Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Descartes research

Stochastic Models for Self-Aware Computing in Data Centers

Samuel Kounev

Chair of Software Engineering
University of Würzburg

http://descartes-research.net/
http://descartes.tools/

ASMTA 2017 Keynote, Newcastle-upon-Tyne, UK, July 10, 2017

Selected References

- 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
- 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]

Dagstuhl-Seminar

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

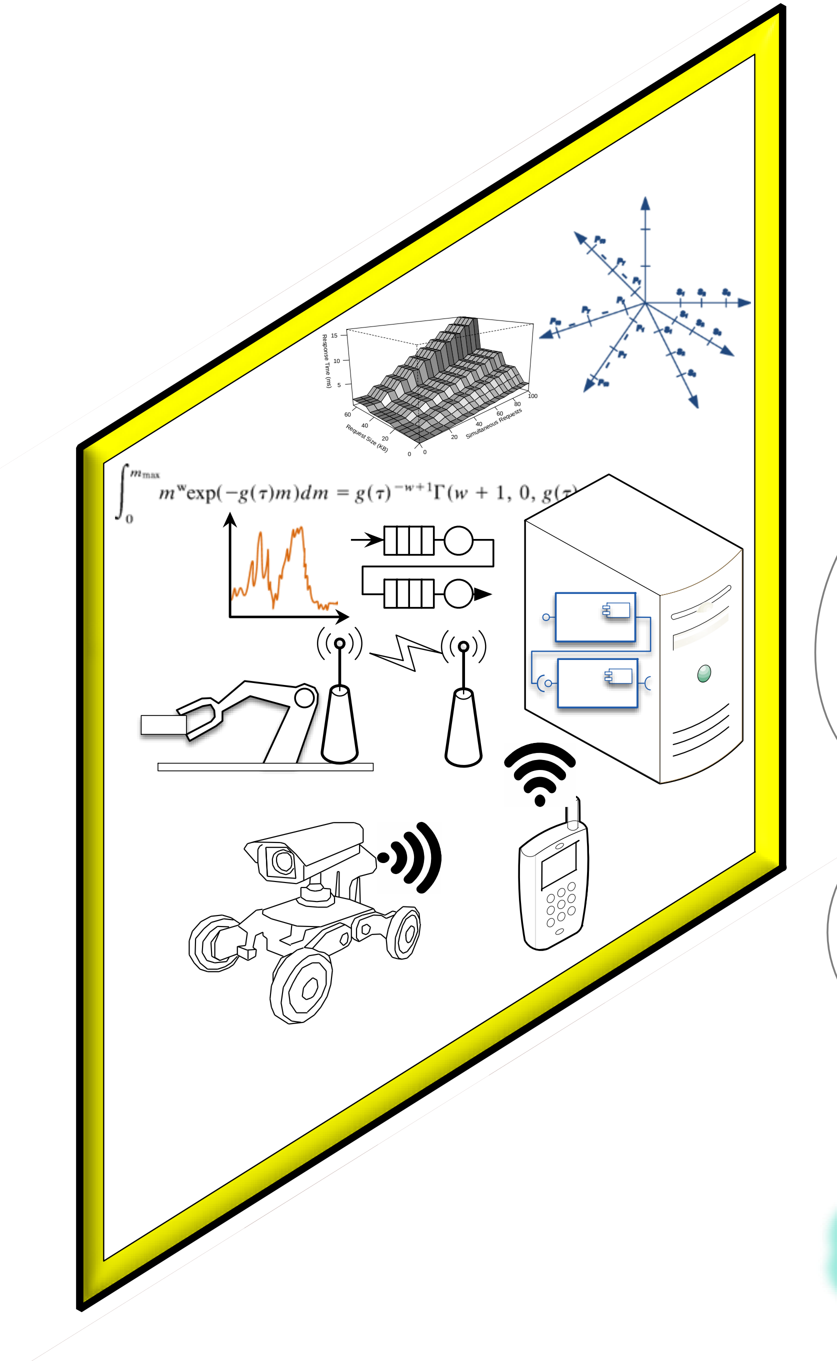
Organizers

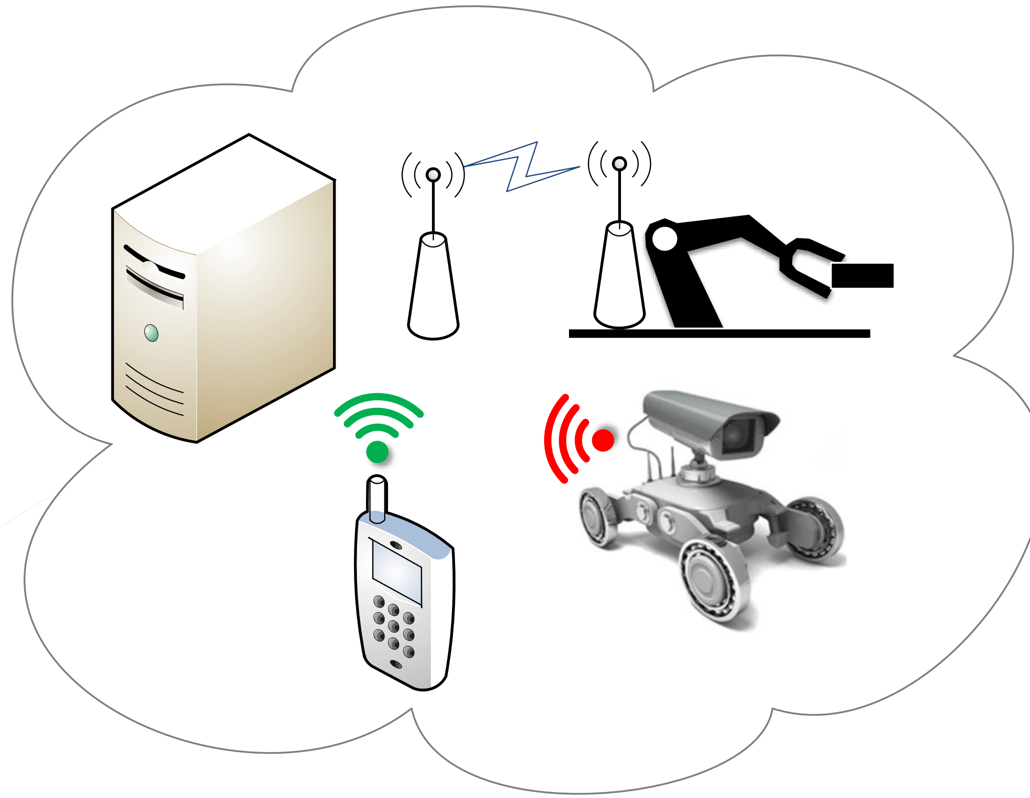
Jeffrey O. Kephart (IBM TJ Watson Research Center, US)
Samuel Kounev (Universität Würzburg, DE)
Marta Kwiatkowska (University of Oxford, GB)
Xiaoyun Zhu (VMware, Inc., US)

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

The Vision

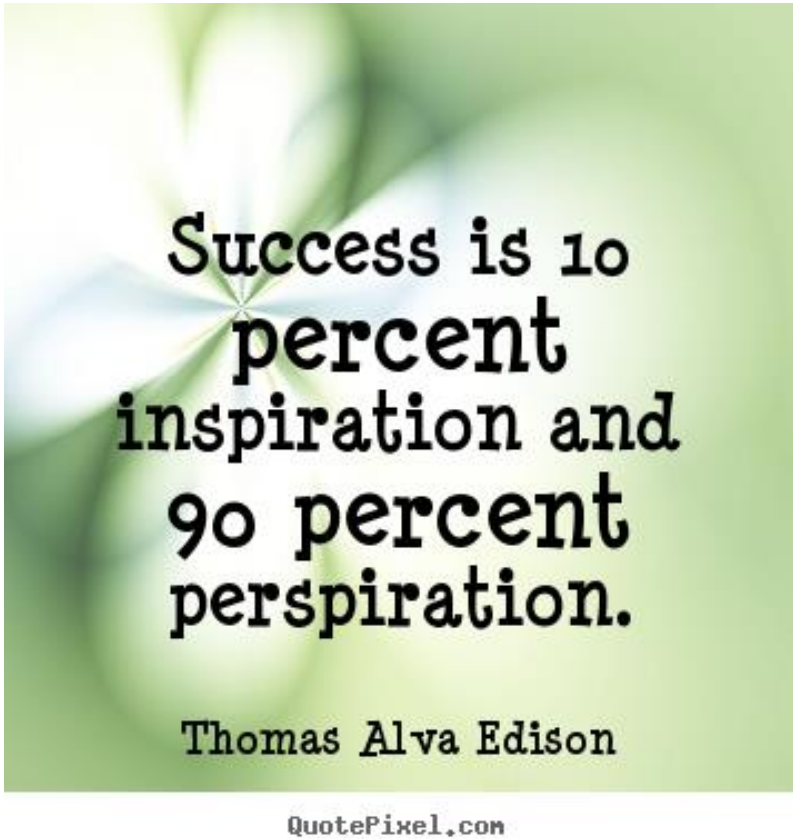




Self-Aware Computing

Inspiration vs. Perspiration

- „Whoever has visions should go to the doctor“



Helmut Schmidt

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

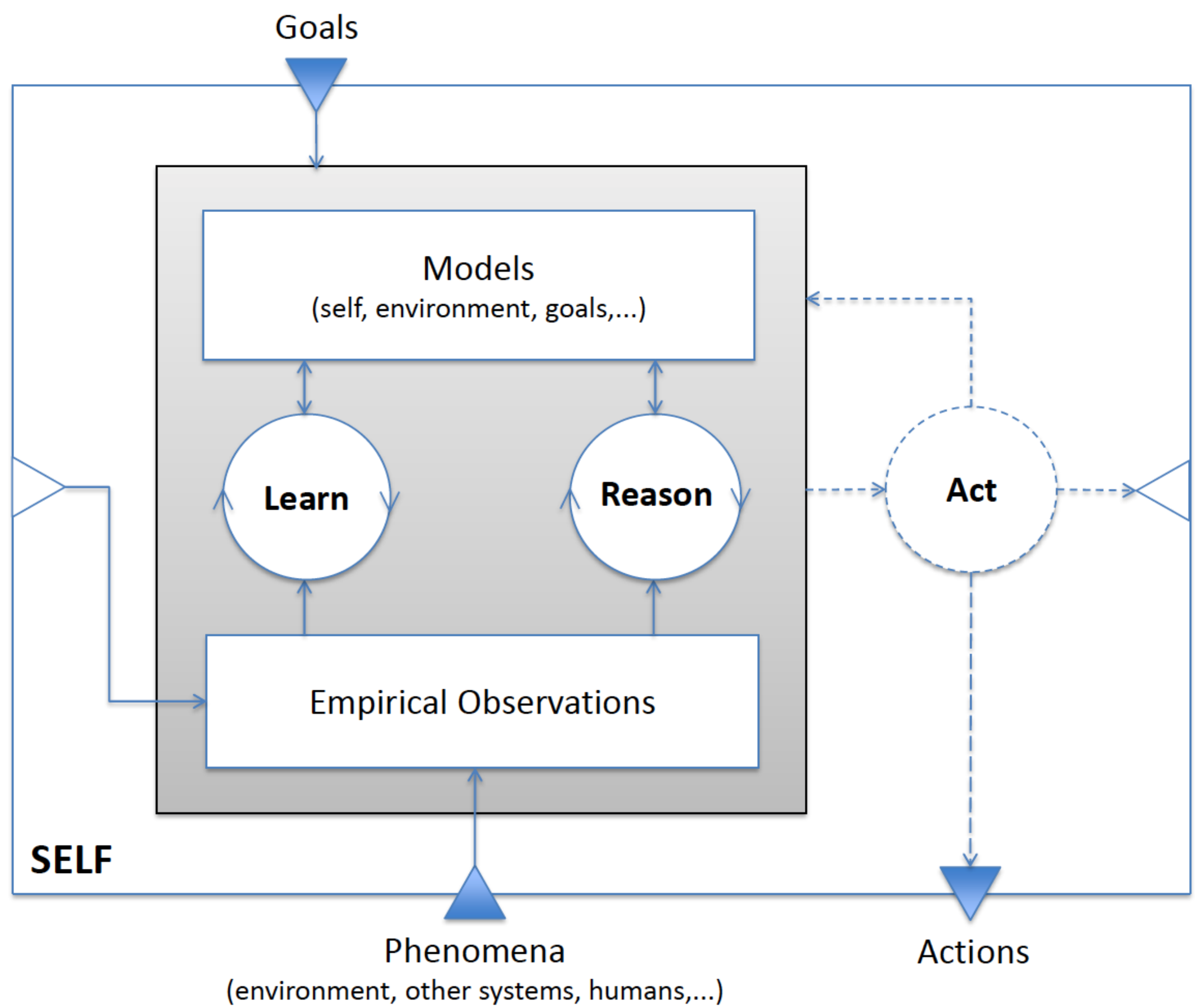
Self-aware Computing Systems are computing systems that:

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

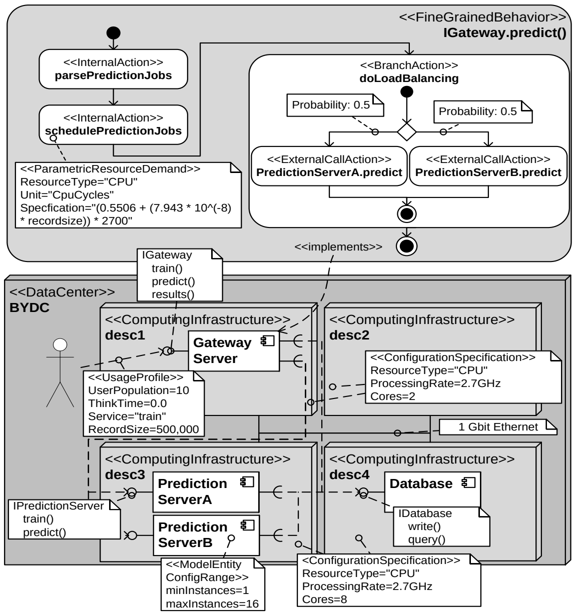
Descriptive Models

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

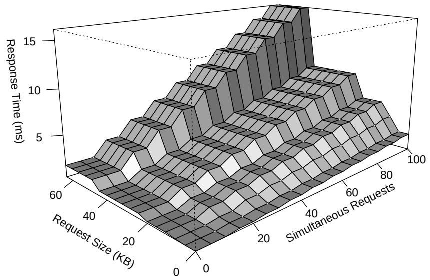
(Predictive) Analysis Models

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

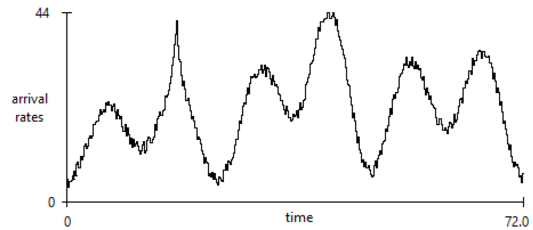
Examples of Models



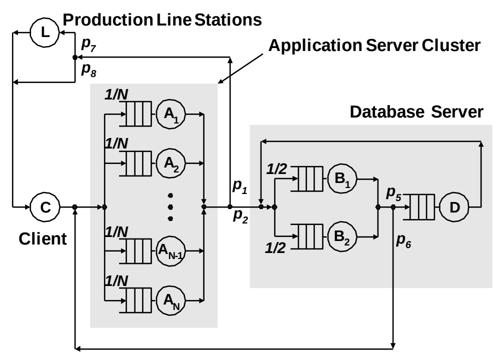
Descriptive MOF-based models



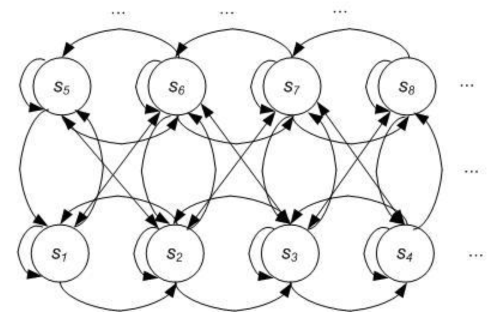
Statistical regression models



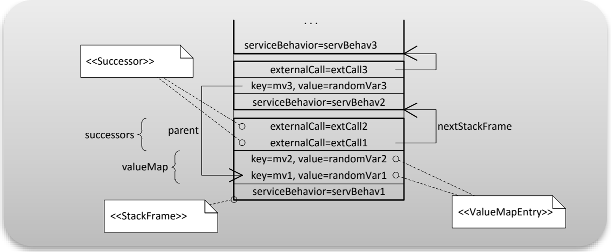
Load forecasting models



Queueing network models



Markov models



Simulation models

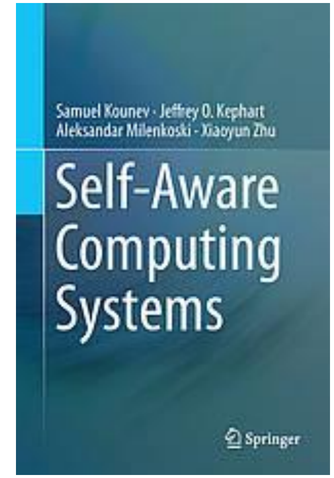
$$R \geq \max\left[N \times \max\{D_i\}, \sum_{i=1}^{K} D_i\right] \qquad X_0 \leq \min\left[\frac{1}{\max\{D_i\}}, \frac{N}{\sum_{i=1}^{K} D_i}\right]$$

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

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

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Descartes research

BACK TO:
Self-Aware Computing in Data Centers

Main References

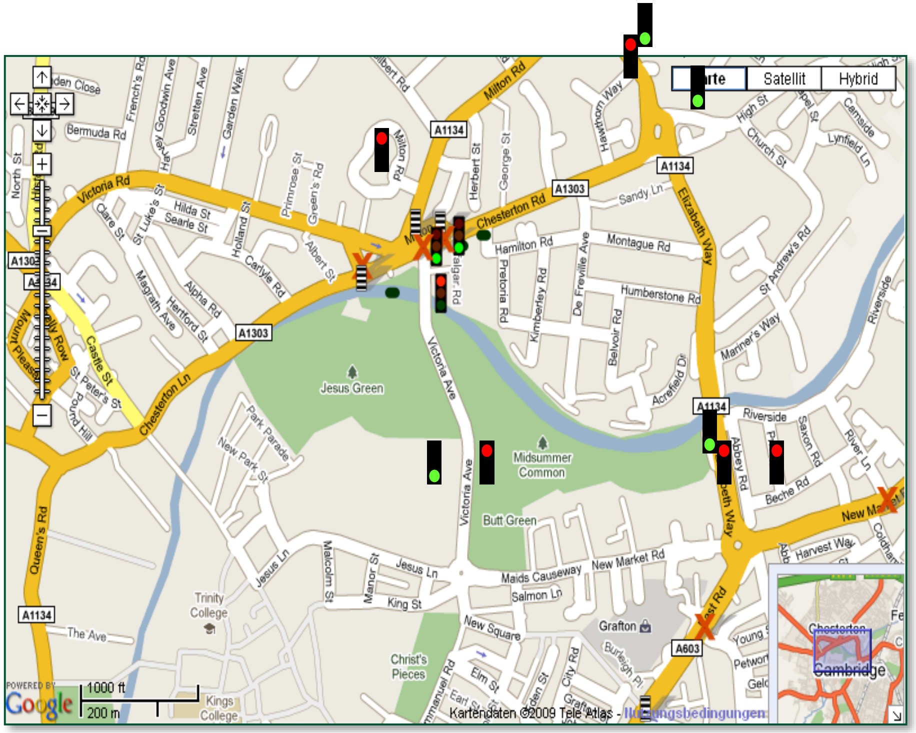
S. Kounev, N. Huber, F. Brosig, and X. Zhu. ***A Model-Based Approach to Designing Self-Aware IT Systems and Infrastructures***. IEEE Computer, 49(7):53–61, July 2016.

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.

See also Tutorial at ICPE 2017 →
Slides available at http://descartes.tools

Motivating Example

Traffic Monitoring System

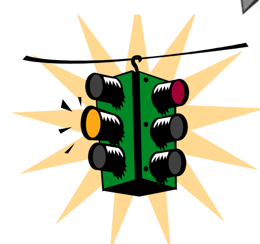


UNIVERSITY OF CAMBRIDGE

Induction Loops

GPS Sensors

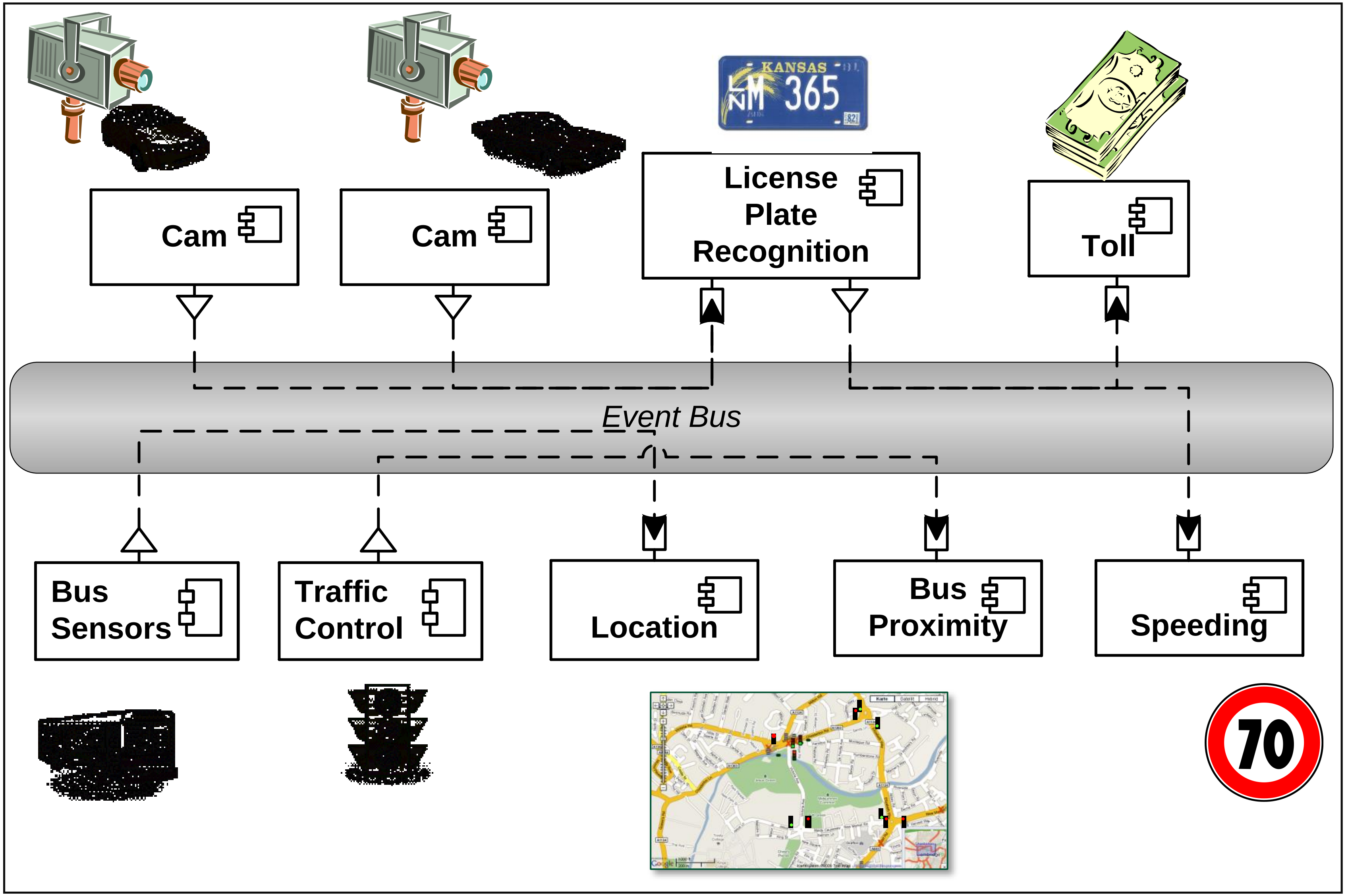
Traffic Cameras



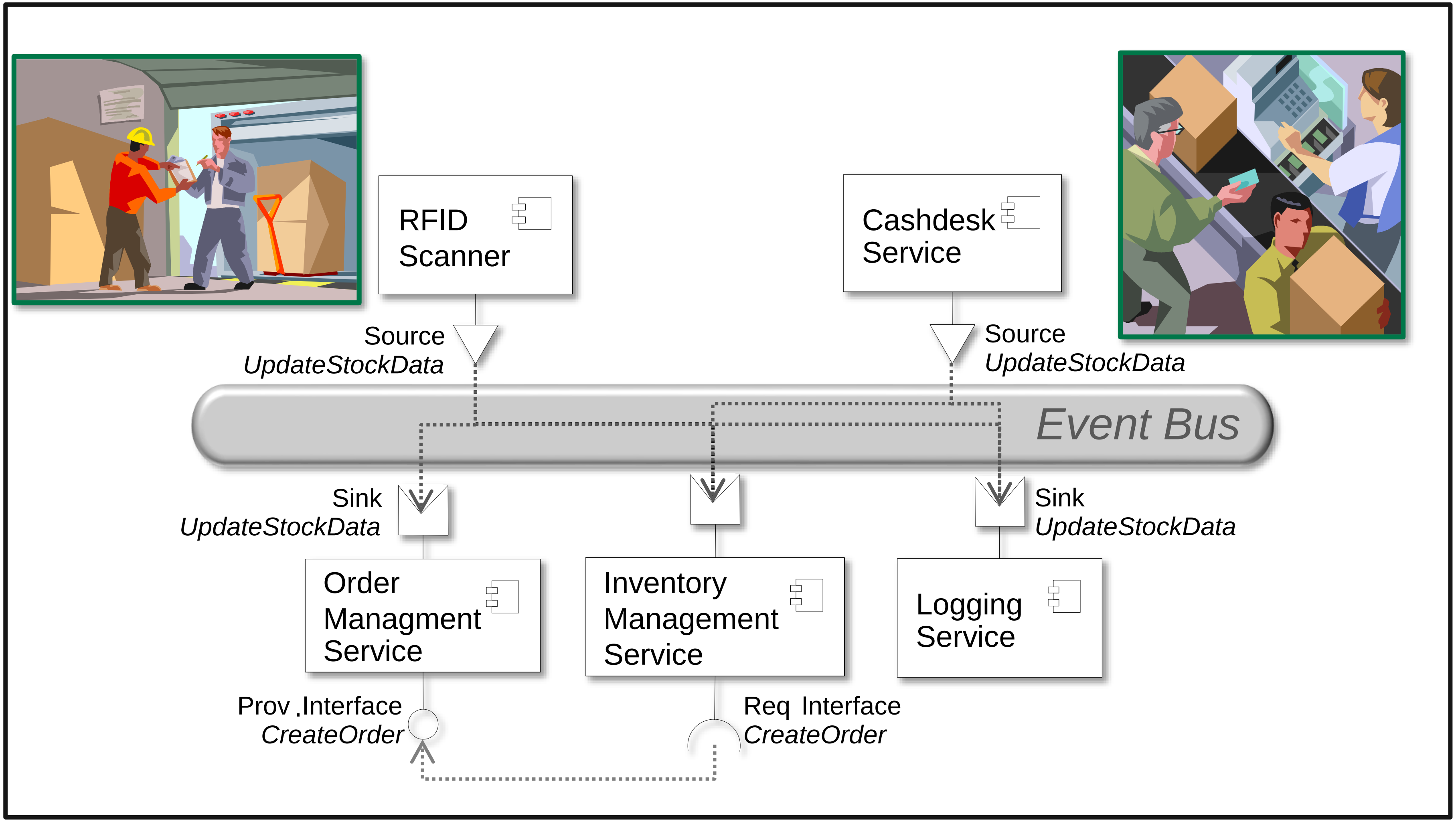
Traffic Light Status

http://www.cl.cam.ac.uk/research/time/

Ex 1: Traffic Monitoring System

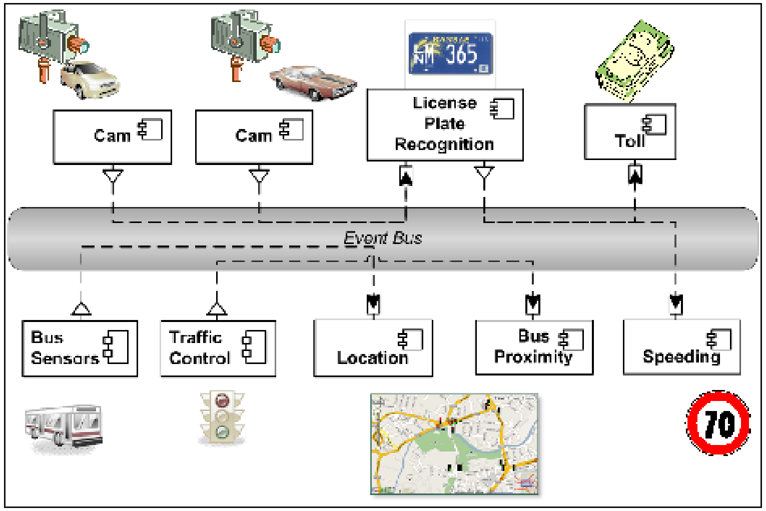


Ex 2: Inventory Management System

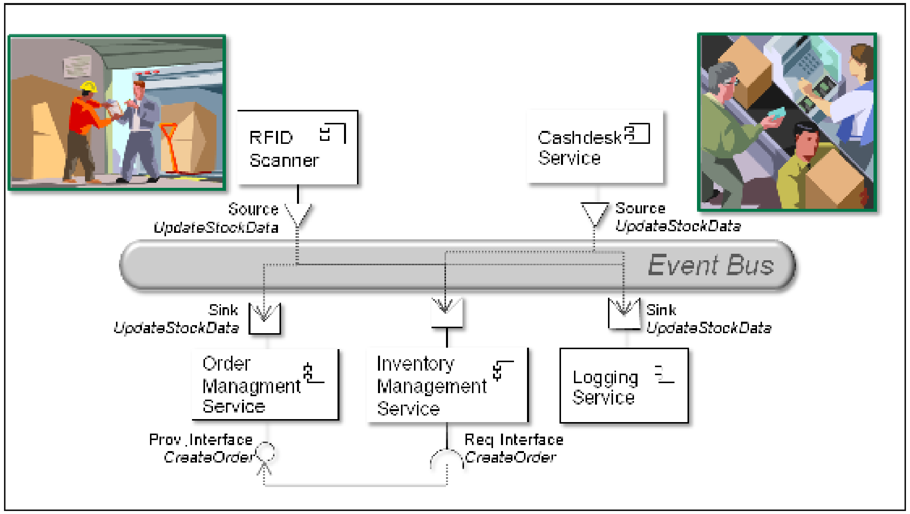


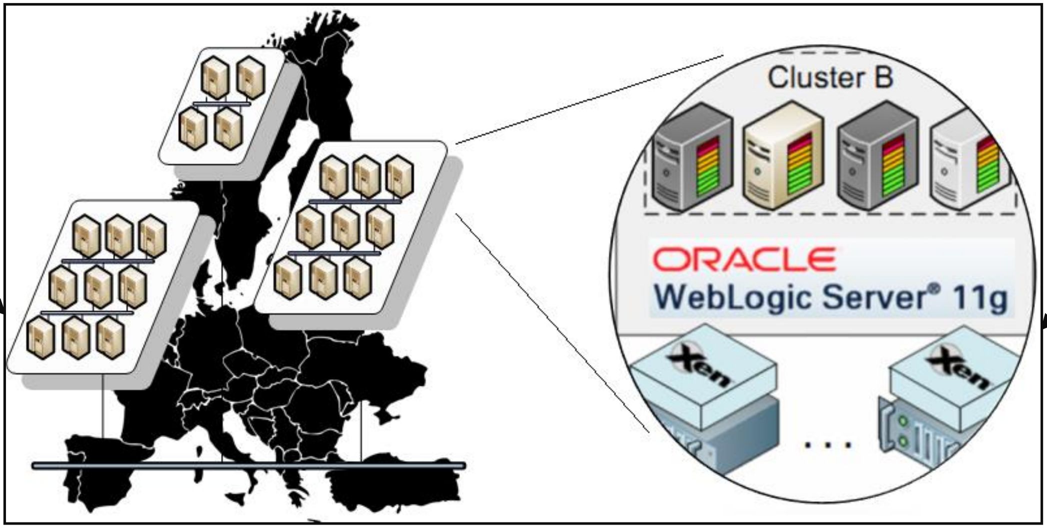
Increasing Complexity & Dynamics

Traffic Monitoring System



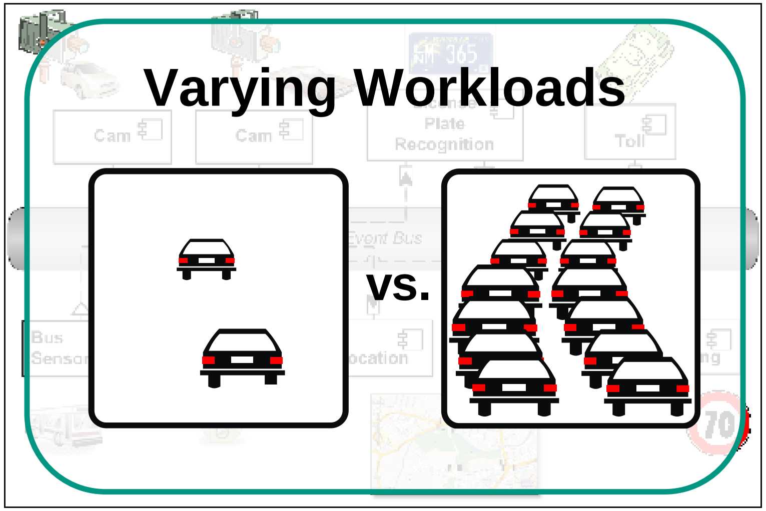
Inventory Management System



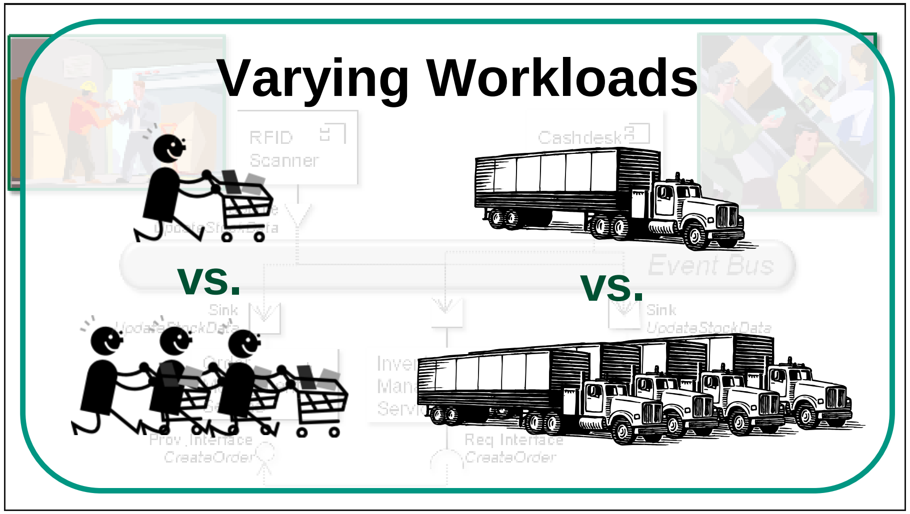


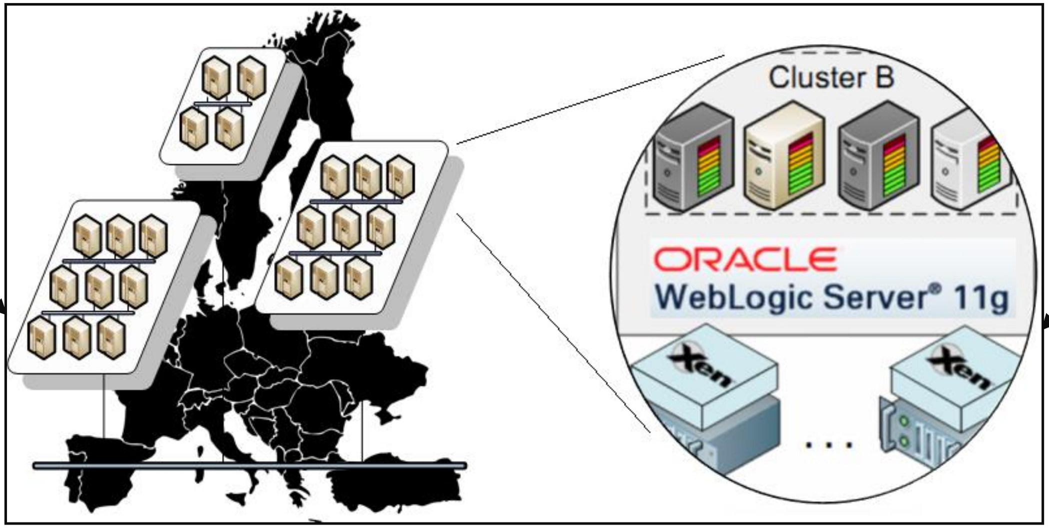
Increasing Complexity & Dynamics

Traffic Monitoring System



Inventory Management System



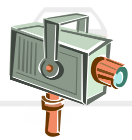


Increasing Complexity & Dynamics

Traffic Monitoring System

System Evolution

- New streets / bus lines
- New features and services
- Upgraded cameras

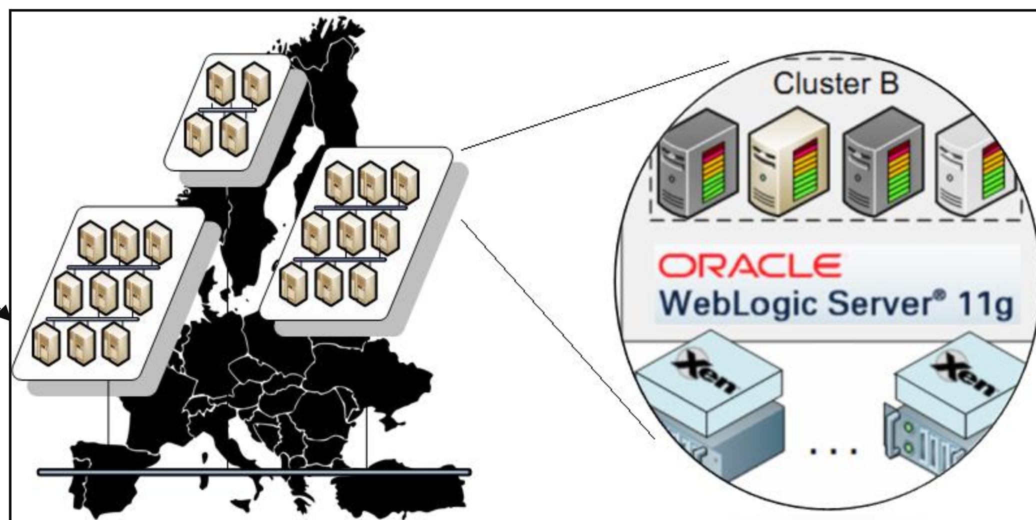
 **vs.** 

Inventory Management System

System Evolution

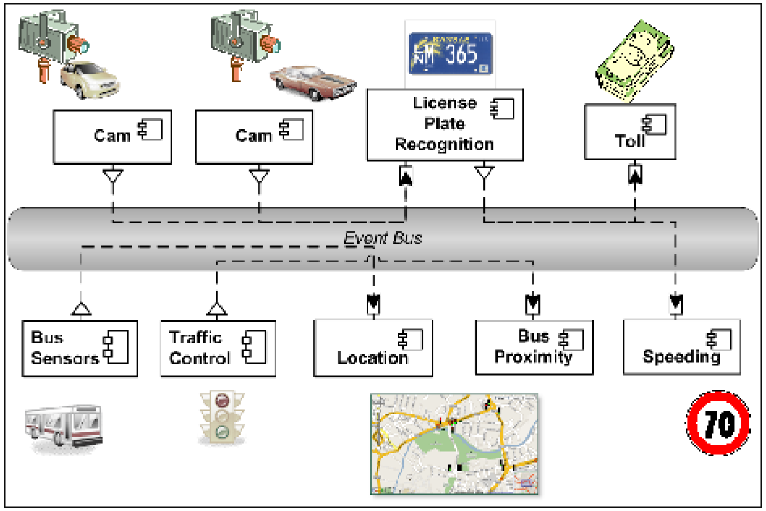
- New supermarket stores
- New features and services
- Upgraded RFID readers

 **vs.** 

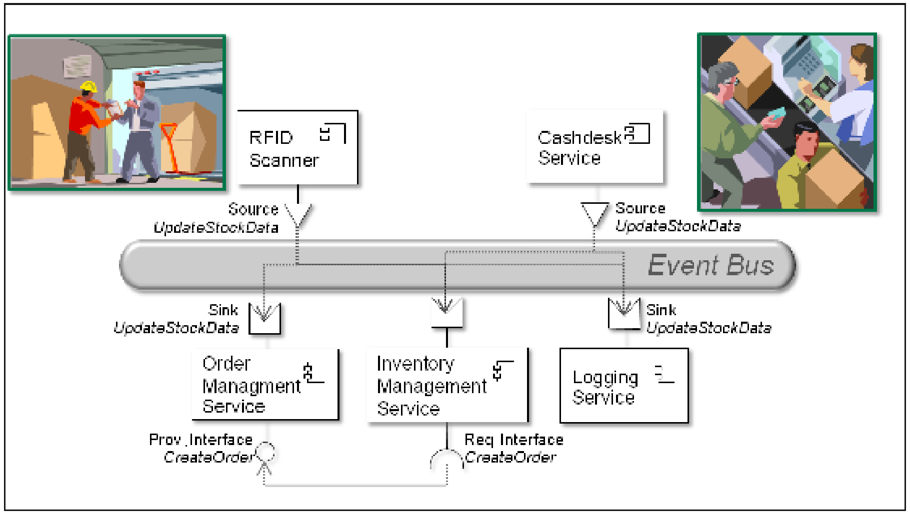


Increasing Complexity & Dynamics

Traffic Monitoring System

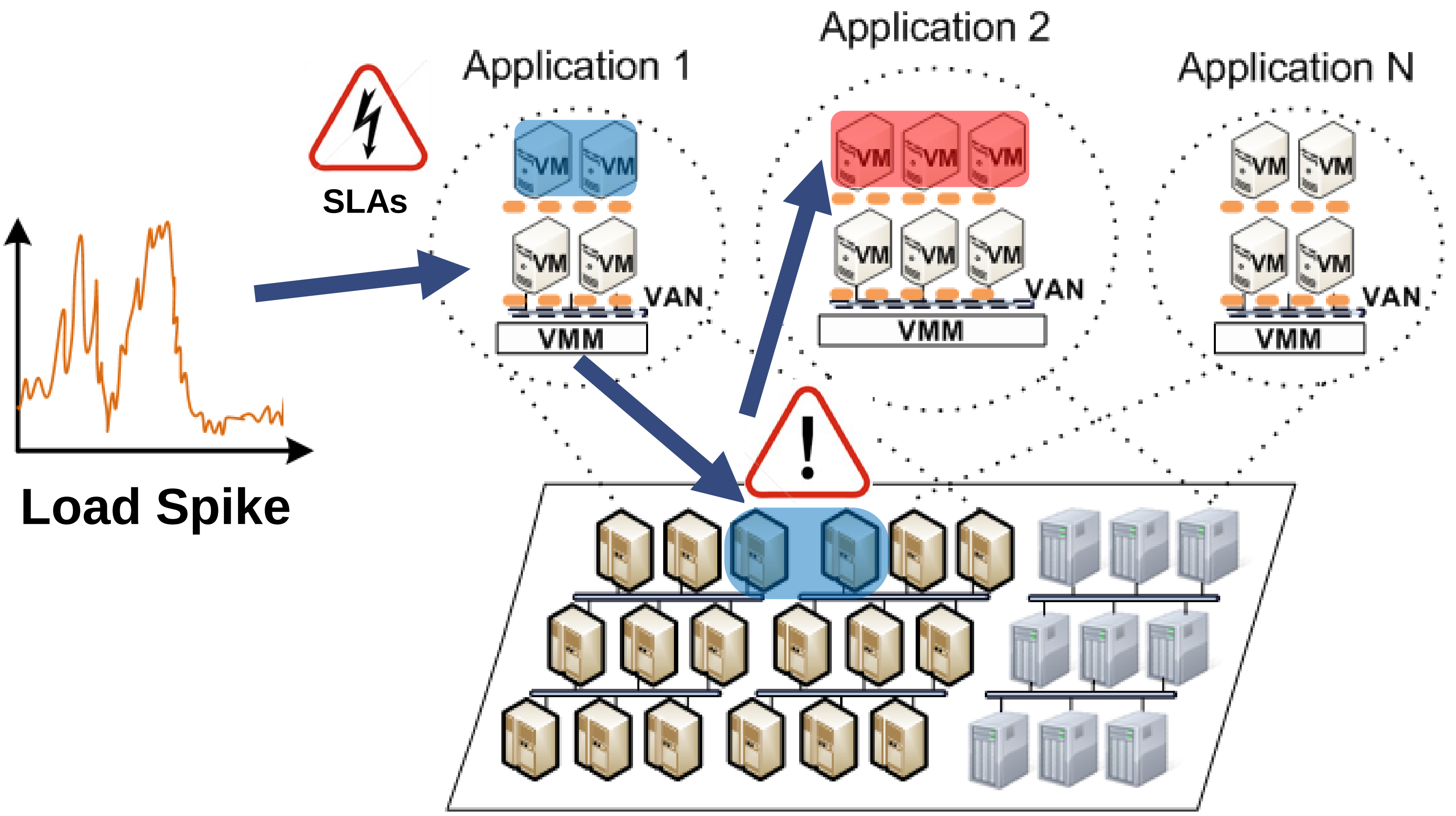


Inventory Management System

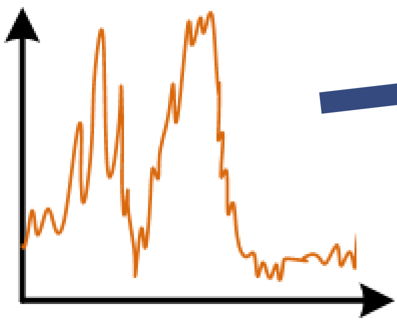


- Software systems increasingly **complex** and **dynamic**
- Must be **reconfigured at run-time** more and more frequently
  - Component instances, application configuration
  - Deployment topology, resource allocations
- Two issues:
  - Determine **WHEN** exactly reconfigurations are necessary?
  - Determine **WHAT** exactly each reconfiguration should do?

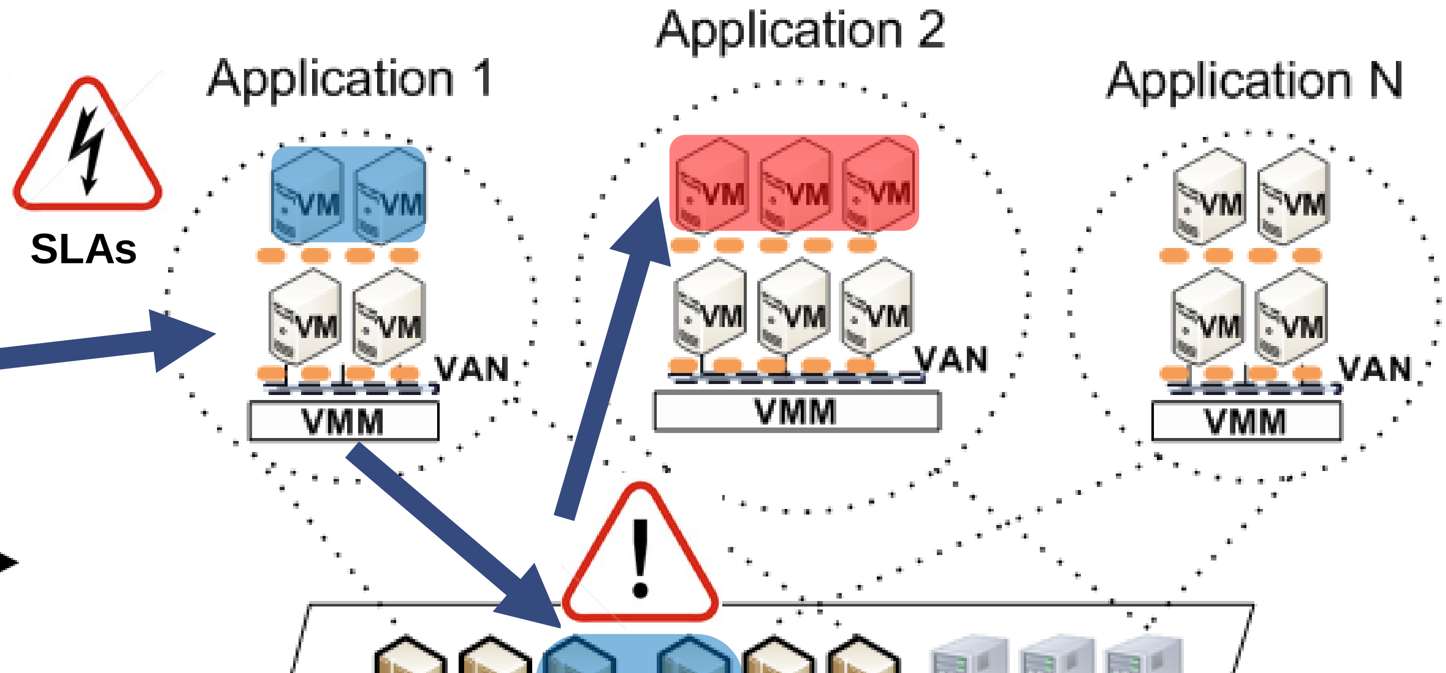
Challenges: Availability & Performance



Challenges: Availability & Performance



Load Spike



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?

Autumn 2015: Overload in Data Centers of the Sparkasse Bank

- 94 Sparkasse branch banks are down
- Cause: „overload of the network infrastructure "

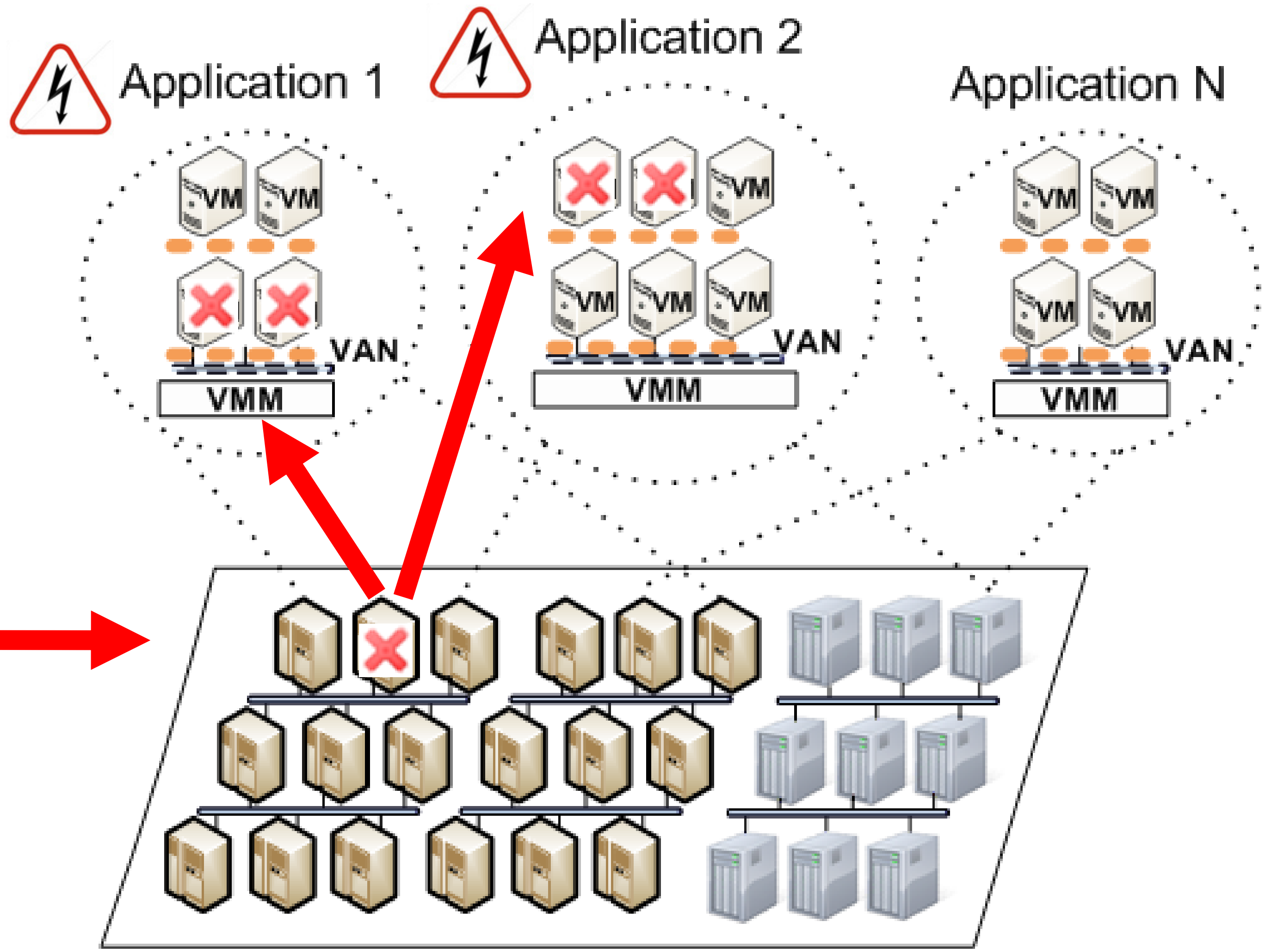
Sparkasse

Frankfurter Allgemeine

9. Juni 2016: Software-Panne: Kunden leiden unter IT-Schwäche der Banken

[http://www.faz.net/aktuell/finanzen/meine-finanzen/sparen-und-geld-anlegen/kunden-leiden-unter-it-schwaeche-der-banken-14276587.html]

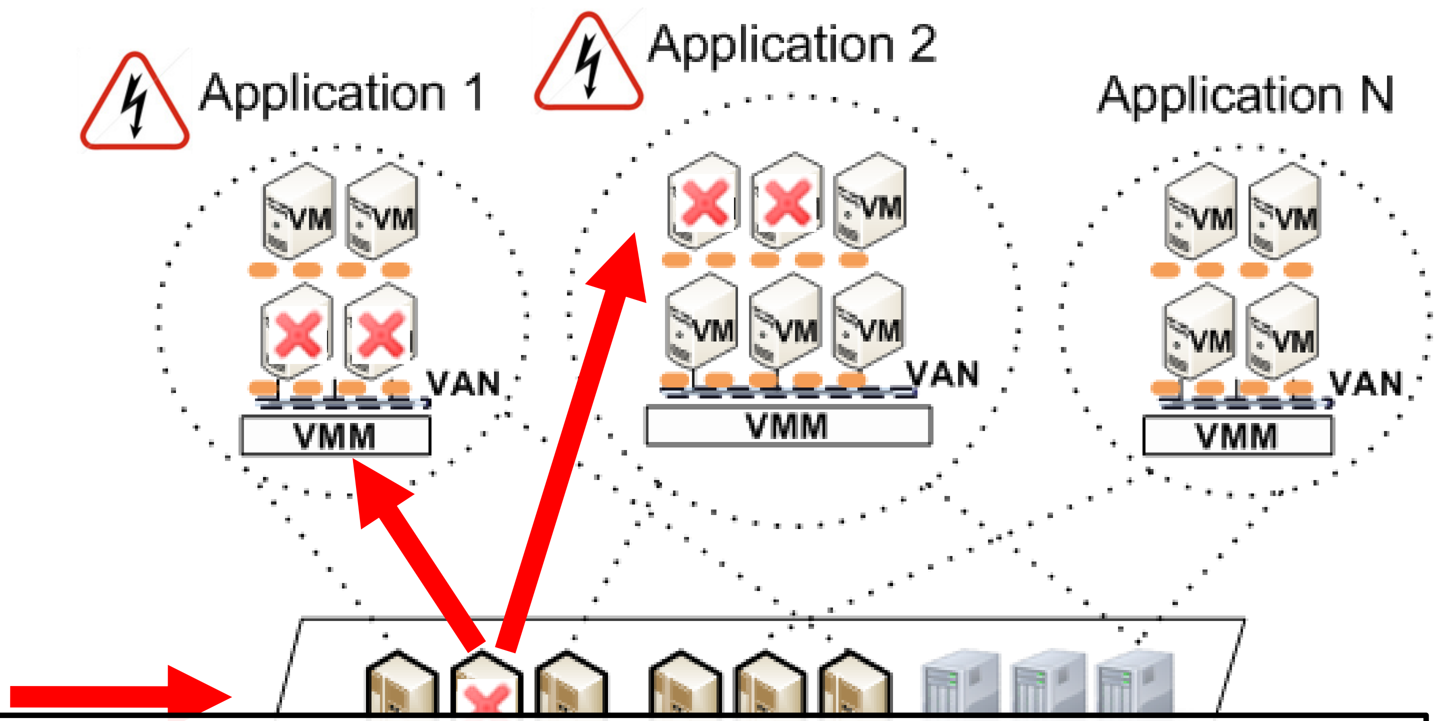
Challenges: Reliability



Hardware or Software Failure

Challenges: Reliability

Hardware or Software Failure



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

Software Crash @ Deutsche Bank

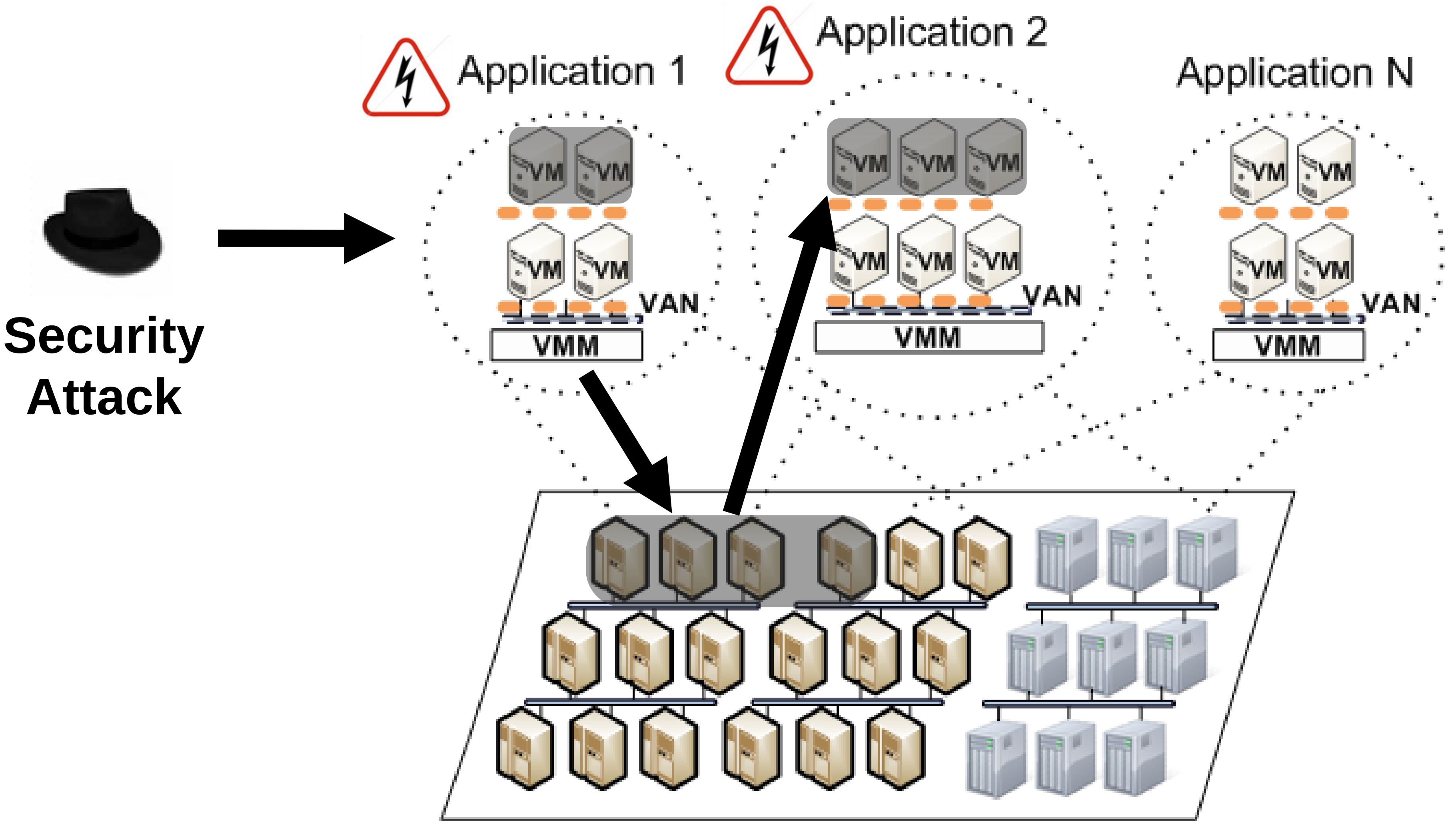
- 60,000 customers cannot use their bank card
- **2.9 million accounts** → show wrong balance!
- Numerous double bookings
- ...

Frankfurter Allgemeine

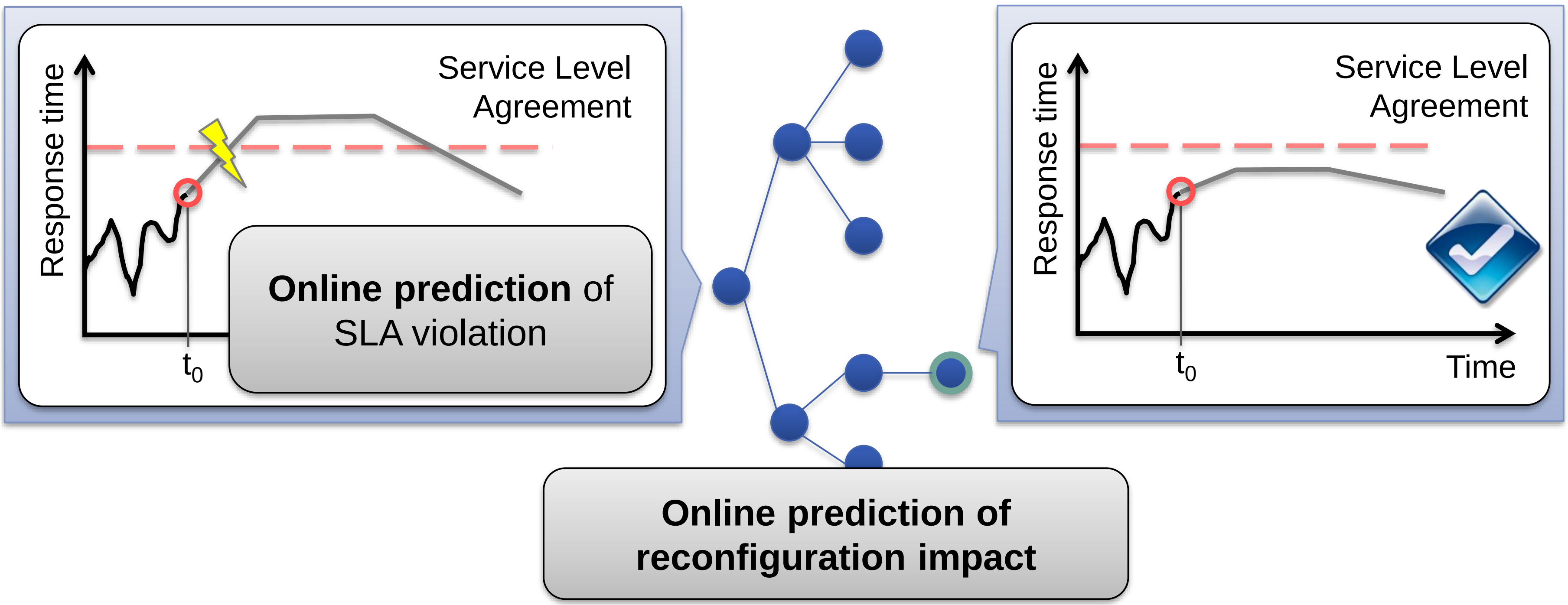
9. Juni 2016: Software-Panne: Kunden leiden unter IT-Schwäche der Banken

[http://www.faz.net/aktuell/finanzen/meine-finanzen/sparen-und-geld-anlegen/kunden-leiden-unter-it-schwaeche-der-banken-14276587.html]

Challenges: Security



Self-Aware Data Center



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

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

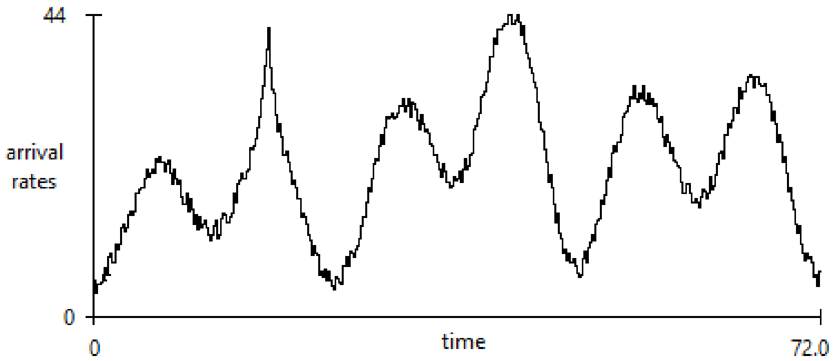
http://descartes.tools

Mailing list available...

LIMBO Tool

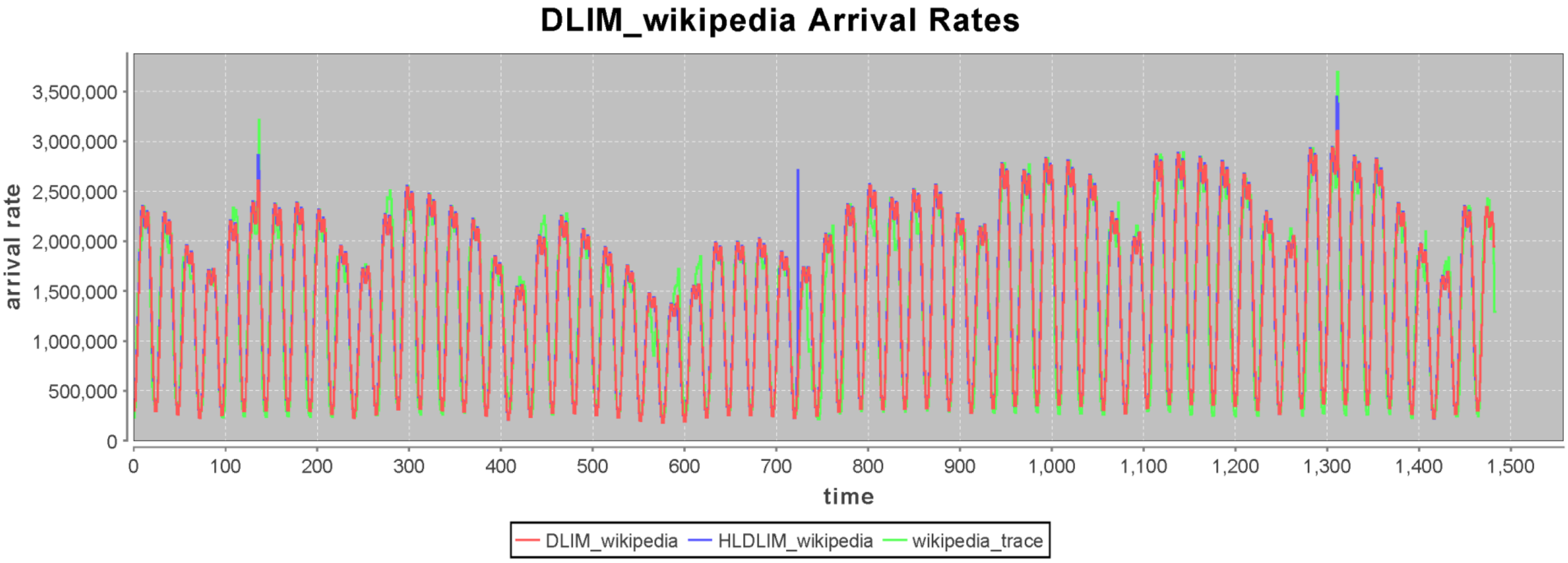
- Problem:
  - How to capture the load intensity variations (e.g., requests per sec) in a compact mathematical model?
  - How to forecast the load intensity (requests per sec) in future time horizons?

- **Load Intensity Modeling & Forecasting Tool**

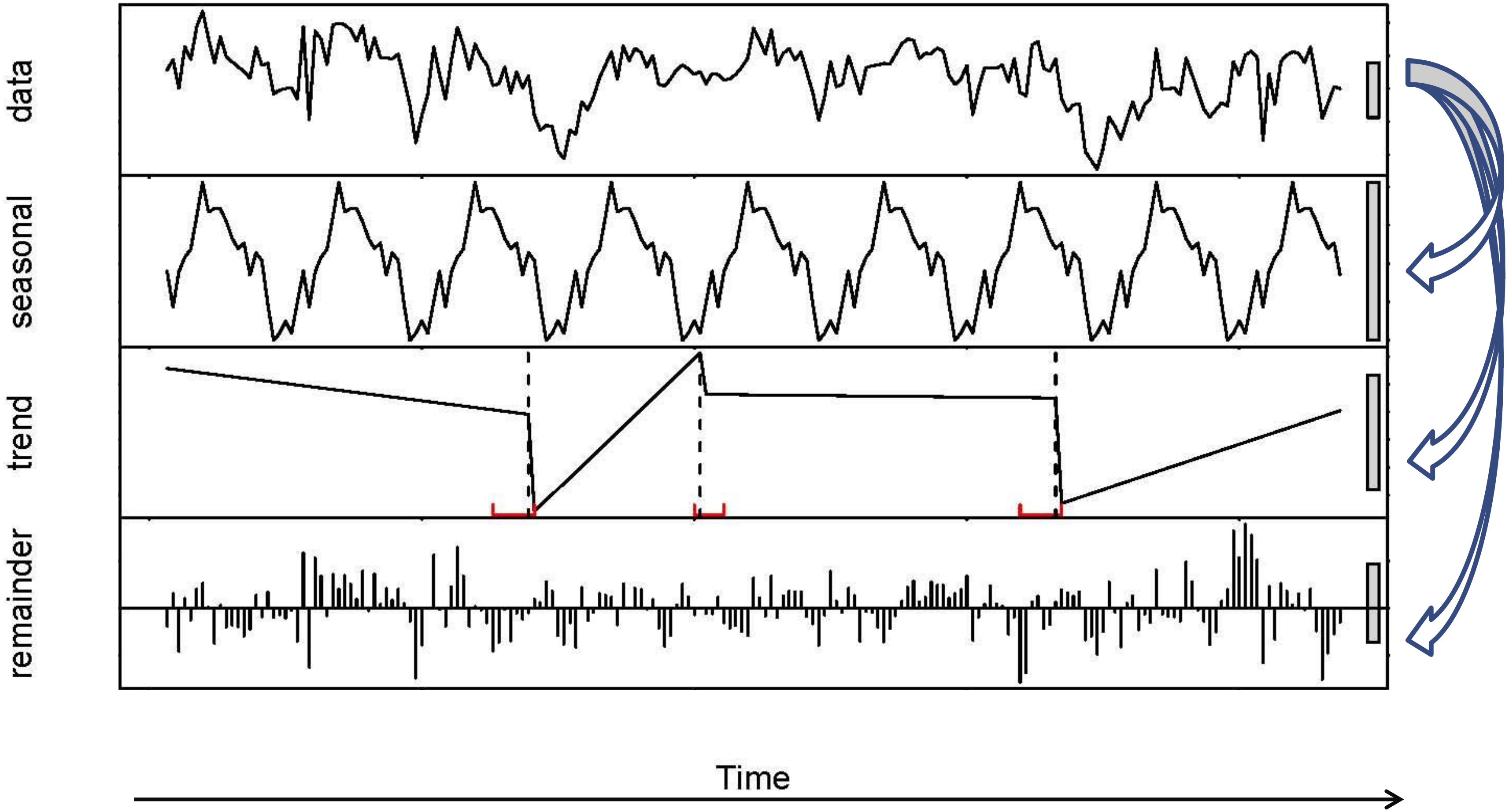


http://descartes.tools/limbo

Example: Wikipedia Workload



Time Series Analysis



[BFAST]

Applied Forecasting Methods

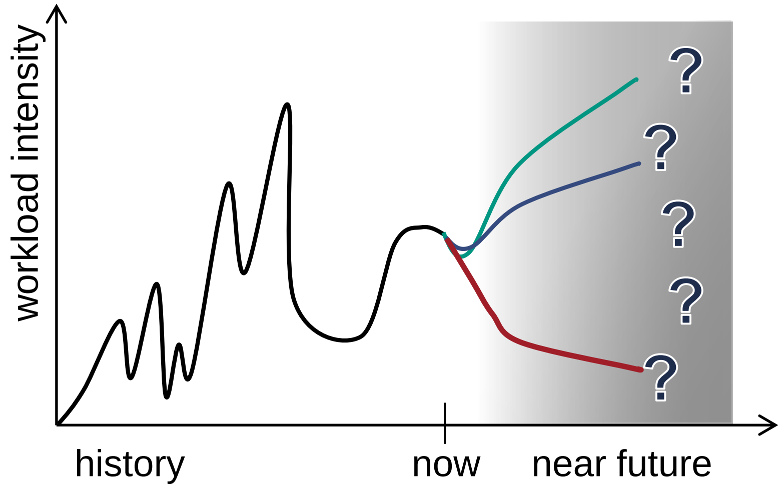
Basic Methods	**(initial)**
Naïve, Moving Averages, Random Walk	

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 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

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

LibReDE

Google Research Award

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]

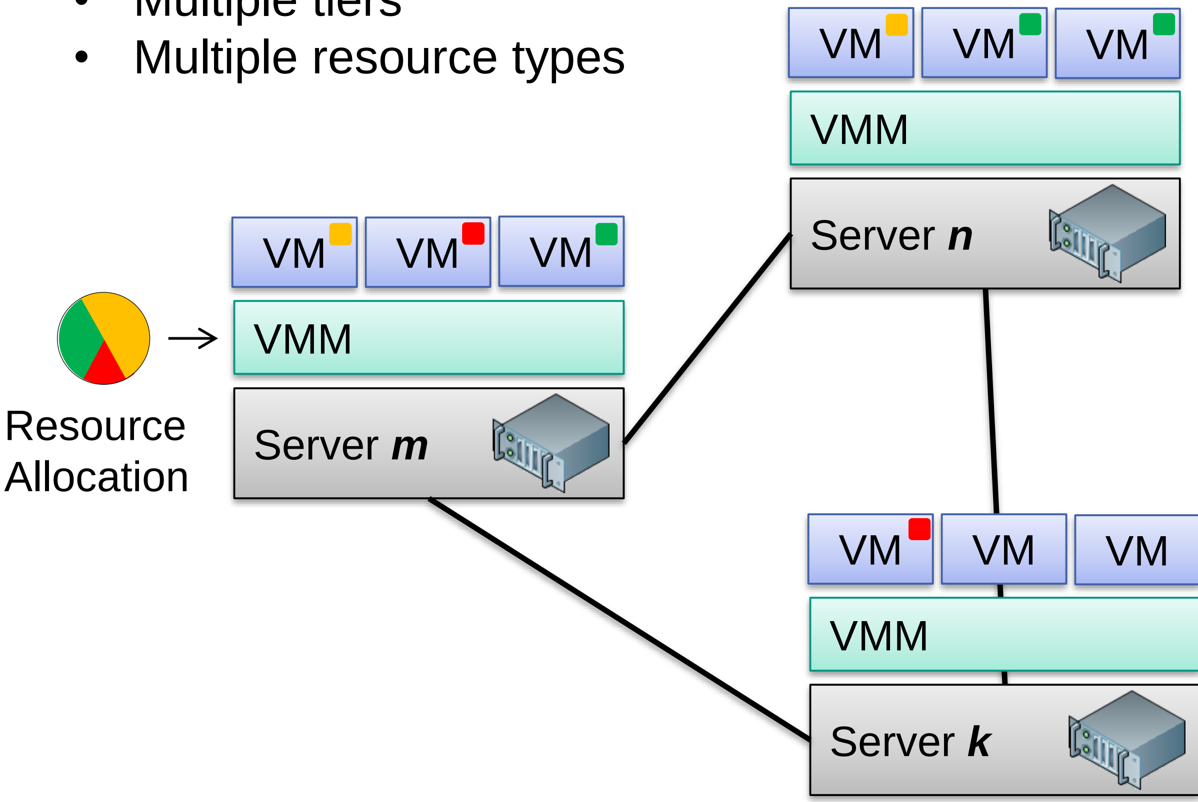
Estimation Approaches

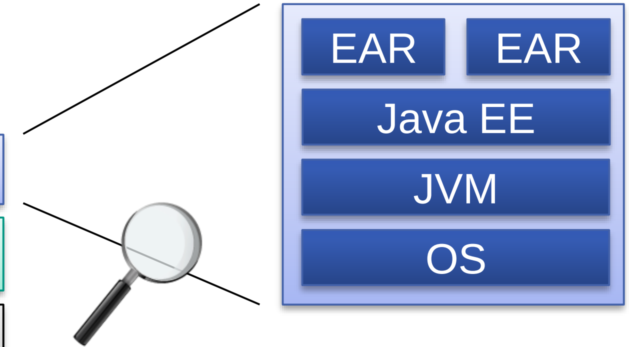
Technique	Variant	References
Approximation with response times		Urgaonkar et al. [13] Nou et al. [14] Brosig et al. [15]
Service Demand Law		Lazowksa [4] Brosig et al. [15]
Linear regression	Least squares	Bard and Shatzoff [16] Rolia et al. [17, 18] Pacifici et al. [19] Kraft et al. [20, 21]
	Least absolute differences	Zhang et al. [22, 23, 24]
	Least trimmed squares	Casale et al. [25, 26]
Kalman filter		Zheng et al. [27, 28] Kumar et al. [29] Wang et al. [30, 31]
Optimization	Non-linear constrained optimization	Zhang et al. [32] Menascé [33]
	Quadratic programming	Liu et al. [34, 35, 36] Kumar et al. [37]
Machine learning	Clusterwise linear regression	Cremonesi et al. [38]
	Independent component analysis	Sharma et al. [39]
	Support vector machine	Kalbasi et al. [40]
	Pattern matching	Cremonesi et al. [41, 42]
Maximum likelihood estimation		Kraft et al. [20] Perez et al. [21]
Gibbs sampling		Sutton and Jordan [43] Wang et al. [44]

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]

Semantic Gap Problem

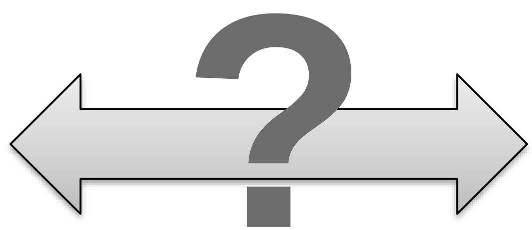
Applications
- Multiple tiers
- Multiple resource types





Complex Software Stacks
- Multiple layers
- Heterogeneous

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



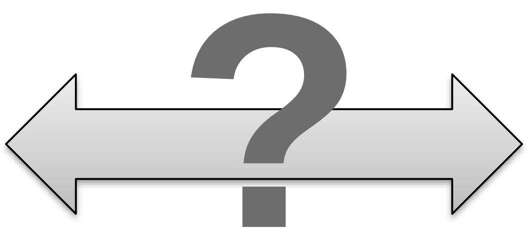
Configuration of System Components, Layers & Tiers

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

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:

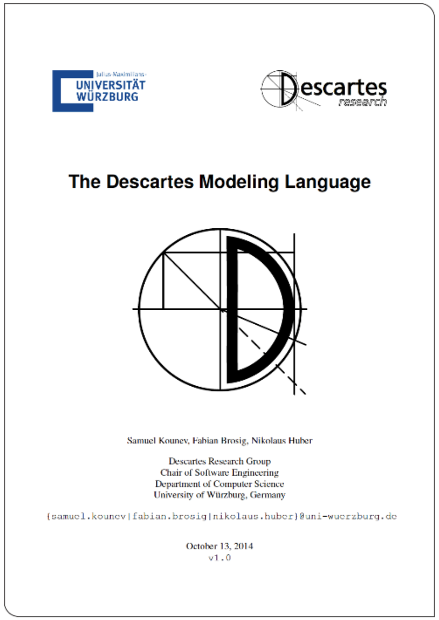
- Univariate Interpolation Library

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


UNIVERSITÄT WÜRZBURG

Descartes research

The Descartes Modeling Language

Samuel Kounev, Fabian Brosig, Nikolaus Huber

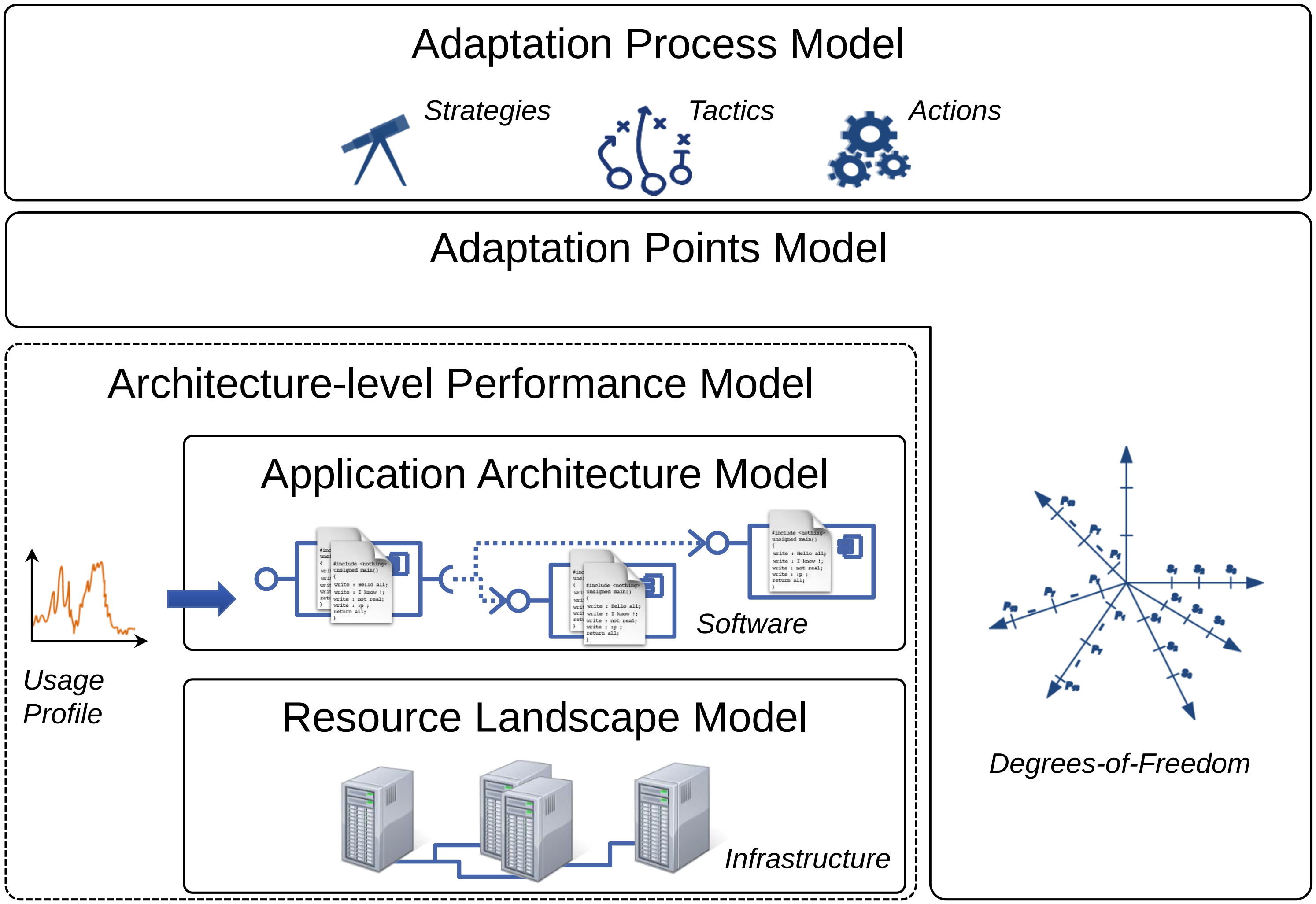
Descartes Research Group
Chair of Software Engineering
Department of Computer Science
University of Würzburg, Germany

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

October 13, 2014
v1.0

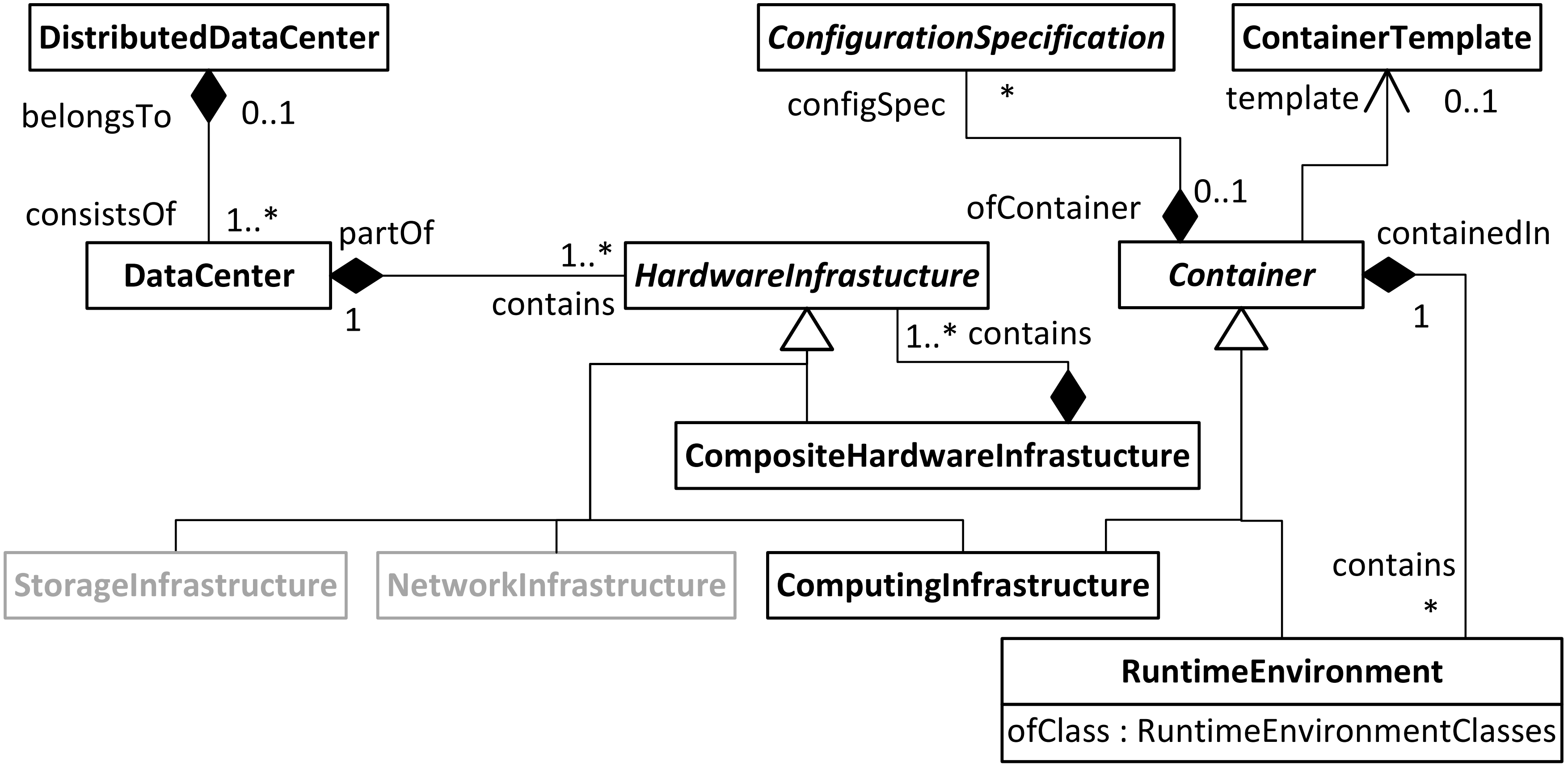
http://descartes.tools/dml

DML Sub-Models

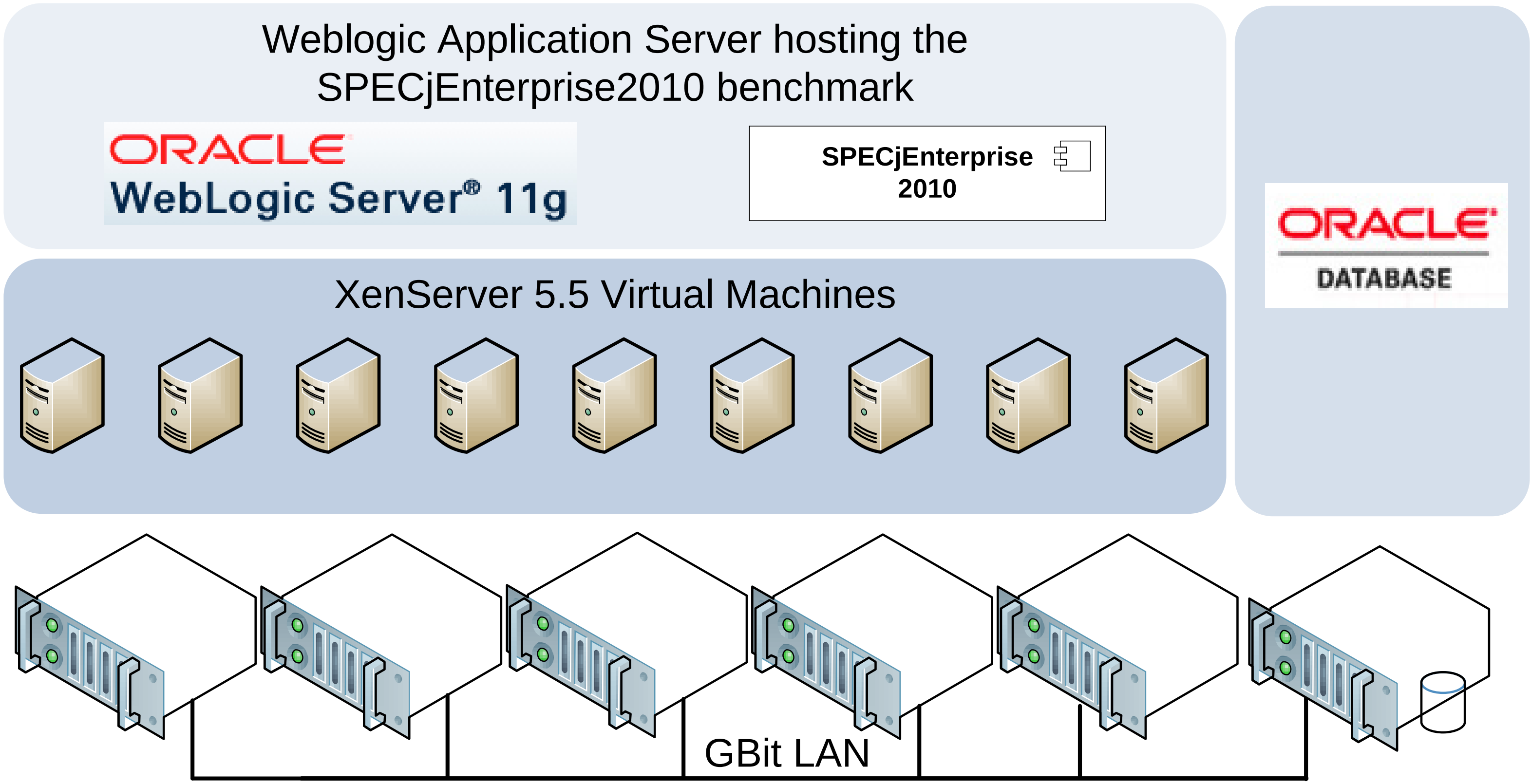


Resource Landscape Meta-Model

(Selected Top Level Modeling Elements)

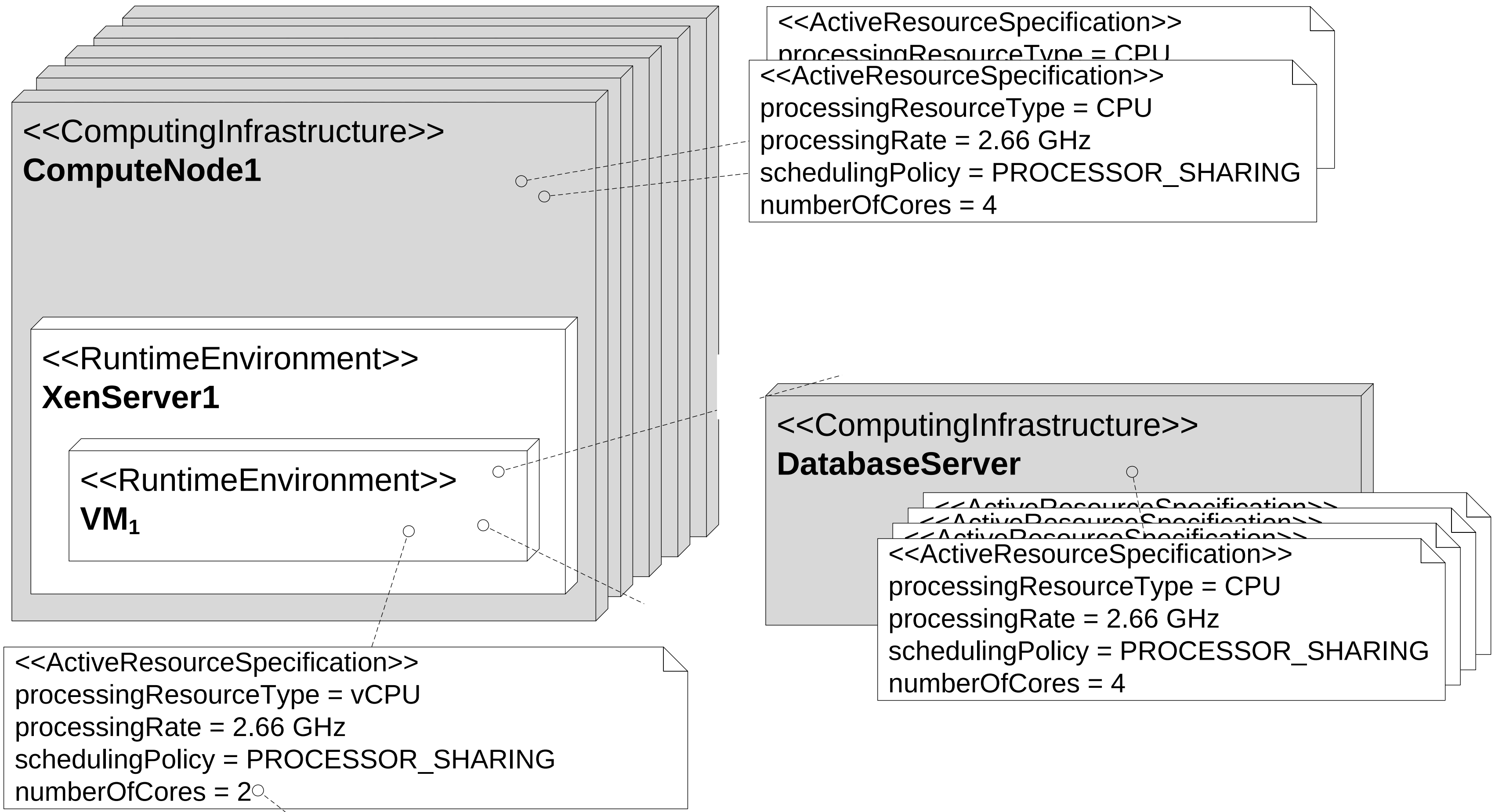


Example: WebLogic Server Cluster
(Resource Landscape)



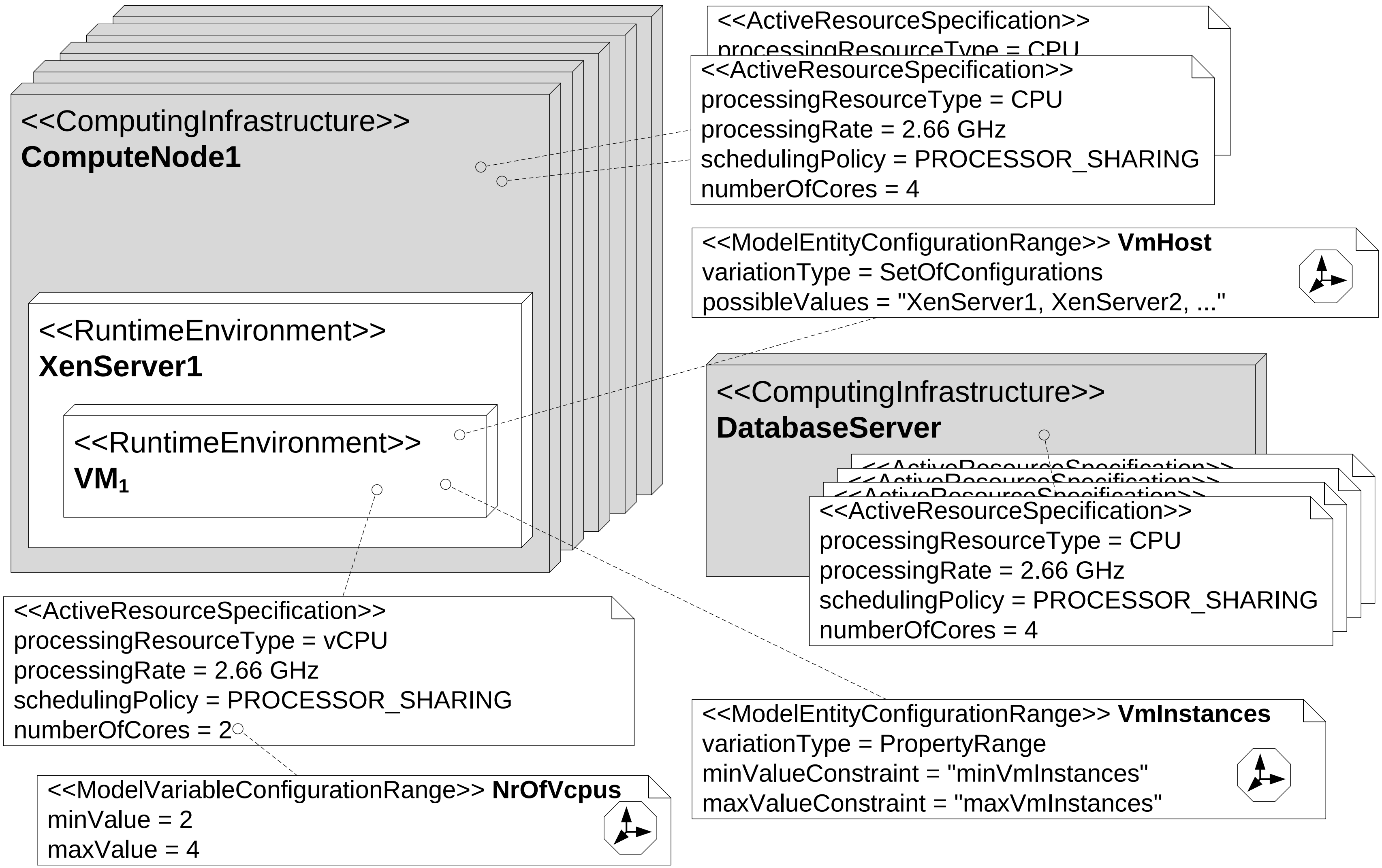
Example: WebLogic Server Cluster

(Resource Landscape Model)

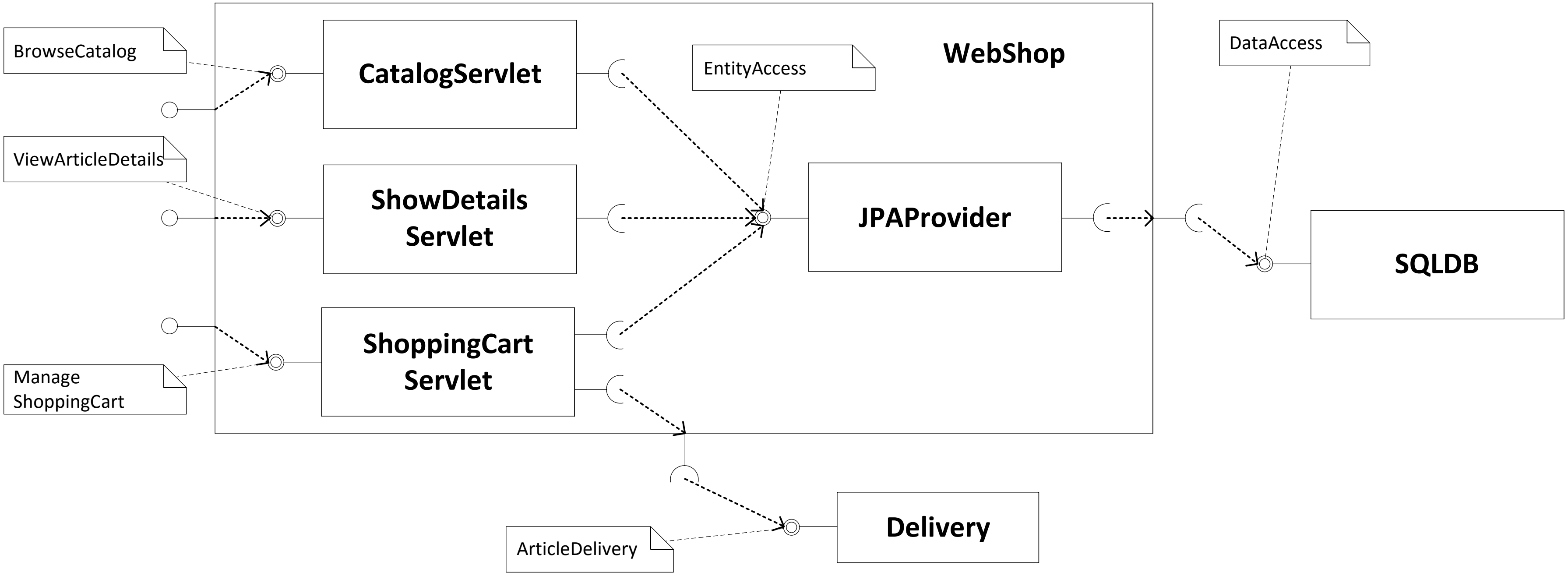


Example: WebLogic Server Cluster

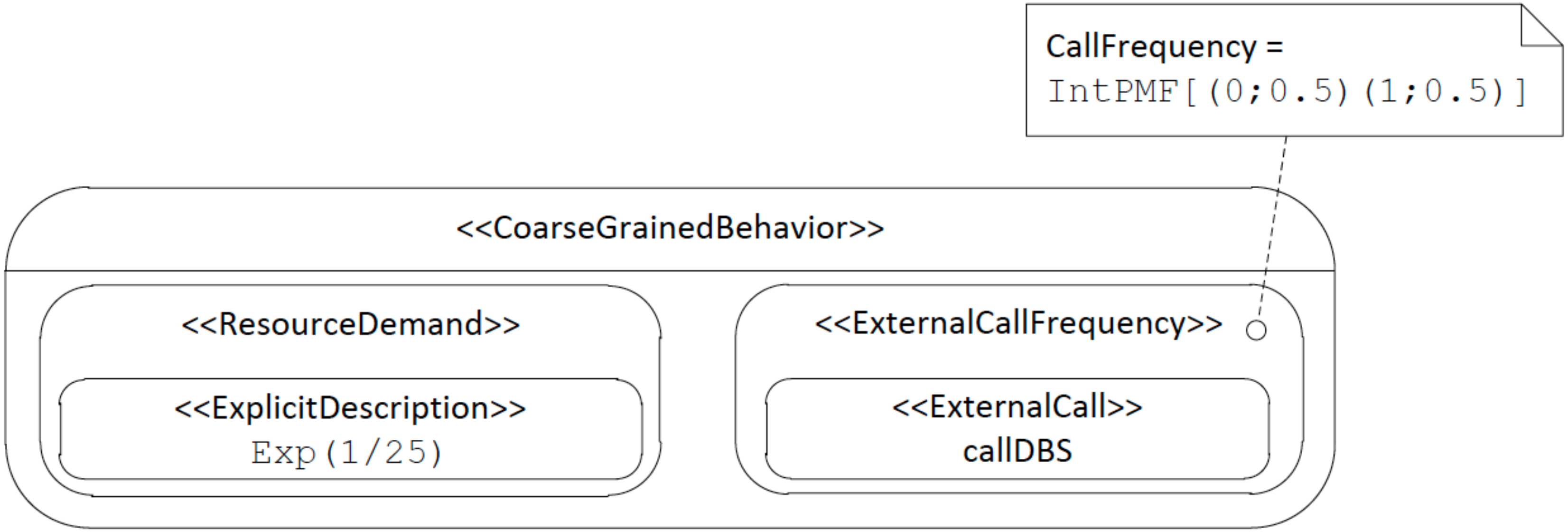
(Resource Landscape Model) + (Adaptation Points Model)



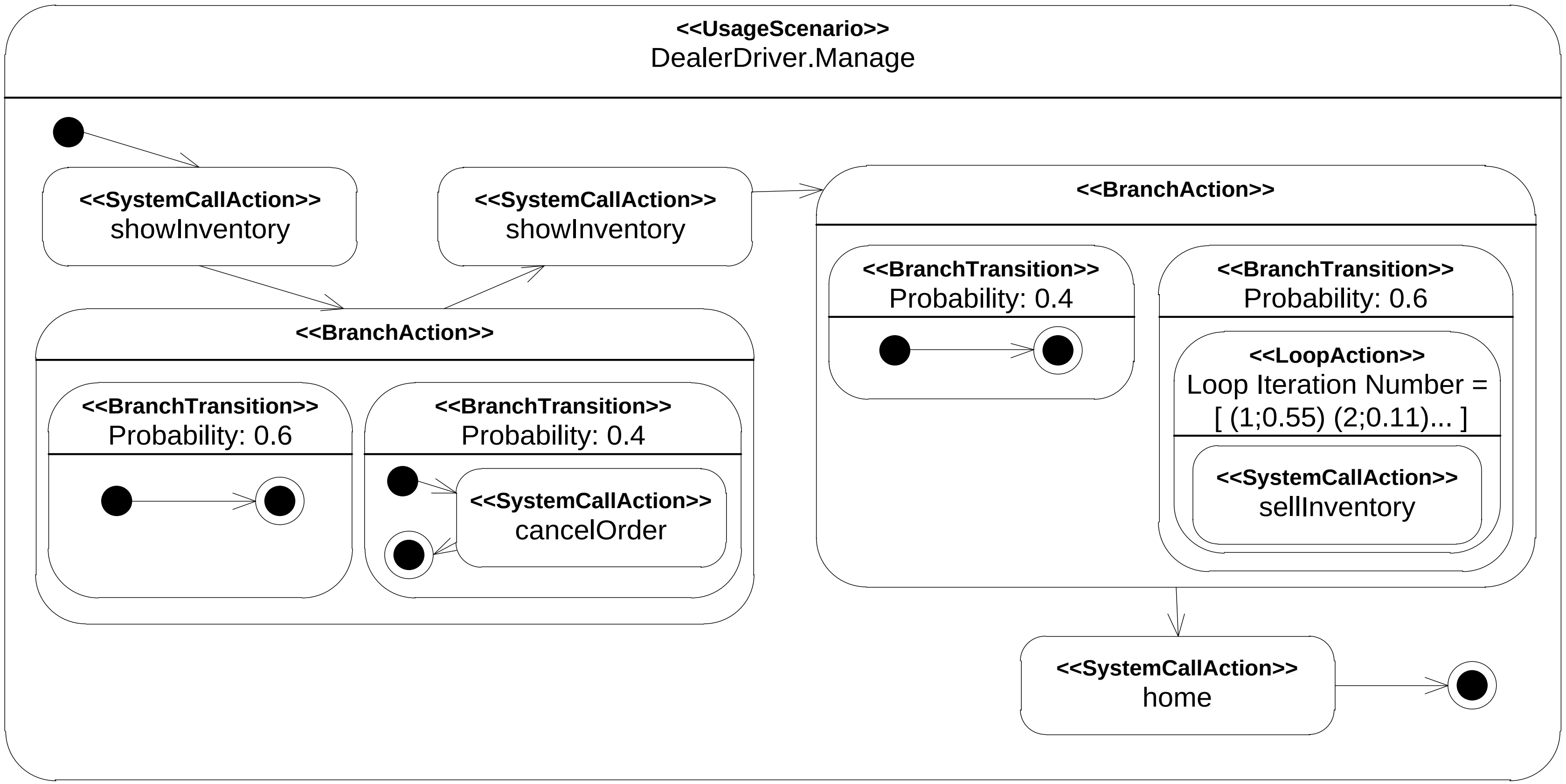
Example
(Application Architecture Model)



Example
(Coarse-Grained Service Behavior Model)



Example
(Fine-Grained Service Behavior Model)

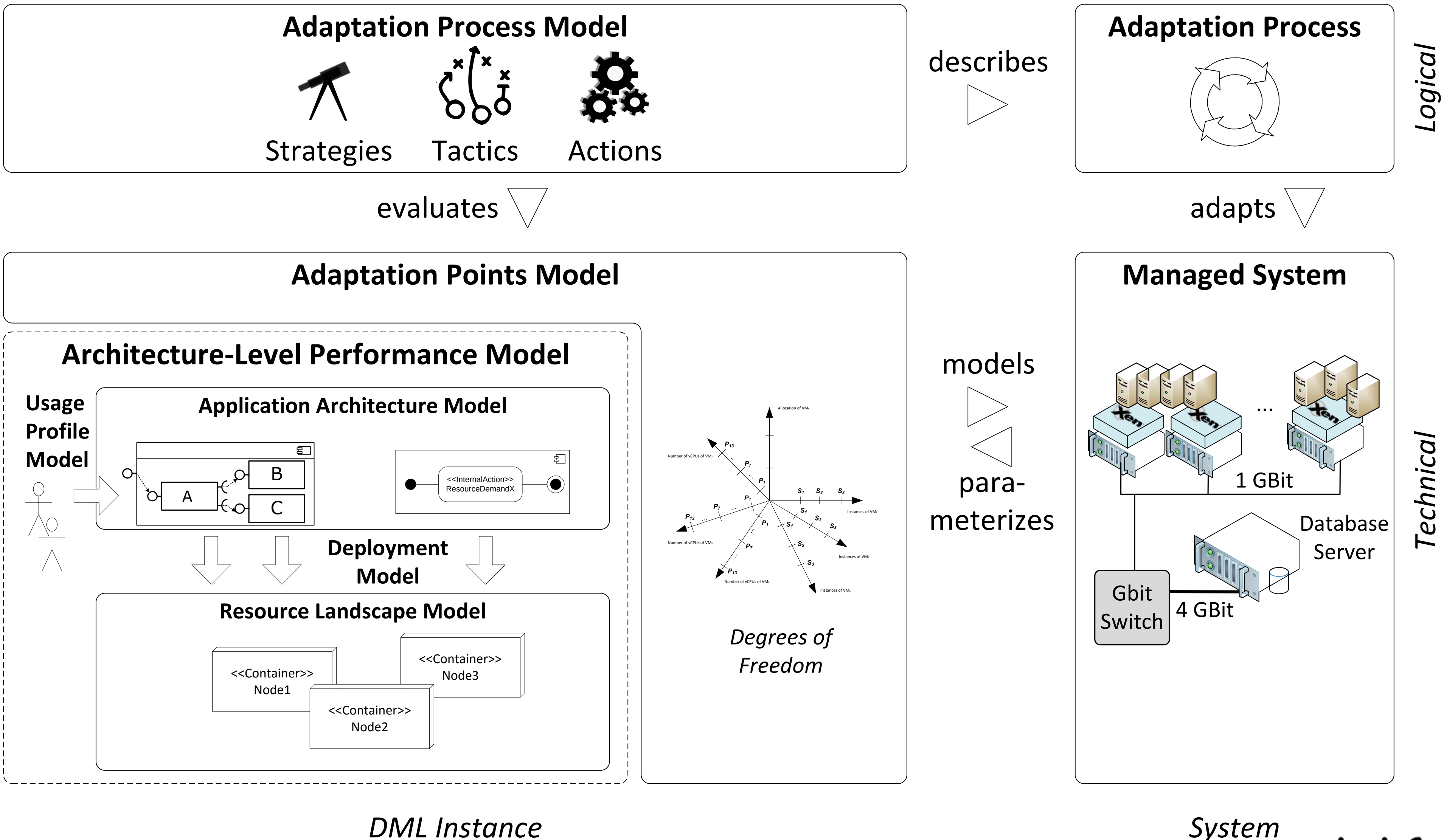


DNI - Descartes Network Infrastructure Modeling

- 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

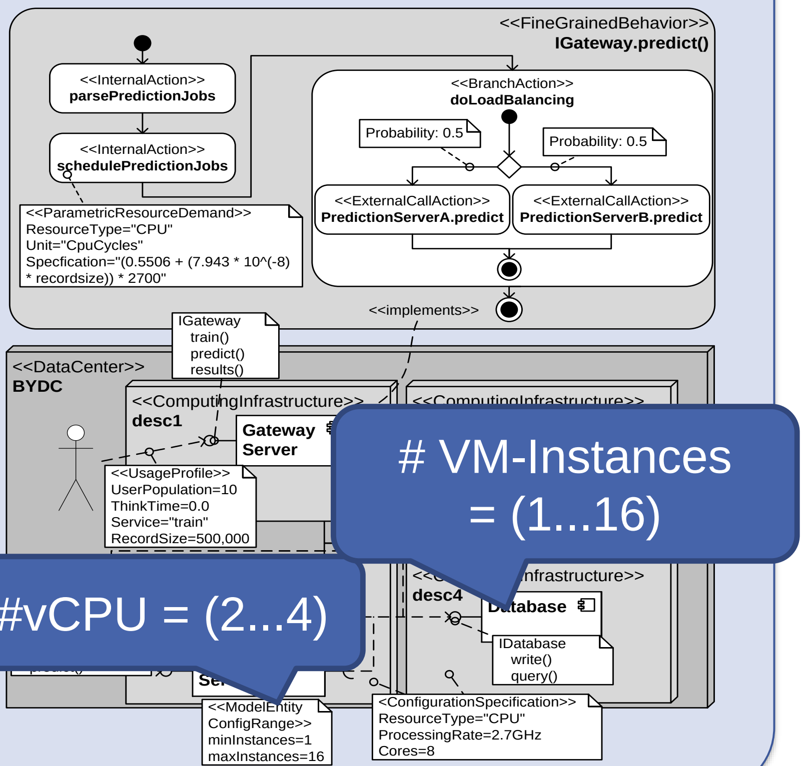
Big Picture



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

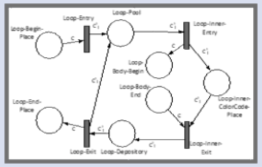
Architecture-Level Performance Model

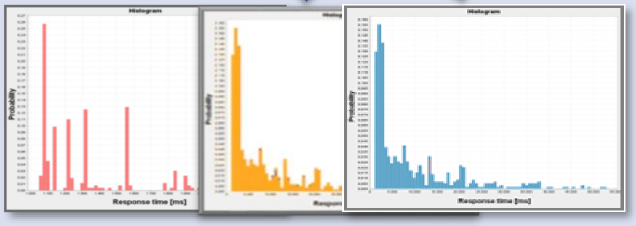


Online Performance Prediction

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

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





Autonomic Decision Making

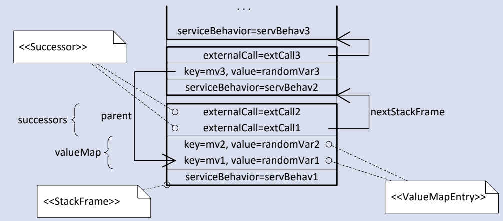
Tailored Model Solution

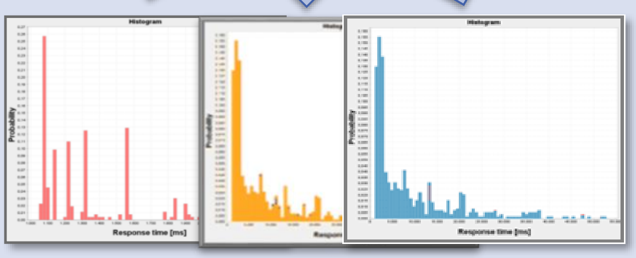
Analytical Analysis

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

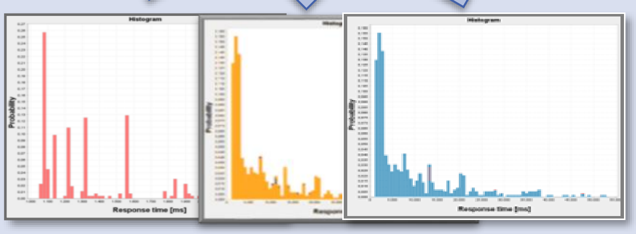
$$\frac{N}{\max\{D_i\}[K+N-1]} \le X_0 \le \frac{N}{avg\{D_i\}[K+N-1]}$$

Simulative Analysis





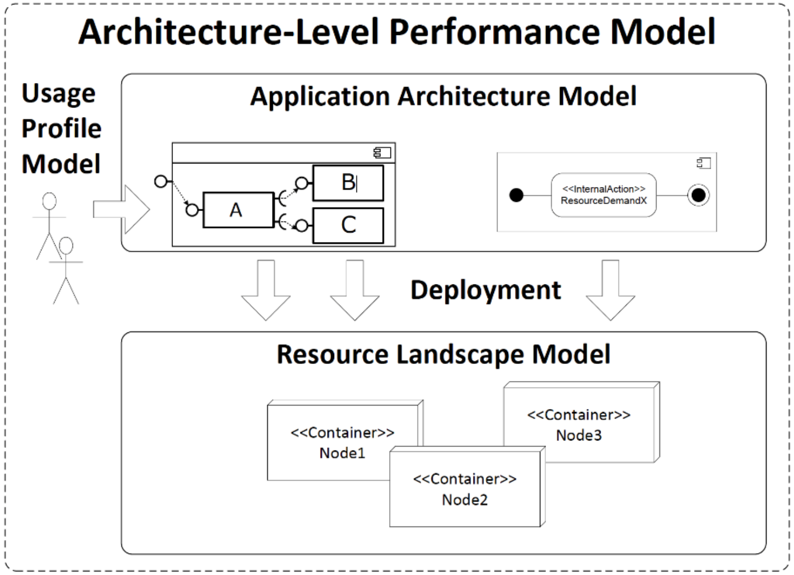
Analysis Results



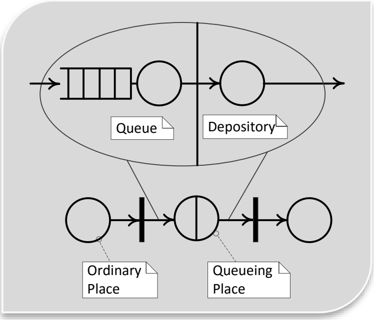
Analysis Results

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]

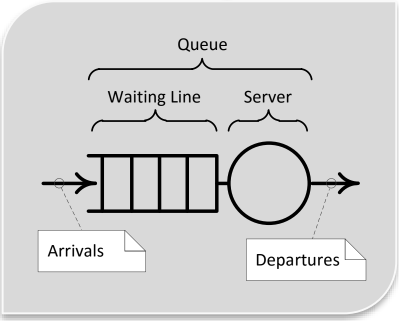
Transformations to Predictive Models



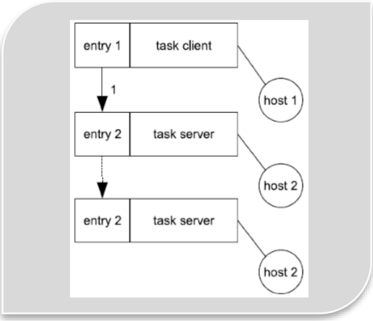
DML Instance



Queueing Petri Net



Bounds Analysis Model



Layered Queueing Network

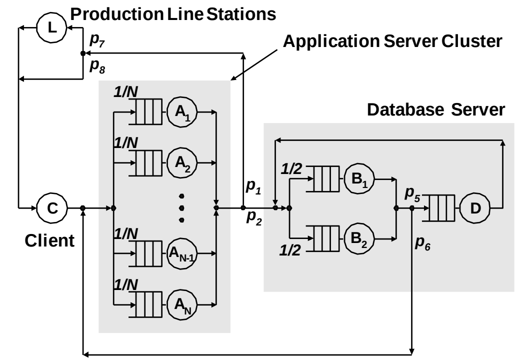
Example Predictive Models

Schranken-Analyse

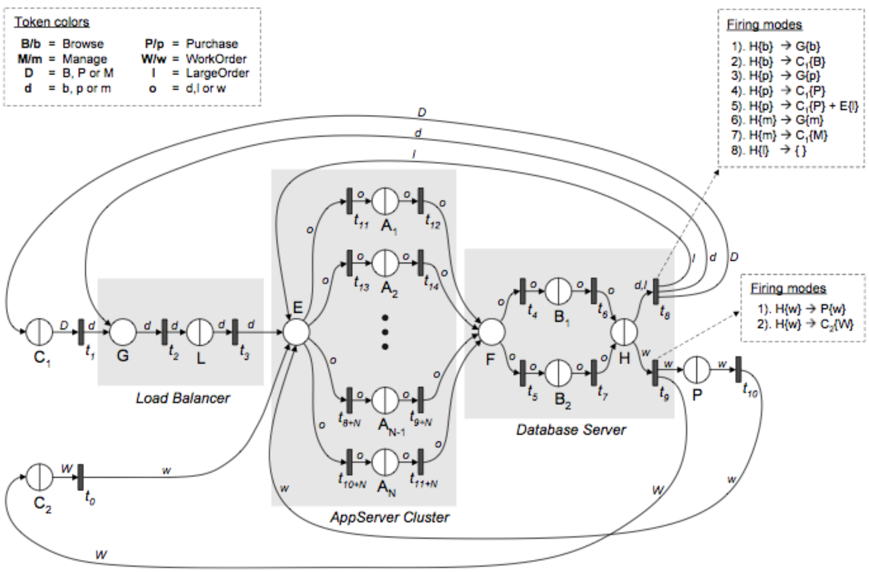
$$R \geq \max\left[N \times \max\{D_i\}, \sum_{i=1}^{K} D_i\right] \qquad X_0 \leq \min\left[\frac{1}{\max\{D_i\}}, \frac{N}{\sum_{i=1}^{K} D_i}\right]$$

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

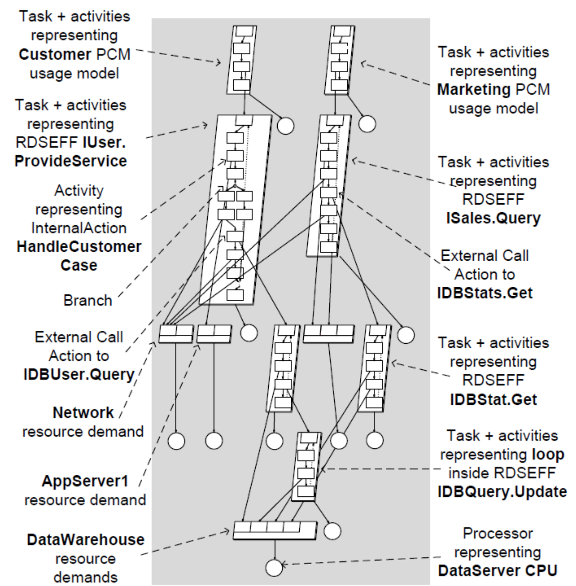
Warteschlangennetz



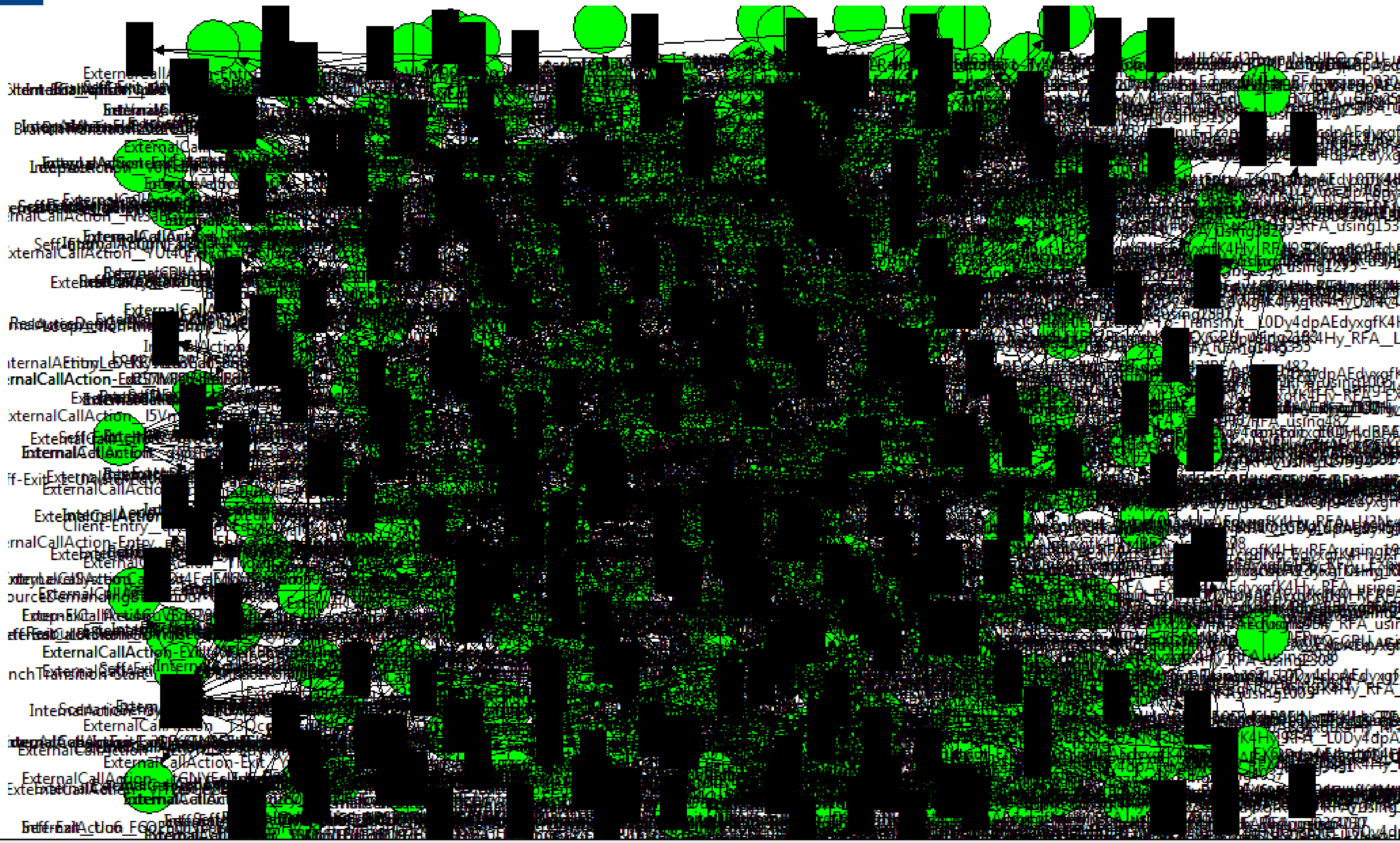
Warteschlangen-Petri-Netz (Queueing-Petri-Net)



Geschachteltes Warteschlangennetz (LQN)

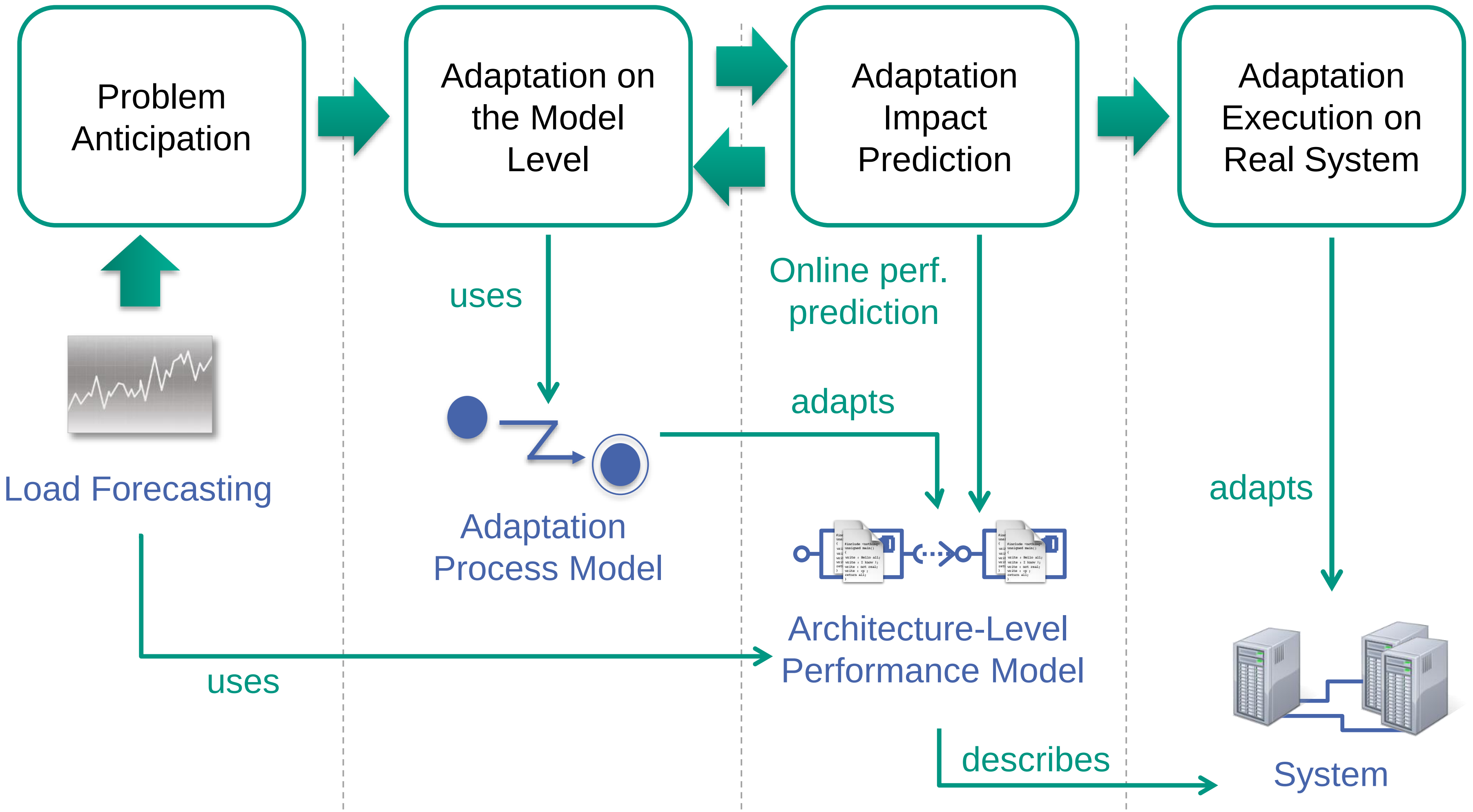


Case Study: Process Control System (ABB)

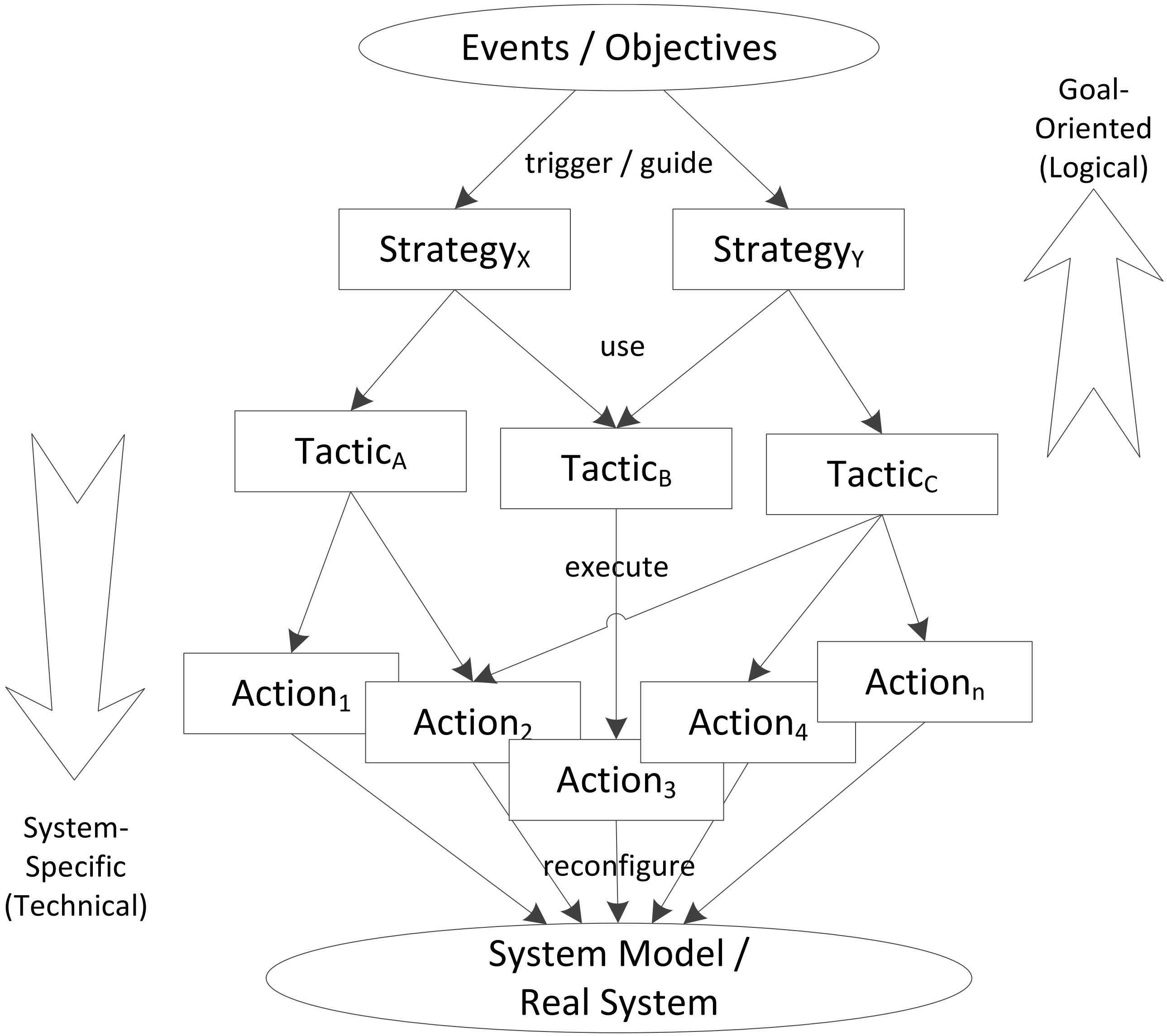


P. Meier, S. Kounev, and H. Koziolek. **Automated transformation of component-based software architecture models to queueing petri nets.** In *19th IEEE/ACM Intl. Symp. on Modeling, Analysis and Simulation of Computer and Telecomm. Systems (MASCOTS), Singapore, July 25-27*, 2011. [.pdf]

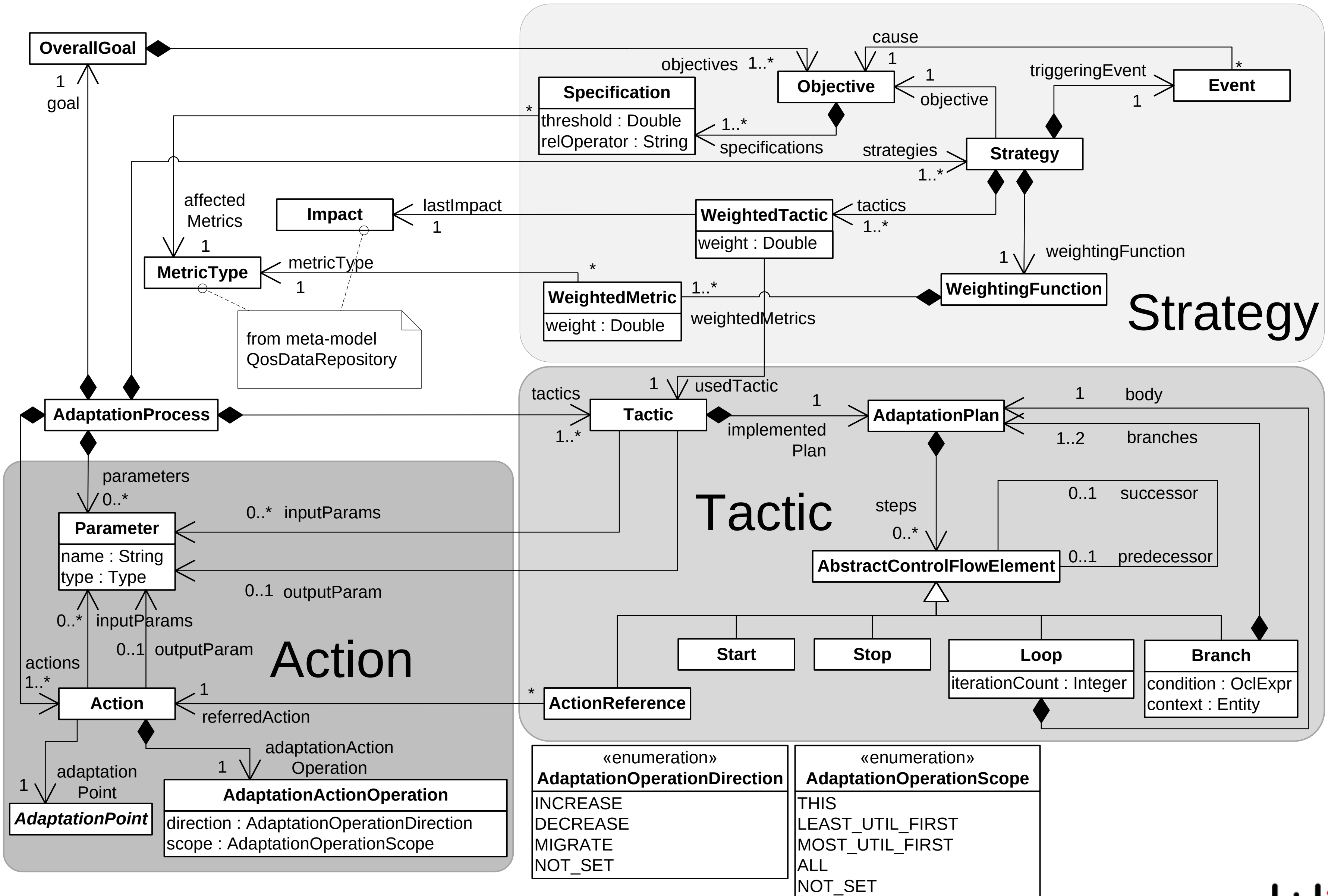
Model-Based System Adaptation



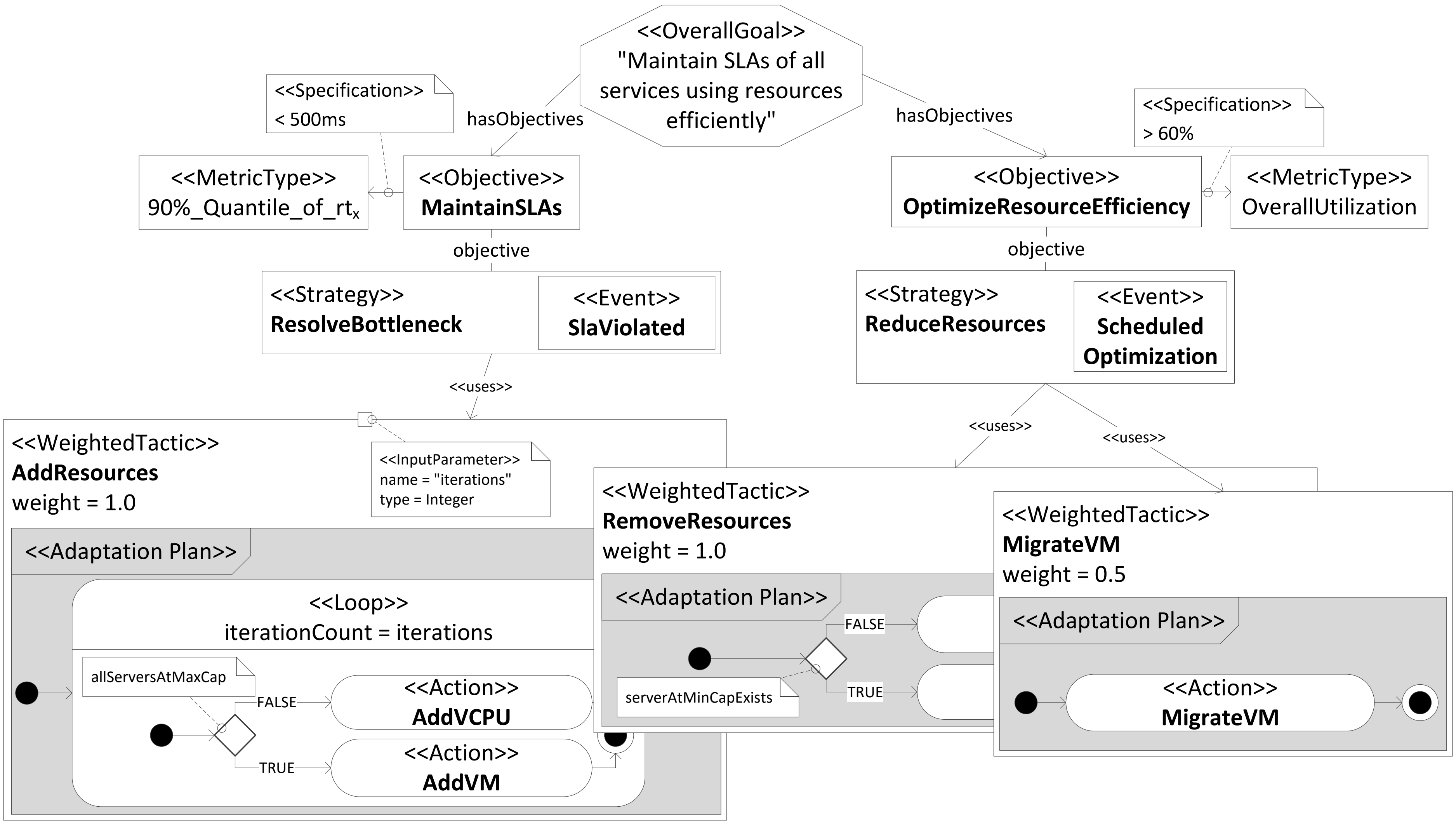
Adaptation Process Model



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



Example: Adaptation Process Model



Applied Modeling Techniques

Descriptive Architecture-level Models

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

Predictive Performance 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

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.

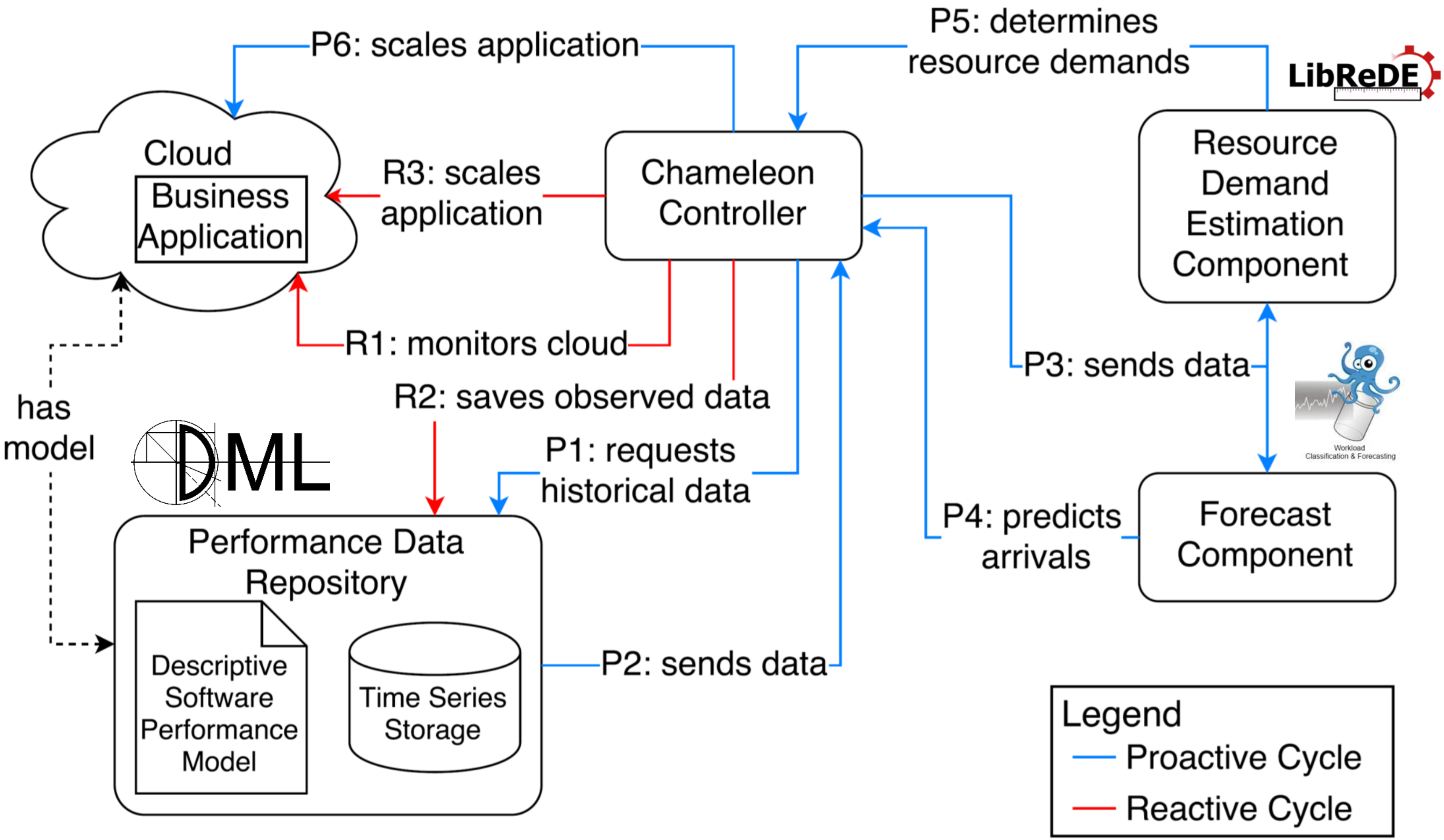
N. Huber, F. Brosig, S. Spinner, S. Kounev, and M. Bähr. ***Model-Based Self-Aware Performance and Resource Management Using the Descartes Modeling Language***. IEEE Transactions on Software Engineering (TSE), PP(99), 2017.

Putting it All Together

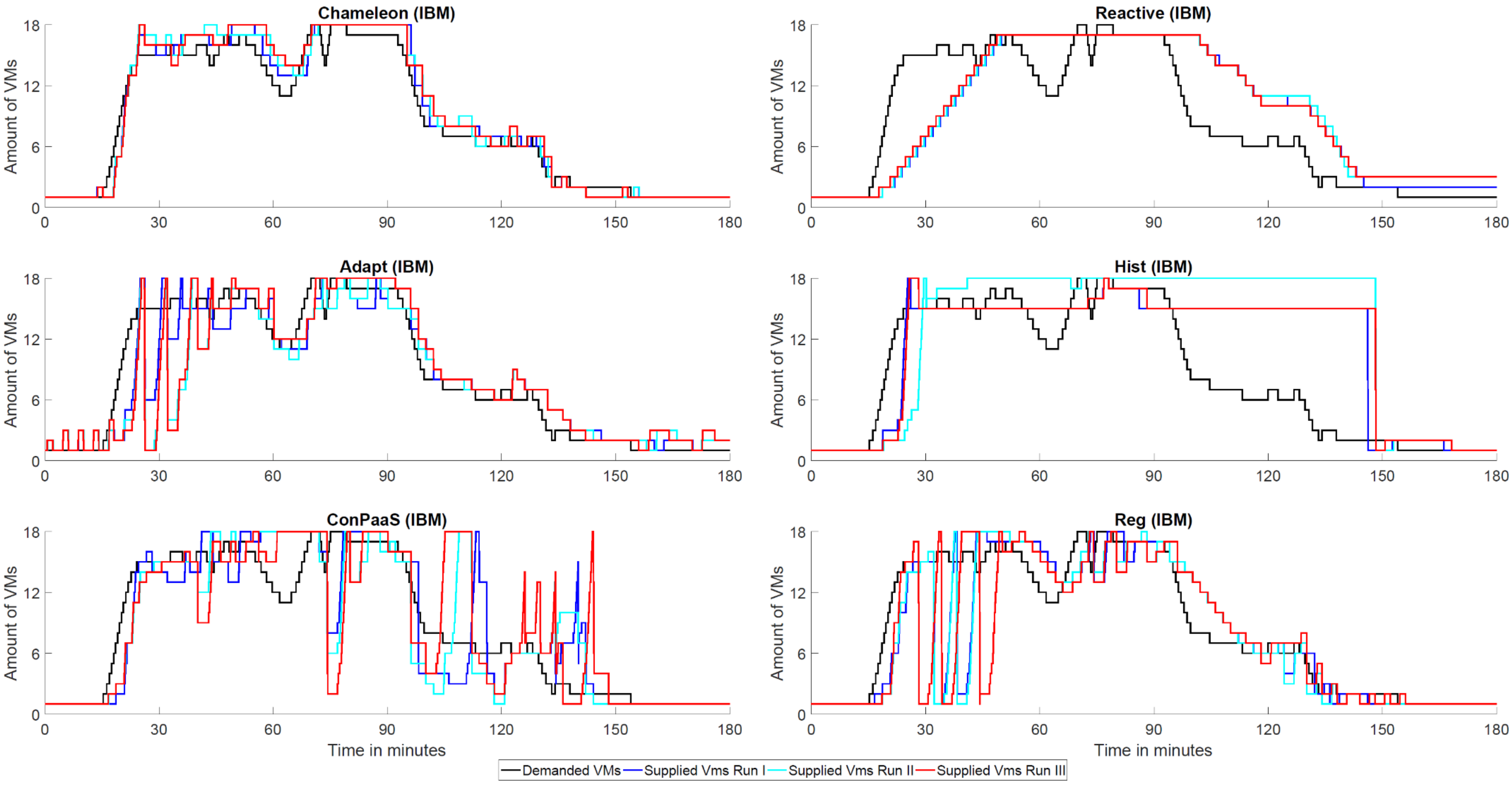
DESIGN AND EVALUATION OF A PROACTIVE, APPLICATION-AWARE AUTO-SCALER

CHAMELEON

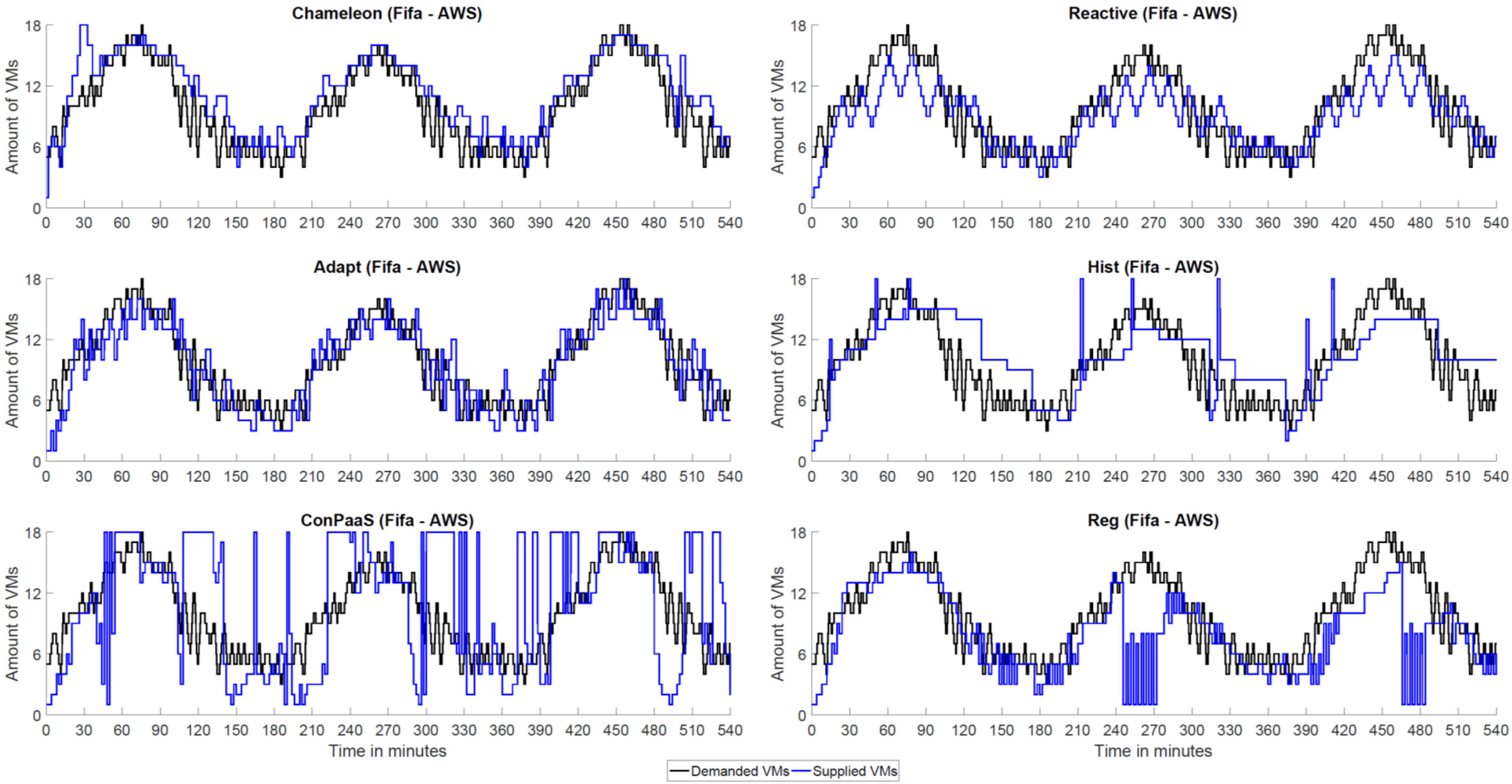
Chameleon‘s Architecture



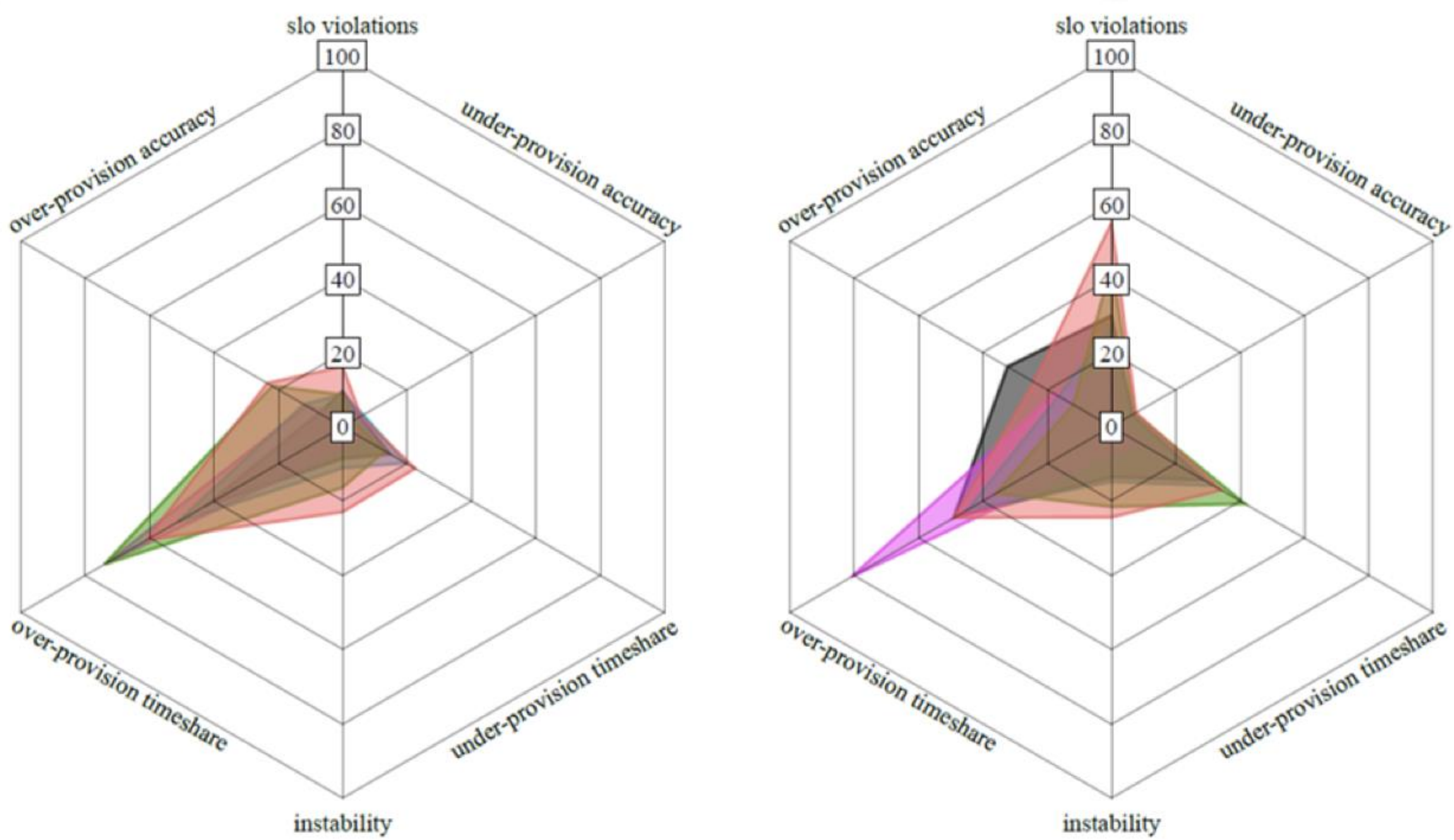
IBM Trace - 1 Day (3 runs)



3 Days Fifa 1998 in AWS EC2



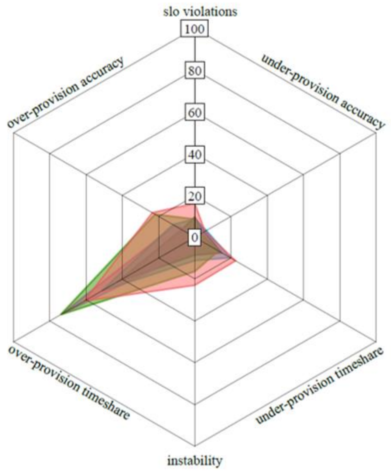
EVALUATION SUMMARY



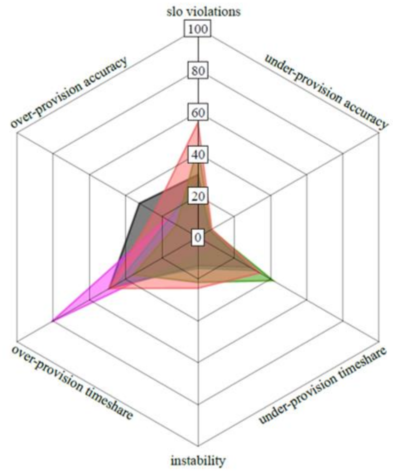
Metric overview Chameleon.

Metric overview Adapt.

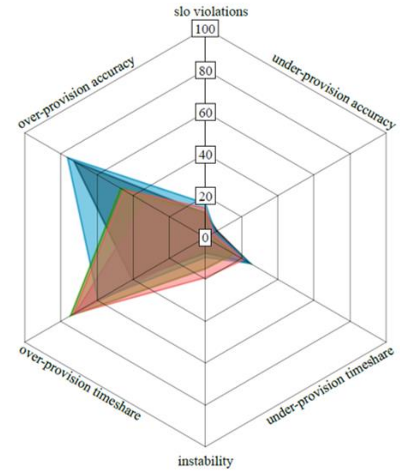
EVALUATION SUMMARY



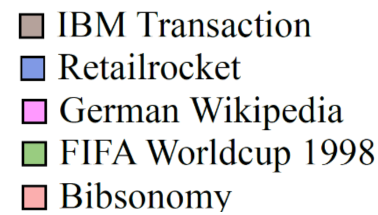
Metric overview Chameleon.

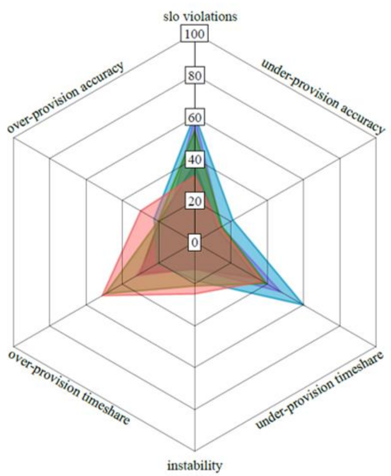


Metric overview Adapt.

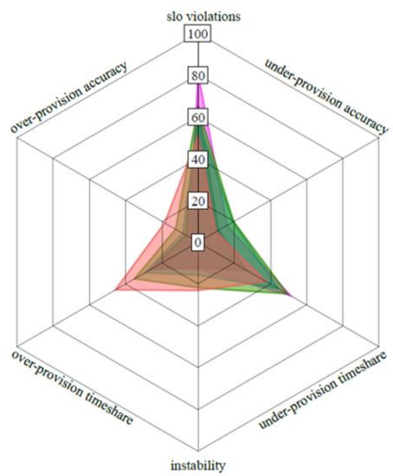


Metric overview Hist.

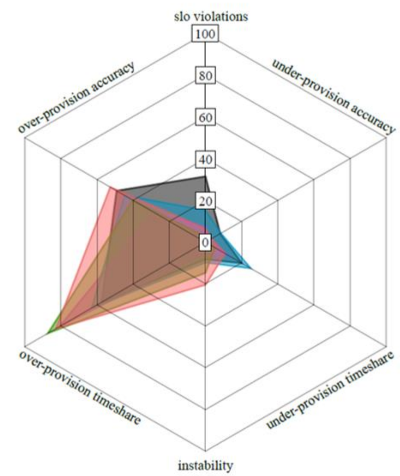




Metric overview ConPaaS.



Metric overview Reg.



Metric overview Reactive.

Mailing list at http://descartes.tools/

All measurements will be soon online on http://descartes.tools/chameleon

For further information see the Auto-Scaler Tutorial @ http://descartes.tools/

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]

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.

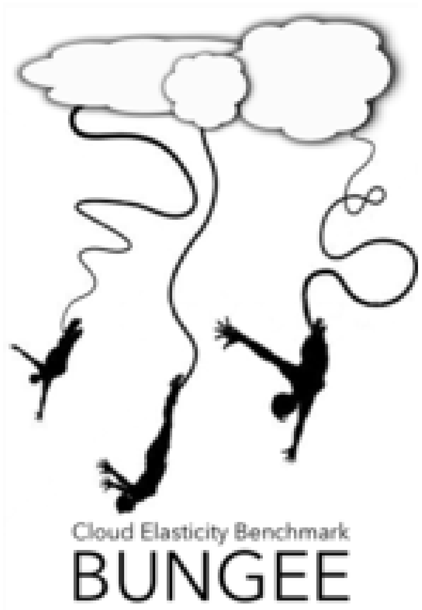
N. Herbst, S. Kounev and R. Reussner
Elasticity in Cloud Computing: What it is, and What it is Not.
in Proceedings of the 10th International Conference on Autonomic Computing (ICAC 2013), San Jose, CA, June 24-28, 2013.
[slides | http | .pdf]

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

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)

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

Acknowledgements

spec
spec Research
vmware
ORACLE
salesforce

- "Real-life" Scenarios
- Workload Traces
- Case Studies
- Results Dissemination

IBM
SIEMENS
SAP
ABB
HUAWEI
CAS SOFTWARE
blue yonder

Imperial College London
TU Delft
Universidade de Coimbra
Universität Stuttgart
Università della Svizzera italiana
UCD Dublin
Universitat Politècnica de Catalunya
Barcelona Supercomputing Center – Centro Nacional de Supercomputación
Technische Universität Berlin

Julius-Maximilians-
UNIVERSITÄT WÜRZBURG

Descartes research

Questions?

skounev@acm.org

http://descartes.tools

http://descartes-research.net