RobMoSys:
Better Models, Better Tools, Better Systems

Progress made so far…

RobMoSys Workshop
ERF, Tampere, March 14th, 2018

RobMoSys

CEA · Hochschule Ulm University of Applied Sciences · KU LEUVEN · TECHNISCHE UNIVERSITÄT MÜNCHEN · COMAU · PAL ROBOTICS · SIEMENS · EUnited Robotics | European Robotics Association · eclipse

RobMoSys Contribution to EU Digital Industrial Platform for Robotics
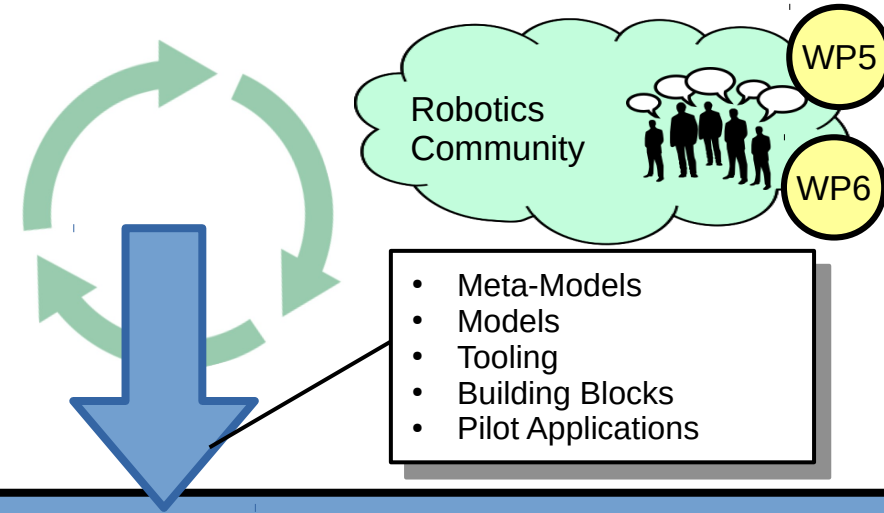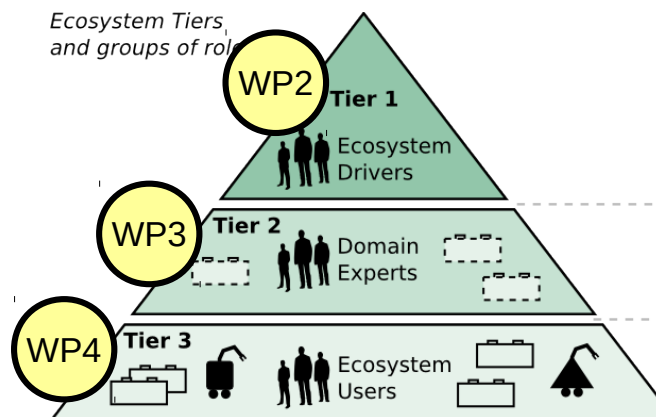
RobMoSys

RobMoSys is more than just another project...

- RobMoSys as community **moderator**

- **Join forces** to address the most challenging questions together

- Be **most inclusive**

- "Build a positive **atmosphere** and team spirit in the **community**"

- Release cycles and accessibility: **transparency has priority** over completeness (shown e.g. via the wiki)

RobMoSys

Links to Work Packages:
- WP2: Methodology, (Meta)Models, Tooling
- WP3: Basic Building Blocks
- WP4: Pilots
- WP5: Open Calls
- WP6: Dissemination and Community Building

Ecosystem Tiers and groups of roles

WP2 — **Tier 1** Ecosystem Drivers
WP3 — **Tier 2** Domain Experts
WP4 — **Tier 3** Ecosystem Users

Robotics Community — **WP5** **WP6**

- Meta-Models
- Models
- Tooling
- Building Blocks
- Pilot Applications

EU Digital Industrial Platform for Robotics

RobMoSys: Progress made so far...

- First round of successfully identifying, structuring, explicating body of knowledge and making it available via the RobMoSys Wiki

- First Round of successfully making existing Eclipse-based tooling conformant to RobMoSys and making it available

- First Open Call resulted in 6 funded ITPs

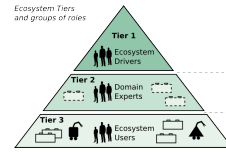- Preparing the Pilots (follow the Wiki updates to see how they evolve)

RobMoSys: Roles and Views

Structures

RobMoSys Composition Structures expressed in **meta-models:**

- Robotic Behavior Metamodel
- Communication-Object Metamodel
- Communication-Pattern M.-M.
- Component-Definition Metamodel
- Deployment Metamodel
- Functional Architecture Metamodel
- Cause-Effect-Chain …
- … and its Analysis Metamodels
- Service-Definition Metamodel
- System Component Architecture MM
- …

Views

Each role uses dedicated **views** to work on models, the modeling twin and the building block, e.g.

- Communication Pattern View
- Component Development View
- Service Design View
- System Configuration View
- Performance View
- Service Architecture View
- …
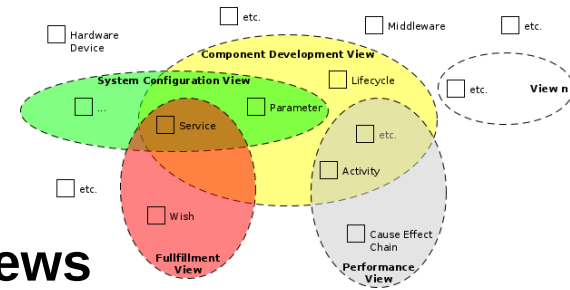
Participants

The participants in the ecosystem take one or several **roles** to use and supply building blocks:

- Behavior Developer
- Component Supplier
- Function Developer
- Performance Designer
- Safety Engineer
- Service Designer
- System Architect
- System Builder
- …

Tooling

… powered by SmartSoft … eclipse

Which patterns and structures form the *Sweet Spot* between *Freedom of Choice* and *Freedom from Choice*?

*Support as much freedom as possible while still ensuring **composability** despite **separation of roles***
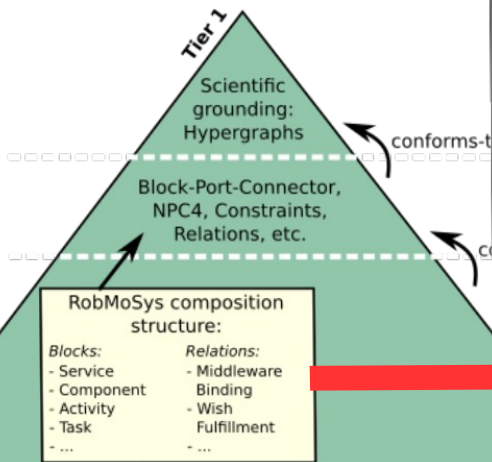
The RobMoSys Wiki: The number one source of information

- RobMoSys wiki is the number one source of information (accessible via the RobMoSys web page)

- Release cycles and accessibility: **transparency has priority** over completeness

- Evolving **live** RobMoSys Wiki vs. Frozen **Snapshots** for Call Applications

  - http://robmosys.eu/wiki/ vs.
  - http://robmosys.eu/wiki-sn-01/

- **Proposals** based on the 1st RobMoSys Wiki snapshot but we **encourage** you to trace the updates in the live RobMoSys Wiki for selected pages of your interest.

- RobMoSys **Core-Consortium** will **moderate, harmonize and integrate structures into the RobMoSys wiki**

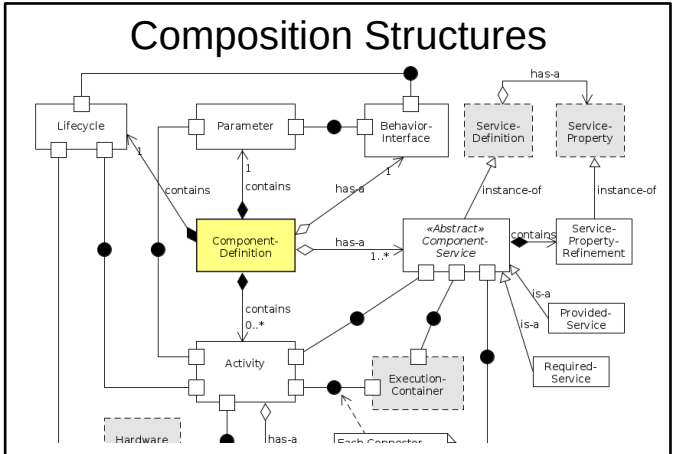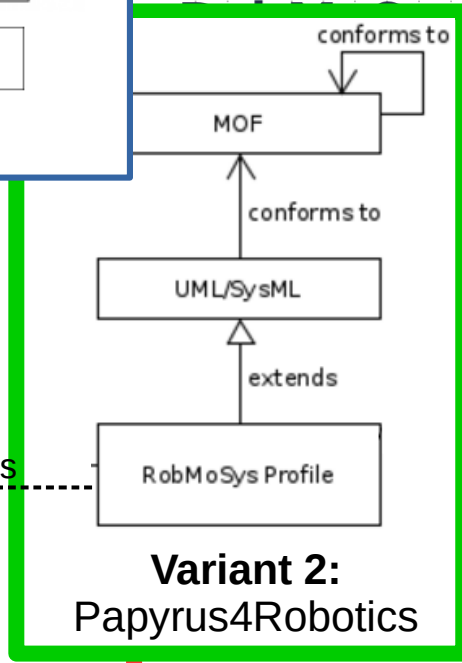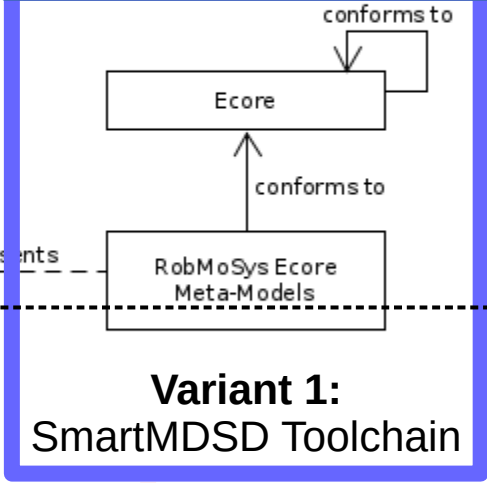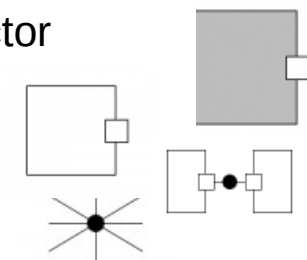- A **continuous** process starting just now with the ITPs

RobMoSys Tier 1 in Detail
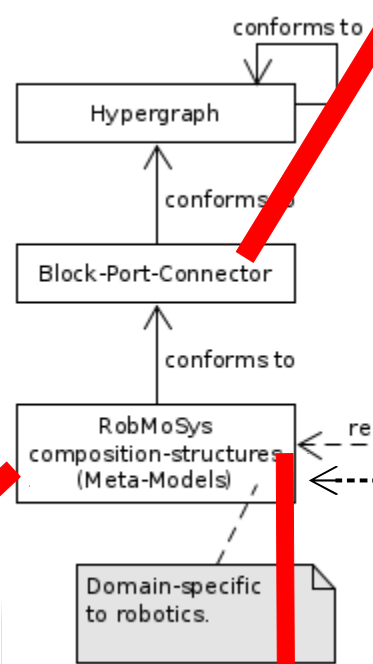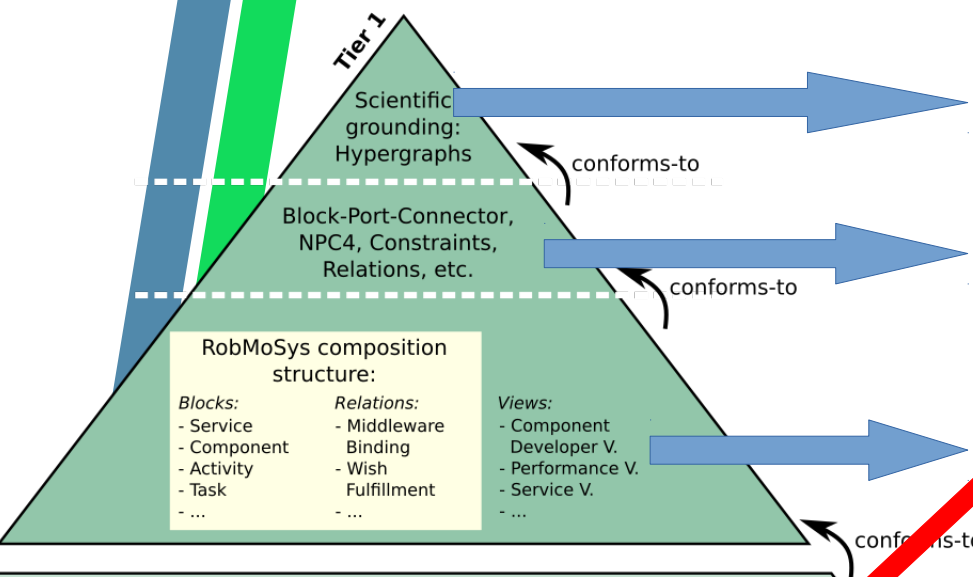
Block-Port-Connector (BPC) Model
- Block, port, connector
- Has-a
- Contains
- Connection
- Collection
- ...

Variant 1:
SmartMDSD Toolchain

Variant 2:
Papyrus4Robotics

Composition Structures

Role-Specific Views

Realization in RobMoSys Tooling
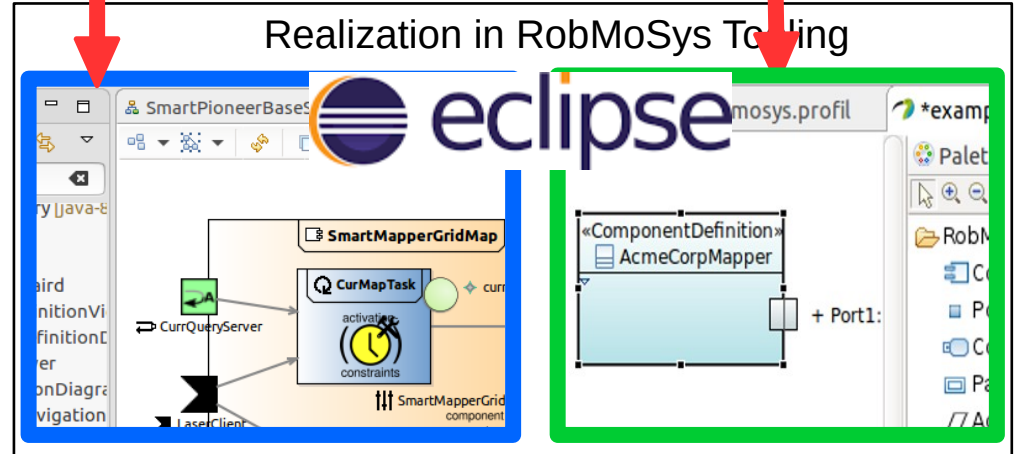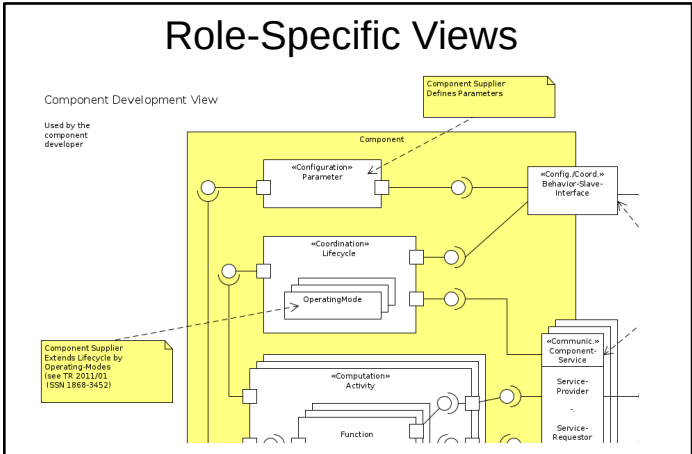
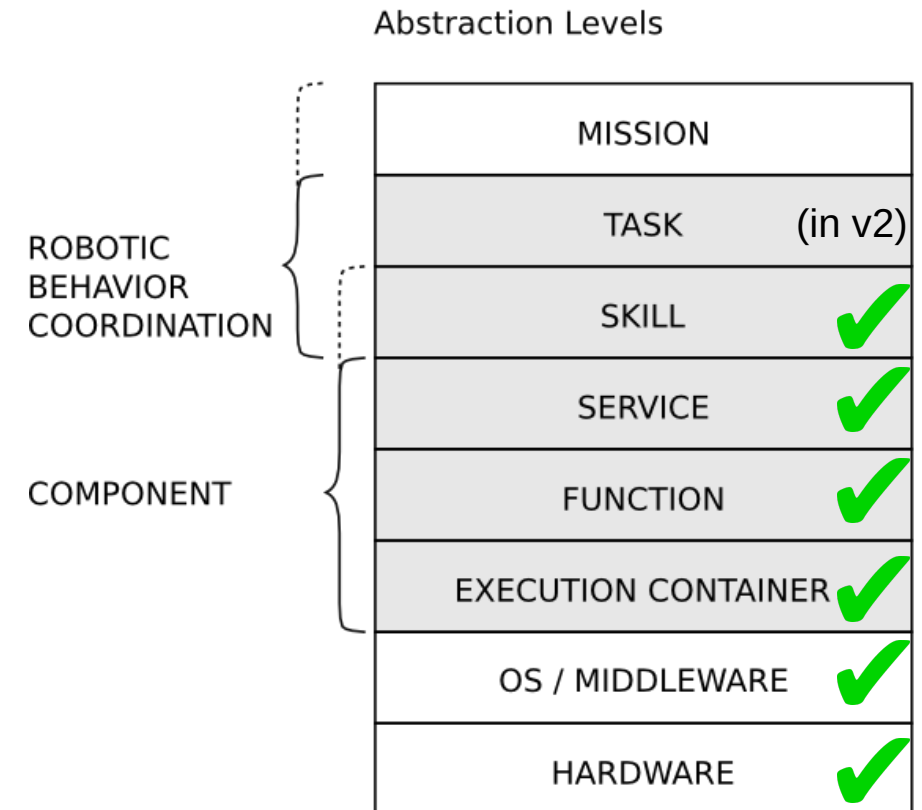# Tooling Variant 1: Focus and Strengths of the SmartMDSD Toolchain

- Very strong in modeling and developing concrete software components

- Very strong in code-generation resulting in executable software components

- Very strong in building **real systems**

- **Content** available for RobMoSys Tier 2 and Tier 3 available for download!

  - Domain Models, Software Components, Systems

- Also includes:

  - Analysis via SymTA/S

  - Simulation (Player/Stage, Morse, Gazebo)

Current State and Roadmap

RobMoSys

- Released March 2nd, Tooling ready to use, ongoing stability improvements

- Full support for vertical levels of abstraction complete

- Features and Meta-Models in transition
  - See next slide

- Content in transition to v3:
  - ~60 components in transition
  - Navigation stack already available
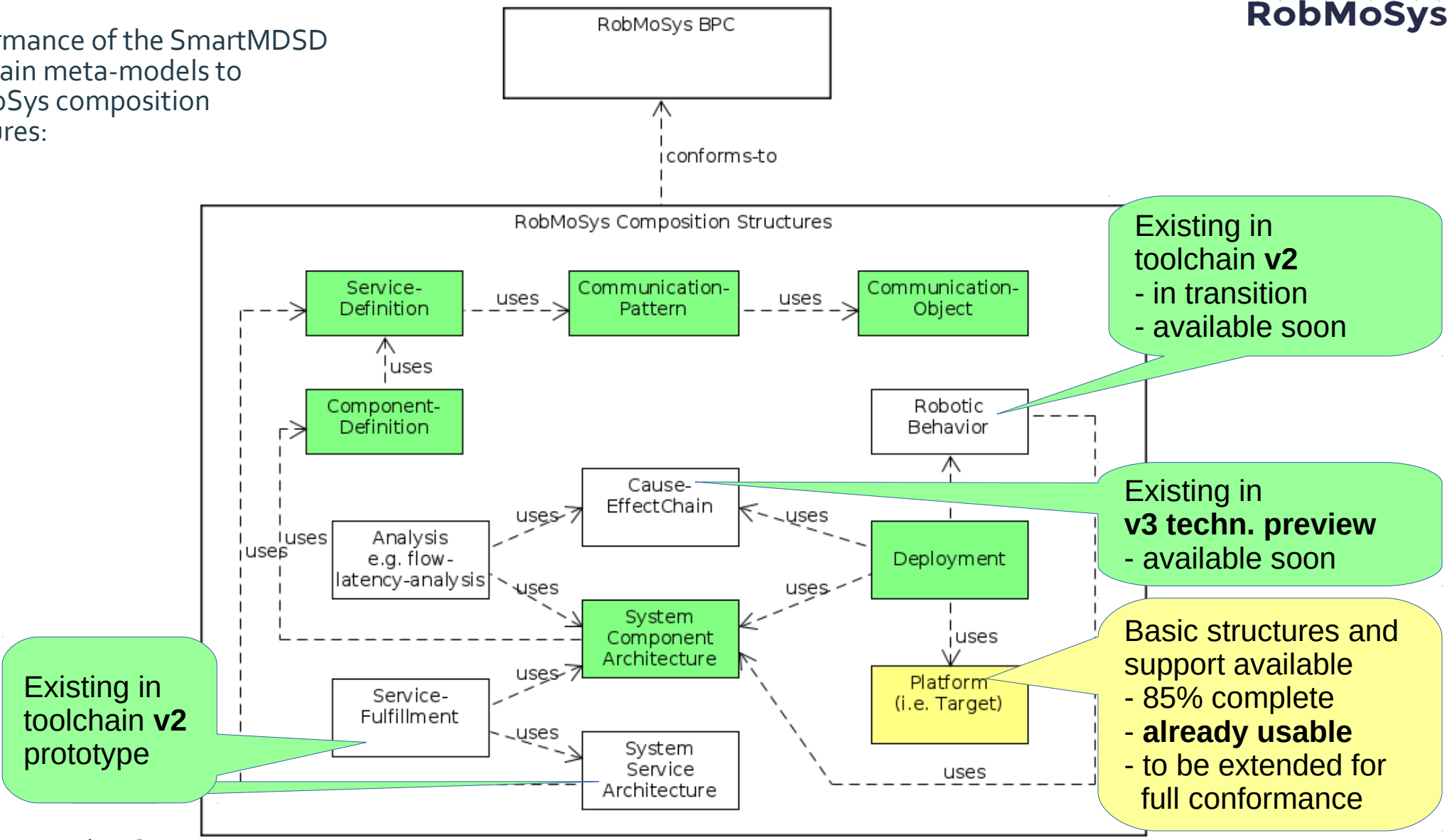 - TIAGo and Pioneer robot already available

Abstraction Levels

MISSION	
TASK	(in v2)
SKILL	✔
SERVICE	✔
FUNCTION	✔
EXECUTION CONTAINER	✔
OS / MIDDLEWARE	✔
HARDWARE	✔

ROBOTIC BEHAVIOR COORDINATION

COMPONENT

Conformance to RobMoSys

- Conformance of the SmartMDSD Toolchain meta-models to RobMoSys composition structures:

RobMoSys BPC

conforms-to

RobMoSys Composition Structures

Service-Definition — uses → Communication-Pattern — uses → Communication-Object

Existing in toolchain **v2**
- in transition
- available soon

Component-Definition (uses → Service-Definition)

Robotic Behavior

Cause-EffectChain

Analysis e.g. flow-latency-analysis

uses

Deployment

Existing in **v3 techn. preview**
- available soon

System Component Architecture

uses

Platform (i.e. Target)

uses

Basic structures and support available
- 85% complete
- **already usable**
- to be extended for full conformance

Service-Fulfillment

System Service Architecture
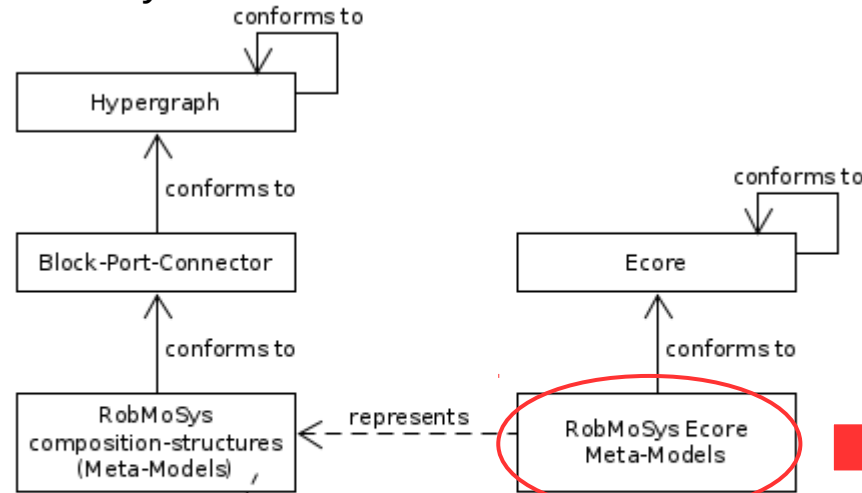
uses

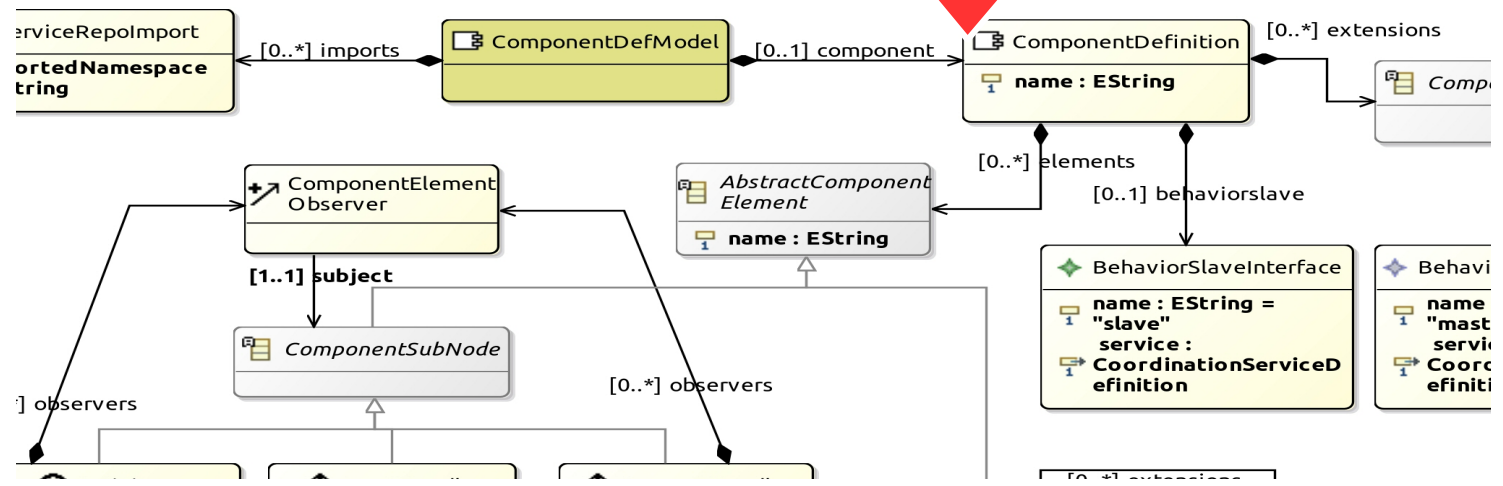Existing in toolchain **v2** prototype

RobMoSys

The SmartMDSD Toolchain implements the RobMoSys composition structures using Ecore. RobMoSys **structures become accessible** to Tier 2 and Tier 3 users!
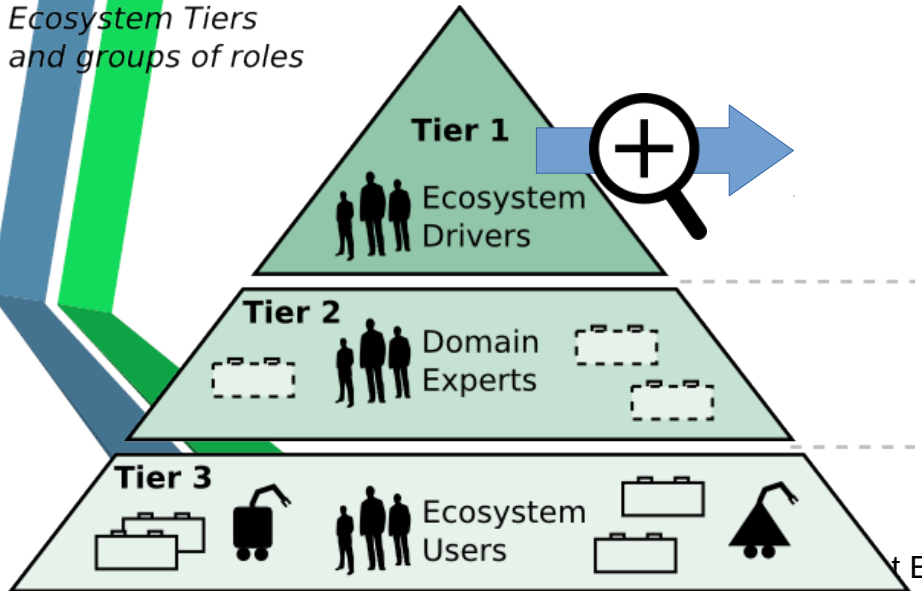
Example: Component Definition Meta-Model

RobMoSys Structures:

e.g. ecore meta-model for Component definition
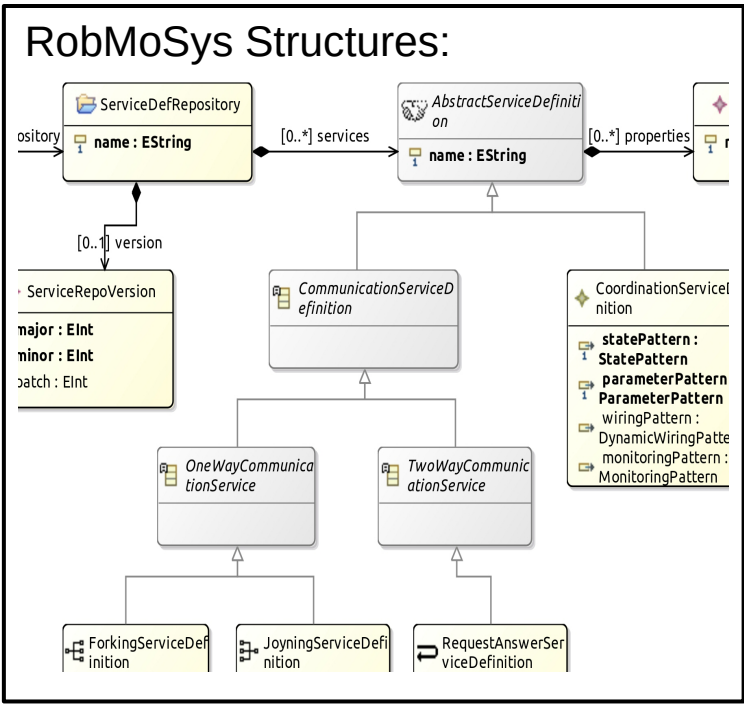
Ecosystem Tiers and groups of roles

RobMoSys

The SmartMDSD Toolchain supports in modeling domain structures (**domain models)** according to the RobMoSys composition structures.

Example: **Service Definitions**

RobMoSys Structures:

Role-Specific View:

```
NavigationObjects.services ⊠

        ActivationType=CommBasicObjects.CommBatteryPa
        EventType=CommBasicObjects.CommBatteryEvent
        EventStateType=CommBasicObjects.CommBatterySta
    >
}

/**
 * Planner service definitions
 */
ForkingServiceDefinition PlannerGoalService {
    PushPattern < DataType = CommNavigationObjects.Com
}

ForkingServiceDefinition PlannerEventService {
    EventPattern <
        ActivationType=CommNavigationObjects.CommPlann
        EventType=CommNavigationObjects.CommPlannerEve
        EventStateType=CommNavigationObjects.PlannerEv
    >
}

/**
 * Mapping service-definitions
```
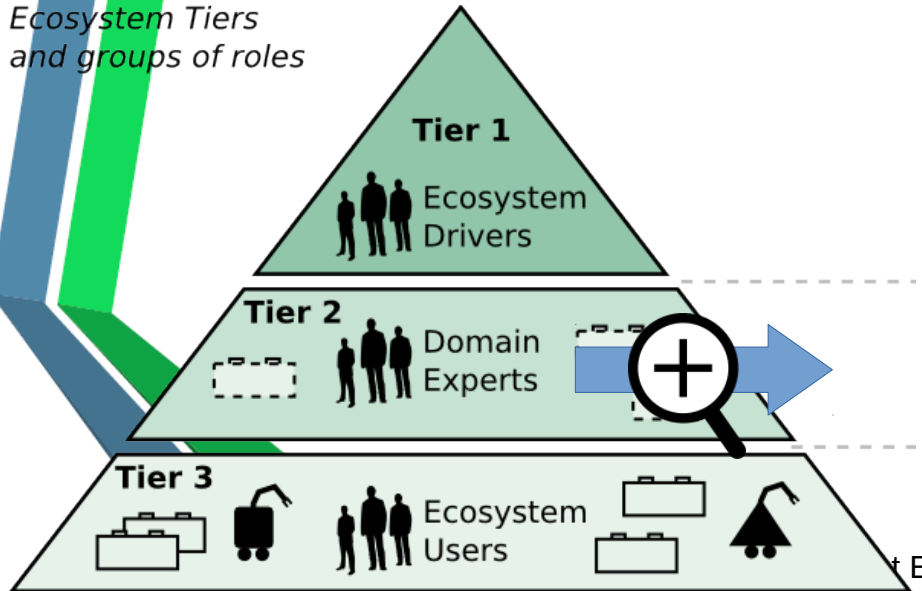
Available content: Domain Models
See https://github.com/Servicerobotics-Ulm/DomainModelsRepositories
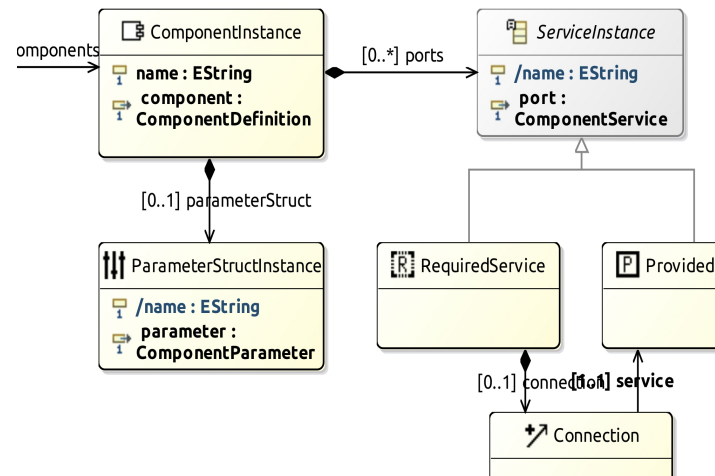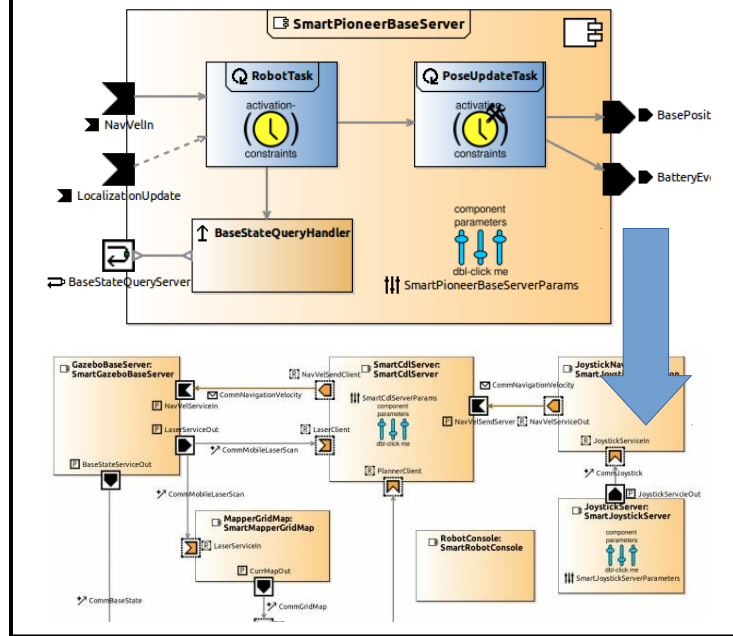

Ecosystem Tiers and groups of roles

RobMoSys

The SmartMDSD Toolchain supports in **developing components** and in **composing** previously developed components to **systems**.
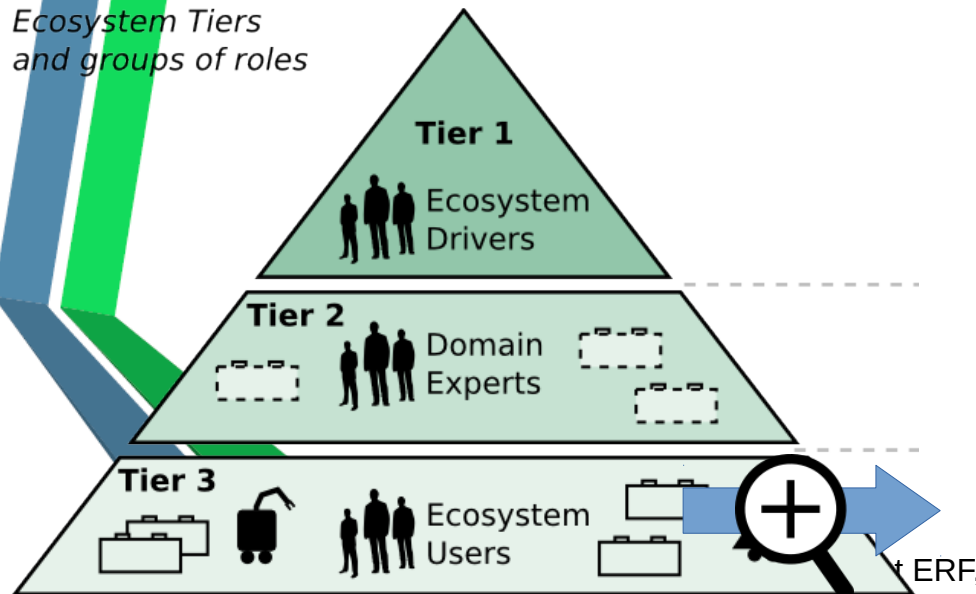
Example: TiaGO Navigation

RobMoSys Structures:
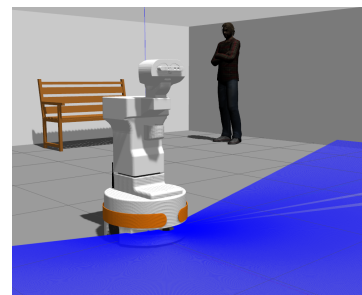e.g. Component Definition
and System Component Architecture

Role-Specific View:

Ecosystem Tiers and groups of roles

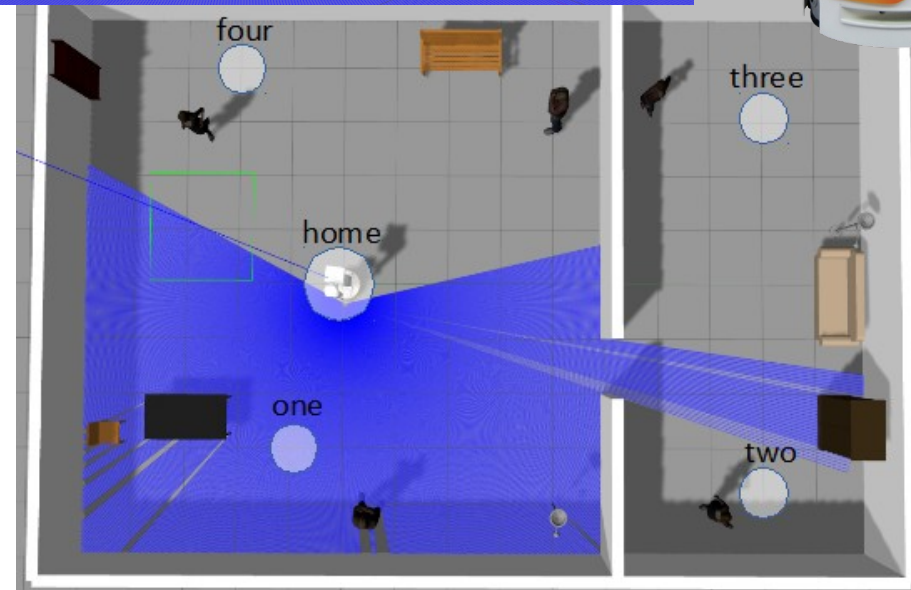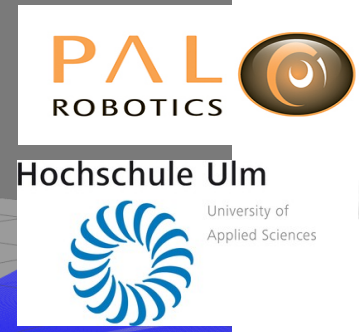Available content: previously developed/modeled building blocks:
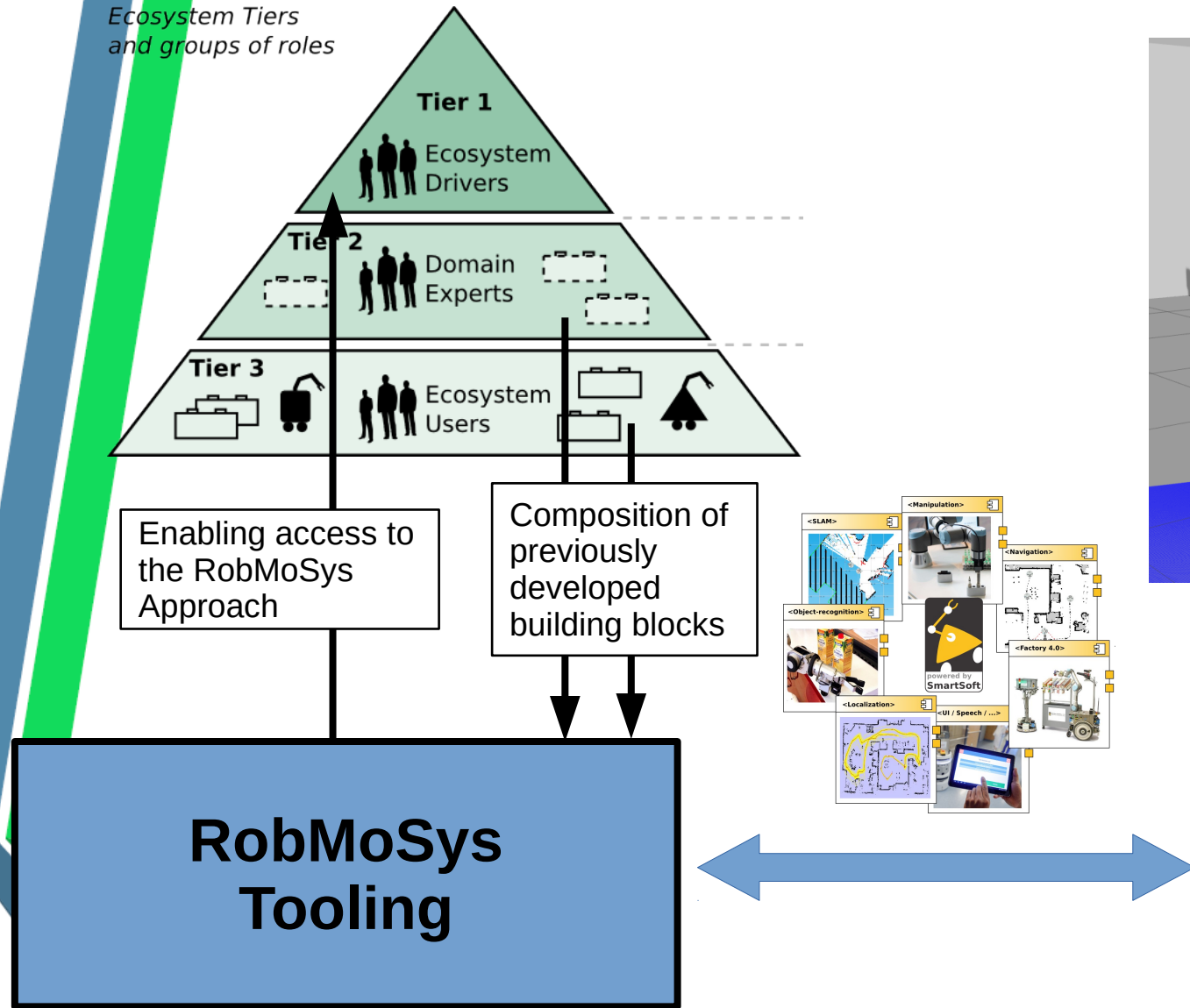See https://robmosys.eu/wiki/baseline:components:smartsoft

Enabling access to the RobMoSys Approach

Composition of previously developed building blocks
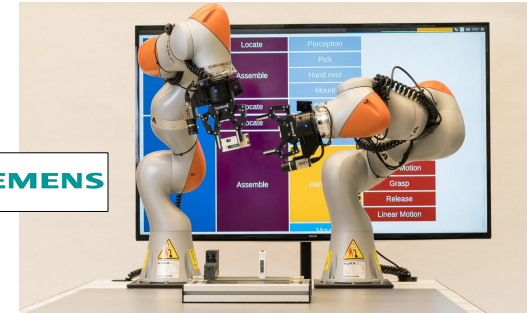
RobMoSys Tooling

Preparing the RobMoSys Pilots

Intralogistic
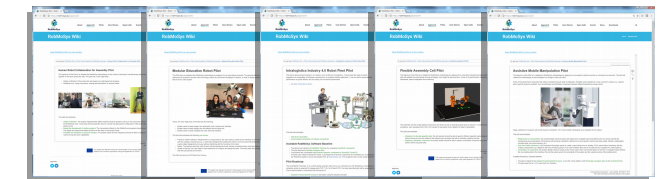Industry 4.0 Robot Fleet

Healthcare Assistive Robot

Flexible Assembly Cell

Modular Educational Robot

Human-Robot Collaboration
for Assembly

Broad description of all pilots on the
RobMoSys wiki