Karlsruhe Institute of Technology

Descartes research

Towards Self-Aware Dependability Management in Virtualized Service Infrastructures

Samuel Kounev

www.descartes-research.net

Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft

Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Agenda

- Descartes Research Group @ KIT
- Challenges Posed by Cloud Computing
- Resource Management in the Cloud
- Vision and Research Roadmap
- Initial Steps and Preliminary Proof-of-Concept

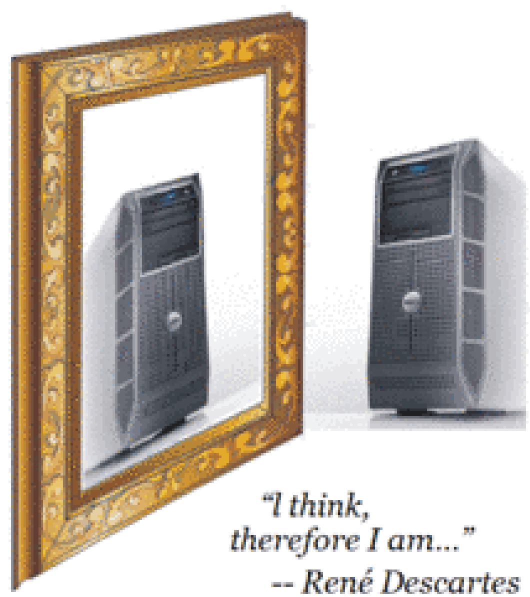
S. Kounev, F. Brosig, N. Huber, and R. Reussner. **Towards self-aware performance and resource management in modern service-oriented systems**. In *Proc. of the 7th IEEE Intl. Conference on Services Computing (SCC 2010), July 5-10, Miami, Florida, USA*.

The Descartes Research Group @ KIT

Karlsruhe Institute of Technology

- Named after the french philosopher René Descartes
- Funding: DFG, KIT, EU, Industry
- Focus: Engineering of *Self-Aware* Software Systems
- www.descartes-research.net

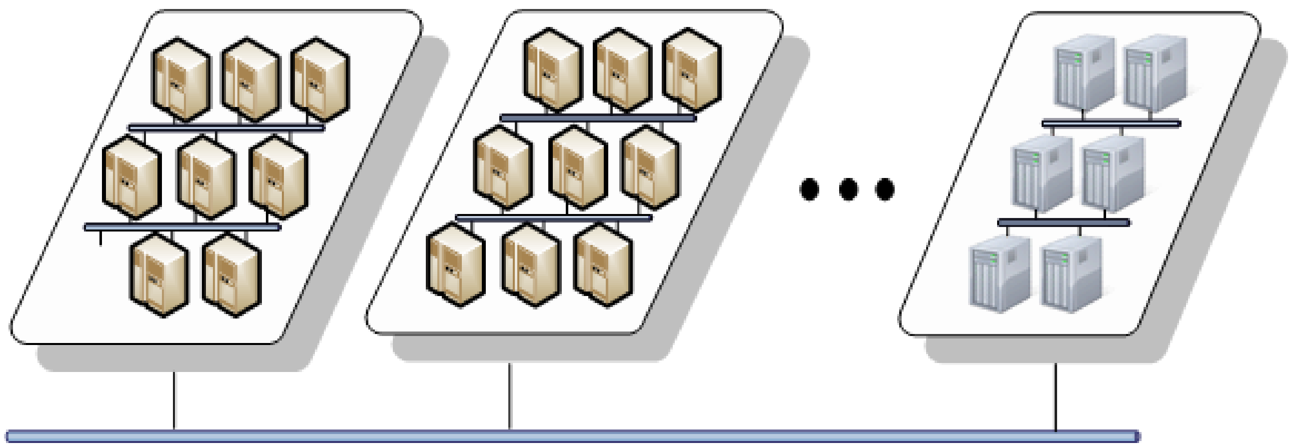
Descartes research



Emmy Noether-Programm Deutsche Forschungsgemeinschaft DFG

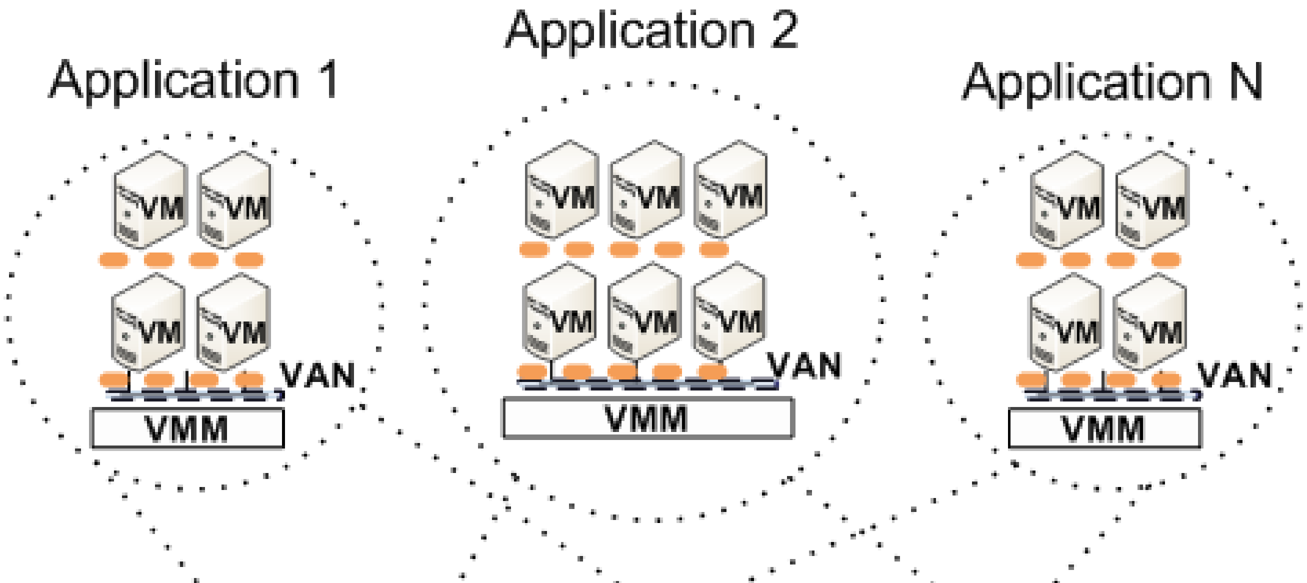
Deutsche Forschungsgemeinschaft DFG

Traditional Data Center Infrastructures



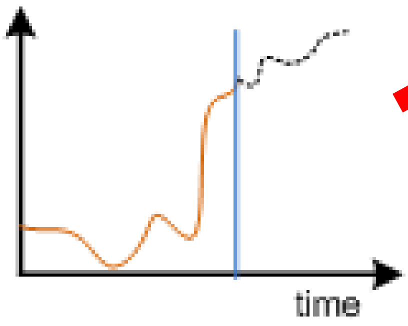
- Applications running on dedicated hardware
- Over-provisioned system resources
- Poor resource utilization and energy efficiency
- Increasing number of servers → rising operating costs

Cloud Computing Infrastructures

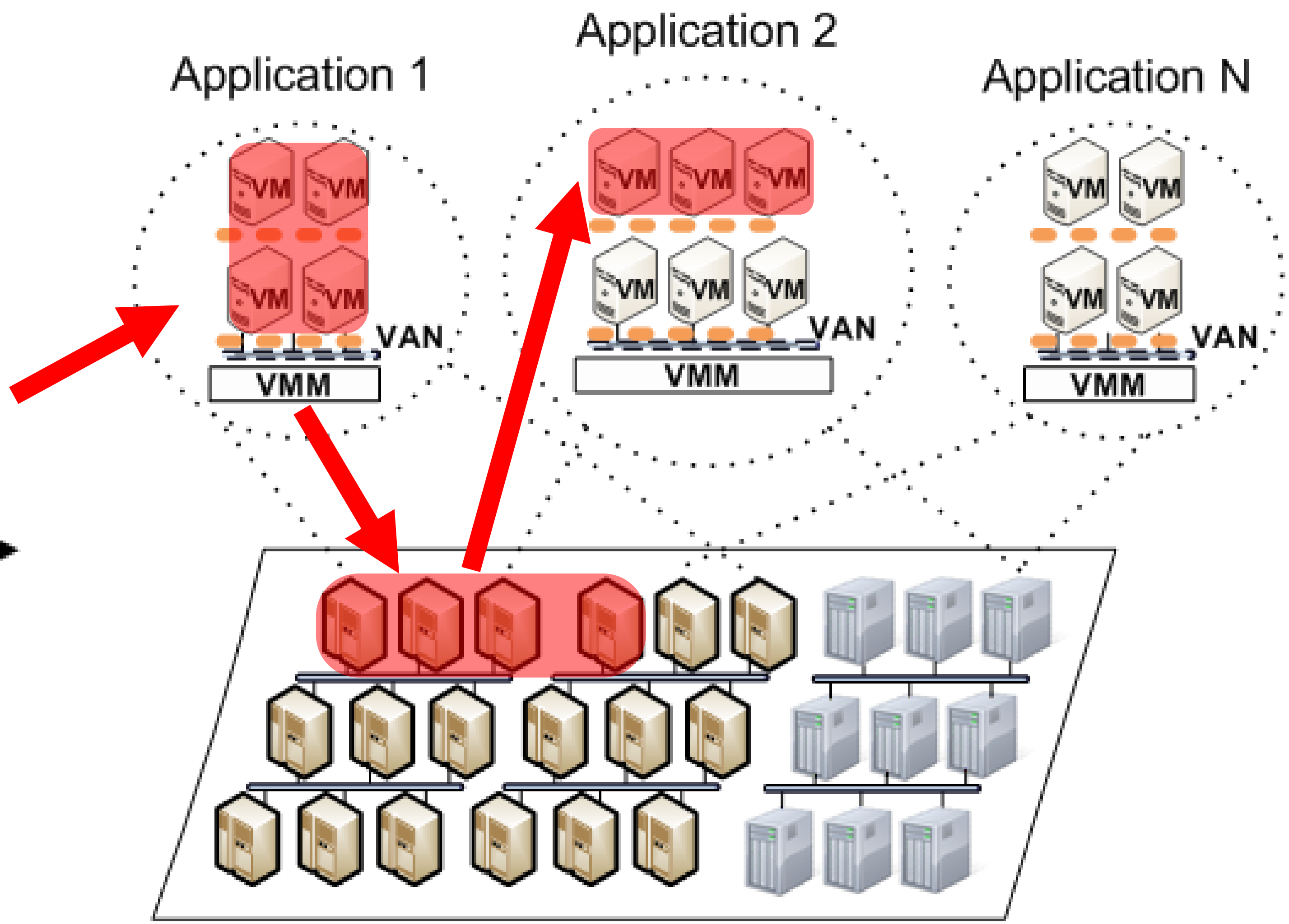


- Applications running in a virtualized environment
- Shared physical infrastructure
- Flexible mapping of logical to physical resources
- Higher resource utilization & energy efficiency
- Lower operating costs

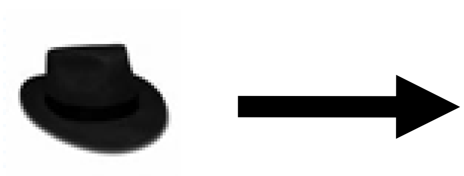
Cloud Computing Infrastructures (2)



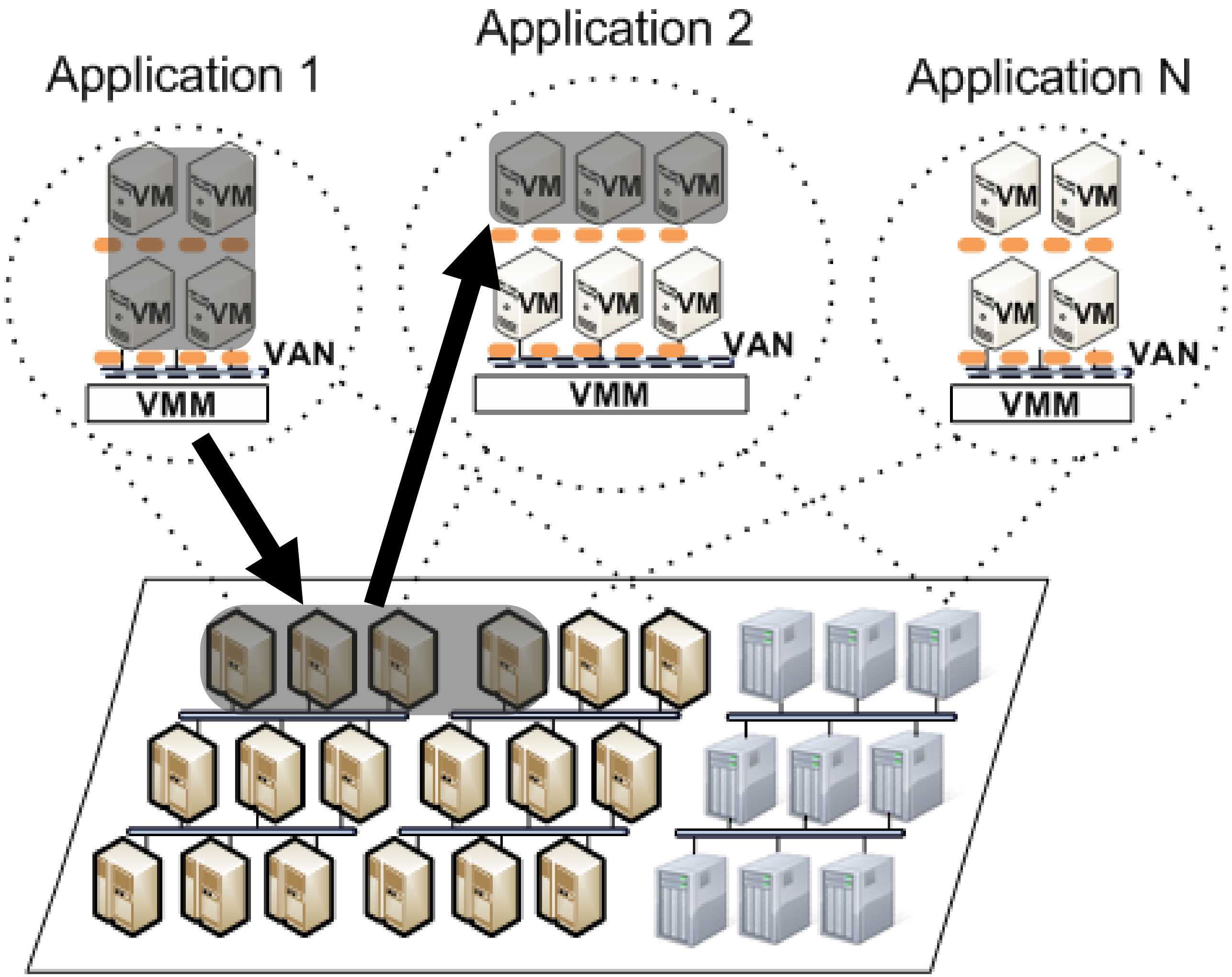
Load Spike



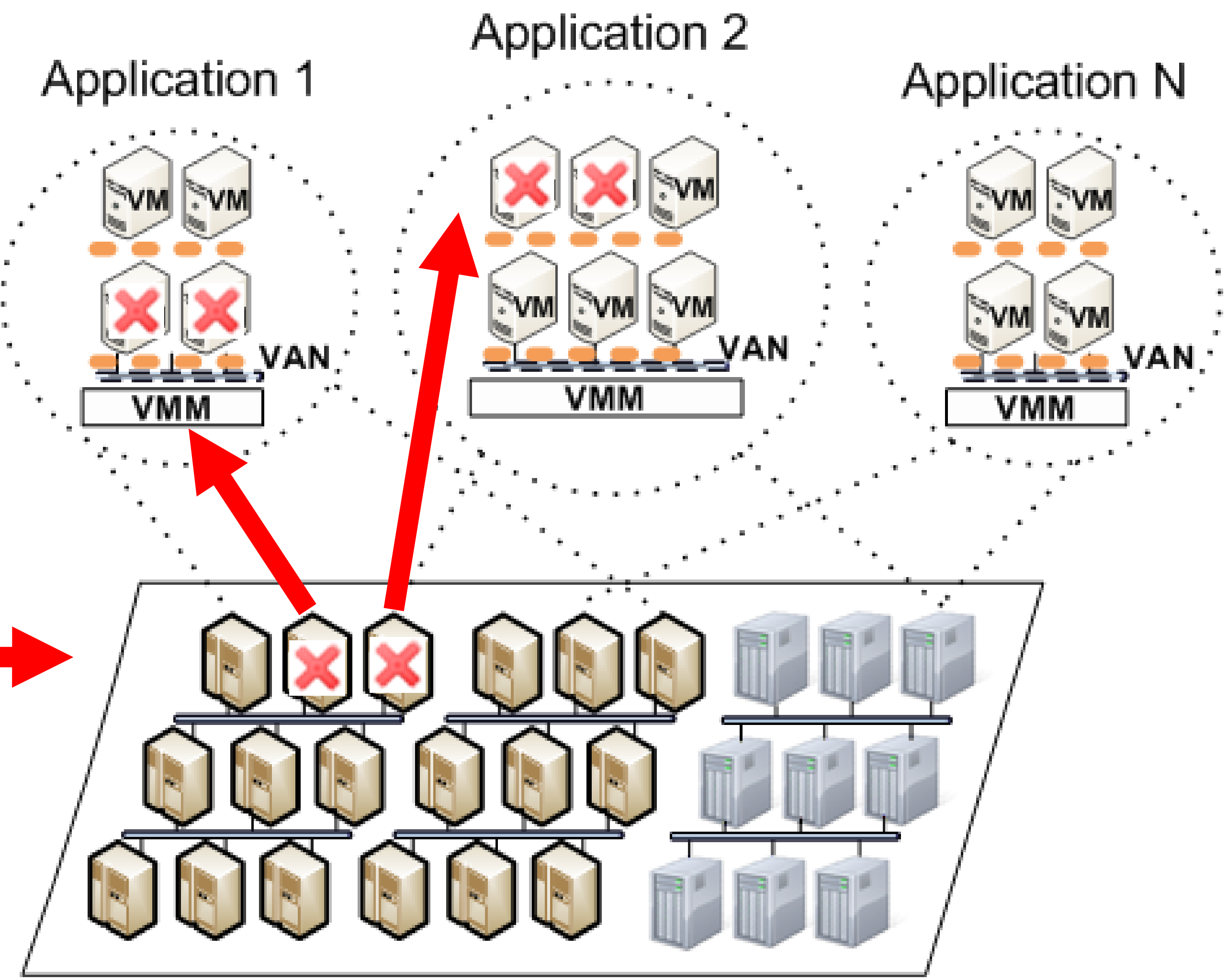
Cloud Computing Infrastructures (3)



Network Attack / Intrusion



Cloud Computing Infrastructures (4)

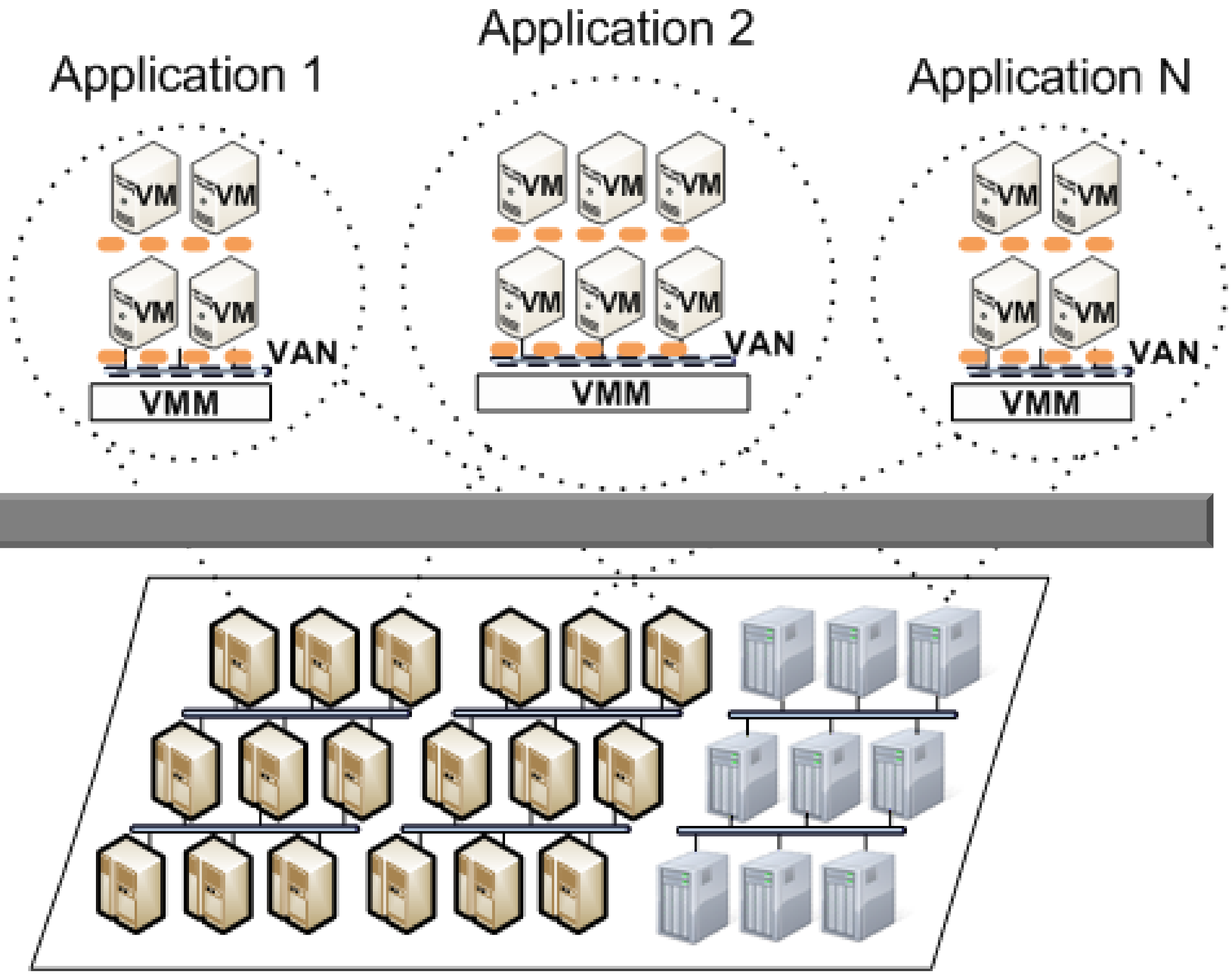


Hardware Failure

Cloud Computing Infrastructures (5)

Service Provider (SP)

Infrastructure Provider (IP)



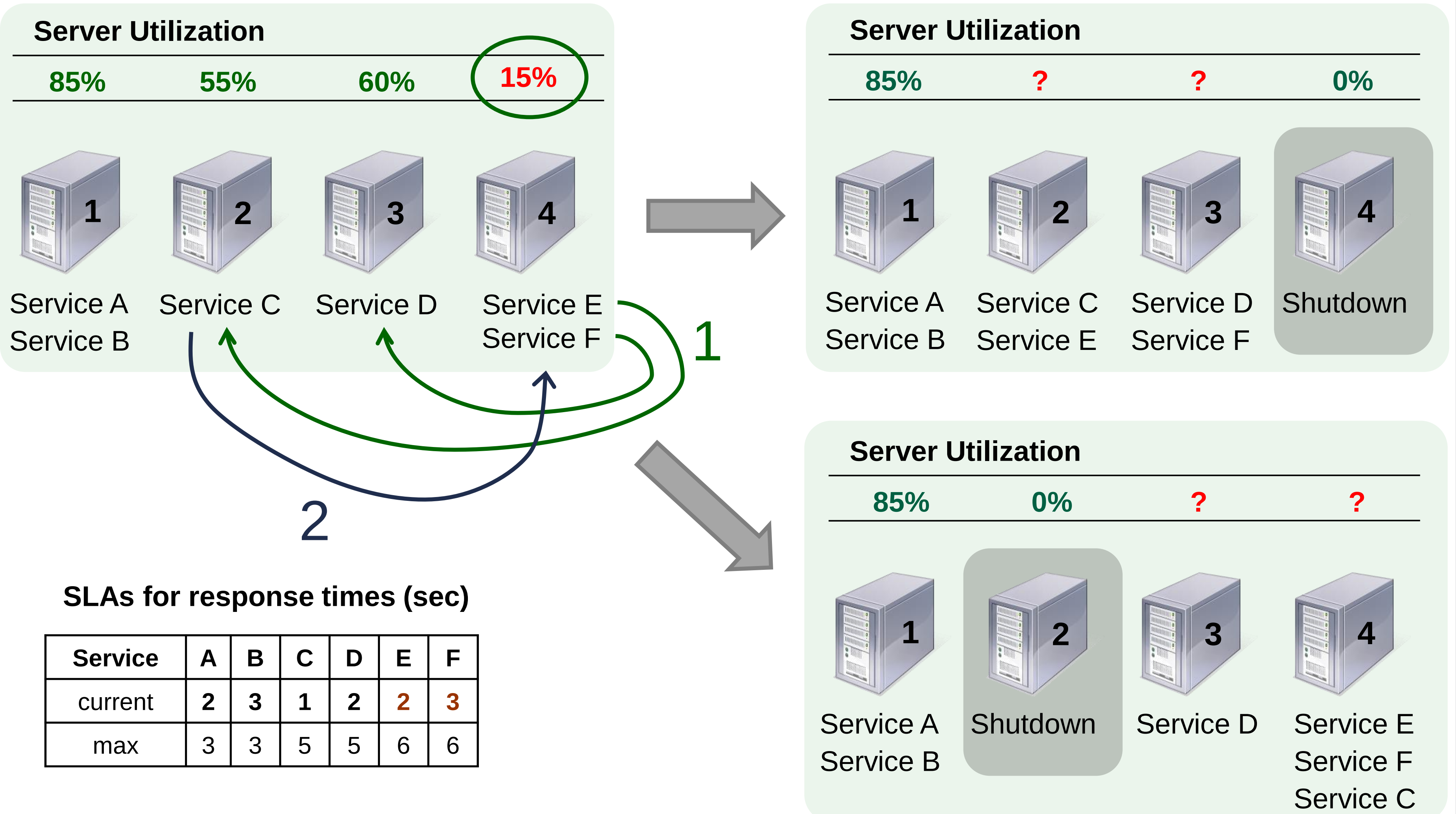
Challenges Posed by Cloud Computing

- Increased system complexity and dynamics
- Lack of direct control over underlying hardware
- New threats and vulnerabilities due to resource sharing
- Separation of service providers and infrastructure providers

↓

- Inability to provide QoS and dependability guarantees
- Lack of trust

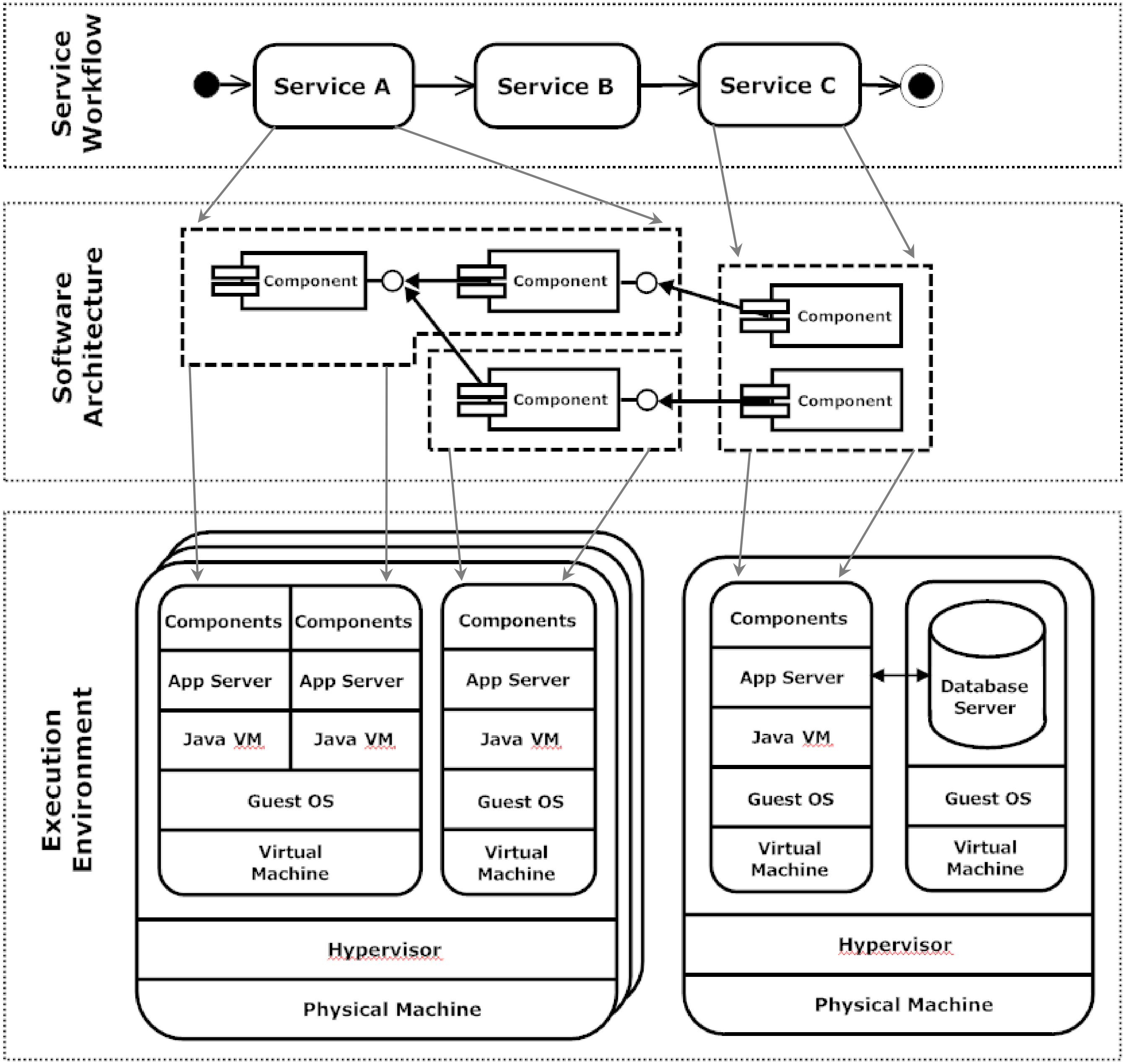
Run-time Performance Management



SLAs for response times (sec)

Service	A	B	C	D	E	F
current	**2**	**3**	**1**	**2**	**2**	**3**
max	3	3	5	5	6	6

Modern Service-Oriented System



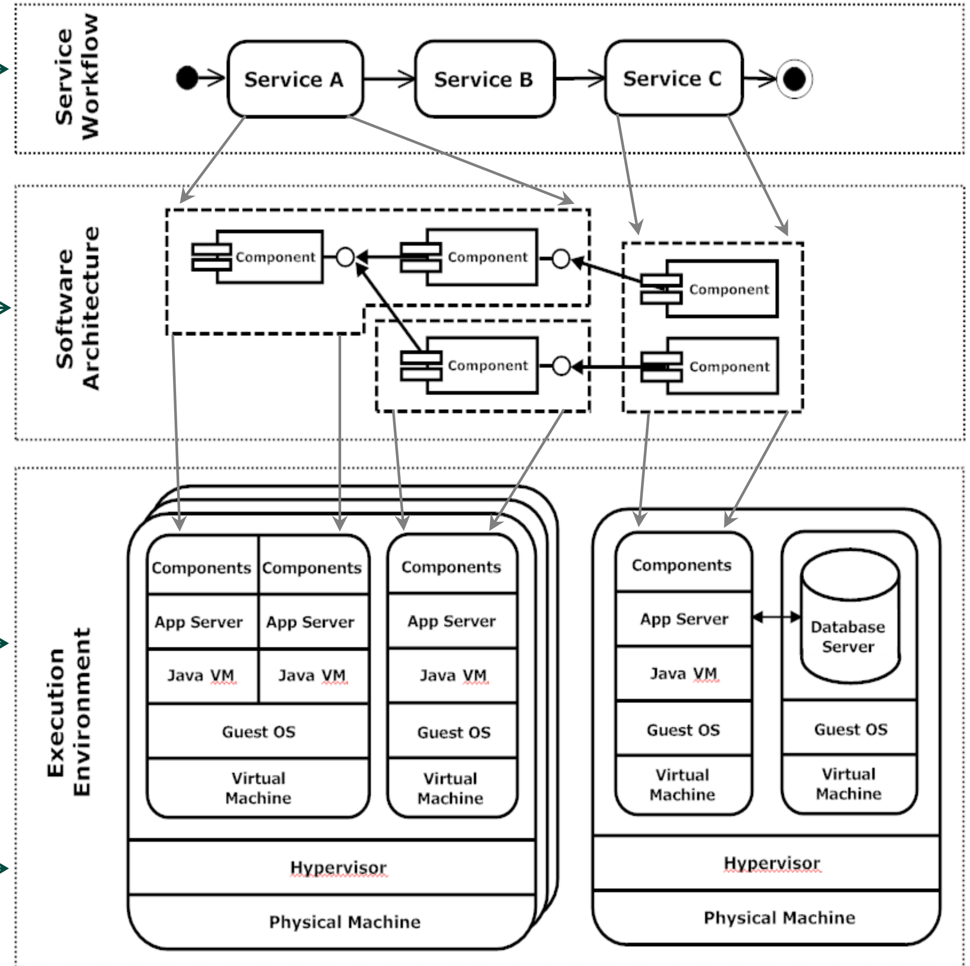
Modern Service-Oriented System

System workload and usage profile
- Number and type of clients
- Input parameters and input data
- Data formats used
- Service workflow

Software architecture
- Connections between components
- Flow of control and data
- Component resource demands
- Component usage profiles

Execution environment
- Number of component instances
- Server execution threads
- Amount of Java heap memory
- Size of database connection pools

Virtualization layer
- Physical resources allocated to VMs
  - – number of physical CPUs
  - – amount of physical memory
  - – secondary storage devices



Network bandwidth between system nodes

State-of-the-Art

Karlsruhe Institute of Technology

1. Performance prediction at design & deployment time

- Descriptive architecture-level performance meta-models
  - E.g., PCM, SPE-MM, CSM, CBML, KLAPER, UML SPT, UML MARTE
- Automated transformation to predictive models
  - E.g., layered queueing networks, stochastic Petri nets

Main issues:

- Overhead in building and analyzing models
- Models assume static system architecture
- Maintaining models during operation is prohibitively expensive

[M. Woodside et al], [D. Petriu et al], [R. Reussner et al], [C. Smith et al],
[R. Mirandola et al], [K. Trivedi et al], [V. Cortellessa et al], [I. Gorton et al],
[J. Merseguer et al], [D. Menasce et al], [E. Eskenazi et al], [J. Murphy et al],…

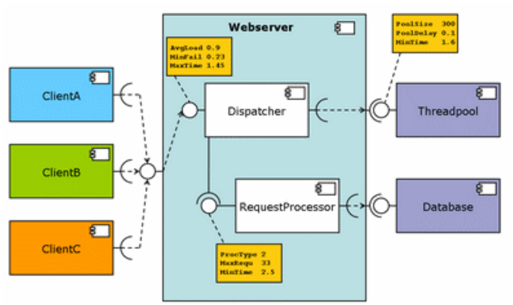
State-of-the-Art (2)

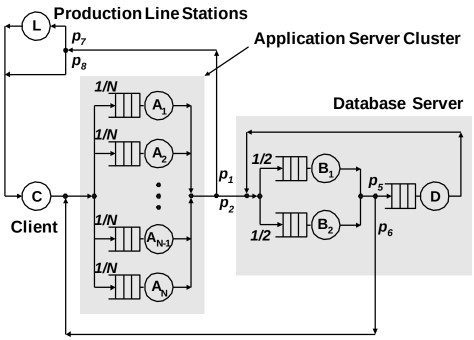
2. Performance and resource management at run-time

- Simple models used that abstract the system at very high level
- Services modeled as black boxes
- Restrictive assumptions often imposed, e.g.:
  - Single workload class
  - Homogeneous servers
  - Single-threaded components
  - Exponential request interarrival times and service times
- Layers of the execution environment not modeled explicitly

[G. Pacifici et al], [A. D'Ambrogio et al], [G. Tesauro et al], [D. Menasce et al],
[C. Adam et al], [Rashid A. Ali et al], [I. Foster er al], [S. Bleul et al],
[A. Othman et al], [P. Shivam et al], [R. Berbner et al], [H. Song et al],…

The Past

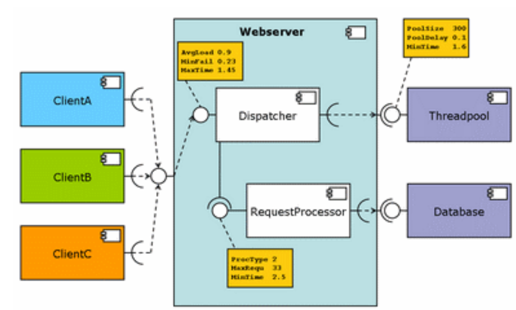


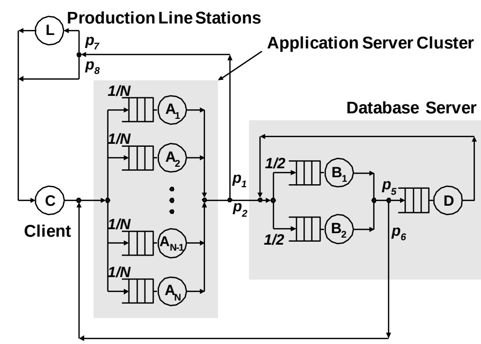


System

Models

The Future





System **Models**

The Future

"I think, therefore I am …"
-- René Descartes

Vision

Next generation ***self-aware*** software systems:

1. Aware of their architecture and the environment they are running in

2. Aware of internal and external changes and able to predict their effect
   a) External changes, e.g., evolving service workloads
   b) Internal changes, e.g., dynamically undertaken reconfiguration actions

   "thought (cogitatio) is what happens in me such that I am immediately conscious of it…" -- Rene Descartes

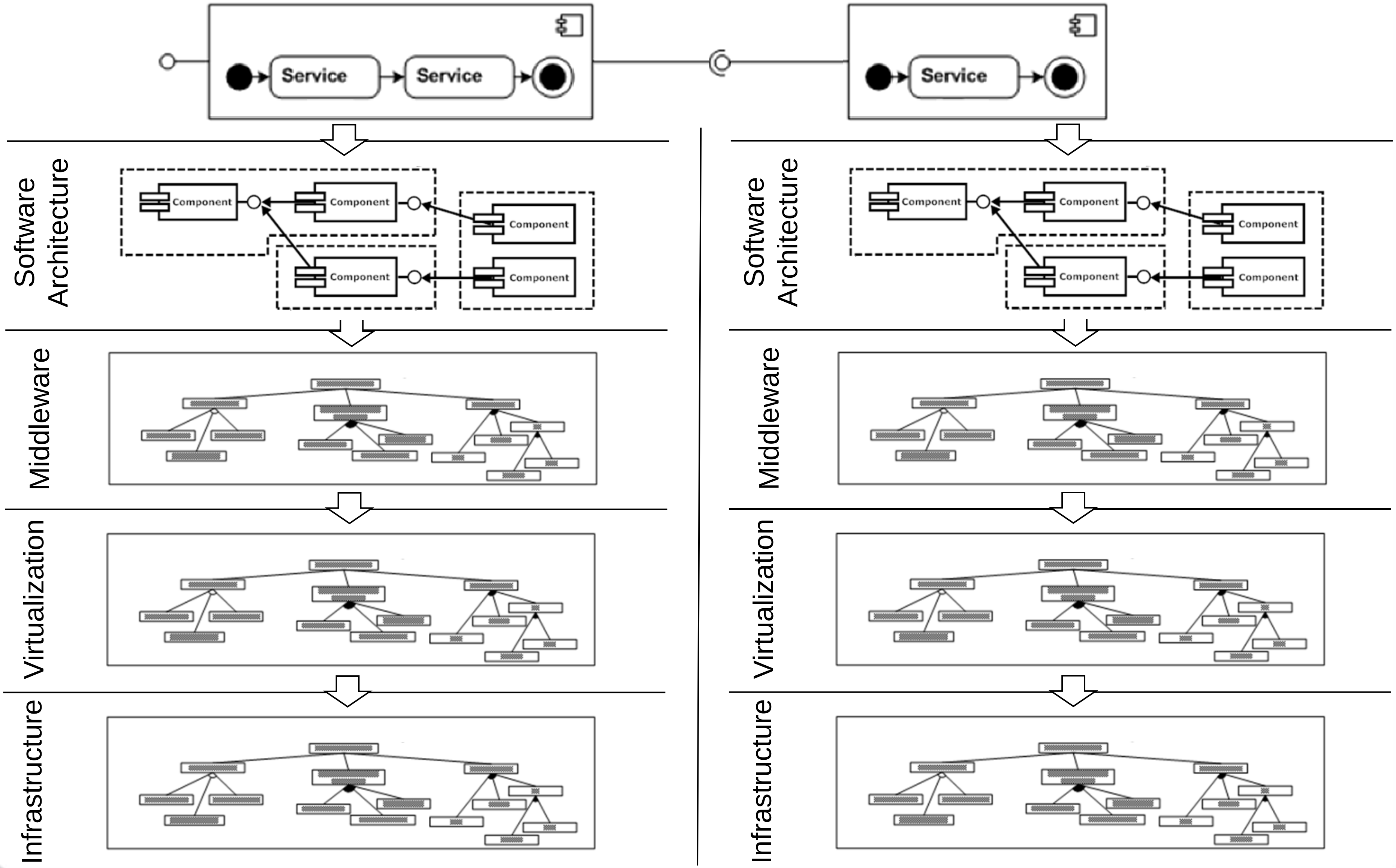
3. Proactively adapting to enforce QoS and resource efficiency

   "For it is not enough to have a good mind: one must use it well" -- Rene Descartes

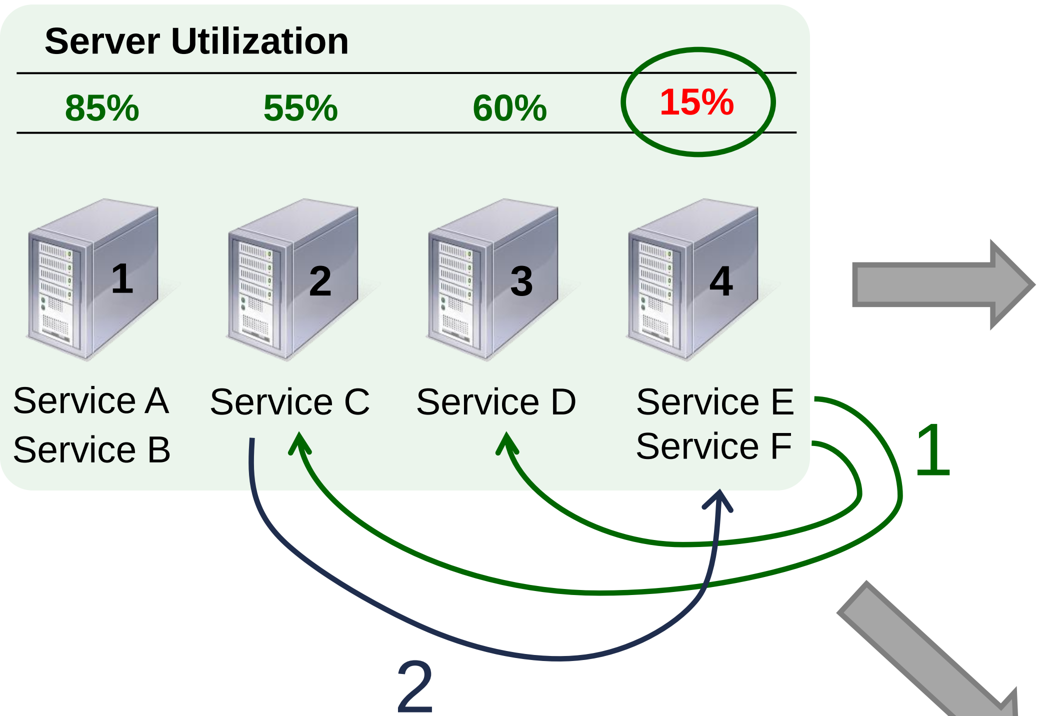
4. Based on integrated *dynamic* QoS prediction models

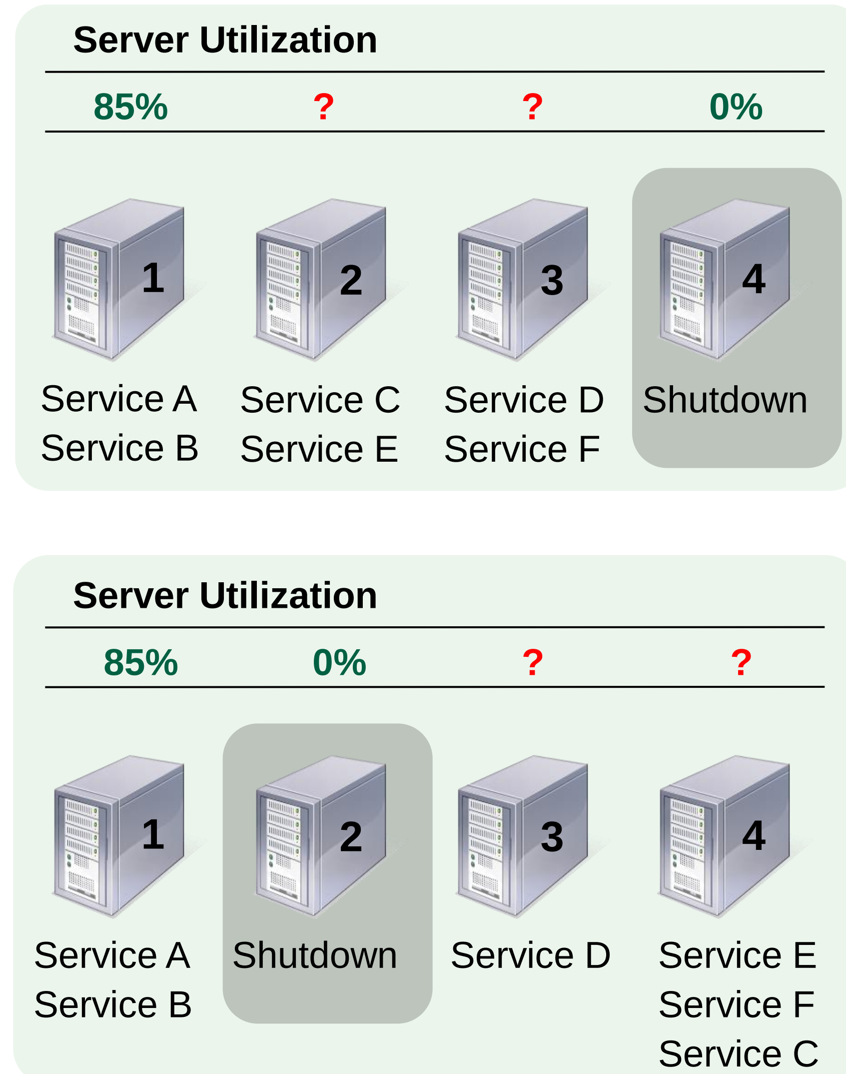
   Dualism: "the mind controls the body,
   but that the body can also influence the mind" -- Rene Descartes

Dynamic Model Composition



Run-time Performance Management

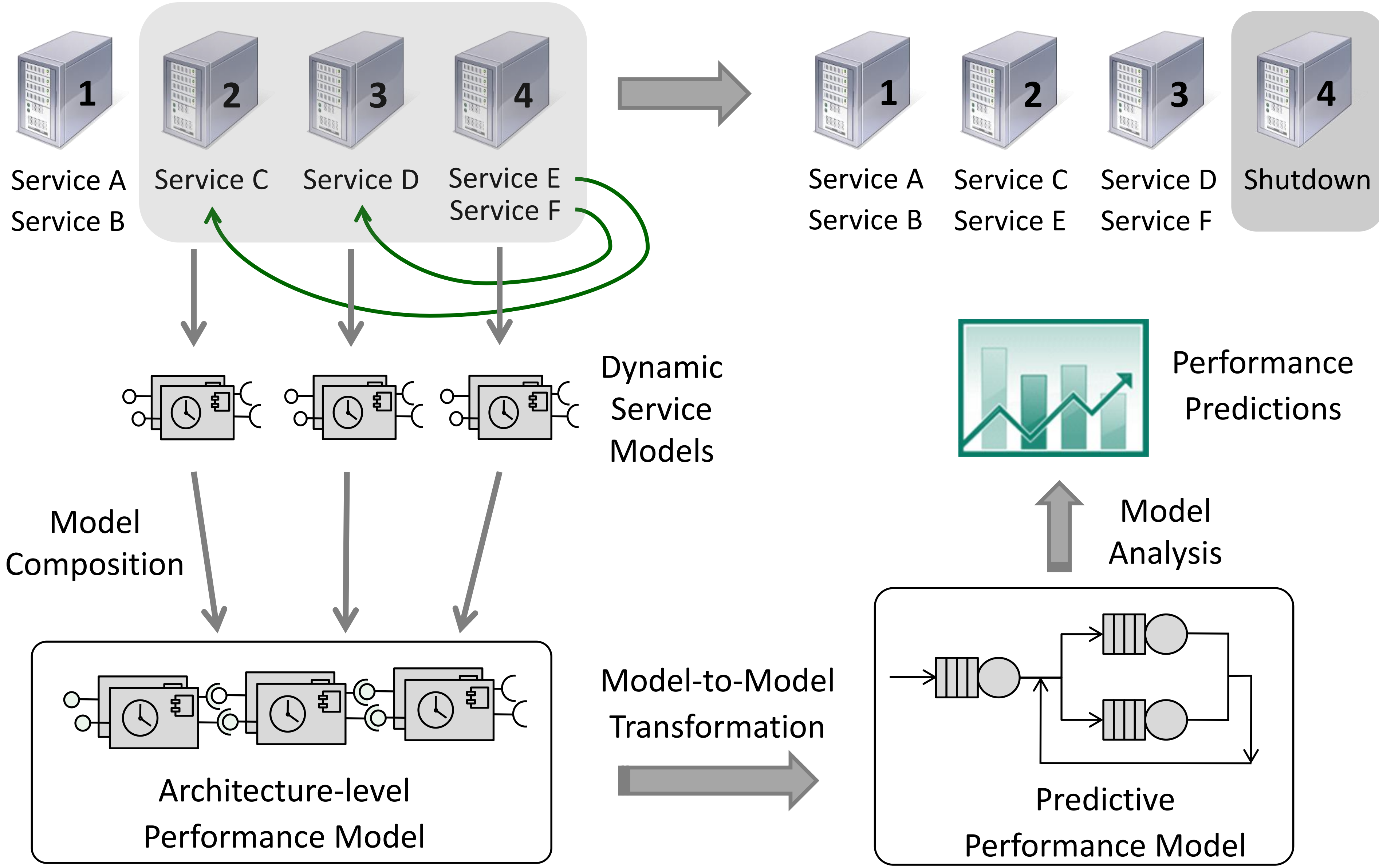




SLAs for response times (sec)

Service	A	B	C	D	E	F
current	2	3	1	2	2	3
max	3	3	5	5	6	6

Performance Prediction On-The-Fly



Performance Prediction On-The-Fly (2)

Server Utilization

85%	55%	60%	30%

SLAs for Resp. Times (sec)

Service	A	B	C	D	E	F
current	2	3	1	2	2	3
max	3	3	5	5	6	6

1

Server Utilization

85%	70%	80%	0%

SLAs for Resp. Times (sec)

Service	A	B	C	D	E	F
current	2	3	3	4	4	5
max	3	3	5	5	6	6

2

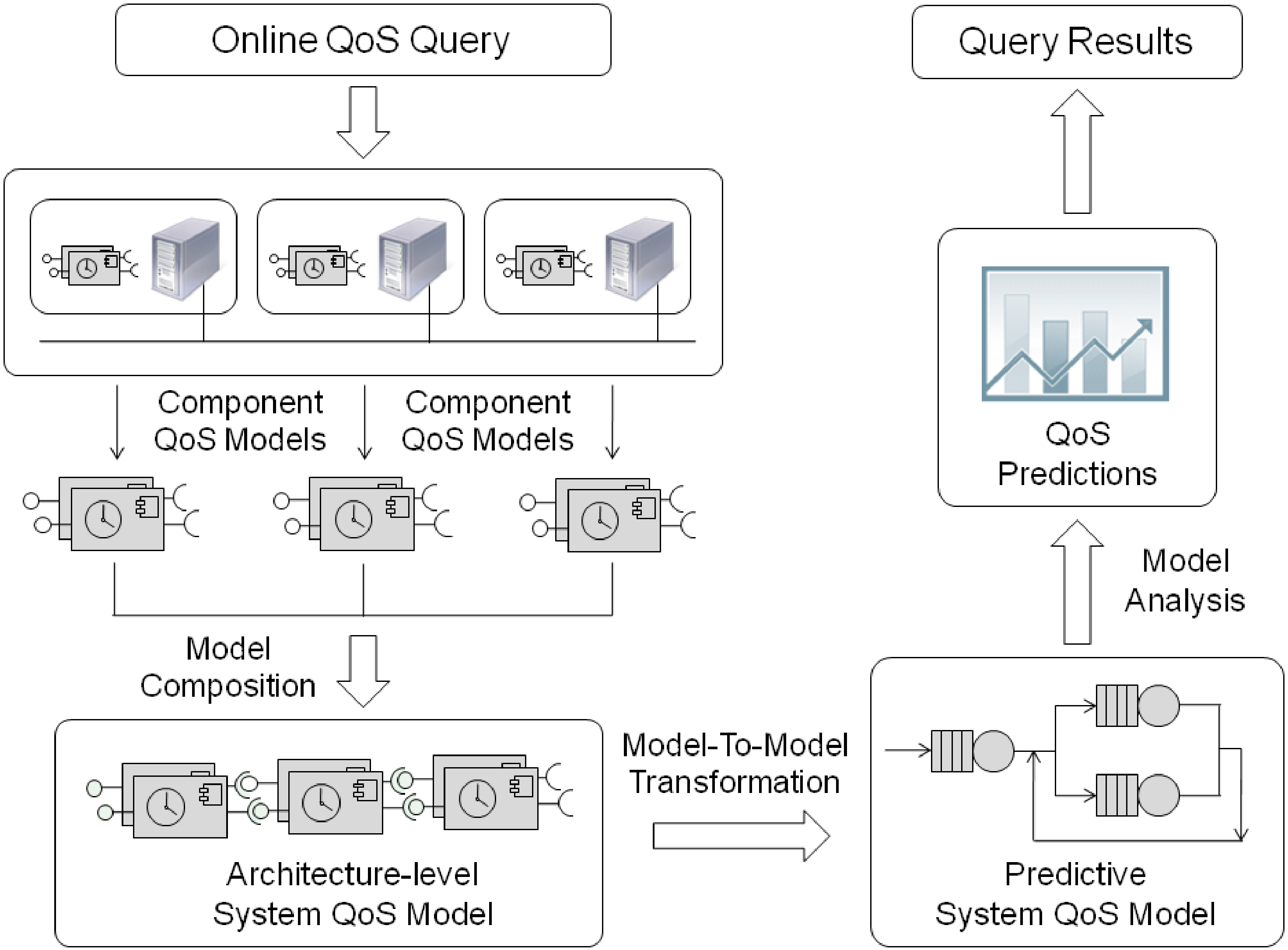
Server Utilization

85%	0%	60%	90%

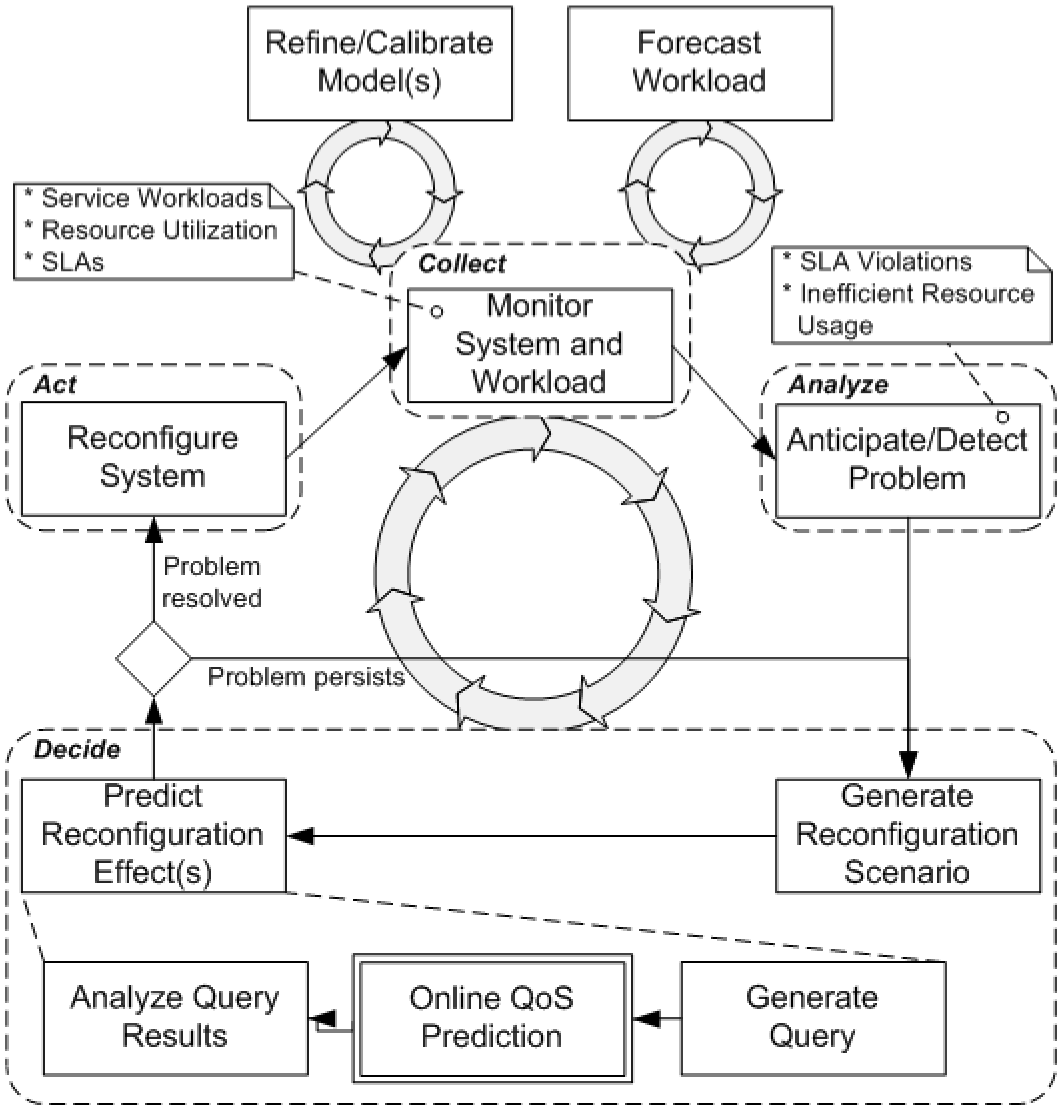
SLAs for Resp. Times (sec)

Service	A	B	C	D	E	F
current	2	3	4	2	6	7
max	3	3	5	5	6	6

Generalized Online Prediction Process



System Control Loop

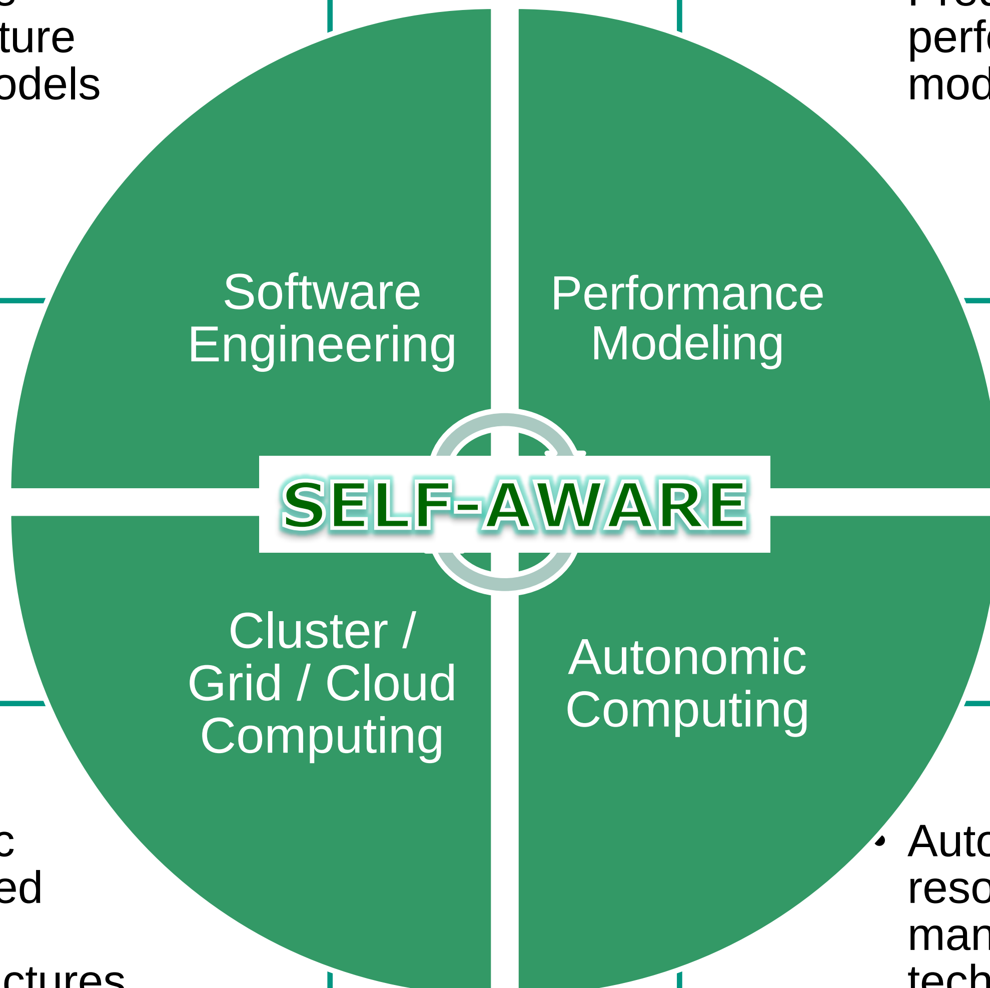


Input from Multiple Communities

Karlsruhe Institute of Technology

- Software architecture meta-models

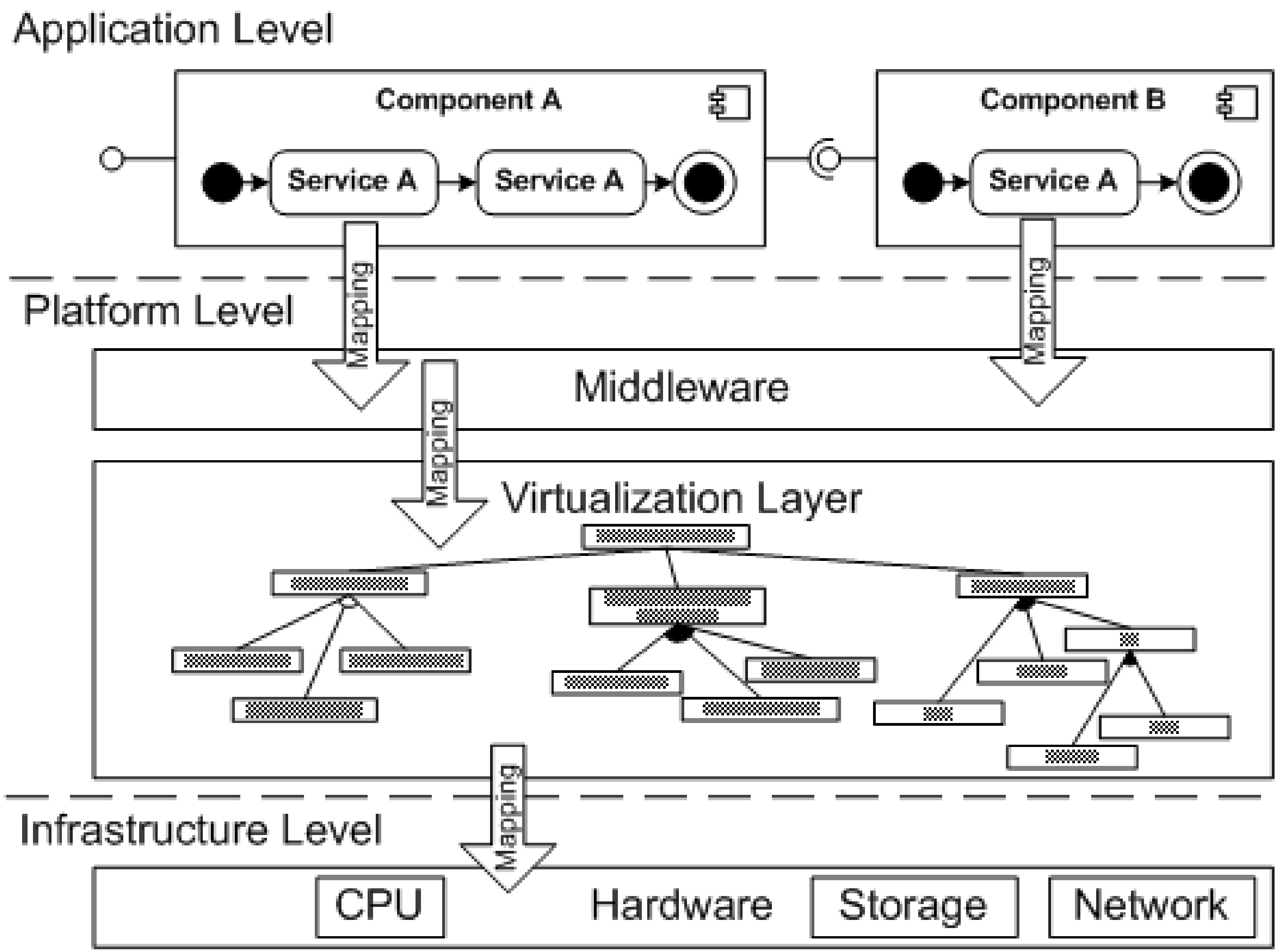
- Predictive performance models



- Dynamic virtualized service infrastructures

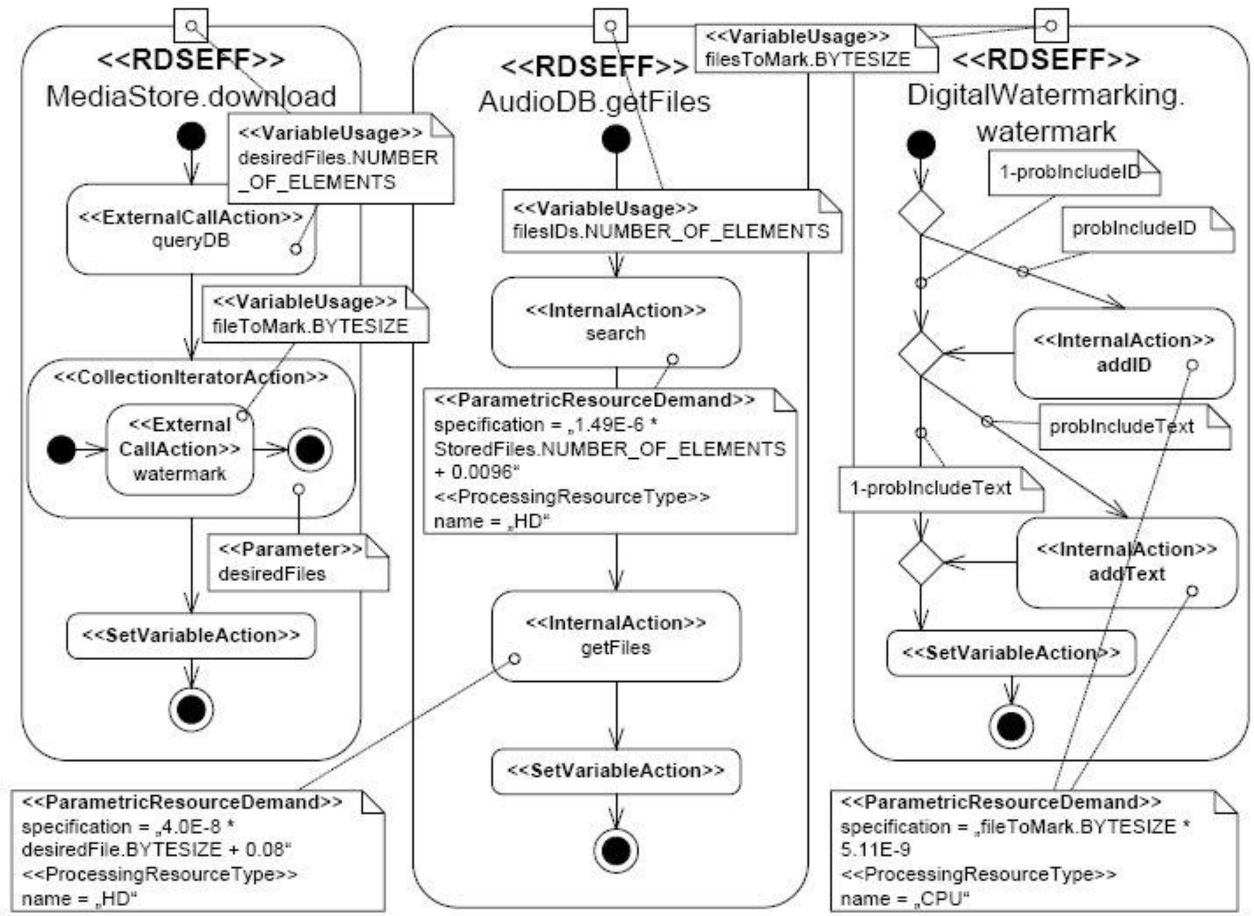
- Autonomic resource management techniques

Initial Steps: Meta-Model for Dynamic Systems



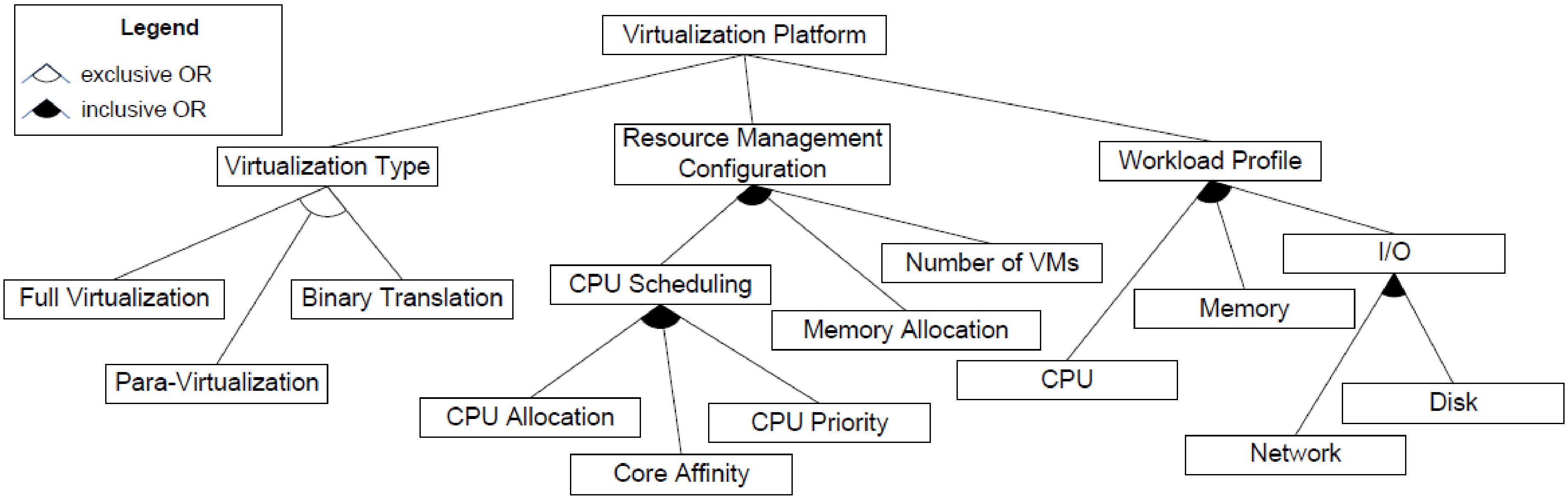
Initial Steps: Application Level

Karlsruhe Institute of Technology



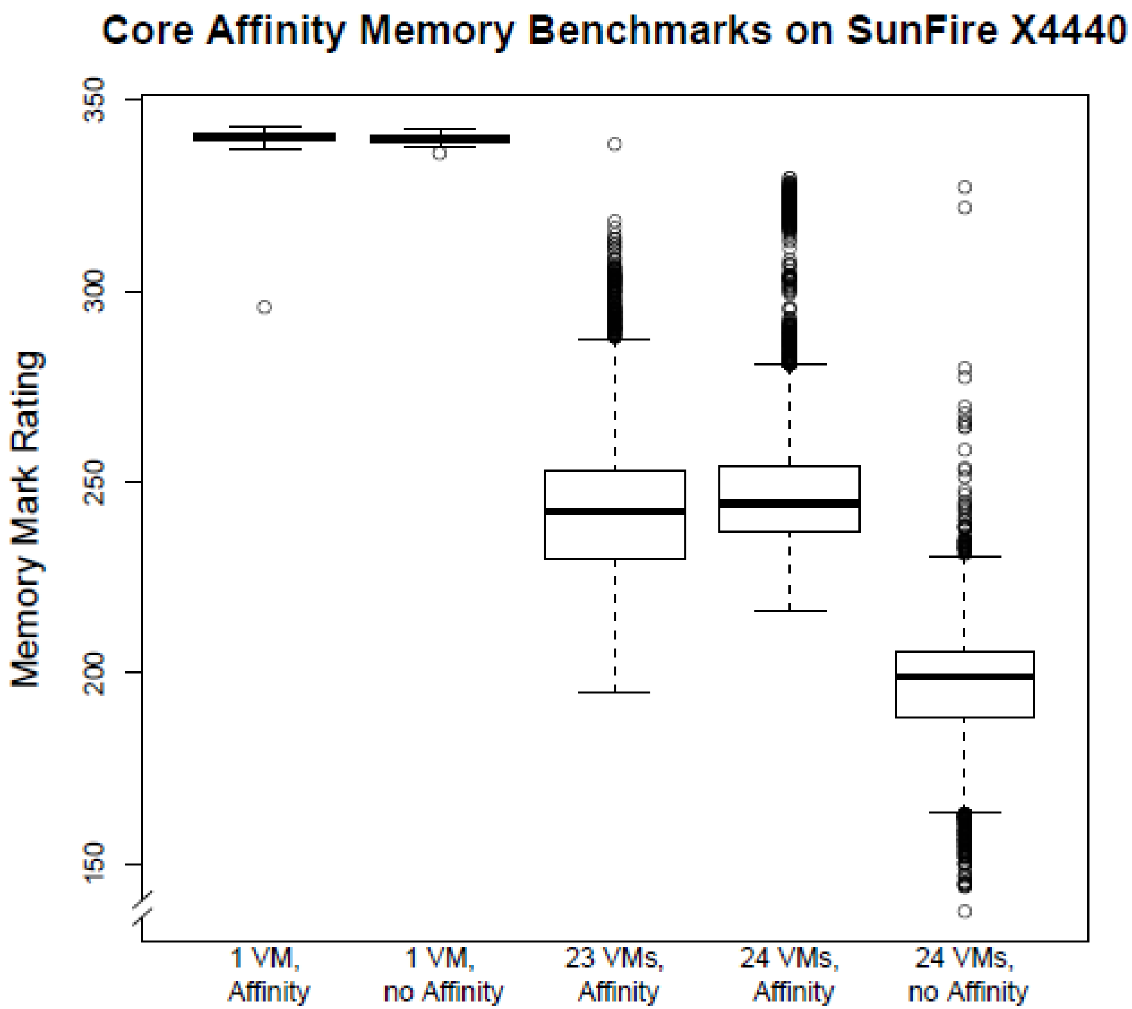
S. Becker, H. Koziolek, and R. Reussner. **„The Palladio component model for model-driven performance prediction“**. *Journal of Systems and Software*, 82:3-22, 2009.

Initial Steps: Platform Level



Nikolaus Huber, Marcel von Quast, Fabian Brosig, and Samuel Kounev. **”Analysis of the Performance-Influencing Factors of Virtualization Platforms”**. In *OTM 2010 Conferences - Distributed Objects, Middleware, and Applications (DOA'10)*. Springer Verlag, 2010.

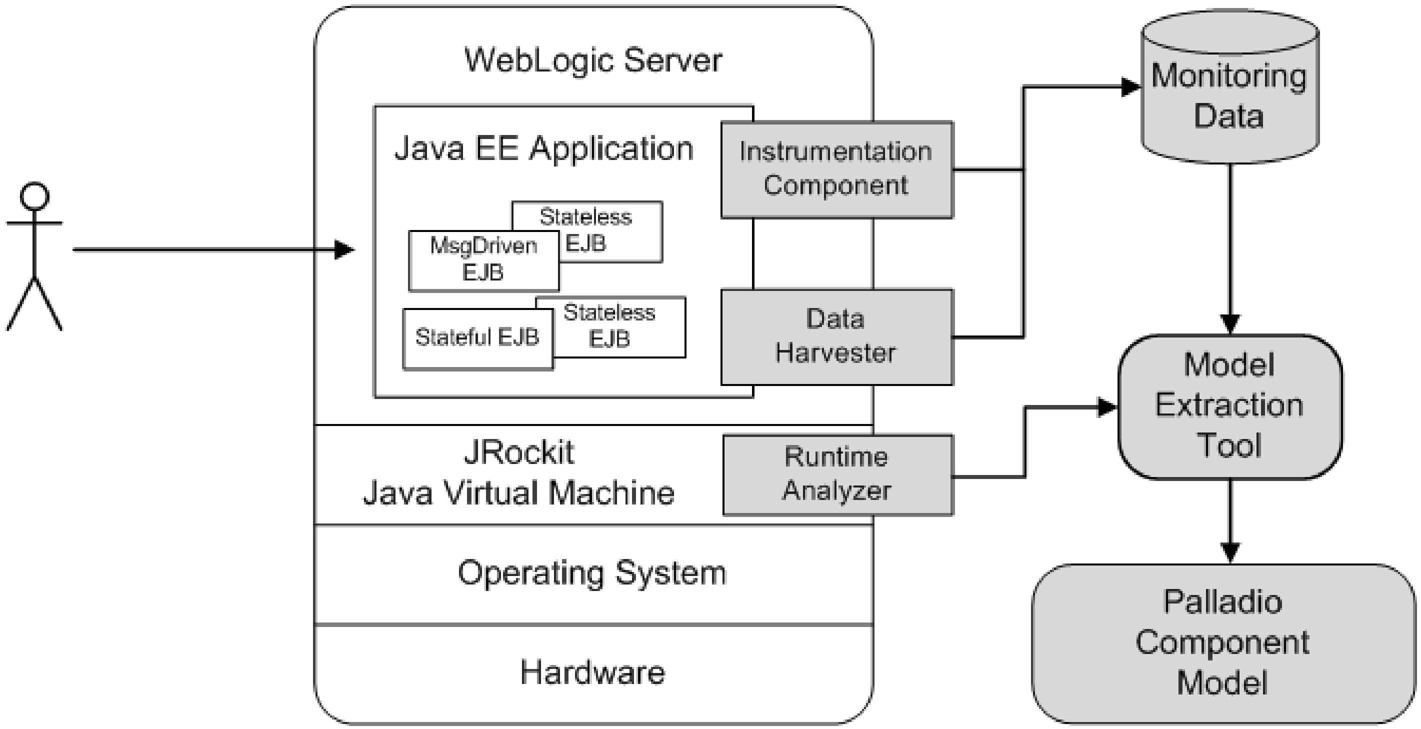
Scaling Number of Co-Located VMs



Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft
Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Case Study: Automated Model Extraction

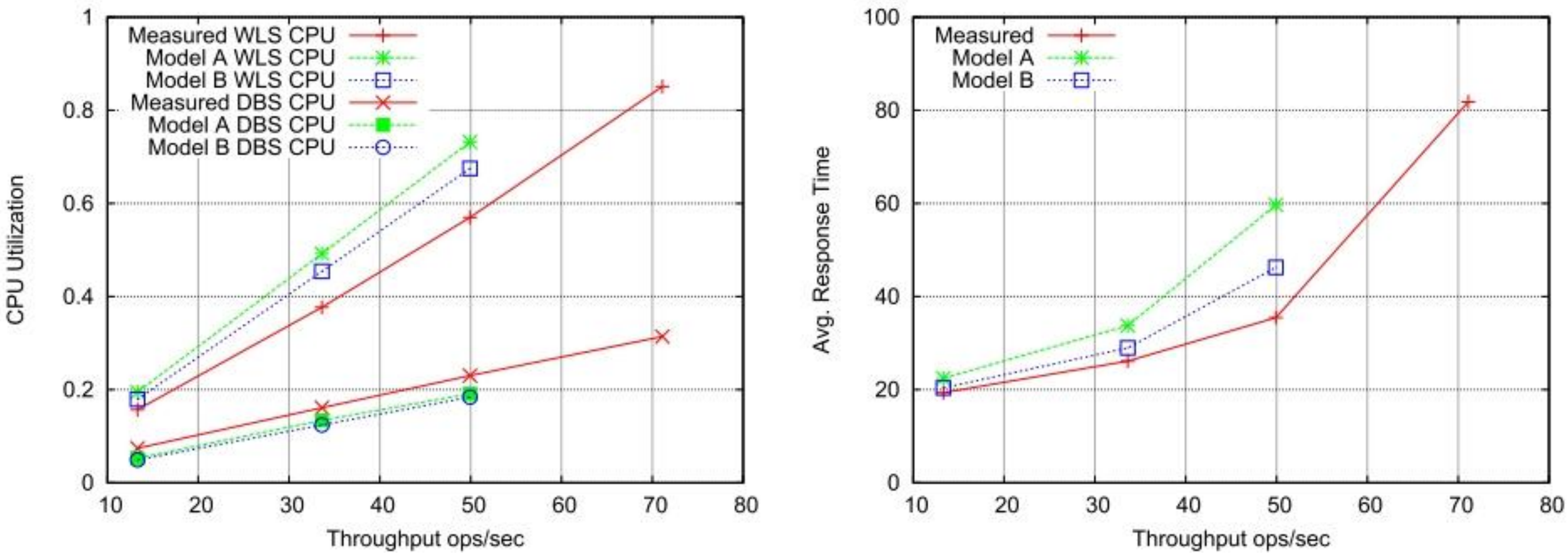
Extracting architecture-level performance models from online monitoring data



F. Brosig, S. Kounev, and K. Krogmann. „**Automated Extraction of Palladio Component Models from Running Enterprise Java Applications”**. In Proc. of ROSSA 2009. ACM Press.

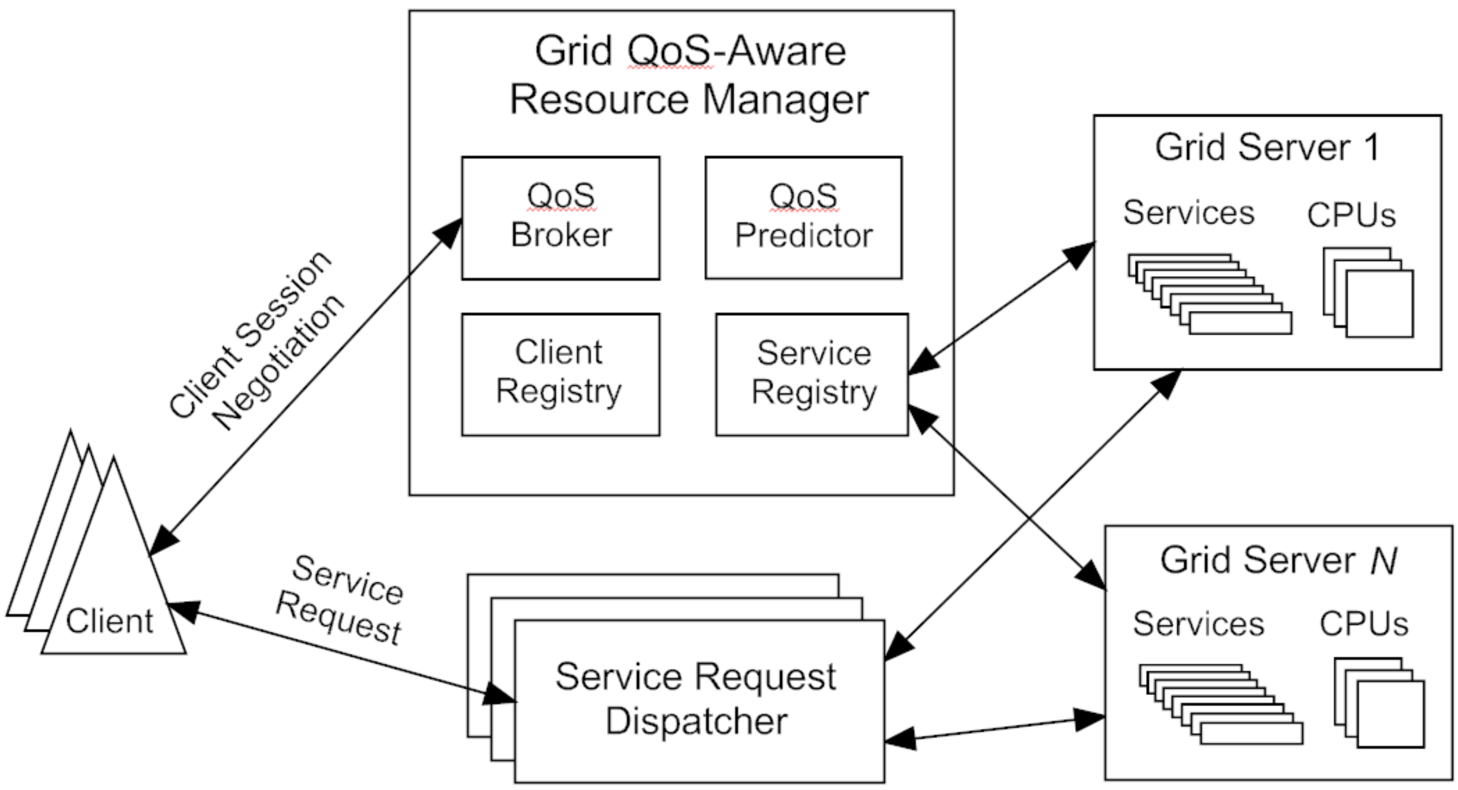
Case Study: Automated Model Extraction (2)

- Model A: Resource demands approximated with measured response times
- Model B: Resource demands estimated based on utilization and throughput data



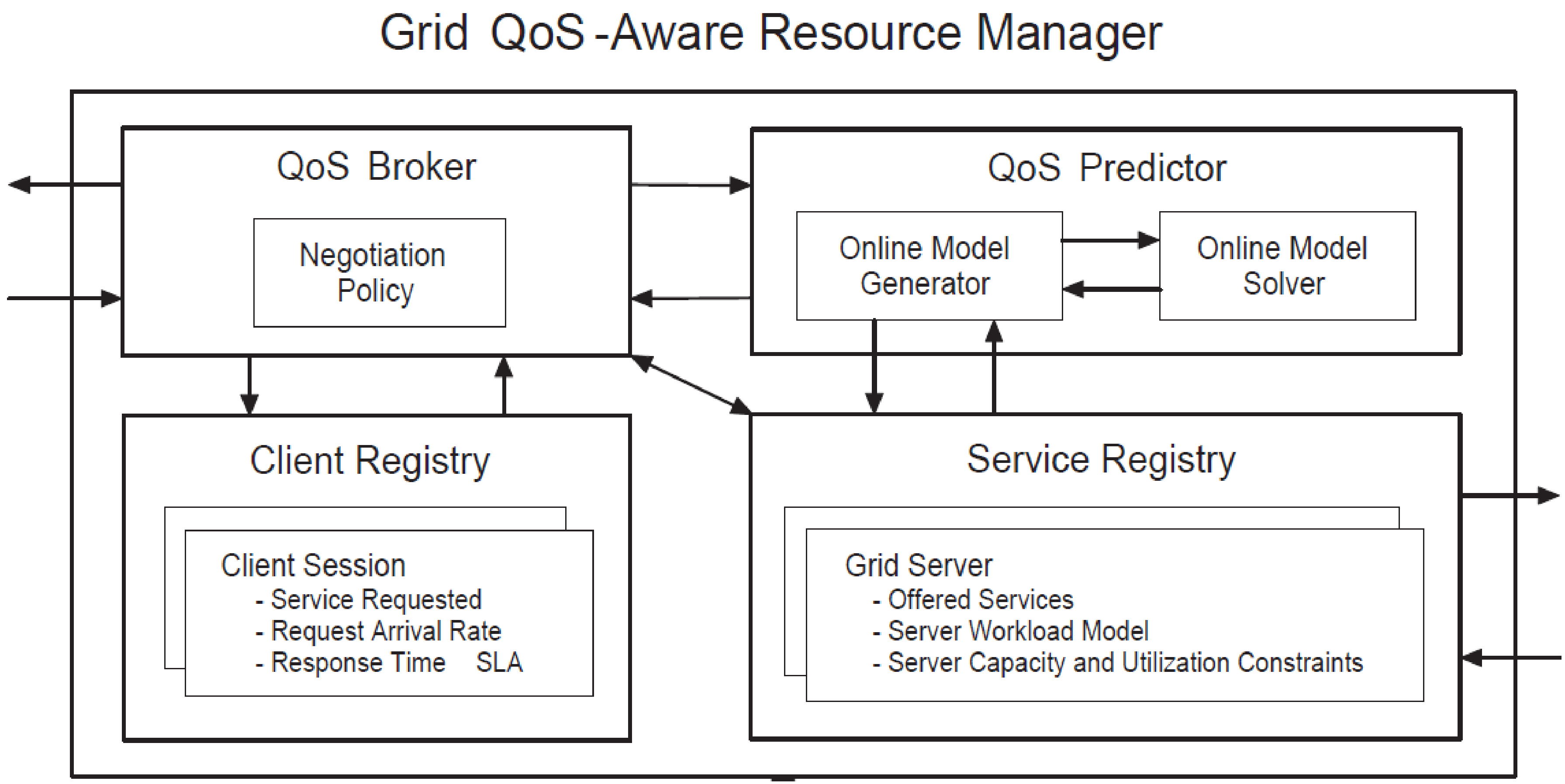
Model A: $U_{WLS\_CPU} = 0.12$, Model B: $U_{WLS\_CPU} = 0.81$, Steady State Time: 1020 sec

Case Study: Online QoS Control



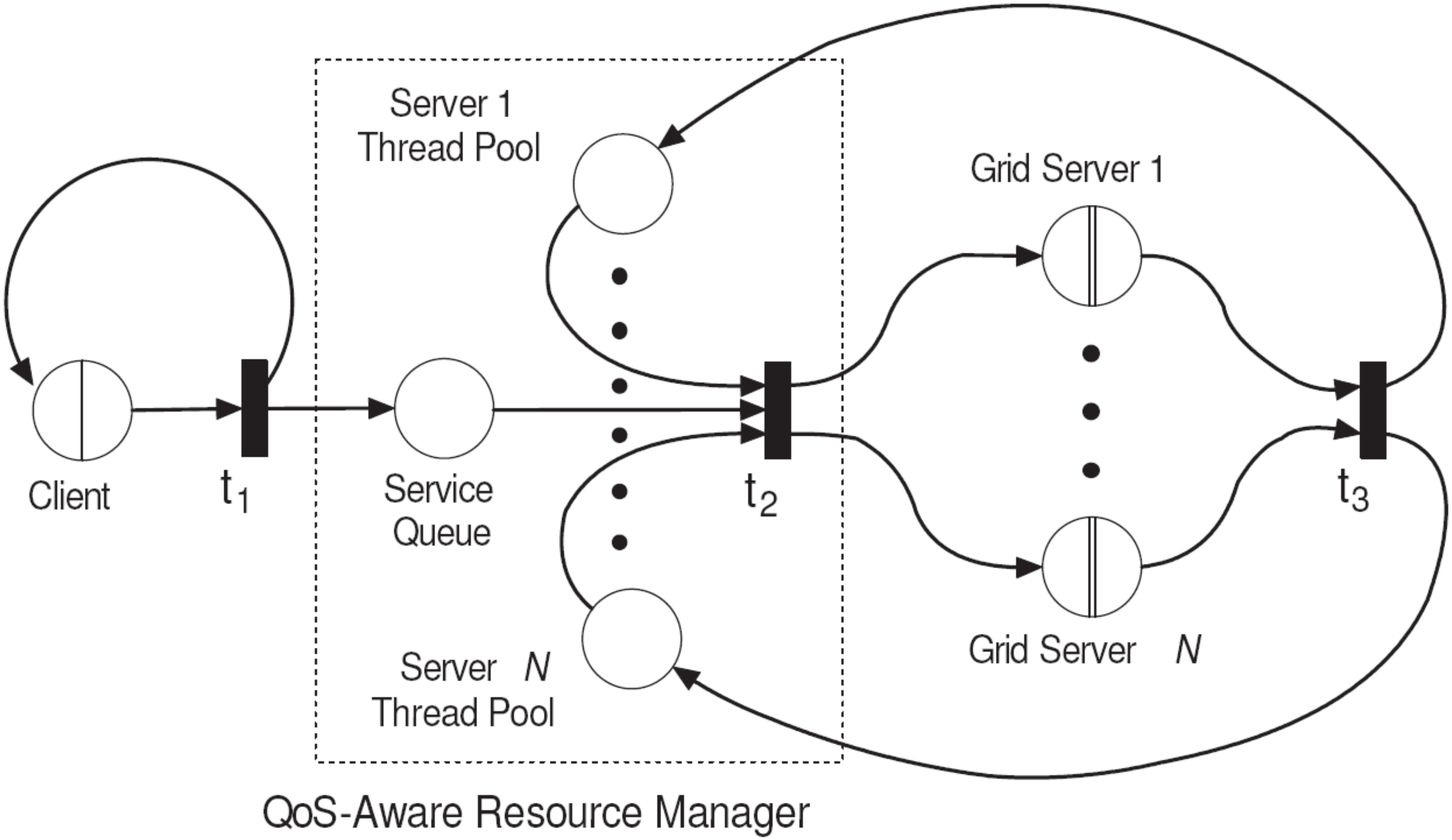
R. Nou, S. Kounev, F. Julia, and J. Torres. „**Autonomic QoS control in enterprise Grid environments using online simulation**". *Journal of Systems and Software*, 82(3):486-502, March 2009.

Case Study: Online QoS Control

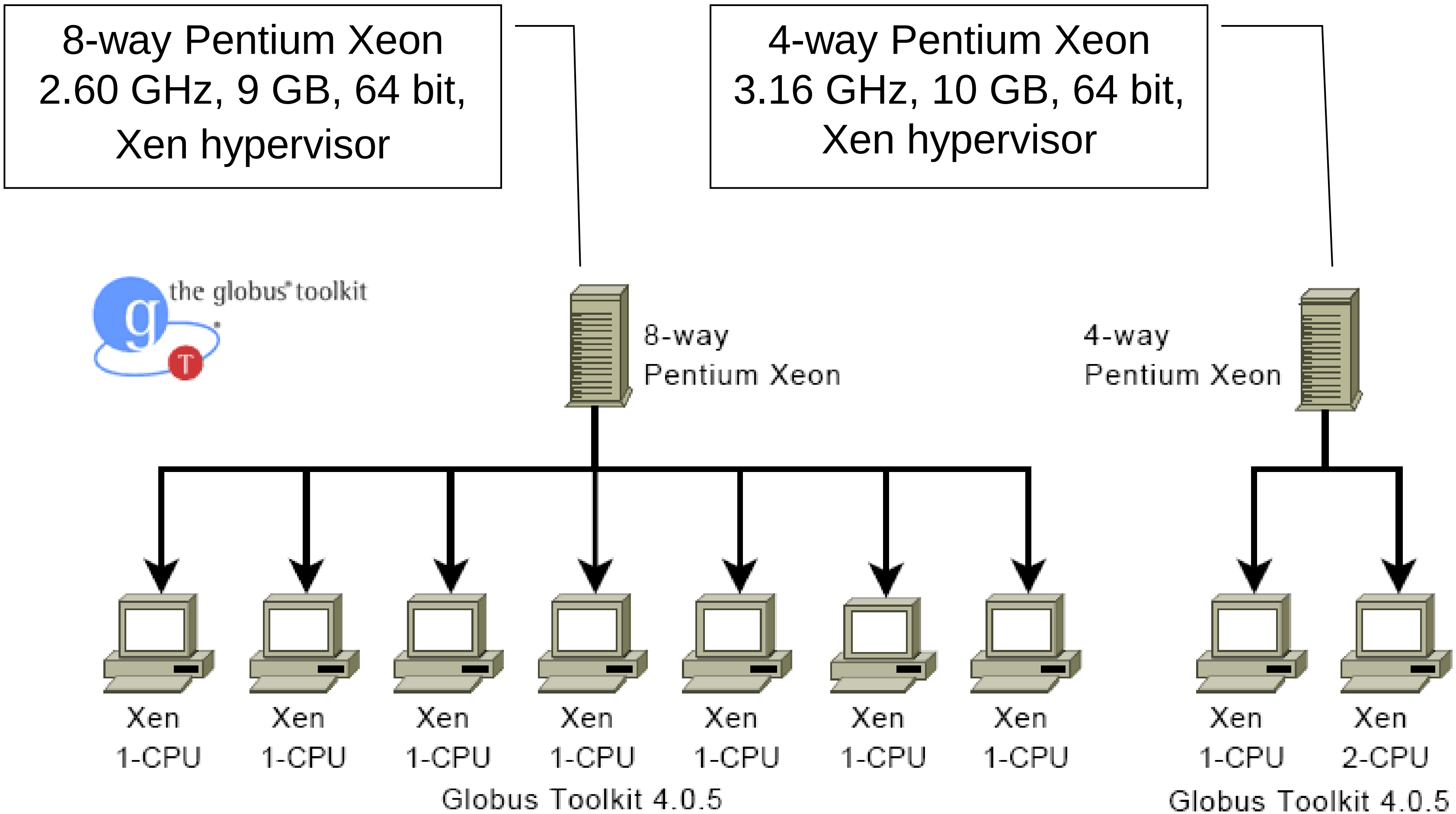


R. Nou, S. Kounev, F. Julia, and J. Torres. „**Autonomic QoS control in enterprise Grid environments using online simulation**". *Journal of Systems and Software*, 82(3):486-502, March 2009.

Case Study: Online QoS Control (cont.)



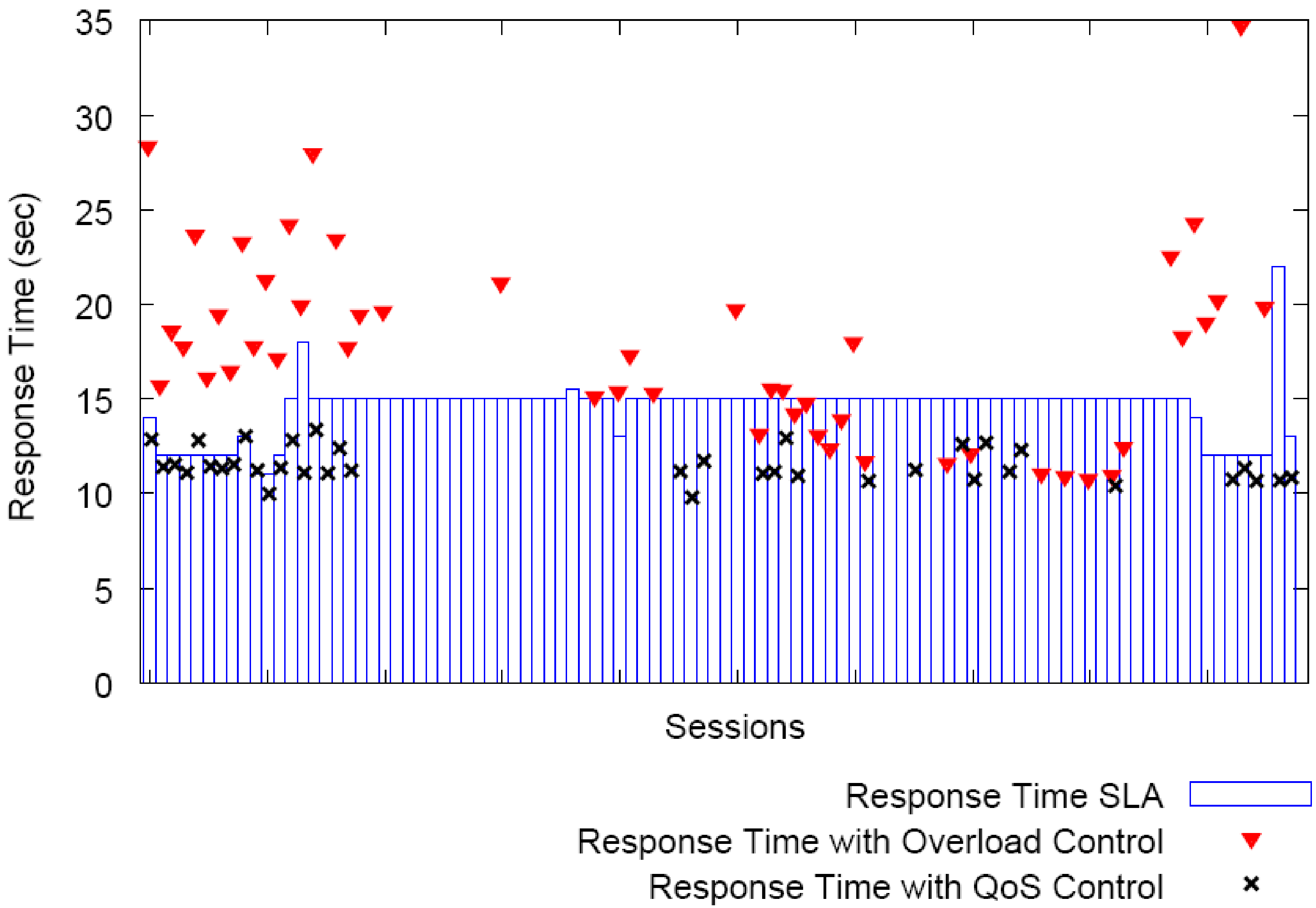
Case Study: Online QoS Control (cont.)



Case Study: Online QoS Control (cont.)

- 99 session requests executed over period of 2 hours
- Run until all sessions complete
- Average session duration 18 minutes (92 requests)
- Will compare two configurations
  - **Without QoS Control**
    - Incoming requests simply load-balanced
    - Reject session requests when servers saturated
  - **With QoS Control**
    - QoS-aware admission control enforced

Case Study: Online QoS Control (cont.)



Case Study: Online QoS Control (cont.)

- Config 1: Without QoS Control
  - 96% of sessions admitted, SLAs observed by only 22% of them
- Config 2: QoS Control / workload model available
  - 54% of sessions accepted
- Config 3: QoS Control / workload characterization on-the-fly
  - Rejects only 14 sessions (16%) more compared to Config 2

Config	SLA fulfilled	SLA violated	Sessions rejected
1	19	63	3
2	46	2	37
3	34	0	51

Case Study: Online QoS Control (cont.)

- Config 1: All servers online / without QoS control.
- Config 2: All servers online / with QoS control.
- Config 3: Servers added on demand / without QoS control.
- Config 4: Servers added on demand / with QoS control.

Configuration	SLA fulfilled	SLA violated	Sessions rejected
1	19	63	3
2	46	2	37
3	15	61	9
4	45	7	33

Case Study: Online QoS Control (cont.)

- Up to five server failures emulated during the run
- Points of server failures chosen randomly during the 2 hours
- Sessions reconfigured after each server failure

Failures emulated	Without QoS Control			With QoS Control		
	SLA fulfilled	SLA violated	Sessions rejected	SLA fulfilled	SLA violated	Sessions rejected
1	14	62	9	37	1	47 (0)
2	16	57	12	39	3	43 (1)
3	10	58	17	40	3	42 (2)
4	3	56	26	38	1	46 (6)
5	4	45	36	31	4	50 (13)

Summary

- CC promises to revolutionize the way software is built and run
- QoS issues and lack of trust → major show stoppers
- **Self-Aware Software Systems**
  - Systems with integrated dynamic prediction models
  - Models composed dynamically at run-time
  - Used for autonomic QoS management
- **Major challenges**
  - Platform-independent meta-model for dynamic software systems
  - Trade-off accuracy vs. management overhead
  - Bridging the gap between service provider and infrastructure provider

Thank You!

www.descartes-research.net

Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft
Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825