



SmartSoft

*Structures, tools and building blocks for
robotics software development*

Christian Schlegel, Alex Lotz,
Matthias Lutz, Dennis Stampfer
Ulm University of Applied Sciences





- What is SmartSoft?
- A brief history of SmartSoft
- Parts of the SmartSoft World
- Benefits of using SmartSoft

What is SmartSoft?



Service Robotics Ulm
autonomous mobile service robots

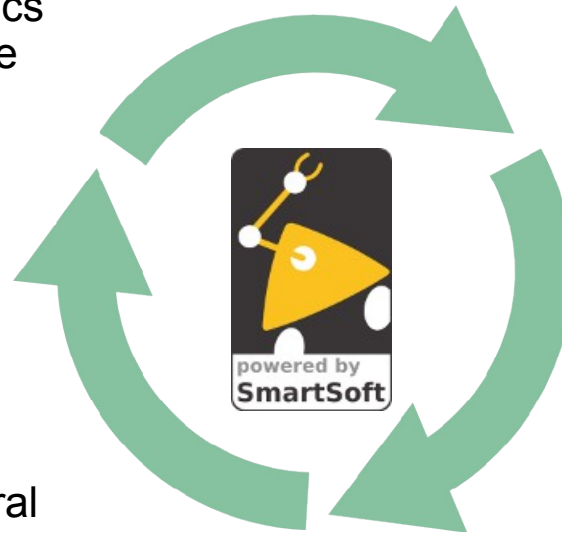
The **SmartSoft World** includes:

Structures

Stable structures guiding robotics software development to enable the flexible **composition** of **building blocks** to robotics systems in an **ecosystem** approach.

Infrastructure

Exchangeable reference **implementations** of the SmartSoft Framework for several platforms and operating systems.



Tooling

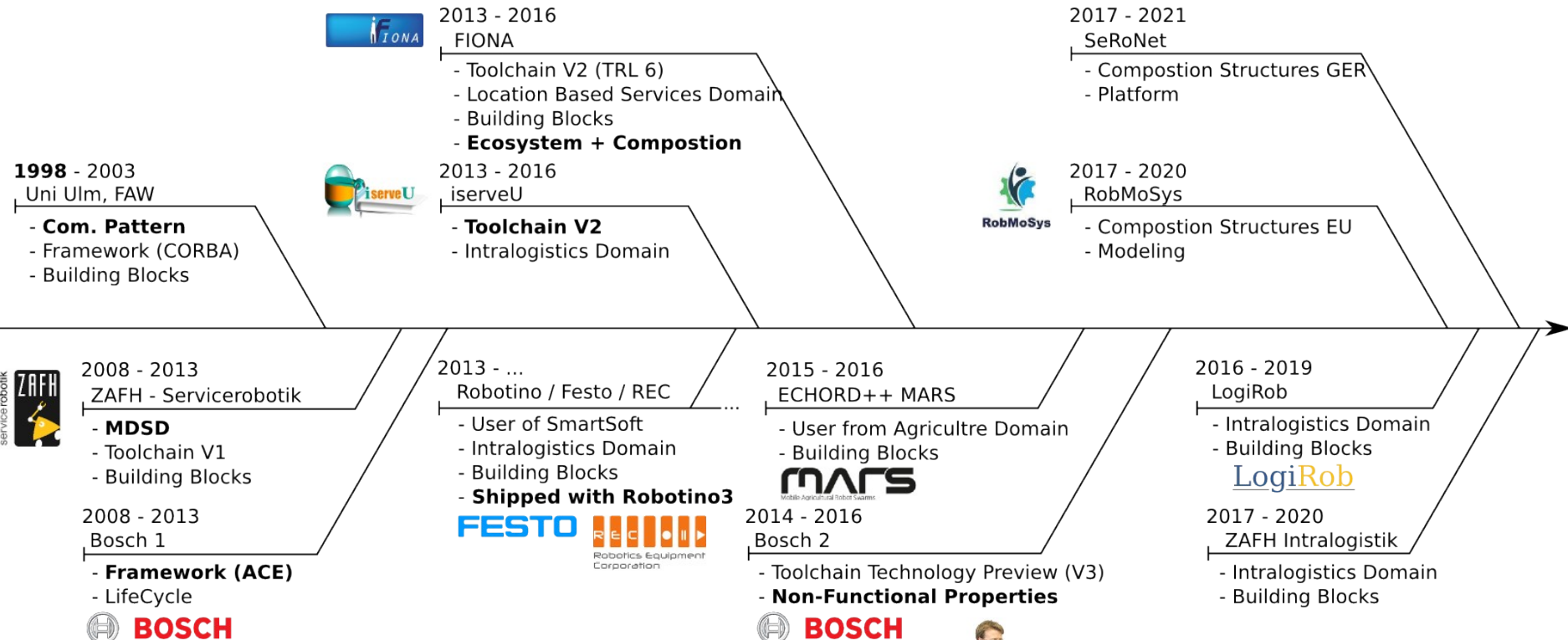
Realize the structures via **MDSD**/meta-models to **make structures accessible** and guide users in applying them.

Software Components

A collection of **building blocks** for **immediate composition** to new robotic systems.

SmartSoft is an umbrella term for **structures, tools and building blocks** to build robotics systems: a systematic development methodology, best practices, implementations and software components.

A Brief History of SmartSoft



The SmartSoft World includes: Structures



We work on structures for:

- Composition workflow
- Composability
- Separation of Roles
- Robotic Behavior

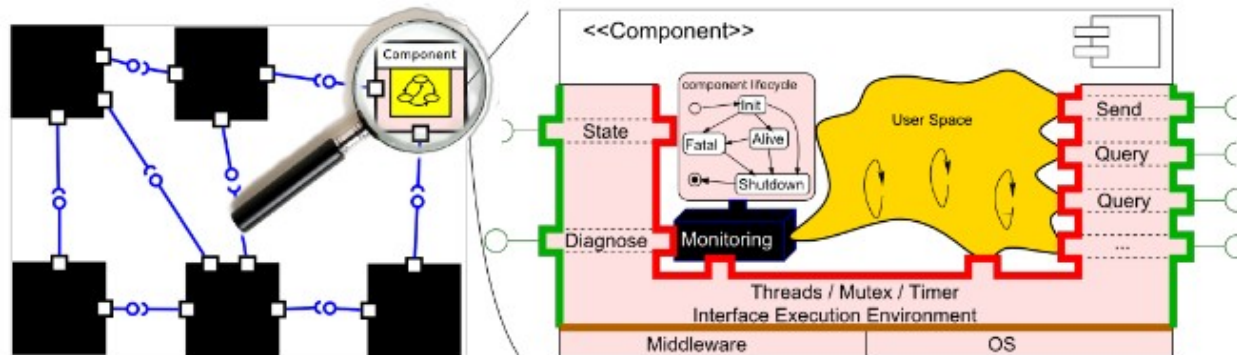
We apply:

- Component-Based Software Engineering
- Service-Oriented Architectures
- Freedom from choice

*Think **SOA** rather than message centric:*

A SOA (service-oriented architecture) has to ensure that services don't get reduced to the status of interfaces, rather they have an identity of their own

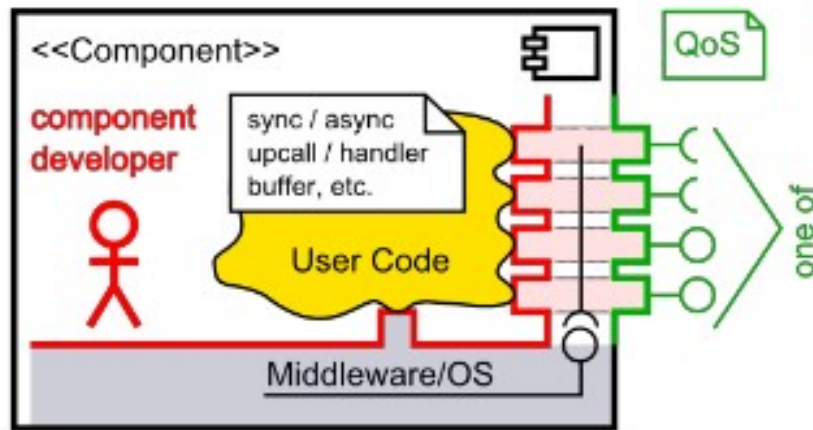
*Gain control over component hull
all relevant properties and parameters
explicated at the component hull*



The SmartSoft World includes: Structures



- Communication Patterns:
Mastering the link between component inside / outside view



Comm. Pattern	Description
Send	one-way communication
Query	two-way request-response
PushNewest	1-to-n publish-subscribe
PushTimed	1-to-n publish-subscribe

Config. Pattern	Description
Parameter	component configuration
State/Lifecycle	activate/deactivate comp. services
DynamicWiring	dynamic component wiring
Event	asynchronous notification
Monitoring	introspection of components <i>(internally based on communication patterns)</i>



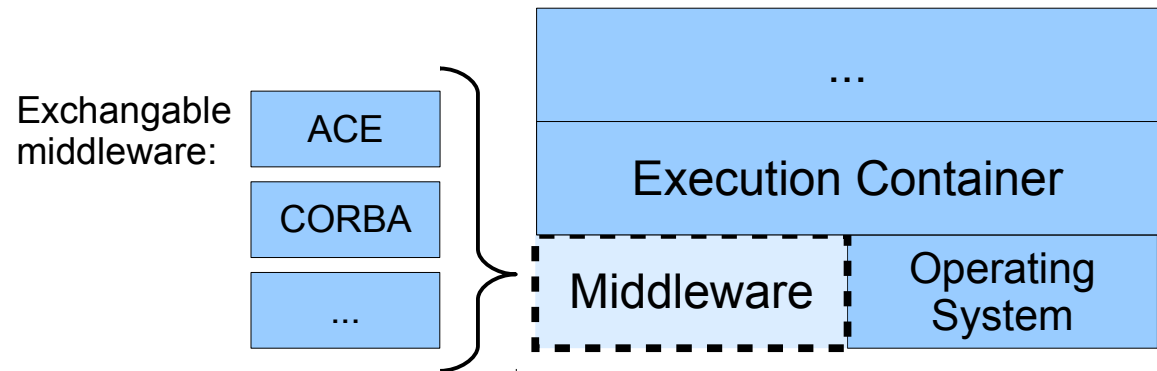
Principles of good service design: [Sprott&Wilkes, 2004]

- reusable: use of service, not reuse by copying of code / implementation
- abstracted: service is abstracted from the implementation
- published: precise, published specification functionality of service interface, not implementation
- formal: formal contract between endpoints places obligations on provider and consumer
- relevant: functionality is presented at a granularity recognized by the user as a meaningful service

The SmartSoft World includes: Infrastructure



- Two exchangeable reference implementations of the SmartSoft Framework
 - Current: ACE middleware, former: CORBA middleware
- Additional tooling and build infrastructure
- Support for several platforms:
 - Linux, but also Windows, iOS, macOS
 - PC, but also ARM/RaspberryPi, powerPC

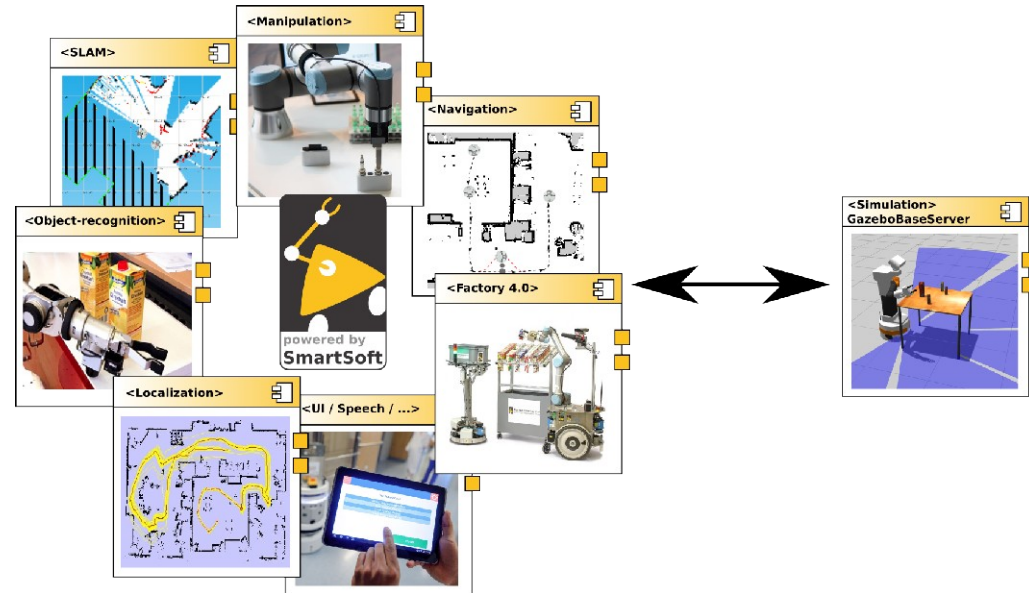


The SmartSoft World includes: Software Components

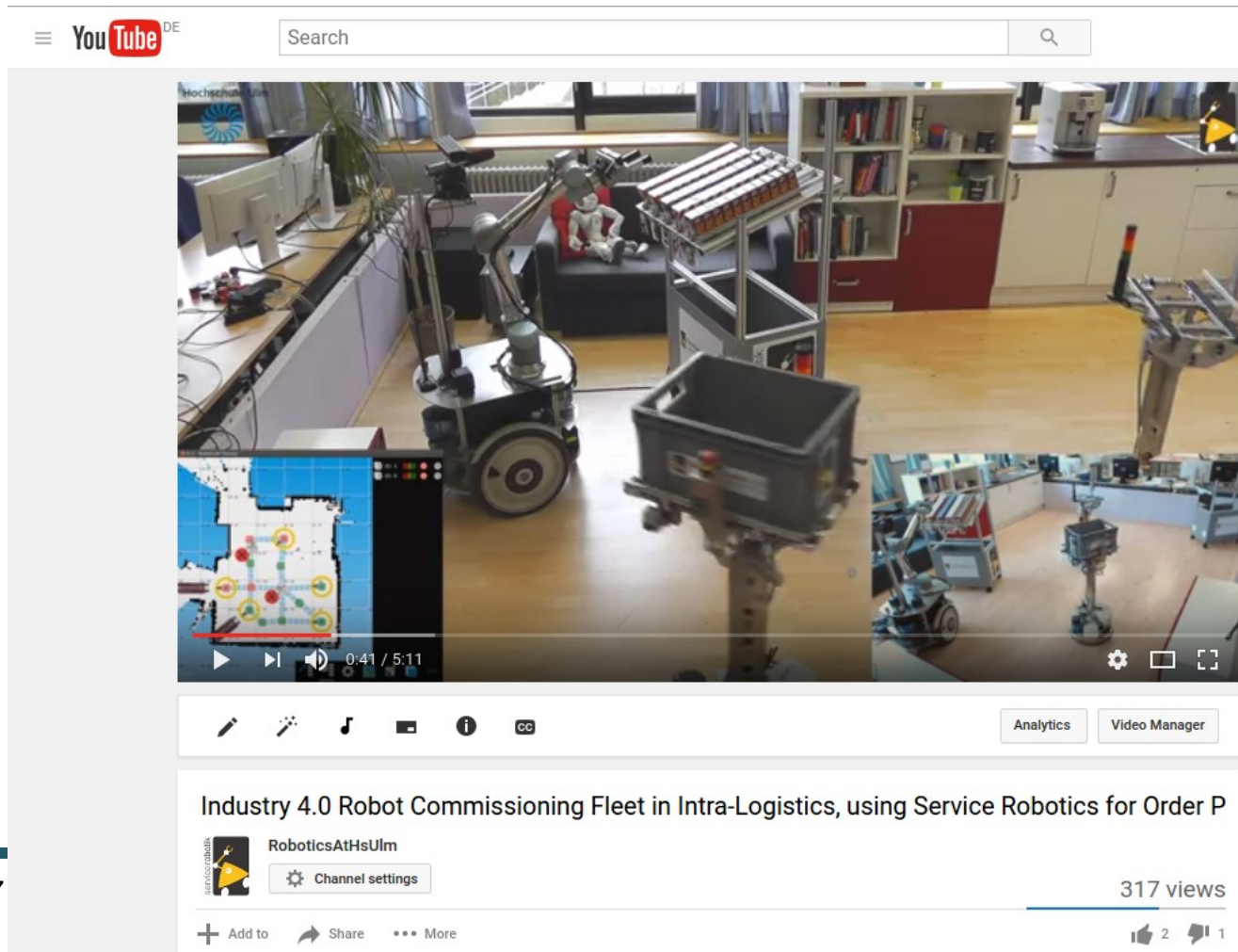


Service Robotics Ulm
autonomous mobile service robots

- A collection of **SmartSoft components** is readily available under **Open Source Licenses** at <http://www.servicerobotik-ulm.de>
- They have been developed using the SmartMDSD Toolchain and are **available for immediate reuse**, for example in robotics systems for:
 - Object Recognition
 - Navigation
 - Manipulation
 - Localization
 - Fleet coordination
 - Simulation
 - Human-Machine-Interaction
 - Machine-to-machine communication



- Industry 4.0 Robot Commissioning Fleet in Intra-Logistics, using Service Robotics for Order Picking (<https://www.youtube.com/watch?v=qRSDxBOUVx0>)



YouTube DE Search

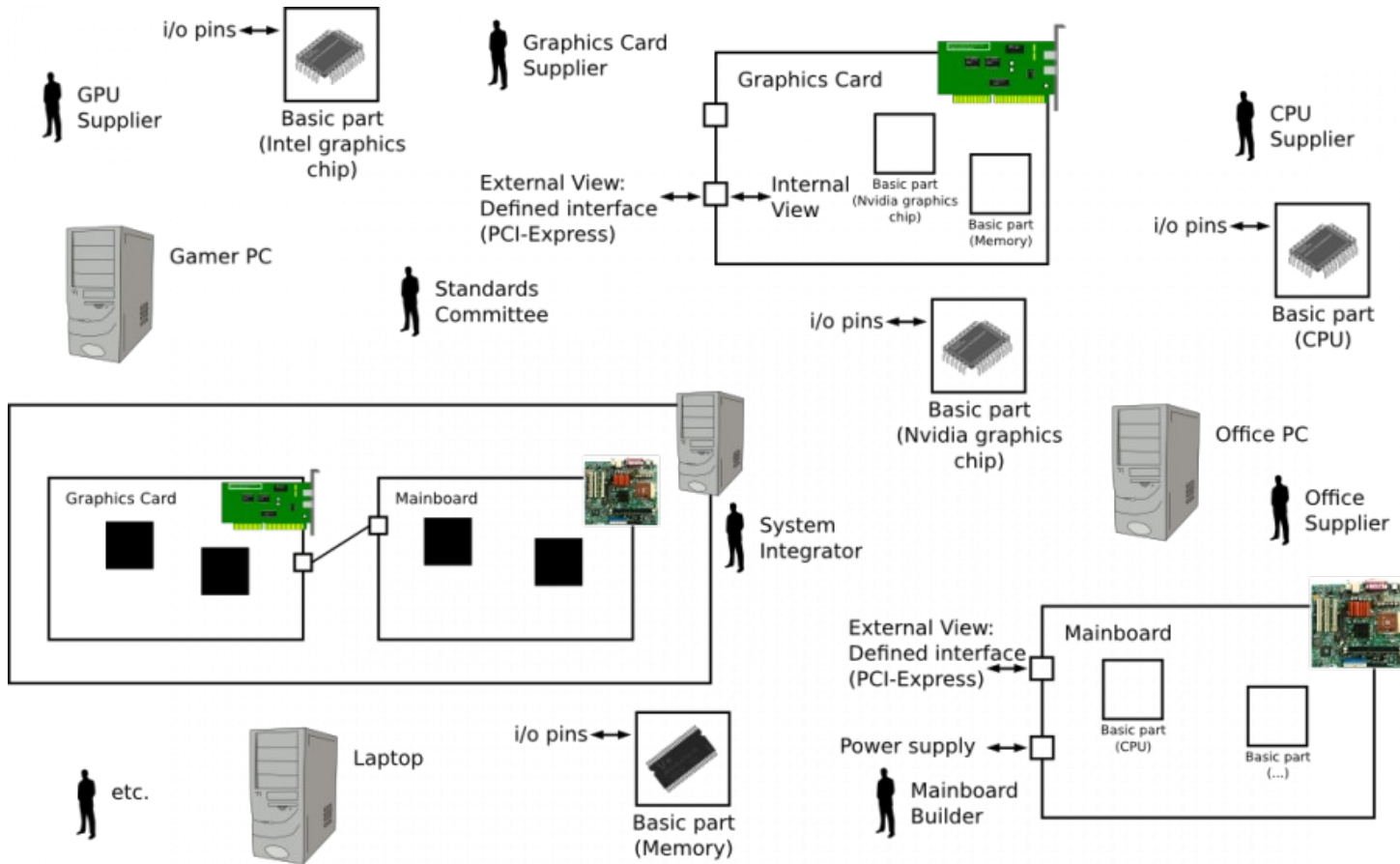
Industry 4.0 Robot Commissioning Fleet in Intra-Logistics, using Service Robotics for Order P

RoboticsAtHsUlm Channel settings

317 views

Add to Share More

An Analogy: The PC Domain



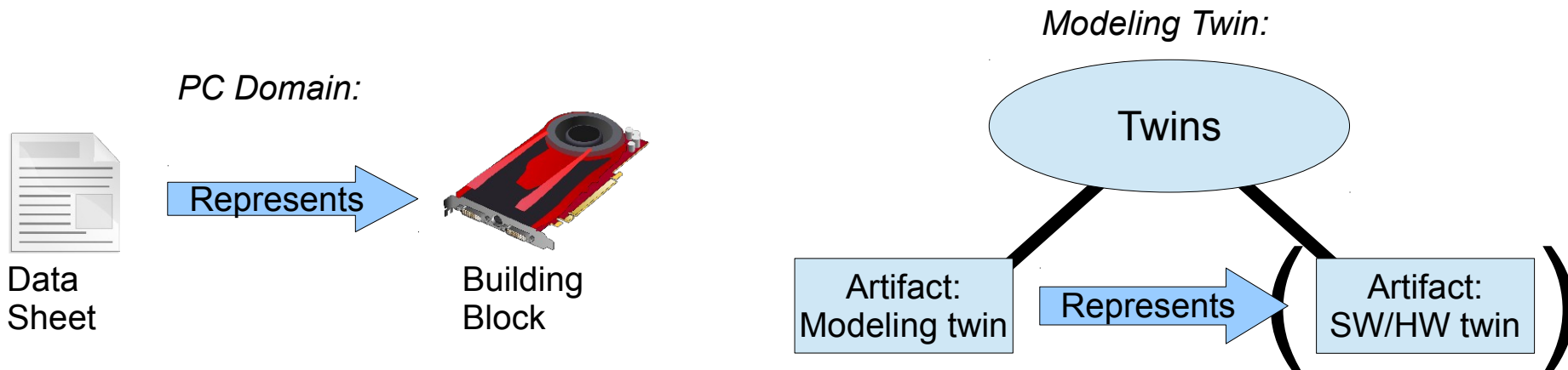
- building blocks with data sheets
- different stakeholders in different roles
- composition instead of integration

An Analogy: The PC Domain



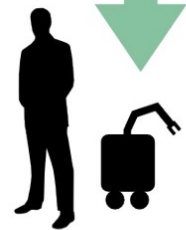
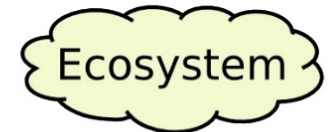
What enables composition of building blocks in the PC domain?

- Building blocks adhere to **superordinate structures** (e.g. PCIe) and explicate properties in **data sheets** (e.g. power supply, form factor, thermal information)
- This enables:
 - Views
 - Decoupled supply and use
 - Flexible IP



Ecosystem Participant "Max"

- Imagine you as an integrator are willing to develop an application which **needs a localization** module and you are interested in **integrating the third-party localization** software on your intralogistics mobile platform.
- You are
 - a **SME** that wants to **access robotics technology** and that wants to **build a robot** application
- You want
 - to **select components** from the market matching your expressed needs
 - your application to be **correct by construction**: you expect that building blocks seamlessly fit together
 - to view components as **grey-boxes** and use them **"as-is"**: adjust only at explicated variation points within modeled boundaries, do not modify source code



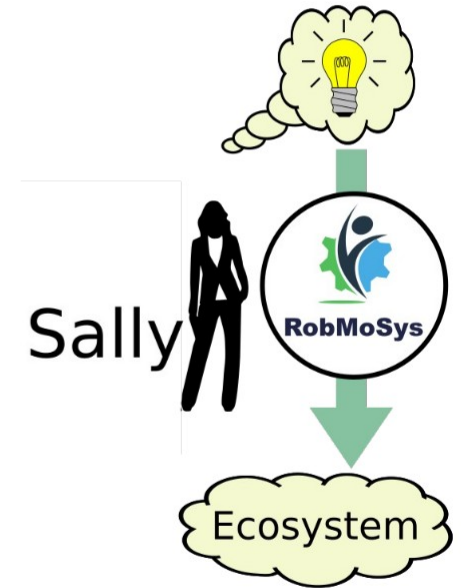
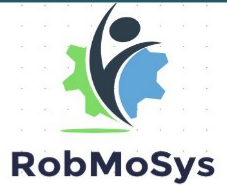
Max



„Models for Composition in an Ecosystem with Separation of Roles“, European Robotics Forum (ERF), Edinburgh, March 2017.

Ecosystem Participant "Sally"

- Imagine you have **developed software** to localize a robot in the environment and you are interested in **making it available** in robotics.
- You are
 - a **SME**, specialized in a certain domain
 - e.g. a **component supplier** for robot navigation
- You want to
 - express your offer with pivotal features such that others can find your component (yellow pages)
 - ensure that others can use your component (composability+compositionality)
 - explicate non-functional properties of your component and define its variation points

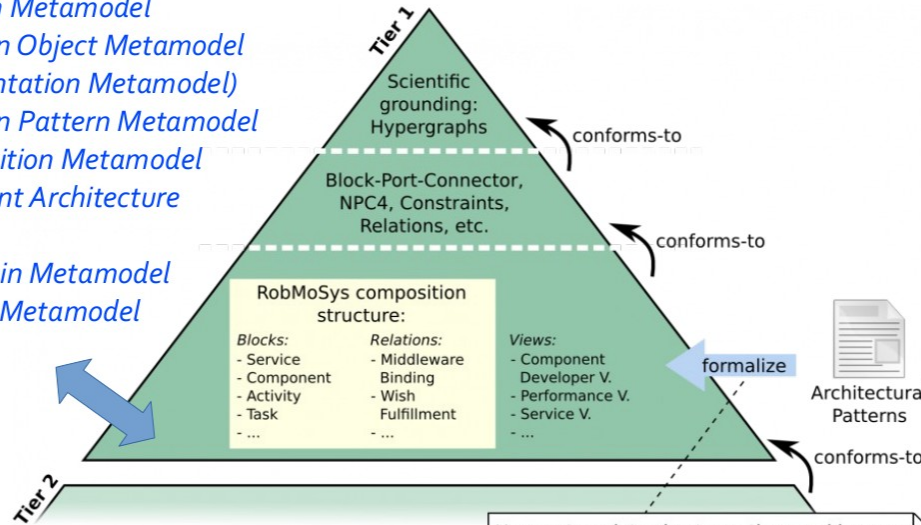


„Models for Composition in an Ecosystem with Separation of Roles“, European Robotics Forum (ERF), Edinburgh, March 2017.

Tier 1 Modeling Foundations

Tier 1 provides the **general structures for composition**. Three levels can be distinguished:

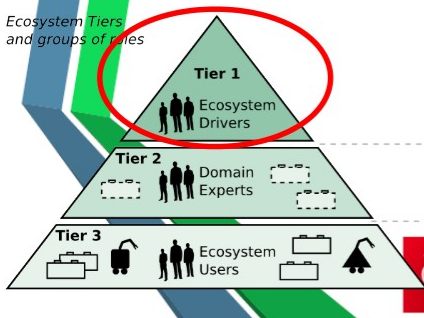
- *Service-Definition Metamodel*
- *Communication Object Metamodel (Data Representation Metamodel)*
- *Communication Pattern Metamodel*
- *Component Definition Metamodel*
- *System Component Architecture Metamodel*
- *Cause-Effect-Chain Metamodel*
- *Robotic Behavior Metamodel*
- ...



Architectural Patterns:

- levels / concerns
- composability
- separation of roles

		Concerns			
		Communication	Communication	Contribution	Configuration
Mission					
Task					
Skill					
Service					
Function					
Execution Container		Process Resources	Process Resources	Process Resources	Process Resources
Operating System / Middleware		Process Resources	Process Resources	Process Resources	Process Resources
Hardware		Process Resources	Process Resources	Process Resources	Process Resources



Human translates best practices and lessons learned as described in architectural patterns into formal models using the RobMoSys Block-Port-Connector meta-models to result in the RobMoSys composition-structure.



„Modeling Principles and Modeling Foundations“, RobMoSys Brokerage Day, Leuven, July 2017.

Benefits of using SmartSoft



Better Development

- separation of roles
- modeling
- workflow
- collaborate by structure rather than management

Better Systems

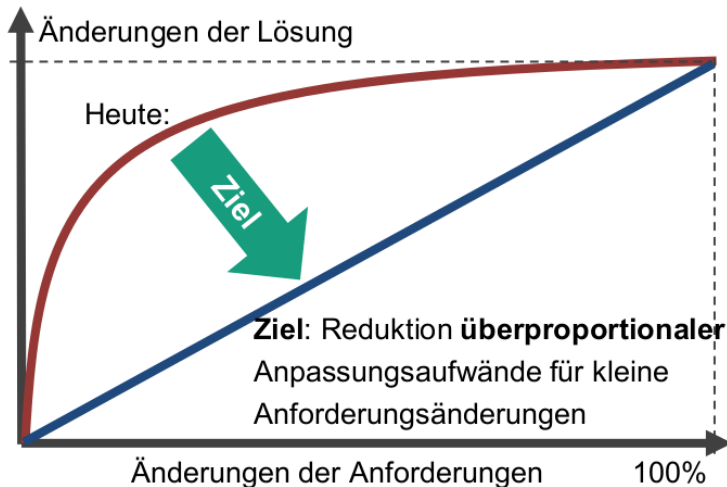
Ensured system properties and managed interfaces: roles and components can rely on them; enables analysis.

Flexibility and Efficiency

- composition of building blocks
- complexity reduction
- consistency



In BMBF SeRoNet:



In EU H2020 RobMoSys:

<p>Reduction of development time</p>	<p>Reduced development costs</p>	<p>Commoditisation of base components</p>	
<p>Composable components</p>	<p>Replaceable components</p>	<p>Re-usable</p>	<p>Predictable properties</p>

Mehr Informationen



- <http://www.servicerobotik-ulm.de>
- <http://www.youtube.com/roboticsathsulm>
 - incl. Screencast-Tutorials for the SmartMDSD Toolchain
- <http://www.robmosys.eu/wiki>
 - SmartSoft contributes to and conforms to the structures of the RobMoSys approach.

