

Model-Based Approach to Designing Self-Aware IT Systems and Infrastructures

Samuel Kounev

CHAIR OF SOFTWARE ENGINEERING, DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF WÜRZBURG

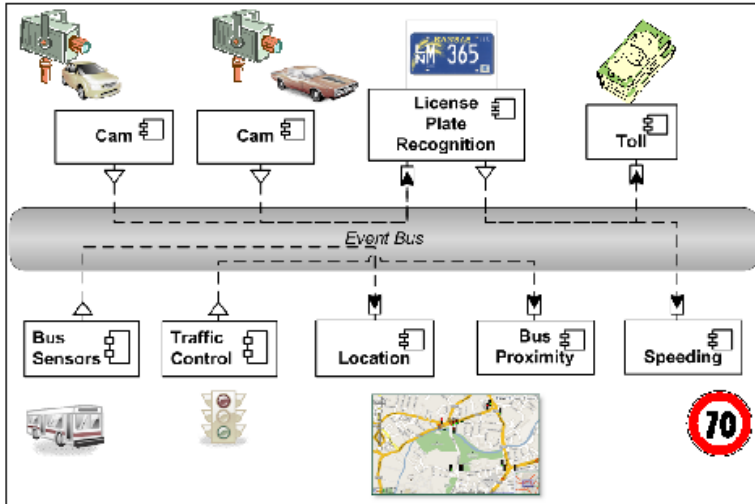


Referenzen

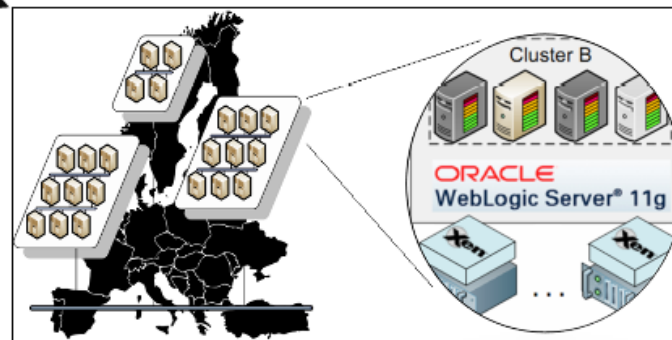
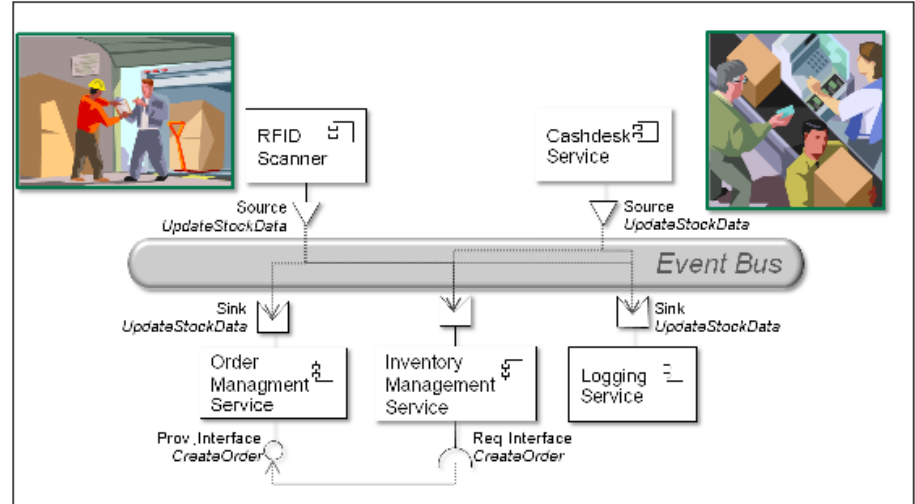
- S. Kounev, N. Huber, F. Brosig, and X. Zhu. **Model-Based Approach to Designing Self-Aware IT Systems and Infrastructures.** *IEEE Computer Magazine*, 2016, IEEE. Accepted for Publication, To appear in 2016. *Available on request.*
- S. Kounev, F. Brosig, and N. Huber. **The Descartes Modeling Language.** 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. Koziolok, 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. Koziolok, H. Koziolok, 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](#)]
- 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, N. Huber, S. Kounev and E. Amrehn. **Self-Adaptive Workload Classification and Forecasting for Proactive Resource Provisioning.** *Concurrency and Computation - Practice and Experience*, 26(12):2053-2078, John Wiley and Sons, Ltd., 2014. [[DOI](#) | [http](#) | [.pdf](#)]
- J. v. Kistowski, N. Herbst, D. Zoller, S. Kounev, and A. Hotho. **Modeling and Extracting Load Intensity Profiles.** In *Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2015)*, Firenze, Italy, May 18-19, 2015. [[DOI](#) | [slides](#) | [.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](#)]
- N. Herbst, S. Kounev, A. Weber, and H. Groenda. **BUNGEE: An Elasticity Benchmark for Self-Adaptive IaaS Cloud Environments.** In *Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2015)*, Firenze, Italy, May 18-19, 2015. [[slides](#) | [.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.
- A. Milenkoski, B. Payne, N. Antunes, M. Vieira, S. Kounev, A. Avrtizer, and M. Luft. **Evaluation of Intrusion Detection Systems in Virtualized Environments Using Attack Injection.** In *The 18th International Symposium on Research in Attacks, Intrusions, and Defenses (RAID 2015)*, Kyoto, Japan, November 2015. Springer. November 2015.

Motivation

Verkehrsüberwachungssystem

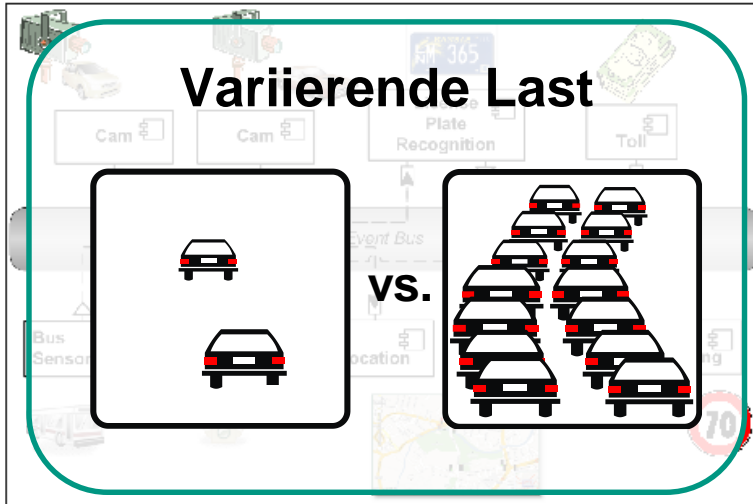


Warenwirtschaftssystem

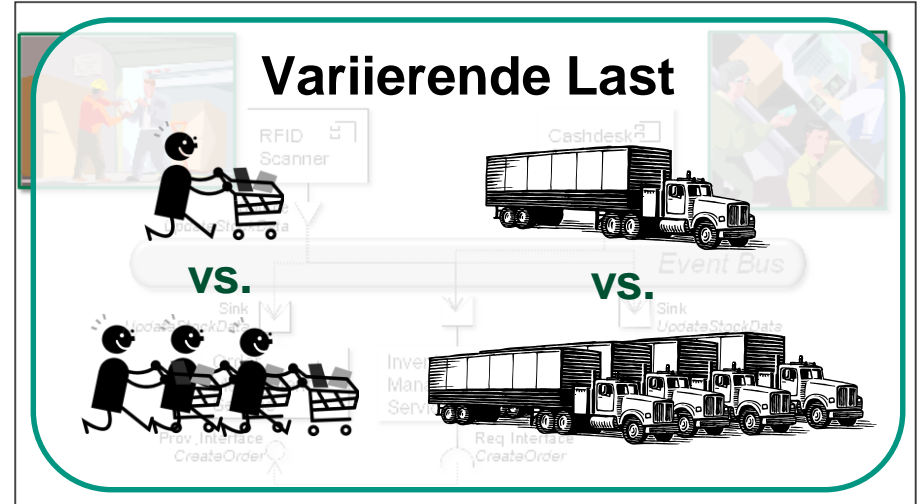


Motivation

Verkehrsüberwachungssystem



Warenwirtschaftssystem



- Hohe Komplexität und Dynamik
→ neue Herausforderungen für die Verlässlichkeit
- Systeme müssen **zur Laufzeit** immer häufiger **rekonfiguriert werden**
 - Skalieren durch Replikation von Komponenten / virtuellen Maschinen
 - Anpassung der Ressourcenzuteilungen / Systemkonfiguration
 - Neustarten von Diensten / Komponenten

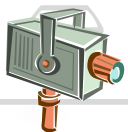


Motivation

Verkehrsüberwachungssystem

Laufzeit-Fehler / Abstürze

- Software-Komponenten
- Betriebssystem/Middleware
- Hardware



vs.



Warenwirtschaftssystem

Laufzeit-Fehler / Abstürze

- Software-Komponenten
- Betriebssystem/Middleware
- Hardware



vs.



- Hohe Komplexität und Dynamik

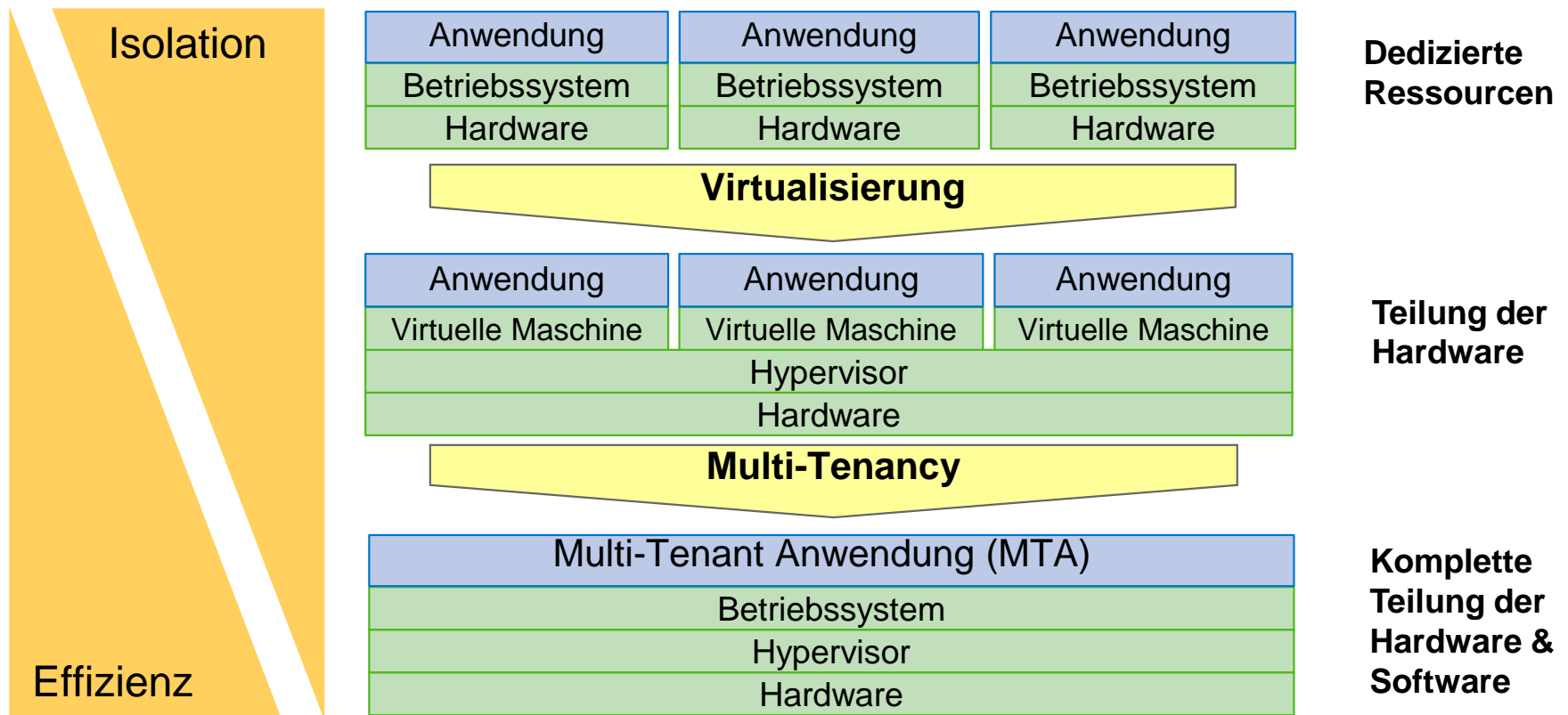
→ neue Herausforderungen für die Verlässlichkeit

- Systeme müssen **zur Laufzeit** immer häufiger **rekonfiguriert werden**
 - Skalieren durch Replikation von Komponenten / virtuellen Maschinen
 - Anpassung der Ressourcenzuteilungen / Systemkonfiguration
 - Neustarten von Diensten / Komponenten

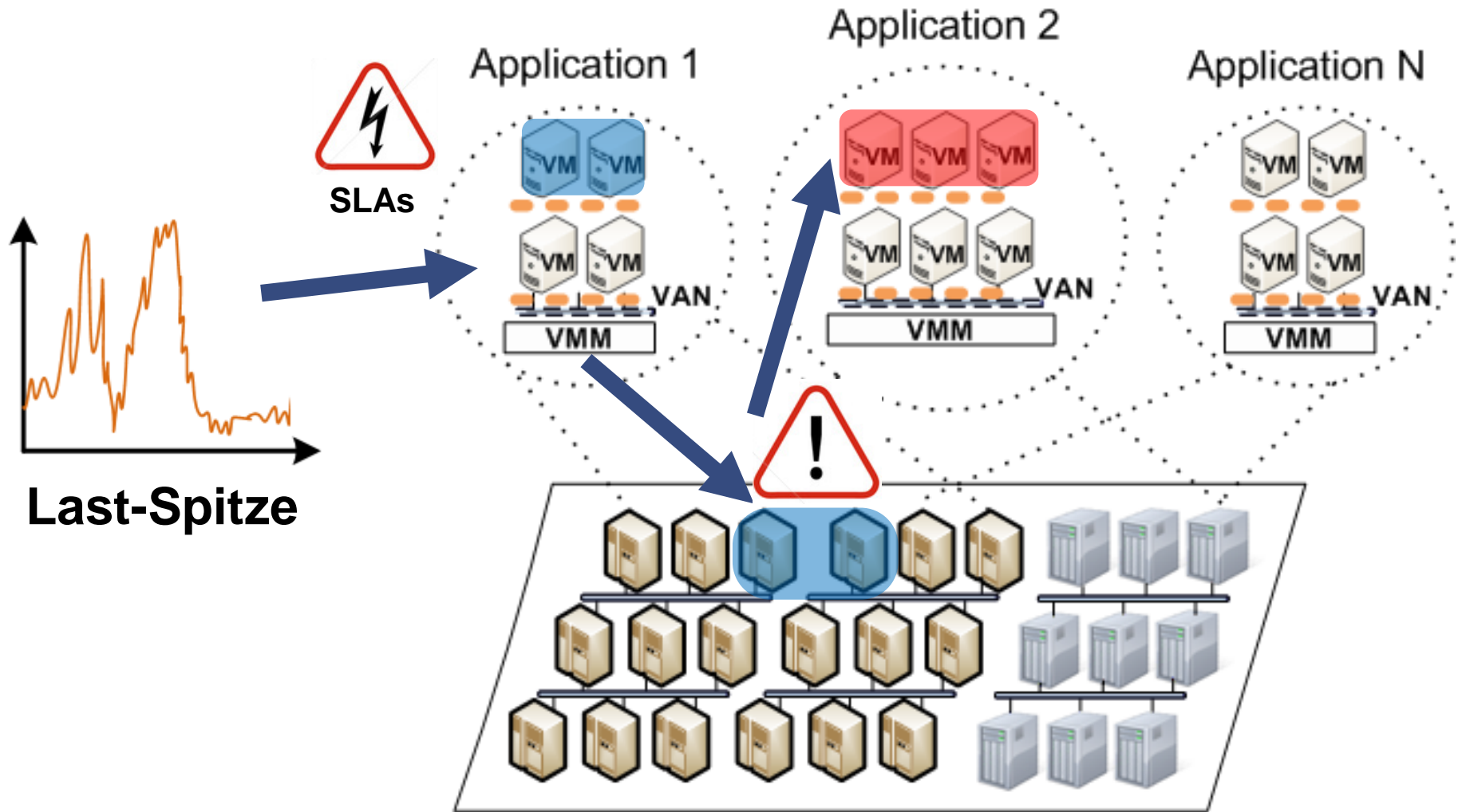


Zunehmender Druck zur Steigerung der Effizienz

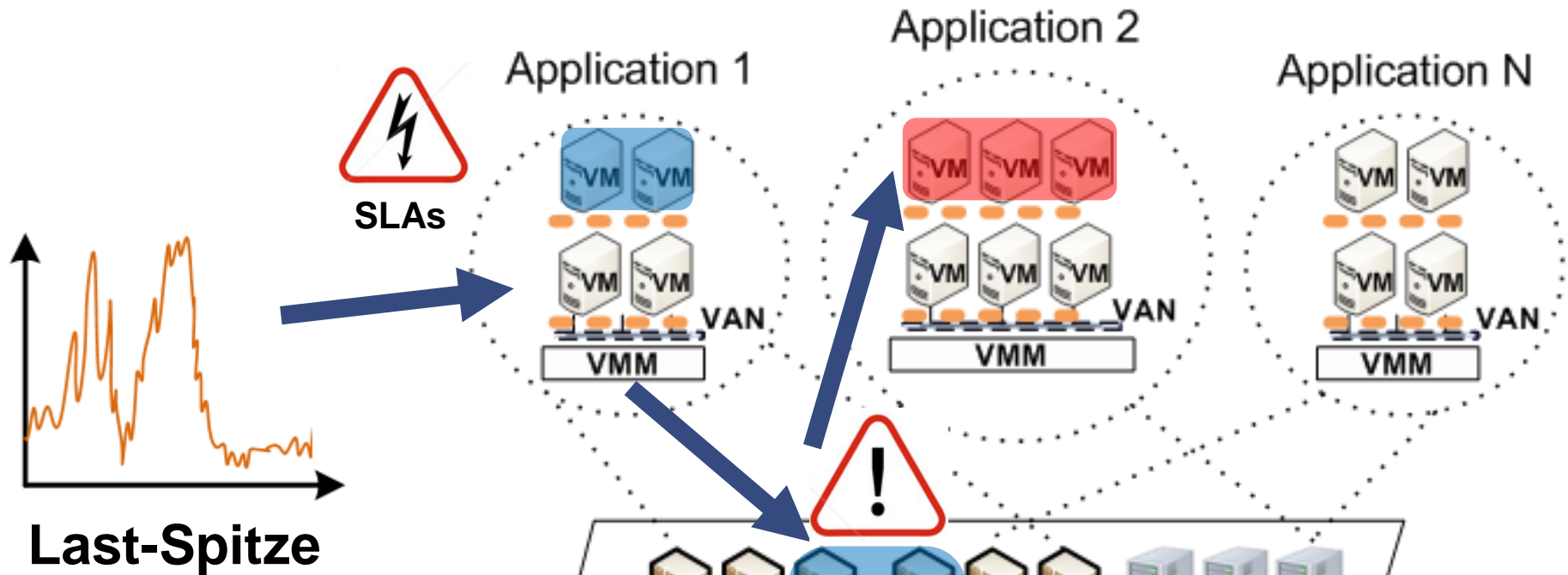
- Verschiedene Ansätze zur Teilung der IT-Ressourcen
 - Netzwerk-, Storage-, und Recheninfrastruktur
 - Software-Schichten



Verfügbarkeit & Performance



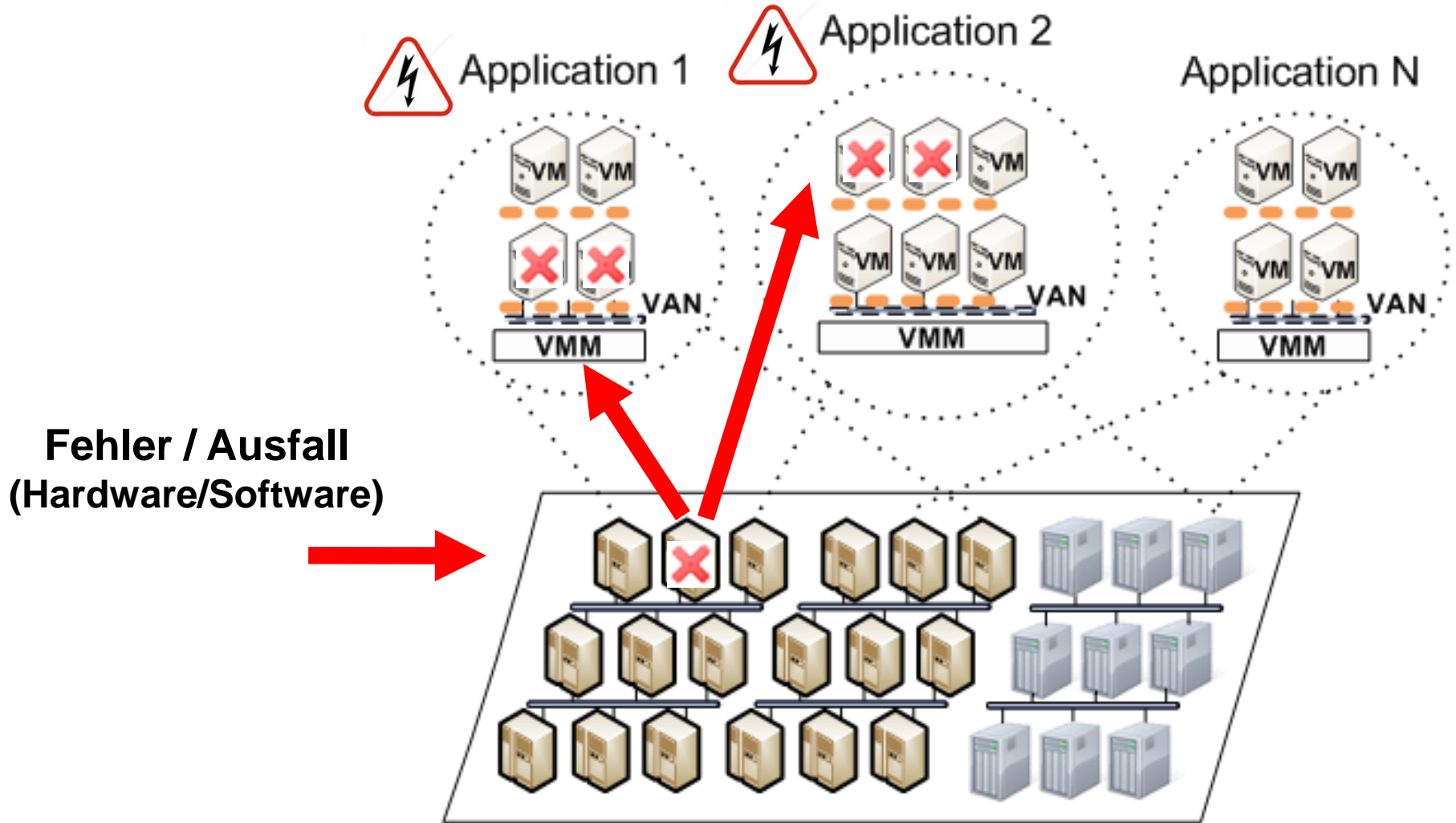
Verfügbarkeit & Performance



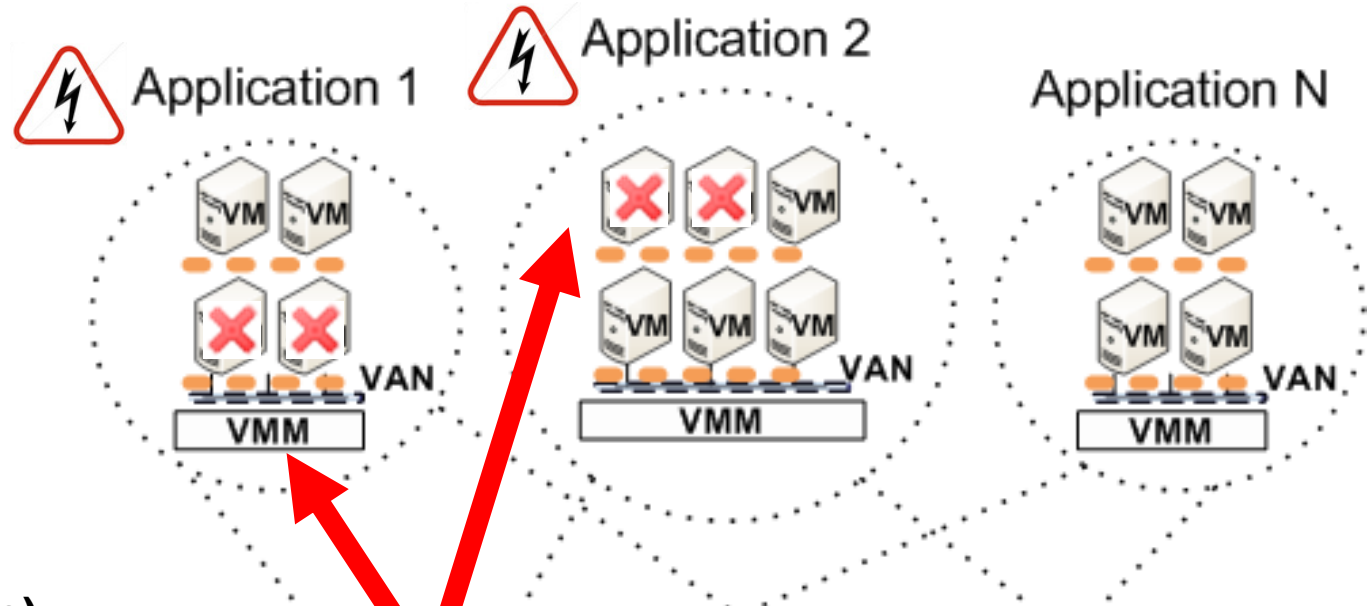
Elastische Skalierung der Ressourcen zur Laufzeit

- Wie kann man die Last-Spitze vorhersagen?
- Wann genau eine Rekonfiguration (Skalierung) auslösen?
- Welche konkreten Ressourcen skalieren? In welcher Granularität?

Zuverlässigkeit



Zuverlässigkeit



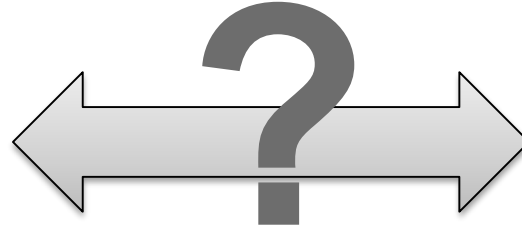
Fehler / Ausfall (Hardware/Software)

- Wie kann man Fehler/Abstürze vorhersagen und vermeiden?
- Wann genau eine Rekonfiguration auslösen?
- Welche Dienste sollen neu gestartet werden?



Semantic-Gap-Problem

Verlässlichkeit und
Effizienz auf
Anwendungsebene



Konfiguration des
Systems auf
technischer Ebene

▪ Verfügbarkeit & Performance

- Dienste funktionsbereit 99.99% der Zeit
- Antwortzeit einer Anfrage < 20 ms
- Durchsatz: Aufträge pro Sekunde > 1000
- Server-Auslastung > 60%
- ...

▪ Zuverlässigkeit

- Dienste funktionieren korrekt
- „Time to recover after a failure“ < 1 min

BEISPIELE

- Wie viele vCPUs / wie viel Hauptspeicher sollen einer VM zugewiesen werden?
- Wann genau soll eine Rekonfiguration ausgelöst werden?
- Welche Ressourcen sollen dabei skaliert / repliziert / migriert werden?
- Wie schnell und in welcher Granularität?
- Welche Dienste sollen neu gestartet werden?
- ...



Gliederung

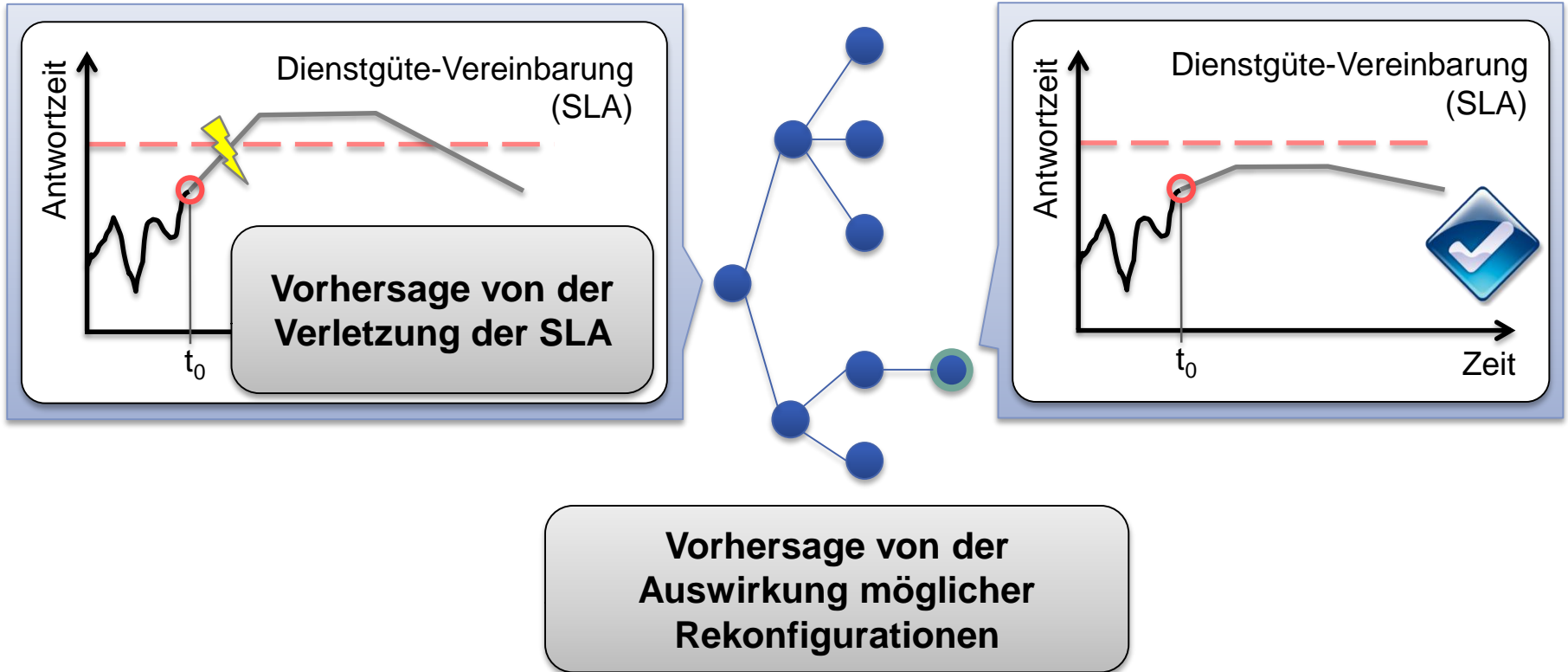
1. Einführung
2. Forschungsfragen
3. Wissenschaftlicher Ansatz
4. Ausgewählte Fallstudie
5. Vision & Ausblick

DFG Emmy Noether Projekt „Descartes“

- Modellierungs-Ansätze zur **Vorhersage zur Laufzeit** von der Auswirkung dynamischer Änderungen auf die Dienstgüte (engl. Quality-of-Service) des Systems
 - Ursprünglicher Fokus: Performance und Verfügbarkeit
 - Später: Zuverlässigkeit
- Modell-basierte Algorithmen und Techniken zur **autonomen System-Adaption** während Betrieb



Beispiel: Vorhersage zur Laufzeit



→ **Self-Aware Computing** (mehr dazu später)

Übergeordnete Forschungsfragen

- Was sind angemessene Abstraktionen zur Modellierung der Systemarchitektur um eine **Vorhersage der Auswirkung dynamischer Änderungen zur Laufzeit** zu ermöglichen?
- Wie kann der große Zustandsraum möglicher Rekonfigurationen in den Griff bekommen werden?
- Welche Modell-Analyseverfahren und Optimierungsmethoden ermöglichen eine schnelle und effiziente Rekonfiguration?
- ...

Beispiel-Einflussfaktoren der System-Architektur

Nutzungsprofil, z.B:

- Menge und Art der Dienstnehmer
- Eingabeparameter & Eingabedaten
- Benutzte Datenformate
- Arbeitsablauf (engl. Workflow)

Software-Architektur, z.B:

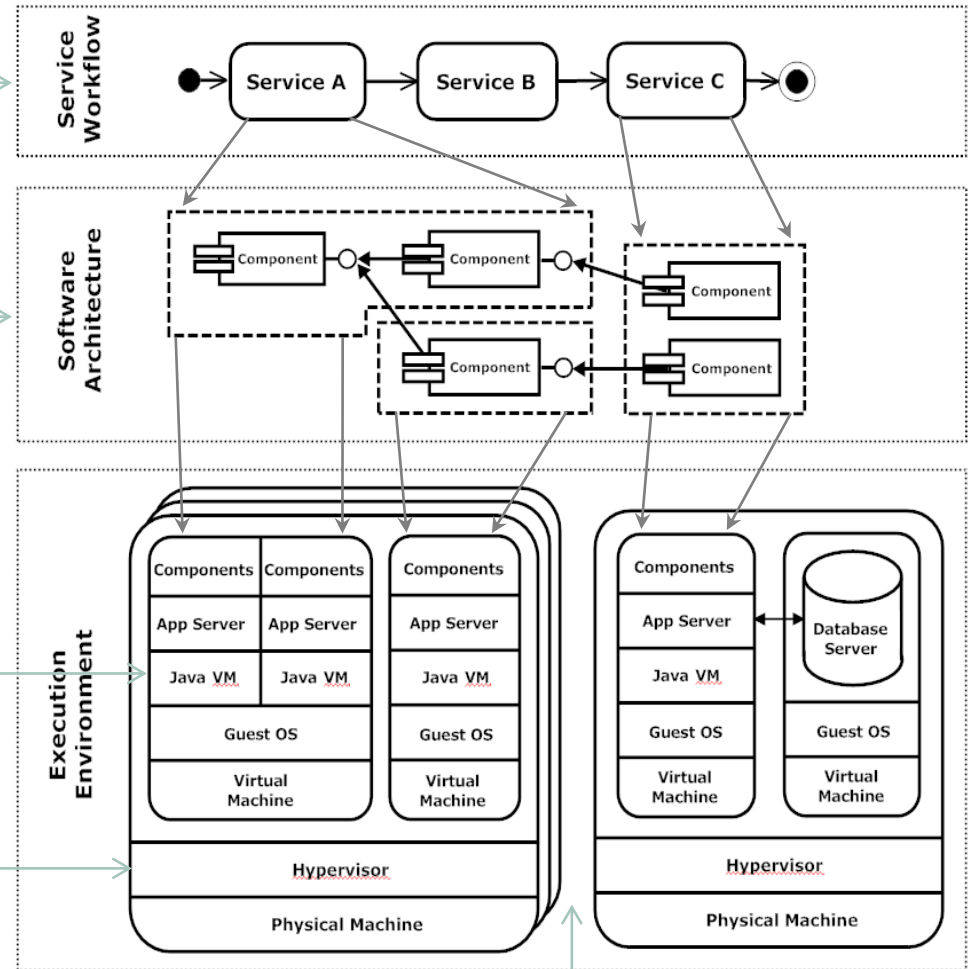
- Verbindungen zw. Komponenten
- Kontrollfluss und Datenfluss
- Komponenten-Nutzungsprofil

Ausführungsumgebung, z.B:

- Anzahl der Komponenten-Instanzen
- Server-Ausführungsfäden
- Hauptspeicher (Java Heap)
- Größe der DB-Connection-Pools

Virtualisierungsschicht, z.B:

- Zuweisung physikalischer Ressourcen
 - Anzahl Prozessoren
 - Hauptspeicher
 - Sekundärer Speicher

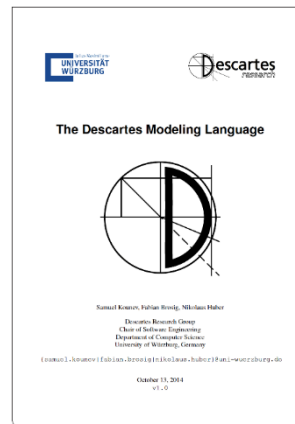


Gliederung

1. Einführung
2. Forschungsfragen
3. Wissenschaftlicher Ansatz
4. Ausgewählte Fallstudie
5. Vision & Ausblick

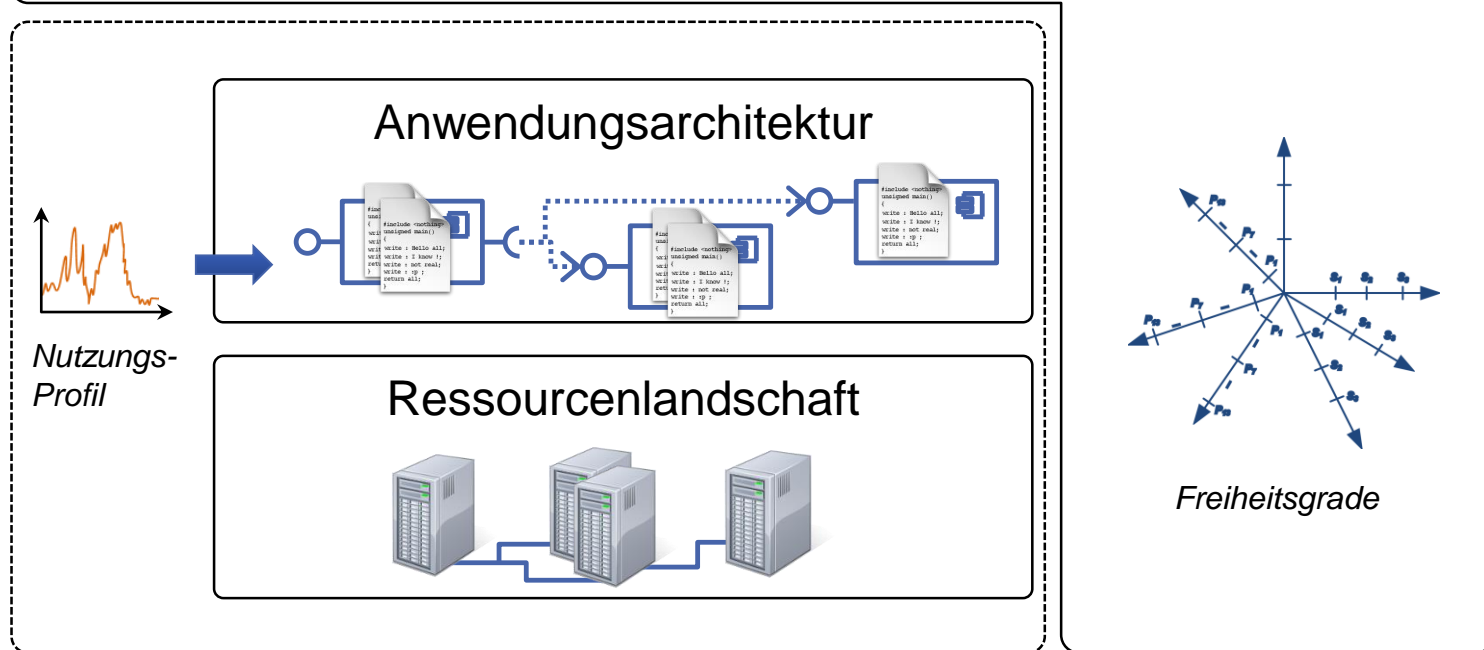
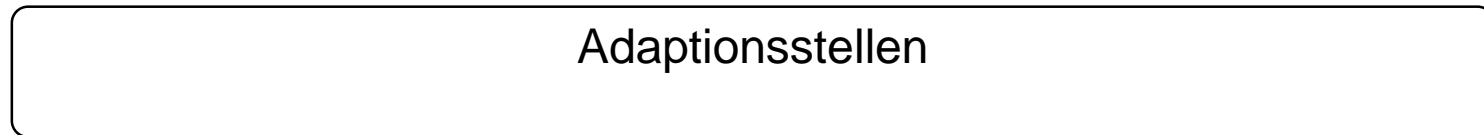
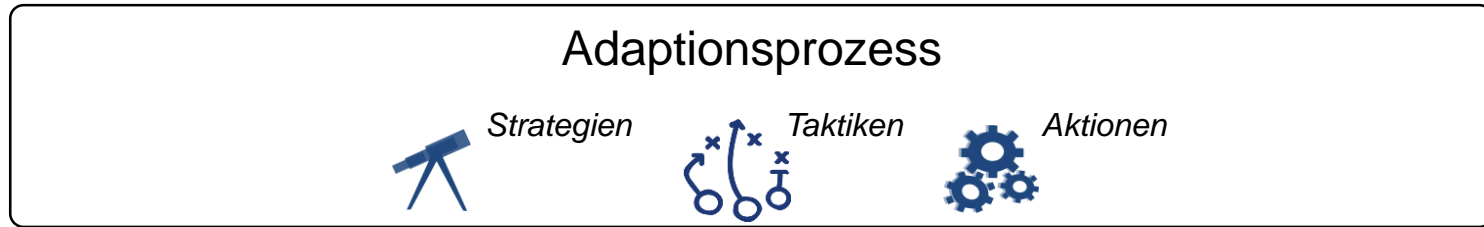
Descartes Modeling Language (DML)

- Modellierungssprache zur Modellierung der Architektur von verteilten Software-Systemen und IT-Infrastrukturen
- Basierend auf OMG's Meta-Object Facility (MOF) meta-models
- Deskriptive Modelle als Basis für
 - Vorhersage der Auswirkung dynamischer Änderungen zur Laufzeit
 - Autonome System-Adaption



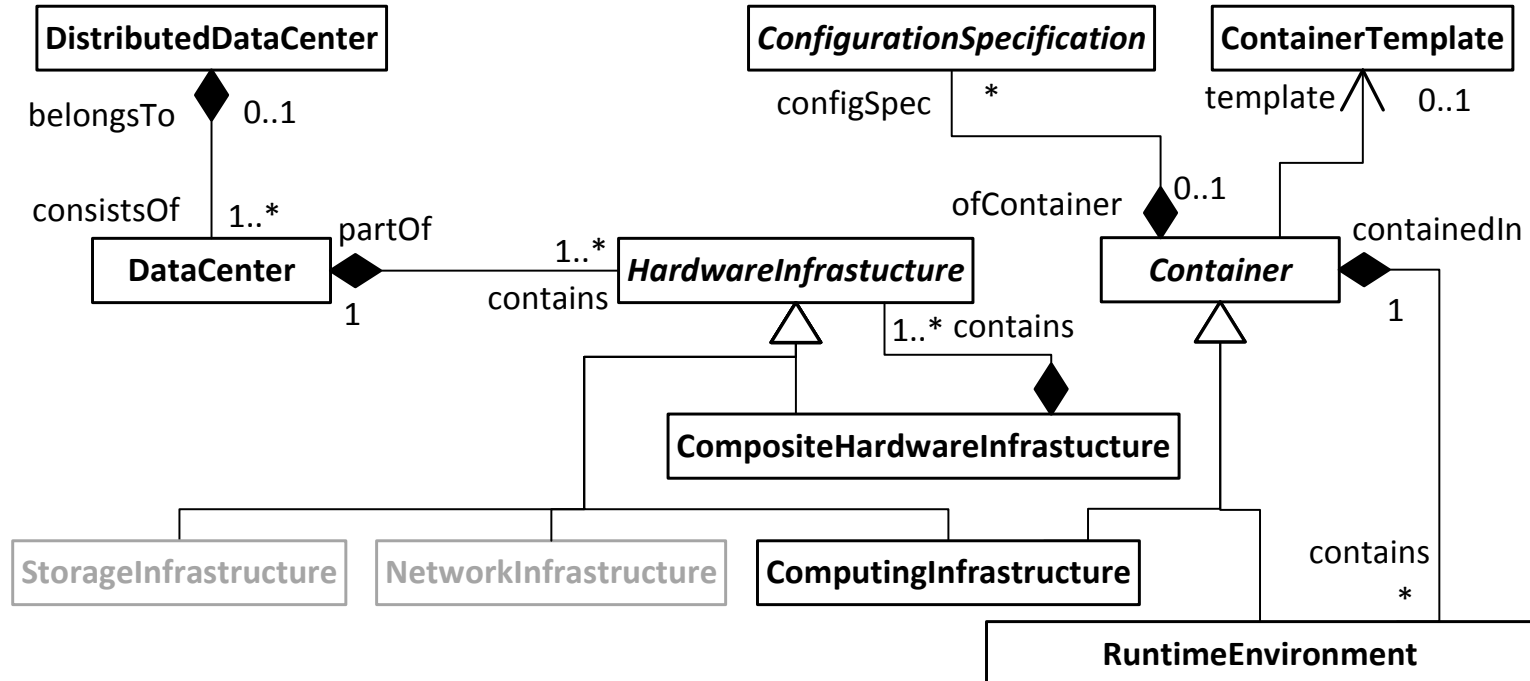
<http://descartes.tools/dml>

DML Sub-Modelle

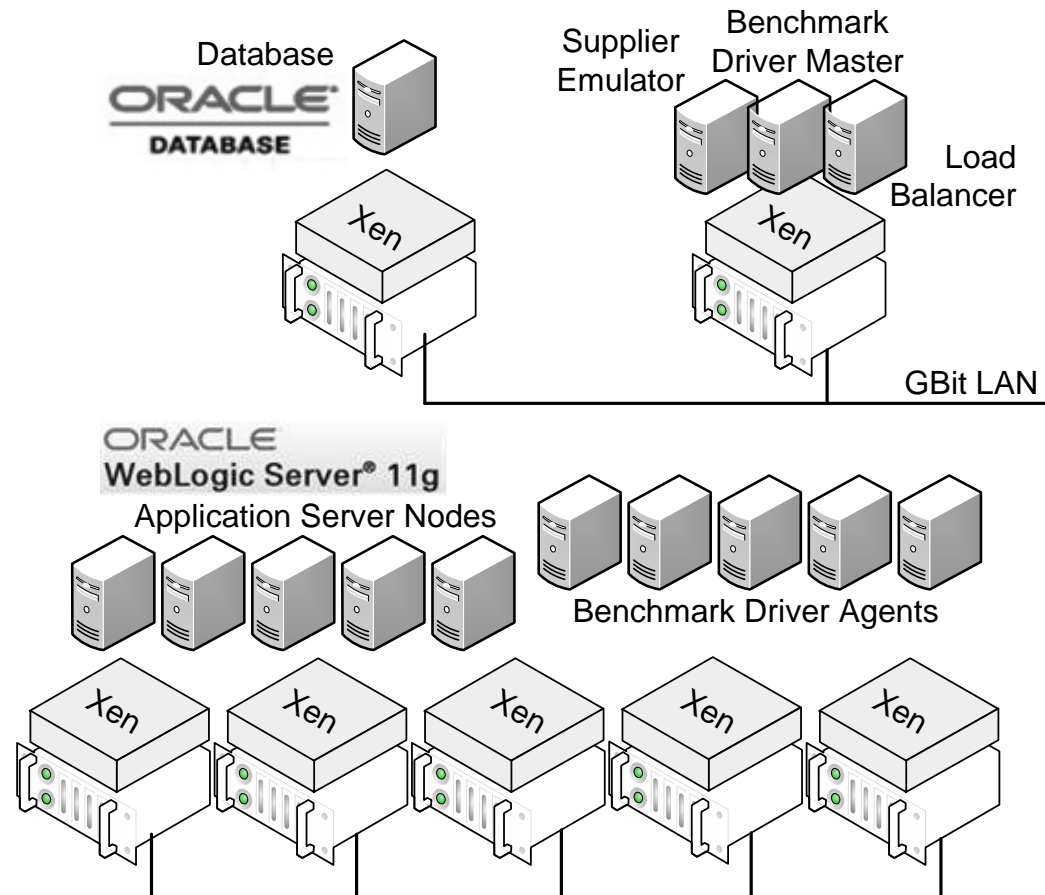


DML Implementierung mit EMF

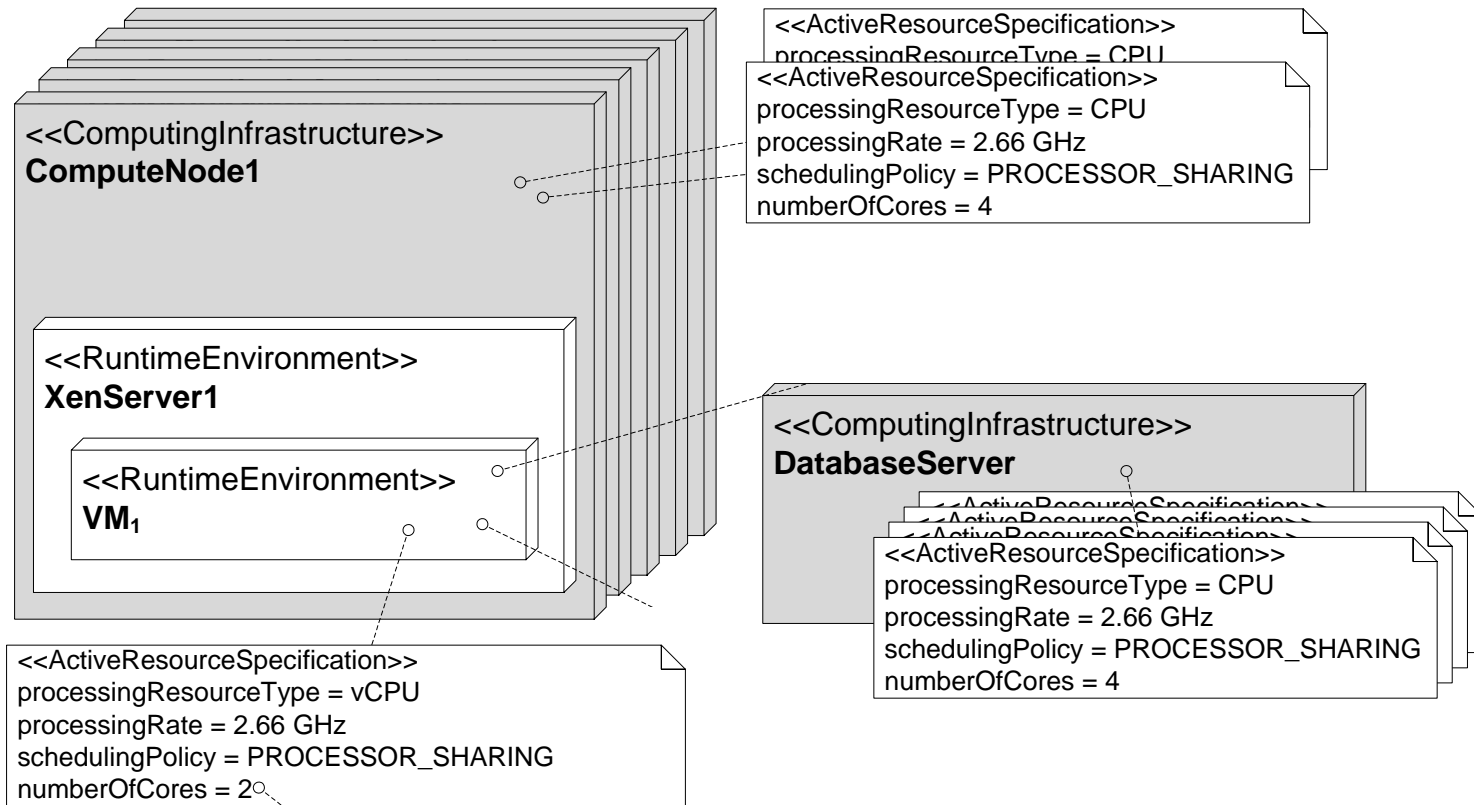
- Implementierung mit Ecore (Eclipse Modeling Framework)
- Beispiel Meta-Model: Ressourcenlandschaft



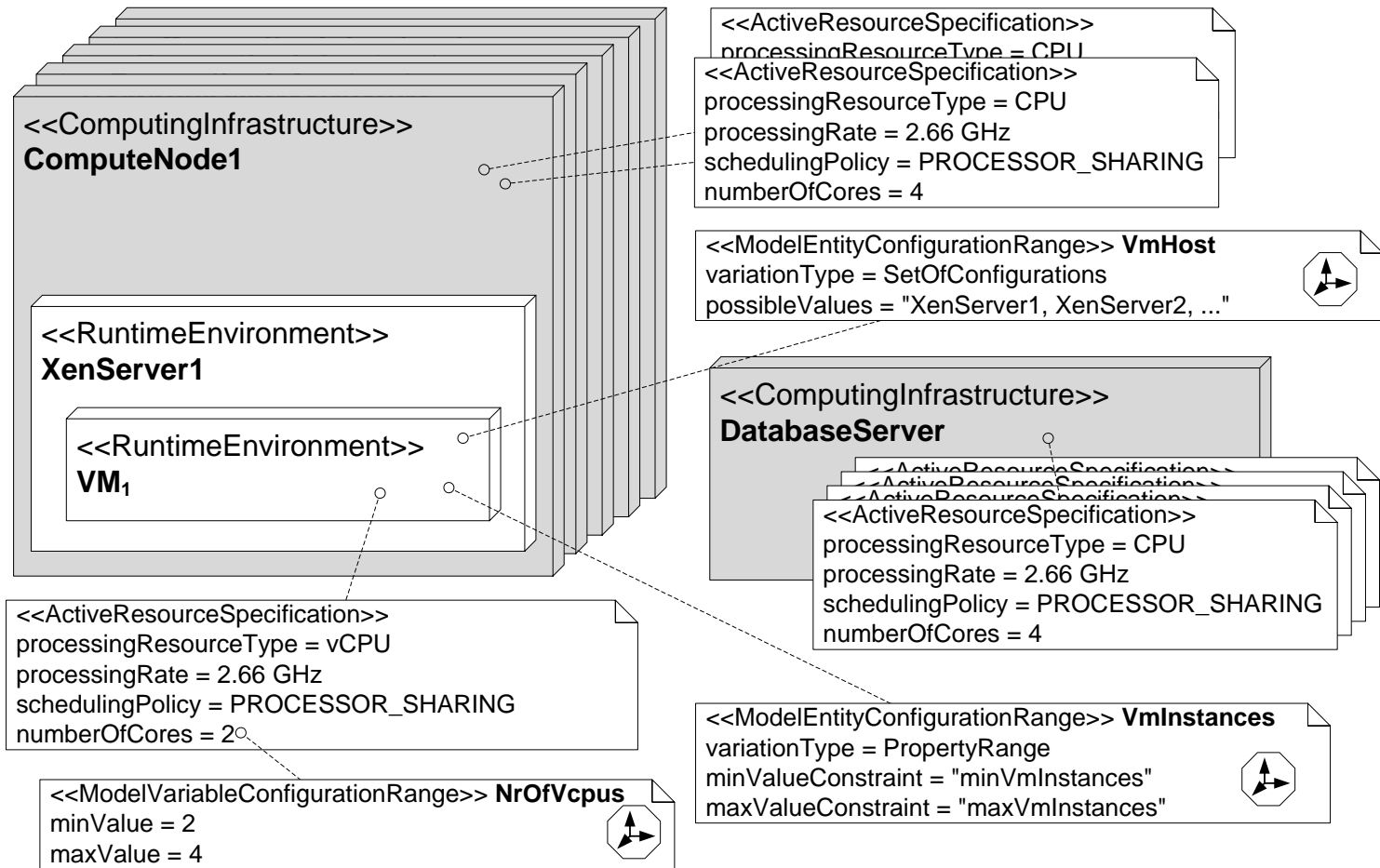
Beispiel: WebLogic Server Cluster



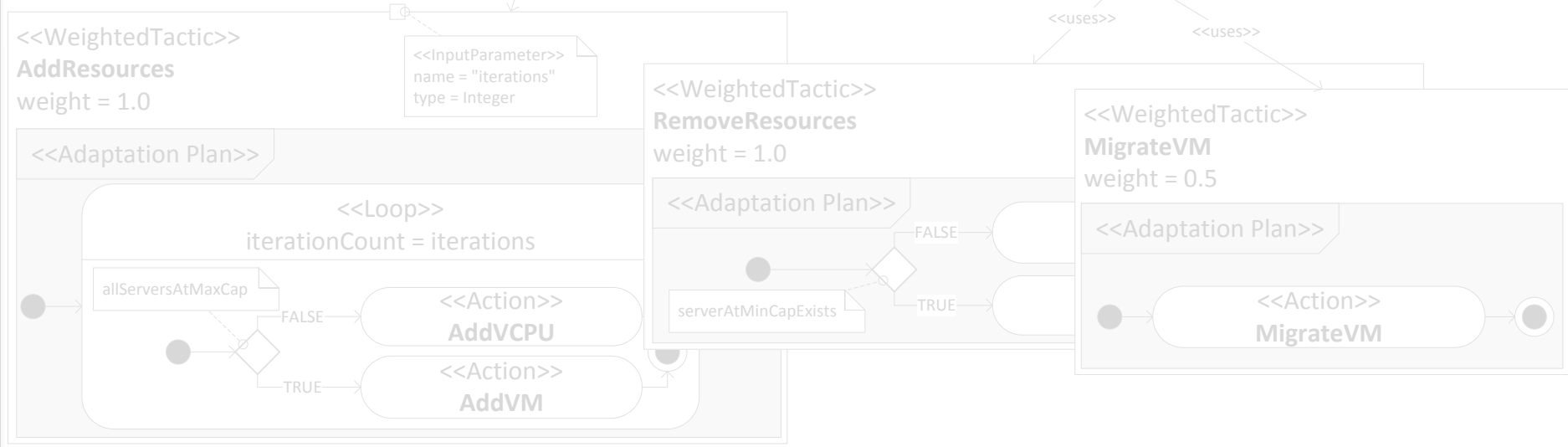
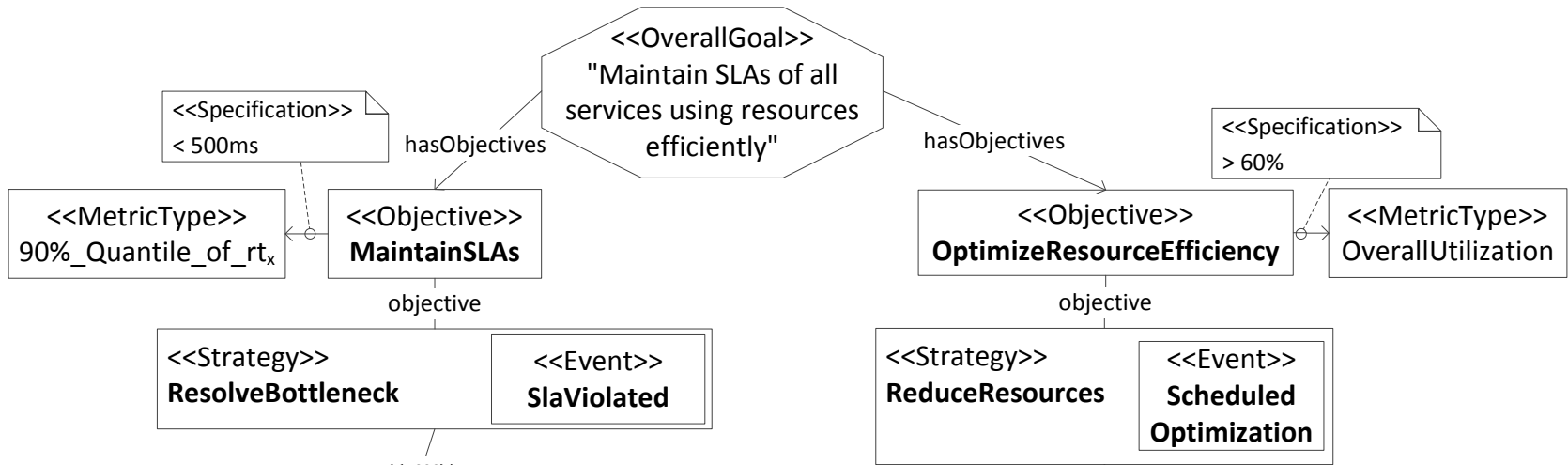
Beispiel: DML Model Instance



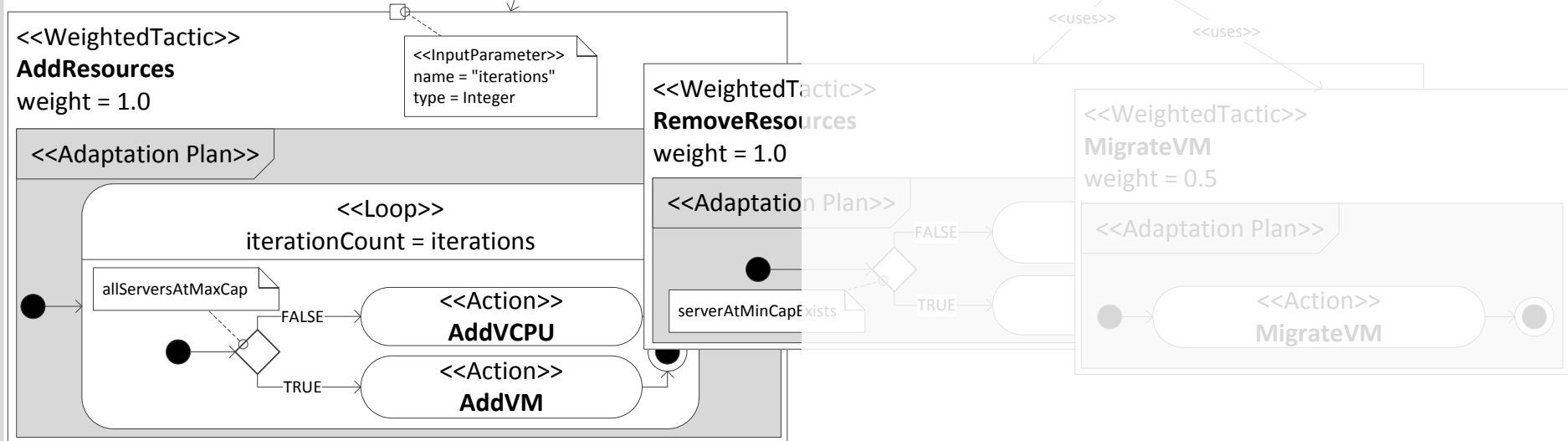
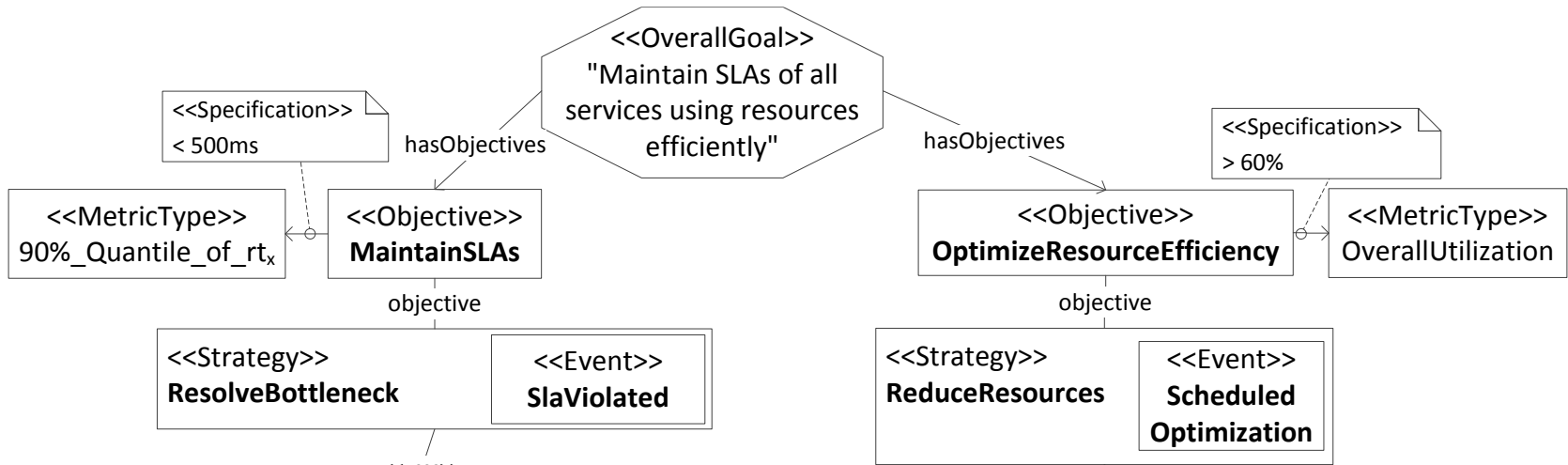
Beispiel: DML Model Instance



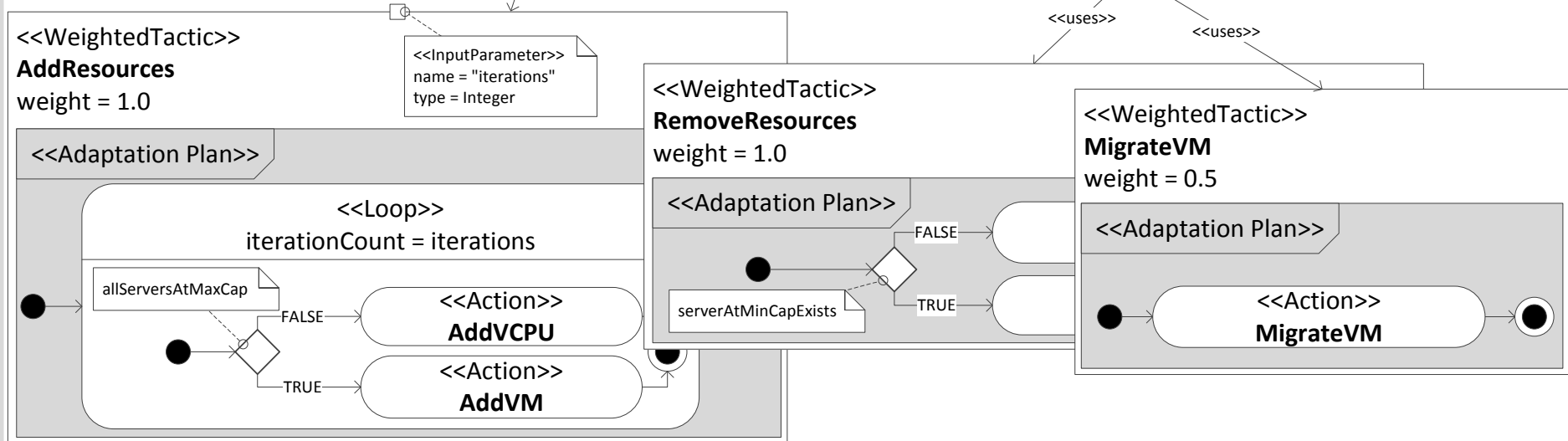
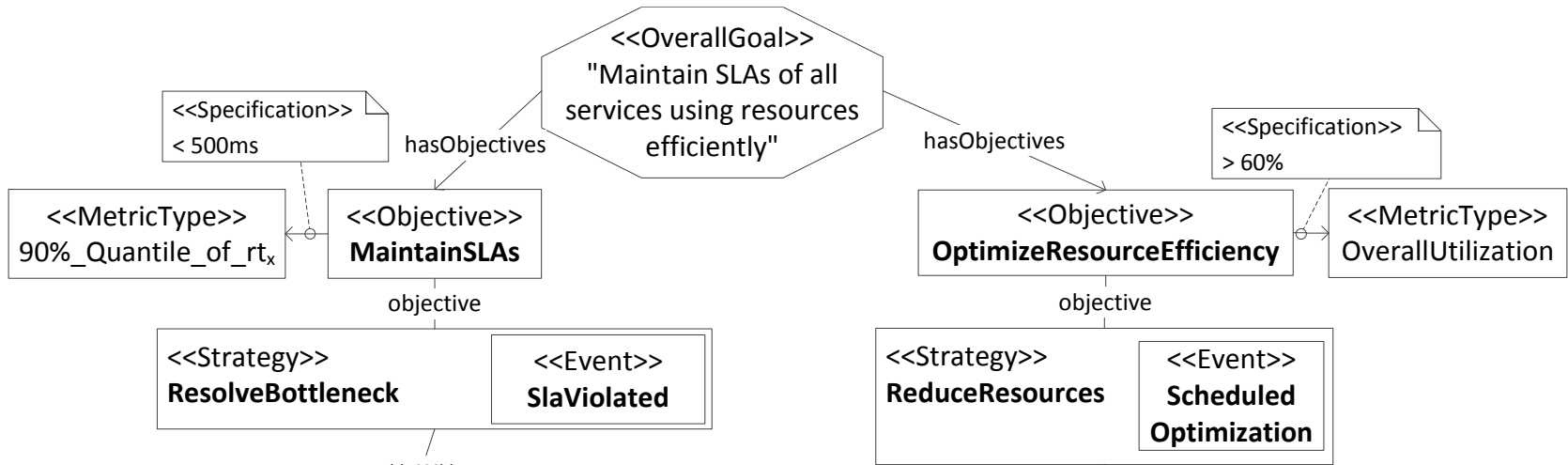
Beispiel: DML Model Instance



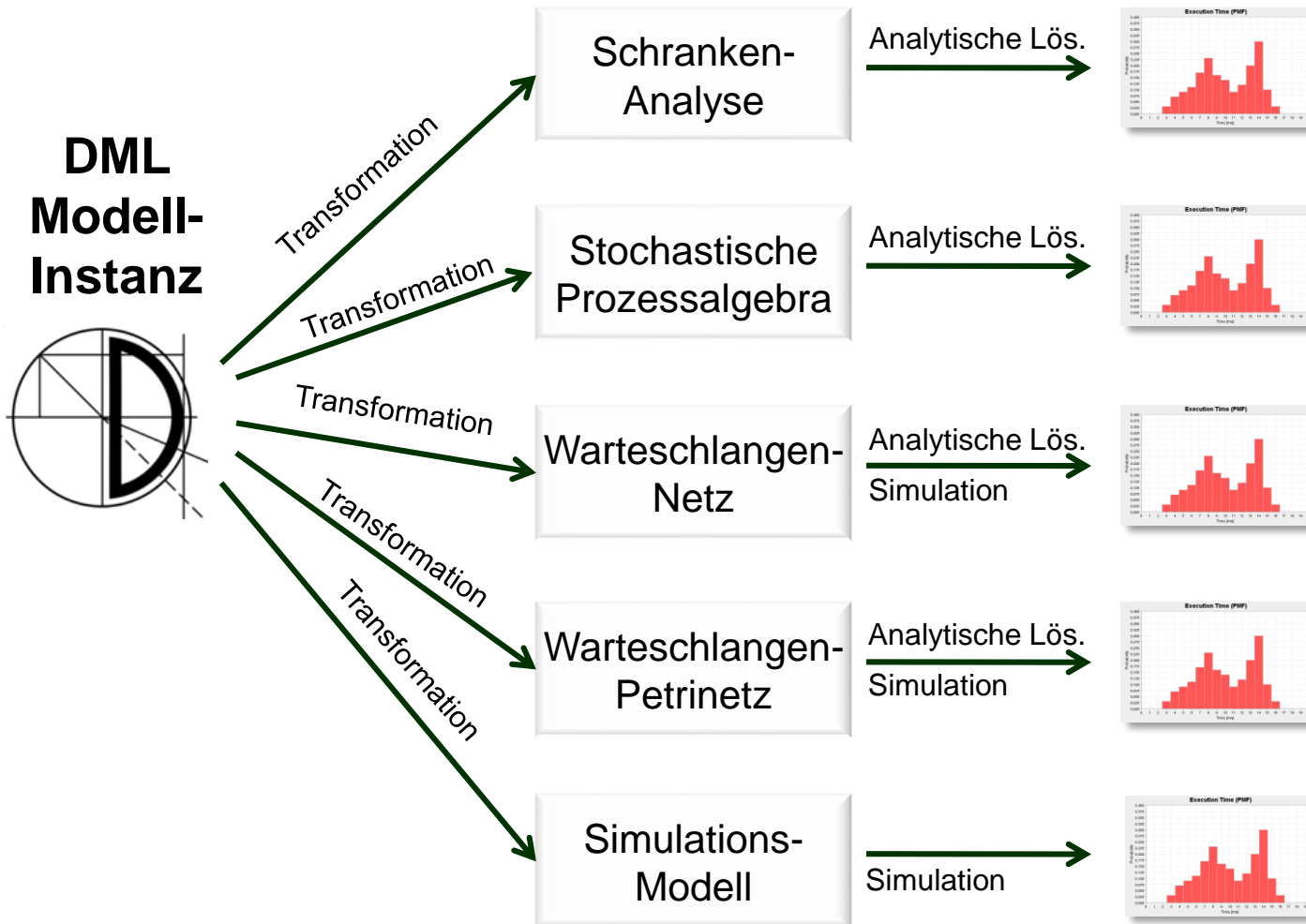
Beispiel: DML Model Instance



Beispiel: DML Model Instance



Performance-Vorhersage zur Laufzeit durch Modell-zu-Modell Transformationen



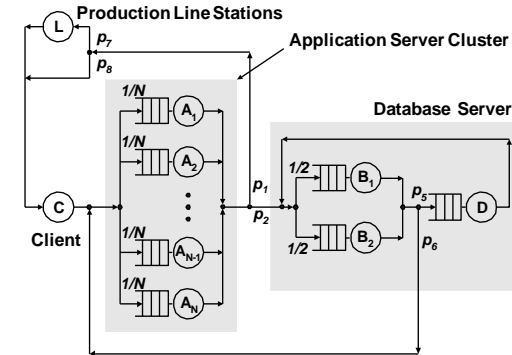
Beispiel-Transformationen

Schranken-Analyse

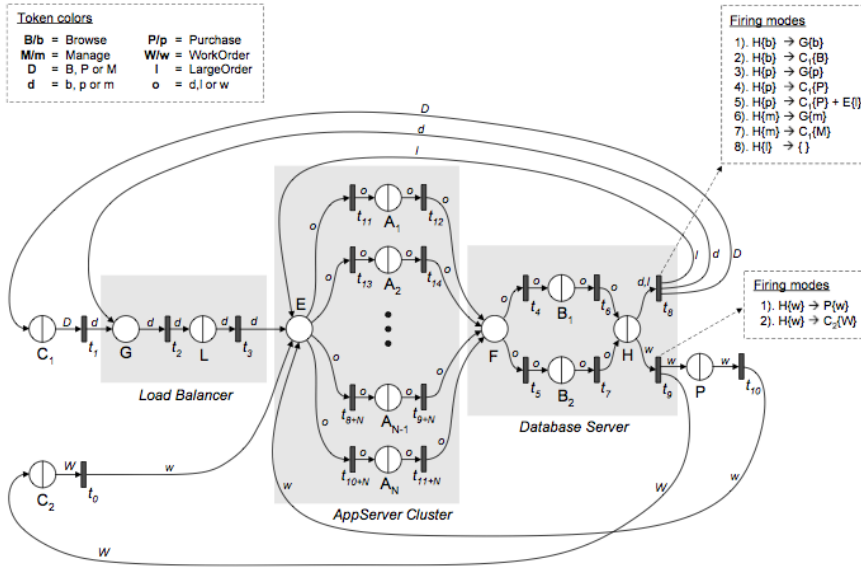
$$R \geq \max \left[N \times \max \{ D_i \}, \sum_{i=1}^K D_i \right] \quad X_0 \leq \min \left[\frac{1}{\max \{ D_i \}}, \frac{N}{\sum_{i=1}^K D_i} \right]$$

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

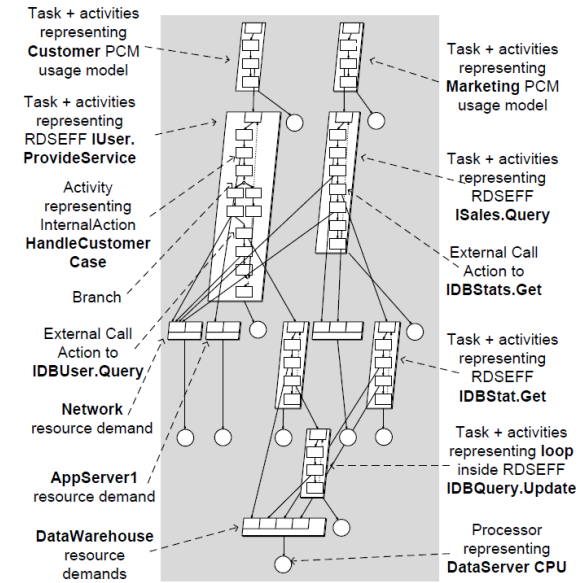
Warteschlangennetz



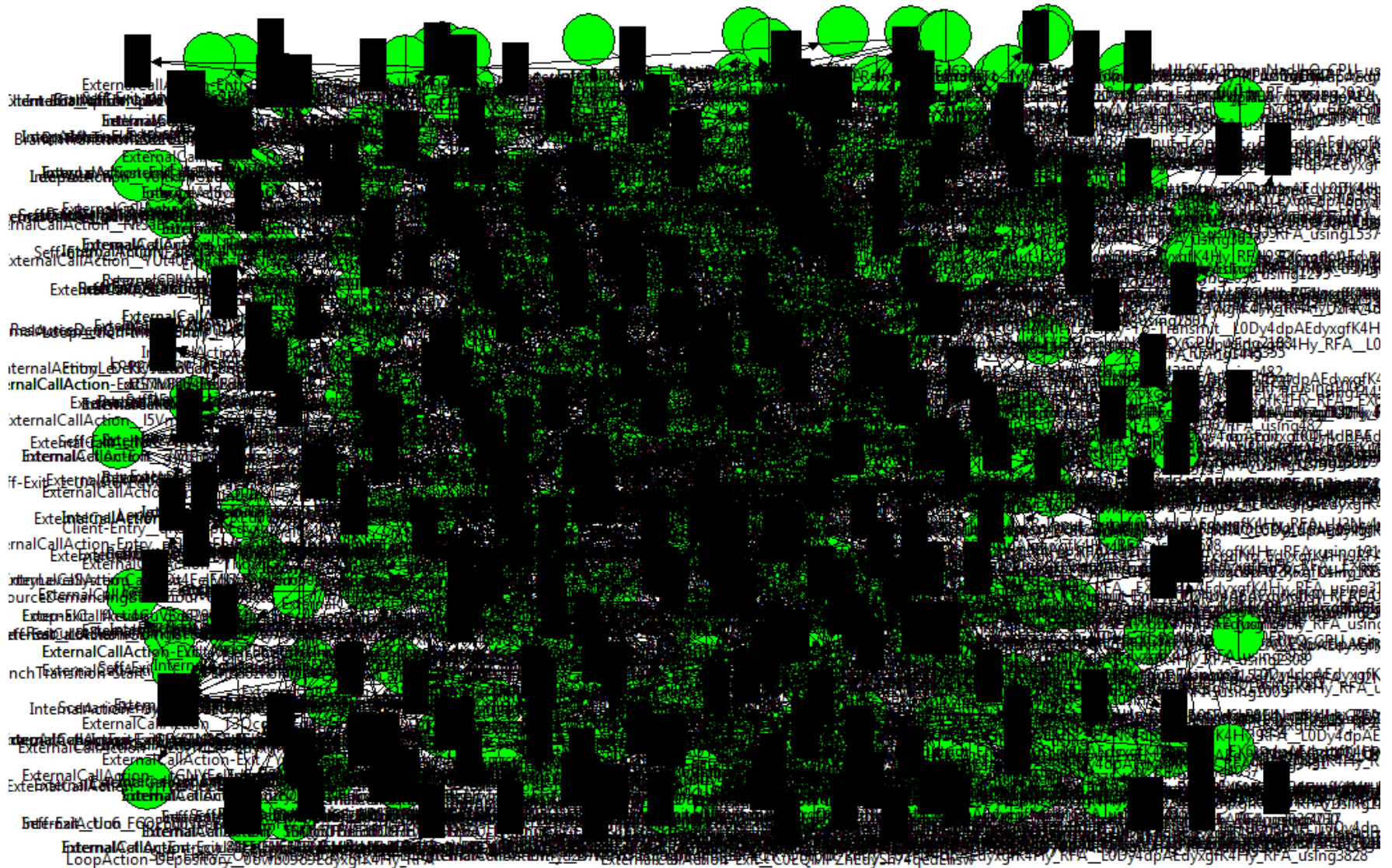
Warteschlangen-Petri-Netz (Queueing-Petri-Net)



Geschichtetes Warteschlangennetz (LQN)



Fallstudie: Prozess-Kontroll-System (ABB)



Descartes-Tool-Chain



<http://descartes.tools>

Julius-Maximilians-
UNIVERSITÄT WÜRZBURG

Chair of Computer Science II
Software Engineering

Imprint + Privacy Policy | Sitemap

« Fakultät für Mathematik und Informatik

« Institut für Informatik

« Lehrstuhl für Informatik II

News

People ▶

Research ▶

Publications ▶

Projects ▶

Tools ▼

DML Bench ▶

DNI

LIMBO ▶

WCF

LibReDE ▶

SPA

DQL

BUNGEE

hInjector

Descartes Tools

Below you see a list of the tools we develop. Please click on the tool name to get more information:

Descartes Modeling Language:

- [DML Specification](#)
Implementation in EMF (Eclipse Modeling Framework)
- [DML Bench](#)
- [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 \(Storage Performance Analyzer\)](#)

Declarative Performance Engineering:


- [DQL \(Descartes Query Language\)](#)

Benchmarking:





- [BUNGEE Cloud Elasticity Benchmark](#)
- [hInjector Hypercall Attack Injector](#)

Stochastic Modeling:

- [QPME \(Queueing Petri net Modeling Environment\)](#)



Important Links

- [SPEC Research Group](#)

- [Relate FP7 ITN](#)

- [Descartes Modeling Language \(DML\)](#)

- [Queueing Petrinet Modeling Environment \(QPME\)](#)

- [Interval Standard Working Group P1788](#)

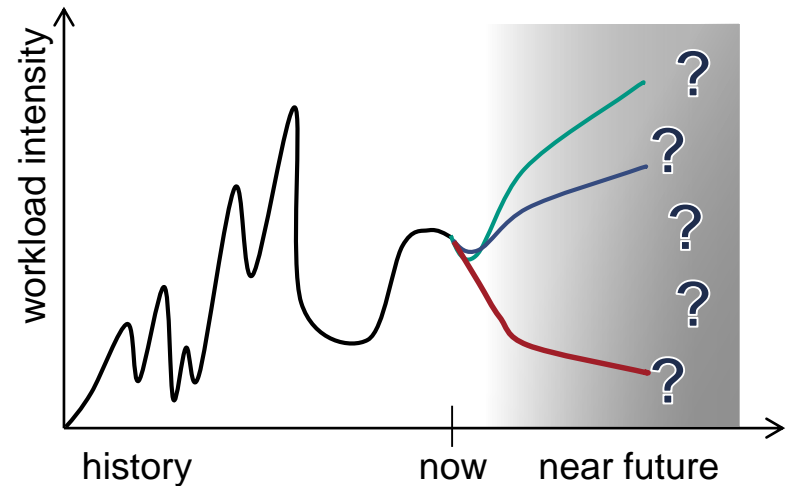
Upcoming Events

- [Int. Conference on Performance Engineering \(ICPE\)](#)
- [Dagstuhl Seminar on Self-Aware Computing](#)
- [Int. Conference on Autonomic](#)

- **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** – Dependability benchmarking tool ([homepage](#), [publications](#))
- Queueing Petri Net Modeling Environment (QPME)

Beispiel-Tool: LIMBO

- **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>



Beispiel-Tool: LibReDE

- **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](#)]

Verwendete Methoden

Descriptive Architecture-level Models

- OMG Meta Object Facility (MOF)
 - MOF-based meta-models
- (UML MARTE)
- (UML SPT)

Predictive Models

- Bounding techniques
- Operational analysis
- Statistical regression models
- Stochastic process algebras
- (Extended) queueing networks
- Layered queueing networks
- Queueing Petri nets
- Reinforcement learning models
- Detailed simulation models

Workload Forecasting

AR(I)MA

Extended exp. smoothing

tBATS

Croston's method

Cubic smoothing splines

Neural network-based

Resource Demand Estimation

Regression-based techniques

Kalman filter

Nonlinear optimization

Maximum likelihood estimation

Independent component analysis

Regression Analysis

MARS

CART

M5 trees

Cubist forests

Quantile regression forests

Support vector machines

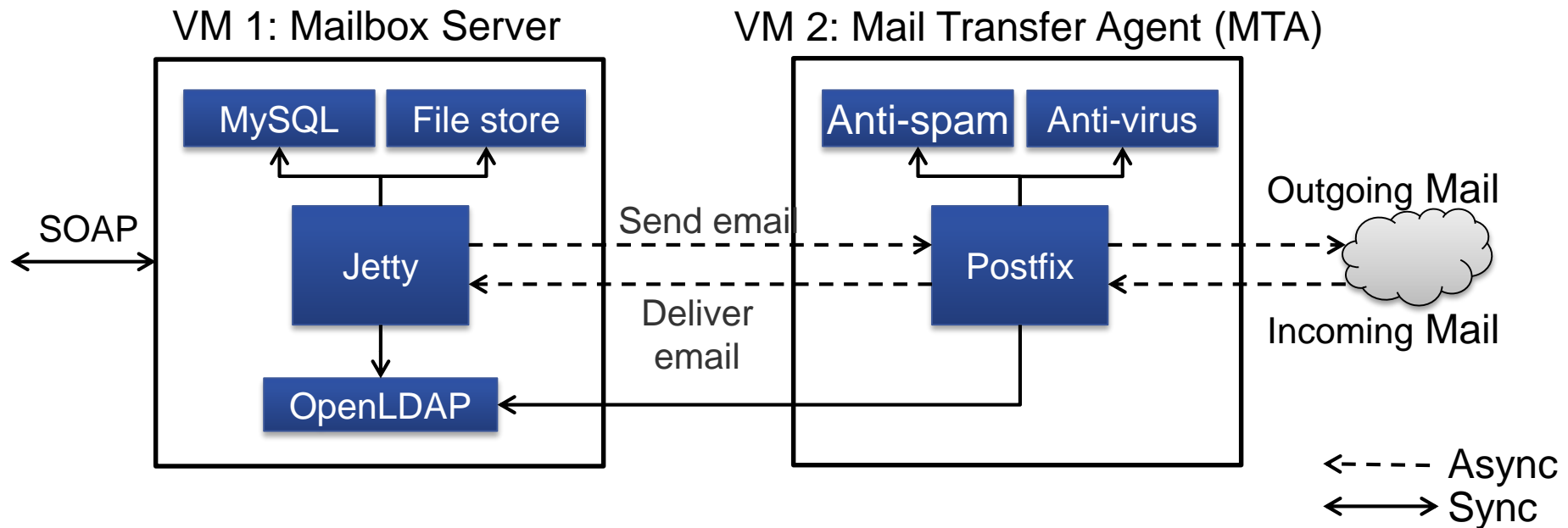


Gliederung

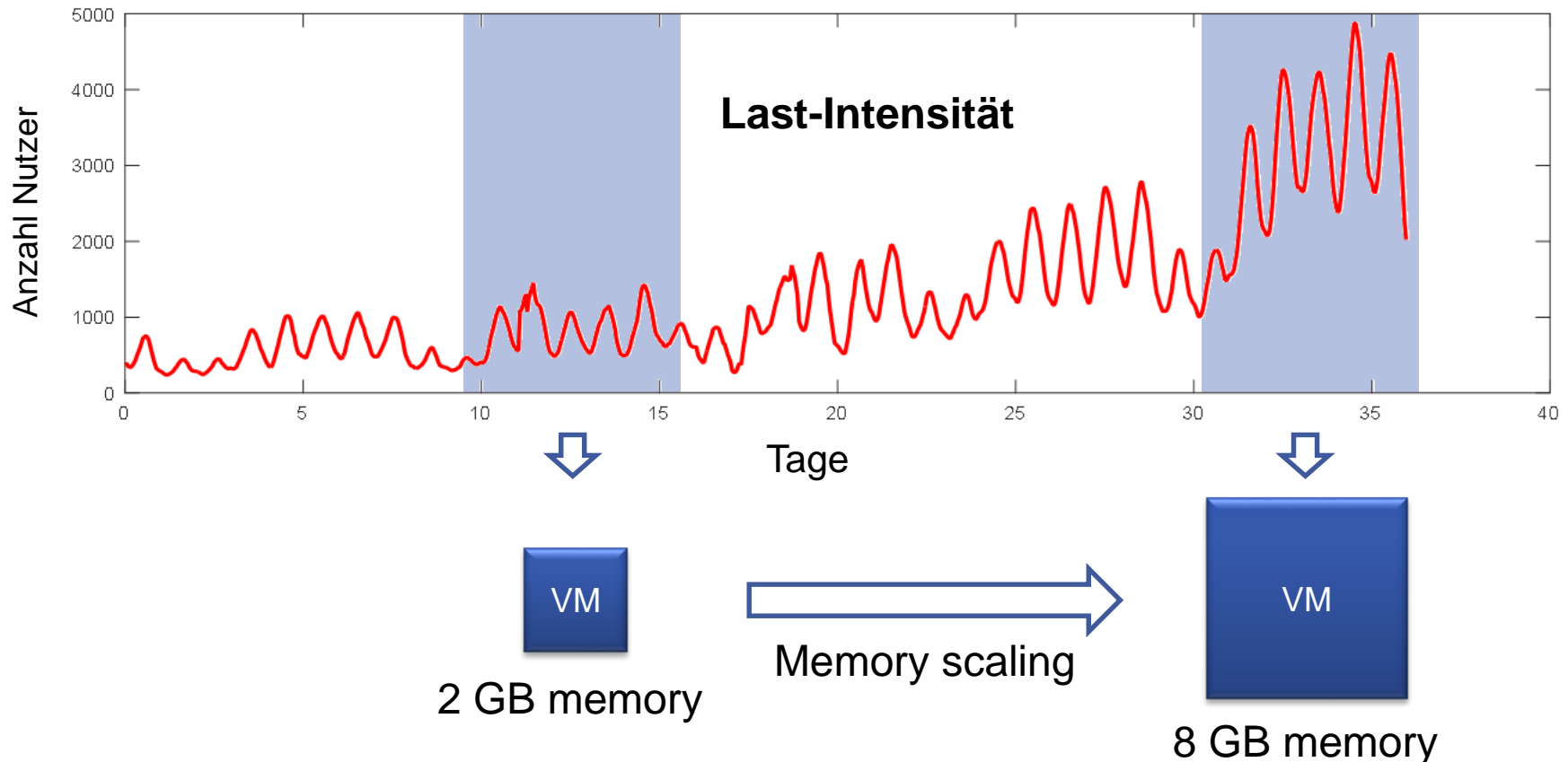
1. Einführung
2. Forschungsfragen
3. Wissenschaftlicher Ansatz
4. Ausgewählte Fallstudie
5. Vision & Ausblick

Fallstudie: Proaktive Speicher-Skalierung

- Zimbra collaboration server
- Idee: “Leverage **memory hot-add** mechanisms of modern hypervisors to **avoid service unavailability** due to memory bottlenecks”



Fallstudie: Proaktive Speicher-Skalierung

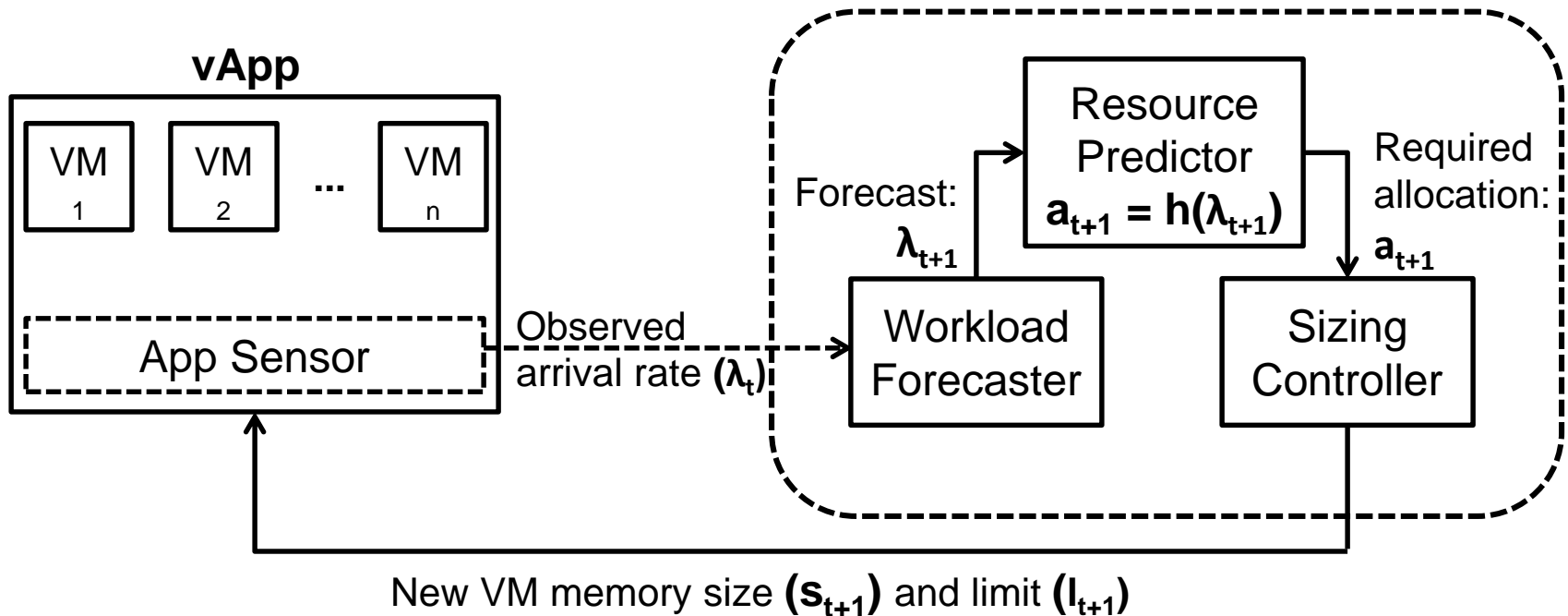


VM Hauptspeicher Allokation:

- Zu wenig → schlechte Performance, Verfügbarkeit / Zuverlässigkeit
- Zu viel → Ineffizienz, hohe Kosten

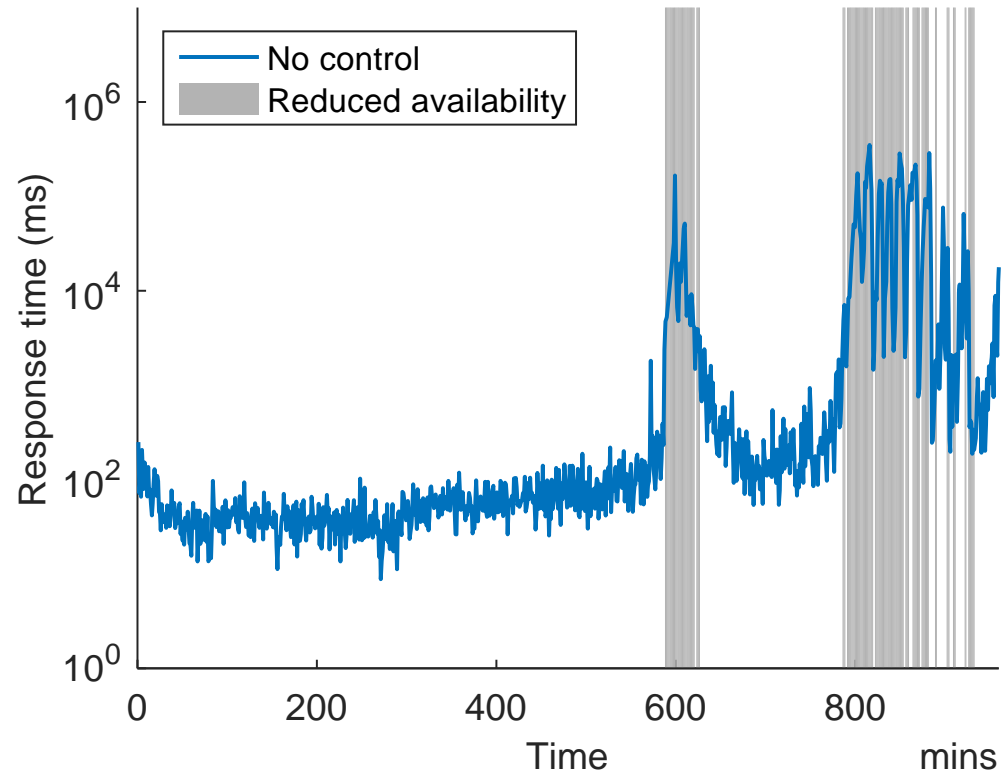
Ansatz

- Vorhersage der Last-Intensität (workload)
- Proaktive Rekonfiguration der Hauptspeicher-Zuweisung



Beispiel Szenario

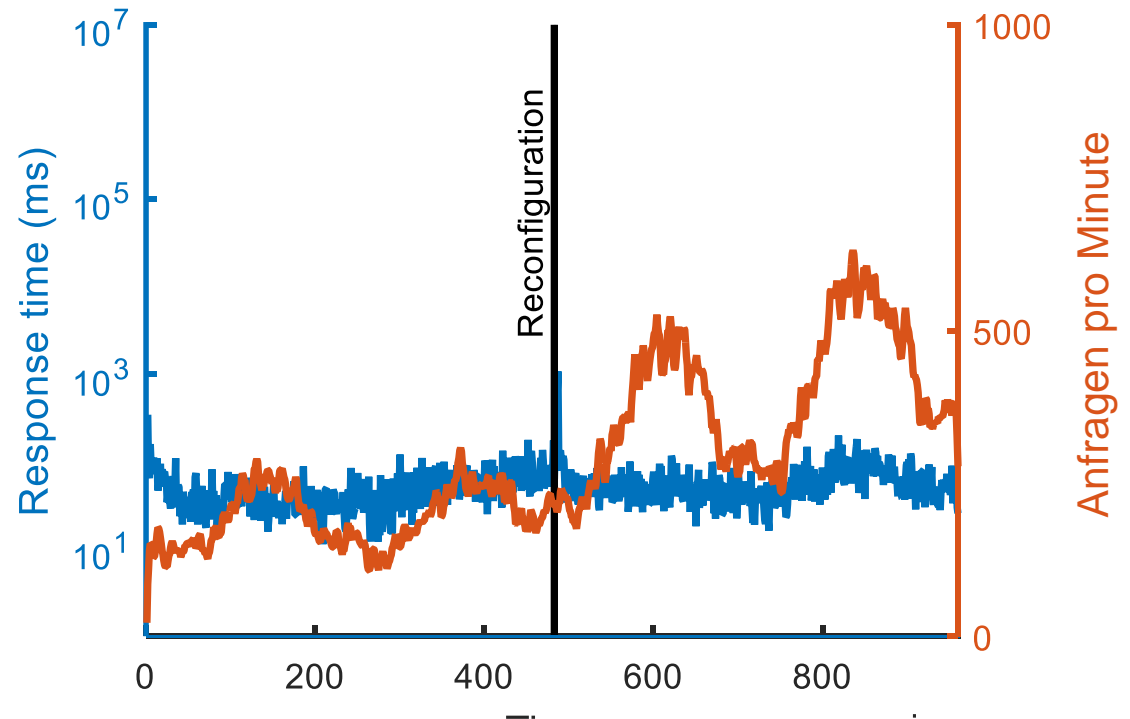
— Antwortzeit
■ schlechte Verfügbarkeit



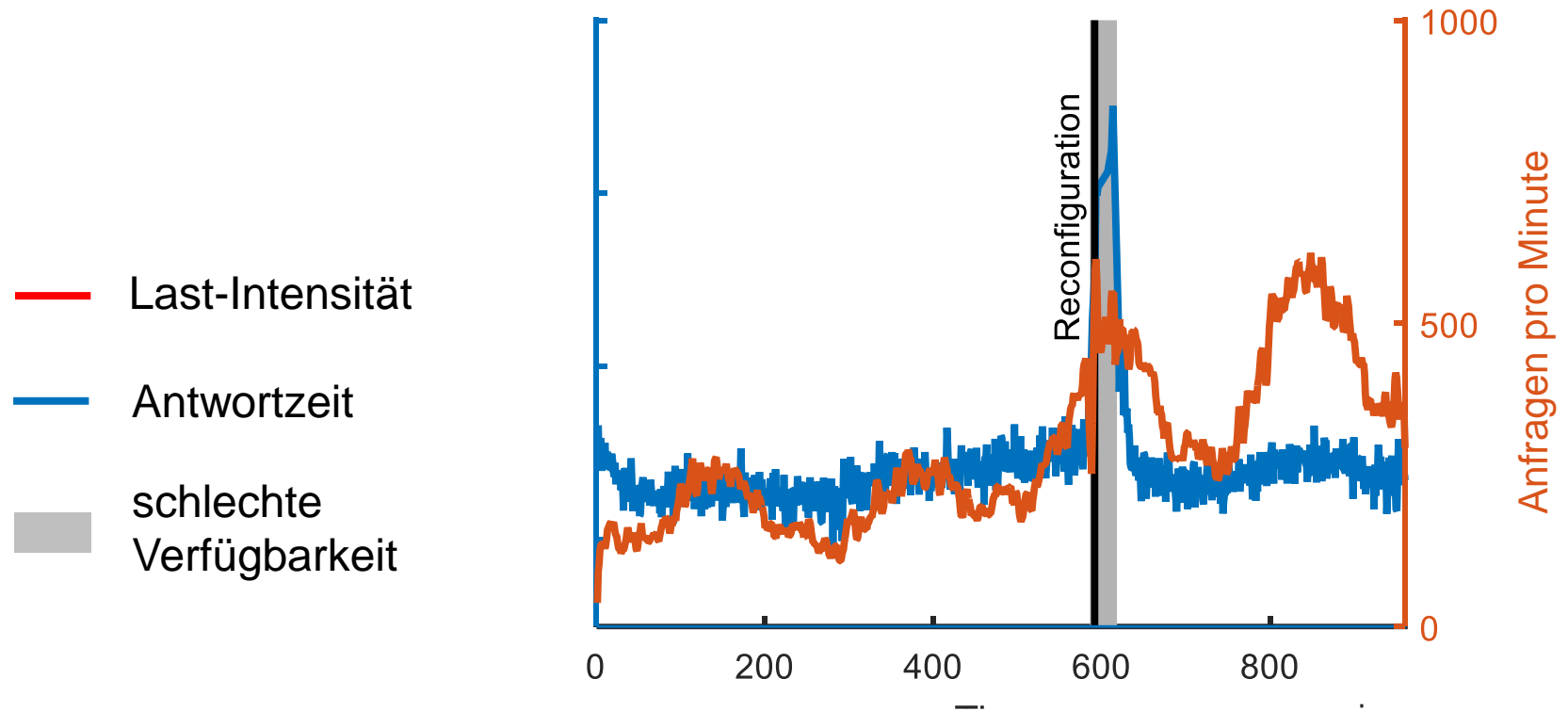
→ 176 Minuten schlechte Verfügbarkeit (timeouts)

Evaluation: „Proactive Controller“

- Last-Intensität
- Antwortzeit
- schlechte Verfügbarkeit



Evaluation: „Reactive Controller“



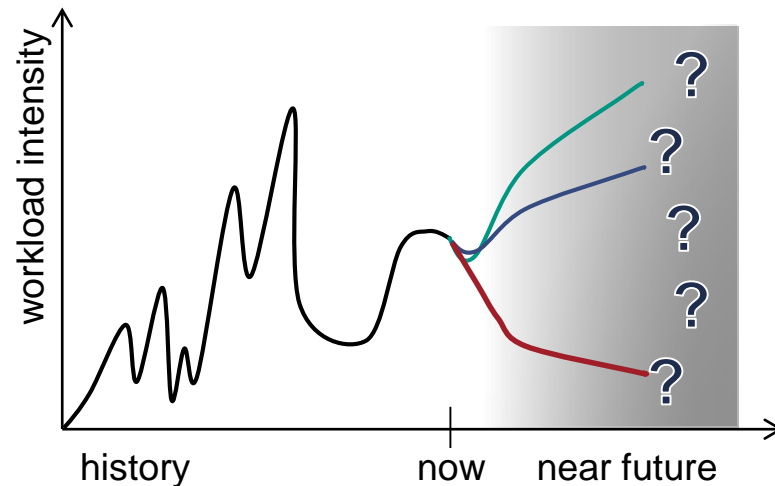
→ 33 Minuten schlechte Verfügbarkeit (timeouts)

Evaluation: Zusammenfassung

	No control	Reactive	Proactive
Mean Resp. Time [ms]	7,567	1,211	52
Timeouts	84	285	0
Errors	8593	1485	337
Time of reduced availability [min]	176	33	4

LIMBO Tool

- Modellierung und Vorhersage von Last-Intensität
 - z.B. Anfragen / Aufträge pro Sekunde

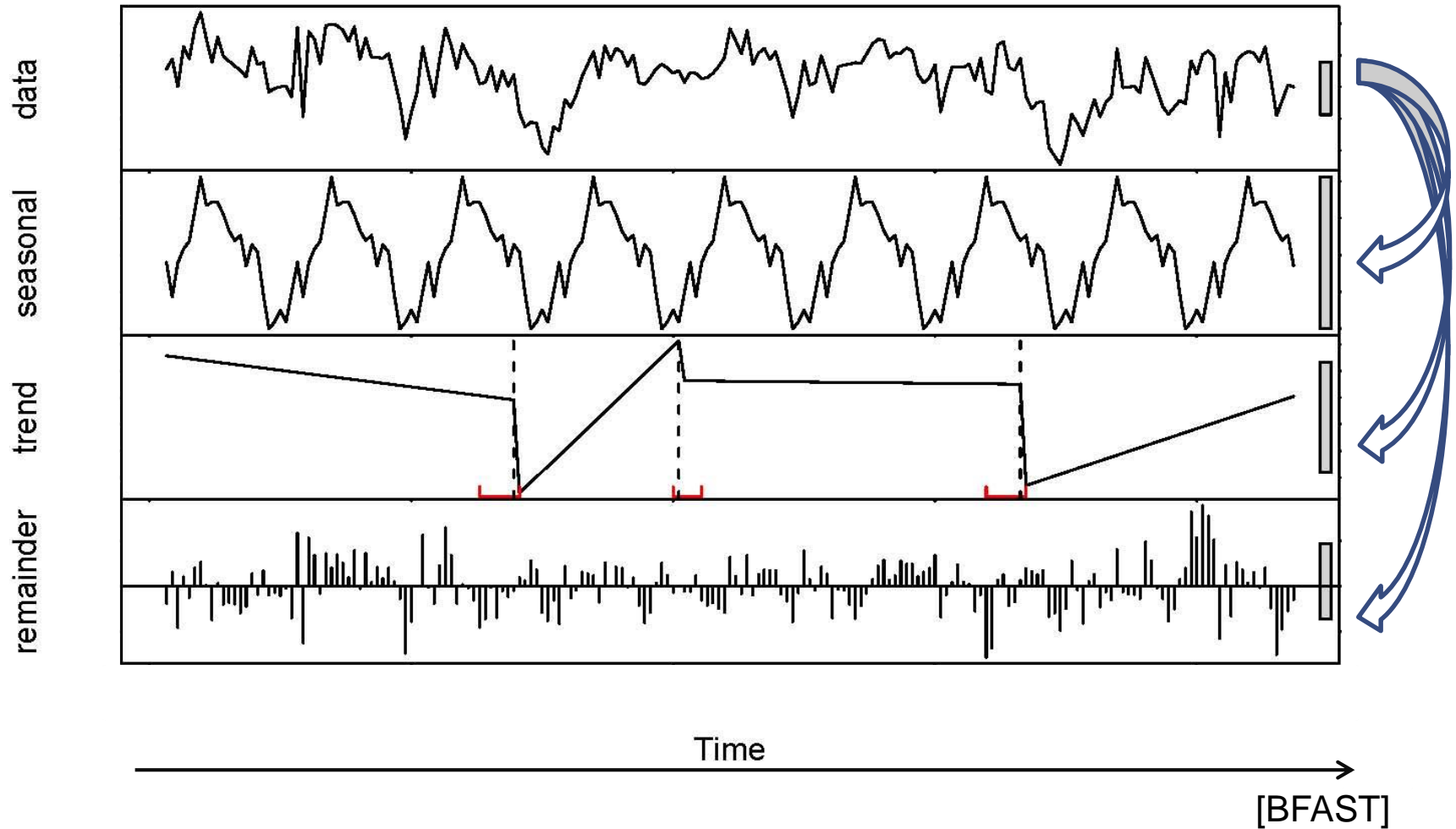


<http://descartes.tools/limbo>

<http://descartes.tools/wcf>



LIMBO: Zeitreihen-Analyse



LIMBO: Verwendete Methoden

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]

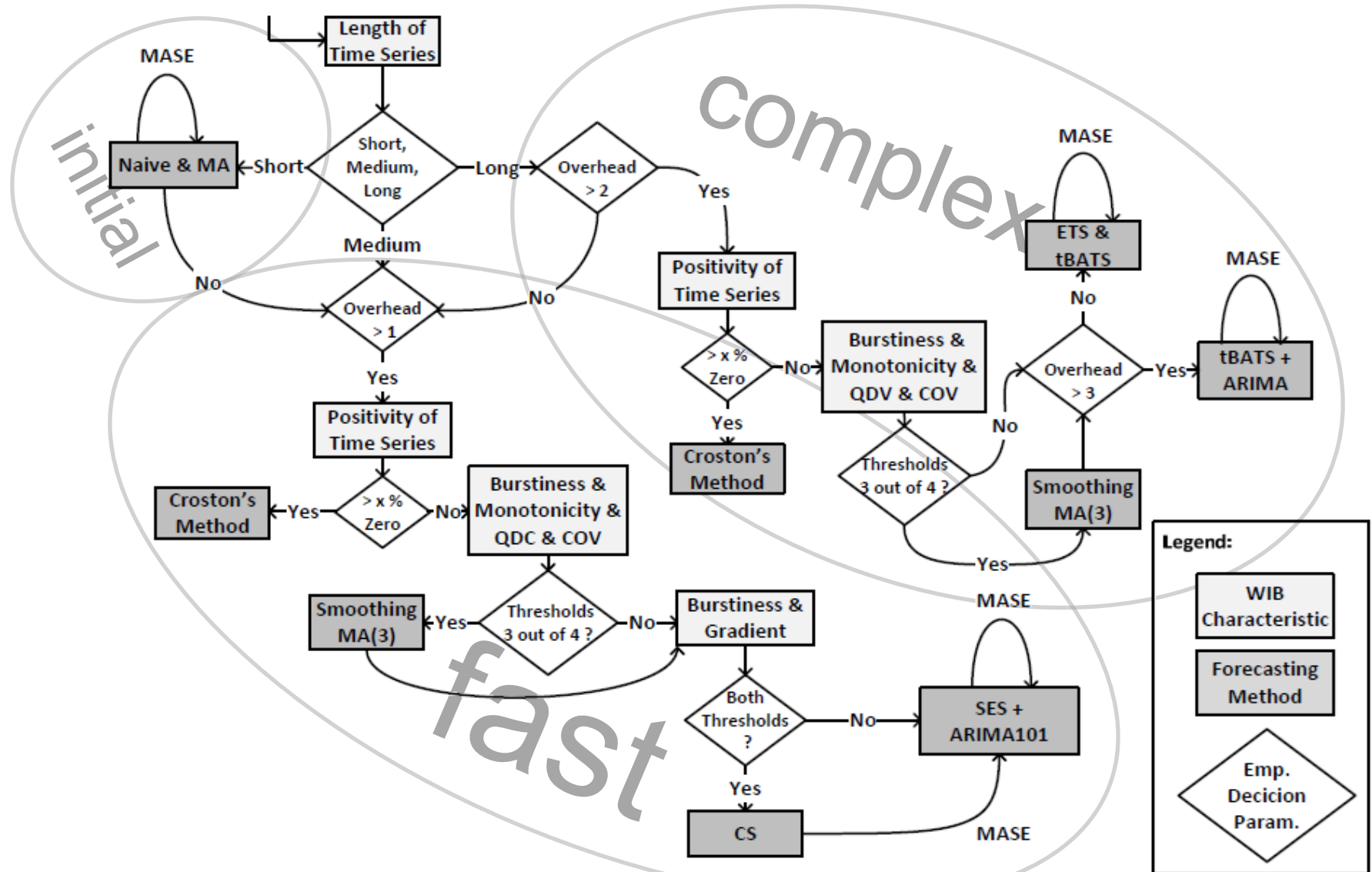
Estimation and Modeling of Seasonal Patterns (complex)

Extended Exponential Smoothing (ETS) [Hynd08, Hyn08]

ARIMA framework with automatic model selection [Box08, Hynd08]

tBATS for complex seasonal patterns [Live11]

LIMBO: Auswahl der Methode



Gliederung

1. Einführung
2. Forschungsfragen
3. Wissenschaftlicher Ansatz
4. Ausgewählte Fallstudie
5. Vision & Ausblick

Dagstuhl Seminar „Self-Aware Computing“

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

Organizers

Samuel Kounev (Universität Würzburg, DE)

Jeffrey O. Kephart (IBM TJ Watson Research Center, US)

Marta Kwiatkowska (University of Oxford, GB)

Xiaoyun Zhu (VMware, Inc., US)

Community:

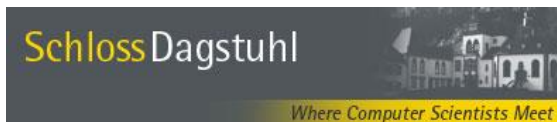
<http://descartes.tools/self-aware>

Dagstuhl Report:

<http://drops.dagstuhl.de/opus/volltexte/2015/5038/>

Seminar Page:

<http://www.dagstuhl.de/15041>



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, X. Zhu, J. O. Kephart and M. Kwiatkowska, editors. **Model-driven Algorithms and Architectures for Self-Aware Computing Systems** (Dagstuhl Seminar 15041). Dagstuhl Reports, vol. 5, No. 1. pp. 164-196, Dagstuhl, Germany, 2015. <http://drops.dagstuhl.de/opus/volltexte/2015/5038>

Community page: <http://descartes.tools/self-aware>

Extended Definition

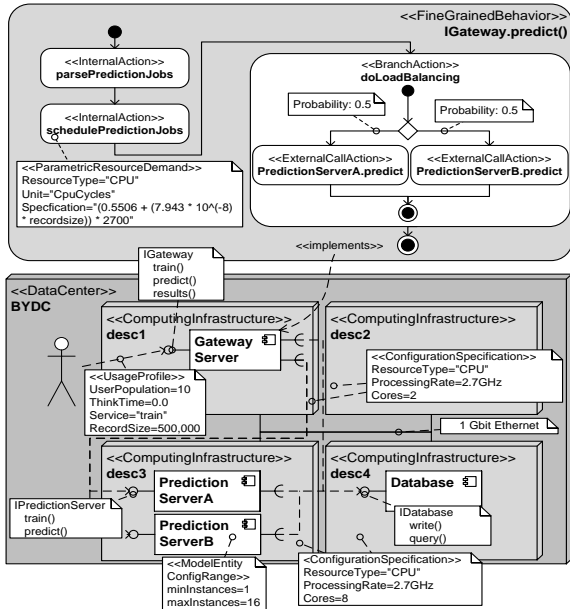
Self-aware Computing Systems are computing systems that:

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)

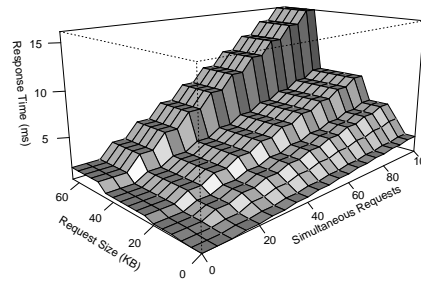
in accordance with **higher-level goals**, which may also be subject to change.



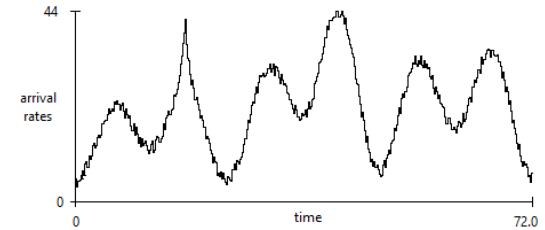
Examples of Models



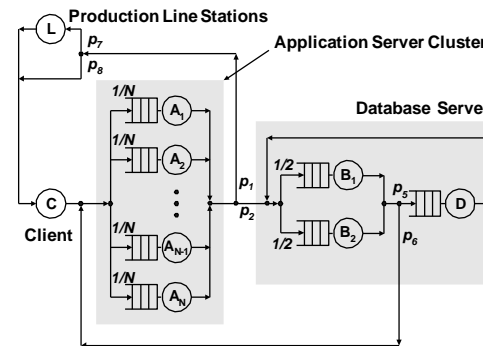
Descriptive MOF-based models



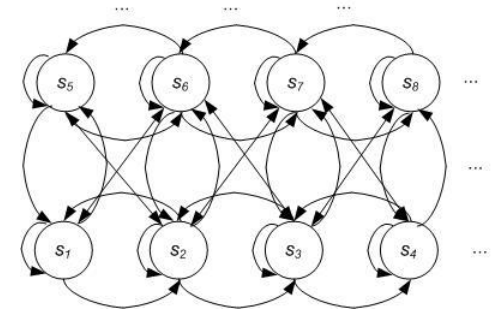
Statistical regression models



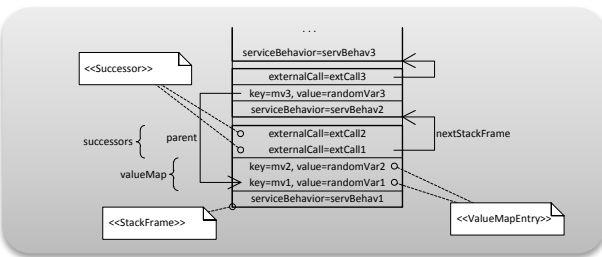
Load forecasting models



Queueing network models



Markov models



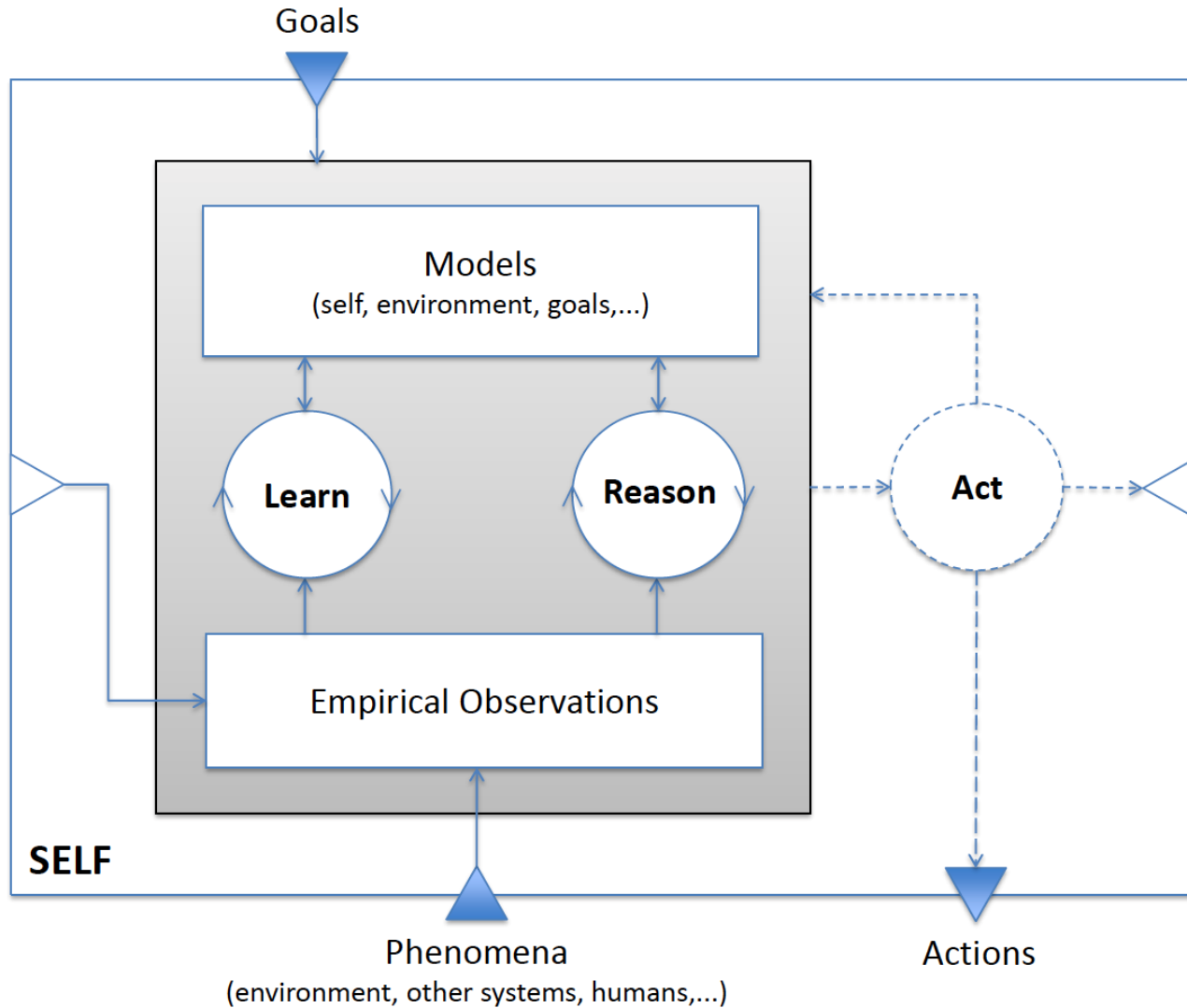
Simulation models

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

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

Analytical analysis models

Self-Aware Learning & Reasoning Loop



The Vision



Self-Aware Computing

Laufende Initiativen

Springer-Buch „Self-Aware Computing Systems“

Samuel Kounev (Universität Würzburg, DE)

Jeffrey O. Kephart (IBM TJ Watson, US)

Marta Kwiatkowska (University of Oxford, GB)

Xiaoyun Zhu (VMWare, Inc. – Palo Alto, US)

Aleksandar Milenkoski (Universität Würzburg, DE)



Antrag **DFG-Schwerpunktprogramm** in Arbeit

Samuel Kounev, Universität Würzburg (Coordinator)

Kurt Geihs, Universität Kassel

Lars Grunske, Humboldt-Universität Berlin

Sabina Jeschke, RWTH Aachen

Gisela Lanza, Karlsruhe Institute of Technology



July 19-22, 2016 – Würzburg – Germany

ICAC 2016

The 13th IEEE International Conference on Autonomic Computing



ICAC 2016 Call for Papers

<http://icac2016.uni-wuerzburg.de>

13th IEEE International Conference on Autonomic Computing (ICAC 2016)
Würzburg, Germany, July 19-22, 2016



ICAC 2016 Workshops

1. 4th Self-IoT: Self-Aware Internet of Things
2. 2nd DAS: Distributed Adaptive Systems
3. SISSY: Workshop on Self-Improving System Integration
4. International Workshop on Models@run.time
5. Feedback Computing 2016
6. Self-Organizing Self-Managing Clouds Workshop (SOSeMC 2016)
7. International Workshop on “Spacecraft Autonomy”
8. Workshop on Self-Adaptivity and Security
9. Annual Meeting of SPEC RG DevOps Performance Working Group

Vielen Dank!



<http://www.descartes-research.net>

<http://descartes.tools>