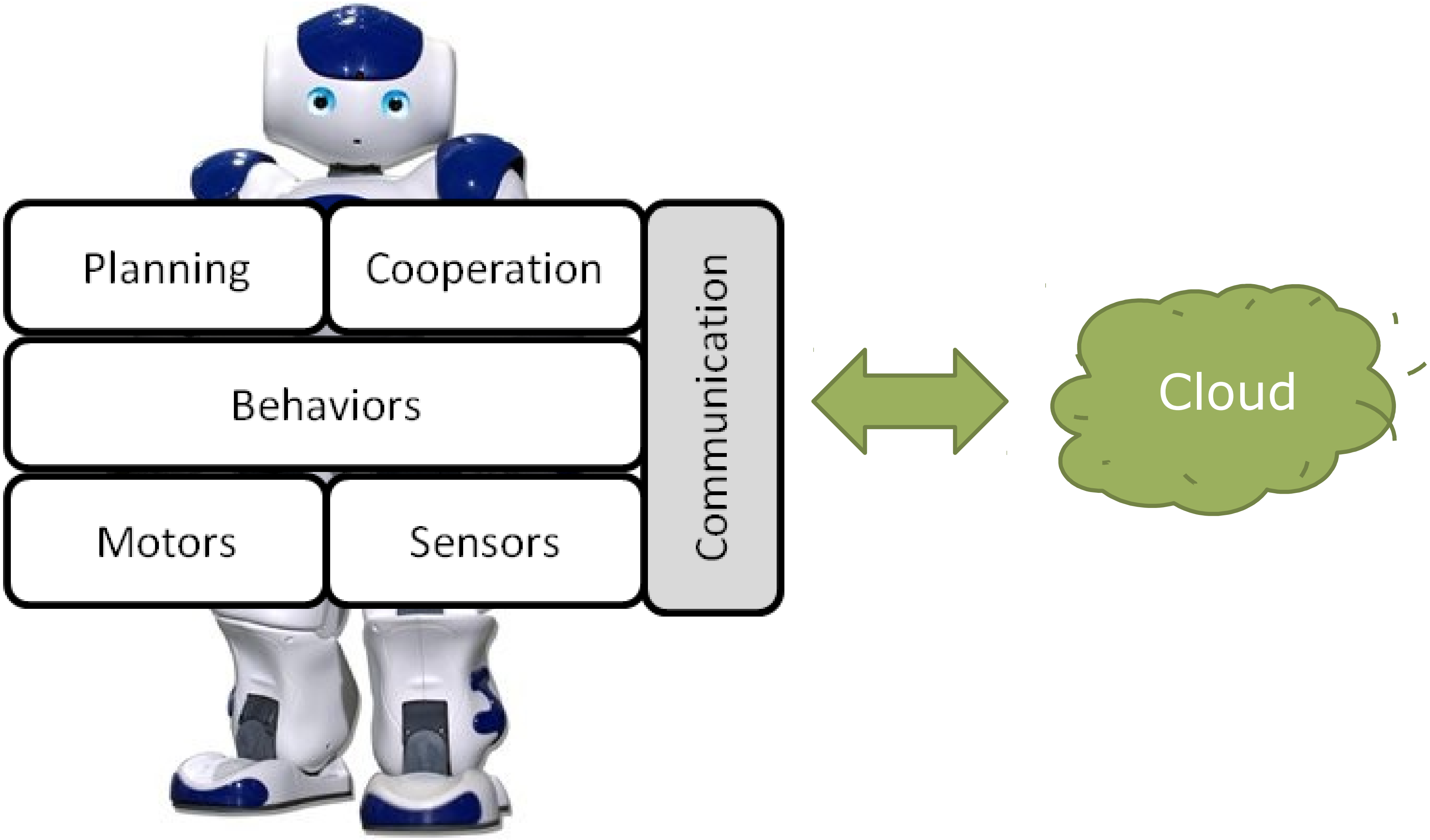
Roomla

Kobuki

50kHz Sensor data rate

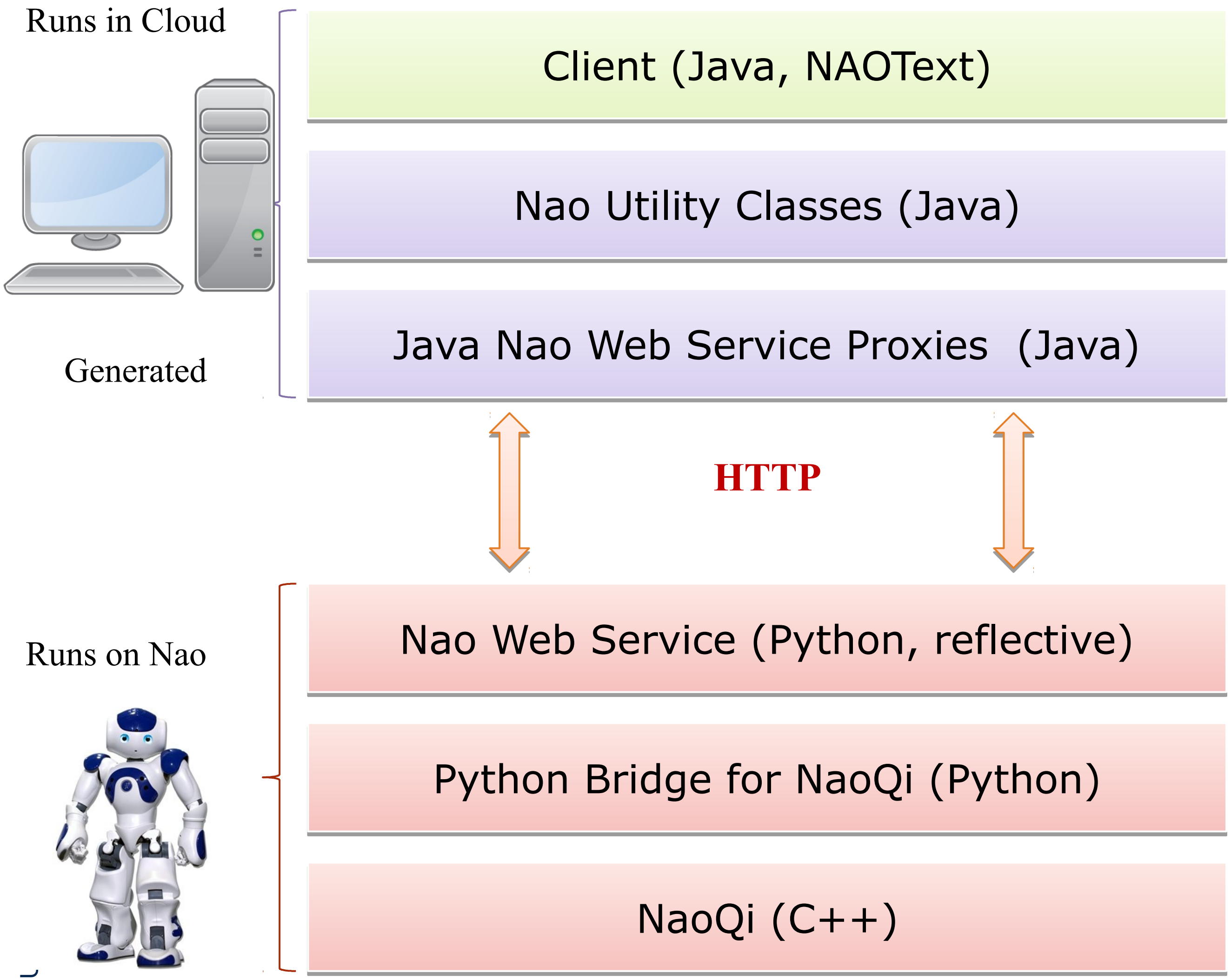
http://wiki.ros.org/Robots/TurtleBot
http://www.turtlebot.com

ResUbic Lab:
NAO Web Service Architecture



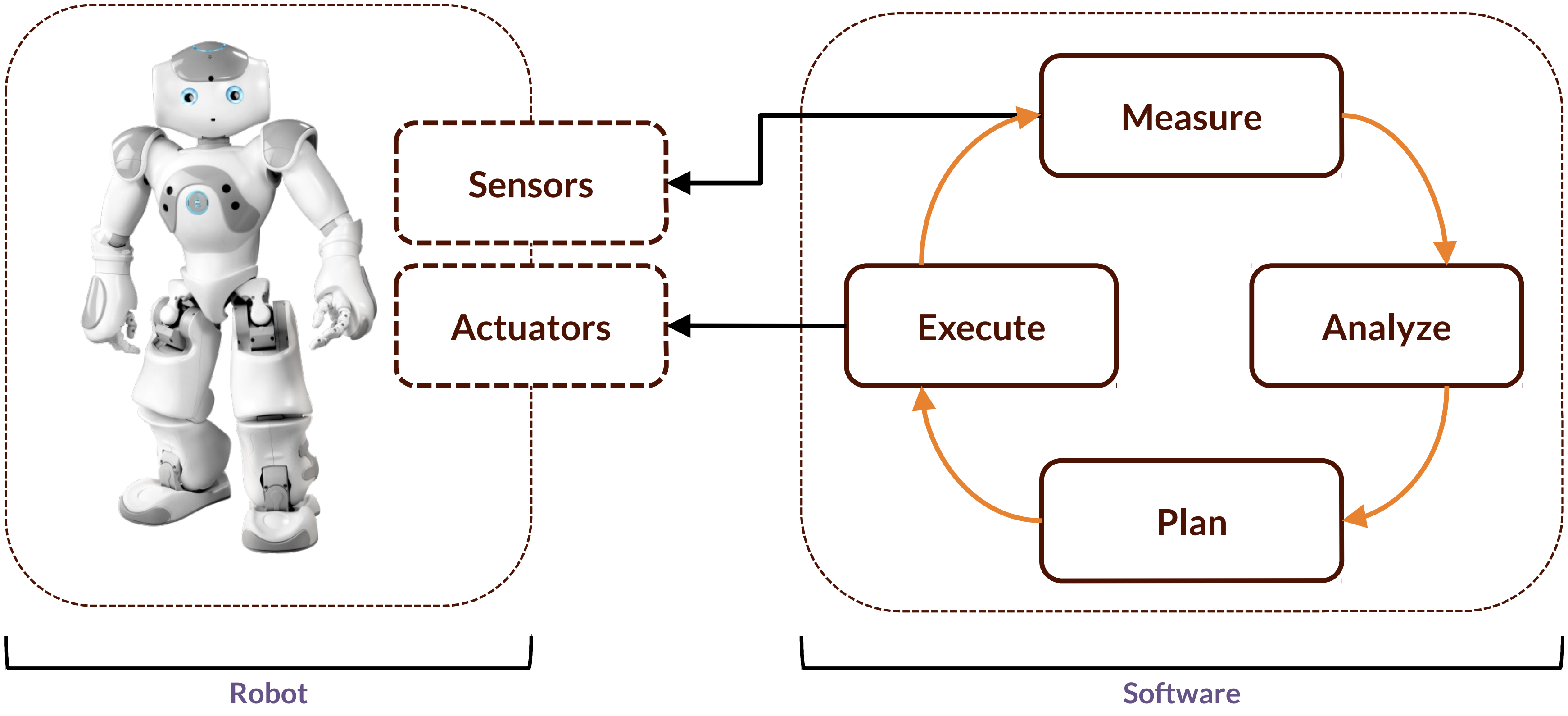
http://code.google.com/p/naoservice/

NAO Web Service and Communication Framework


Runs in Cloud
Generated
Client (Java, NAOText)
Nao Utility Classes (Java)
Java Nao Web Service Proxies (Java)
HTTP
Runs on Nao
Nao Web Service (Python, reflective)
Python Bridge for NaoQi (Python)
NaoQi (C++)

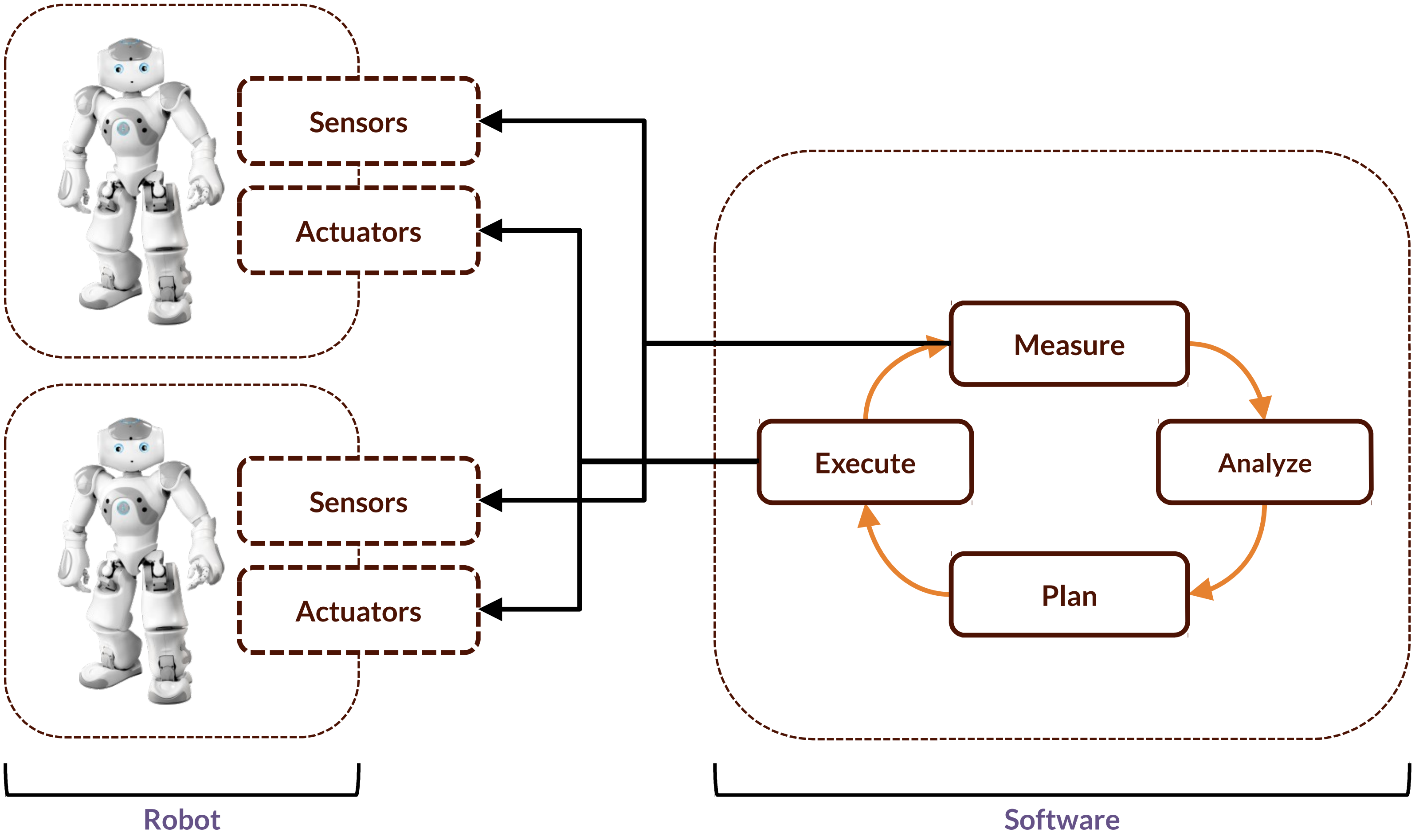
https://github.com/max-leuthaeuser/naoservice

Cloud Robots are Adaptive Systems (MAPE Loop)



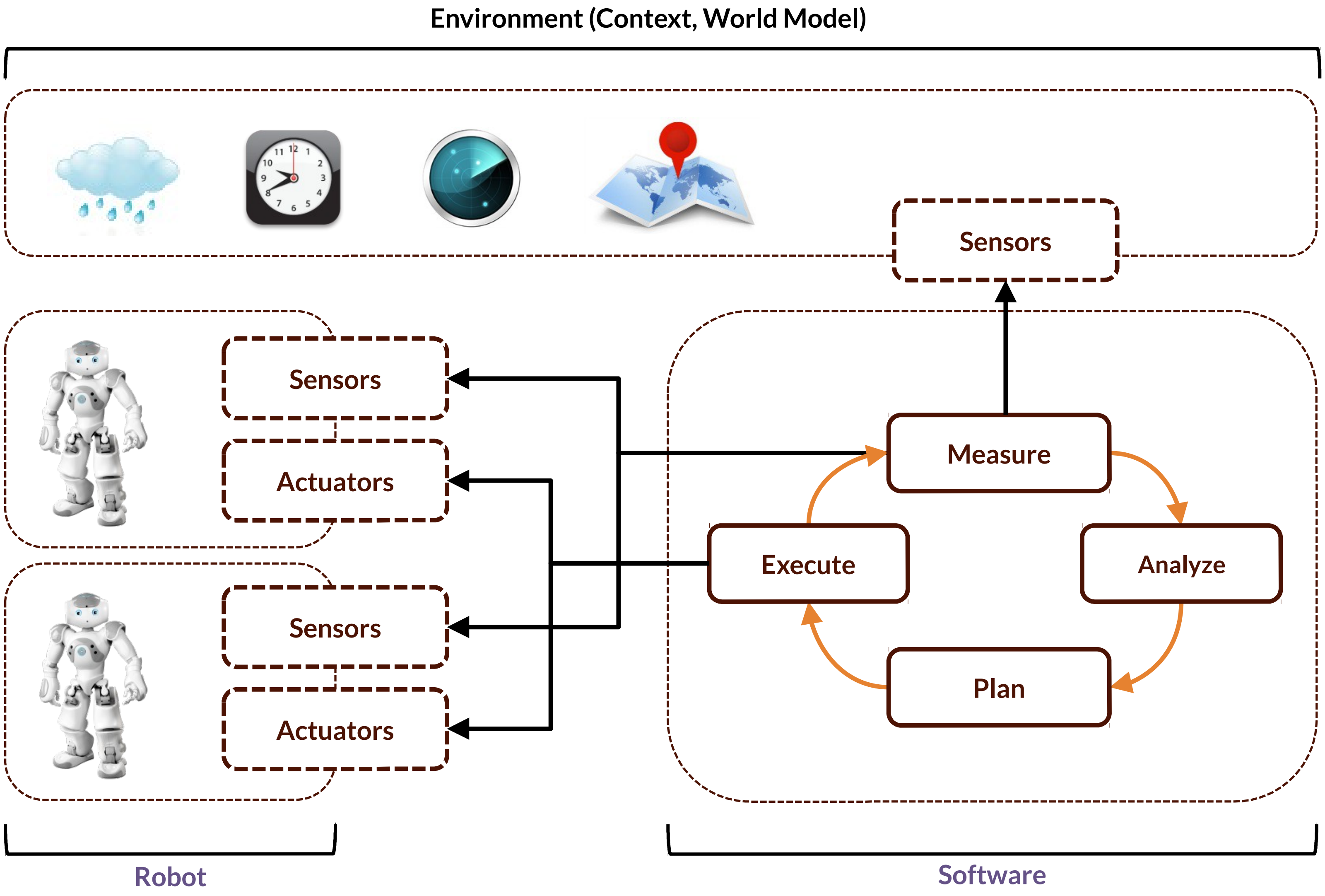
Elastic Architecture

Cloud Robots are Multi-Adaptive Systems



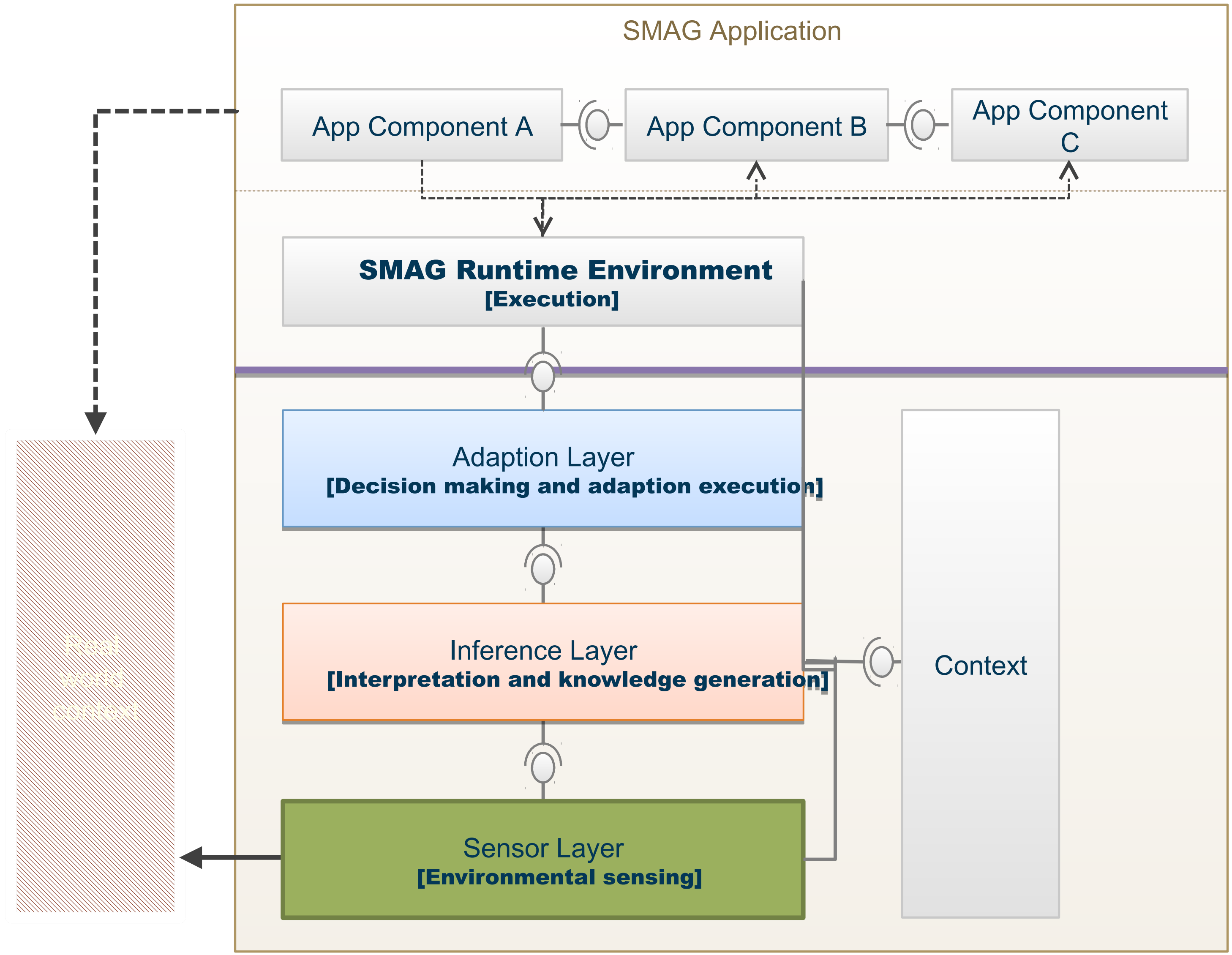
Elastic Architecture

Cloud Robots are Context-Adaptive Systems



Elastic Architecture

The Solution:
Smart Applications on Smart App Grid Infrastructure



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

Donuts Should be Individual....

And

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

Situation Today

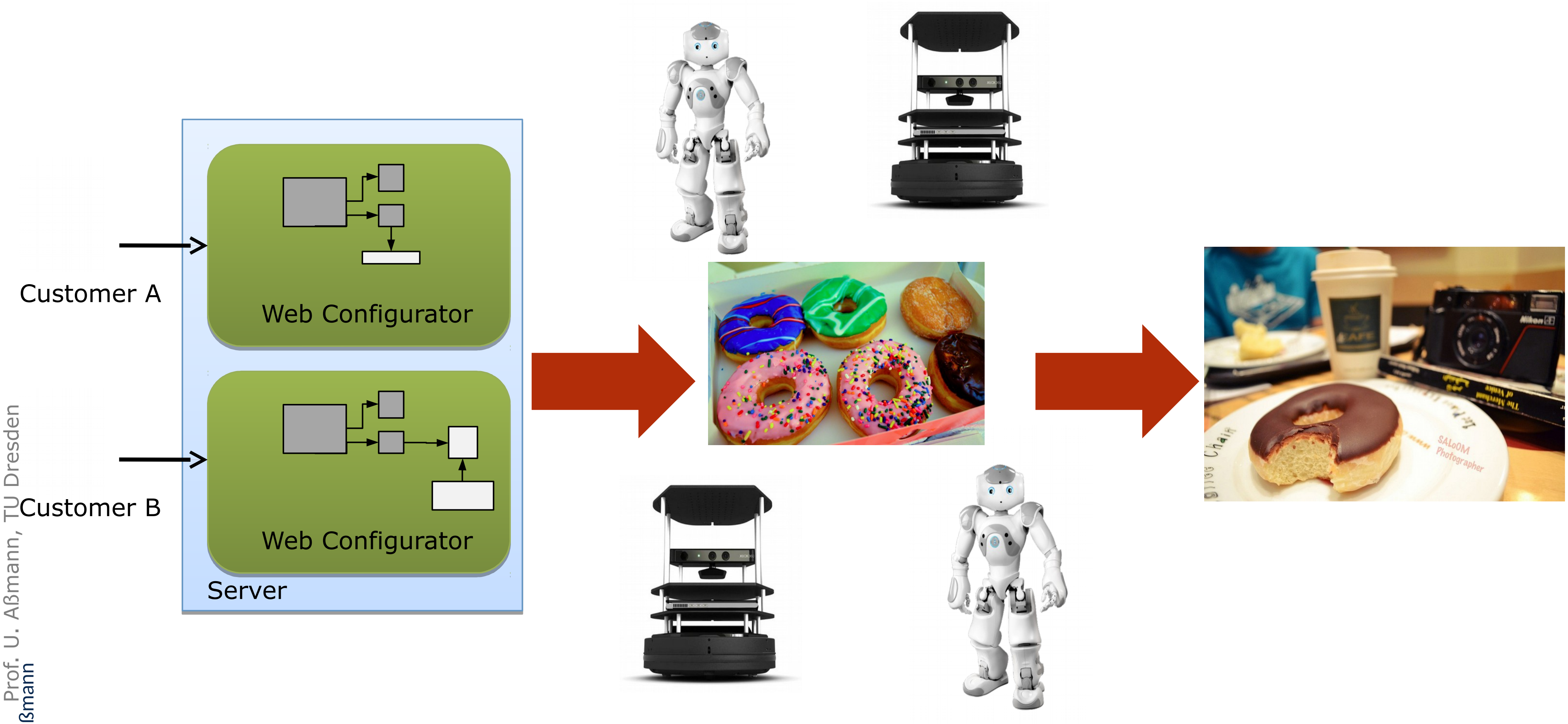
Glazing
(グレージング)
両面がゴールデンブラウンになったドーナツは、
「グレーズのカーテン」をくぐって、できたて
オリジナル・グレーズド・ドーナツに

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

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

Donut Industry-4.0: Pulling Individual Donuts out in Nachtsprung

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



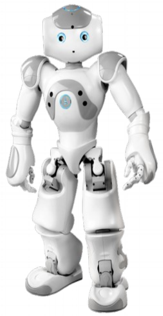
Configuring in the evening

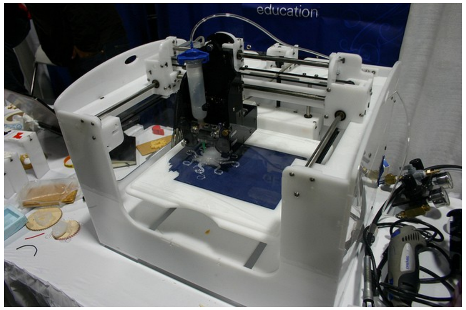
Producing in the night

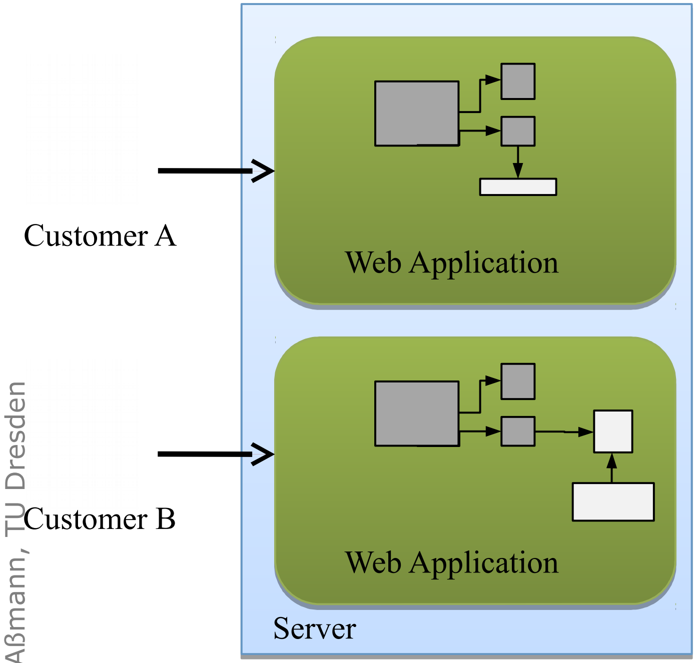
Shipping in the early morning

Industry-4.0: Economic Consequences

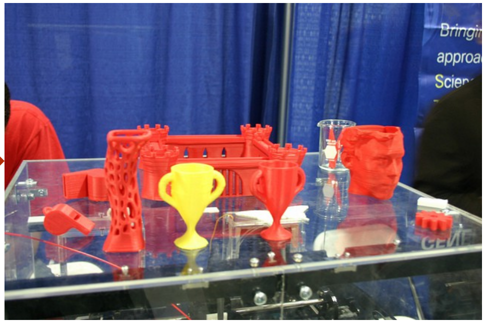
Amazon 2.0

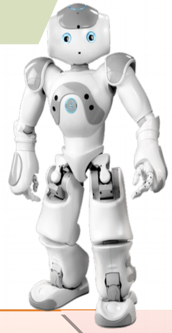






Any Individualized product

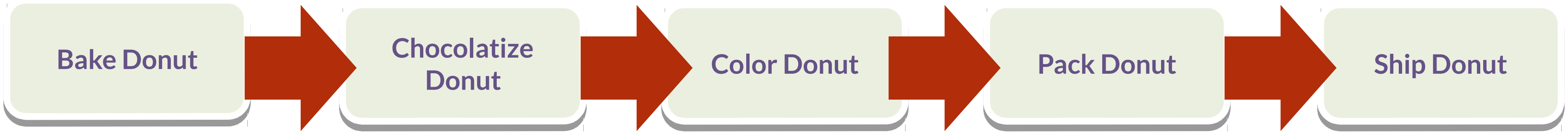




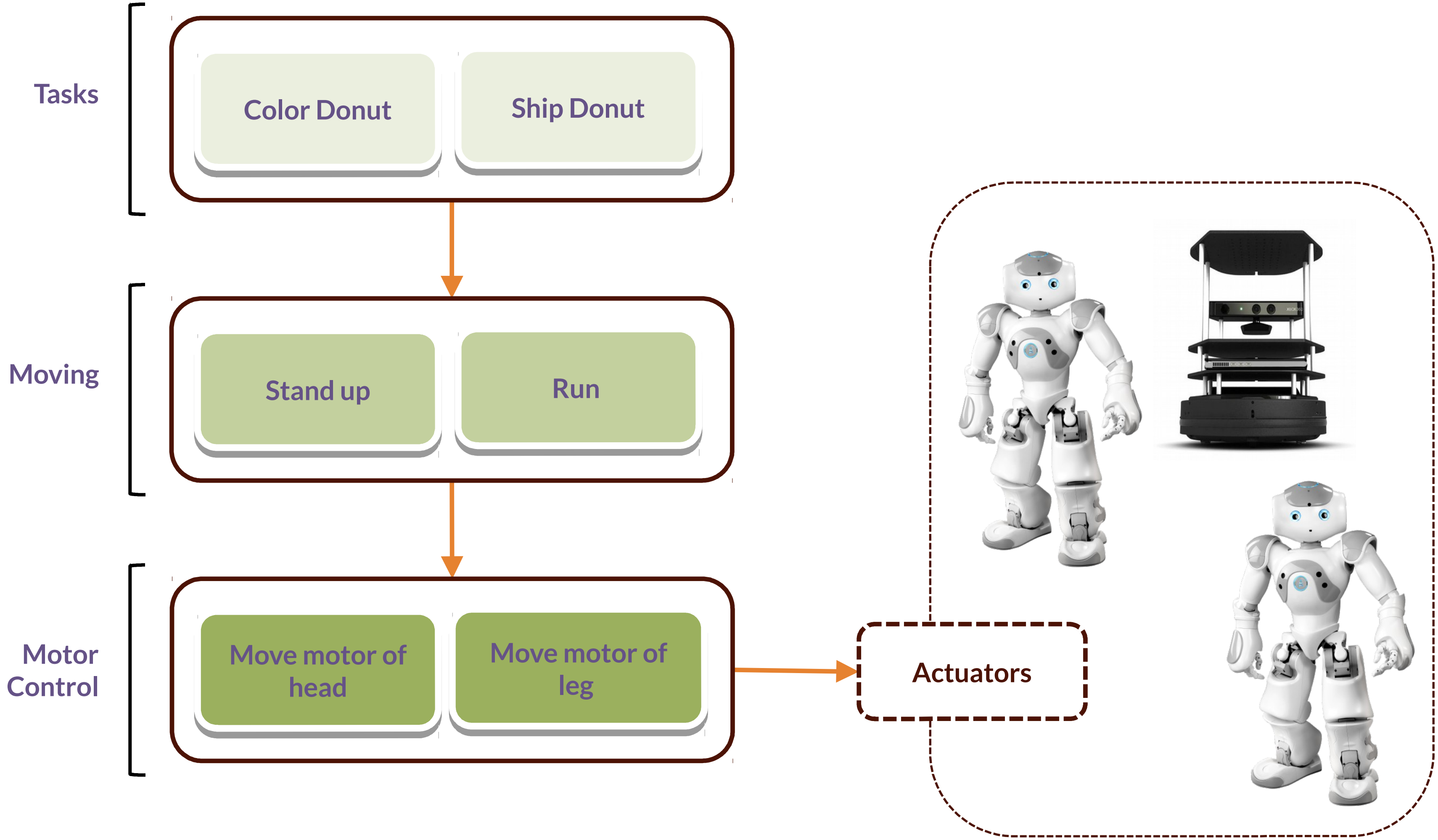
Individualization → Manufactures → Individual Logistics

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

Industry-4.0:
Cloud Robots Produce Things in Workflows



Bake Donut → Chocolatize Donut → Color Donut → Pack Donut → Ship Donut



Tasks: Color Donut, Ship Donut

Moving: Stand up, Run

Motor Control: Move motor of head, Move motor of leg

Actuators

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

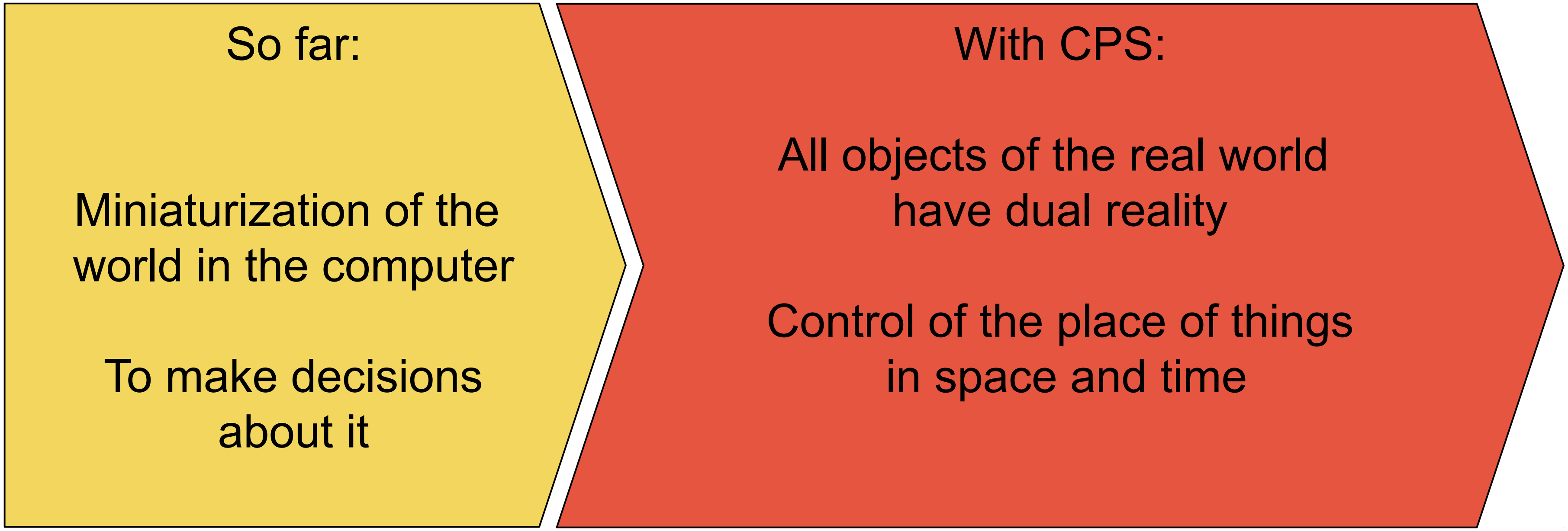
Smart Traffic/Transport/Logistics mit CPS

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

http://www.railcab.de

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



So far:

Miniaturization of the world in the computer

To make decisions about it

With CPS:

All objects of the real world have dual reality

Control of the place of things in space and time

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

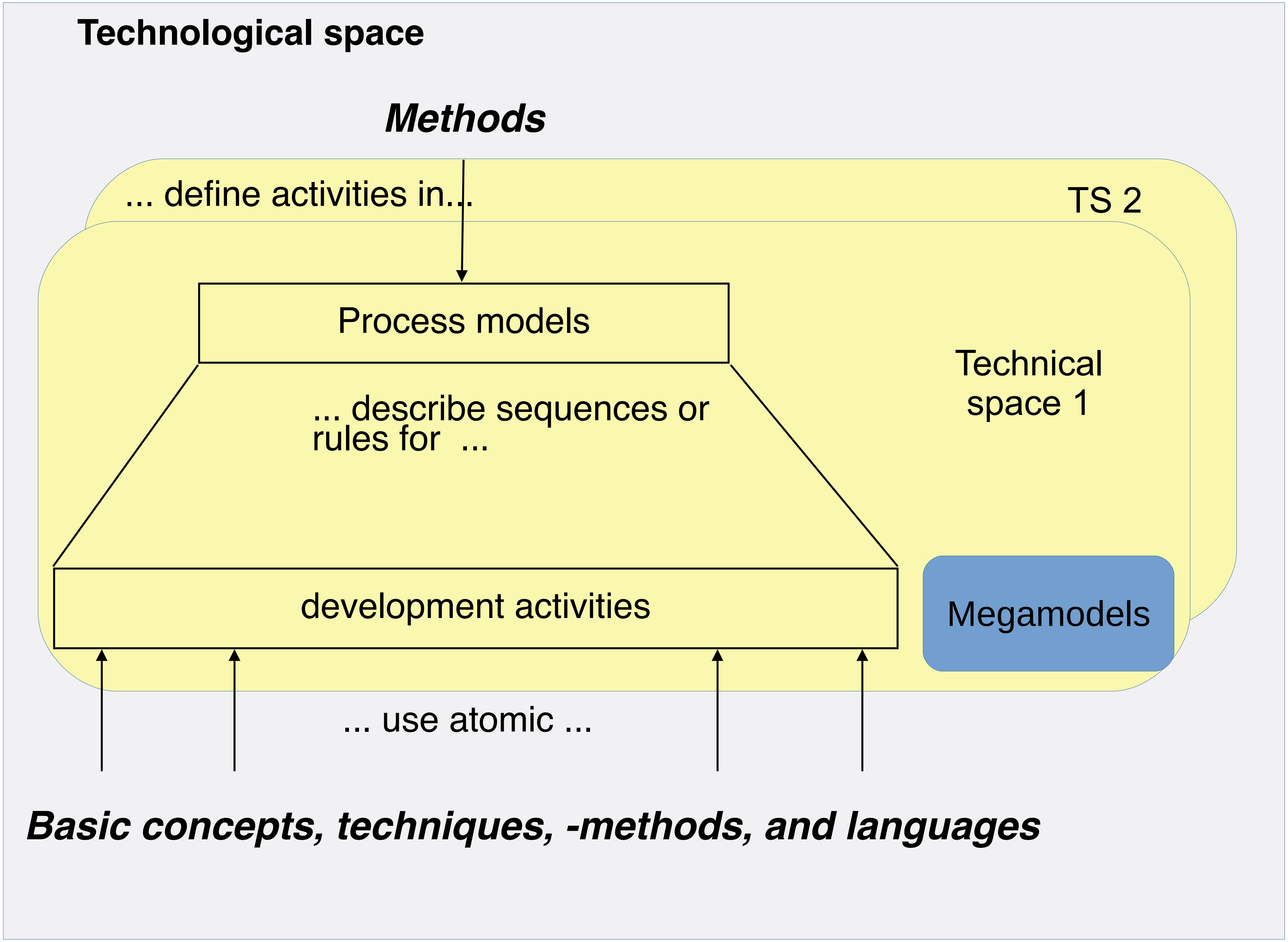
- Metamodels of languages (on M2)
- Models (on M1)
- Repositories (on M0)

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

Software Development is Heterogeneous

[TLS]

[Raasch]



Technological space

Methods

... define activities in...

TS 2

Process models

... describe sequences or rules for ...

Technical space 1

development activities

Megamodels

... use atomic ...

Basic concepts, techniques, -methods, and languages

Method Engineering (Process and Workflow Engineering)

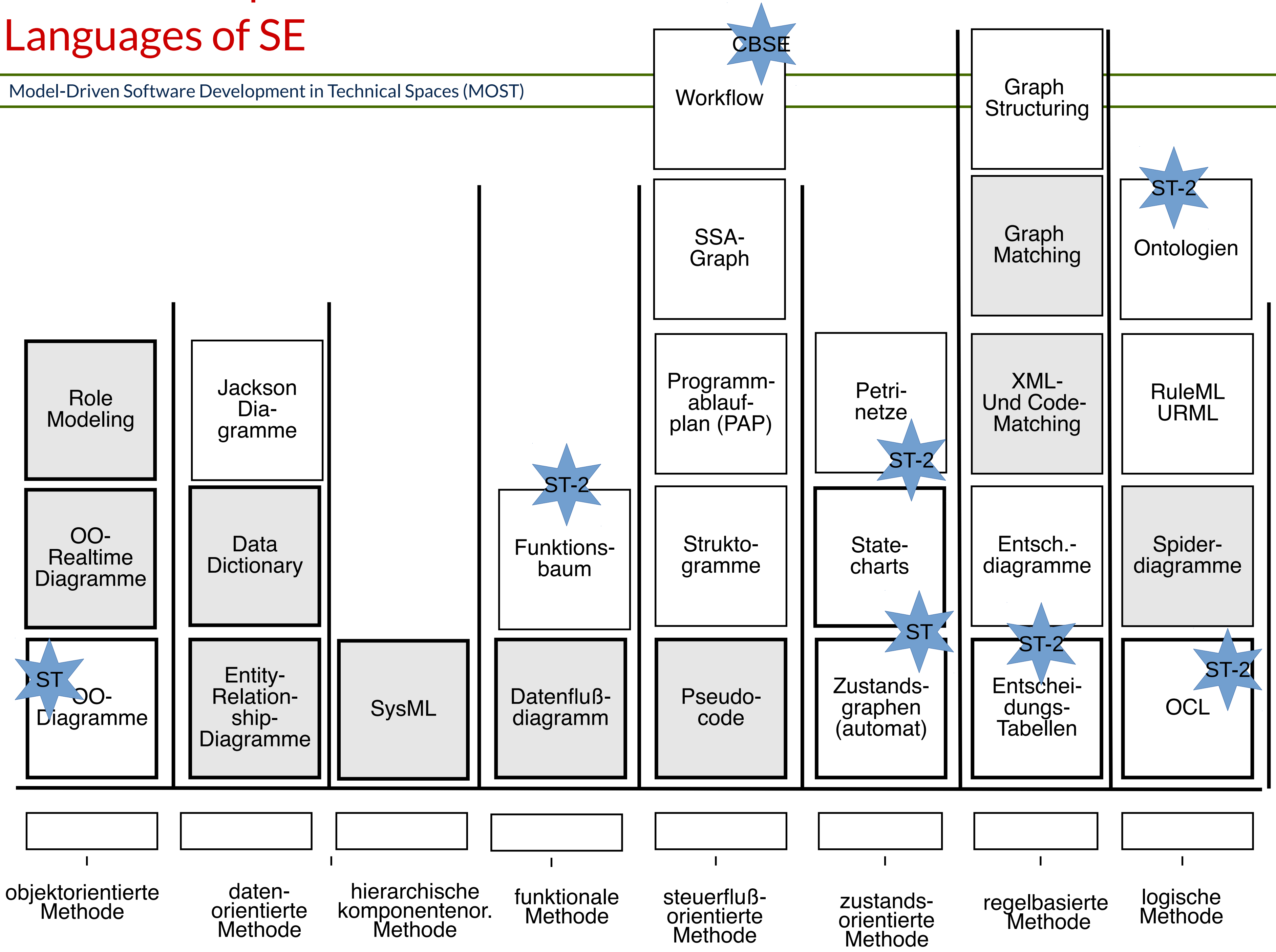
Process Engineering (Method Engineering) is the discipline of specifying and constructing methods and processes for a team of people to conduct a project.
Software Process Engineering (Software Method Engineering) focuses on software development processes.

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)

Basic Techniques and Languages of SE



Quelle: angelehnt an [BAL]

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

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)

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 roles?
- Explain why SMAGs works together with standard platforms easily