RobMoSys Technical Wiki Contents Overview

And Explanations about what we are looking for in the Open Call Proposals

2nd RobMoSys Brokerage Day
Frankfurt, August 24th, 2017

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732410.

RobMoSys Wiki – RobMoSys - Chromium

RobMoSys Wiki – Rc

robmosys.eu/approach/

RobMoSys

About | Approach | Pilots | User Stories | Open Calls | Events | News | Downloads

RobMoSys Wiki

Open RobMoSys Wiki in a new window

You are here: **RobMoSys Wiki**

start

RobMoSys Wiki

RobMoSys enables the composition of robotics applications with managed, assured, and maintained system-level properties via model-driven techniques. It establishes structures that enable the management of the interfaces between different robotics-related domains, different roles in the ecosystem, and different levels of abstractions. Documents that provide an overview and introduction:

- "Section 1 / Excellence": excerpt of RobMoSys Grant Agreement, Annex 1 (part B)
- Presentation of the RobMoSys project at European Robotics Forum 2017, Edinburgh
- Presentation "Modeling Principles and Modeling Foundations" at the RobMoSys Brokerage Day, July 5th 2017, Leuven

The **RobMoSys Wiki** provides technical details on the RobMoSys approach including examples realizing the RobMoSys structures. For general information about the RobMoSys project or its open calls, please refer to the project website.

Please note: The RobMoSys consortium is continuously updating this wiki to provide early insights. See the changelog. If you came here through a RobMoSys document, please see the jumppage to find referred pages.

Glossary and FAQ

The glossary contains descriptions of used terms. The technical FAQ provides answers to frequently asked questions.

Your Role in the RobMoSys Ecosystem

robmosys.eu/approach/

RobMoSys

Why is the RobMoSys Wiki of interest for you?

RobMoSys

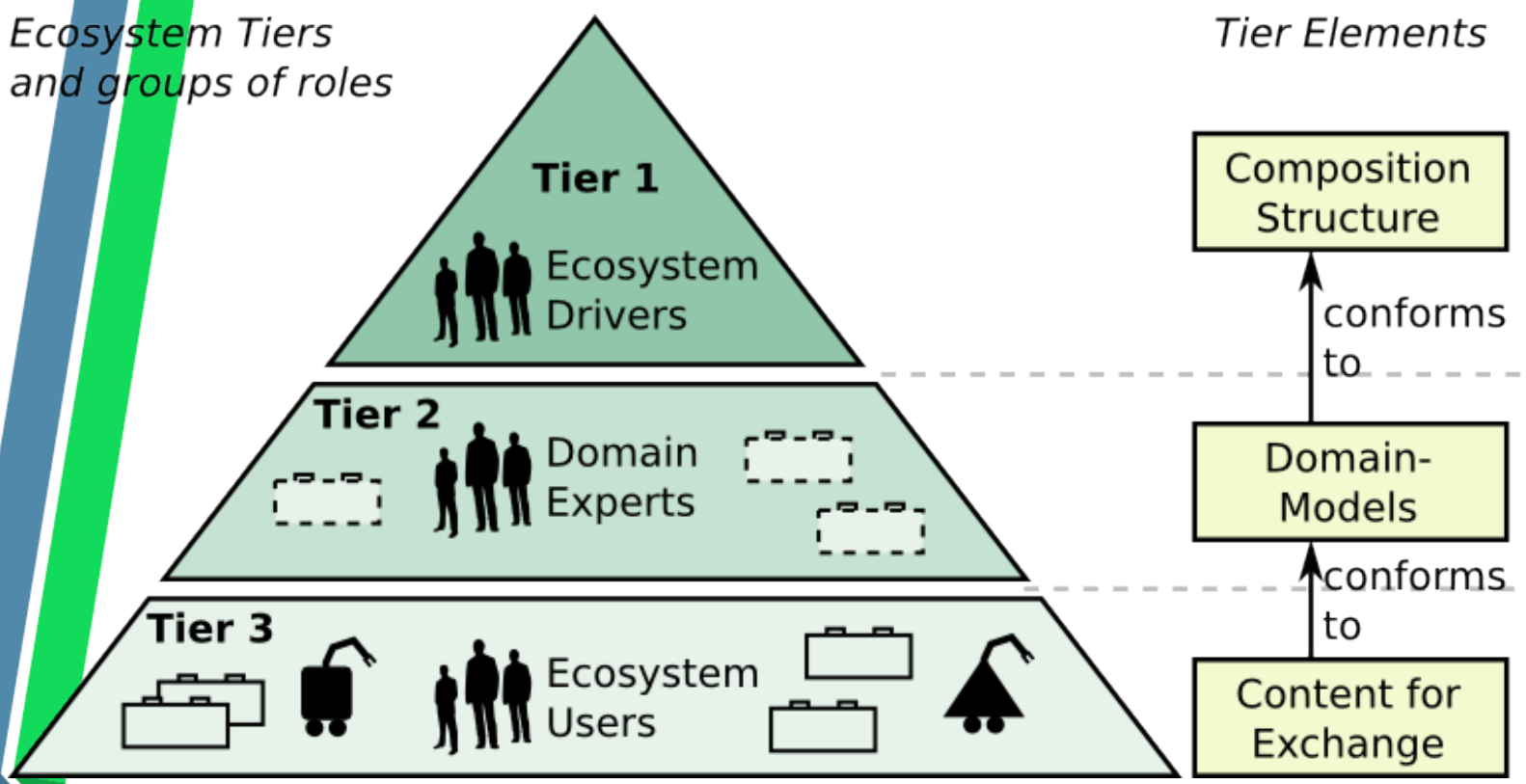
- In RobMoSys we do not start from scratch, instead we already have an **existing software and tooling baseline** to build onto and to extend it which serves as the technical background for **hot topics** in RobMoSys **to apply for**
- The possibility to experience new ways of **efficient engineering** of robotic systems by using structures and tools provided by RobMoSys
- The opportunity **to shape and to realize**, together with the RobMoSys consortium, the next generation **of engineering methods and tools** for robotics software development
- The opportunity to participate in a **structured robotics Ecosystem**
  - Interact with application-domain experts who seek for your component
  - Interact with robotics experts who can provide components that you need for your system

cea · Hochschule Ulm University of Applied Sciences · KU LEUVEN · TUM Technische Universität München · COMAU · PAL Robotics · SIEMENS · EUnited Robotics European Robotics Association · eclipse

Live Wiki and Wiki Snapshots

- Continuously evolving **live Wiki** (with most recent updates):
  - http://robmosys.eu/wiki/
- **1st Wiki Snapshot** for Open Call 1 Applicants (frozen contents):
  - http://robmosys.eu/wiki-sn-01/
- **2nd Wiki Snapshot** for Open Call 2 Applicants (frozen contents):
  - Scheduled at the end of March 2019
- You can write your **proposals** based on the Wiki snapshots but we encourage you to trace the updates in the live Wiki for selected pages of your interest (an updated change-log helps you easily finding new contents)
  - **We will consolidate your contributions and incorporate them into the live Wiki**, so you can directly profit from your own inputs

RobMoSys Ecosystem Organization



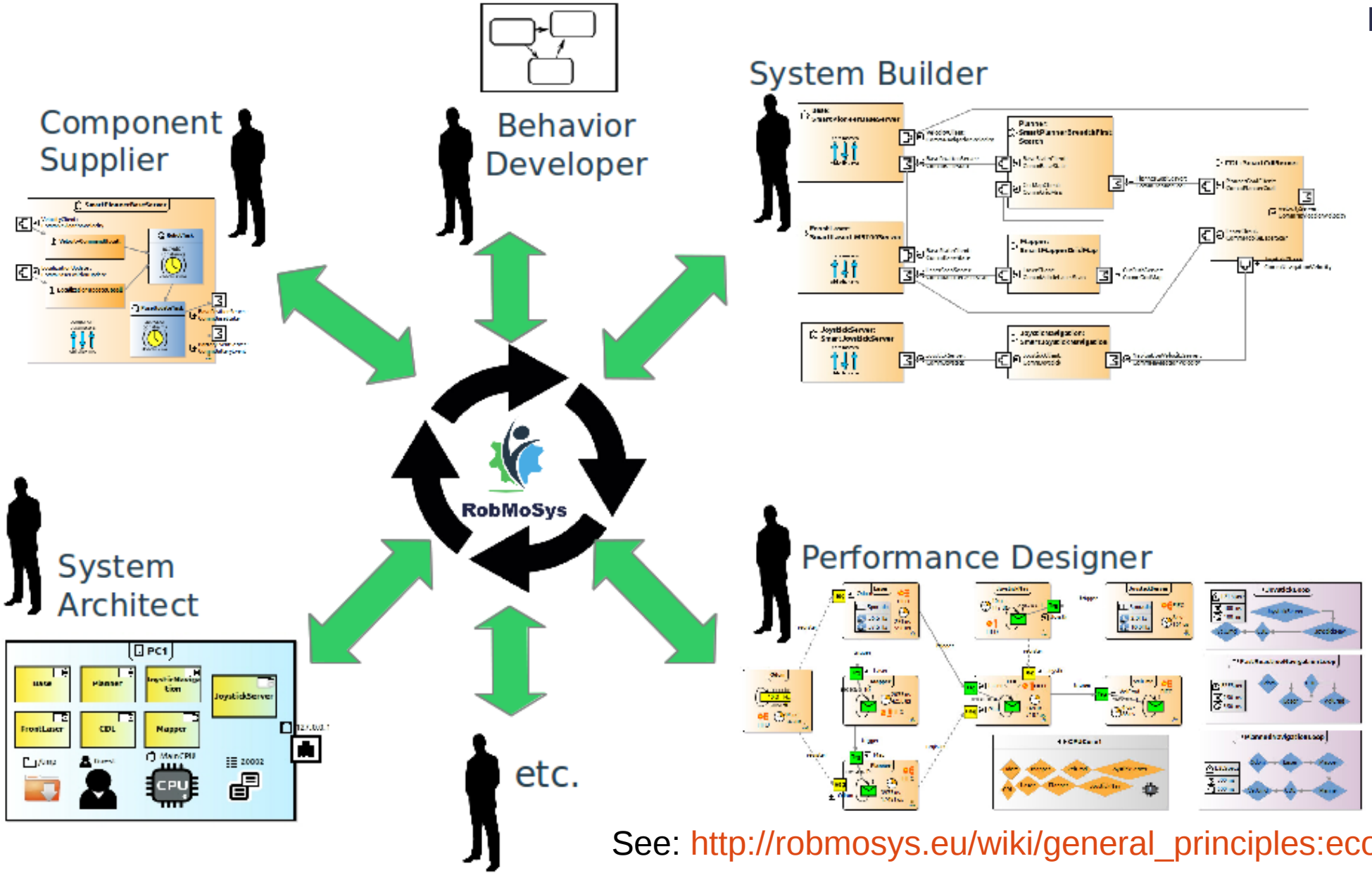
RobMoSys envisions a robotics **business ecosystem** in which a large number of loosely interconnected participants depend on each other for their mutual effectiveness and individual success. The **modeling foundation guidelines** and the meta-*-model structures are driven by the needs of the typical tiers of an ecosystem and the needs of their stakeholders. The different **tiers are arranged along levels of abstractions**.

See: http://robmosys.eu/wiki/general_principles:ecosystem:start

RobMoSys Ecosystem Organization

- **Tier 1:** structures the ecosystem in general for robotics. It is shaped by the drivers of the ecosystem that define an overall composition structure which enables composition and which the lower tiers conform to (similar to, for example, the ecosystem of the Debian GNU/Linux OS and its structures). Tier 1 is shaped by few representative experts for ecosystems and composition. This is kick-started by the RobMoSys project.
  - Structures defined on Tier 1 can be compared to structures that are defined for the PC industry. The personal computer market is based on stable interfaces that change only slowly but allow for parts changing rapidly since the way parts interact can last longer than the parts themselves and there is a huge amount of cooperating and competing players involved. This resulted in a tremendous offer of composable systems and components.
- **Tier 2:** conforms to these foundations, structuring the particular domains within robotics and is shaped by the experts of these domains, for example, object recognition, manipulation, or SLAM. Tier 2 is shaped by representatives of the individual sub-domains in robotics.
- **Tier 3** conforms to the domain-structures of Tier 2 to supply and to use content. Here are the main "users" of the ecosystem, for example component suppliers and system builders. The number of users and contributors is significantly larger than on the above tiers as everyone contributing or using a building block is located at this tier.

Your Role in the RobMoSys Ecosystem



See: http://robmosys.eu/wiki/general_principles:ecosystem:roles

Glossary and FAQ

Check your understanding of the terminology!

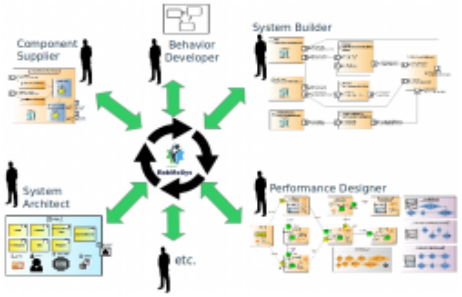
The glossary contains descriptions of used terms. The technical FAQ provides answers to frequently asked questions.

Your Role in the RobMoSys Ecosystem

Start reading here to see what your role is in the RobMoSys ecosystem or learn more about Roles in the Ecosystem. Main ecosystem users are:

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

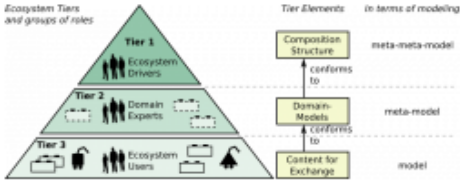
Participate as a role that best serves your needs



Besides the ecosystem participants, there are also other roles like the Model-Driven Engineering tool developers (see RobMoSys Composition Structures) and framework builders (see Software Baseline). Read a quick introduction to the role of open call applicants in the project-level FAQ.

General Principles

RobMoSys manages the interfaces between different roles and separates concerns in an efficient and systematic way by making the step change to a set of fully model-driven methods and tools for composition-oriented engineering of robotics systems. The following list of pages provide some fundamental principles in RobMoSys.



- Separation of Levels and Separation of Concerns
- Architectural Patterns
- Ecosystem Organization and Tiers
- User-Stories
- PC Analogy: Explaining RobMoSys by the example of the PC domain

Hot topics for open call proposals!

See: http://robmosys.eu/wiki

User-Stories

- Find the user-story in the Wiki that best fits your objectives
- Propose your approach to address these user-stories
- Make sure to build onto the RobMoSys software baseline
- Demonstrate that your approach makes a step-change for one (or some) of the user-stories

See: http://robmosys.eu/wiki/general_principles:user_stories

Composable commodities for robot navigation with traceable and assured properties

A very generic but extremely important user story illustrating the full scope of RobMoSys by a single example: Based on model-driven tools, develop and provide composable navigation components with all their explicated properties, variation points, resource requirements etc. (the modeling twin / data sheet). Become able to compose your navigation system out of these readily available commodity building blocks according to your needs and be sure that your needs are being matched, that the properties become traceable etc.

- I, as system builder, just want to become able to compose robotics navigation out of commodity building blocks according to my needs with predictable properties, assured matching with my requirements, free from interference. It is just astonishing that this is not yet possible in robotics. (with MoveBase being exactly an example of 1how it should not be)

Description of building blocks via model-based data sheets

RobMoSys achieves a specific level of quality and traceability in building blocks, their composition and the applications.

as a **component supplier**

- I want my component to become part of as many systems as possible to ensure return-of-investment for development costs and to make profit.
- I need to offer my software component (building block) such that others can easily decide whether it fits their needs and how they can use it.
- I want to offer my software component with a data sheet in form of a digital model (see xxx). A data sheet contains everything you need to know to become able to use that software component in a proper way (interface between the component and its environment) while protecting intellectual property. It contains information about the internals of the software component only as long as this is needed for a proper use.

as a **system builder**

- I want to select from available components the one which best fits my requirements and expectations (provided quality, required resources, offered configurability, price and licensing, etc.)
- I want to check via the data sheet (in form of a digital model) whether that building block with all its strings attached fits into my system given the constraints of my system and given the variation points of the building block. Thereto, I want to be able to import it into my system design to perform e.g. a what-if analysis etc.
- I want to extract from my system design the specification of a missing building block such that someone else can apply for providing a tailored software component according to my needs
- I want to use components as grey-box, use them “as-is” and only adjust them within the variation points expressed in the data-sheet without the need to examine or modify source code.

Replacement of component(s)

A hardware device is broken and the identical device is not available anymore (deprecated, discontinued, only next version available). As a system builder,

- I want to check whether all my relevant system level properties and constraints are matched when I use the new device.
- I also want to know how I need to configure it for that.

The very same holds true for software components where a software library used is not available anymore with updates of other libraries etc.:

- As a system builder, when I remove a software component from a system, I want to know which constraints define the now white spot in my design in order to fill in another one with the proper configuration to again match the system level properties.

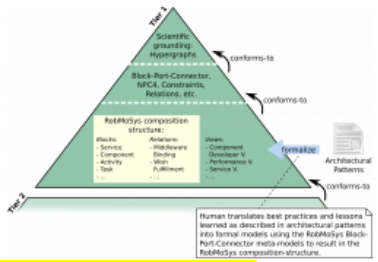
Wiki Main Contents

- For MDSE tool developers: Tier 1: “**Modeling Foundations**” is of particular interest to you
- For robotics domain experts (e.g. SLAM expert): Tier 2: “**Examples of Domain Models**” is where you can contribute
- For Ecosystem users (e.g. component developers, system builders, etc.): the “**RobMoSys Tools and Software Baseline**” is what you can use already now and here you can contribute with your components and systems

See:
http://robmosys.eu/wiki

Tier 1: Modeling Foundations

RobMoSys considers Model-Driven Engineering (MDE) as the main technology to realize the so far independent RobMoSys structures and to implement model-driven tooling. The wiki pages below collect some basic modeling principles related to realizing the RobMoSys structures.



- Roadmap of MetaModeling
- Modeling Principles
  - Modeling Twin
  - Realization Alternatives
- Tier 1 Structure
  - Scientific Grounding: Hypergraph and Entity-Relation model
  - Block-Port-Connector
  - RobMoSys Composition Structures
  - Views which are used by roles

RobMoSys Metamodels for tool developers

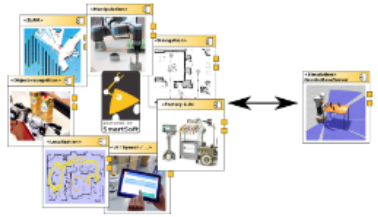
Tier 2: Examples of Domain Models

RobMoSys allows the definition of domain-specific models and structures at composition Tier 2. To illustrate this concept, RobMoSys defines the following extendable content for Tier 2.

- Introduction
- Motion, Perception, Worldmodel Stack
- Flexible Navigation Stack
- Active Object Recognition
- etc.

Tools and Software Baseline

RobMoSys provides a set of tools and a software baseline that already conform to the RobMoSys approach. This set can serve as a starting-point for implementations or demonstrations.



Tooling Baseline

- Roadmap of Tools and Software
- Development Environments and Tools
  - SmartSoft World
  - Papyrus for Robotics
  - to be extended

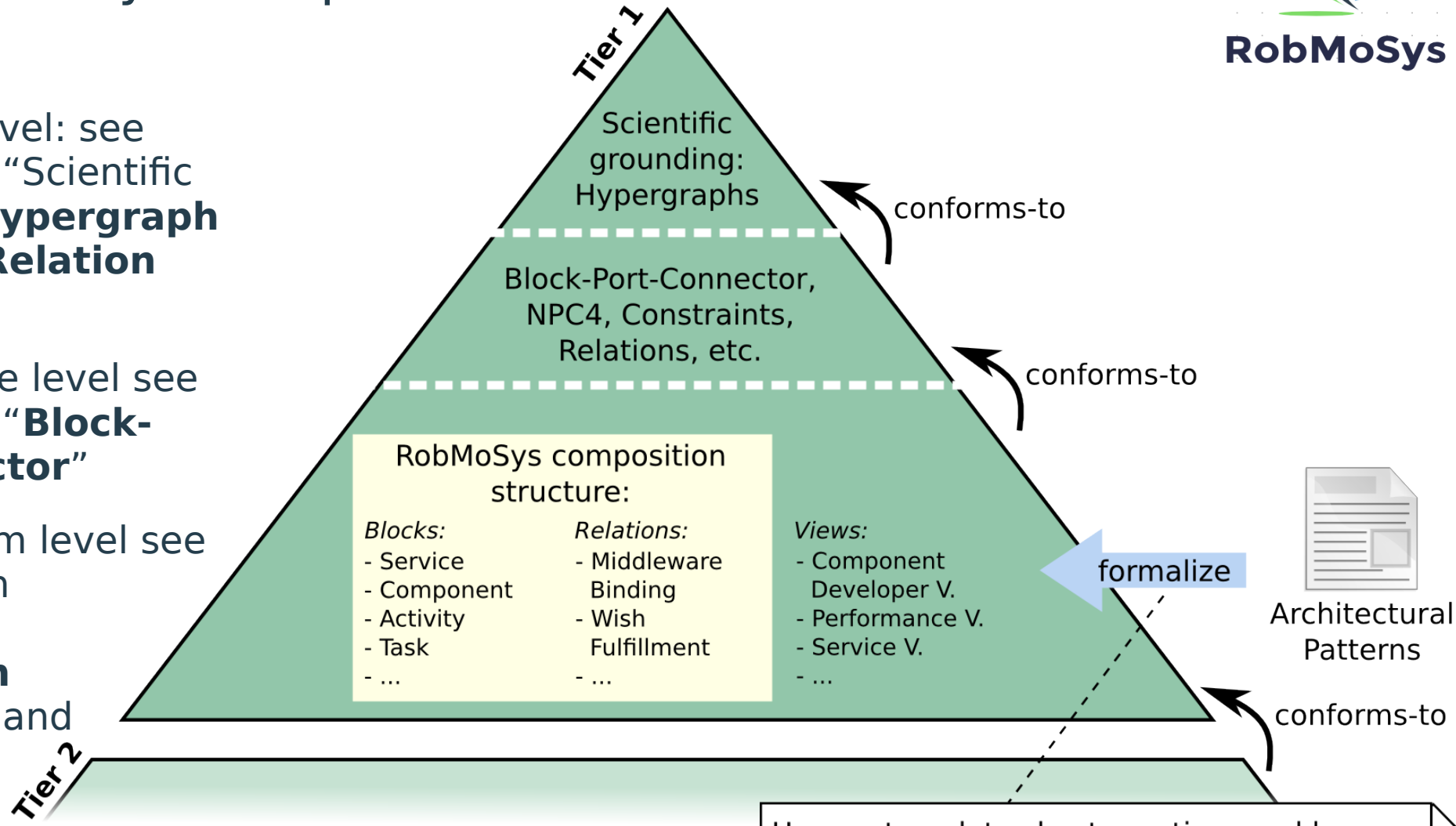
Tier 3: Existing Building Blocks and Scenarios

- Components
  - SmartSoft Components
- Scenarios and Systems
  - Gazebo/Tiago/SmartSoft Scenario

Ready to use Components and Systems for Ecosystem users

Tier 1: RobMoSys Composition Structures

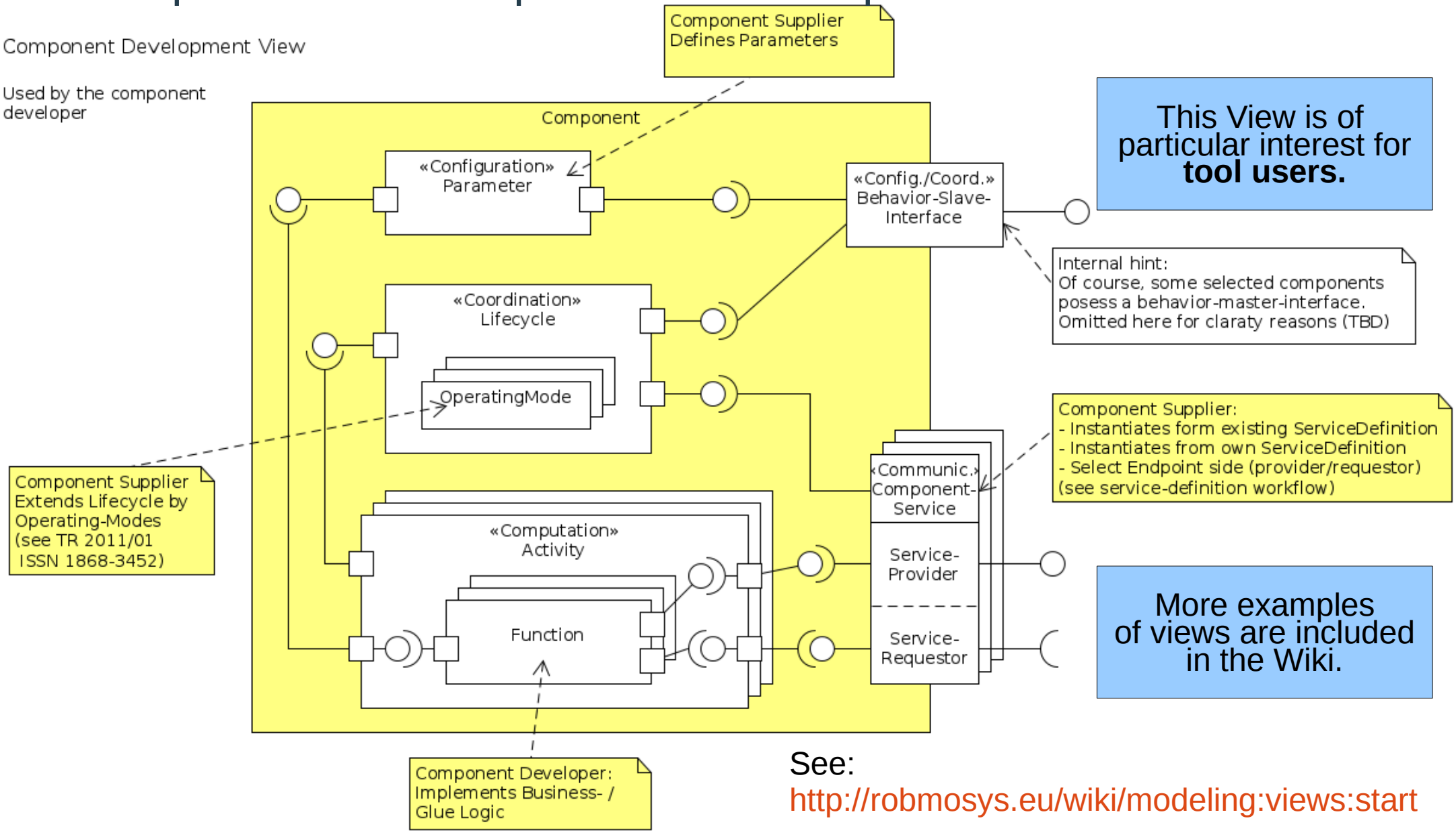
- For the top level: see Wiki page on “Scientific Grounding: **Hypergraph and Entity-Relation model**”
- For the middle level see Wiki page on “**Block-Port-Connector**”
- For the bottom level see Wiki pages on “**RobMoSys Composition Structures**” and on “**Views**”



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.

See: http://robmosys.eu/wiki/modeling:tier1

Tier 1: Structural Model of a Component: The Component Developer View Example

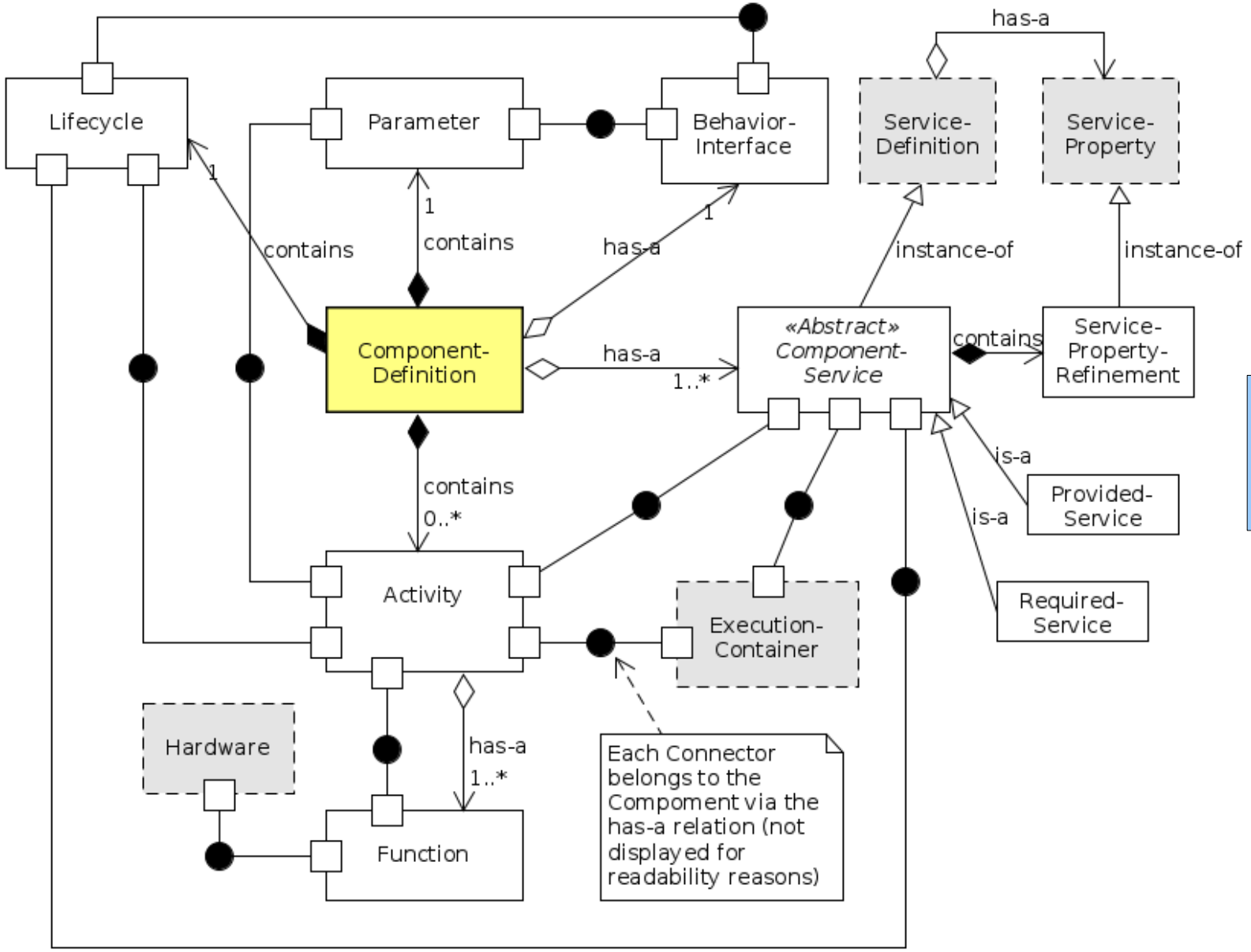


This View is of particular interest for **tool users.**

More examples of views are included in the Wiki.

See:
http://robmosys.eu/wiki/modeling:views:start

Tier 1: RobMoSys Composition Structures: The Component Metamodel Example



This **formal representation** is of particular interest for **tool developers.**

This Metamodel is based on the RobMoSys **Block-Port-Connector** specification (see Wiki)

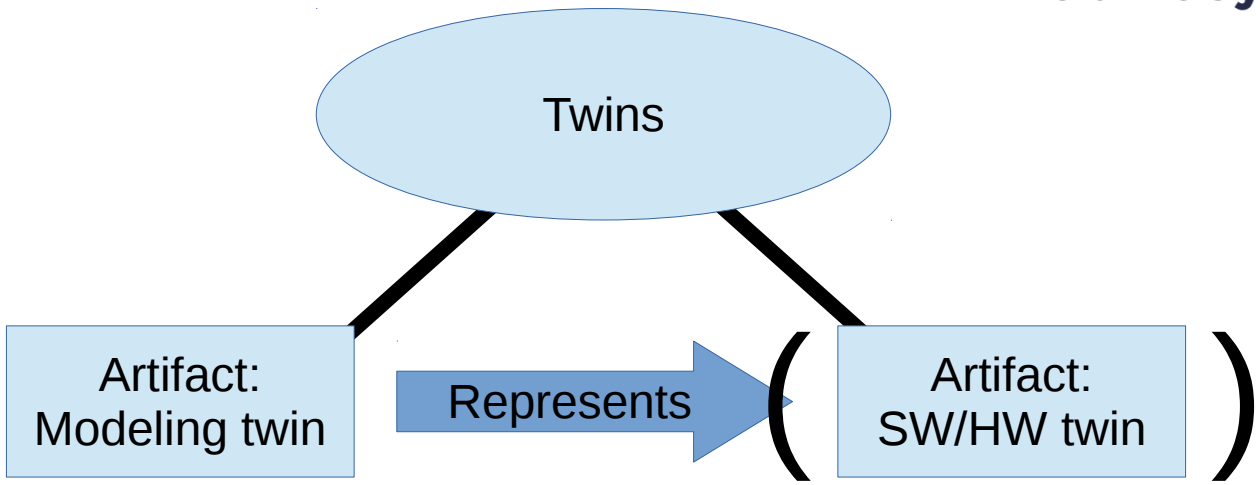
A full list of Meta-Models are included in the Wiki: http://robmosys.eu/wiki/modeling:composition-structures:start

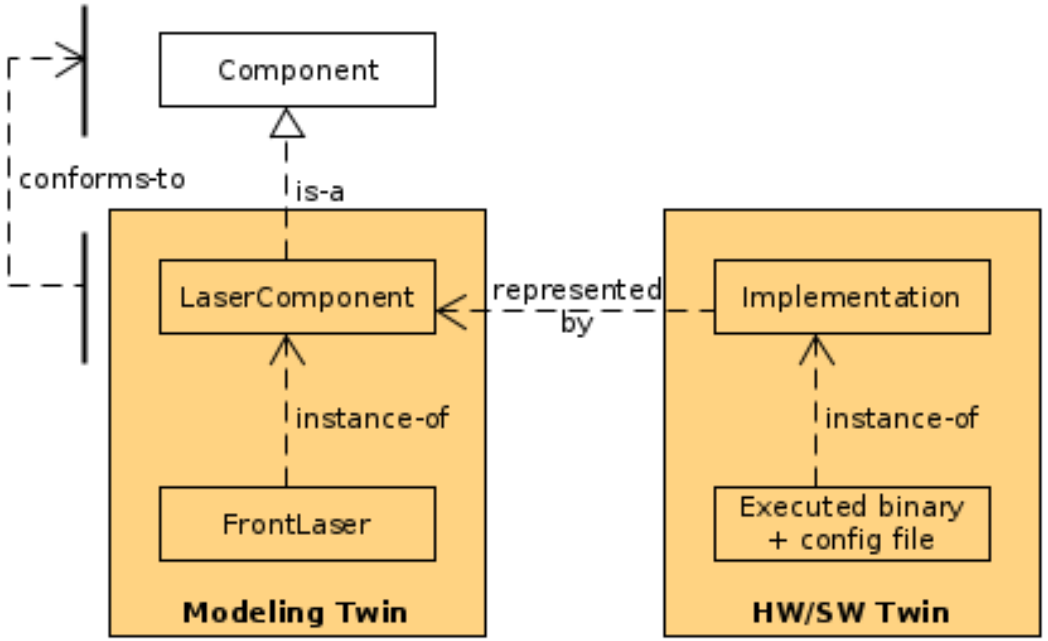
Modeling Twin

- All entities in the market come as twins
- Always supply and work with modeling twin.
- SW/HW twin might be supplied later (or might not exist at all if not needed)
- Entities in the market will never be just artifacts without a modeling twin as then the artifact cannot be used
- You can continue working independently with only the modeling twin

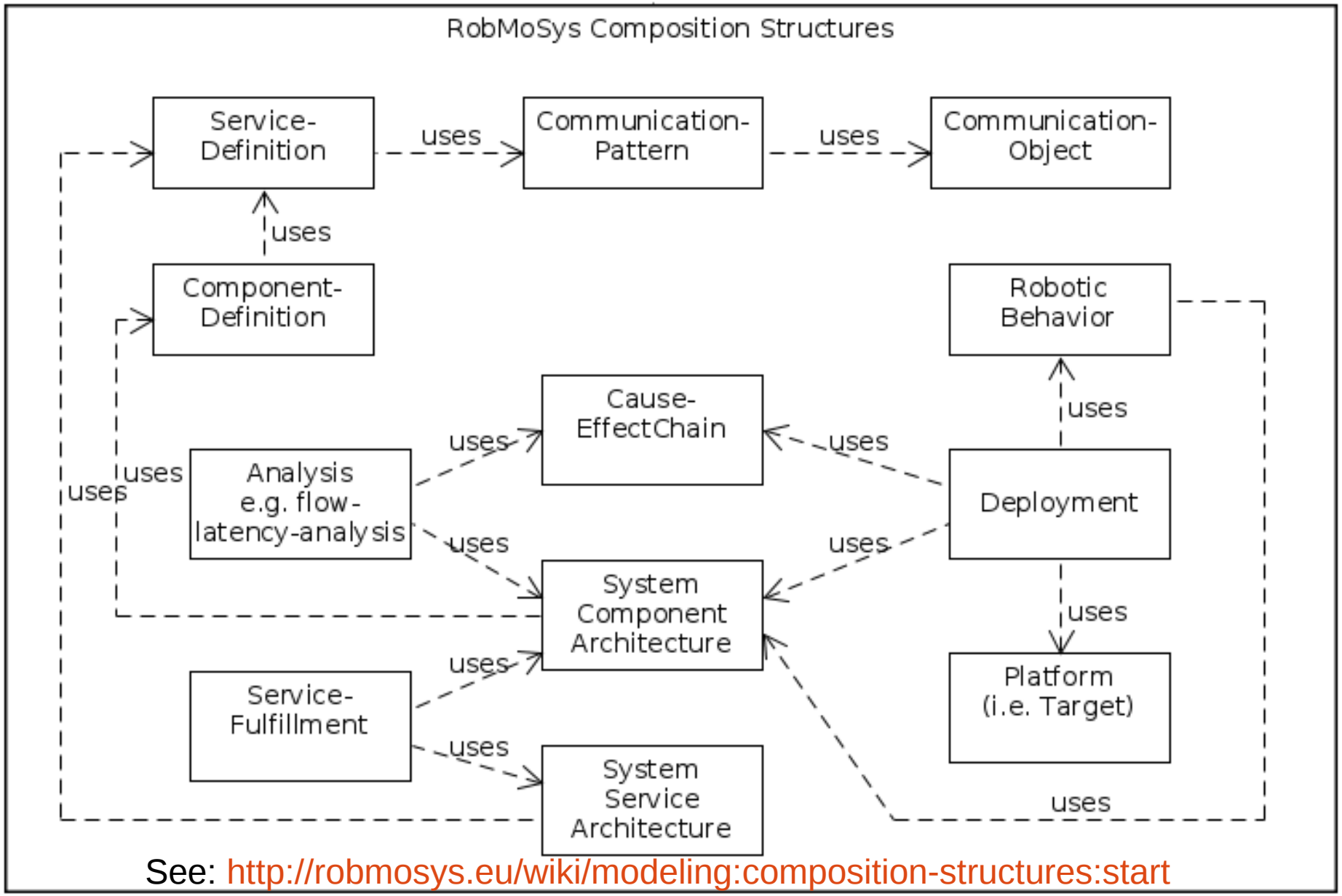
http://robmosys.eu/wiki/modeling:principles:modeling-twin

Modeling in RobMoSys:





Tier 1: RobMoSys Composition Structures

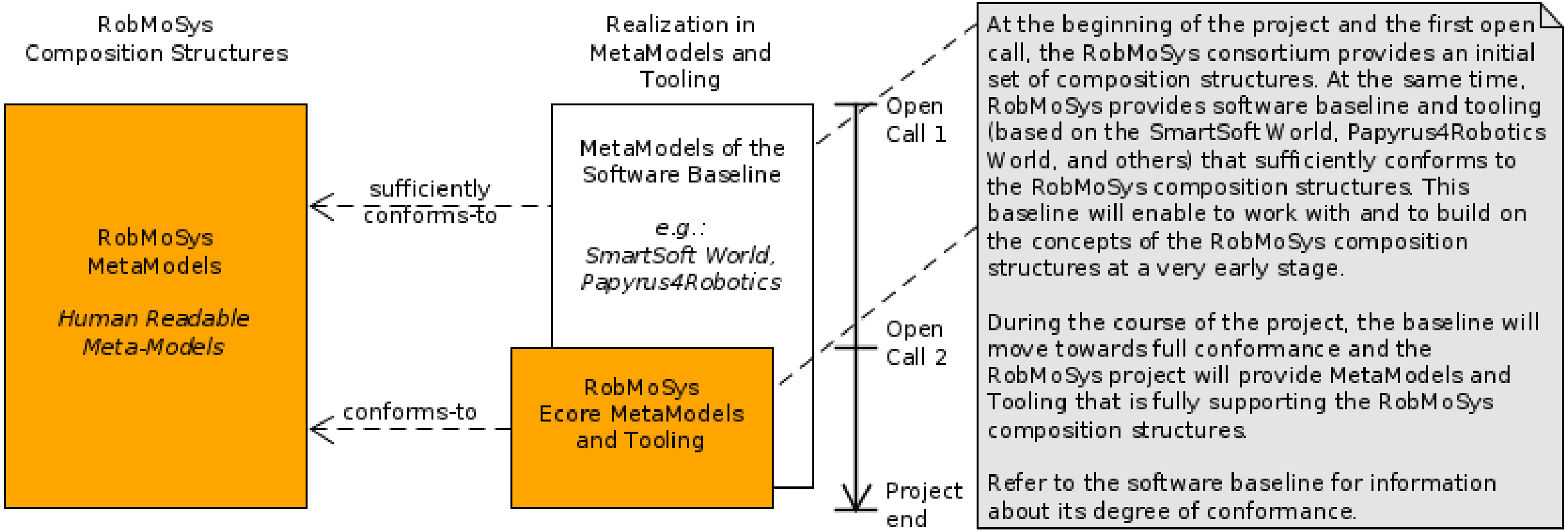


See: http://robmosys.eu/wiki/modeling:composition-structures:start

Each box represents a full meta-model

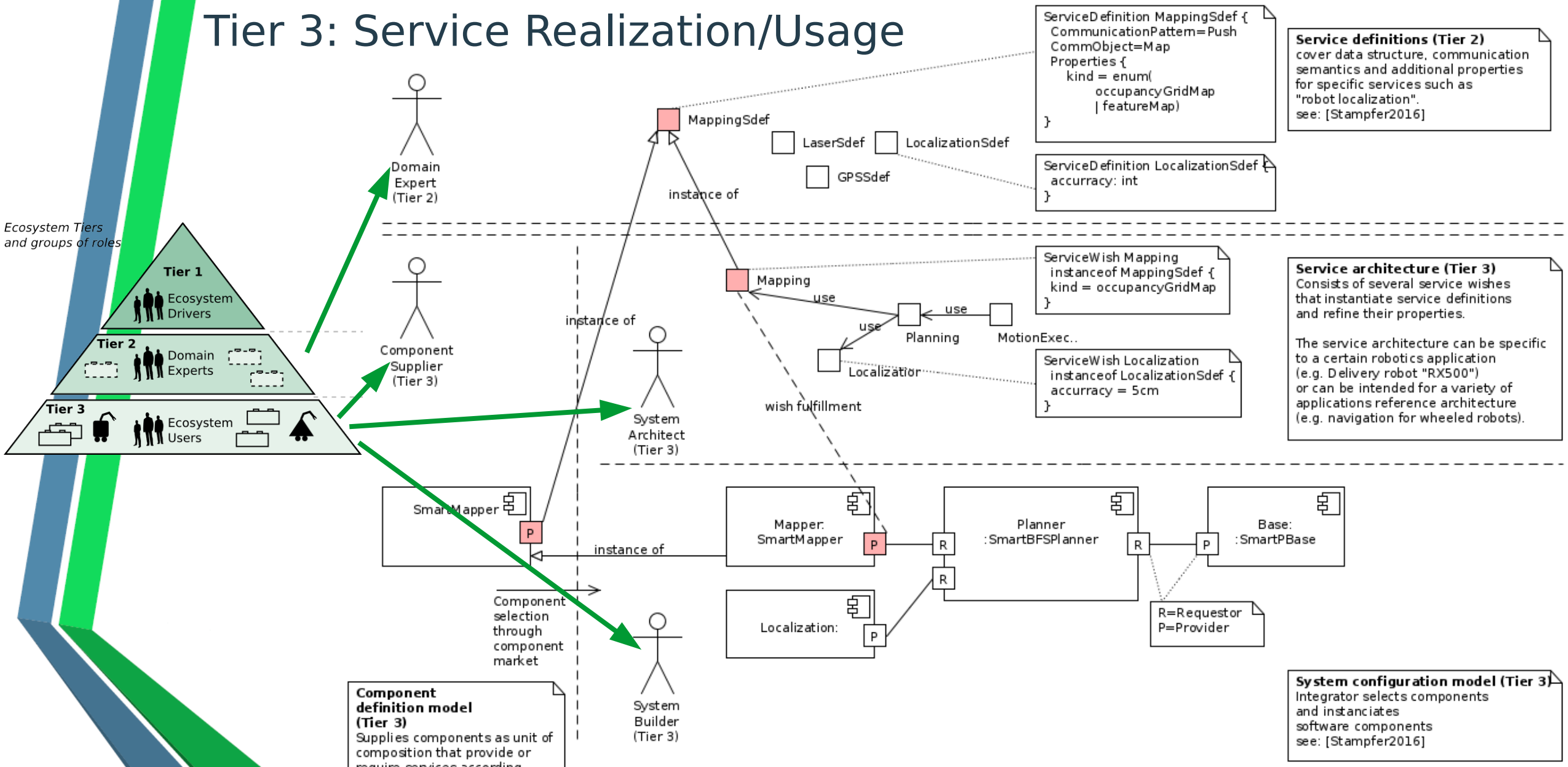
Meta-models have mutual dependencies

RobMoSys Software Baseline (tooling, components and applications)



See: http://robmosys.eu/wiki/baseline:roadmap

Tier 2: Service Definition and Tier 3: Service Realization/Usage



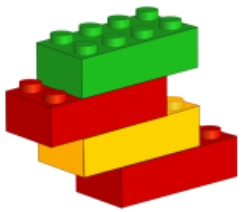
See: http://robmosys.eu/wiki/composition:service-based-composition:start

RobMoSys Pilots and Composition Basics

- RobMoSys uses **pilots** to demonstrate the use of its approach through the development of **full applications with robots**. Pilots span different domains and different kind of applications. The pilots can be **provided to project contributors** to support designing, developing, testing, benchmarking and demonstrating their contribution.
- RobMoSys is open for constructive suggestions from the community for **further pilots** or **extensions to existing pilots**, as long as "**platform**", "**composability**" and "**model-tool-code**" are first-class citizens of those suggestions.

Composition in an Ecosystem

RobMoSys adopts a composition-oriented approach to system integration that manages, maintains and assures system-level properties, while preserving modularity and independence of existing robotics platforms and code bases, yet can build on top of them.



- Introduction to Composition in an Ecosystem
- We illustrate composition, for example, using:
  - Task composition
  - Service-based composition of software components
  - Composition of algorithms

Pilots: Demonstrating the RobMoSys Approach

RobMoSys uses pilots to demonstrate the use of its approach through the development of full applications with robots. Pilots span different domains and different kind of applications. The pilots can be provided to project contributors to support designing, developing, testing, benchmarking and demonstrating their contribution.

- Goods Transport in a Company:
  - Intralogistics Industry 4.0 Robot Fleet Pilot
- Mobile Manipulation for manufacturing applications on a product line:
  - Flexible Assembly Cell Pilot
  - Human Robot Collaboration for Assembly Pilot
- Mobile manipulation for assistive robotics in a domestic environment or in care institutions:
  - Assistive Mobile Manipulation Pilot
- Modular Educational Robot Pilot

The project is open for constructive suggestions from the community for further pilots or extensions to existing pilots, as long as "platform", "composability" and "model-tool-code" are first-class citizens of those suggestions.

Other Approaches in the RobMoSys Context

RobMoSys follows a reuse-oriented approach. This means that reinvention should be kept to a minimum and existing approaches should be used wherever possible. The following list provides some common approaches that are considered relevant within the RobMoSys context.

- General Purpose Modeling Languages (SysML/UML) and Dynamic-Realtime-Embedded (DRE) domains (AADL, MARTE, etc.)
- Robotics Approaches (ROS, YARP, RTC, etc.)
- Middlewares (DDS)

See:
http://robmosys.eu/wiki

Explanations about what we are looking for in the Open Call Proposals

(see http://robmosys.eu/open-calls/ for further information)

http://robmosys.eu/open-calls/

RobMoSys

About | Approach | Pilots | User Stories | Open Calls | Events | News | Downloads

Open Calls

1st Open Call open!

Contribute to the RobMoSys ecosystem with great research funding opportunities

The first RobMoSys Open Call **opened on July 10th** and will be **closing on October 9th, 2017 (5 pm Brussels time)**. There will be approximately 6-7 projects selected to **start on March 1st 2018** with a runtime of 12 months.

– What is it about?

Project RobMoSys, co-funded from the European Union's Horizon 2020 research and innovation programme under agreement No 732410, foresees as an eligible activity the **provision of financial support to third parties**, as means to achieve its own objectives.

The vision of RobMoSys is to create **better models, as the basis for better tools and better software, which then allow to build better robotic systems**.

+ What are we looking for in the first RobMoSys Open Call?

+ Which types of activities qualify for financial support?

+ Come to our Brokerage Days

+ Download Call Documents

Apply here for the Open Call!

Contact: opencalls@robmosys.eu

For any further questions, please contact us via email.

Download Call documents

RobMoSys Funding Agreement Template
Download
396.05 KB 51 downloads

Proposal Template Word
Download
43.78 KB 66 downloads

Call text
Call Text
Download

What do we generally expect in the open call proposals?

RobMoSys

- **your proposal needs to be in line with the overall RobMoSys project setting and goals**. You subscribe to the “RobMoSys” methodology.
- the vision of RobMoSys is to create **better models**, as the basis for **better tools** and **better software**, which then allow to build **better robotic systems**. Thereby, the focus is on **composability** and on **separation of roles** via model-driven approaches.
- we ask for contributions that illustrate a **step change in system-level composition for robotics** in the context of RobMoSys and that demonstrate this **in real-world scenarios**.
- examples of what we want to achieve are given by **user stories** available in the RobMoSys Wiki. We prefer project proposals that are in line with the overall setting as laid out by the user stories. **Deriving your project ideas from there is highly recommended**.

General guidelines for open call proposals

RobMoSys

- contribute to **better models, better tools, better systems**
- the more **robotics centric** a proposal is, the better
- the more you address **links (i.e. interactions) between roles, views, levels, concerns,** etc., the better. It is not sufficient to deal with them in isolation.
- a contribution needs to come with a **clear motivation,** relevance, significance and potential **for the overall RobMoSys objectives**.
- we prefer **open-source** contributions and expect at least the **(meta-)models** and their transformations to be an open source license.
- the contribution needs to consistently fit into and **conform to the RobMoSys meta-models and models** (composability, conformance).

cea Hochschule Ulm University of Applied Sciences KU LEUVEN TUM Technische Universität München COMAU PAL Robotics SIEMENS EUnited Robotics European Robotics Association eclipse

What should be explicated in a proposal?

- fitting to the RobMoSys methodology:
  - **explain how your contribution fits into** (i.e. adheres to) and is composable with the **RobMoSys** modeling ideas
  - the more you deviate from the RobMoSys structures, the better a motivation needs to be why that is necessary (**motivate and explain the extraordinary benefit** that this could give)
- approach:
  - **describe and motivate the envisioned approach**, why it should now be considered within the RobMoSys setting, how the balance between risks and benefits for RobMoSys looks like
- results:
  - describe and **specify what you will deliver** and how you will deliver it

Build tandems and benefit from complementary expertise (here are some examples)

RobMoSys

- You are a **MDSE tool developer** and not experienced with robotics needs?
  - **team up with a robotics related project** in a tandem and provide a metamodel extension that fits within the RobMoSys baseline and that demonstrates the benefit (i.e. a step change) with respect to one (or several) of the user-stories
- You are a **robotics expert** and not experienced with using (or developing) model-driven tooling?
  - team up with a tooling expert (see above)
  - or at least **use the provided RobMoSys baseline** tooling to demonstrate that the RobMoSys structures helped you in specifying your components and/or systems and that **your components have been used by others** to build their systems or that you have been able to **use components from others to build your system**

Register to find potential partners at: http://robmosys.eu/online-brokerage-registration/

cea · Hochschule Ulm University of Applied Sciences · KU LEUVEN · TUM Technische Universität München · COMAU · PAL Robotics · SIEMENS · EUnited Robotics European Robotics Association · eclipse

Involvement with the RobMoSys consortium

- we require your commitment to participate in RobMoSys **inter-project workshops**
- in your proposal:
  - **underpin your background** with respect to the involved technologies in your proposal and the proposed approach
  - **motivate why you setup a tandem** or what the reason is for not going with a tandem
- **the RobMoSys core consortium will assist** you within the RobMoSys inter-project workshops in the learning curve in using / attaching to the RobMoSys software baseline and the RobMoSys baseline tools

What is RobMoSys *not* about?

RobMoSys

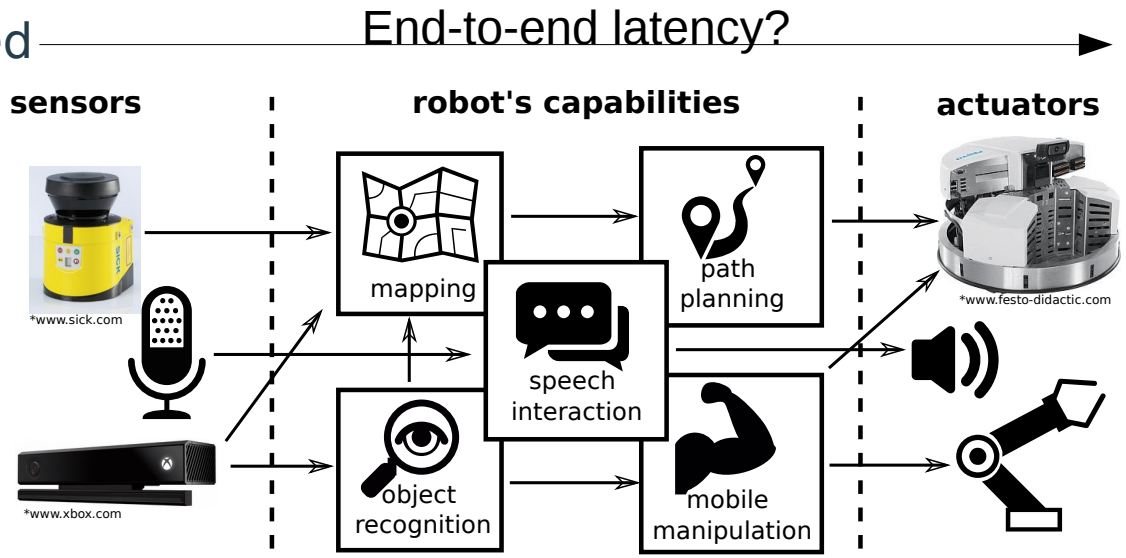
- *it is not just about Software Engineering, Model Driven Engineering or DSLs (domain specific languages) in itself*. Instead, **RobMoSys exploits** as far as possible **these means for the purpose of robotics**. Thus, there needs to be a clear focus on robotics needs. Reuse whatever already exists (e.g. from related domains) and reuse it whenever it fits for RobMoSys.
- *it is not just about realizing meta-models and models*. It is about **engineering models** which allow to build systems and which need to come with solvers. *It is also not about just using an equivalent technology and showing that you can do it with that technology as well* (avoid re-inventing the wheel). Instead, we favor **building on existing technologies** (e.g. make them accessible or extend them, in particular from related domains when available there). **Build transformations between worlds:** use a technology from another domain and make it accessible to robotics, e.g. use timing analysis (e.g. from automotive), introduce sanity checks at run-time, manage fail-operational modes, enable graceful degradation etc. **Always make clear the benefit for the RobMoSys objectives**.
- *it is not just about realizing show-cases* but about **realizing show-cases with the RobMoSys approach** and illustrating that doing them **in the RobMoSys way gives you advantages** which otherwise cannot be easily achieved. Such advantages are for example predictable properties of composed systems, knowledge about the impact of replacing components, traceability about properties, etc., (see the user stories for further details)

cea · Hochschule Ulm University of Applied Sciences · KU LEUVEN · TUM Technische Universität München · COMAU · PAL Robotics · SIEMENS · EUnited Robotics European Robotics Association · eclipse

An ideal setting of an open call proposal explained by an existing example (i)

RobMoSys

- Addressing the user-story on "**Management of Non-Functional Properties**"
  - I want to design and to predict **end-to-end latencies** in a system built of components
- Starting point:
  - **existing MDSE tools** allow the construction of component-based systems
  - an existing **navigation stack** is provided
  - an analysis methodology and tool exists (e.g. **SymTA/S**) from automotive domain
- A **tandem** with experts from the automotive domain is established
- This example is part of RobMoSys, see the Wiki Page for more details: http://robmosys.eu/wiki/modeling:metamodels:performance



cea · Hochschule Ulm University of Applied Sciences · KU LEUVEN · TUM Technische Universität München · COMAU · PAL Robotics · SIEMENS · EUnited Robotics European Robotics Association · eclipse

An ideal setting of an open call proposal explained by an existing example (ii)

RobMoSys

- Approach:
  - The component and system **meta-model have been extended** such that non-functional aspects related to end-to-end guarantees can be modeled
  - **Refined** responsibilities of **existing roles** (e.g. component supplier and system builder) and **introduced a new role** called “performance expert”
  - The existing **MDSE tool has been extended** to adhere to the extended meta-models and to support independent work of these roles
  - A new **model-to-model transformation step** from the system model to the **SymTA/S** representation allows the usage of the analysis results as feedback in the models to predict end-to-end latencies
  - The feasibility of the approach has been demonstrated in an **existing real-world scenario** by adapting the existing navigation stack
- As result: a system now can be designed out of existing components and **non-functional aspect can be infused into existing components afterwards** → this leads to **enhanced composability, predictablility and traceability**

cea Hochschule Ulm KU LEUVEN TUM COMAU PAL ROBOTICS SIEMENS EUnited Robotics European Robotics Association eclipse

Conclusions and Takeaway Messages

RobMoSys

- RobMoSys does not start from scratch, nor does it create models and DSLs in complete isolation. Instead, RobMoSys provides a solid **software baseline** that will be extended during the run-time of the RobMoSys project for:
  - Model-driven tooling (contribute to extending meta-models, improving workflows, etc.)
  - Software Components and Scenarios (e.g. contribute to managing QoS aspects)
- **RobMoSys wants your contributions** towards **better models**, as the basis for **better tools** and **better software**, which then allow to build **better robotic systems**.
- RobMoSys is open for constructive suggestions, as long as "**platform**", "**composability**" and "**model-tool-code**" are first-class citizens of those suggestions.
- Contributions can range from individual **software components and/or systems** (demonstrating a certain challenging user-story), up to extending and refining the **meta-models and model-driven tools** (thus improving the support for robotics component developers and system builders).
- The overall goal of RobMoSys is to achieve a "**EU Digital Industrial Platform for Robotics**" where robotic systems are built from composable models and software and where different experts collaborate to share risks and efforts.

cea Hochschule Ulm KU LEUVEN TUM COMAU PAL ROBOTICS SIEMENS EUnited Robotics European Robotics Association eclipse

RobMoSys

Thank you for your attention!
Any questions so far?

cea · Hochschule Ulm University of Applied Sciences · KU LEUVEN · TUM Technische Universität München · COMAU · PAL Robotics · SIEMENS · EUnited Robotics European Robotics Association · eclipse