# Software Architectures for Self-Aware Computing Systems

**Samuel Kounev**

Chair of Software Engineering
University of Würzburg

**http://se.informatik.uni-wuerzburg.de/**

Leipzig, 19.06.2017

# Selected References


- N. Huber, F. Brosig, S. Spinner, S. Kounev, and M. Bähr. **Model-Based Self-Aware Performance and Resource Management Using the Descartes Modeling Language**. *IEEE Transactions on Software Engineering (TSE)*, PP(99), 2017, IEEE Computer Society. To appear. [ pdf | DOI | http ]
- S. Kounev, N. Huber, F. Brosig, and X. Zhu. **A Model-Based Approach to Designing Self-Aware IT Systems and Infrastructures**. *IEEE Computer*, 49(7):53–61, July 2016, IEEE. [ pdf | DOI | http ]
- S. Kounev, F. Brosig, and N. Huber. **The Descartes Modeling Language**. Technical report, Department of Computer Science, University of Wuerzburg, October 2014. [ http | http | .pdf ]
- F. Brosig, N. Huber, and S. Kounev. **Architecture-Level Software Performance Abstractions for Online Performance Prediction**. *Elsevier Science of Computer Programming Journal (SciCo)*, Vol. 90, Part B:71-92, 2014, Elsevier. [ DOI | http | .pdf ]
- N. Huber, A. van Hoorn, A. Koziolek, F. Brosig, and S. Kounev. **Modeling Run-Time Adaptation at the System Architecture Level in Dynamic Service-Oriented Environments**. *Service Oriented Computing and Applications Journal (SOCA)*, 8(1):73-89, 2014, Springer-Verlag. [ DOI | .pdf ]
- F. Brosig, P. Meier, S. Becker, A. Koziolek, H. Koziolek, and S. Kounev. **Quantitative Evaluation of Model-Driven Performance Analysis and Simulation of Component-based Architectures**. *IEEE Transactions on Software Engineering (TSE)*, 41(2):157-175, February 2015, IEEE. [ DOI | http | .pdf ]
- F. Gorsler, F. Brosig, and S. Kounev. **Performance Queries for Architecture-Level Performance Models**. In *5th ACM/SPEC International Conference on Performance Engineering (ICPE 2014)*, Dublin, Ireland, 2014. ACM, New York, NY, USA. 2014. [ DOI | .pdf ]
- N. Herbst, N. Huber, S. Kounev and E. Amrehn. **Self-Adaptive Workload Classification and Forecasting for Proactive Resource Provisioning**. *Concurrency and Computation - Practice and Experience, John Wiley and Sons, Ltd.*, 26(12):2053-2078, 2014. [ DOI | http | .pdf ]
- S. Spinner, G. Casale, F. Brosig, and S. Kounev. **Evaluating Approaches to Resource Demand Estimation**. *Performance Evaluation*, 92:51 - 71, October 2015, Elsevier B.V. [ DOI | http | .pdf ]
- *N. Herbst, S. Kounev and R. Reussner.* ***Elasticity: What it is, and What it is Not.*** *In 10th Intl. Conference on Autonomic Computing (ICAC 2013), San Jose, CA, June 24-28, 2013.* [ slides | http | .pdf ]
- A. Milenkoski, M. Vieira, S. Kounev, A. Avrtizer, and B. Payne. **Evaluating Computer Intrusion Detection Systems: A Survey of Common Practices**. *ACM Computing Surveys*, 48(1):12:1-12:41, September 2015, ACM, New York, NY, USA. **5-year Impact Factor (2014): 5.949**. [ http ]
- A. Milenkoski, K. R. Jayaram, N. Antunes, M. Vieira, and S. Kounev. Quantifying the Attack Detection Accuracy of Intrusion Detection Systems in Virtualized Environments. In Proceedings of The 27th IEEE International Symposium on Software Reliability Engineering (ISSRE 2016), Ottawa, Canada, October 2016. IEEE, IEEE Computer Society, Washington DC, USA. October 2016.

# Dagstuhl-Seminar

**Model-driven Algorithms and Architectures for Self-Aware Computing Systems, Jan 18-23, 2015, Dagstuhl Seminar 15041**

**Organizers**

Jeffrey O. Kephart (IBM TJ Watson Research Center, US)
Samuel Kounev (Universität Würzburg, DE)
Marta Kwiatkowska (University of Oxford, GB)
Xiaoyun Zhu (VMware, Inc., US)

Community:
http://descartes.tools/self-aware
Dagstuhl Report:
http://drops.dagstuhl.de/opus/volltexte/2015/5038/
Seminar Page:
http://www.dagstuhl.de/15041

Schloss Dagstuhl
Where Computer Scientists Meet

# Definition

<u>**Self-aware Computing Systems**</u> are computing systems that:

1. ***learn models*** capturing knowledge about themselves and their environment ***on an ongoing basis*** and

2. ***reason*** using the models enabling them to ***act*** based on their knowledge and reasoning

in accordance with ***higher-level goals***, which may also be subject to change.

S. Kounev, P. Lewis, K. Bellman, N. Bencomo, J. Camara, A. Diaconescu, L. Esterle, K. Geihs, H. Giese, S. Goetz, P. Inverardi, J. Kephart and A. Zisman. **The Notion of Self-Aware Computing**. In Self-Aware Computing Systems, S. Kounev, J. O. Kephart, A. Milenkoski, and X. Zhu, editors. Springer Verlag, Berlin Heidelberg, Germany, 2017.

# Extended Definition

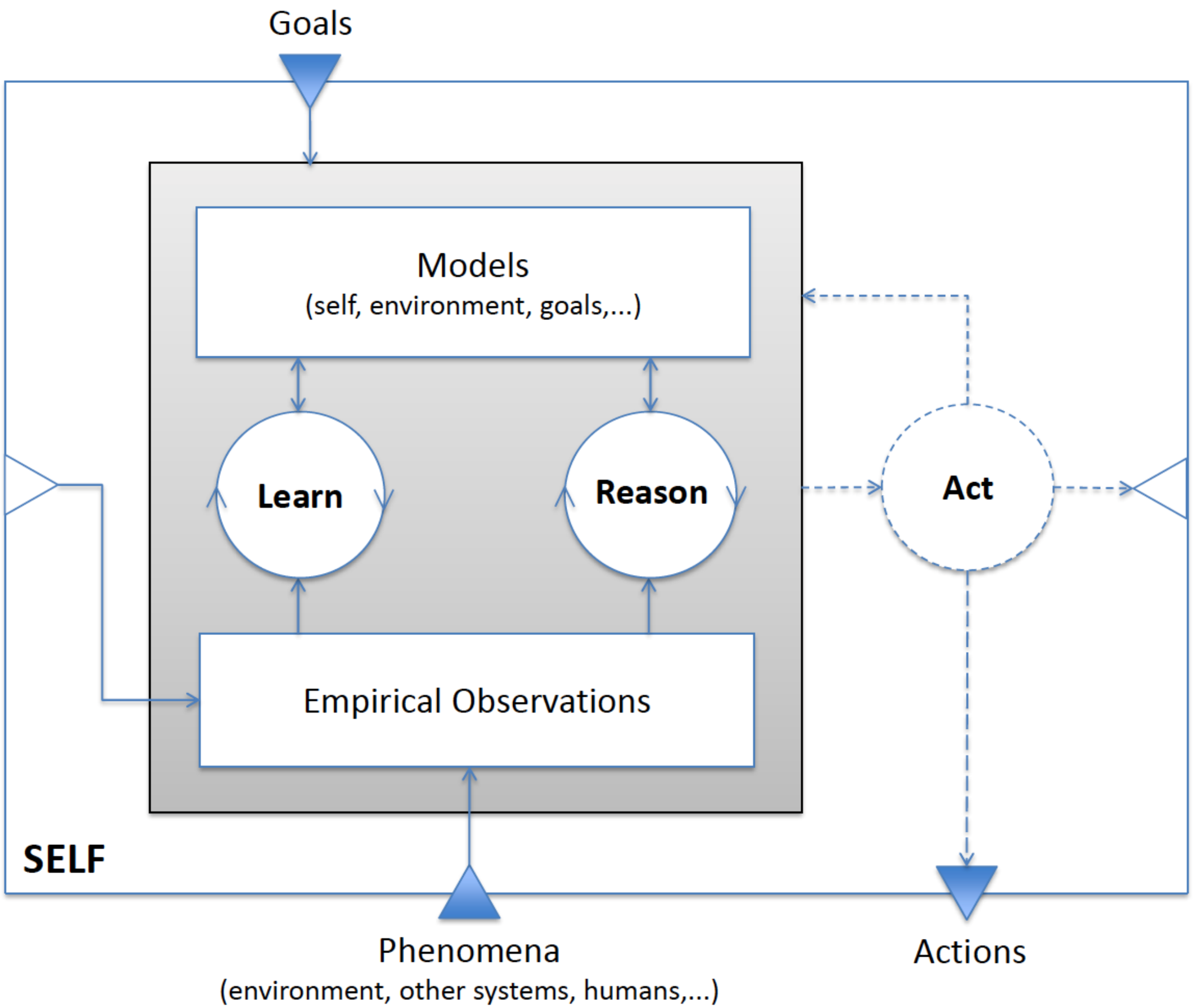
**Self-aware Computing Systems** are computing systems that:

1. ***learn models*** capturing ***knowledge*** about themselves and their environment (such as their structure, design, state, possible actions, and run-time behavior)
on an ongoing basis and

2. ***reason*** using the models (for example predict, analyze, consider, plan) enabling them to ***act*** based on their knowledge and reasoning (for example explore, explain, report, suggest, self-adapt, or impact their environment)

in accordance with ***higher-level goals***, which may also be subject to change.

# Self-Aware Learning & Reasoning Loop



# Models in Software Engineering

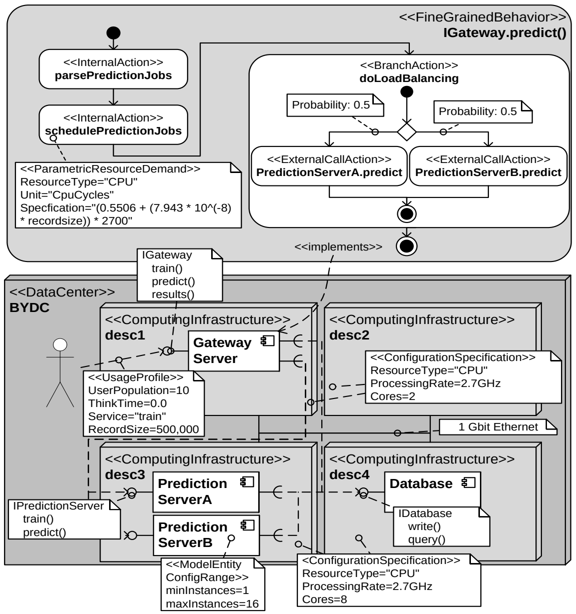
## Descriptive Models

- Capture relevant knowledge about the system and the environment in which it is running
- Describe selected aspects that have influence on the goal fulfilment

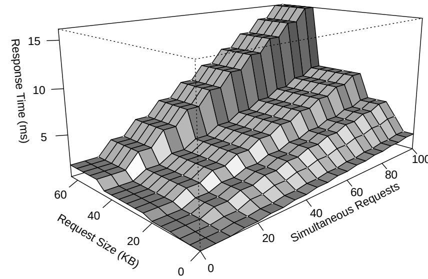
## (Predictive) Analysis Models

- Allow to reason about the system behavior
- Predict the impact of changes on the goal fulfilment

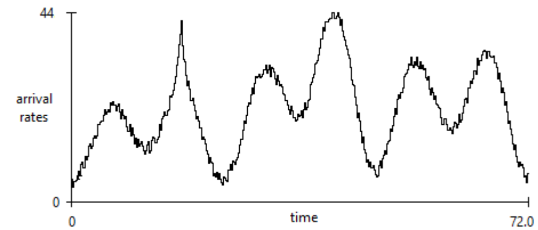
# Examples of Models



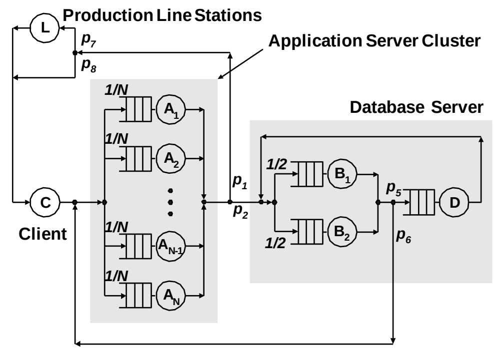

Descriptive MOF-based models



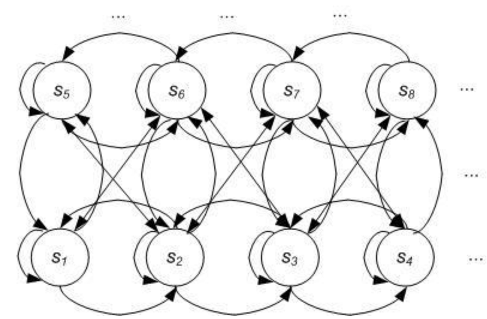

Statistical regression models



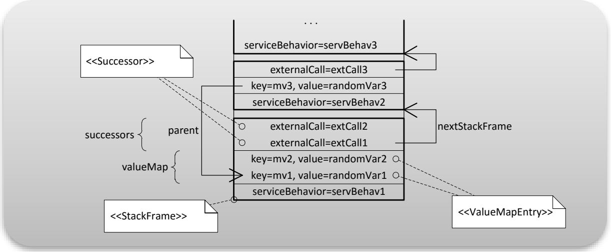

Load forecasting models




Queueing network models




Markov models




Simulation models

$$R \geq \max\left[N \times \max\{D_i\}, \sum_{i=1}^{K} D_i\right] \qquad X_0 \leq \min\left[\frac{1}{\max\{D_i\}}, \frac{N}{\sum_{i=1}^{K} D_i}\right]$$

$$\frac{N}{\max\{D_i\}[K+N-1]} \leq X_0 \leq \frac{N}{avg\{D_i\}[K+N-1]}$$

Analytical analysis models

# New Book

- **„Self-Aware Computing Systems“**
  Samuel Kounev (University of Würzburg, DE)
  Jeffrey O. Kephart (IBM T.J. Watson, USA)
  Aleksandar Milenkoski (University of Würzburg, DE)
  Xiaoyun Zhu (Futurewei Technologies, Huawei, USA)


- 27 chapters, ca 700 pages, ca. 50 authors involved

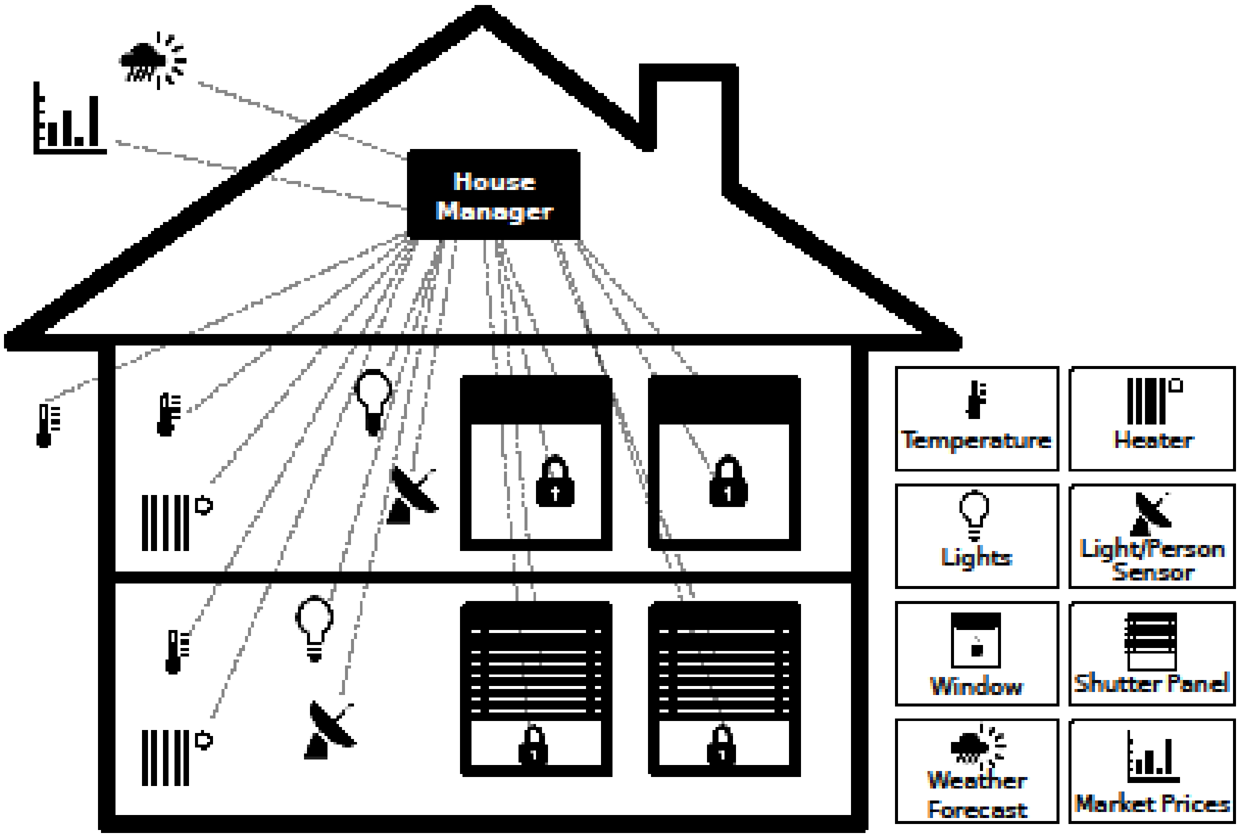
S. Kounev, J. O. Kephart, A. Milenkoski, and X. Zhu. (eds.) **Self-Aware Computing Systems**. Springer Verlag, Berlin Heidelberg, Germany, 2017. http://www.springer.com/de/book/9783319474724

# Implementing LRA-M

- Similar to the feedback loops for self-adaptive software, we argue that the LRA-M loop should be addressed during the architectural design of self-aware computing systems.

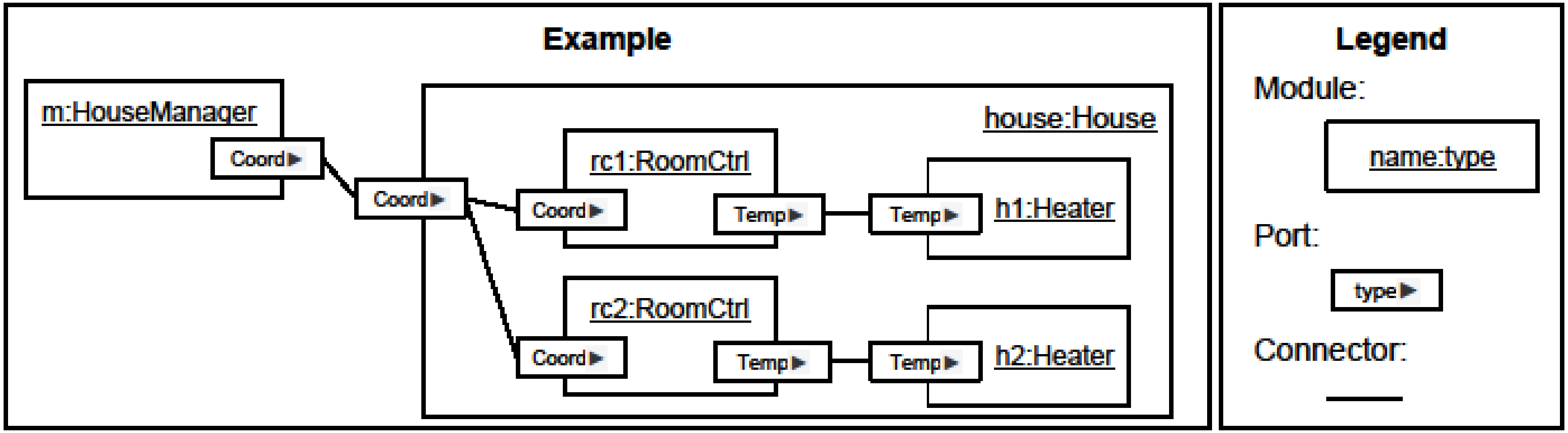
- **Challenges**
  - Visualize architectural concepts to address and make the LRA-M loop visible in the architectural design.
  - How to manifest context-awareness, self-awareness, and meta-self-awareness in an architecture?

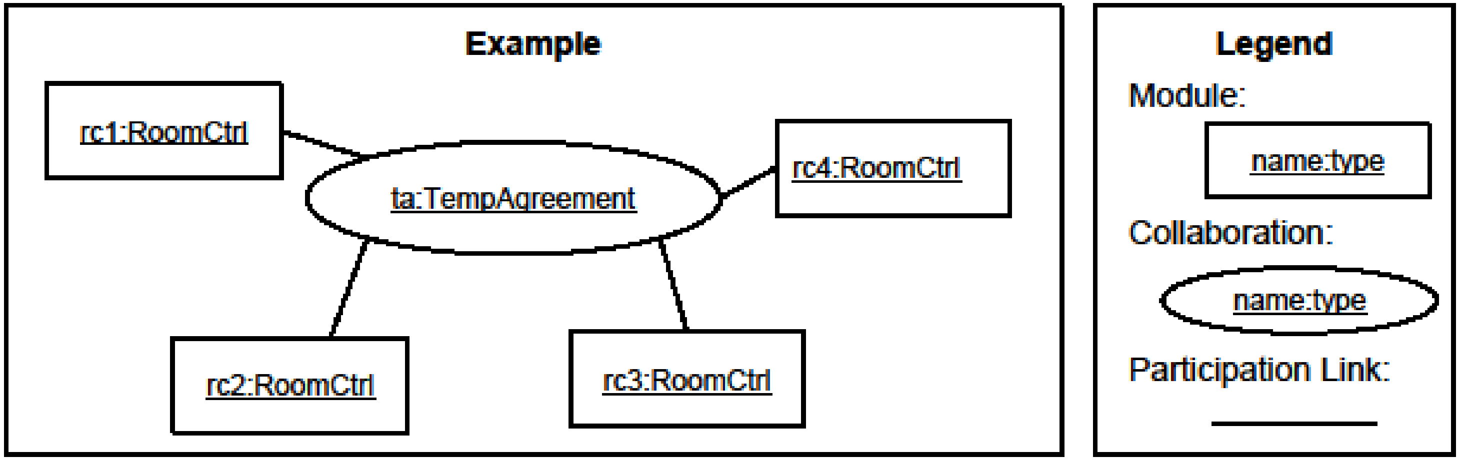
# Running Example: Smart Home



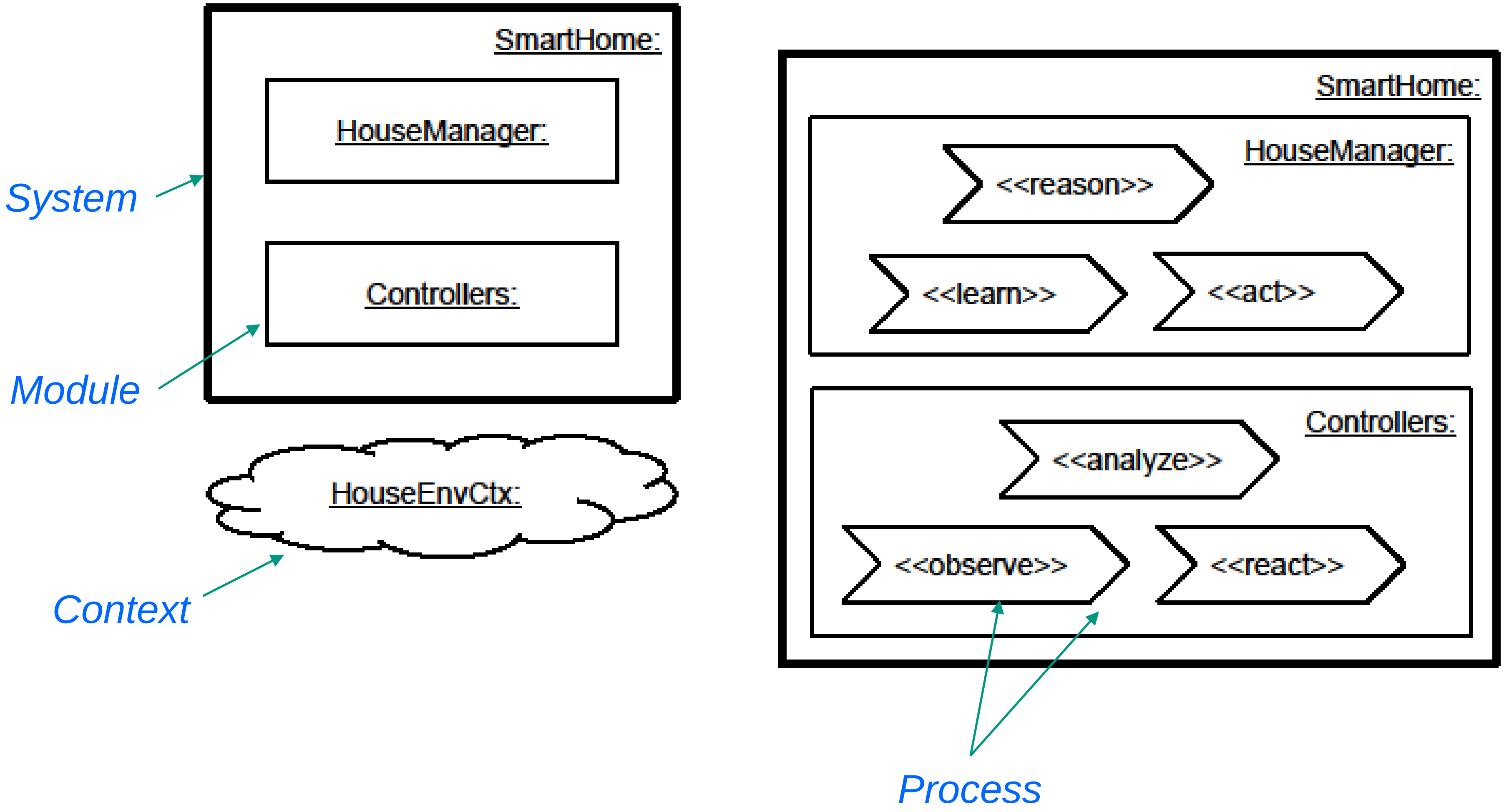

Smart home system architecture of components that control devices in a house and that are coordinated by a *house manager*

# Generic UML elements for architecture and collaboration modeling



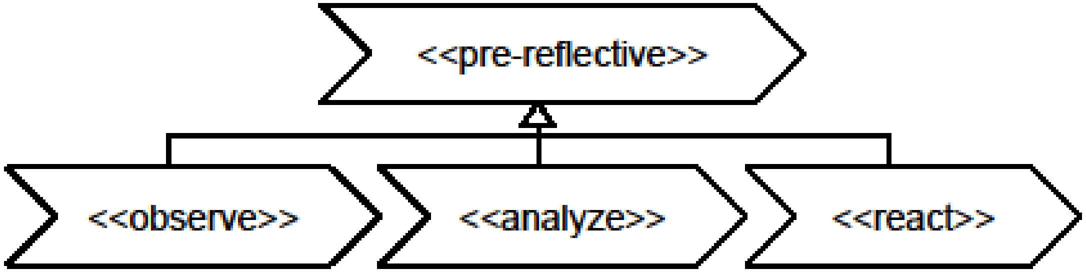



# System, Environmental Context, and Modules

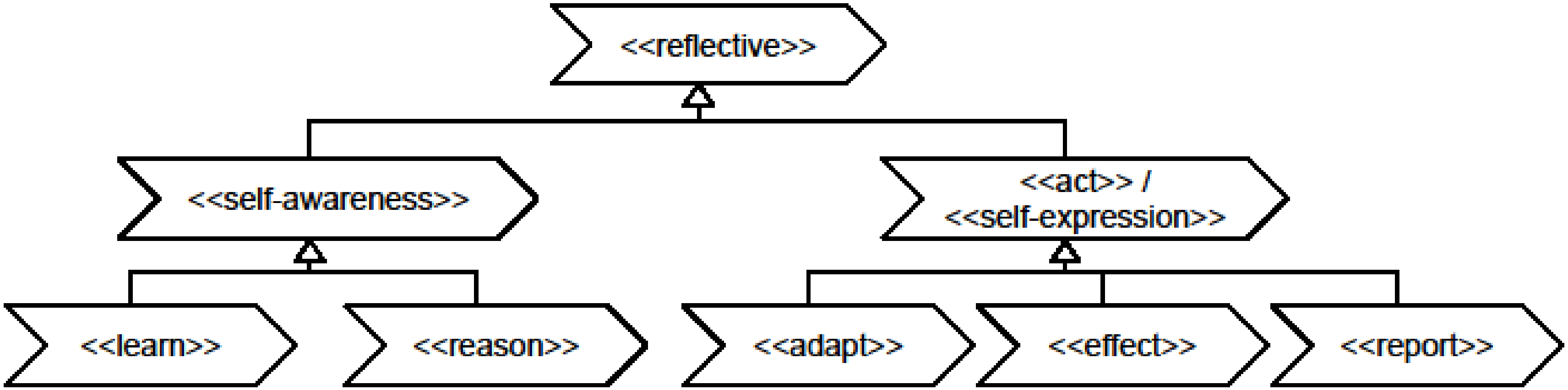


# Process Taxonomy

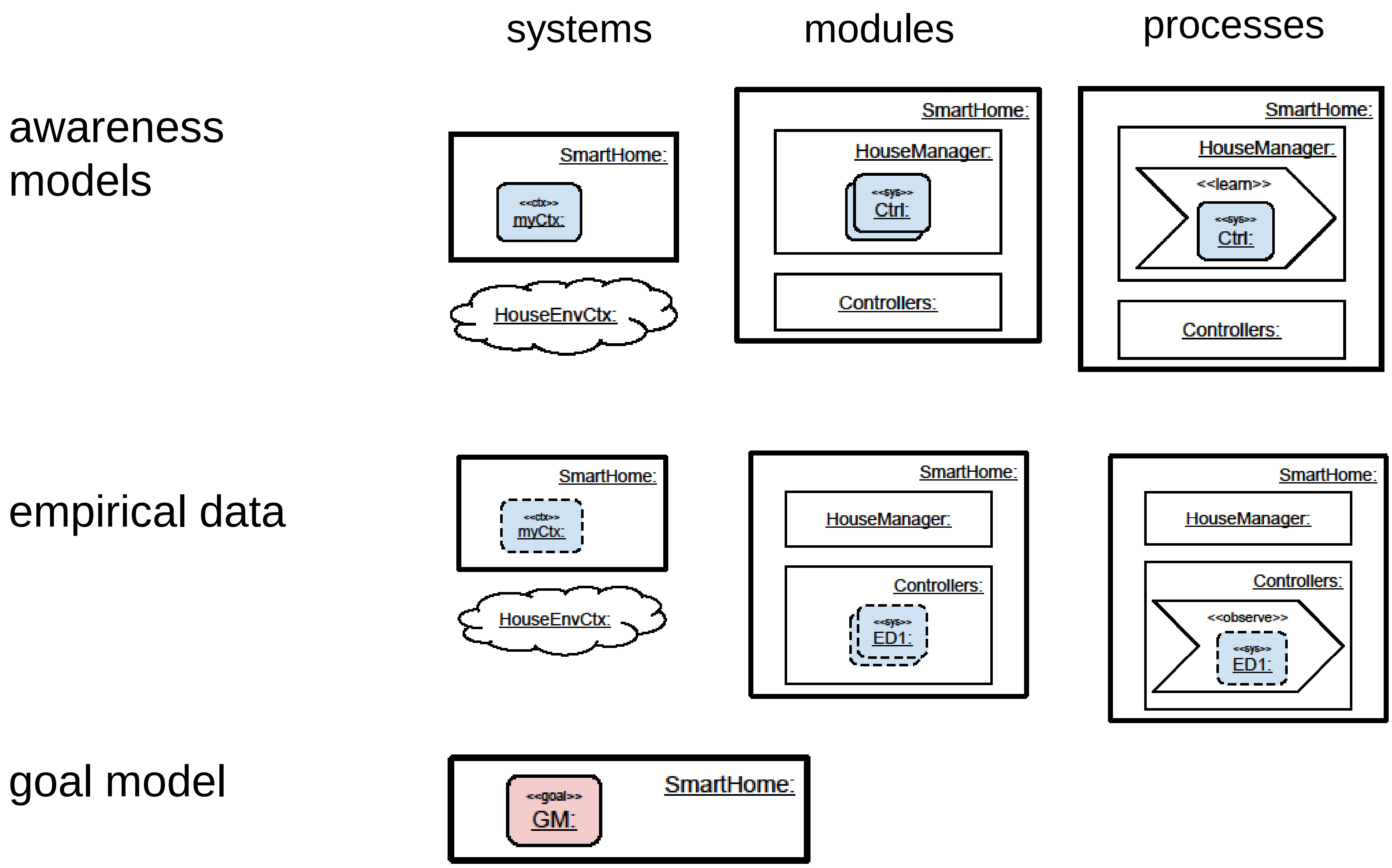
- Pre-reflective



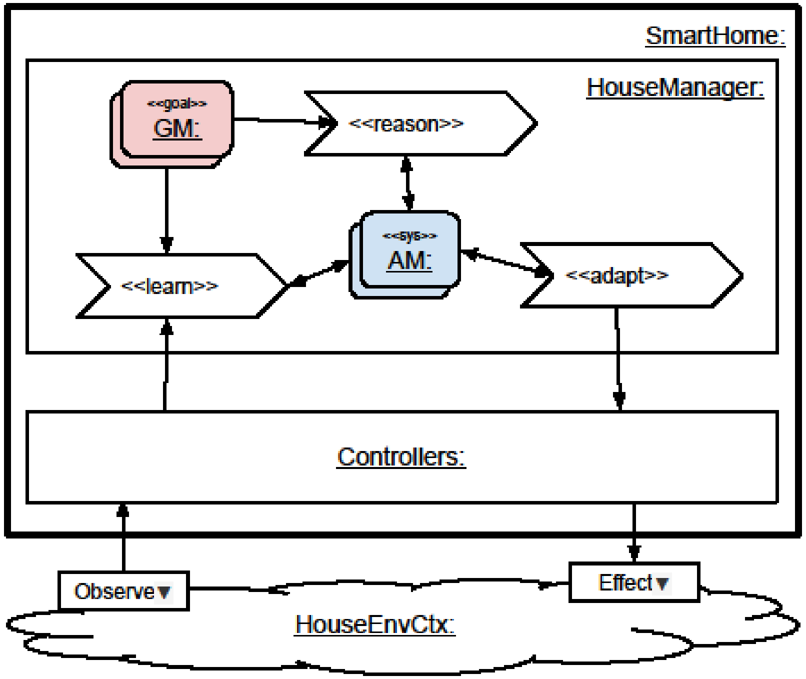

- Reflective

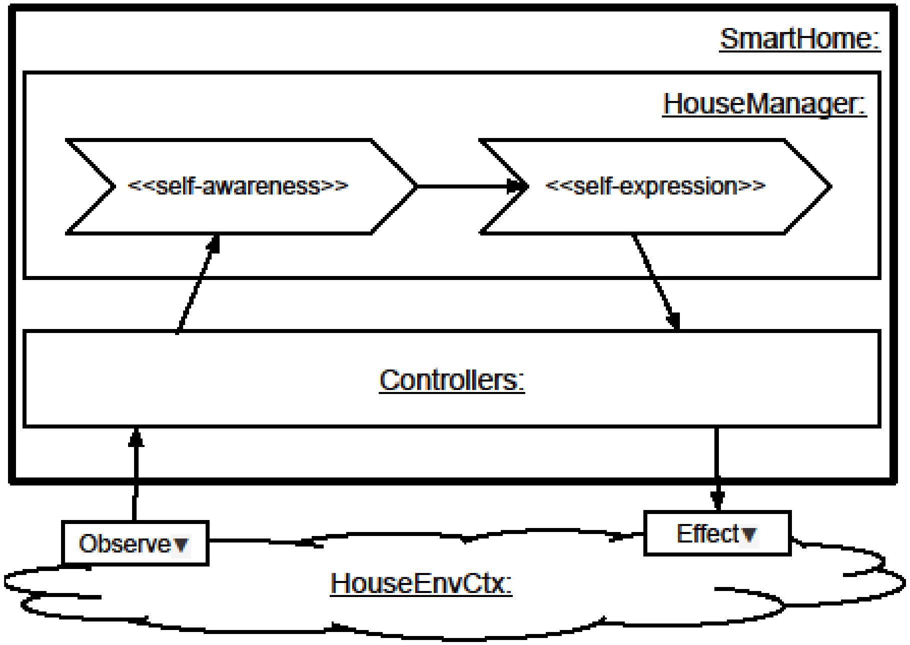


# Awareness Models, Empirical Data, and Goal Models



# Data Flow

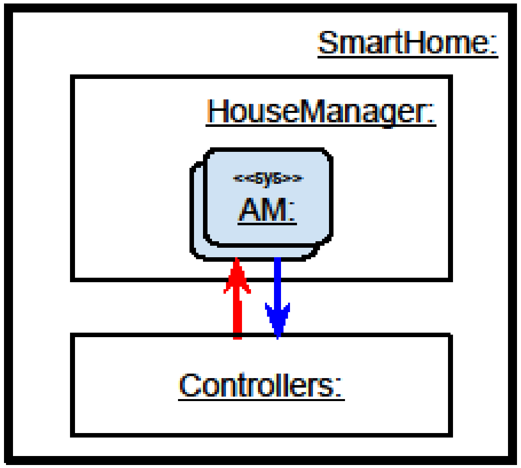


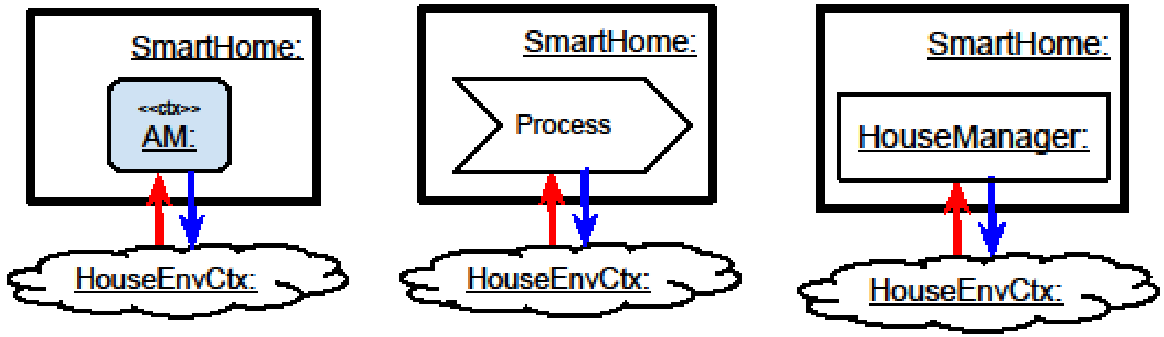



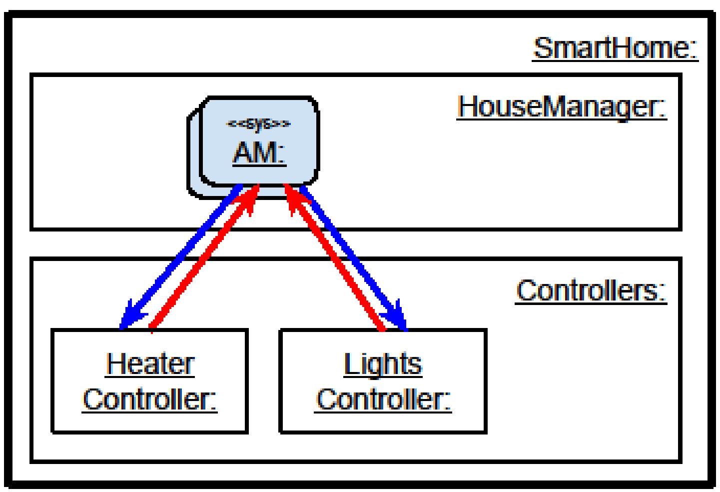

A learn process obtains AMs guided by the goals of the self, a reason process uses the AMs and GMs to reason, and finally, an adapt process uses the AMs to dynamically change a module

Abstraction hiding the employed models only capturing data-flow between these two processes and the Controllers module

# Awareness and Expression




*awareness link*

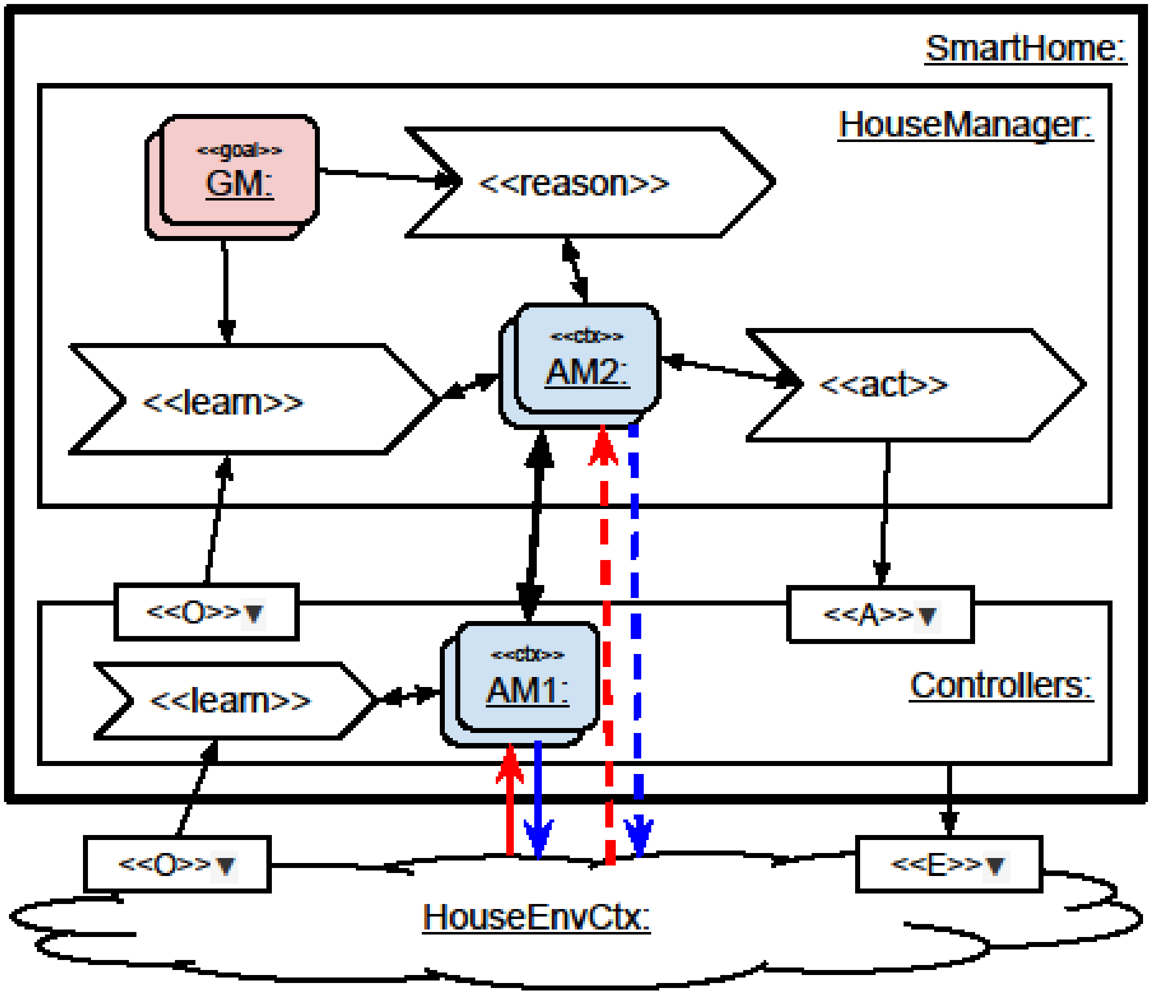
An *awareness link* denotes that a scope is represented by a model maintained by a span. Thus, the span is aware of the scope.

*expression link*

An *expression link* denotes that the span's self-expression impacts the scope.

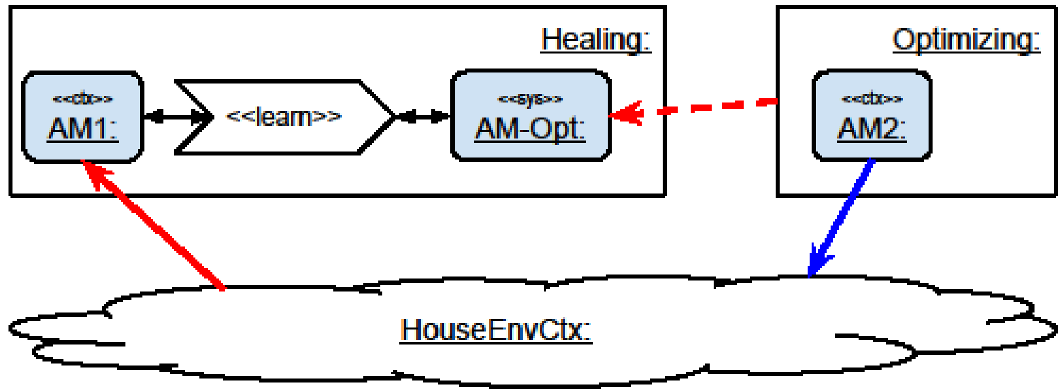
If we want to abstract the models in an architectural view, we link the scope to a process or module containing the (hidden) models

# Direct and Indirect Awareness

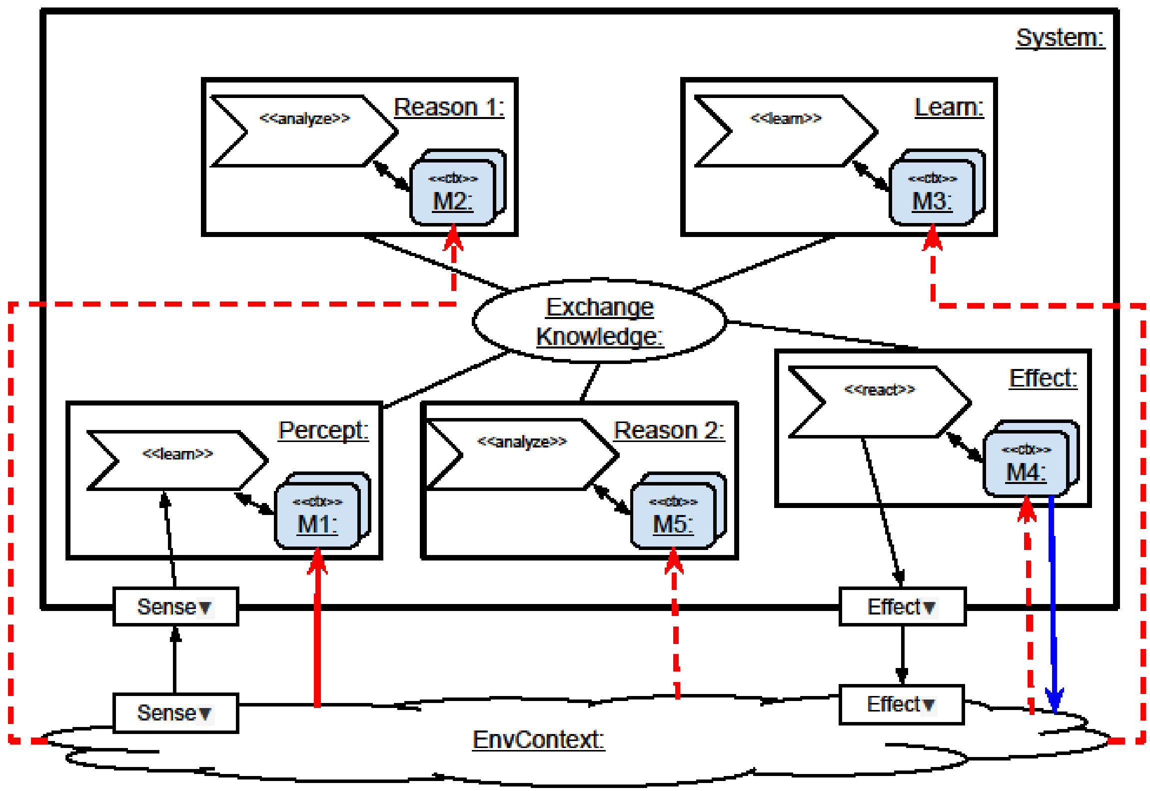


# Direct and Indirect Awareness II

Indirect awareness via the environmental context




Direct and indirect awareness with collaborations



Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Descartes research

# CASE STUDY

# Self-Aware Performance and Resource Management in Shared IT Infrastructures

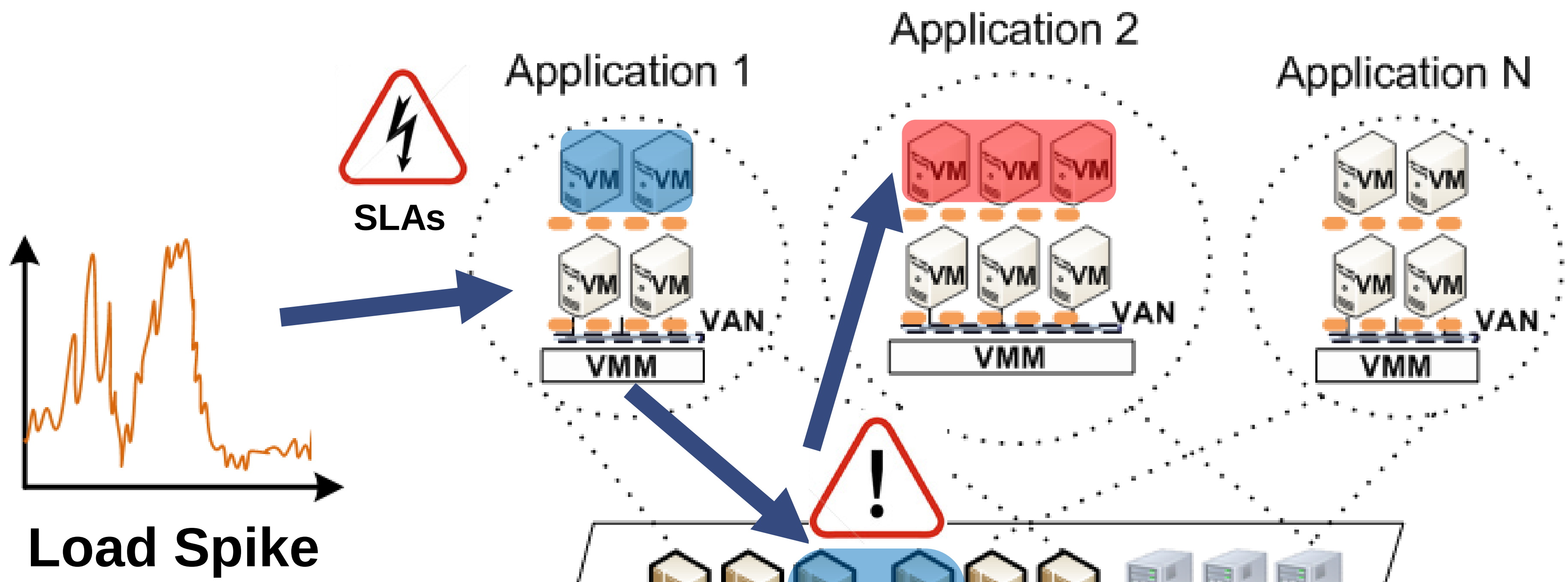
# References


S. Kounev, N. Huber, F. Brosig, and X. Zhu. ***A Model-Based Approach to Designing Self-Aware IT Systems and Infrastructures***. IEEE Computer, 49(7):53–61, July 2016.

N. Huber, F. Brosig, S. Spinner, S. Kounev, and M. Bähr. ***Model-Based Self-Aware Performance and Resource Management Using the Descartes Modeling Language***. IEEE Transactions on Software Engineering (TSE), PP(99), 2017.

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING

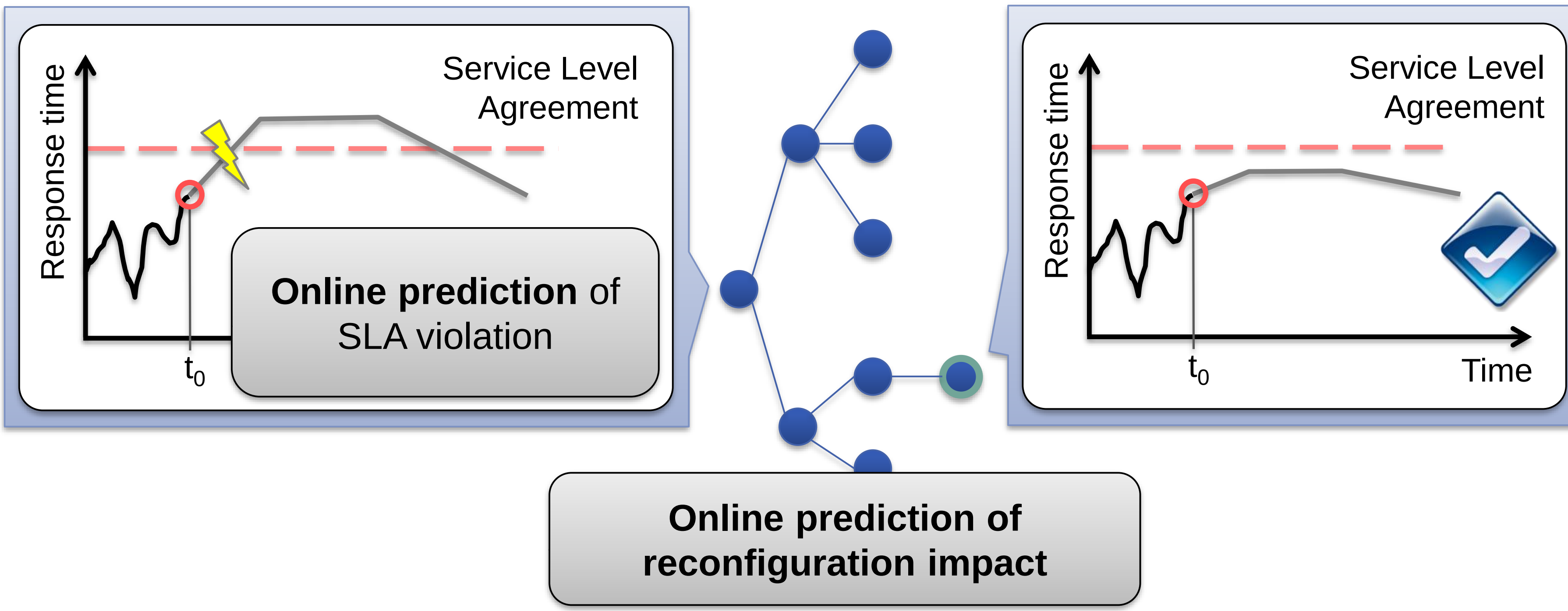
# Challenges: Availability & Performance




**Elastic (auto)-scaling of resources at run-time**

- How can one predict the load spike?
- When exactly should a reconfiguration (scaling) be triggered?
- Which particular resources should be scaled?
- How quickly and at what granularity?

# Proactive Auto-Scaling




→ Example Scenario for Self-Aware Computing

# Descartes Tool Chain

**http://descartes.tools**

# Selected Tools

- **DML** – Descartes Modeling Language (homepage, publications)
- **DML Bench** (homepage, publications)
- **DQL** – Declarative performance query language (homepage, publications)
- **LibReDE** - Library for resource demand estimation (homepage, publications)
- **LIMBO** – Load intensity modeling tool (homepage, publications)
- **WCF** – Workload classification & forecasting tool (homepage, publications)
- **BUNGEE** – Elasticity benchmarking framework (homepage, publications)
- **hInjector** – Security benchmarking tool (homepage, publications)
- Queueing Petri Net Modeling Environment (QPME)
- **Further relevant research**
  - **http://descartes-research.net/research/research_areas/**
  - **Self Aware Computing** (publications)

# Descartes Tools

**Descartes Modeling Language:**

- DML (Descartes Modeling Language)
- DNI (Descartes Network Infrastructures Modeling)

**Workload Characterization & Model Extraction:**

- LIMBO Load Intensity Modeling Tool
- WCF (Workload Classification and Forecasting Tool)
- LibReDE (Library for Resource Demand Estimation)
- SPA (Storage Performance Analyzer)
- PMX (Performance Model eXtractor)

**Declarative Performance Engineering:**

- DQL (Descartes Query Language)

**Benchmarking:**

- BUNGEE Cloud Elasticity Benchmark
- hInjector Hypercall Attack Injector

**Stochastic Modeling:**

- QPME (Queueing Petri net Modeling Environment)

**Black-Box Modeling:**

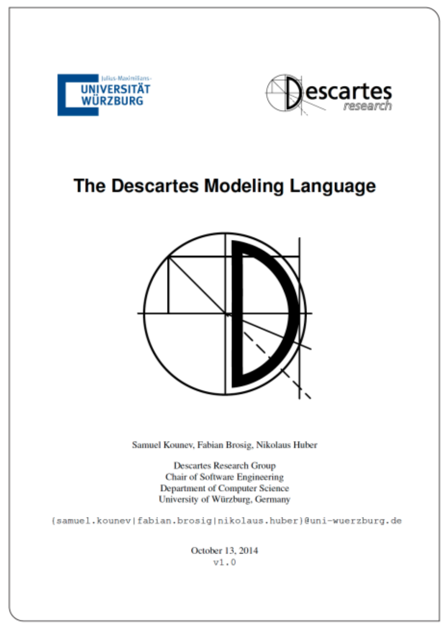
- Univariate Interpolation Library

**http://descartes.tools**

Mailing list available...

# Descartes Modeling Language (DML)

- Architecture-level modeling language for modeling QoS and resource management related aspects of IT systems and infrastructures
  - Prediction of the impact of dynamic changes at run-time
  - Current version focused on performance including capacity, responsiveness and resource efficiency aspects


Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Descartes
research

The Descartes Modeling Language

Samuel Kounev, Fabian Brosig, Nikolaus Huber

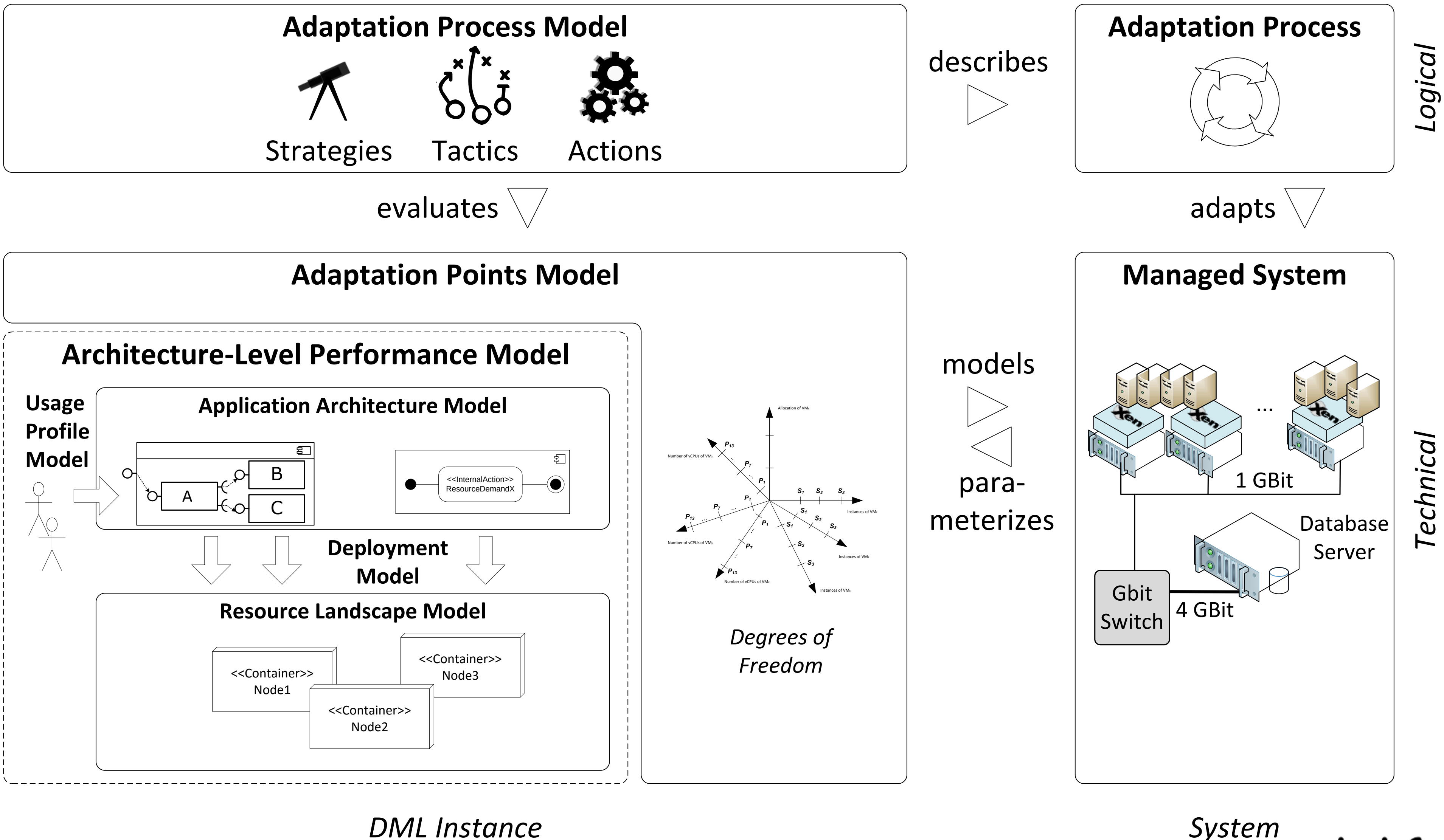
Descartes Research Group
Chair of Software Engineering
Department of Computer Science
University of Würzburg, Germany

{samuel.kounev|fabian.brosig|nikolaus.huber}@uni-wuerzburg.de

October 13, 2014
v1.0

**http://descartes.tools/dml**

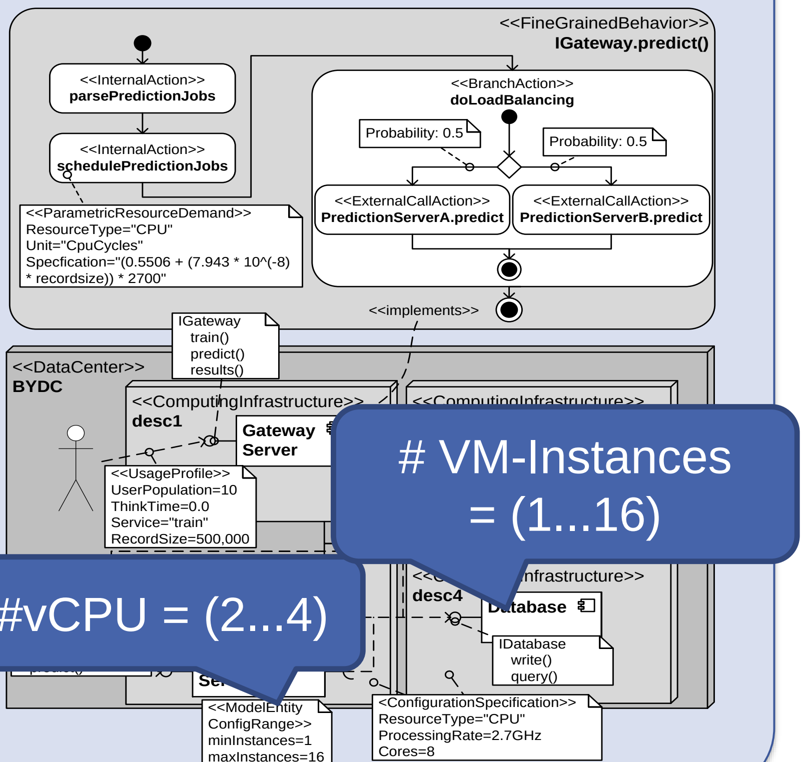
# Big Picture



Adaptation Process Model
Strategies
Tactics
Actions
describes
Adaptation Process
Logical
evaluates
adapts
Adaptation Points Model
Managed System
Architecture-Level Performance Model
Usage Profile Model
Application Architecture Model
A
B
C
<<InternalAction>>
ResourceDemandX
Deployment Model
Resource Landscape Model
<<Container>>
Node1
<<Container>>
Node2
<<Container>>
Node3
Degrees of Freedom
models
para-meterizes
1 GBit
Database Server
Gbit Switch
4 GBit
Technical
DML Instance
System

# Online Performance Prediction

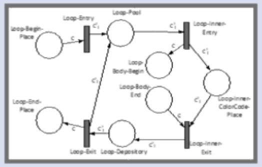
## Architecture-Level Performance Model

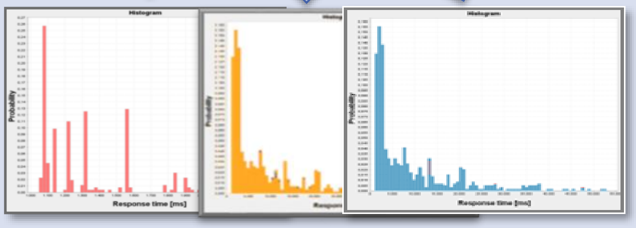



## Online Performance Prediction

$$\overline{X} \le min\left\{\frac{N}{\sum_{i=0}^{n} D_i^{sync}}, min_{1\le i\le n}\left\{\frac{1}{D_i}\right\}\right\}$$

$$\overline{R} = \frac{N}{\overline{X}} \ge max\left\{\sum_{i=0}^{n} D_i^{sync}, N * max_{1\le i\le n}\{D_i\}\right\}$$






## Autonomic Decision Making

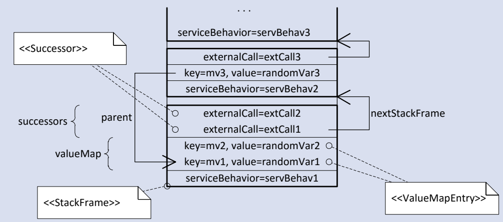
# Tailored Model Solution

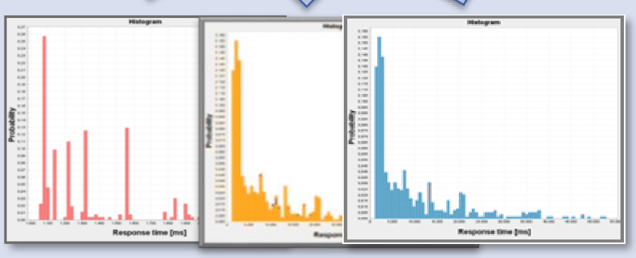
## Analytical Analysis

$$R \geq \max\left[N \times \max\{D_i\}, \sum_{i=1}^{K} D_i\right] \qquad X_0 \leq \min\left[\frac{1}{\max\{D_i\}}, \frac{N}{\sum_{i=1}^{K} D_i}\right]$$

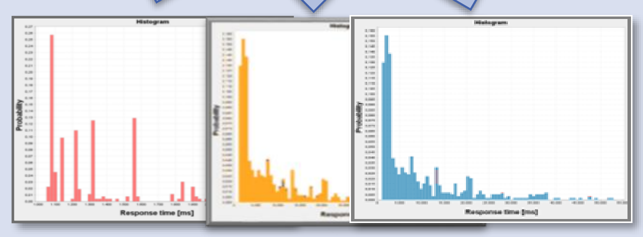
$$\frac{N}{\max\{D_i\}[K+N-1]} \leq X_0 \leq \frac{N}{avg\{D_i\}[K+N-1]}$$

## Simulative Analysis



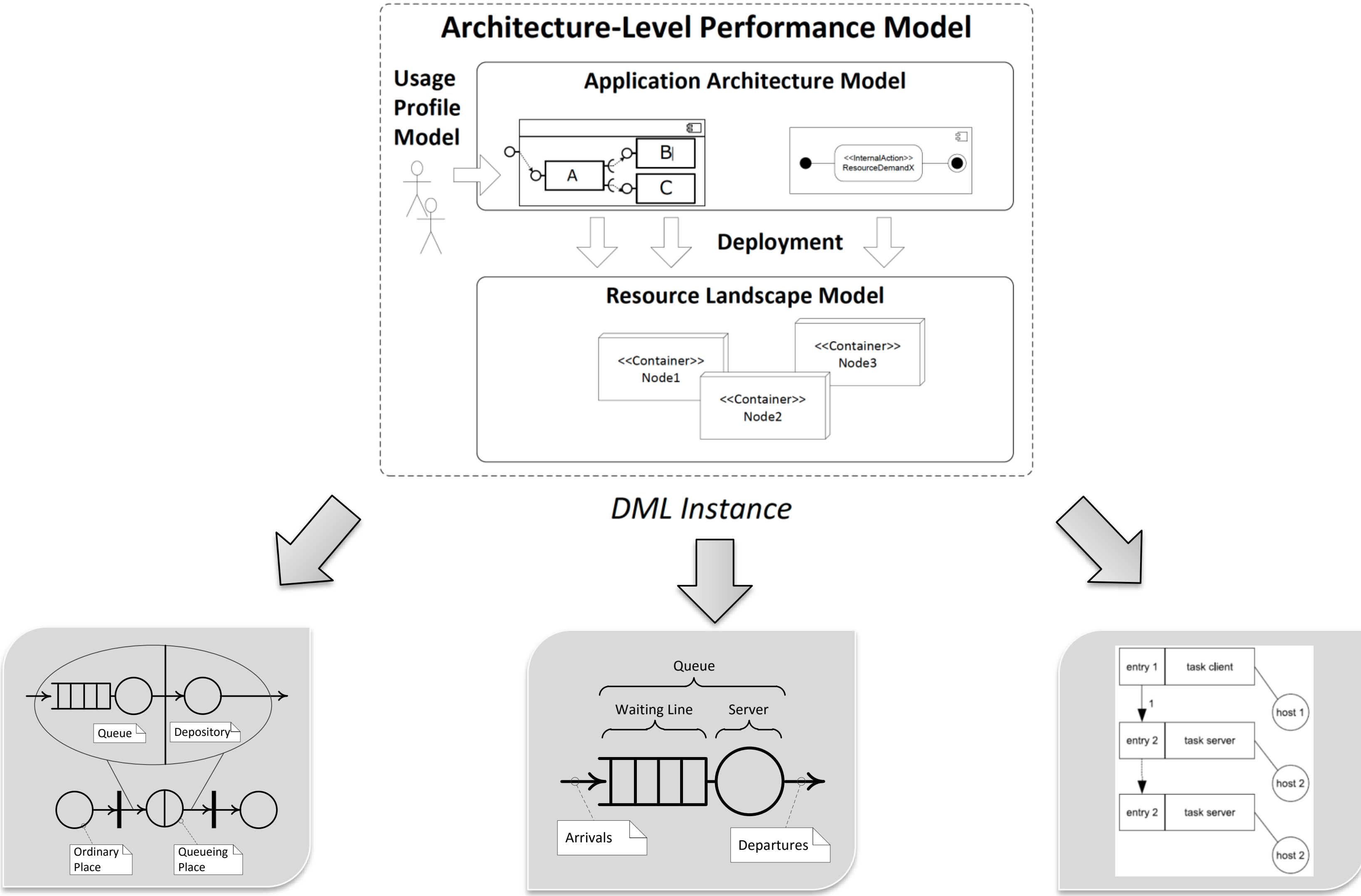
Analysis Results



Analysis Results

Fabian Brosig, Philipp Meier, Steffen Becker, Anne Koziolek, Heiko Koziolek, and Samuel Kounev. **Quantitative Evaluation of Model-Driven Performance Analysis and Simulation of Component-based Architectures**. *IEEE Transactions on Software Engineering (TSE)*, 41(2):157-175, February 2015, IEEE. [ DOI | http | .pdf ]

# Transformations to Predictive Models

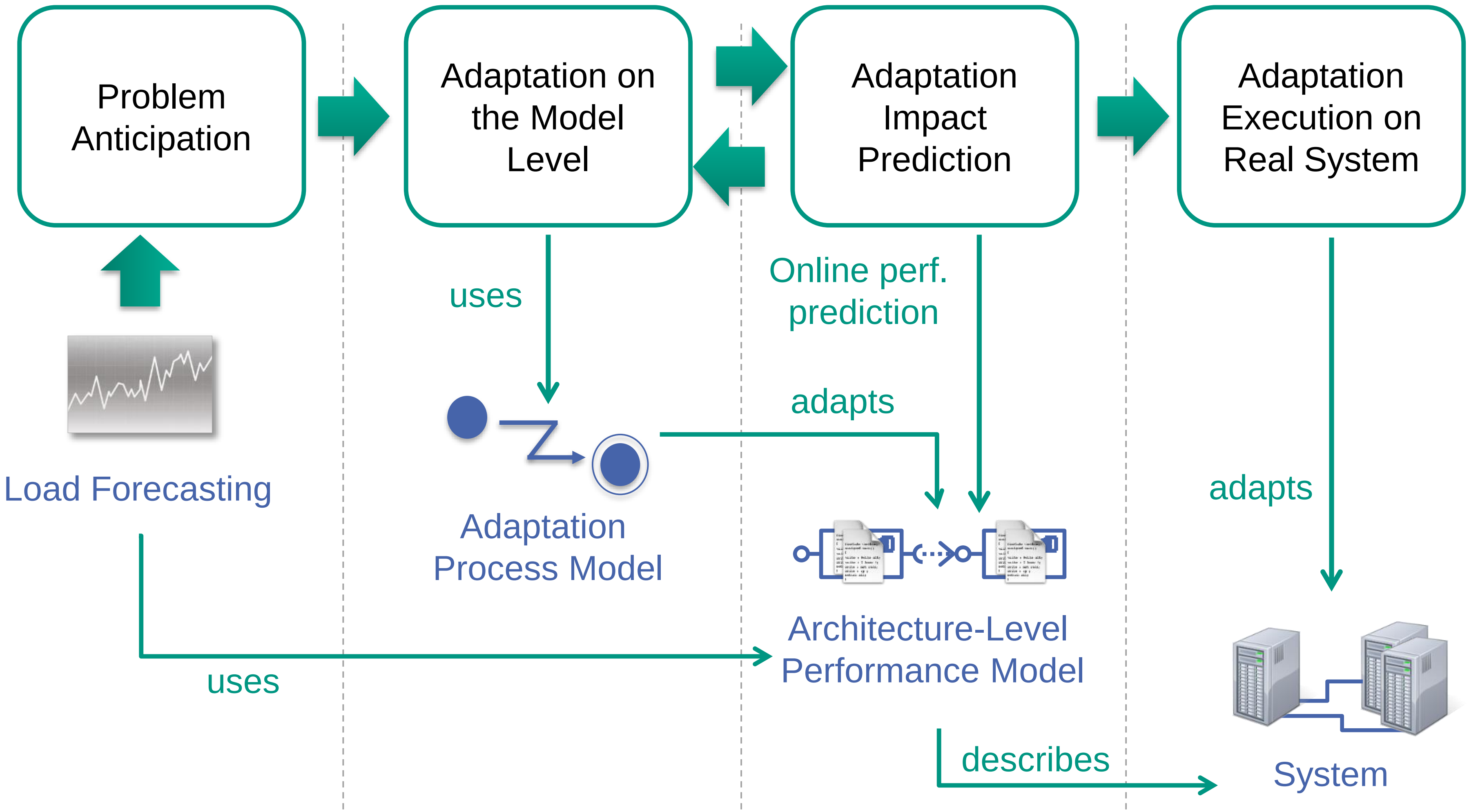



Queueing Petri Net

Bounds Analysis Model

Layered Queueing Network

# Model-Based System Adaptation



# Latest Publications on DML


S. Kounev, N. Huber, F. Brosig, and X. Zhu. ***A Model-Based Approach to Designing Self-Aware IT Systems and Infrastructures***. IEEE Computer, 49(7):53–61, July 2016.

N. Huber, F. Brosig, S. Spinner, S. Kounev, and M. Bähr. ***Model-Based Self-Aware Performance and Resource Management Using the Descartes Modeling Language***. IEEE Transactions on Software Engineering (TSE), PP(99), 2017.

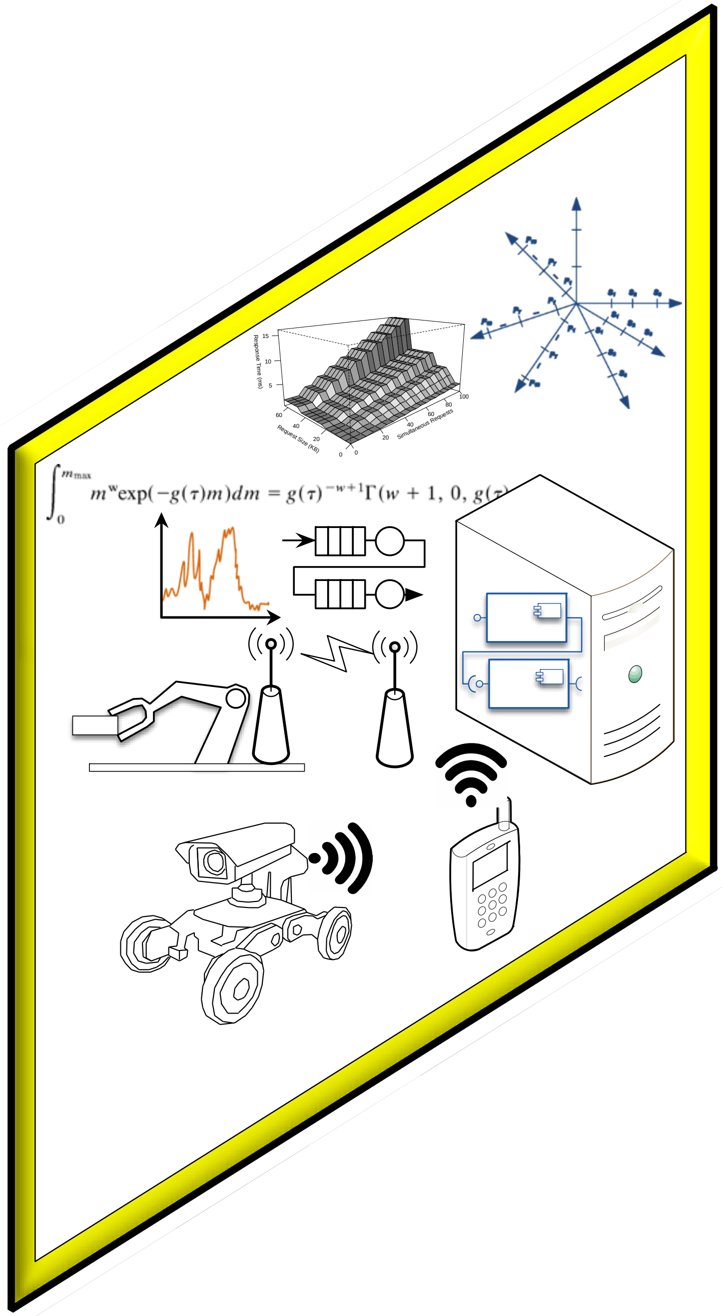
# Links for Further Information

- **DML** – Descartes Modeling Language (homepage, publications)
- **DML Bench** (homepage, publications)
- **DQL** – Declarative query language (homepage, publications)
- **DNI** – Descartes network infrastructure modeling (homepage, publications)
- **LibReDE** - Library for resource demand estimation (homepage, publications)
- **LIMBO** – Load intensity modeling tool (homepage, publications)
- **WCF** – Workload classification & forecasting tool (homepage, publications)
- **BUNGEE** – Elasticity benchmarking framework (homepage, publications)
- **hInjector** – Security benchmarking tool (homepage, publications)
- **Further relevant research**
  - **http://descartes-research.net/research/research_areas/**
  - **Self Aware Computing** (publications)

# Summary

- Pressure to raise efficiency by sharing IT resources
- Resource sharing poses challenges
- 1st Generation Cloud Computing
  - **Simple trigger/rule-based mechanisms**
  - Best effort approach
  - No dependability guarantees
- **Novel model-based approaches** enable self-aware performance and resource management
  - proactive and predictable approach

# The Vision




# Self-Aware Computing

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Descartes research

# Questions?

skounev@acm.org
http://descartes.tools