Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Descartes research

Self-Aware Performance and Resource Management in Shared IT Infrastructures

Samuel Kounev

Chair of Software Engineering
University of Würzburg

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

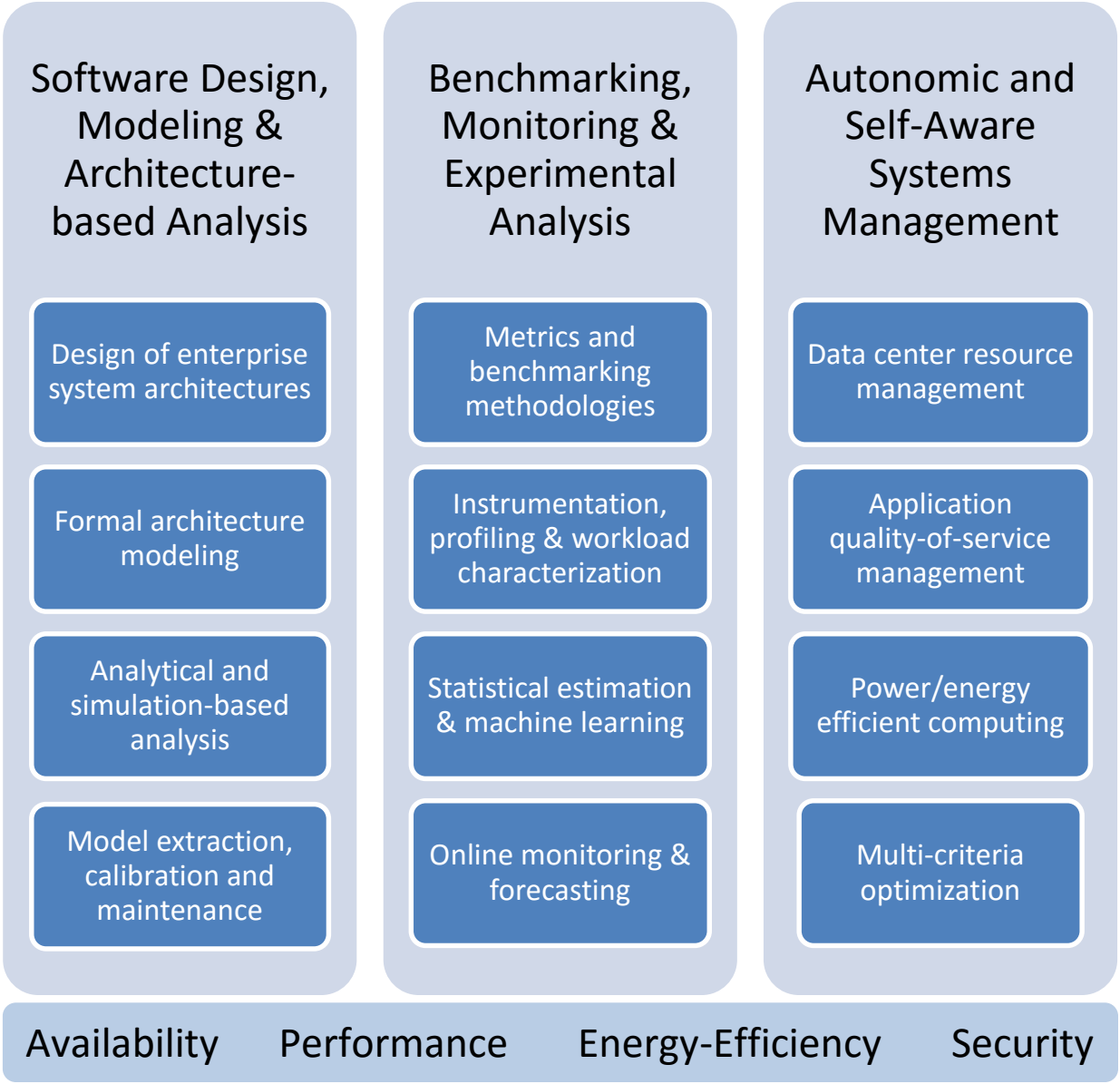
Huawei, Shenzhen, 17.01.17

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.

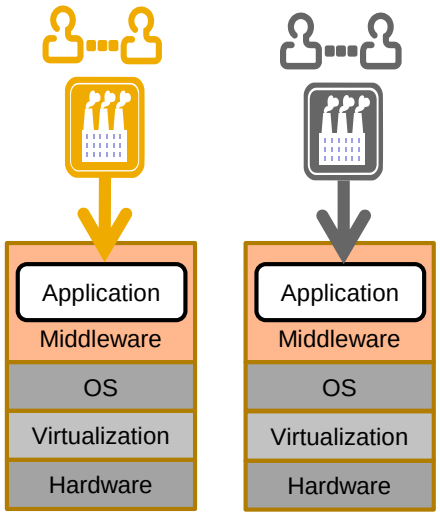
Chair of Software-Engineering (September 2016)

Research Areas

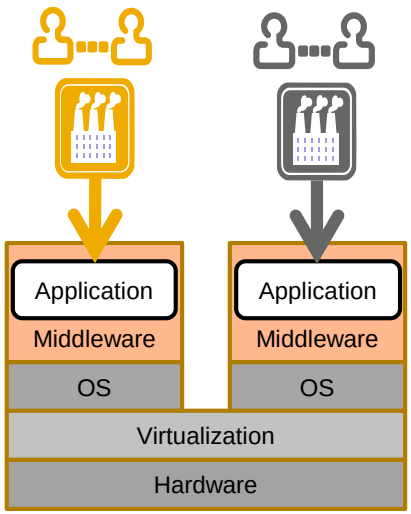


Increasing Pressure to Raise Efficiency

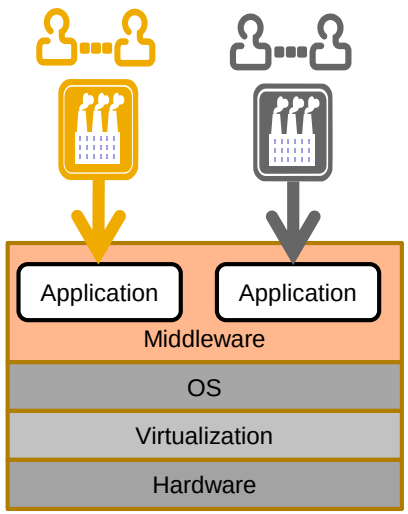
- Proliferation of **shared IT infrastructures** → Cloud Computing
- Different forms of resource sharing (hardware and software)
  - Network, storage, and computing infrastructure
  - Software stacks



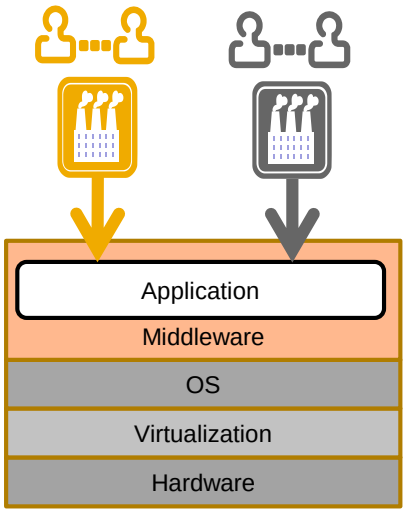
Datacenter Sharing



Hardware Sharing
through
Server Virtualization
(e.g., as in IaaS)



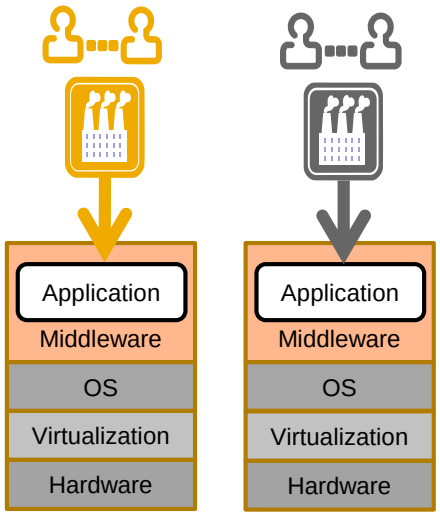
Platform Sharing
through
Shared Middleware
(e.g., as in PaaS)



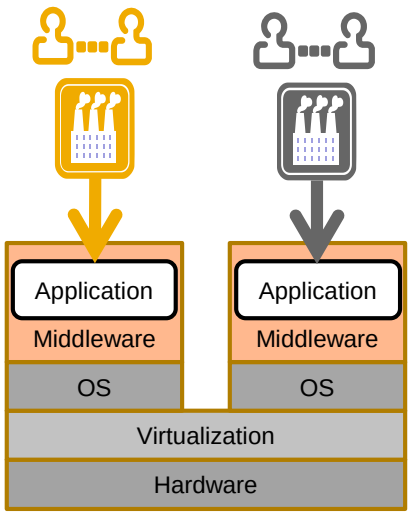
Application Sharing
→ Multi-Tenancy
(e.g., as in SaaS)

Increasing Pressure to Raise Efficiency

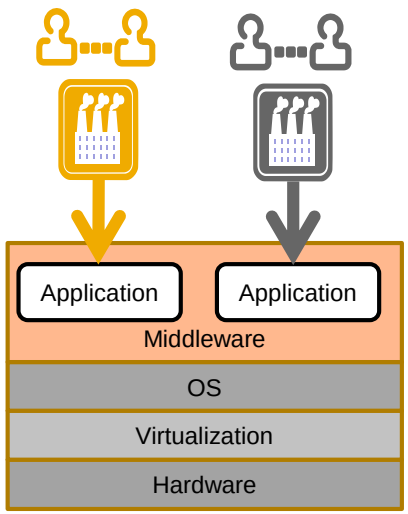
Isolation → **Efficiency**



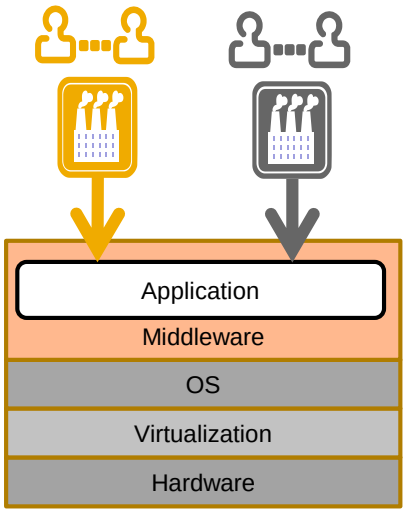
Datacenter Sharing



Hardware Sharing
through
Server Virtualization
(e.g., as in IaaS)



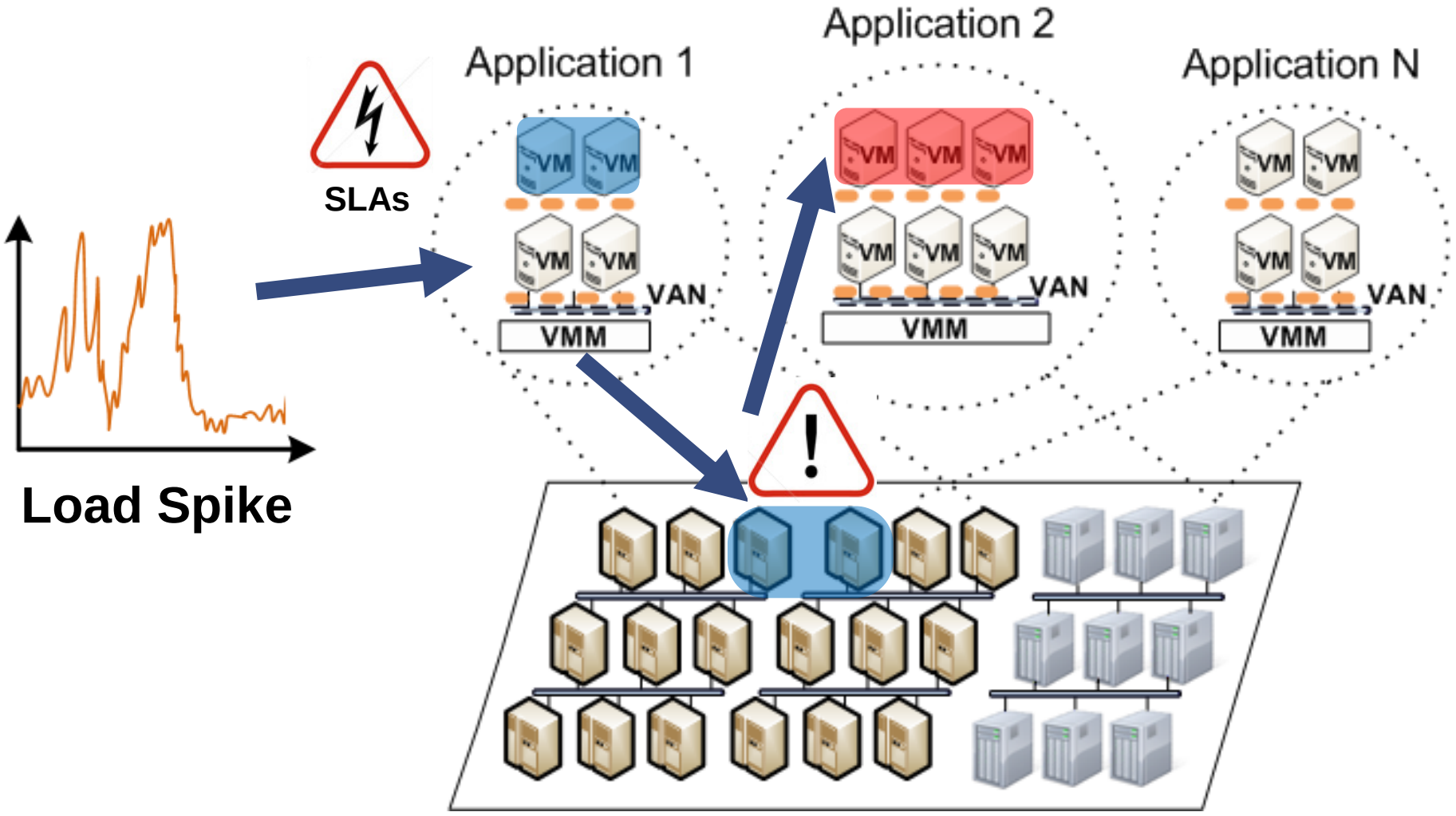
Platform Sharing
through
Shared Middleware
(e.g., as in PaaS)



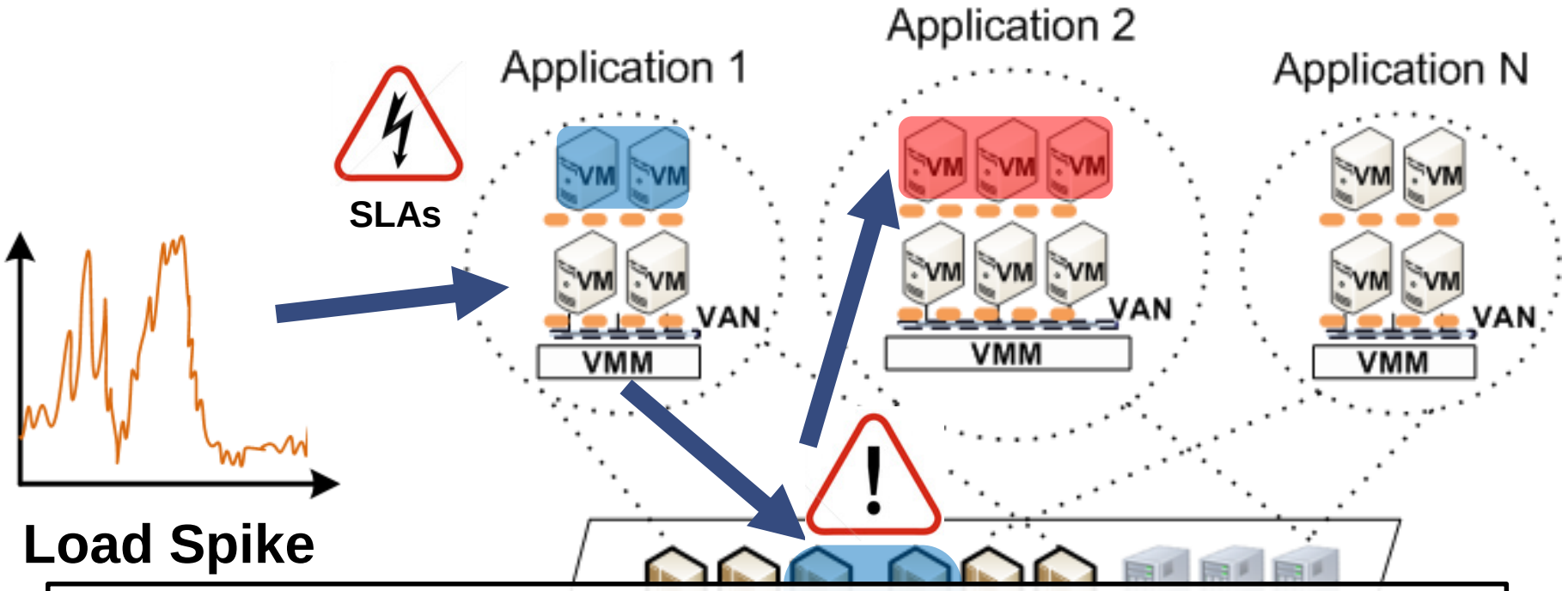
Application Sharing

→ Multi-Tenancy
(e.g., as in SaaS)

Challenges: Availability & Performance



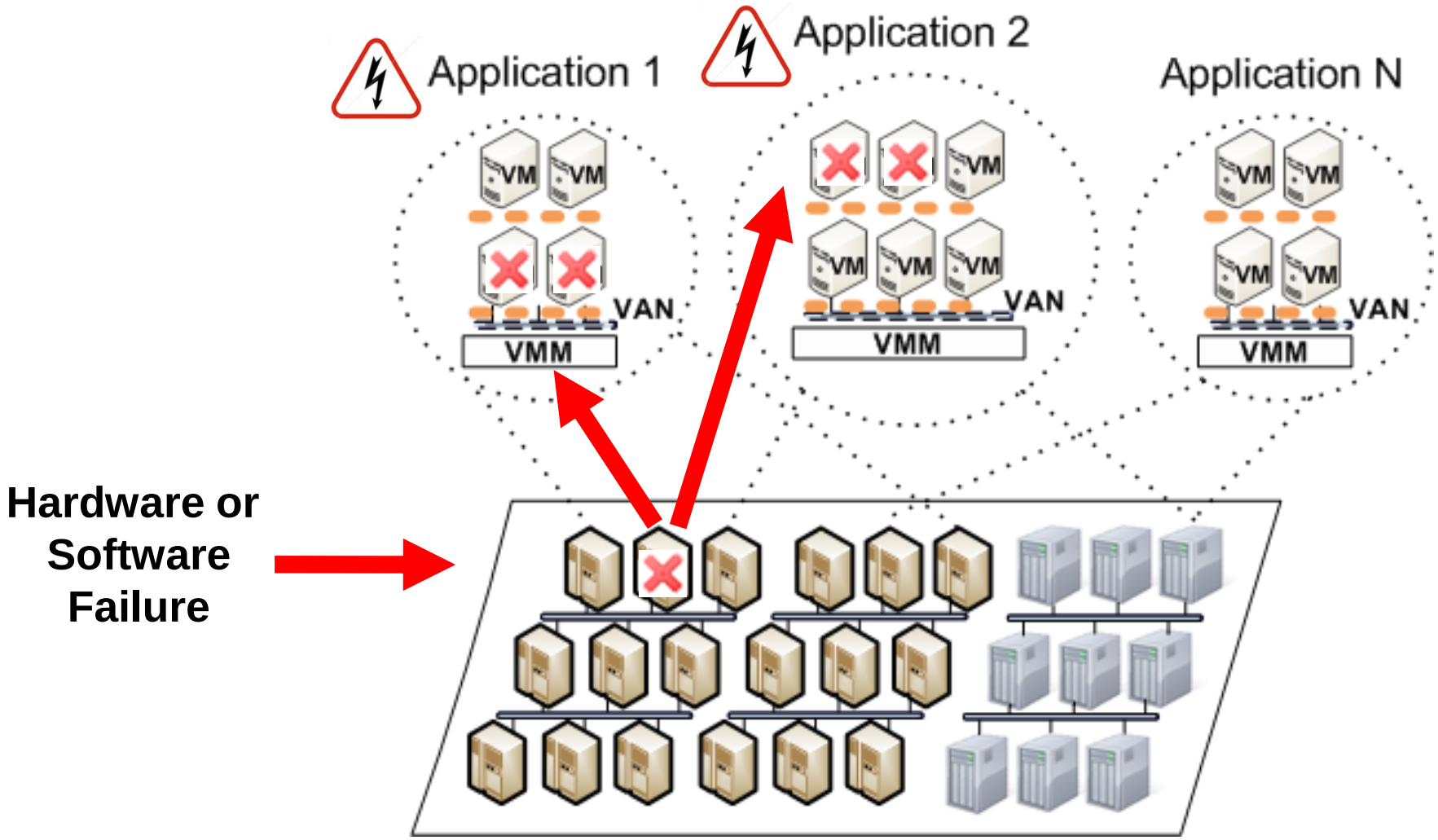
Challenges: Availability & Performance



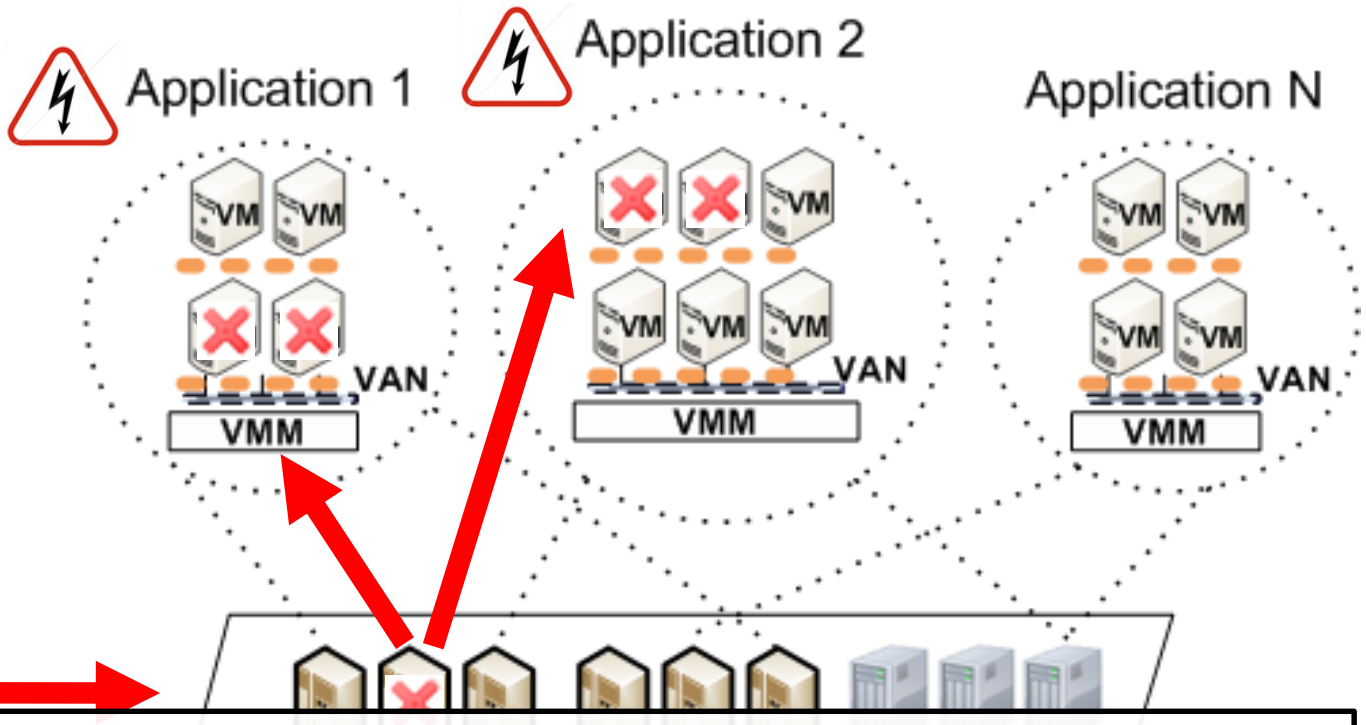
Elastic (auto)-scaling of resources at run-time

- 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?

Challenges: Reliability



Challenges: Reliability



- How can one predict and prevent failures?
- When exactly should a reconfiguration be triggered?
- Which system components / services should be restarted?

State-of-the-Art

- Hard to predict workload changes and their effect on the system performance (system overload and/or system failures)
  - Minimize risks by over-provisioning resources

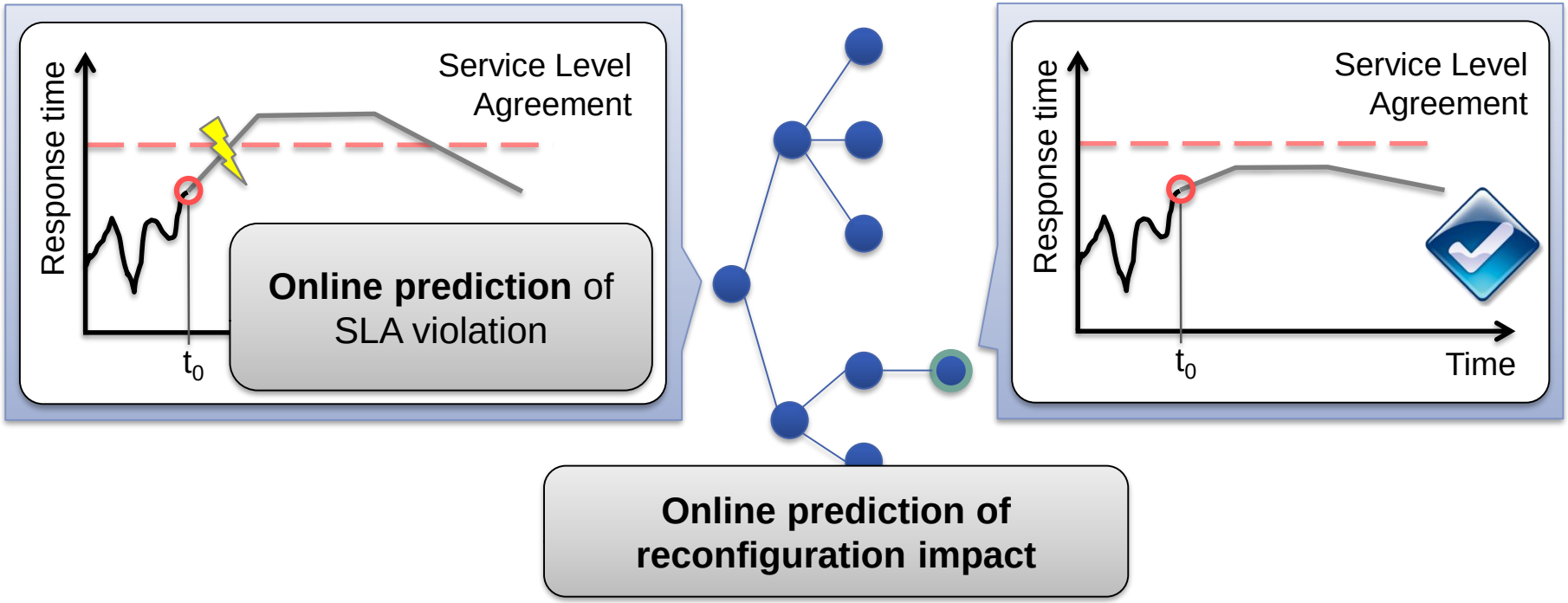
    AND / OR
  - Rely on simple rule-based “best effort” adaptation techniques

- Consequences: Poor resource efficiency
  - Rising energy costs for IT systems
    - 1600% increase by 2025 [Gartner]
  - Rising global CO2 emissions of ICT sector
    - Today: ca 3%, Increase to 10% expected in 10 years [EU]

1st Generation Cloud Computing

- Flexible on-demand resource provisioning
- Increased efficiency through resource sharing
- Challenges
  - Increased system complexity and dynamics
  - Diverse vulnerabilities due to resource sharing
- **Service "dependability"** → major distinguishing factor between cloud platforms
  - Availability, reliability, performance,…
- Simple trigger/rule-based resource management mechanisms
  - Best-effort approach

Proactive Auto-Scaling

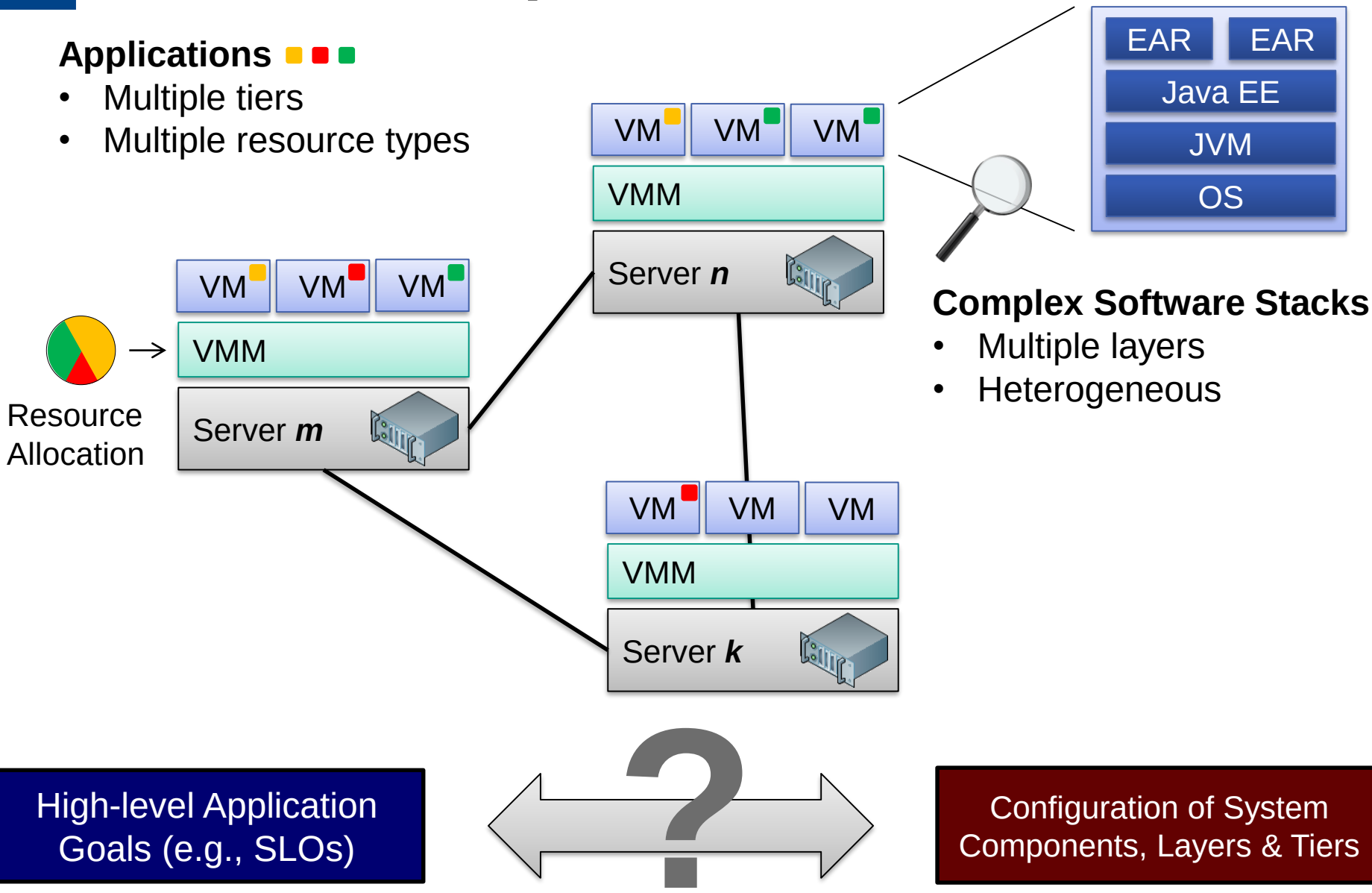


→ **Example Scenario for Self-Aware Computing** (more later)

Semantic Gap Problem

Applications

- Multiple tiers
- Multiple resource types



Complex Software Stacks

- Multiple layers
- Heterogeneous

High-level Application Goals (e.g., SLOs)

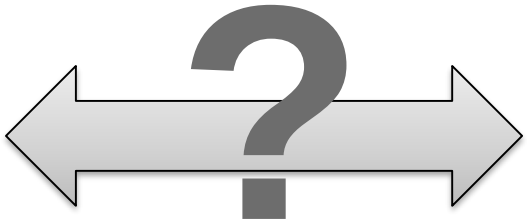
Configuration of System Components, Layers & Tiers

Semantic Gap Problem

- **Availability & Performance**
  - Services available 99.99% of the time
  - Response time of service x < 20 ms
  - Transaction throughput > 1000
  - Server utilization > 60% on average
  - „Time to recover after a failure“ < 1 min
- **Efficiency**
  - Allocate only as much resources as are actually needed
- ...

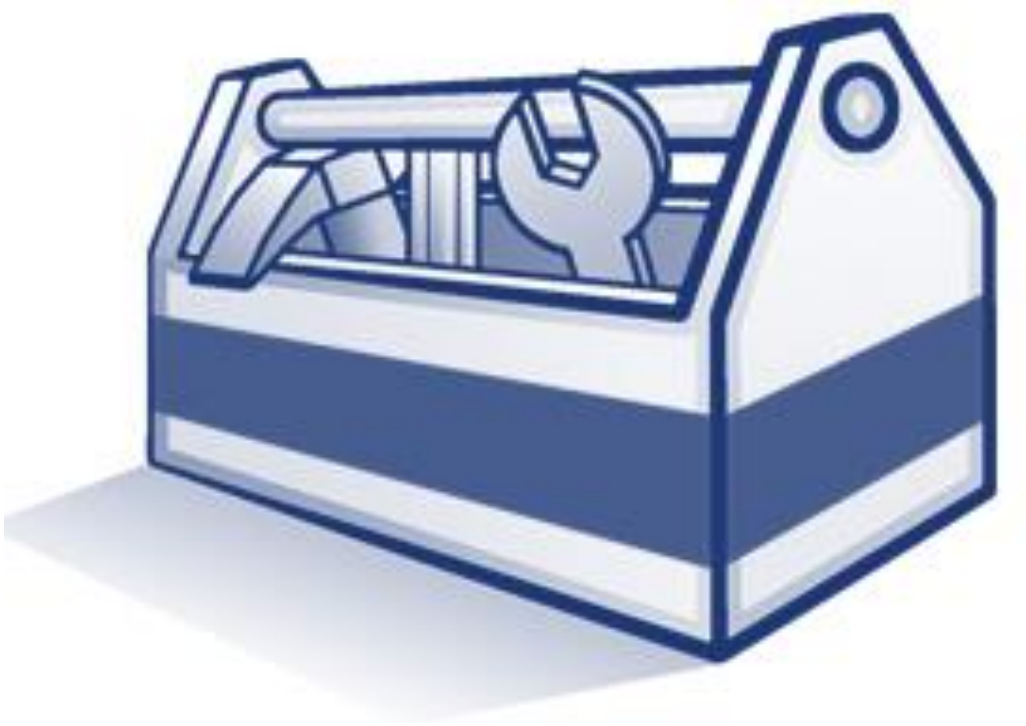
- How many vCPUs to allocate to virtual machine (VM) n?
- How much memory to allocate to VM n?
- When exactly should a reconfiguration be triggered?
- Which particular resources or services should be scaled / replicated / migrated / restarted?
- How quickly and at what granularity?

Service level objectives (SLOs)



Configuration of System Components, Layers & Tiers

Descartes Tool Chain



http://descartes.tools

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

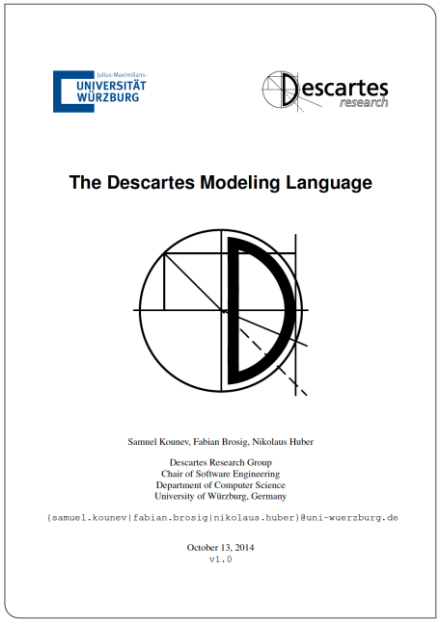
Mailing list available...

Descartes Tools: Links

- **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 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



The Descartes Modeling Language

Samuel Kounev, Fabian Brosig, Nikolaus Huber

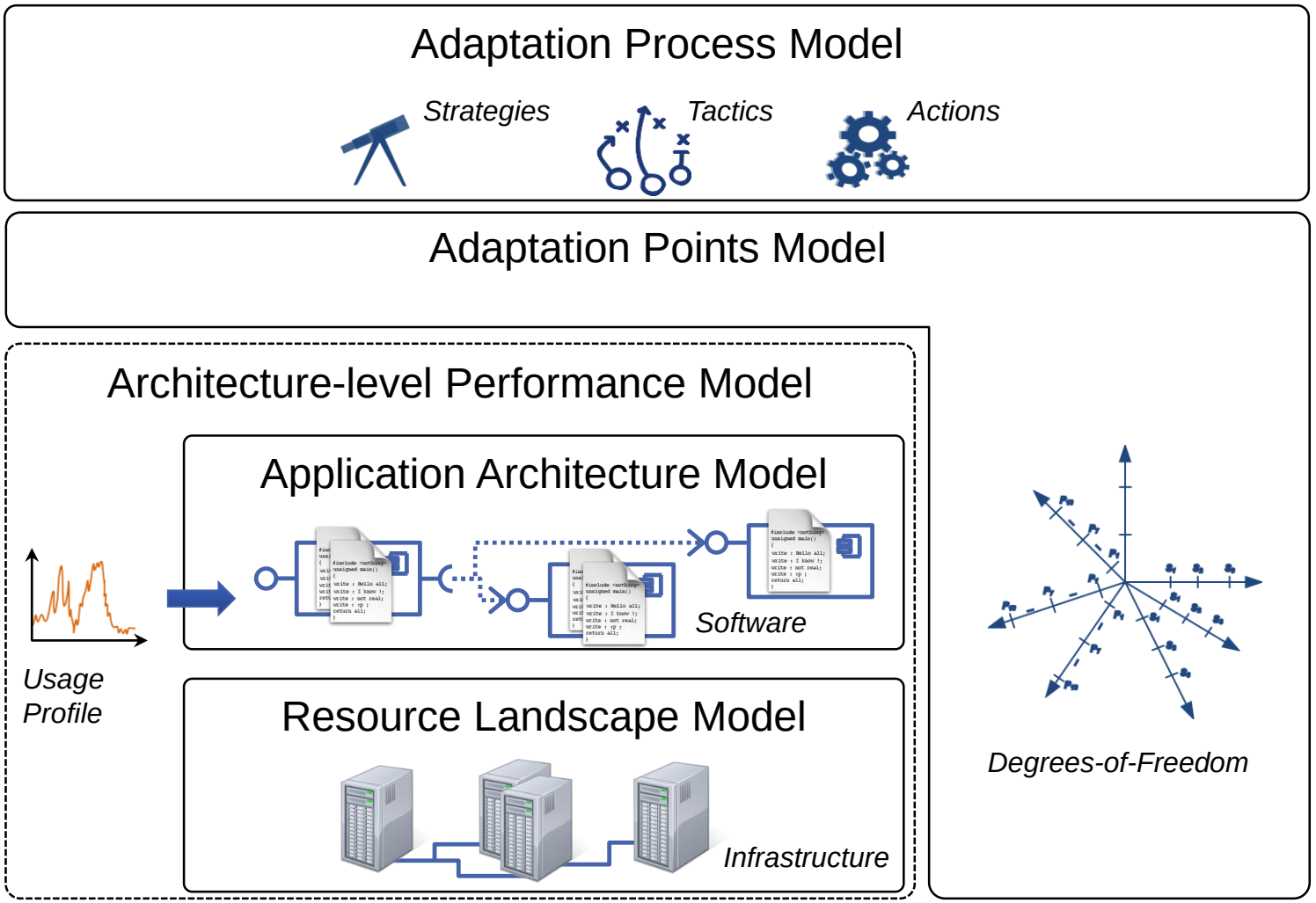
Descartes Research Group
Chair of Software Engineering
Department of Computer Science
University of Würzburg, Germany

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

October 13, 2014
v1.0

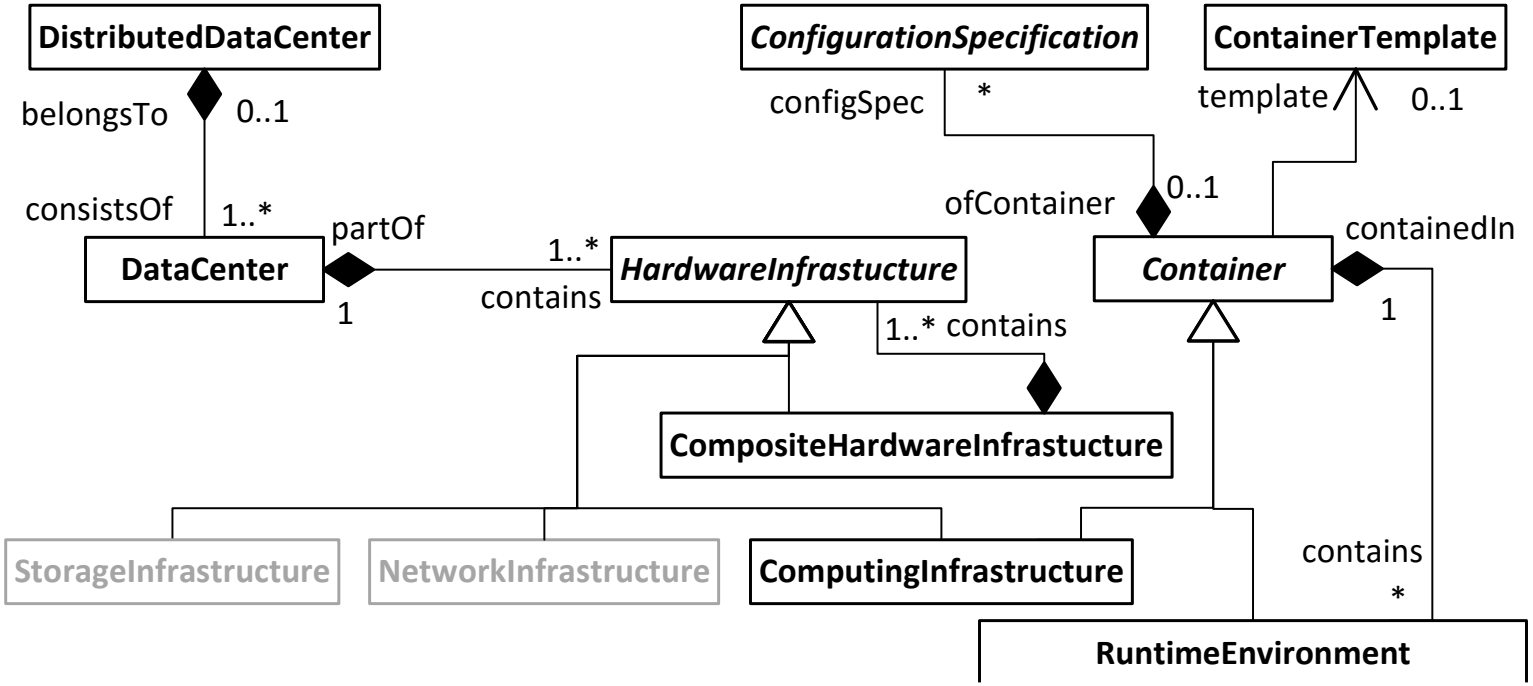
http://descartes.tools/dml

DML Sub-Models



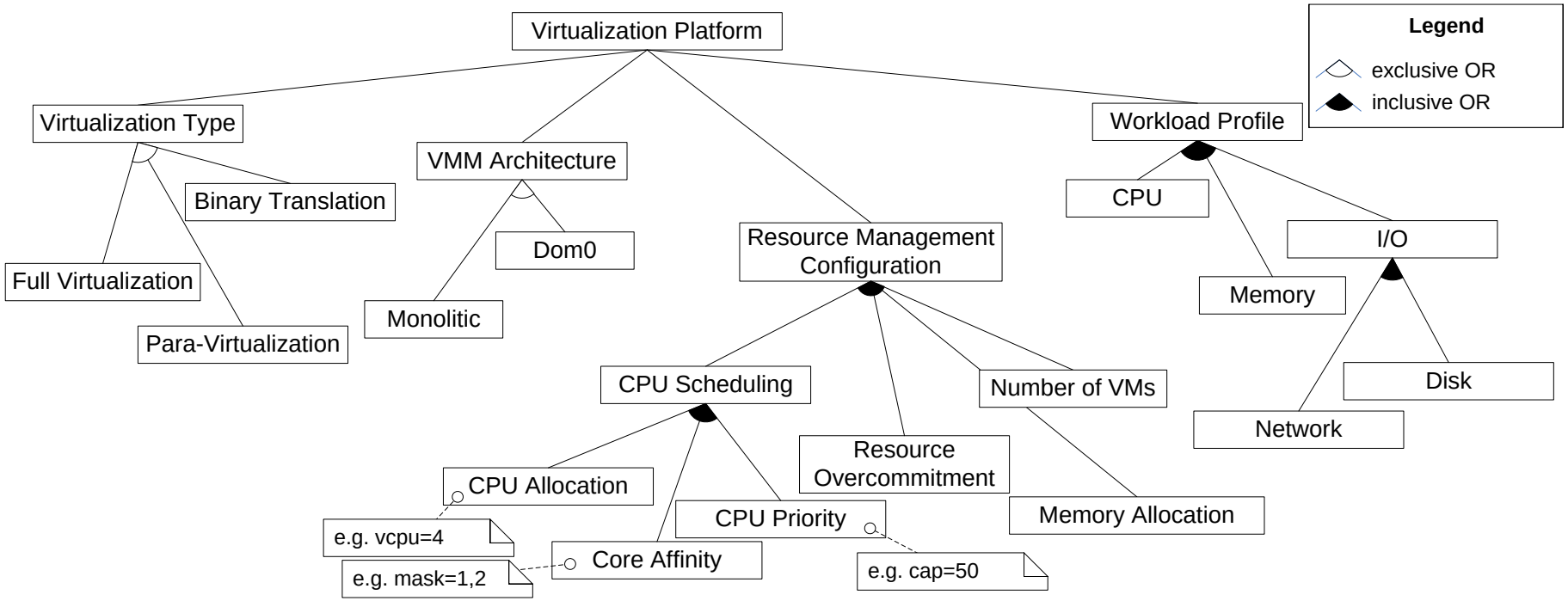
DML Implementation

- Implementation in Ecore (Eclipse Modeling Framework)
- Excerpt from meta-model: resource landscape



Example: Custom Configuration Model

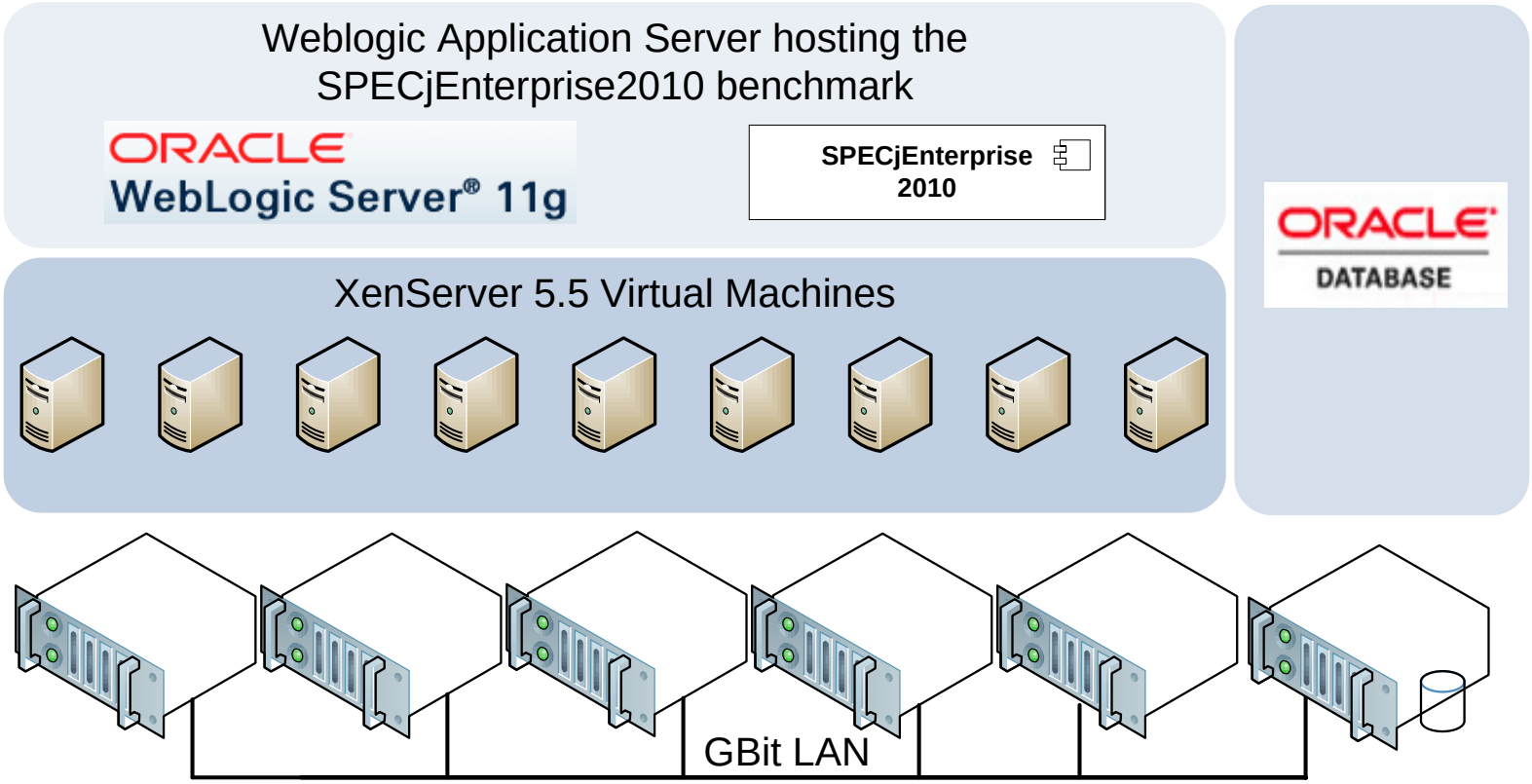
(Feature Model for the Virtualization Platform)



N. Huber, M. Quast, M. Hauck, and S. Kounev. **Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments**. *International Conference on Cloud Computing and Services Science (CLOSER 2011), Noordwijkerhout, The Netherlands*, May 7-9, 2011. Best Paper Award.

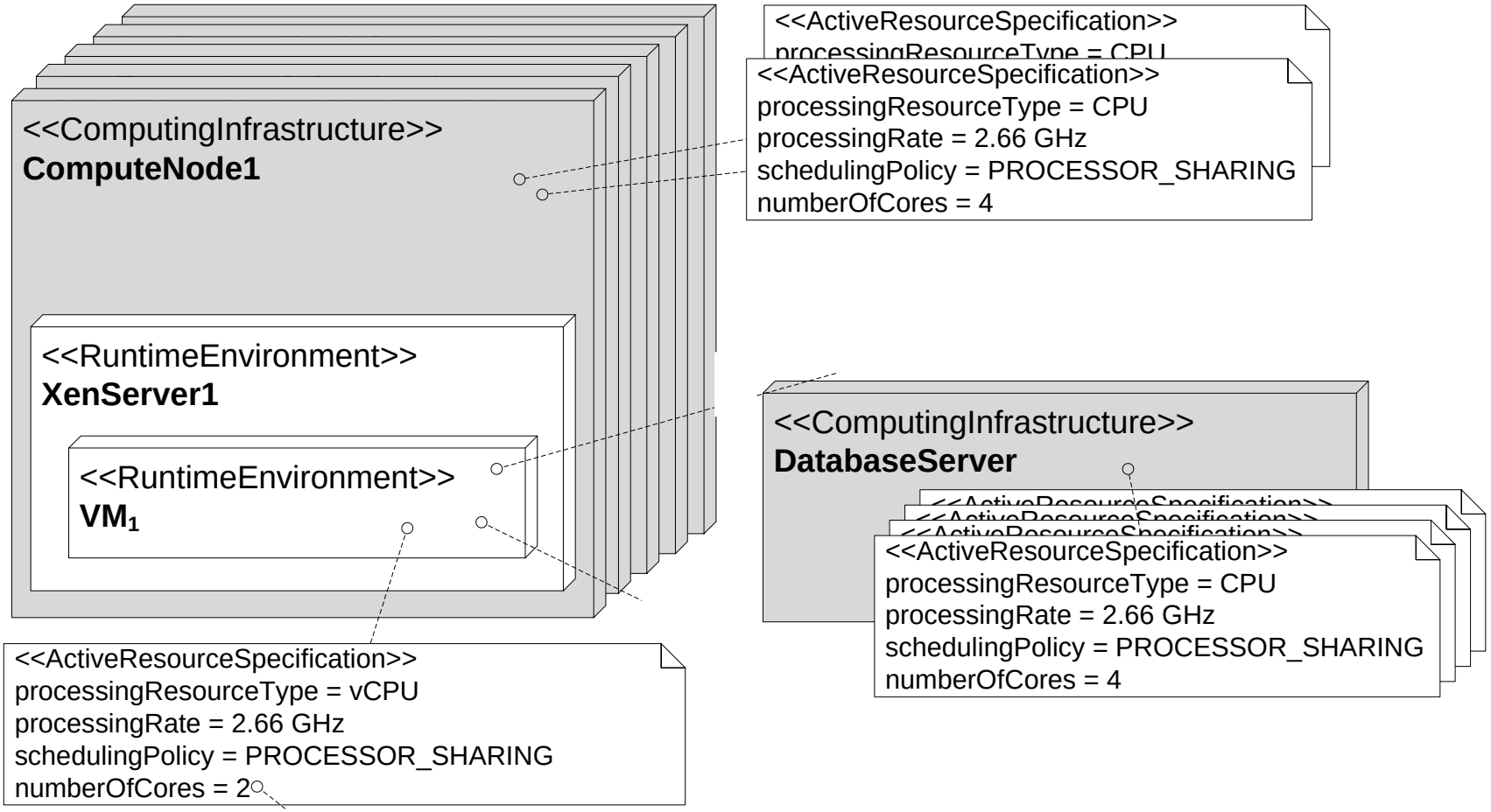
Example: WebLogic Server Cluster

(Resource Landscape)



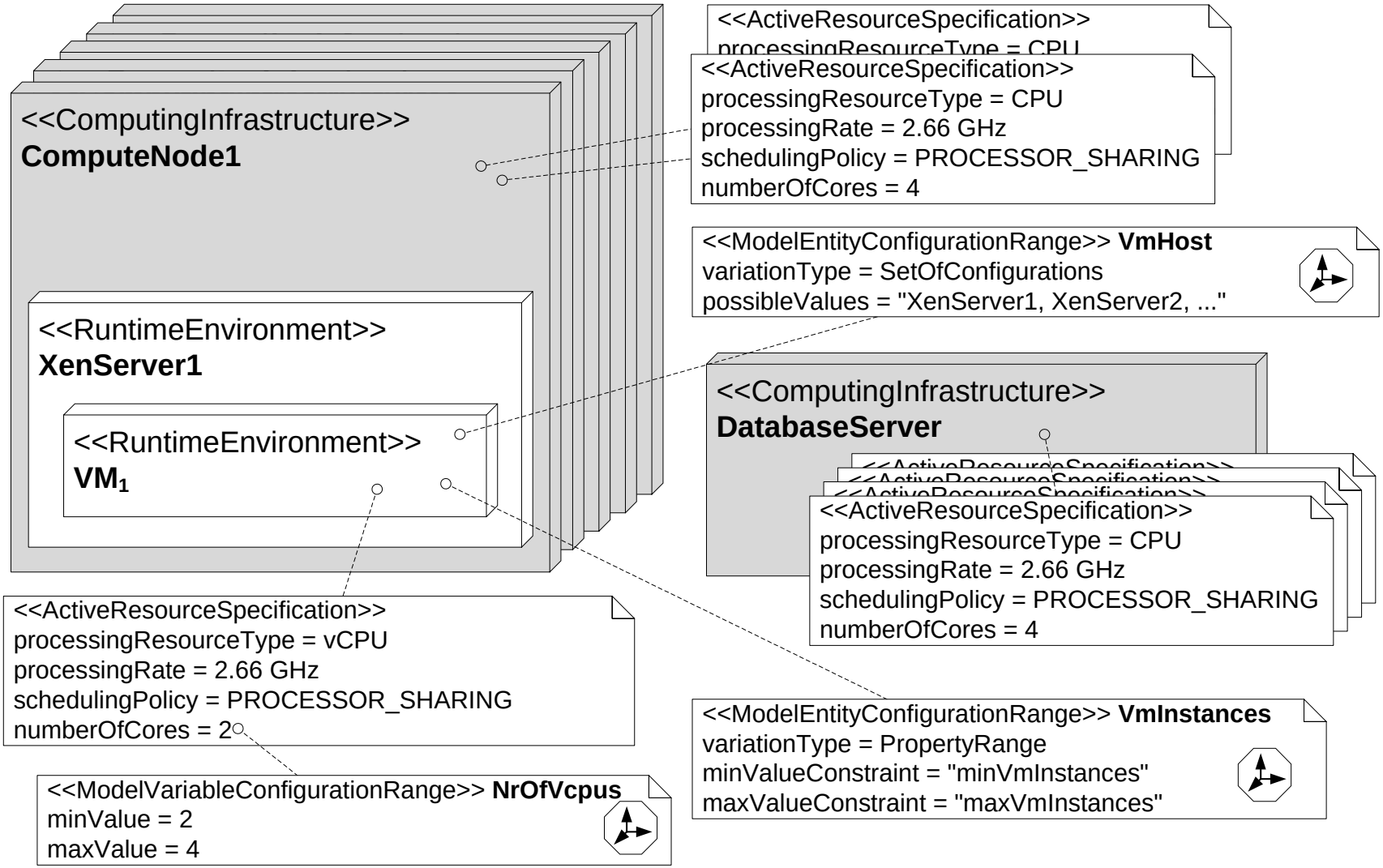
Example: WebLogic Server Cluster

(Resource Landscape Model)

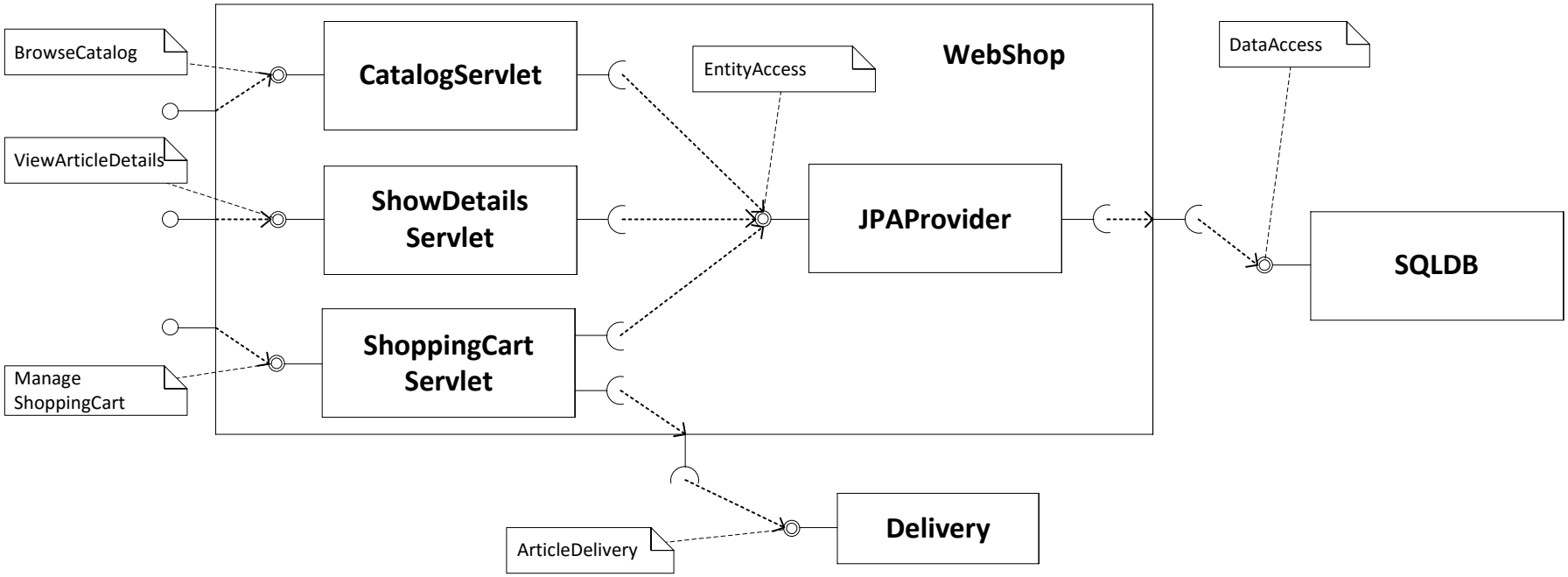


Example: WebLogic Server Cluster

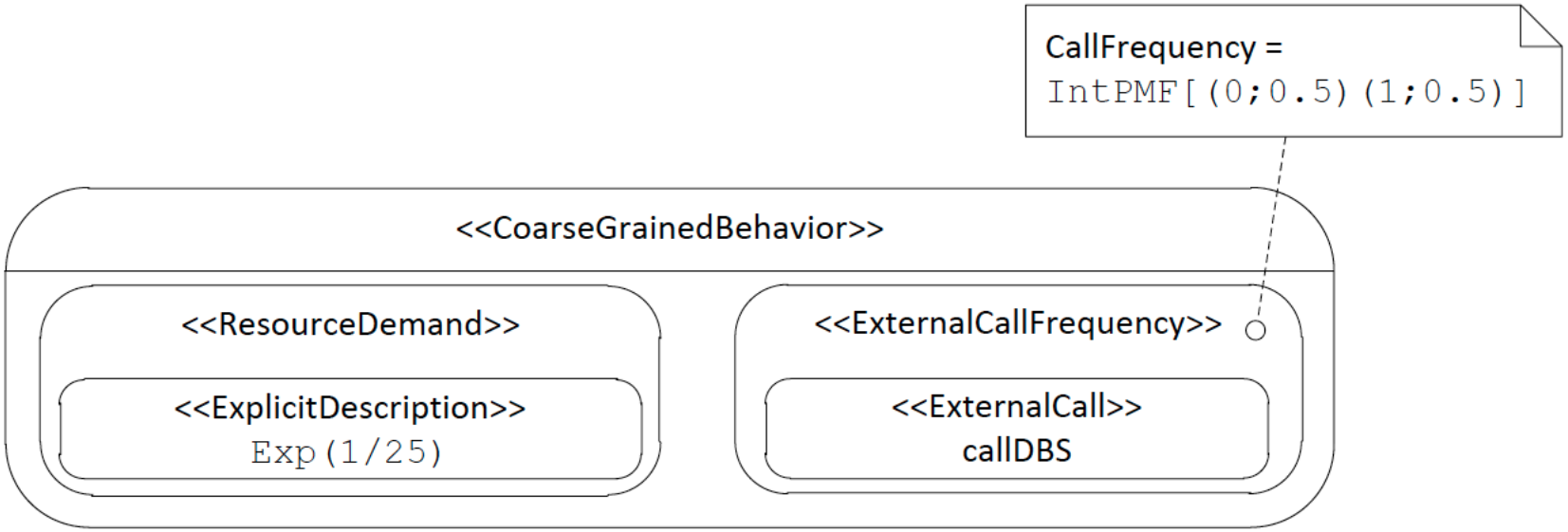
(Resource Landscape Model) + (Adaptation Points Model)



Example
(Application Architecture Model)

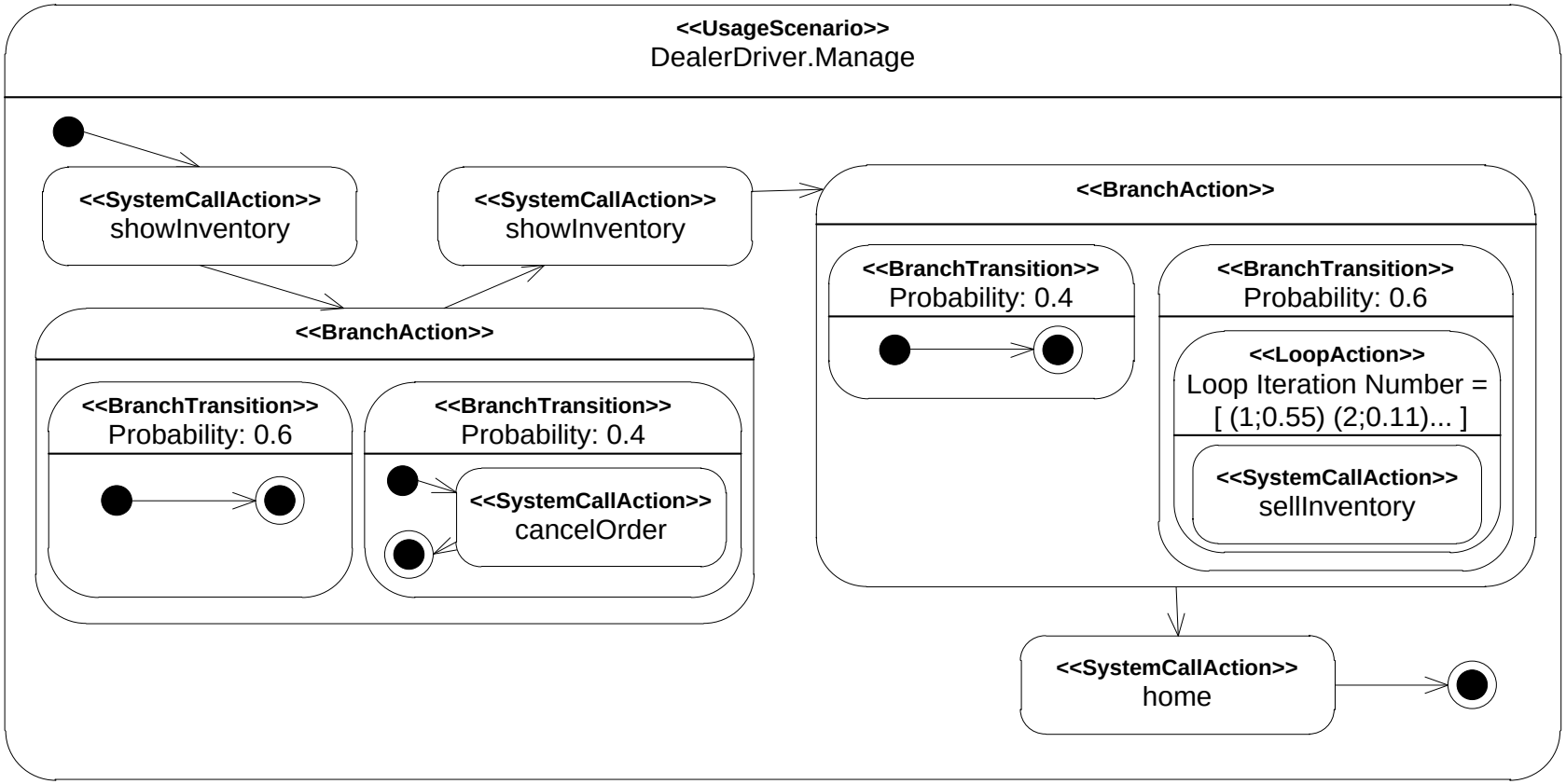


Example
(Coarse-Grained Service Behavior Model)

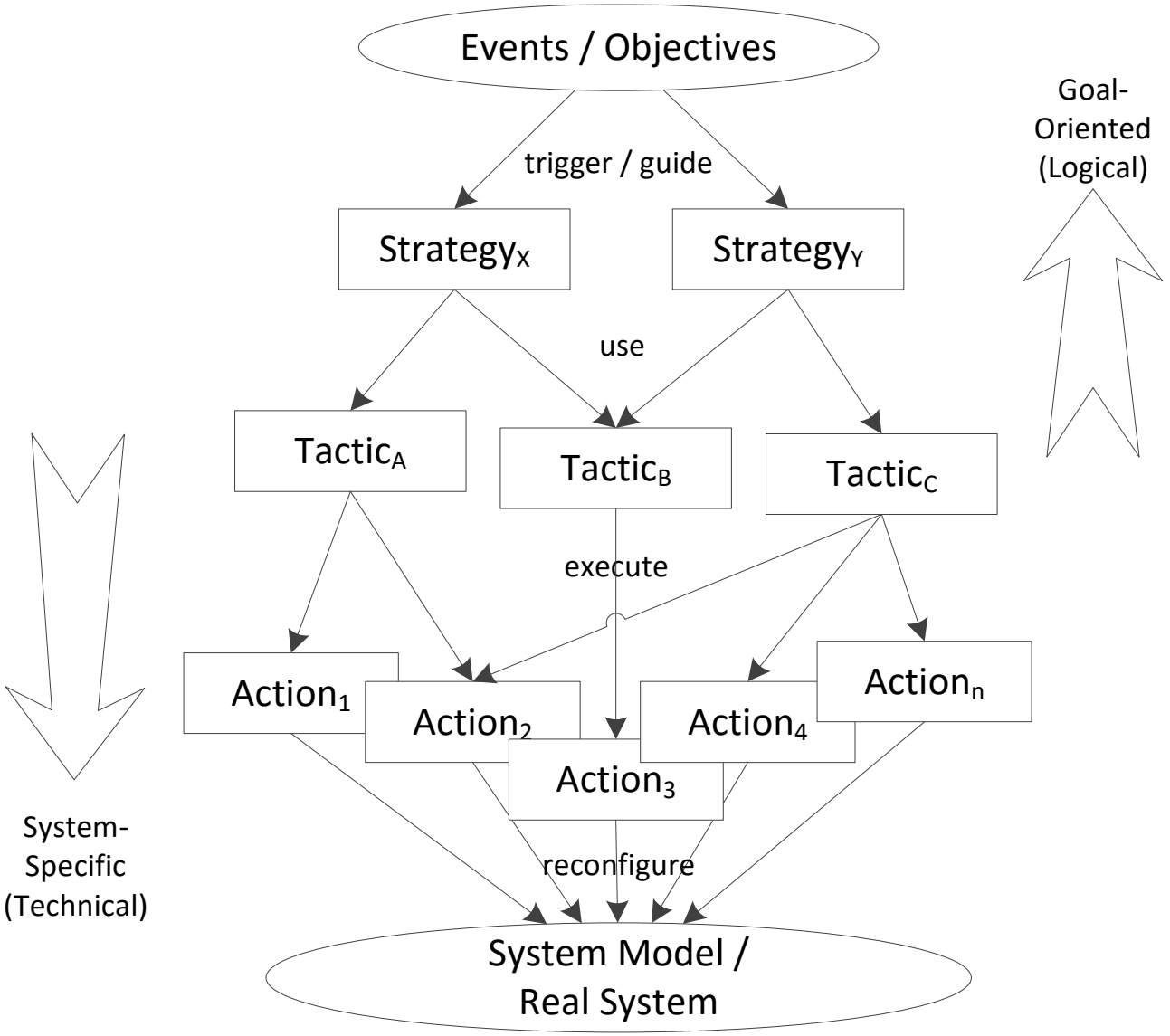


Example

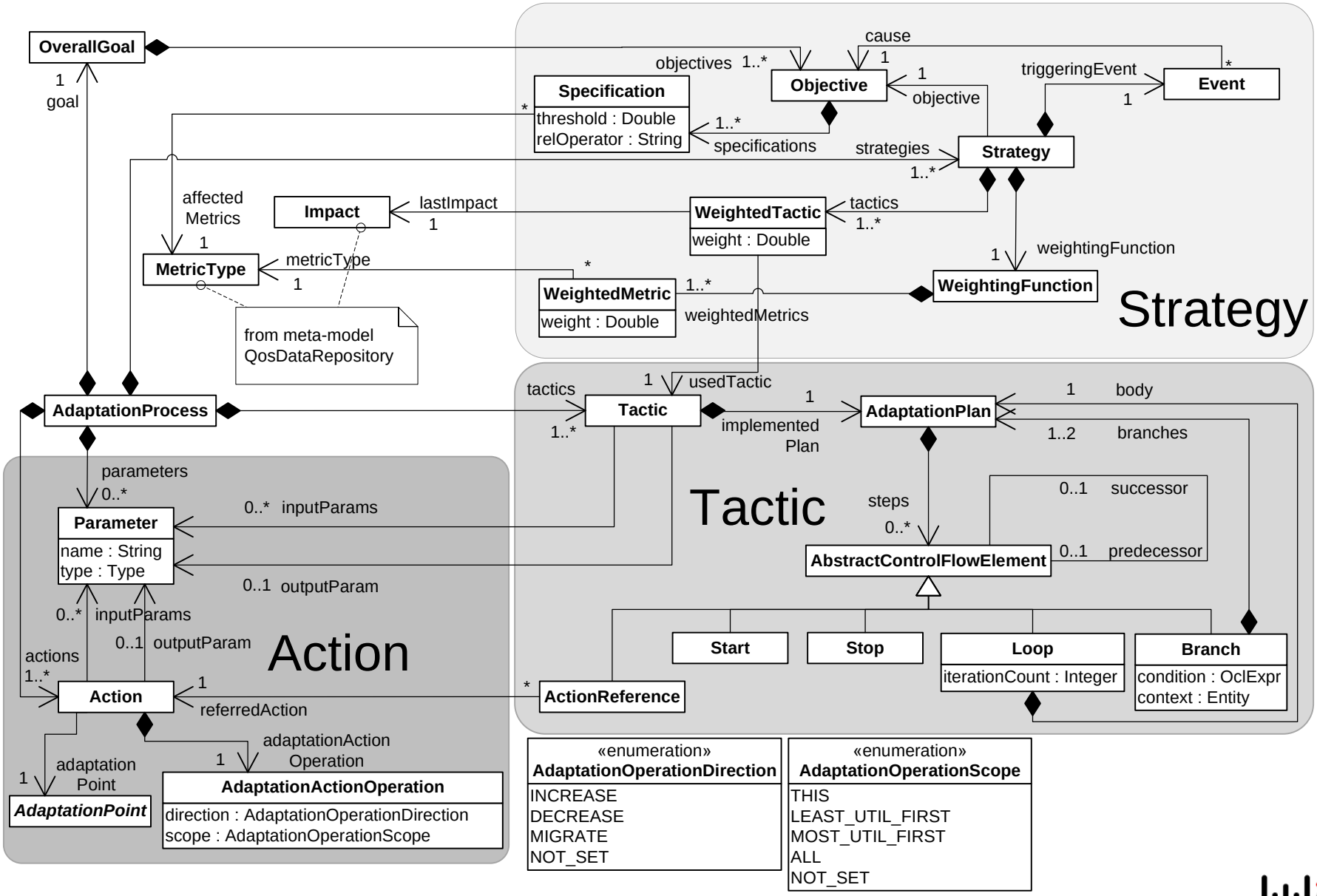
(Fine-Grained Service Behavior Model)



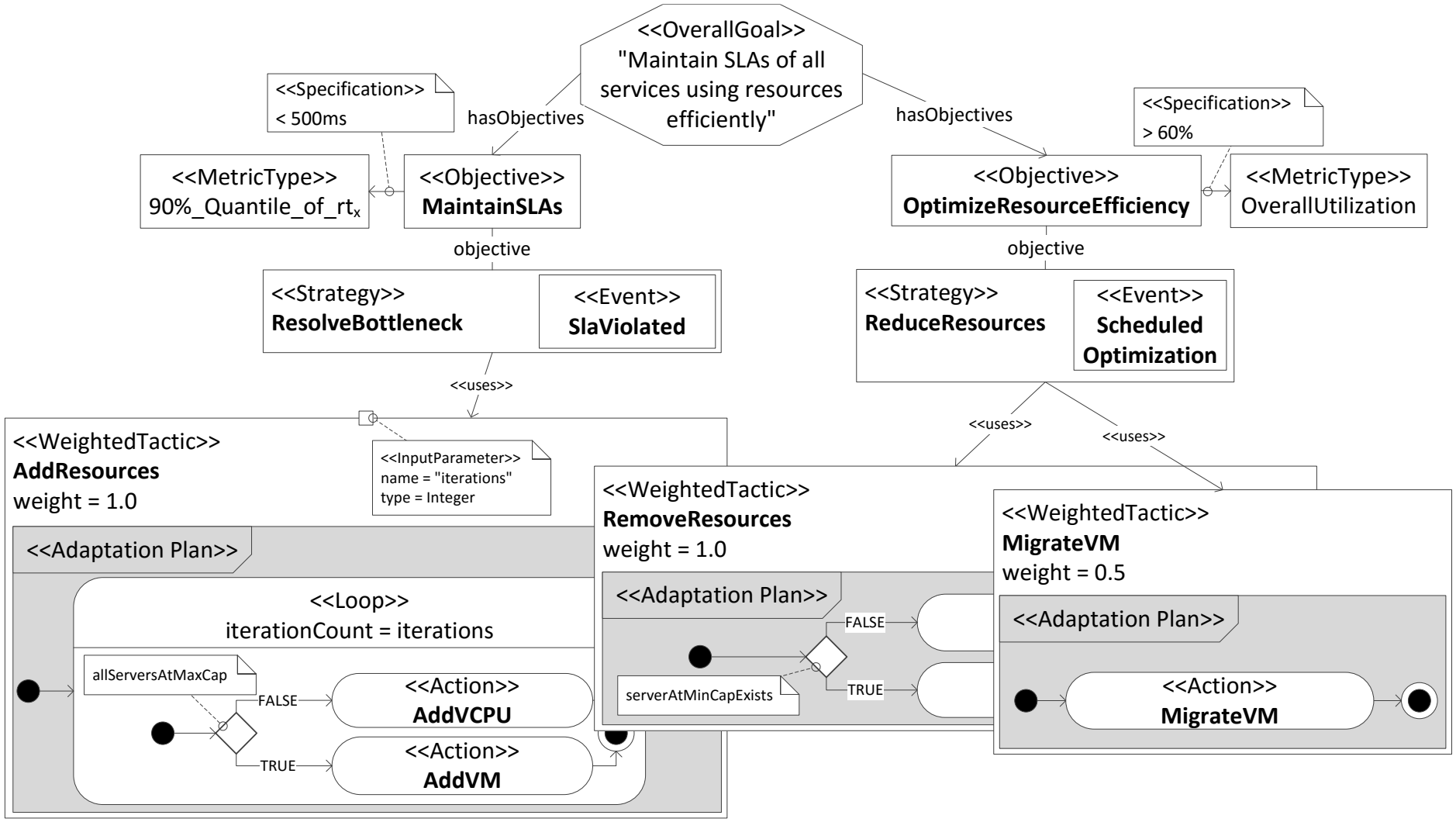
Adaptation Process Model



S/T/A Meta-Model (Strategies, Tactics and Actions)

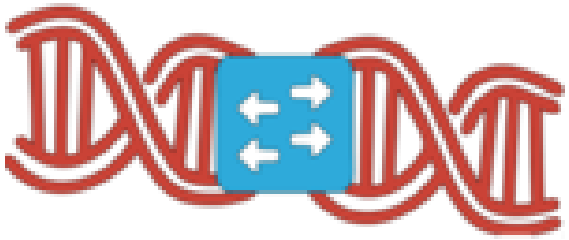


Example: Adaptation Process Model



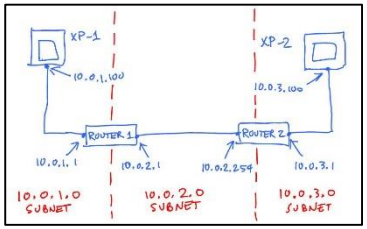
DNI - Descartes Network Infrastructure Modeling

- Language for perf. modeling of data center networks
  - network topology, switches, routers, virtual machines, network protocols, routes, flow-based configuration,...
- Model solvers based on simulation (OMNeT)

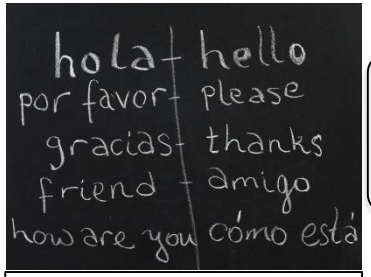


http://descartes.tools/dni

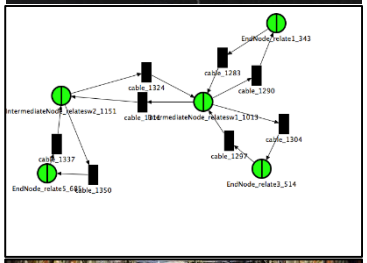
Flexible Modeling of Data Center Networks for Capacity Management



DNI Meta-Model
Generic modeling formalism for SDN- and NFV-based data center networks performance.



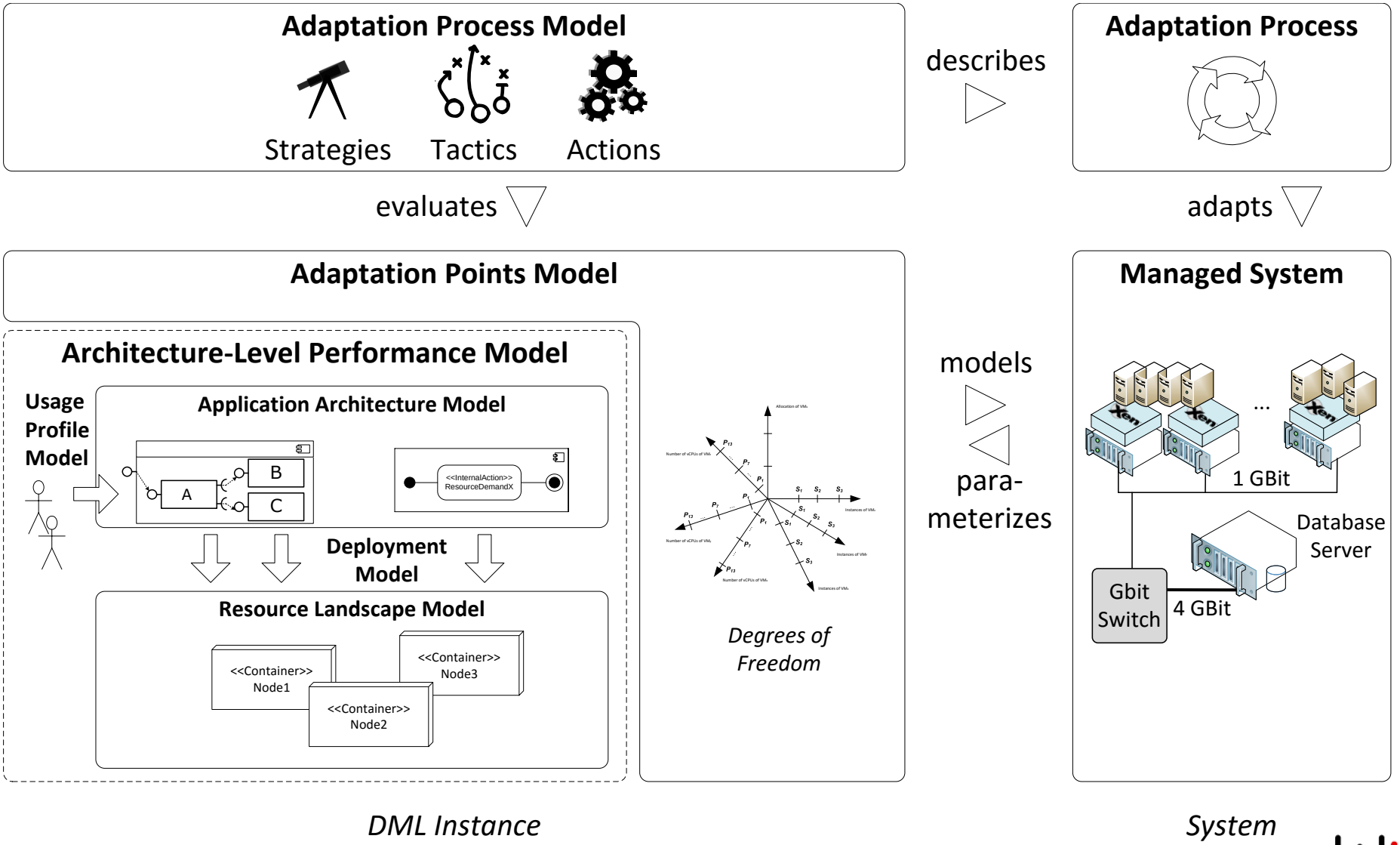
Model Transformations x6
Automated transformations to different predictive models.



Model Solvers ≤10
Solvers supporting trade-offs btw. accuracy and solving time.

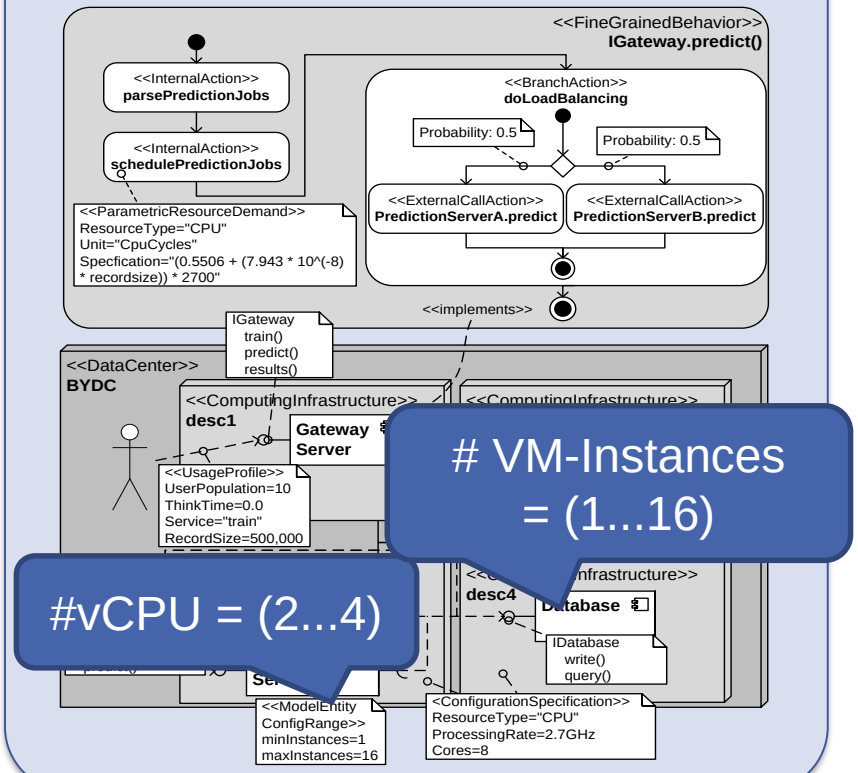
Model Extraction
Traffic models can be extracted automatically from traces.

Big Picture


Adaptation Process Model
Strategies
Tactics
Actions
describes
Adaptation Process
Logical
evaluates
adapts
Adaptation Points Model
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
Managed System
1 GBit
Database Server
Gbit Switch
4 GBit
Technical
DML Instance
System

Online Performance Prediction

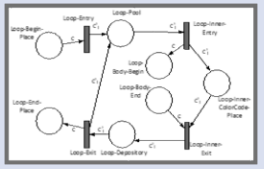
Architecture-Level Performance Model

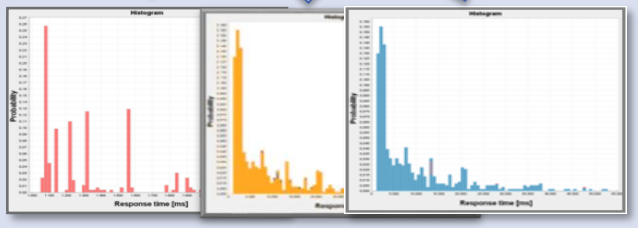


Online Performance Prediction

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

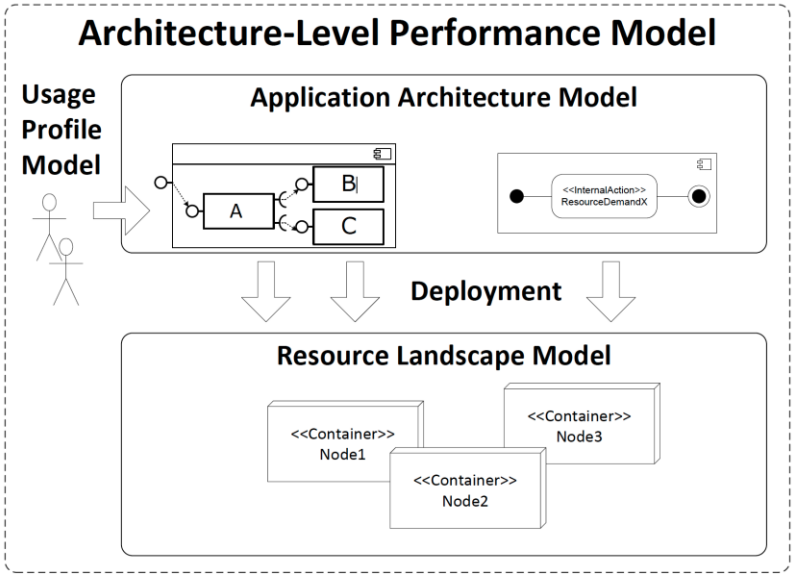
$$\overline{R} = \frac{N}{\overline{X}} \geq max\left\{\sum_{i=0}^{n} D_i^{sync}, N * max_{1\leq i\leq n}\{D_i\}\right\}$$



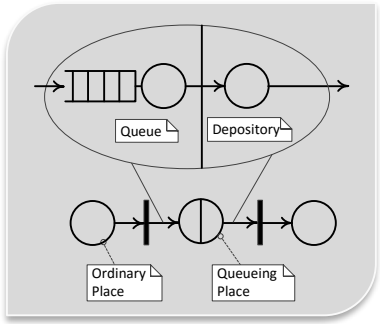


Autonomic Decision Making

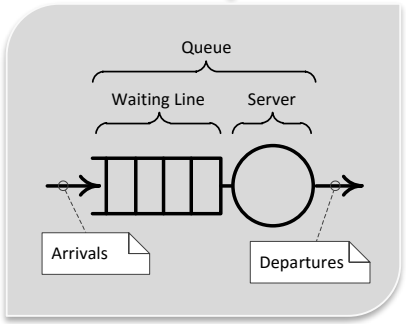
Transformations to Predictive Models



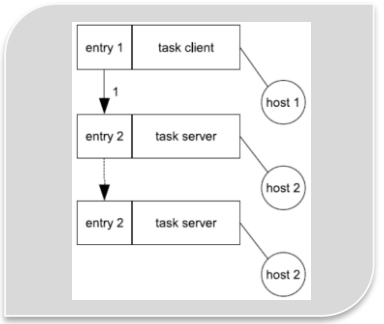
DML Instance



Queueing Petri Net

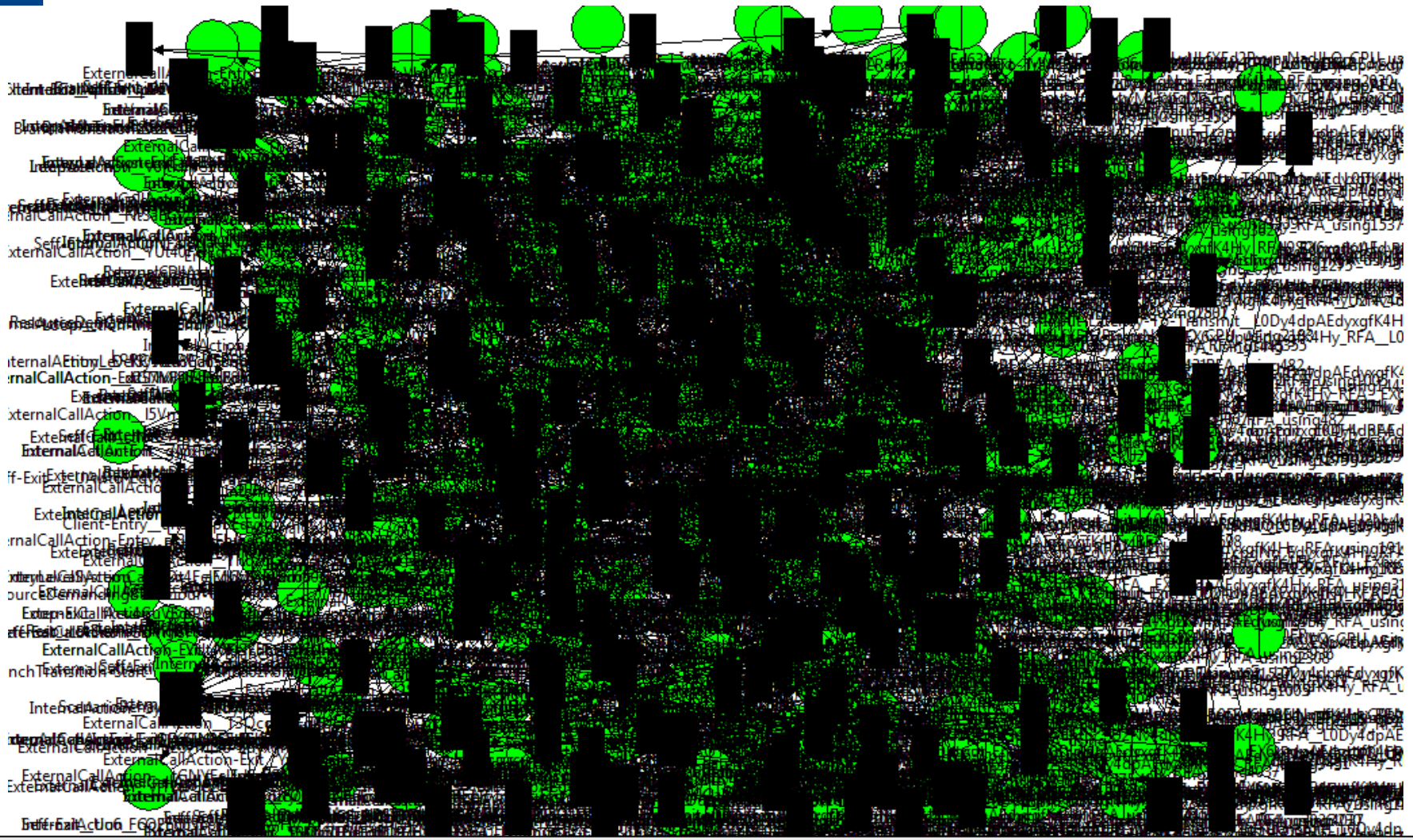


Bounds Analysis Model



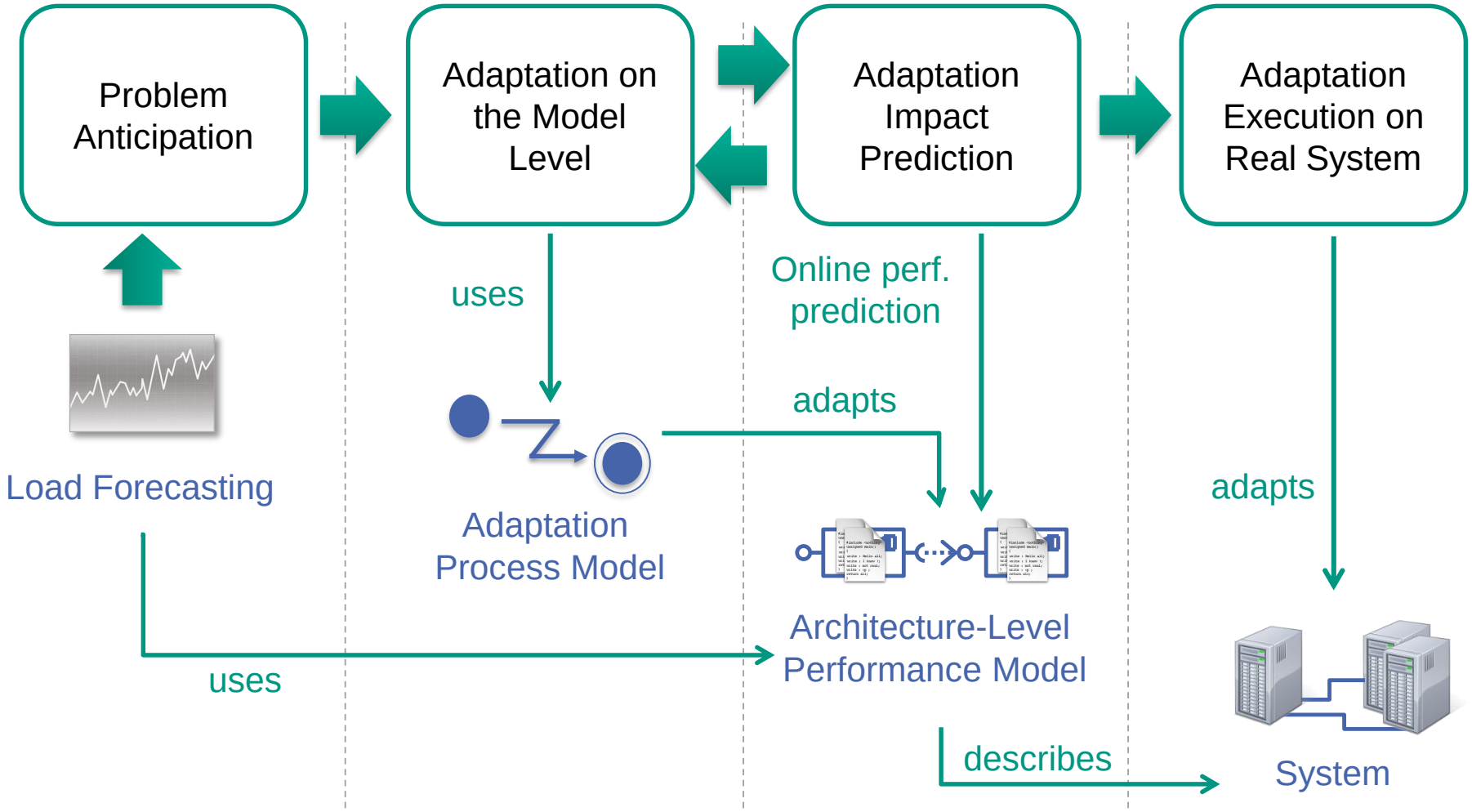
Layered Queueing Network

Case Study: Process Control System (ABB)



P. Meier, S. Kounev, and H. Koziolek. **Automated transformation of component-based software architecture models to queueing petri nets**. In *19th IEEE/ACM Intl. Symp. on Modeling, Analysis and Simulation of Computer and Telecomm. Systems (MASCOTS), Singapore, July 25-27*, 2011. [.pdf]

Model-Based System Adaptation



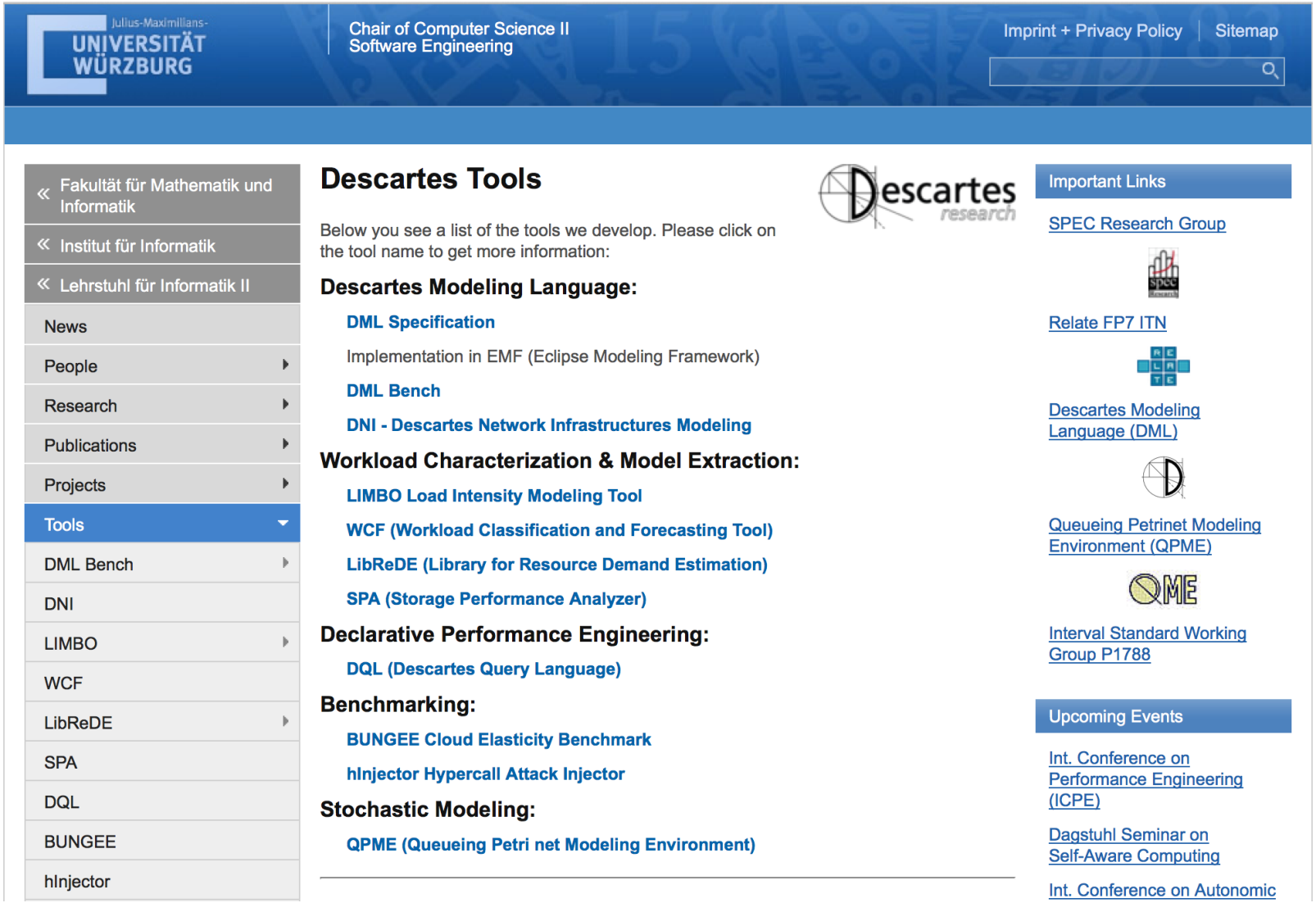
Latest Publications on DML

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.

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING

http://descartes.tools



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

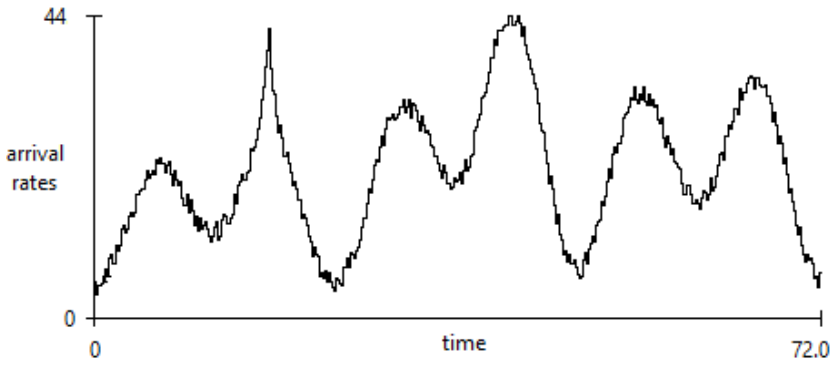
http://descartes.tools

Mailing list available...

LIMBO Tool

- Problem:
  - How to capture the load intensity variations (e.g., requests per sec) in a compact mathematical model?
  - How to forecast the load intensity (requests per sec) in future time horizons?

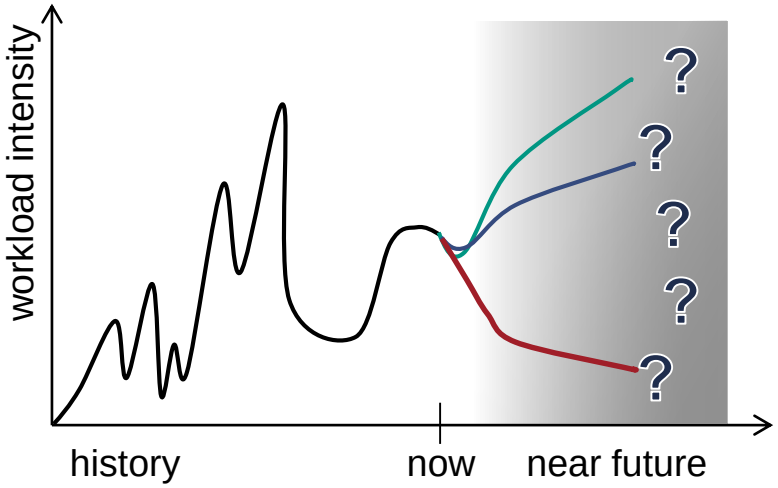
- **Load Intensity Modeling & Forecasting Tool**



http://descartes.tools/limbo

LIMBO Tool (2)

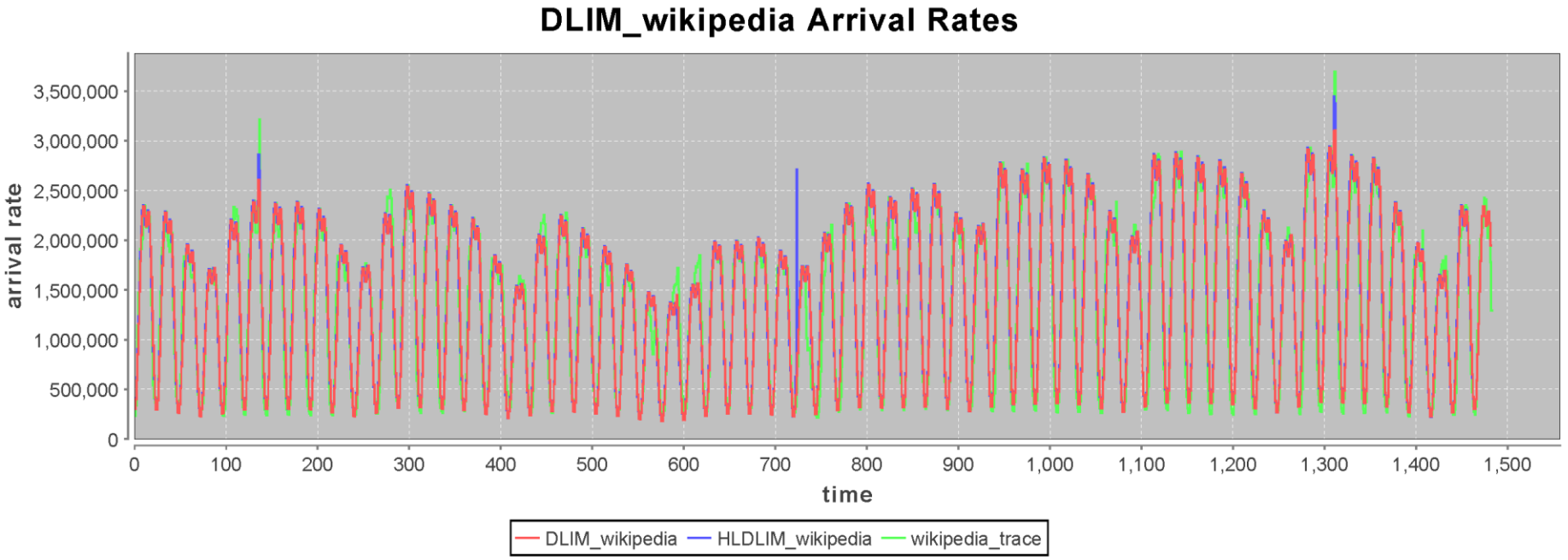
- **Workload Classification & Forecasting (WCF)**
  - Use of multiple alternative forecasting methods in parallel
  - Selection of method based on its accuracy in the past



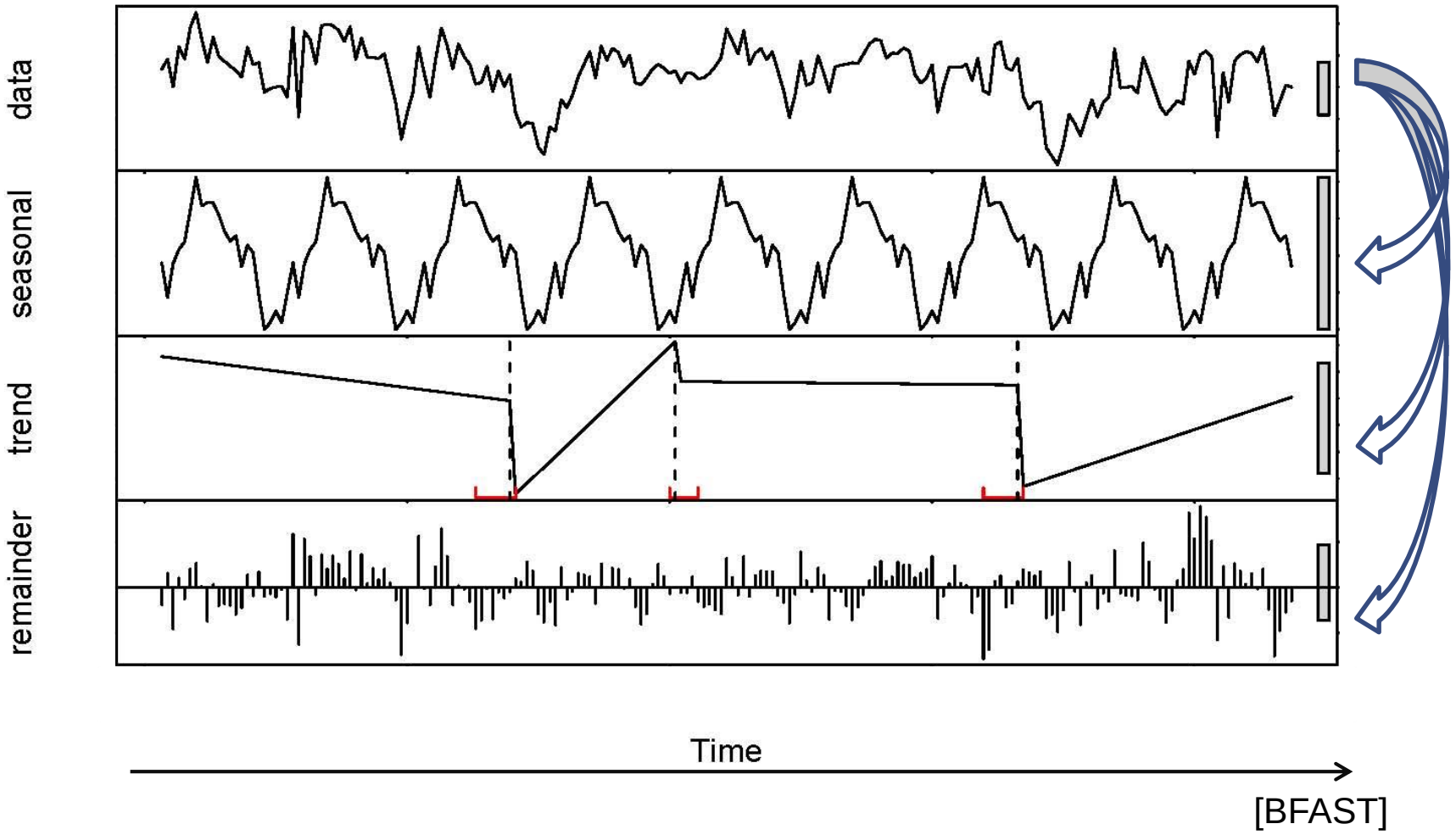
http://descartes.tools/libmo
http://descartes.tools/wcf

IBM PHD FELLOWSHIP AWARD PROGRAMME

Example: Wikipedia Workload



Time Series Analysis



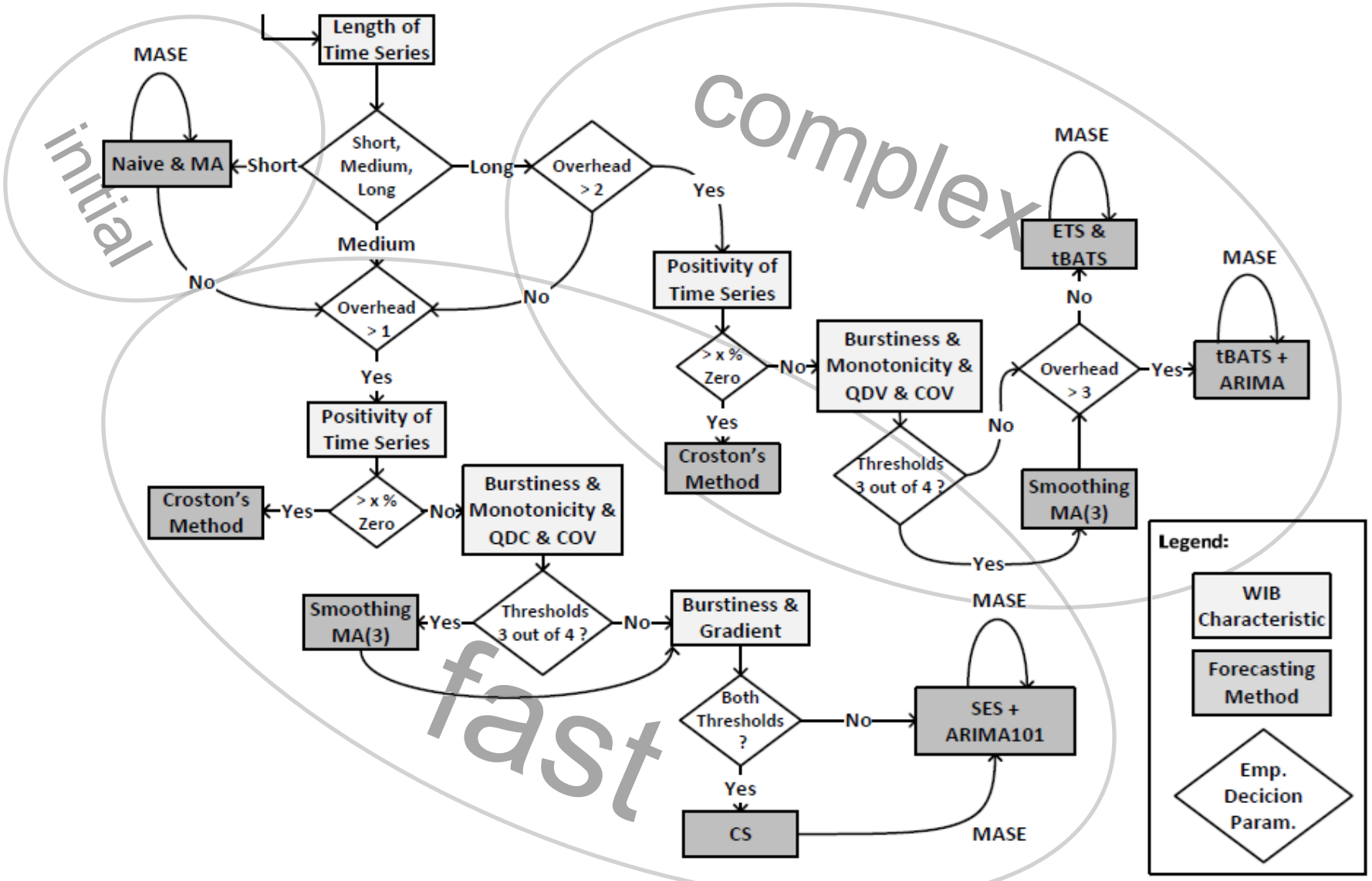
Applied Forecasting Methods

Basic Methods	**(initial)**
Naïve, Moving Averages, Random Walk	

Trend Interpolation	**(fast)**
Simple Exponential Smoothing (SES)	[Hynd08]
Cubic Smoothing Splines	[Hynd02]
Croston‘s method for intermittent time series	[Shen05]
Autoregressive Moving Averages (ARMA11)	[Box08]

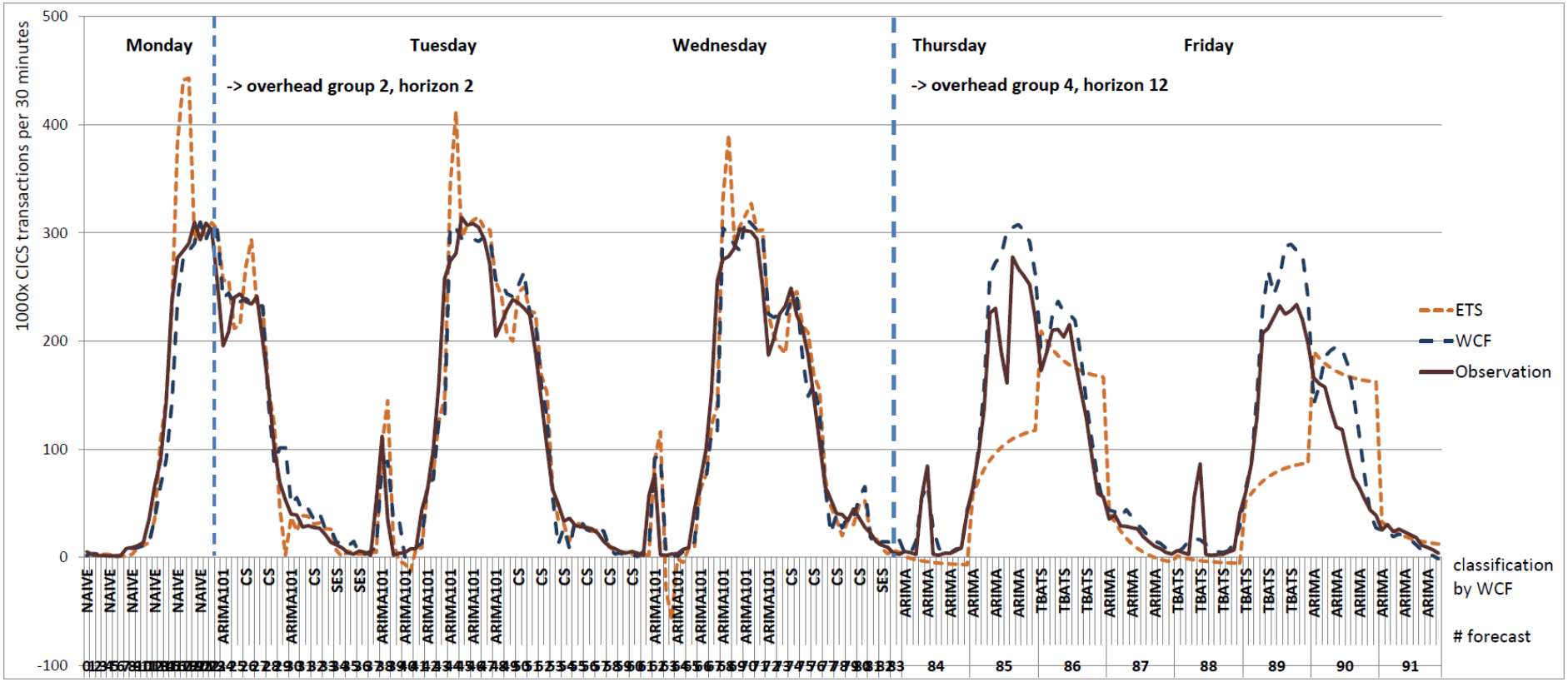
Estimation and Modelling of Seasonal Pattern	**(complex)**
Extended Exponential Smoothing (ETS)	[Hynd08, Hyn08]
ARIMA framework with automatic model selection	[Box08, Hynd08]
tBATS for complex seasonal patterns	[Live11]

LIMBO: Auswahl der Methode



Evaluation

- Real-world workload intensity trace: IBM CICS transactions on System z
- Comparison of **WCF** approach to **ETS** and **Naïve** forecast



LIMBO Tool

http://descartes.tools/limbo

J. von Kistowski, N. Herbst, and S. Kounev. **LIMBO: A Tool For Modeling Variable Load Intensities (Demo Paper)**. In *5th ACM/SPEC International Conference on Performance Engineering (ICPE 2014)*, Dublin, Ireland, March 22-26, 2014, ICPE '14, pages 225-226. ACM, New York, NY, USA. March 2014. [DOI | slides | http | .pdf]

J. von Kistowski, N. Herbst, and S. Kounev. **Modeling Variations in Load Intensity over Time**. In *3rd Intl. Workshop on Large-Scale Testing (LT 2014)*, Dublin, Ireland, March 22, 2014, pages 1-4. ACM, New York, NY, USA. March 2014. [DOI | slides | http | .pdf]

LibReDE Tool

- Problem: How to estimate the total service time of a given type of request/job at a given resource?

- **Library for Resource Demand Estimation**
  - Ready-to-use implementations of estimation approaches
  - Selection of a suitable approach for a given scenario

http://descartes.tools/librede

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]

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...

Systems Benchmarking

Metrics and benchmarks for quantitative evaluation of

1. Cloud elasticity
2. Performance isolation
3. Intrusion detection (and prevention)
4. ...

S. Kounev. **Quantitative Evaluation of Service Dependability in Shared Execution Environments** (Keynote Talk). In 11th Intl. Conf. on Quantitative Evaluation of SysTems (QEST 2014), Florence, Italy, September 8-12, 2014. [slides | extended abstract]

[geek & poke]

Need for Benchmarks

"*To **measure** is to **know**.*" -- Clerk Maxwell, 1831-1879

"*It is much easier to make **measurements** than to **know** exactly what you are measuring*." -- J.W.N.Sullivan (1928)

1. Reliable Metrics

- What exactly should be measured and computed?

2. Representative Workloads

- For which scenarios and under which conditions?

3. Sound Measurement Methodology

- How should measurements be conducted?

Cloud Elasticity

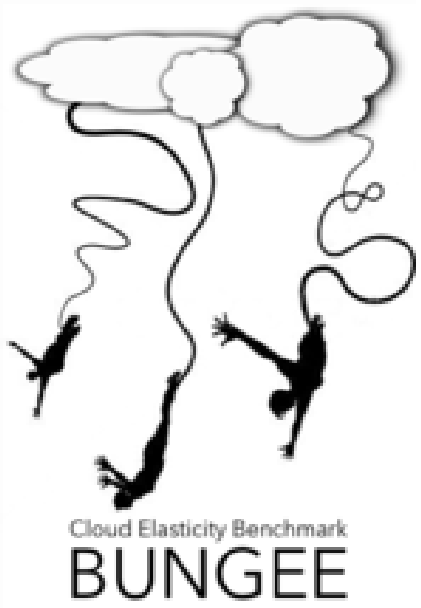
Def: The degree to which a system is able to **adapt** to **workload changes** by **provisioning and deprovisioning** resources in an **autonomic manner**, such that at each point in time the **available resources match** the **current demand** as closely as possible.

N. Herbst, S. Kounev and R. Reussner
Elasticity in Cloud Computing: What it is, and What it is Not.
in Proceedings of the 10th International Conference on Autonomic Computing (ICAC 2013), San Jose, CA, June 24-28, 2013.
[slides | http | .pdf]

http://en.wikipedia.org/wiki/Elasticity_(cloud_computing)

BUNGEE Tool

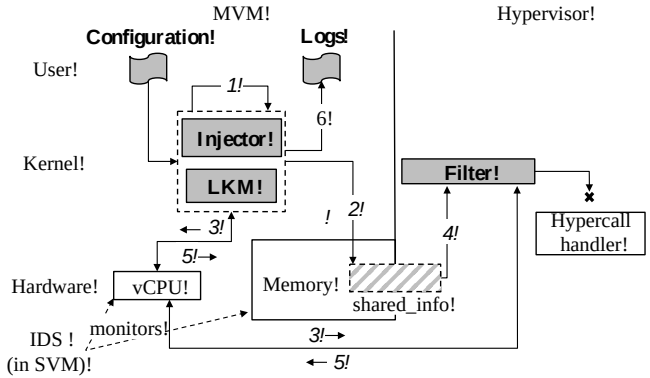
- Problem: How to measure and quantify cloud elasticity?
- Framework for benchmarking elasticity
  - Current focus: IaaS cloud platforms



http://descartes.tools/bungee

Security Research

- Cooperation with
  - University of Coimbra
  - Netflix, Inc.
  - Siemens Corporate Research
  - ERNW



- Security of virtualized infrastructures
  - Vulnerability analysis
  - Evaluation of security mechanisms
  - Intrusion detection techniques

NETFLIX

ERNW providing security.

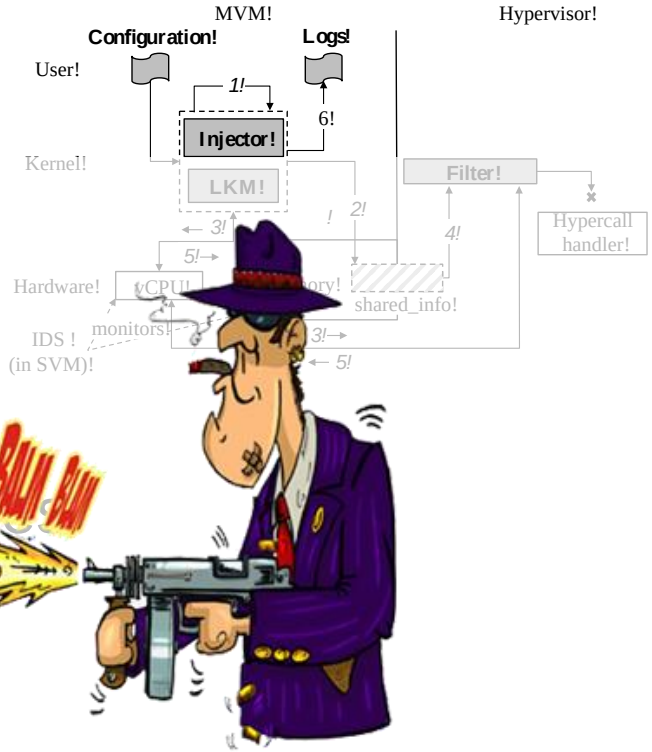
SIEMENS

Academic Partner

U C

Security Research

- Cooperation with
  - University of Coimbra
  - Netfli
  - Siem
  - ERN
- Securi
  - Vulne
  - Evalu
  - Intrus



NETFLIX

ERNW providing security.

SIEMENS

Academic Partner

U C

Security Research

- Cooperation with IBM Research
- Self-protection of critical infrastructures
  - Model-driven self-adaptation
  - Active response to threats and attacks

Academic Partner

Carnegie Mellon University

Security Research

- Active members of the Cloud Security Alliance (CSA)
- Security of production Cloud environments
  - Risk analysis, assessment, and mitigation
  - Security standardization efforts

cloud CSA security alliance SM

RG IDS: Example Results

Aleksandar M., Bryan D. P., Nuno A., Marco V., and Samuel K. Experience Report: An Analysis of Hypercall Handler Vulnerabilities @ The 25th IEEE International Symposium on Software Reliability Engineering (ISSRE 2014) - Research Track, Italy, 2014.

Aleksandar M., Bryan D. P., Nuno A., Marco V., and Samuel K. hInjector: Injecting Hypercall Attacks for Evaluating VMI-based Intrusion Detection Systems (Poster) @ The 2013 Annual Computer Security Applications Conference (ACSAC 2013), USA, 2013.

hInjector available @ https://github.com/hinj/hInjector

Aleksandar M., Marco V., Samuel K., Alberto A., Bryan D. P. Evaluating Computer Intrusion Detection Systems: A Survey of Common Practices @ *ACM Computing Surveys*. Accepted for publication.

Collaboration with industry

Standard Performance Evaluation Corporation

SPEC Research Group

- **Open-Systems-Group (OSG)**
  - Processor and computer architectures
  - Virtualization platforms
  - Java (JVM, Java EE)
  - Message-based systems
  - Storage systems (SFS)
  - Web-, email- and file server
  - SIP server (VoIP)
  - Cloud computing
- **High-Performance-Group (HPG)**
  - Symmetric multiprocessor systems
  - Workstation clusters
  - Parallel and distributed systems
  - Vector (parallel) supercomputers
- **“Graphics and Workstation Performance Group” (GWPG)**
  - CAD/CAM, visualization
  - OpenGL

http://www.spec.org

SPEC Research Group (RG)

- Founded in March 2011: http://research.spec.org
  - Transfer of knowledge btw. academia and industry
- Activities
  - Methods and techniques for experimental system analysis
  - Standard metrics and measurement methodologies
  - Benchmarking and certification
  - Evaluation of academic research results
- Member organizations (Feb 2014)

SPEC RG Members (2015)

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

Dagstuhl-Seminar

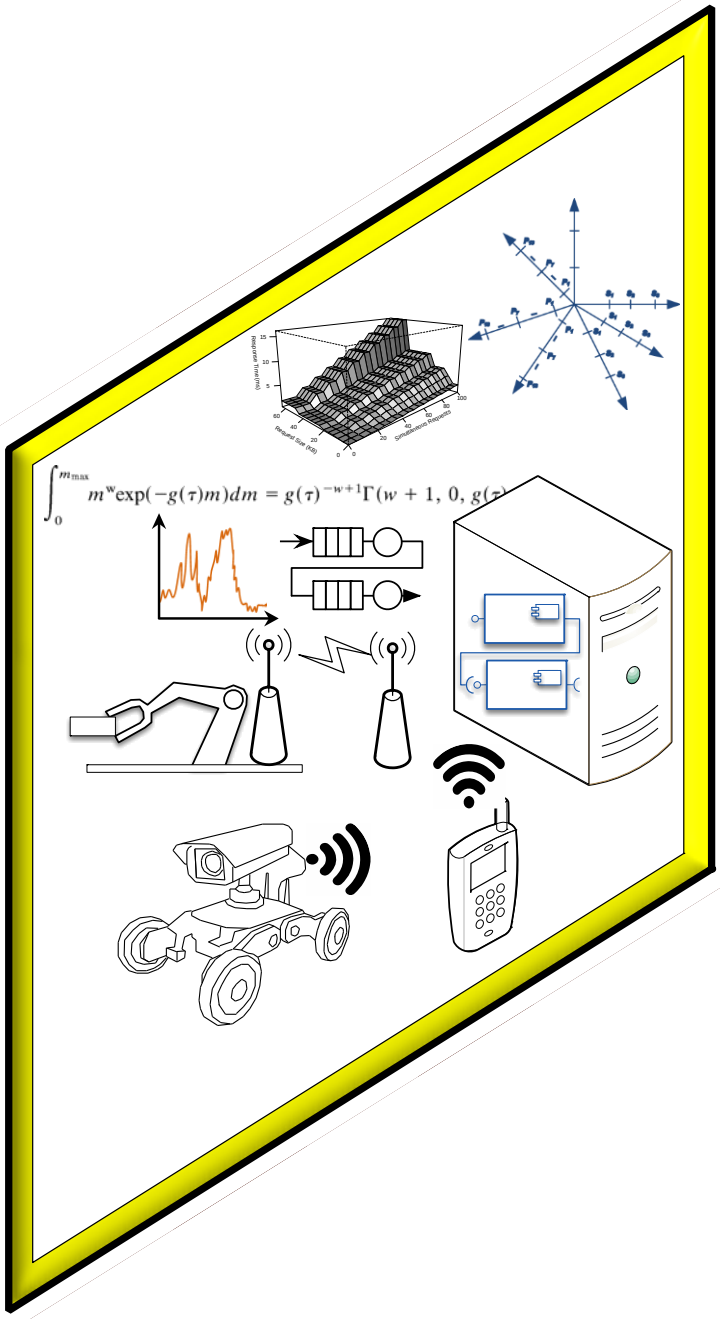
Model-driven Algorithms and Architectures for Self-Aware Computing Systems, Jan 18-23, 2015, Dagstuhl Seminar 15041

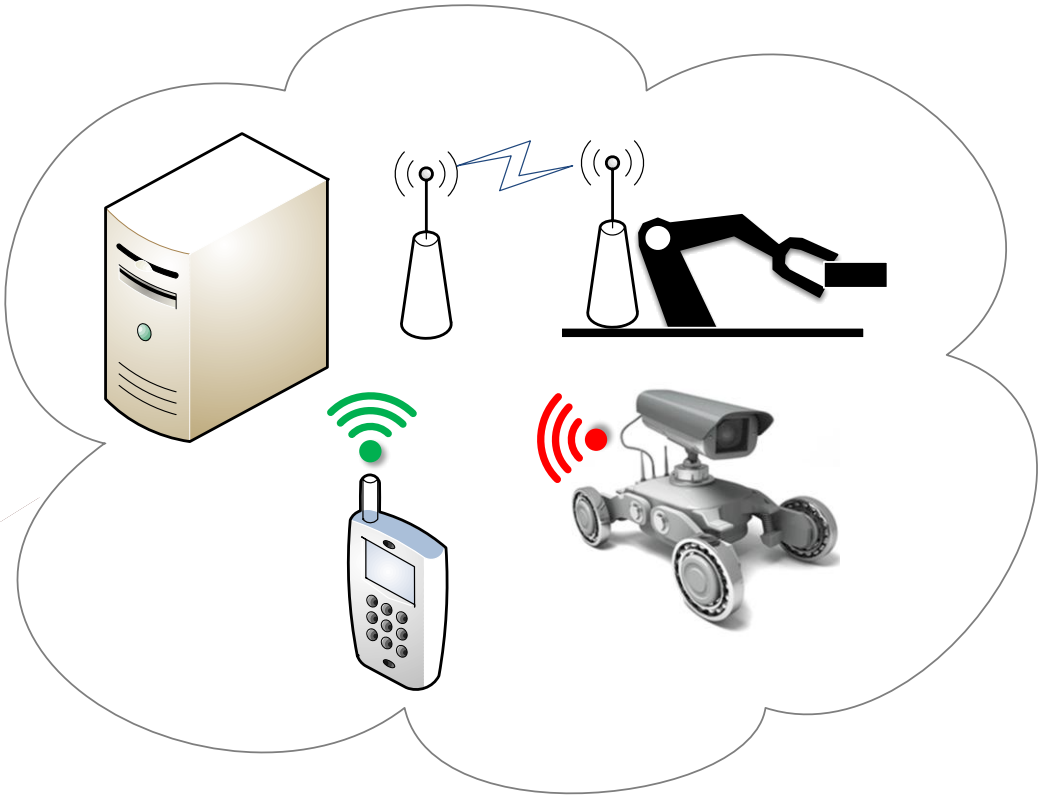
Organizers

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

The Vision





Self-Aware Computing

Definition

Self-aware Computing Systems 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

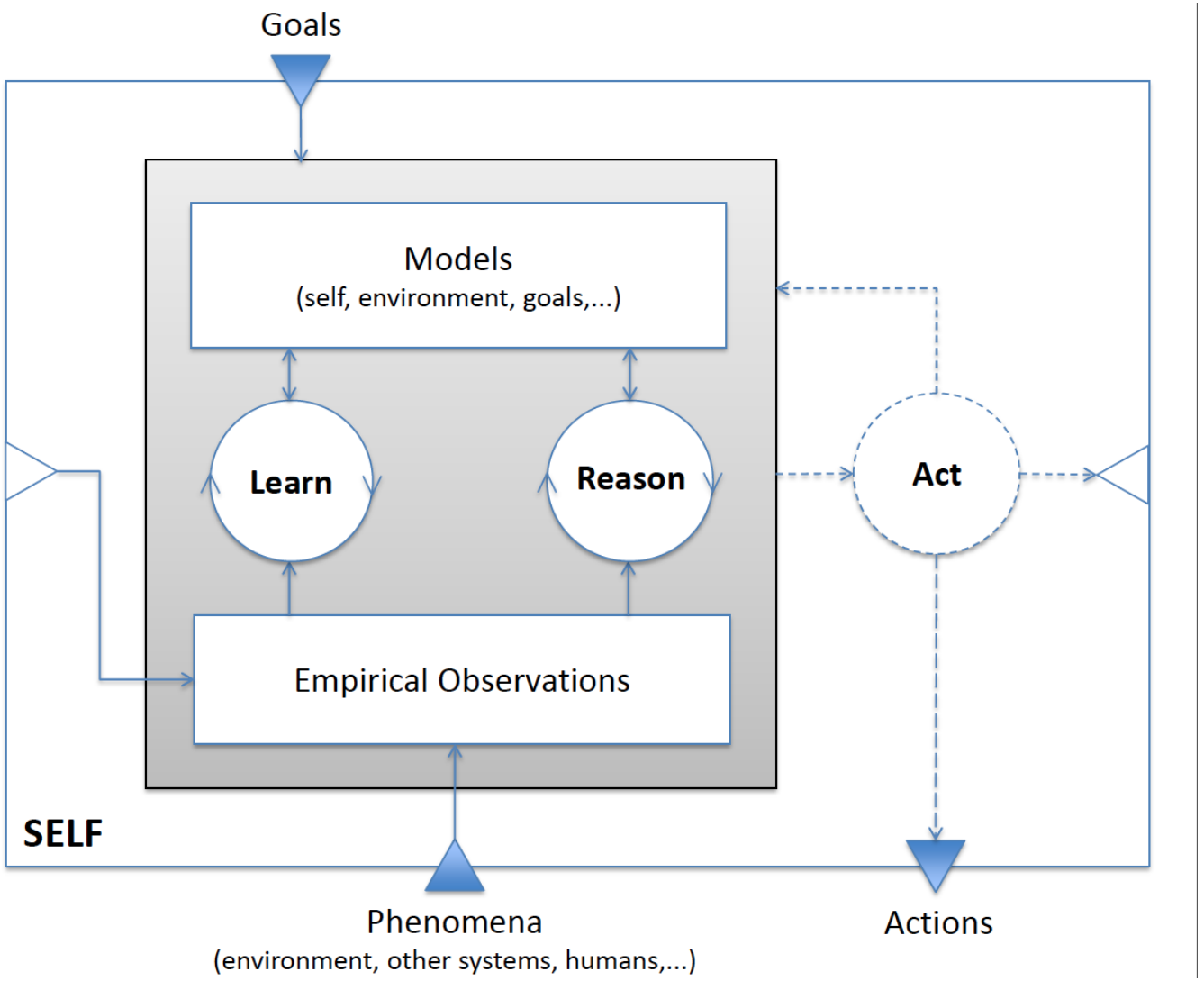
Self-aware Computing Systems are computing systems that:

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

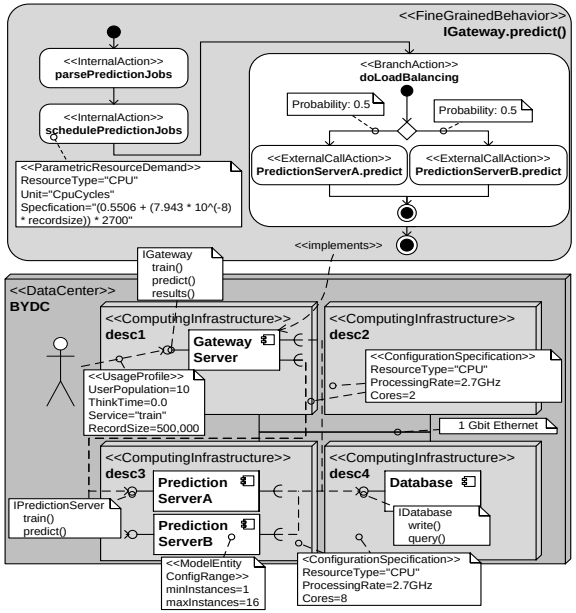
Descriptive Models

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

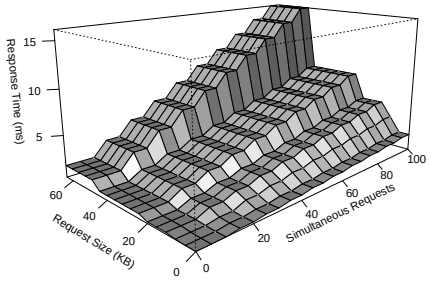
(Predictive) Analysis Models

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

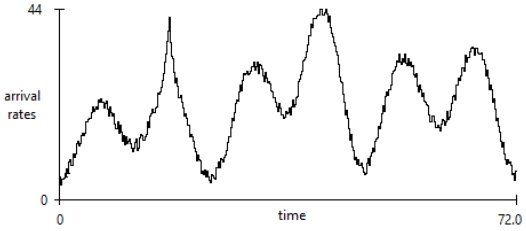
Examples of Models



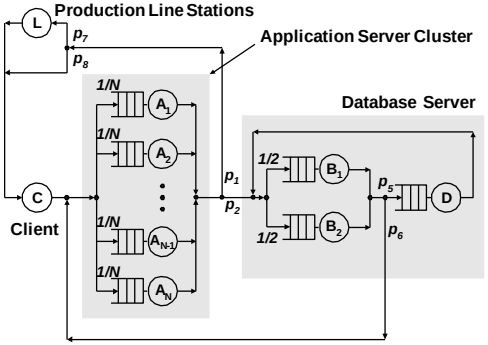
Descriptive MOF-based models



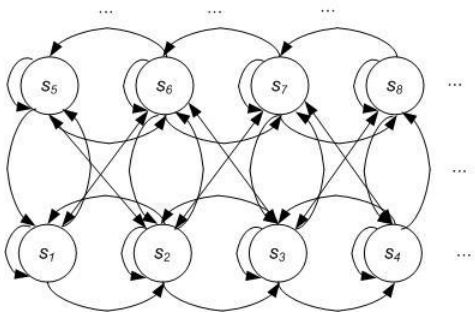
Statistical regression models



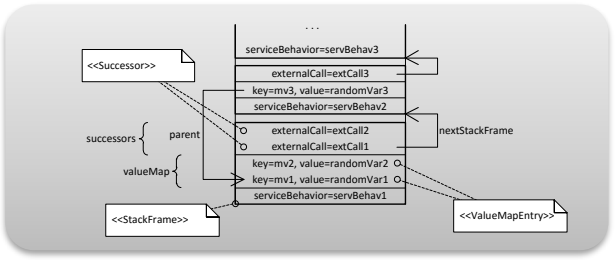
Load forecasting models



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)

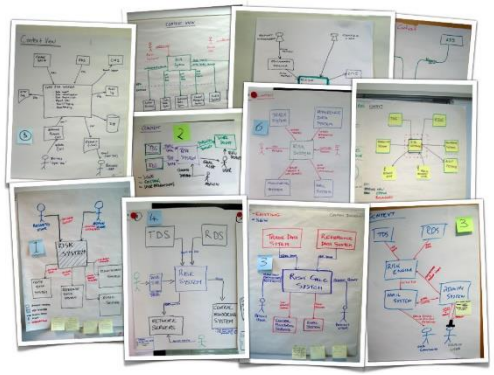
- 27 chapters, ca 700 pages, ca. 50 authors involved

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

Book Preview

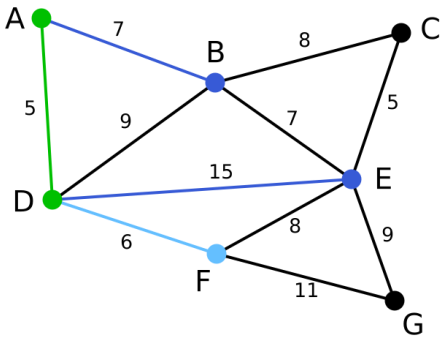
- **Part I: Introduction** (4 chapters)
  - defines self-aware computing systems from multiple perspectives
  - provides a formal unifying definition
  - establishes a taxonomy and a set of reference scenarios

- **Part II: System Architectures** (5 chapters)
  - introduces generic concepts and notations that allow to describe and compare architectures
  - reviews the current state of reference architecture architectural frameworks, and languages



Book Preview

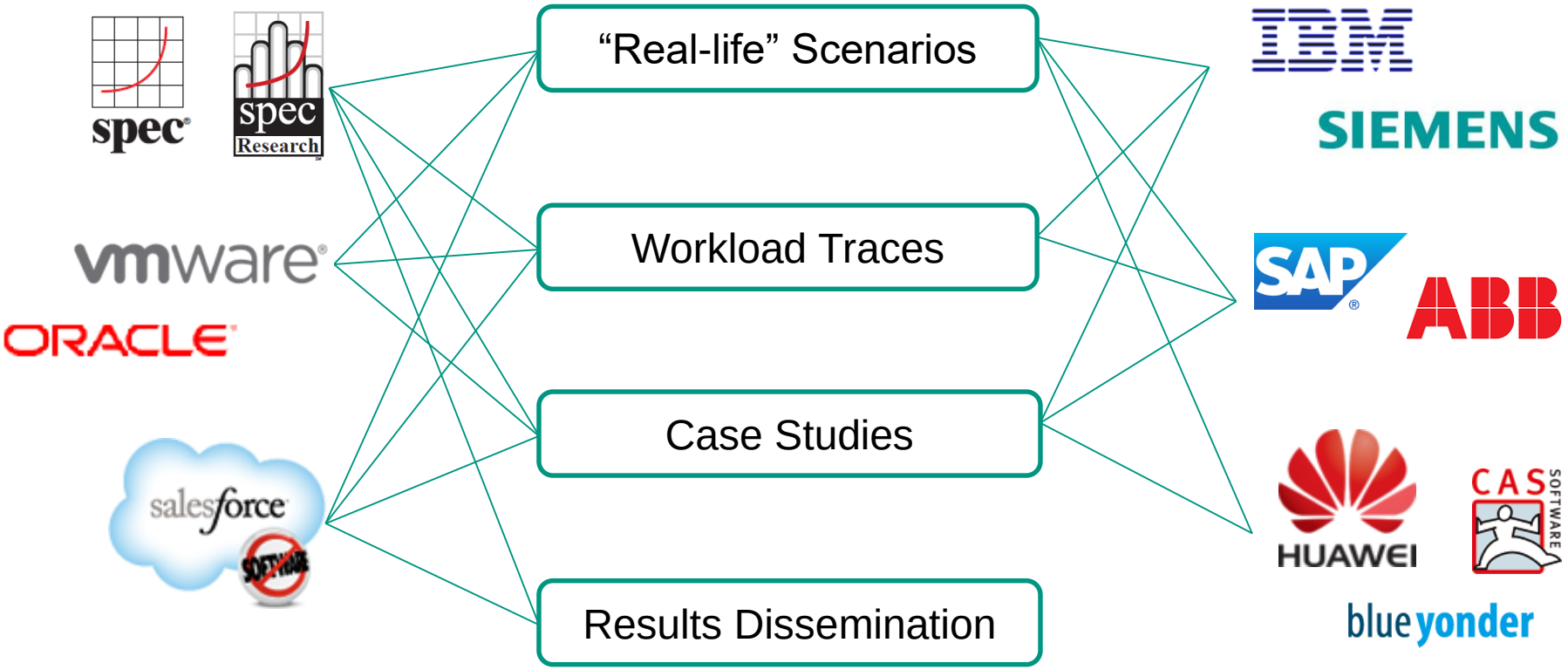
- **Part III: Methods and Algorithms** (7 chapters)
  - focuses on methods and algorithms addressing issues like modeling, synthesis and verification
  - also examines topics such as adaptation, benchmarks and metrics



- **Part IV: Applications and Case Studies** (10 chapters)
  - various domains including cloud computing, data centers, cyber-physical systems, spacecraft applications

- **Part V: Outlook**
  - surveys open challenges and research directions

Research Partners



Imperial College London

TU Delft

U C

Universidade de Coimbra

Universität Stuttgart

Università della Svizzera italiana

UCD DUBLIN

UPC Universitat Politècnica de Catalunya

BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Technische Universität Berlin

Research Funding Record

- Funding Record as PI (2010-2016): 3.6M€
  - Huawei Research Grant (2016-2017)
  - Four DFG Research Grants (2015-2018)
  - Google Research Award (2015-2016)
  - VMware Academic Research Award (2012-2015)
  - ABB Research Grant Award (2014-2015)
  - IBM PhD Fellowship Award (2014-2015)
  - EU Marie Curie ITN Project (2011-2015)
  - DFG Emmy Noether Career Award (2009-2014)

Selected Prizes for Supervised Theses

[Full list at: http://se.informatik.uni-wuerzburg.de/research/awards_and_recognitions]

- **Best Master Thesis Prize** (1000 €) from GFFT e.V., J. v. Kistowski, 2015.
- **IBM Ph.D. Fellowship Award** ($ 20 000) for N. Herbst, KIT, 2014.
- **Best Diploma Thesis Prize** (1000 €) from FZI Karlsruhe, N. Herbst, KIT, 2013.
- **Best Graduate Award** (2500 €) @ KIT, Simon Spinner, 2012.
- **Distinguished Dissertation Award** ($ 1,000) from SPEC, Kai Sachs, 2011.
- **Best Diploma Thesis Prize** (1500 €), ObjektForum Karlsruhe, B. Klatt, 2012.
- **Best Diploma Thesis Prize** (1000 €), FZI Karlsruhe, P. Meier, 2011.
- **Best Diploma Thesis Prize** (1000 €), FZI Karlsruhe, F. Brosig, 2010.

Best Paper Awards

[Full list at: http://se.informatik.uni-wuerzburg.de/research/awards_and_recognitions]

- **Best Poster Award**, Symposium on Software Performance (SOSP 2014), Stuttgart, Germany, November 26-28, 2014.
- **Best Paper Award Nomination**, 25th IEEE Intl. Symp. on Software Reliability Engineering (ISSRE 2014).
- **Best Paper Award**, 2011 International Conference on Cloud Computing and Service Science (CLOSER 2011), Noordwijkerhout, The Netherlands, 2011.
- **Best Paper Award**, 2011 International ICST Conference on Simulation Tools and Techniques (SIMUTools 2011), Barcelona, Spain, 2011.
- **Best Paper Award Nomination**, 7th International ACM SIGSOFT Conference on the Quality of Software Architectures (QoSA 2011).
- **Best Paper Award Nomination**, 11th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2008), Orlando, USA, 2008.
- **Best Paper Award**, 29th International Conference of the Computer Measurement Group (CMG 2003), Dallas, USA, 2003.
- **Best Paper Award**, 2003 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS 2003), Austin, USA.
- **Best Paper Award Nomination**, 28th Intl. Conference on Very Large Data Bases (VLDB 2002), Hong Kong, China, 2002.

Project Ideas

- Development of a platform for elastic NFV applications
- Development of elastic auto-scaling mechanisms for the Huawei cloud platform
- Benchmarking of the Huawei cloud platform
  - Performance, scalability, elasticity
- Performance isolation in shared execution environments
  - Virtualized infrastructures
  - Container-based environments
  - Multi-tenant applications

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Descartes
research

Questions?

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