Department of Telecooperation

AWPS - an architecture for pro-active web performance management

Univ.-Prof. Dr. Gabriele Kotsis (Gabriele.Kotsis@jku.at)

Martin Pinzger (Mpinzger@gmail.com)

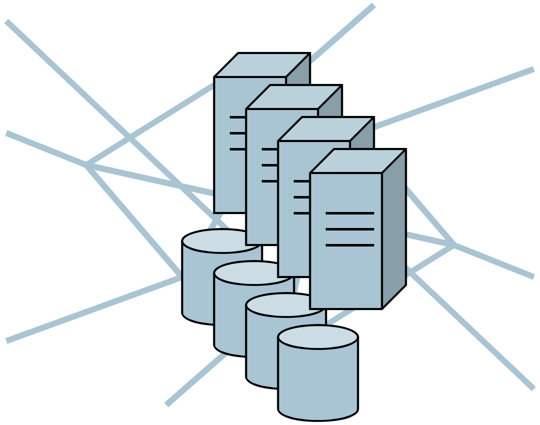
Department of Telecooperation

Johannes Kepler University Linz Austria

Motivation

user requests

user perceived performance (delay, response time)



system performance (utilisation, throughput)

Performance = F (Workload, System)

What is the effect on performance if workload is subject to change?

What is the effect on performance if the system is subject to change? (consider also interdependencies with changes in the workload)

How to design the system to deliver a certain performance for a given workload?

Approaches

- Performance Measurements
  - On real world systems
  - On artificial systems
  - Example: DynaTrace (JKU Spin Off)
    - Measurement and analysis tools for web server performance based on the concept ofexecution paths
- Performance Modelling
  - Off line versus online

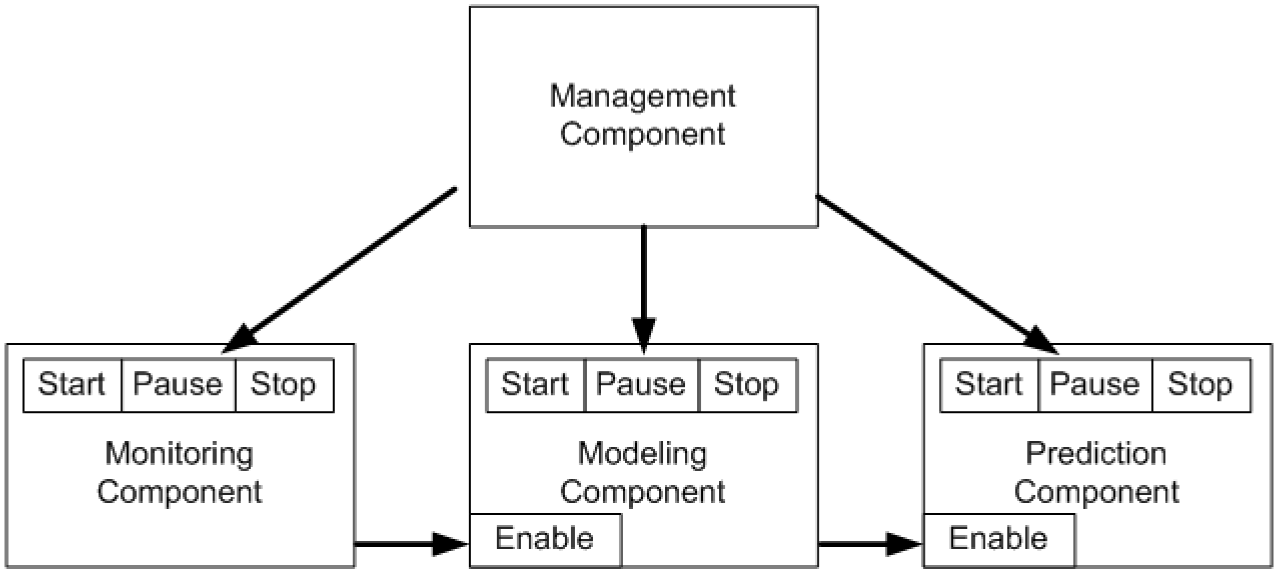
Automated Web Performance System

- **AWPS Concept**
- AWPS Environment Interaction
- **Case Study**
- Conclusion

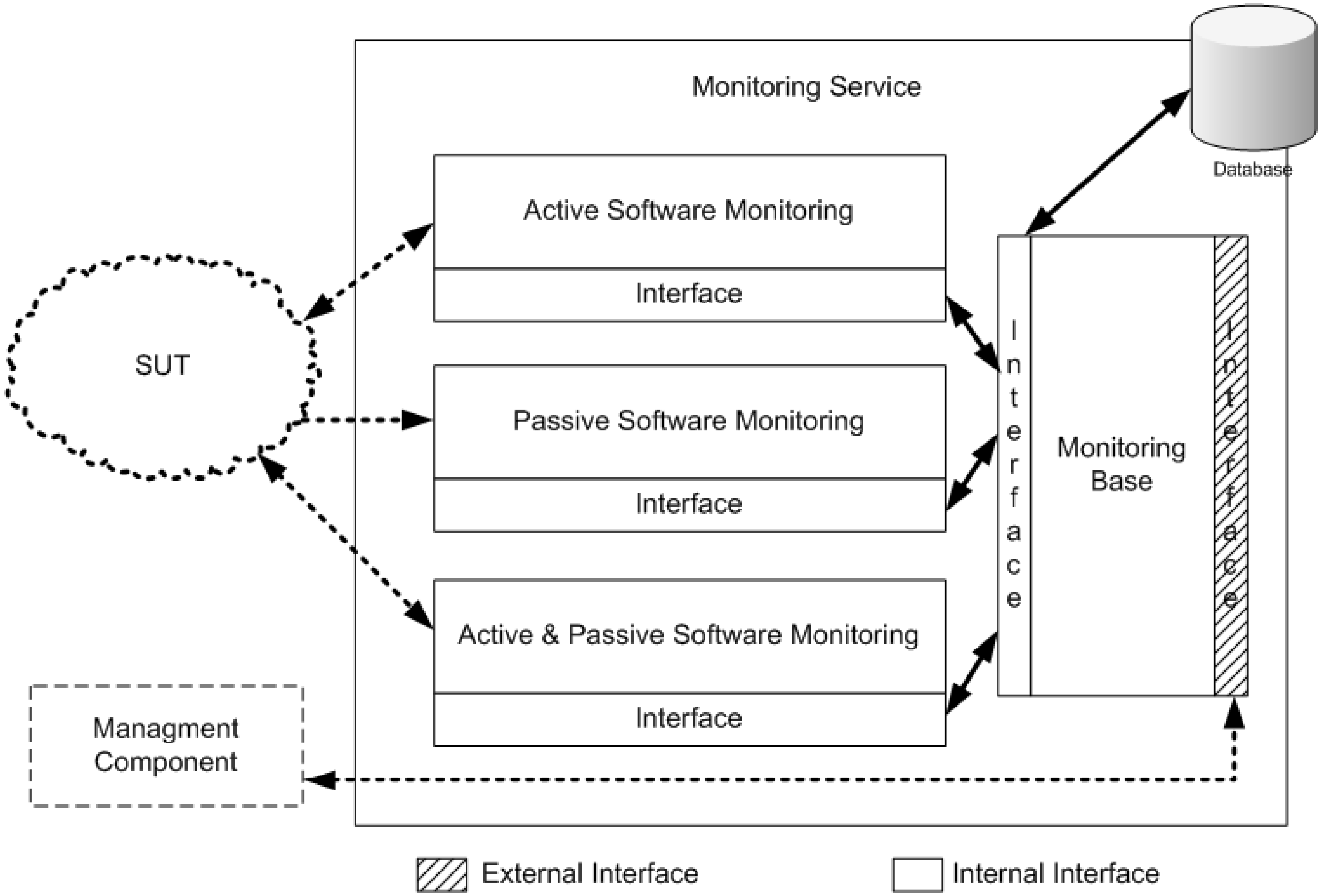
AWPS Concept

- Key Characteristics
  - Automatic
  - Online / „*Realtime*“
  - Pro-active

- Three Key Functions
  - Data Collection
  - Simulation
  - Prediction



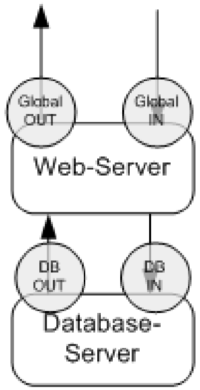
Data Collection - Monitoring



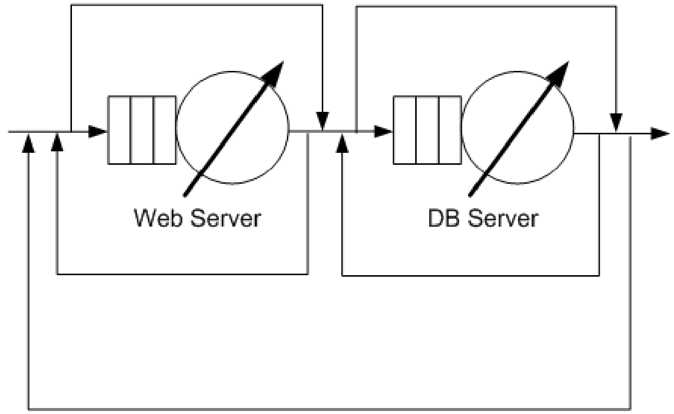
AWPS Concept - Simulation Component

- Model Generation Component
  - Minimum complexity simulation model
  - Maximum complexity simulation model
- Model Comparison Component
- Model Adjustment Component
  - AVG Strategy, Median Strategy, ARMA Strategy
- Model Simulation Component
  - JSIM

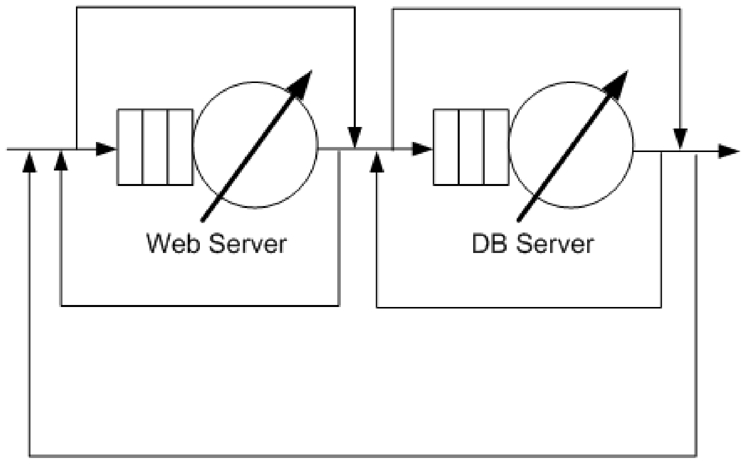
Modeling / Simulation Task



Generation Adaption



Adjustment

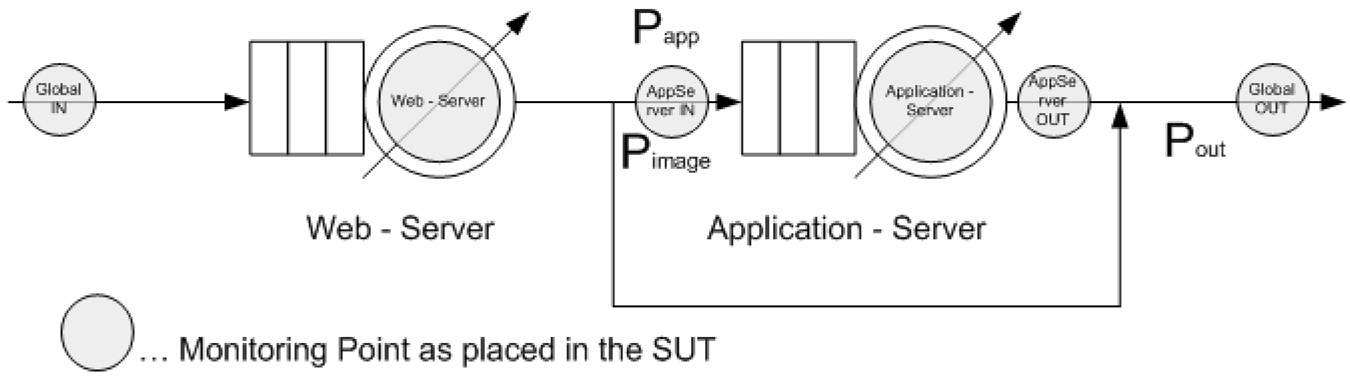


Execution

Comparison Loop

AWPS Concept - Simulation Component

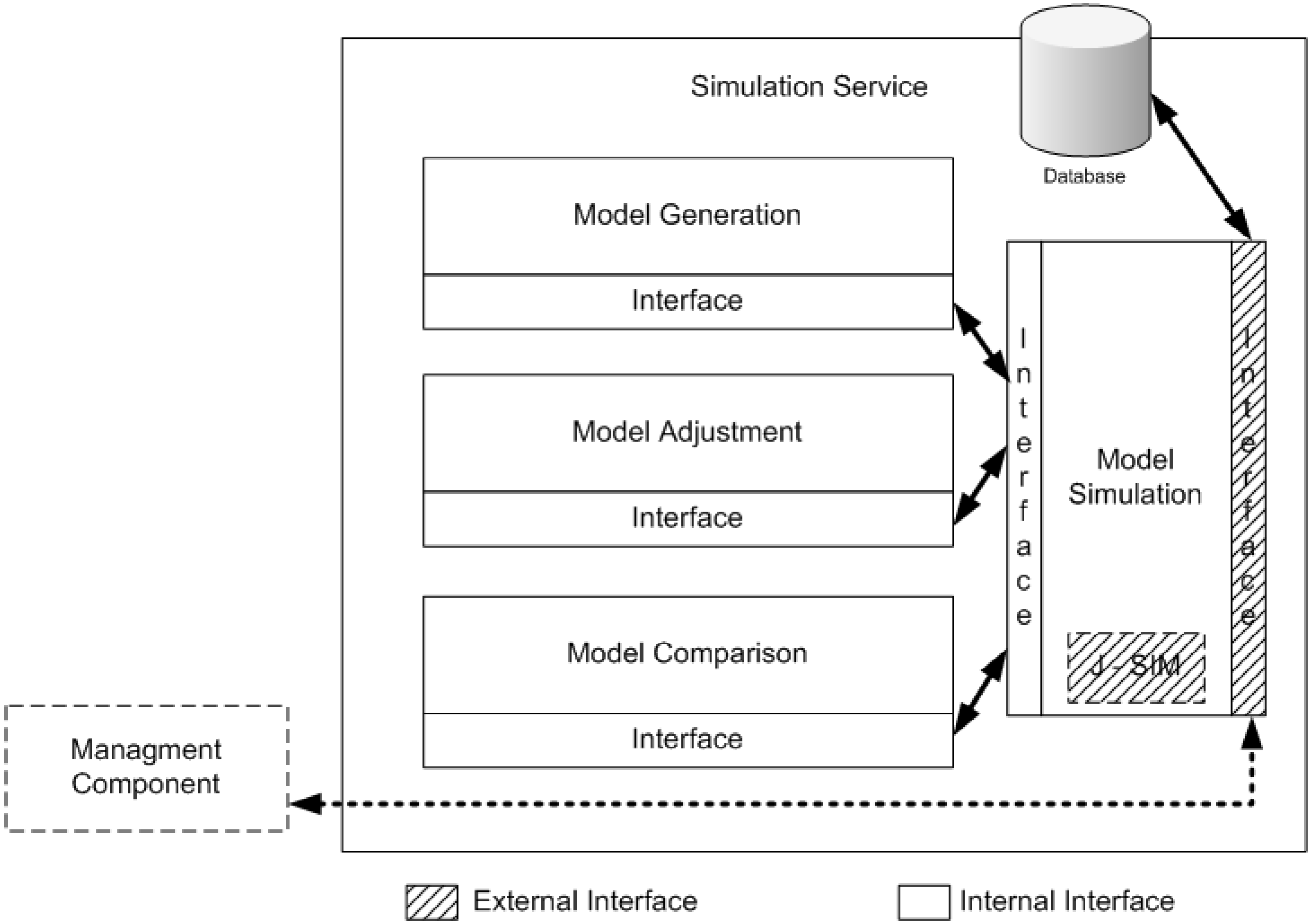
Model Generation Component - Example



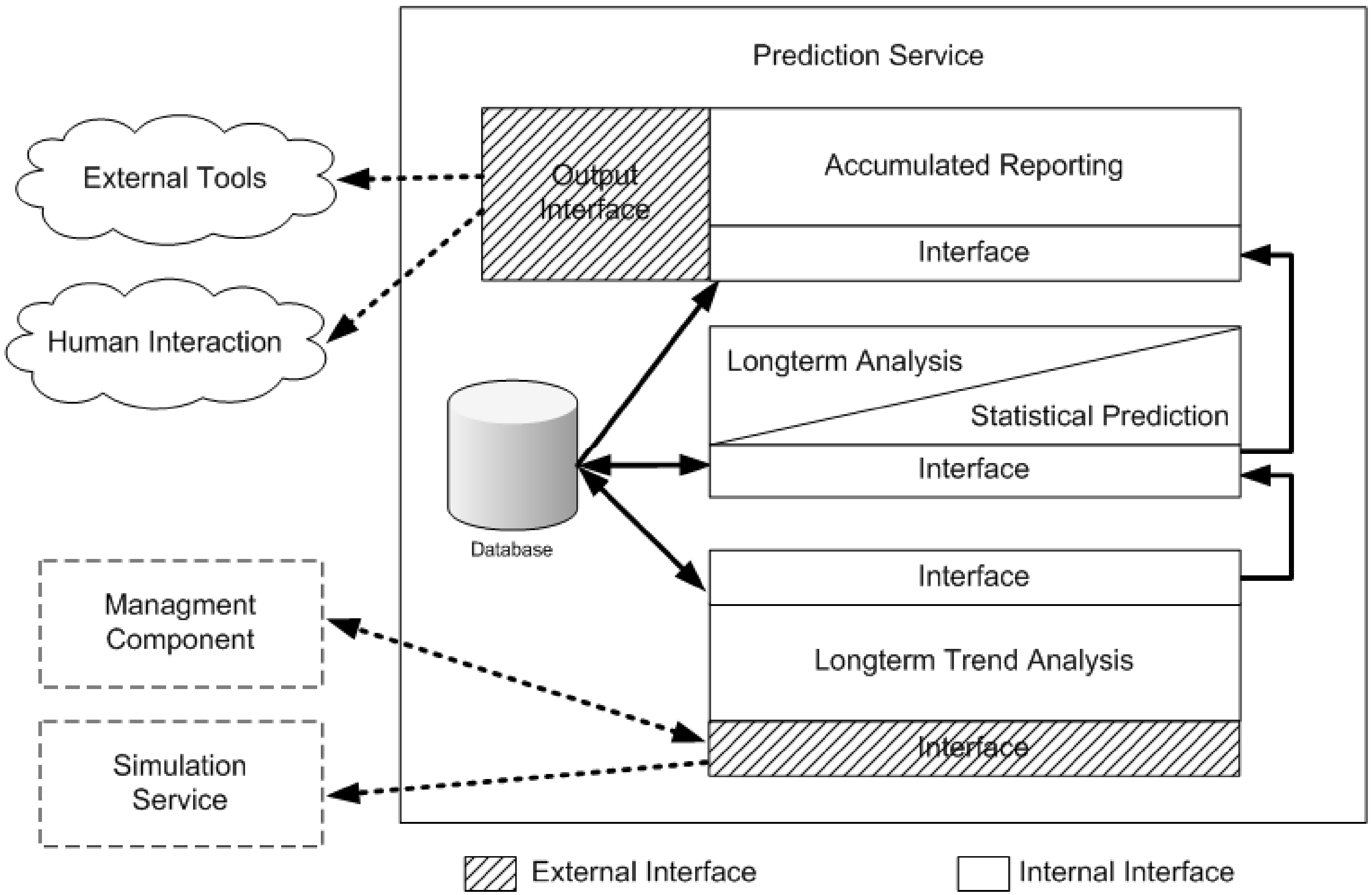
P_{app} … for *special offers* **1.0** for *image requests* **0.0**
P_{image} … for *special offers* **0.0** for *images requests* **1.0**

- **TotalSystemTime** = GlobalOut - GlobalIn
- WebServerTimeA = AppServerIn - GlobalIn
- **AppServerTime** = AppServerOut - AppServerIn
- WebServerTimeB = GlobalOut - AppServerOut
- **WebServerTime** = WebServerTimeA +WebServerTimeB

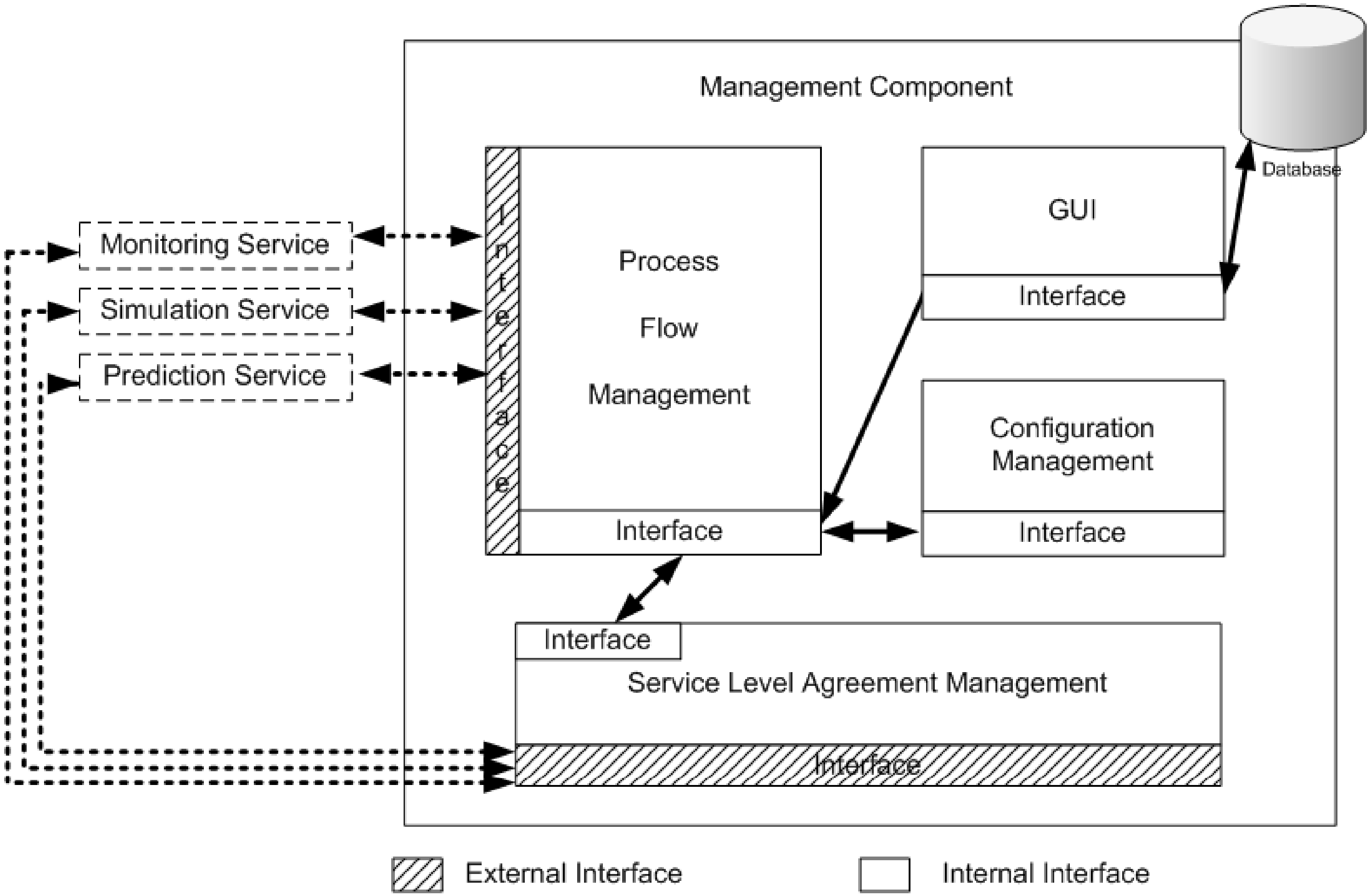
Modeling / Simulation Component



Prediction Component



Management Task



AWPS Environment Interaction

- System Setup
  - Passive Monitoring Strategy *(Network Sniffing)*
  - A Multi Class / Single Queue / Multi Server Model is created automatically
- User Interaction
  - Main Configuration Site
  - Result Presentation Site
  - Online Observation Site
- Influence on the Productive System - **Minor**

AWPS - System Ver. 1.0

EXECUTION RESULTS PREDICTION STATUS SETUP ABOUT CONTACT

If all else fails, read the instructions. LaTeX error help message

Welcome to the AWPS - System

The AWPS - System and it´s base components are described in ASE08 Doctoral Symposium Paper:"Automated Web Performance Analysis"

Abstract

Performance is a key feature in many systems nowadays. There are several tools on the market that ensure and test for adequate performance. They, can be divided into simulation tools and monitoring tools. But only a few automatise and combine both approaches.This paper describes a system capable of automatically creating a web performance simulation and conducting trend analysis of the system under test (SUT). To achieve this the system requires input information, like Monitoring Points and Static-Information about the SUT. The system monitors and analyses the SUT and based on this information generates a simulation model of the system. The simulation model is refinded stepwise e.g. by adding or removing connections between the model components or adjusting the parameters until the aimed accuracy is achieved. With the help of the simulation model a prediction module creates an analysis of the SUT, and thereby can give as much information about the current state of the system and potential trends as possible. This predictive information can be used for pro-active server tuning or other performance optimisations. The focus of my PhD thesis is on the adjustment and prediction part of the system described here. For all other parts, already existing tools and techniques will be used where possible. This initial paper outlines the complete system.

Execution

Settings

Configuration for the basic options concerning the AWPS - Analysis - Tool:

Select Id of Test Run to use:

(-1) New Test Run (ONLINE)

Options only active when mode "**(-1) New Test Run (ONLINE) or (-2) New Test Run (Simulated SUT)**" is selected:

Title of Test Run: TK-Web-Site

Description of Test Run: Online Analysis

Support Multiple Server: ☑

Support Multiple Classes: * e.g. gif,php,css,...

Select maximum number of Request to use during simulation: 50000

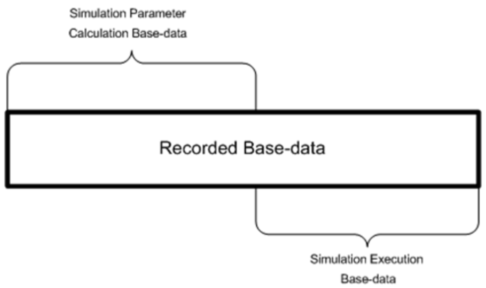
Select after how many Requests a Simulation should be executed: 10

Select method to use for value determination for the simulation parameters:

- ○ AVG - Calculation based on all recorded requests
- ○ AVG - Calculation based on the selected max requests
- ○ Binary - Search
- ◉ ARMA - Calculation based on all recorded requests NO Grouping
- ○ ARMA - Calculation based on all recorded requests WITH Grouping in 60 Seconds Windows
- ○ Median 50 percent

Select Simulation Strategie:

- ◉ TypeD



Number of Requests for Simulation TypeB and TypeD shift: 10

Analysis - Description

Analysis Name: 3.2010 - 06:13 - TK - Web

Analysis Description: Online Analysis - Web

Execute AWPS - System | Clear Form

AWPS - System Ver. 1.0

HOME

Results of the Analysis

Overview

In the following tabel you see the basic results of the executed test runs.

Filter by IdConfRun:

(1739)PERIODIC SPIKE INKL RAND ARMA Step Size 10 (Inkl 10 Threads)

Id	IdConfRun	IdTestRun	SimExCounter	ServerIdentification	Category	AVG-Sim (Sec.)	AVG-Calc-All (Sec.)	AVG-Calc-Sim (Sec.)	AVG-Queue-Length	Nr.Requests
2039427	1739	192	1	MutliServer-All	gif	0.0	0.50015	0.4967	0.0	20
2039428	1739	192	2	MutliServer-All	gif	0.0	0.49996666666666667	0.4996	0.0	30
2039429	1739	192	3	MutliServer-All	gif	0.0	0.50305	0.5123	0.0	40
2039430	1739	192	4	MutliServer-0	gif	0.586880995401449	0.50052	0.4904	0.022595695450916813	50
2039431	1739	192	4	MutliServer-All	gif	0.586880995401449	0.50052	0.4904	0.022595695450916813	50
2039432	1739	192	5	MutliServer-0	gif	0.5584574643211779	0.4992833333333333	0.4931	0.012814381594510106	60
2039433	1739	192	5	MutliServer-All	gif	0.5584574643211779	0.4992833333333333	0.4931	0.012814381594510106	60
2039434	1739	192	6	MutliServer-0	gif	0.5731647177150283	0.4986285714285714	0.4947	0.014249411680278276	70
2039435	1739	192	6	MutliServer-All	gif	0.5731647177150283	0.4986285714285714	0.4947	0.014249411680278276	70
2039436	1739	192	7	MutliServer-0	gif	0.5703540344890223	0.49845	0.4972	0.012140606510588065	80
2039437	1739	192	7	MutliServer-All	gif	0.5703540344890223	0.49845	0.4972	0.012140606510588065	80
2039438	1739	192	8	MutliServer-0	gif	0.497580368145471	0.4971333333333333	0.4866	0.0	90
2039439	1739	192	8	MutliServer-All	gif	0.497580368145471	0.4971333333333333	0.4866	0.0	90
2039440	1739	192	9	MutliServer-0	gif	0.5456886130436212	0.49752	0.501	0.006817794779505936	100
2039441	1739	192	9	MutliServer-All	gif	0.5456886130436212	0.49752	0.501	0.006817794779505936	100
2039442	1739	192	10	MutliServer-0	gif	0.5540319738558908	0.4989818181818182	0.5136	0.006911064592359537	110
2039443	1739	192	10	MutliServer-All	gif	0.5540319738558908	0.4989818181818182	0.5136	0.006911064592359537	110
2039444	1739	192	11	MutliServer-0	gif	0.5450089483222165	0.4999166666666667	0.5102	0.004841764168645552	120

Detail Analysis of the Reqeuest IdConfRun:1739

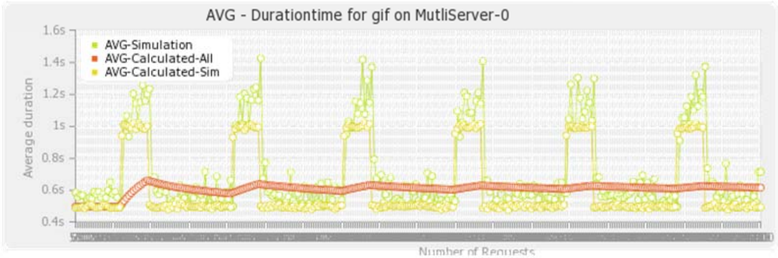
Settings

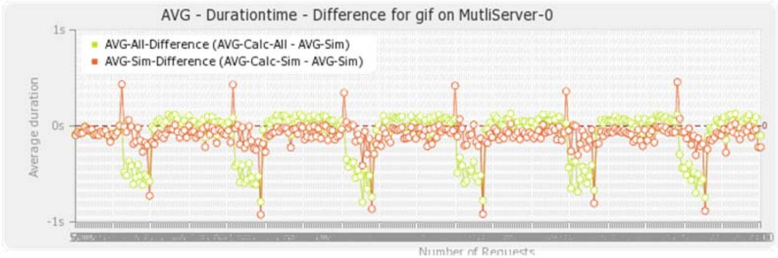
1739;PERIODIC SPIKE INKL RAND ARMA Step Size 10 (Inkl 10

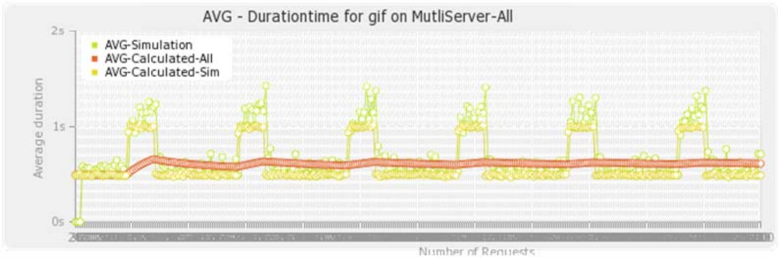
Threads);GET_0,HTTP_0,gif;192;50000;4;10;GET_0;HTTP_0;gif;4;10;1281477475070;f;10;t;t;

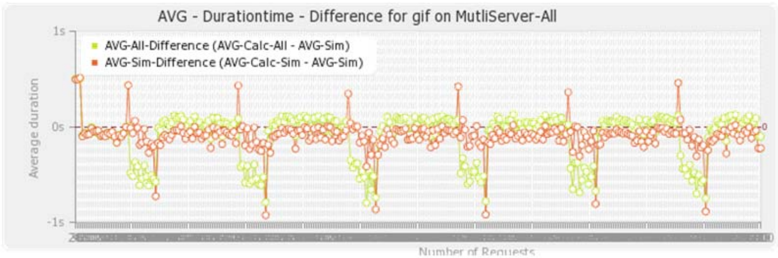
Diagram

In the following diagram you see the basic results of the executed test run, in comparison.









Department of Telecooperation

AWPS - System Ver. 1.0

HOME

System Status

Environment:

- Web Server

  OK

- PostgreSQL Database Server 8.3

  OK

- Socket Communication Service

  OK

Select Conf Run to Monitor Online:

(1740)PERIODIC SPIKE INKL RAND ARMA Step Size 1 (Inkl 10 Threads)

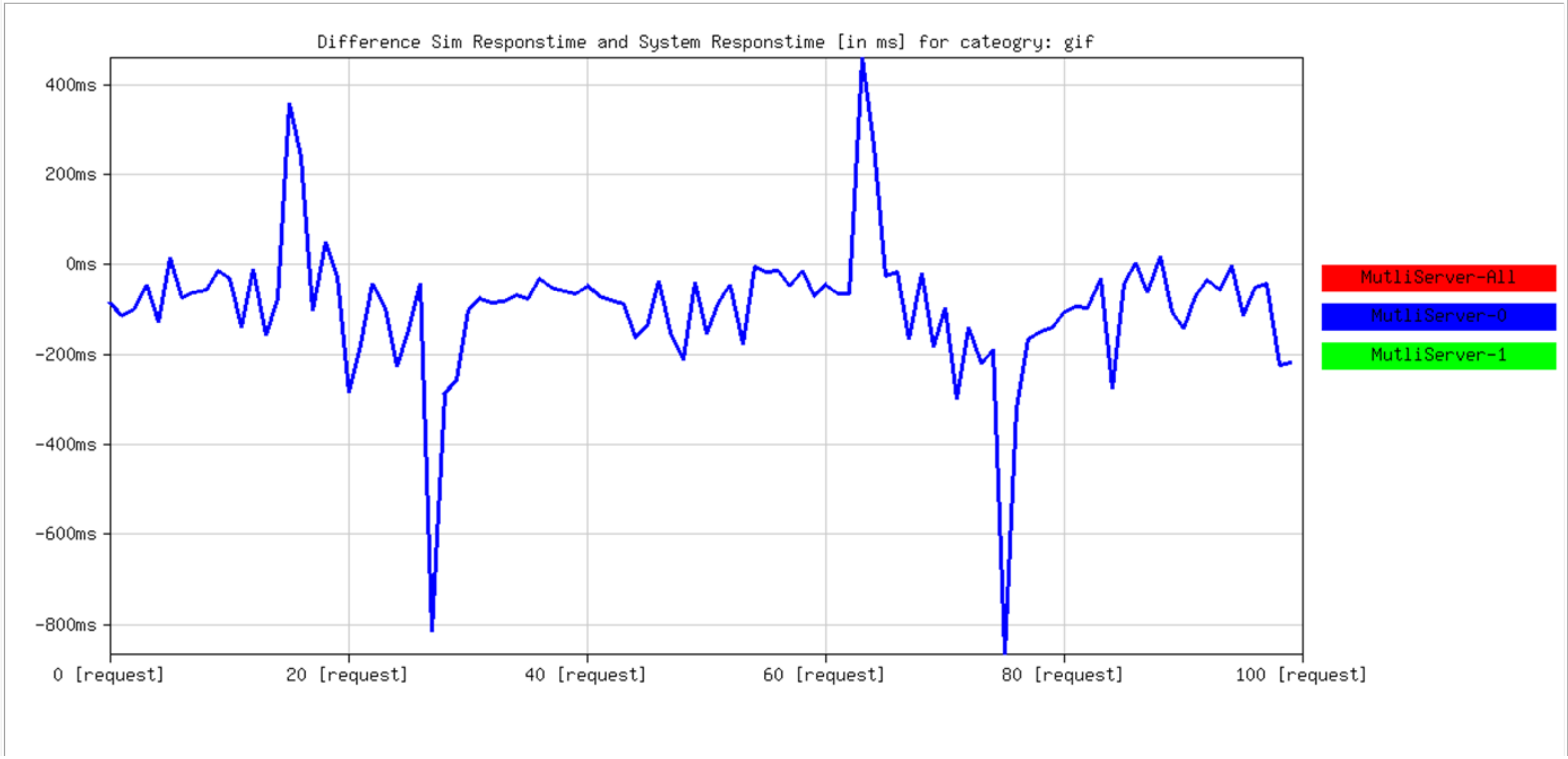
Online - Monitor

Department of Telecooperation

A W P S - S y s t e m V e r . 1 . 0

HOME

System Status ONLINE - Monitor

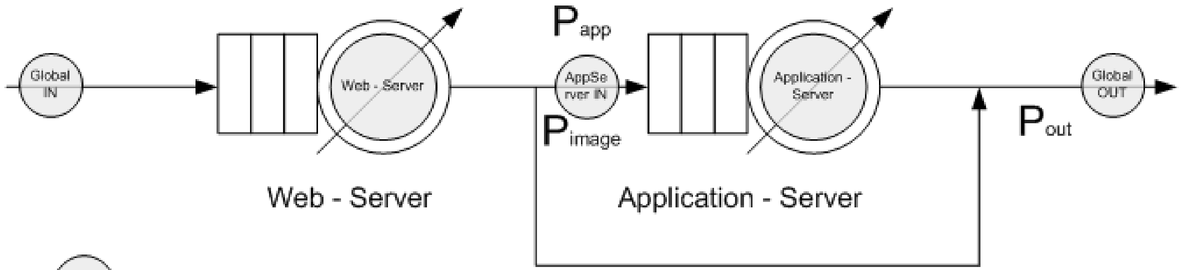


Analysed System Categories

- **Feasibility of the Approach**
  - Synthetic System Offline
    e.g. strictly increasing, constant, complex function
- **Real-time Feasibility**
  - Synthetic System Online
    e.g. by control invoice concerning calculation time consumption
- **Representative Test**
  - Productive System Offline
    TK-WebSite offline analysis (normal load / synthetic load),
    GoSpace offline analysis (synthetic load)
- **Representative Realtime Test**
  - Productive System Online / **Field Test**

Case Study

The case study was done on a two tier web application, which provides as functionality a web page where you can search and book space flights.



… Monitoring Point as placed in the SUT

P_{app} … for *special offers* **1.0** for *image requests* **0.0**

P_{image} … for *special offers* **0.0** for *images requests* **1.0**

TotalSystemTime = GlobalOut - GlobalIn

WebServerTime = AppServerIn - GlobalIn

AppServerTime = GlobalOut - AppServerIn

Case Study – Results - Overview

- Analysis of difference between simulation and reference data
  - T-Test
  - Mean Error / Variation
- Correlation between accuracy and simulation runs
  - Step Size 1000 (Median, AVG, ARMA, ARMA G.)
  - Step Size 100 (Median, AVG, ARMA)
- Realtime / Online Capability

Case Study – Results – T-Test

Strategy	Step Size 1000		Step Size 100		Step Size 10	
	image	special	image	special	image	special
AVG	Yes.	**No (0.980230)**	Yes	Yes (0.935309)	Yes	Yes (0.789822)
Median	Yes.	Yes (0.758424)	Yes	Yes (0.335950)	Yes	Yes (0.002655)
ARMA	Yes.	**No (0.982876)**	Yes	**No (0.990521)**	Yes	**No (0.998247)**
ARMA G.	Yes.	**No (0,992849)**	Yes	Yes (0.289593)	Yes	Yes (0.000000)

Significant difference between simulation and reference data.
The value in the brackets represents the double sided t-Test value.

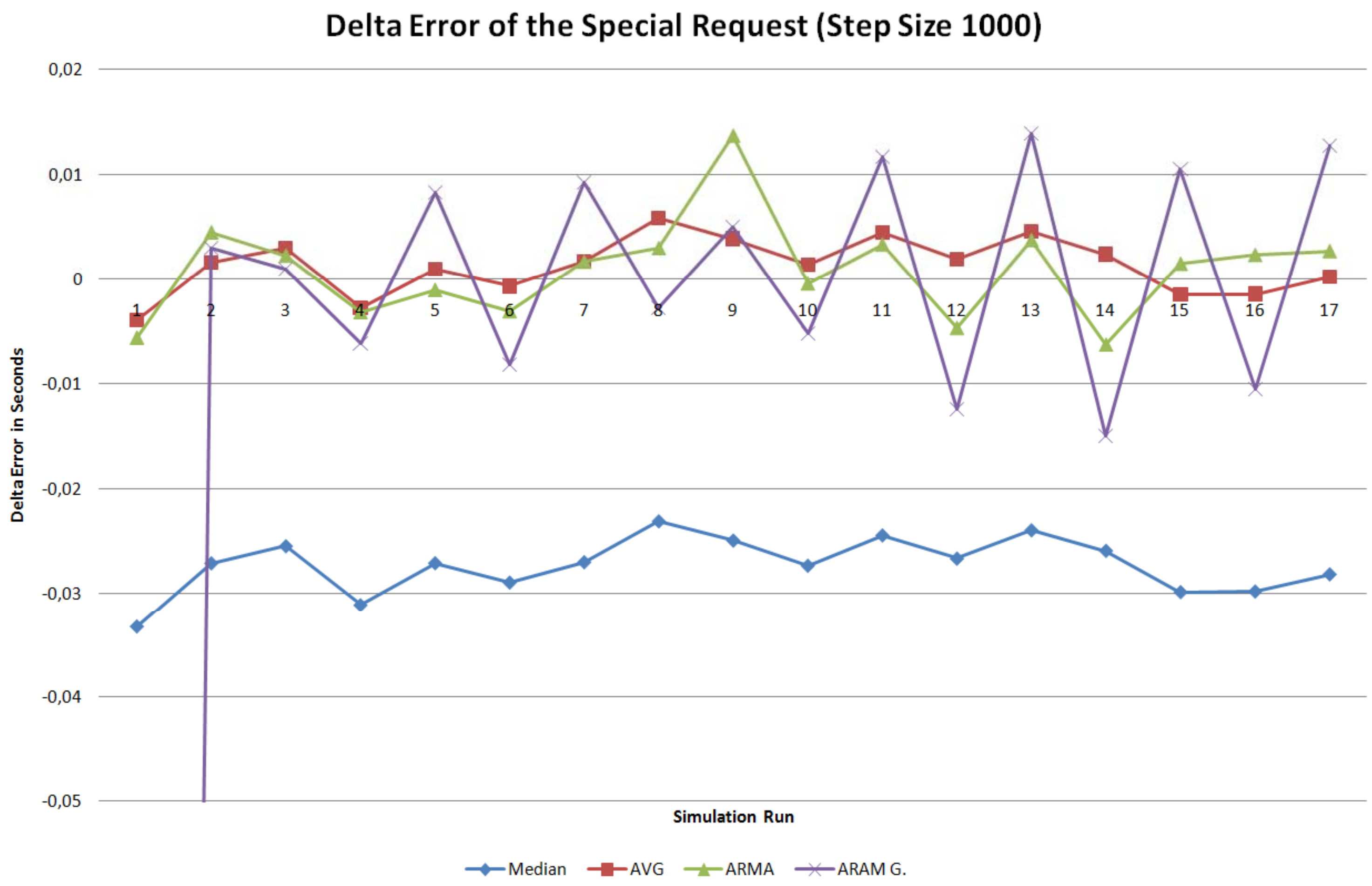
Case Study – Results – Observed Error

Strategy	Step Size 1000		Step Size 100		Step Size 10	
	mean	variance	median	variance	median	variance
AVG	0.002228	0.000002	0.005419	0.000013	0.023877	0.000514
Median	0.017624	0.000052	0.017543	0.000084	0.020431	0.001013
ARMA	0.005091	0.000012	0.006319	0.000022	0.023590	0.000568
ARMA G.	0.006824	0.000019	0.088866	0.055627	0.230810	0.060259

Mean values and Variance values in seconds referring to the delta error, for the special offers (do?action=special) request class.

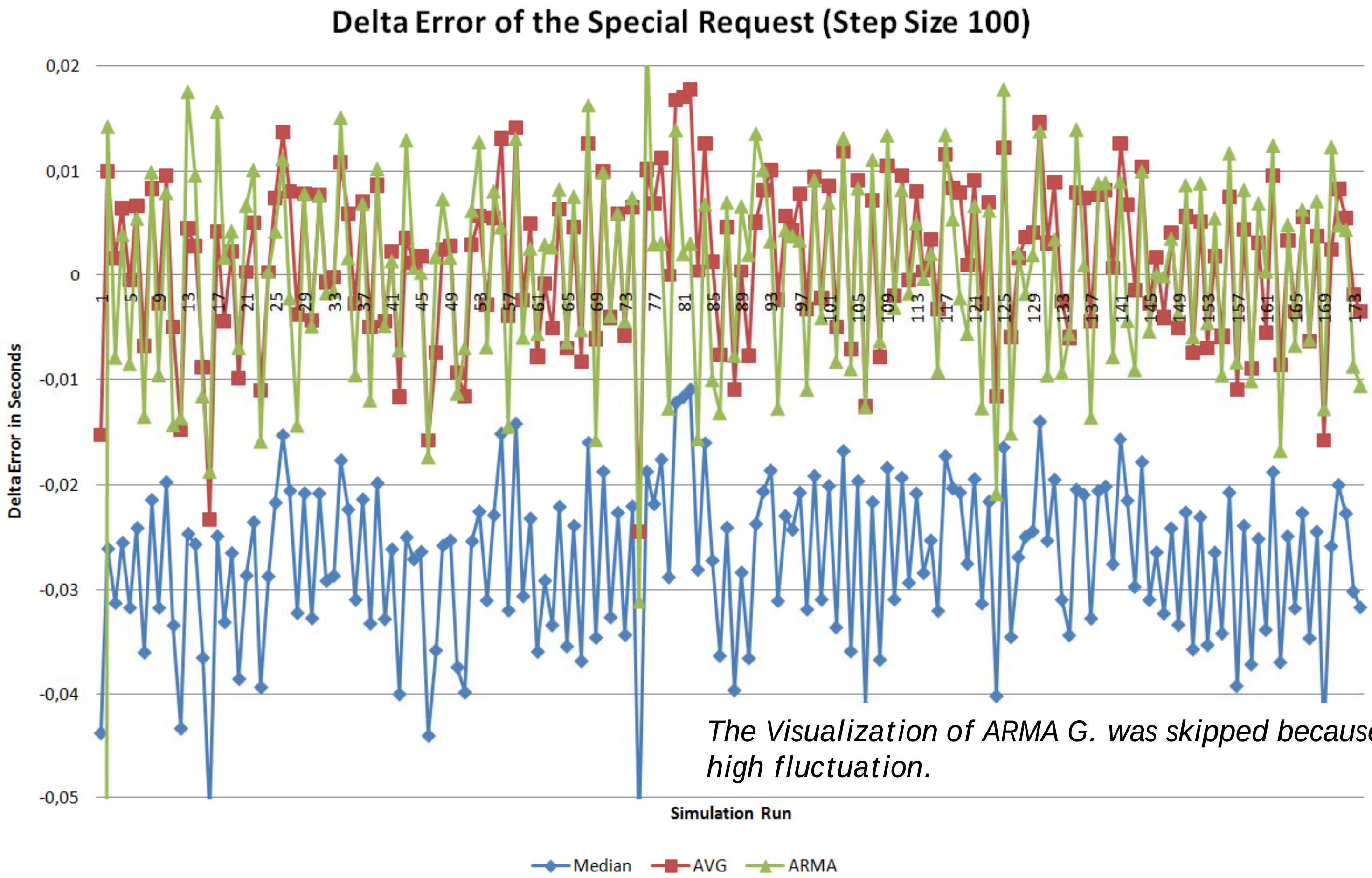
Case Study – Results – Observed Error

Correlation to the Simulation Runs



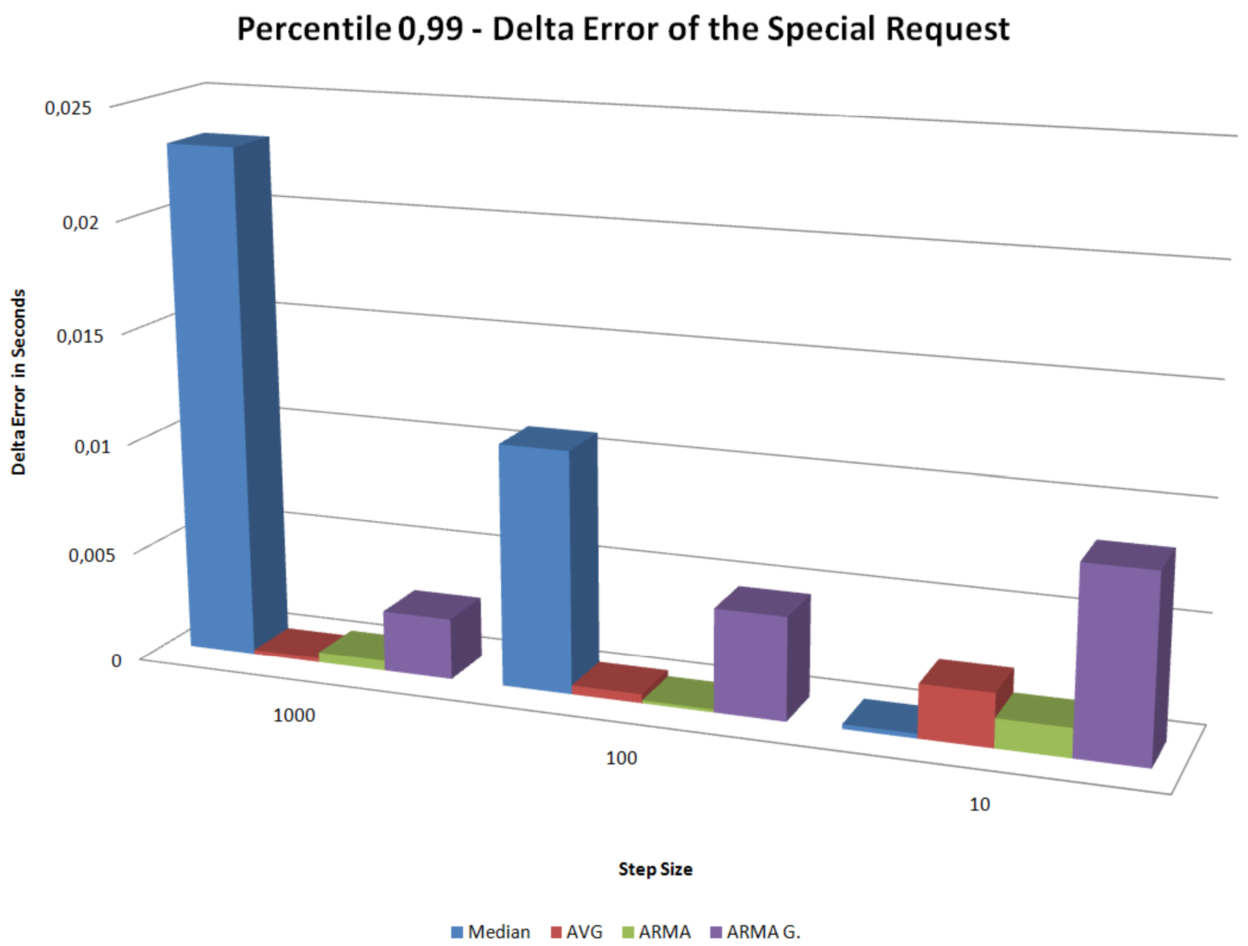
Case Study – Results – Observed Error

Correlation to the Simulation Runs



The Visualization of ARMA G. was skipped because of the high fluctuation.

Case Study – Results – Percentile 0,99



Case Study – Results – Realtime

Strategy	Step Size 1000	Step Size 100	Step Size 10
AVG	244 sec.	511 sec.	2659 sec.
Median	489 sec.	541 sec.	3243 sec.
ARMA	339 sec.	945 sec.	**6706 sec.**
ARMA G.	288 sec.	827 sec.	2947 sec.

Calculation time consumption. Values below 3600 sec. mean that the adjustment method is online-capable.

Conclusion

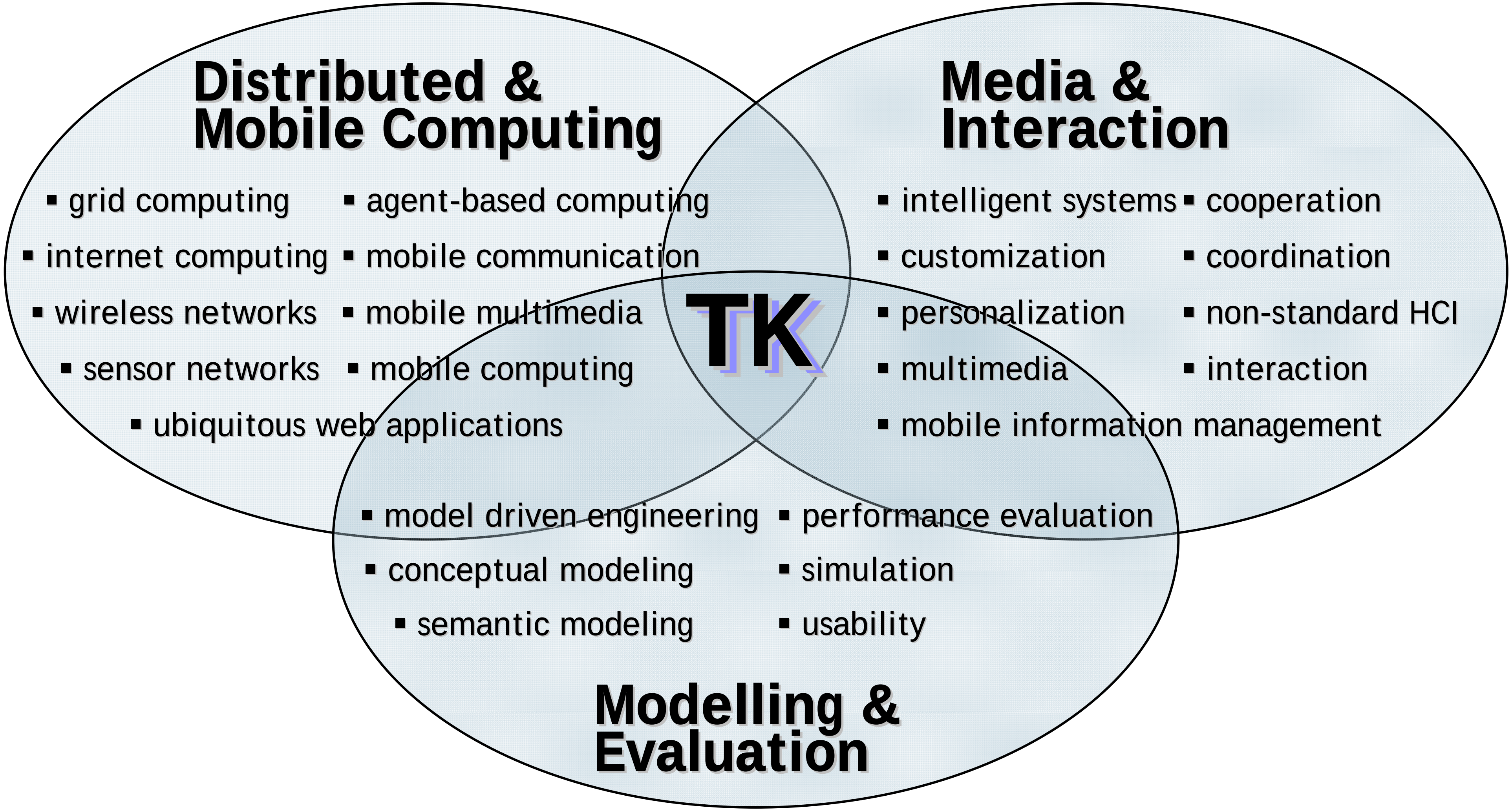
- AWPS works as expected and provides representative results
- simulation model generation process works autonomously and is sufficiently fault-tolerant
- strategies for the adjustment of the simulation model work accurately

- functionality should be enhanced *e.g. adaptive scenario generation*
- additional case studies *e.g. productive system under high (real) load*

The Team

- Head of Department
  - Univ.Prof. Dr. Gabriele Kotsis
- Research and Teaching Assistants
  - Dipl.Ing Kerstin Altmanninger
  - Dipl.Ing. Sabine Bachmayer
  - Dr. Ismail Khalil Ibrahim
  - Dr. Wieland Schwinger
- Project Researchers and PhD Students
  - Stefan Mitsch
  - Martin Pinzger
  - Wolfgang Pointner
  - Martin Wischenbart
  - Elena Zanzani
- Secretary
  - Angela Kohl
- Technical Support
  - Fabian Mergl

Overview of Research

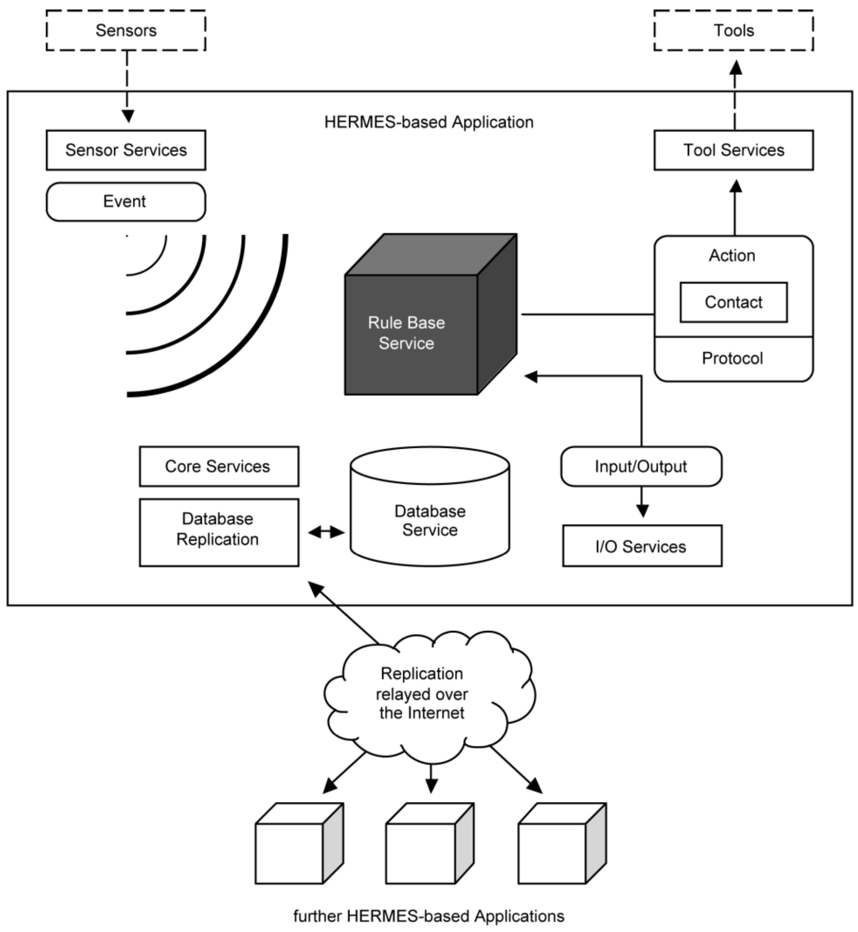


Ubiquitous Communication Management - HERMES

- Vision: support users' **communication needs**
  - **Gain knowledge** of users' communication needs, recognize behavior patterns and learn
  - Provide **appropriate communication tool support** based on precedent data mining and resultant knowledge
  - **Execute** appropriate communication-related actions

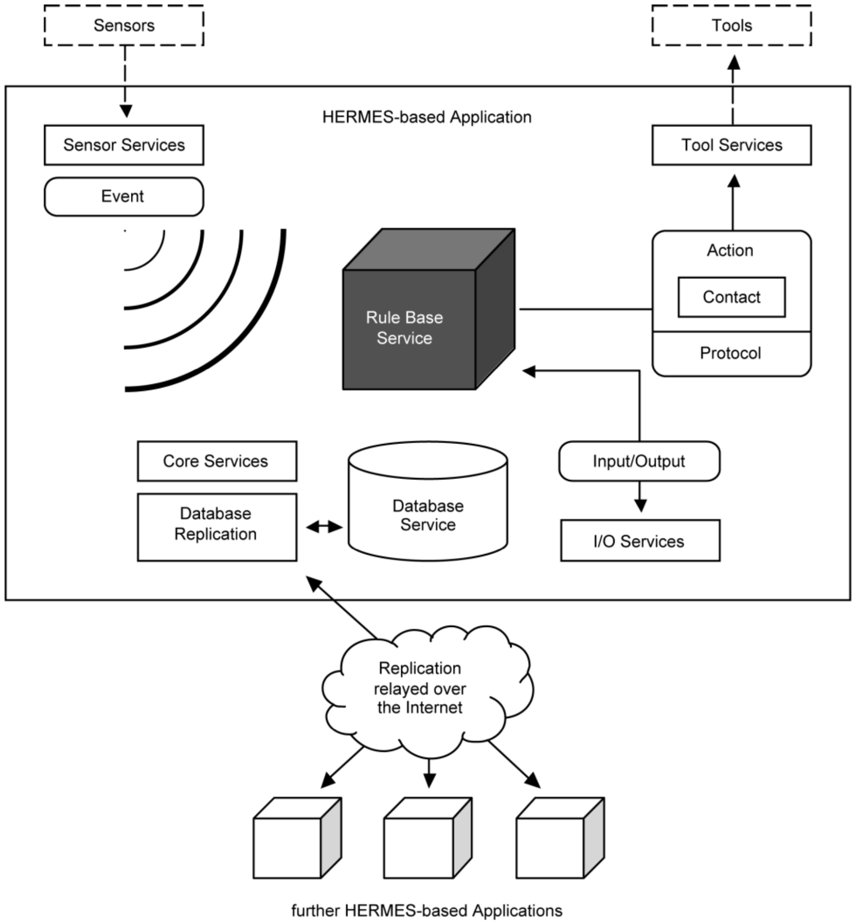
HERMES Architecture Overview

- Operation starts with **Sensors**
  - Specialized data mining components
  - Responsible for gathering data of interest for any communication management-related task
  - Receive new data and publish this to all other components via events
  - Published events are routed to interested recipients, usually Rule Bases
- **Rule Bases** are regarded as brains
  - Represent key idea of a ubiquitous communication management system: ability to learn from previous experience in the domain of communications
  - Supposed to analyze data contained in events and take appropriate communication-related actions



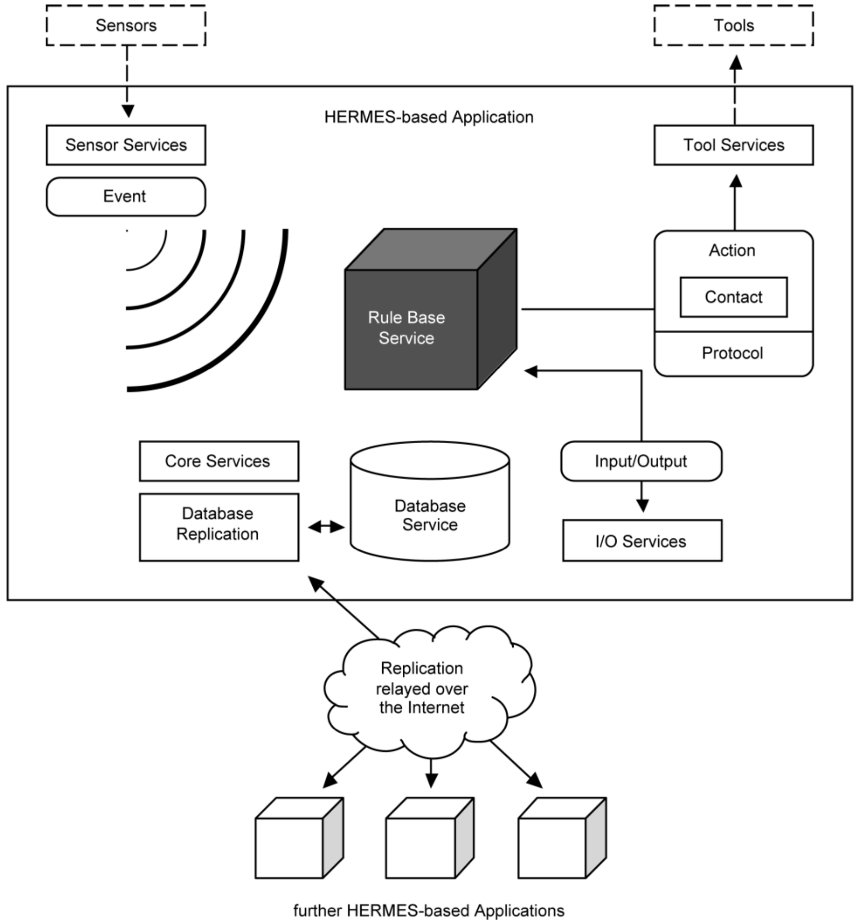
HERMES Architecture Overview

- Possibilities to interact with environment
  - Send Inputs/Outputs to IO Services
  - Send actions to Tool Services
- **Inputs/Outputs** wrap actual values
  - Generic approach enables to switch between Input/Output methods and to develop further methods for different devices
  - Numerous predefined Inputs/Outputs to use out of the box
- **Actions** are communication-related intents
  - Usually passed from Rule Bases to Tools
  - Series of predefined Actions
- **Tools** are responsible for executing received communication-related Actions
  - May forward Actions to existing communication tools and interact with those

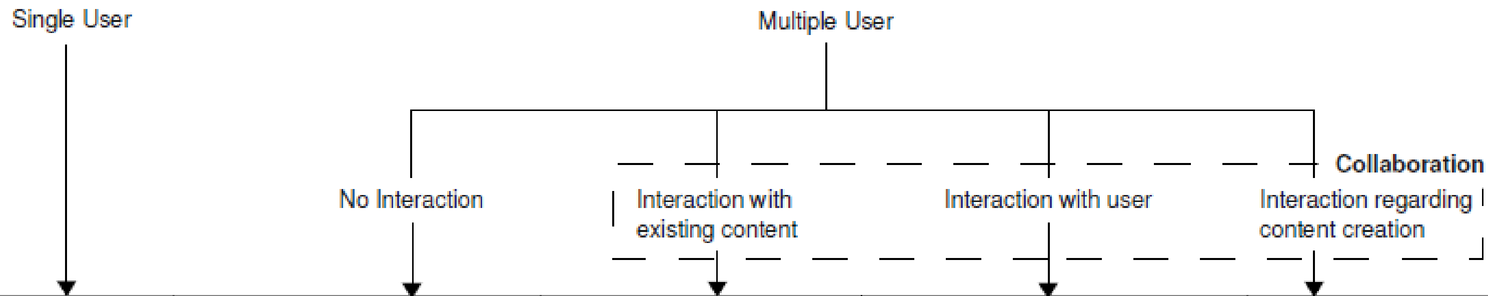


HERMES Architecture Overview

- Knowledge of users' communication needs supposed to be persistent in embedded database
- Communication between framework-based applications powered by XMPP communication platform
  - Primarily intended for simultaneous knowledge replication
- Framework's structure enables framework-based applications to run in distributed heterogeneous networks at the same time

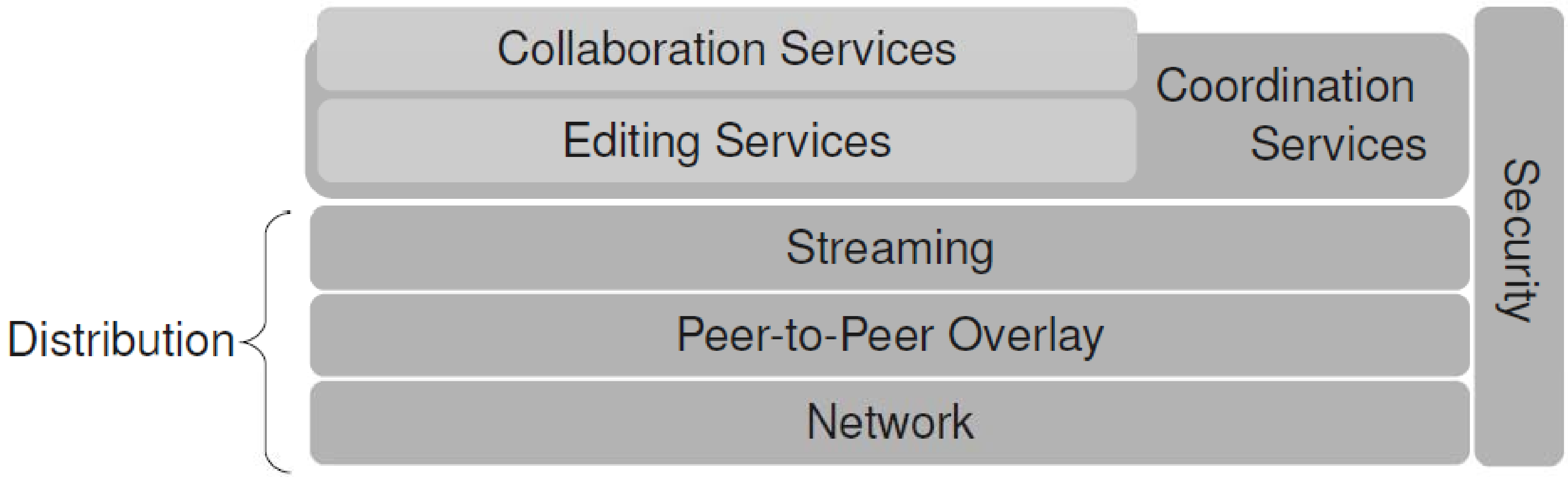


Collaborative Streaming Media



Actions	Create Streaming Session, Watch	Data Acquisition / Distribution	Watch, Control, Edit the Stream	Invite / Join Session, Communicate, (Automated) Awareness	Create, Annotate
Type	Single	Collaborative	Shared / Collaborative	Collaborative	Collaborative
Mode		Synchronous	Synchronous Asyncrhonous	Synchronous Asynchronous	Synchronous Asynchronous
Architecture	Client-Server, CDN	Peer-to-Peer	Client-Server, CDN	Client-Server, CDN	Peer-to-Peer
Tools		Hierarchical Collaborative Multicast [16], CoolStreaming [21], DISCOVR [14], COSMOS [13], [12], STARCast [19], [20]	Zync [18], [15], Comodin [7], [6] [4], [3], [5], coStream [8], [11], [10], [9]	Zync [18], [15], , Comodin [7], [6] [4], [3], [5], coStream [8], [11] [10], [9],	CWaCTool [17]

Collaborative Streaming Architecture

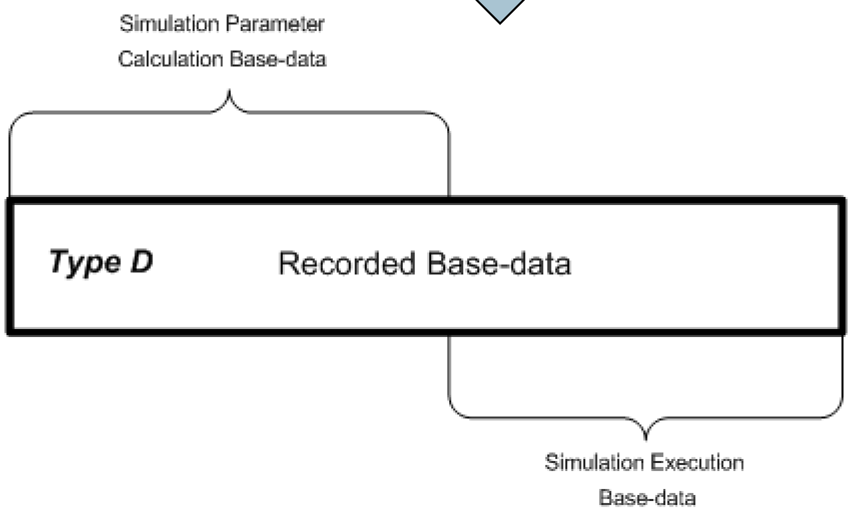


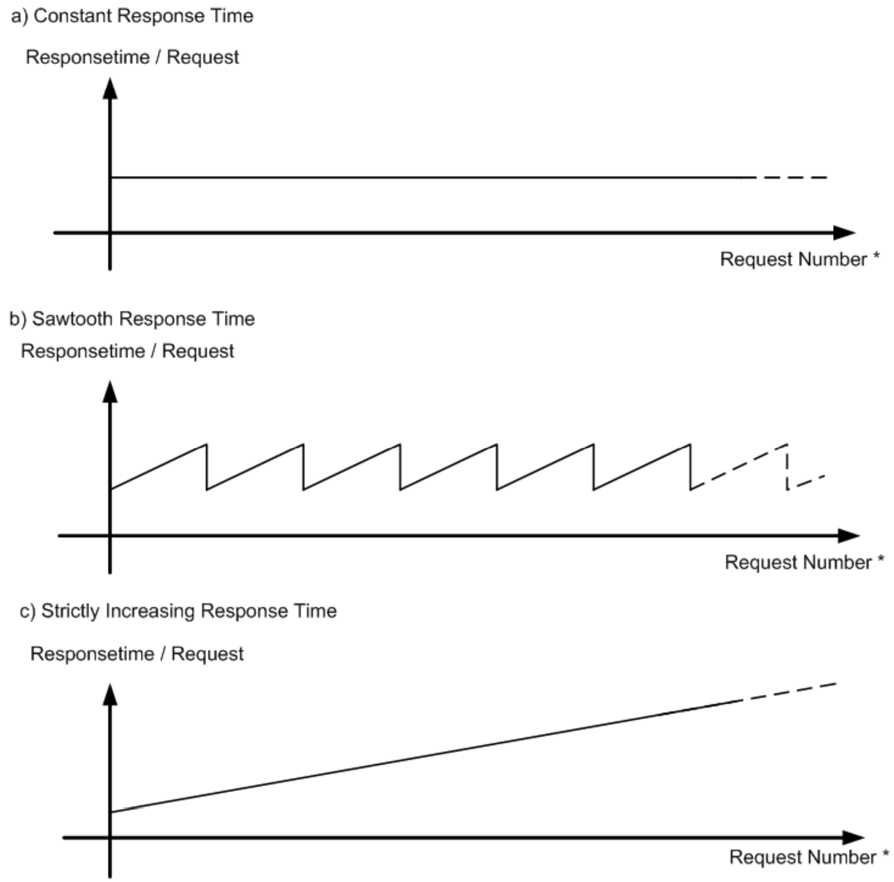
Thank you for your attention!

Case Study 1/5

Automatic Simulation Model Parameter Adjustment

- Case Study Setup
  - Test – Web – Sites
  - Use of Recorded Data





* By Request Number the ID of a request is meant, the number of requests sent to the SUT in one second is assumed constant!

Case Study 2/5

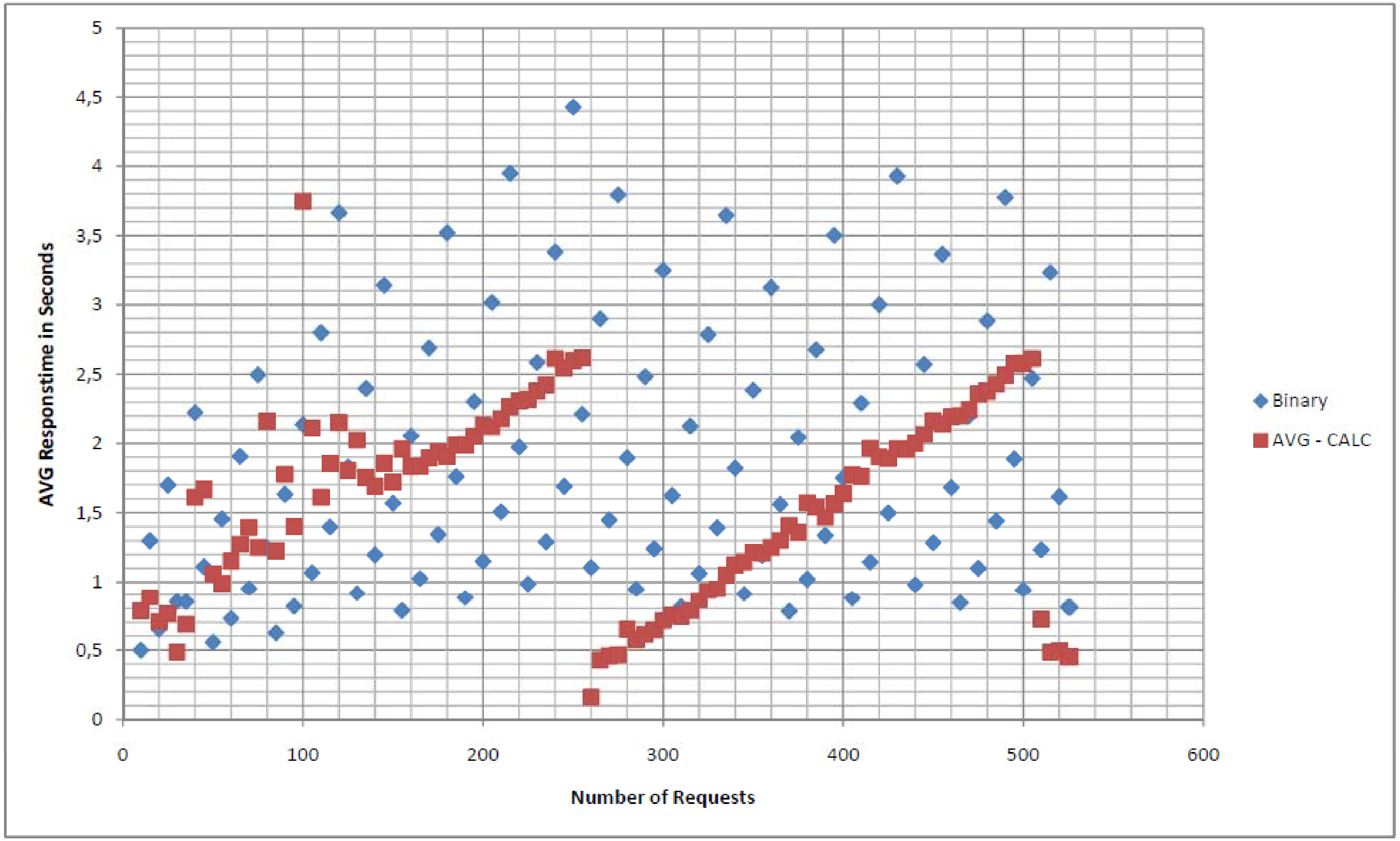
Automatic Simulation Model Parameter Adjustment

❑ Executed Case Studys

- 1 Minute
  - Constant Response Time (**ARMA**, **AVG**, Binary)
  - Sawtooth Response Time (**ARMA**, AVG, Binary)
  - Strictly Increasing Response Time (**ARMA**, **AVG**, Binary)
- 15 Minutes
  - Constant Response Time (**ARMA**, **AVG**, **ARMA G.**, Binary)
  - Sawtooth Response Time (**ARMA**, ARMA G., AVG, Binary)
  - Strictly Increasing Response Time (**ARMA G.**, AVG, ARMA, Binary)
- 60 Minutes
  - Constant Response Time (**ARMA**, **AVG**, ARMA G., Binary)
  - Sawtooth Response Time (**ARMA G.**, AVG, ARMA, Binary)
  - Strictly Increasing Response Time (**ARMA G.**, **Binary**, **AVG**, ARMA)

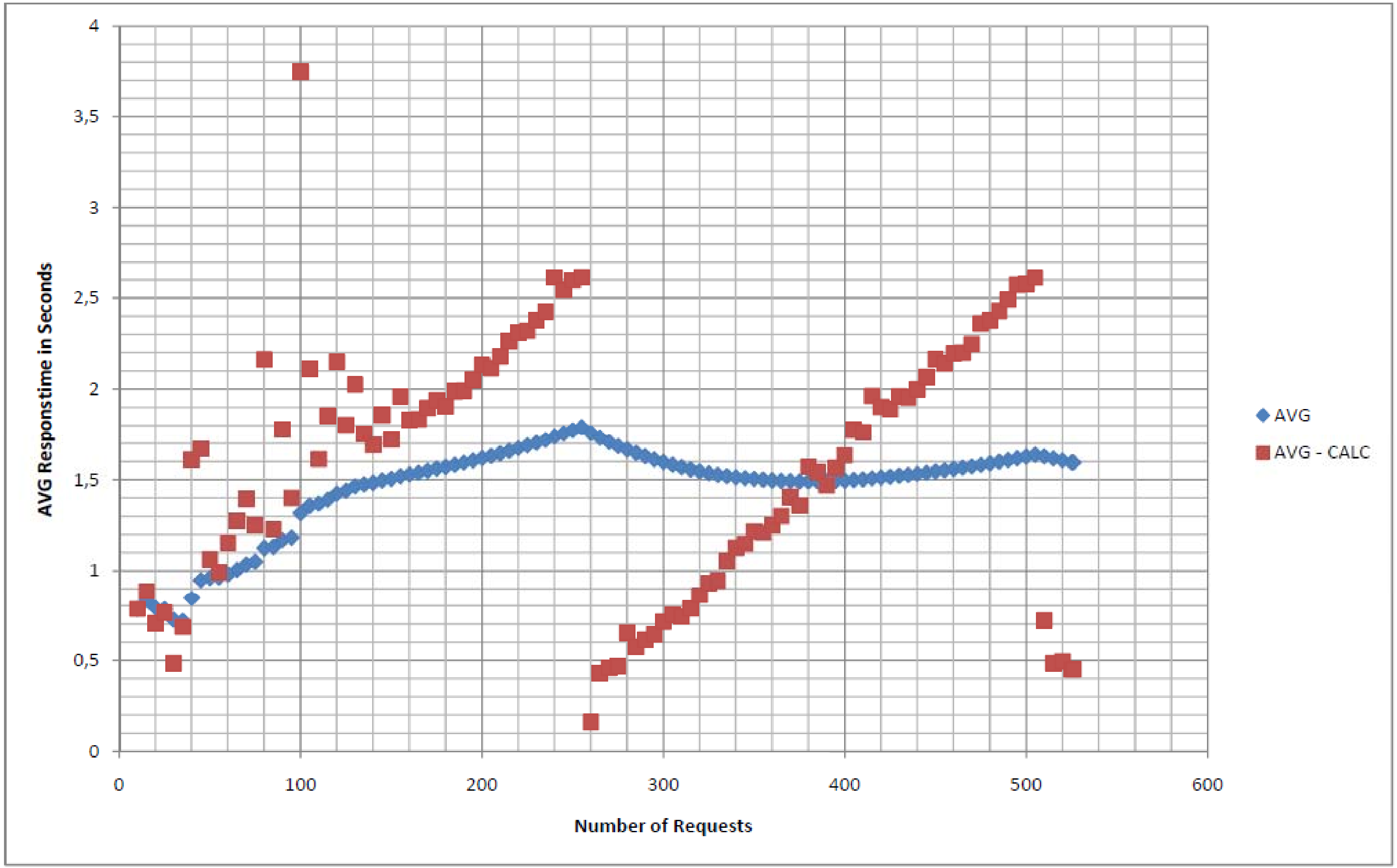
Case Study 3/5 15 Minutes Sawtooth 1/3

Automatic Simulation Model Parameter Adjustment



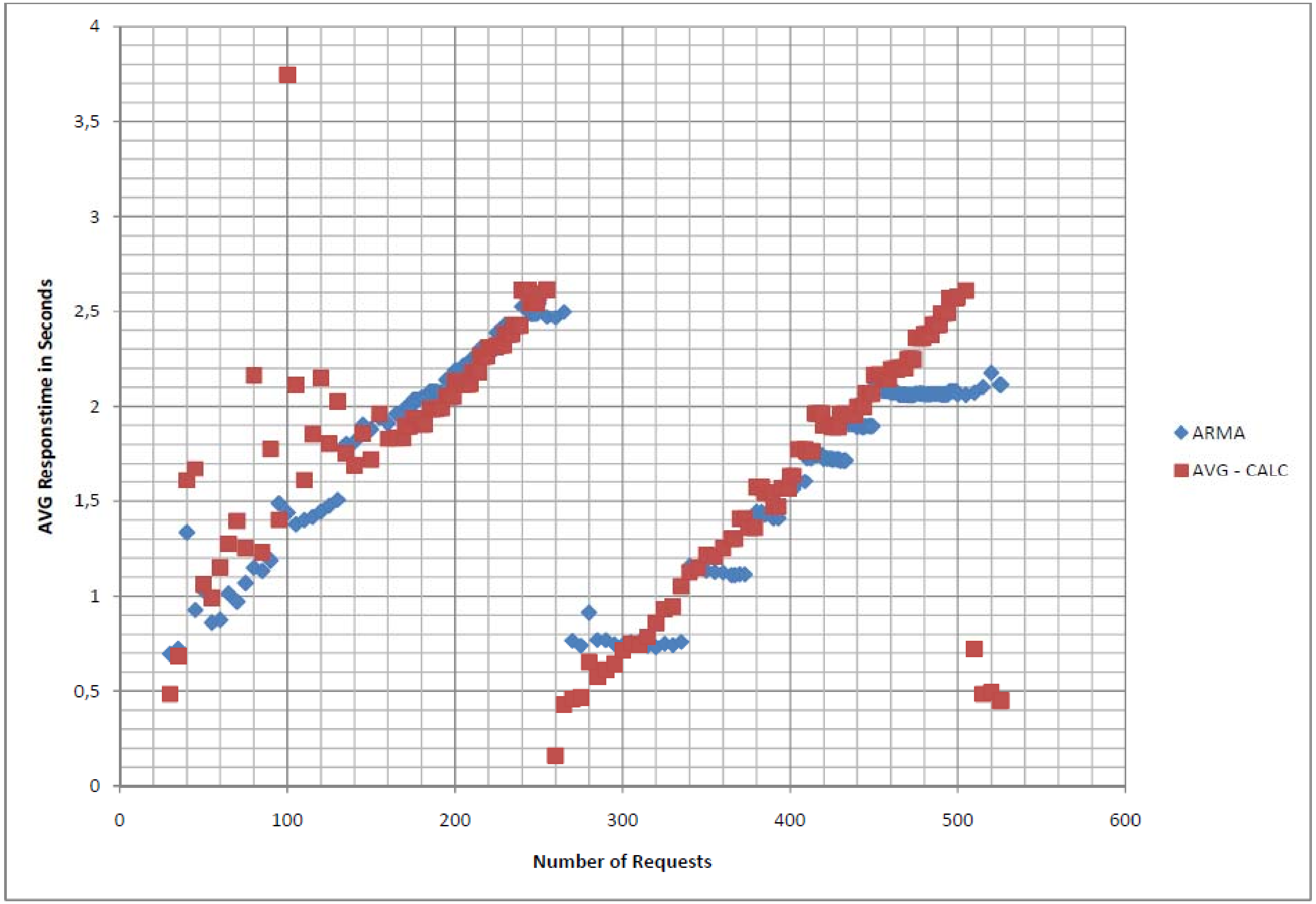
Case Study 3/5 15 Minutes Sawtooth 2/3

Automatic Simulation Model Parameter Adjustment



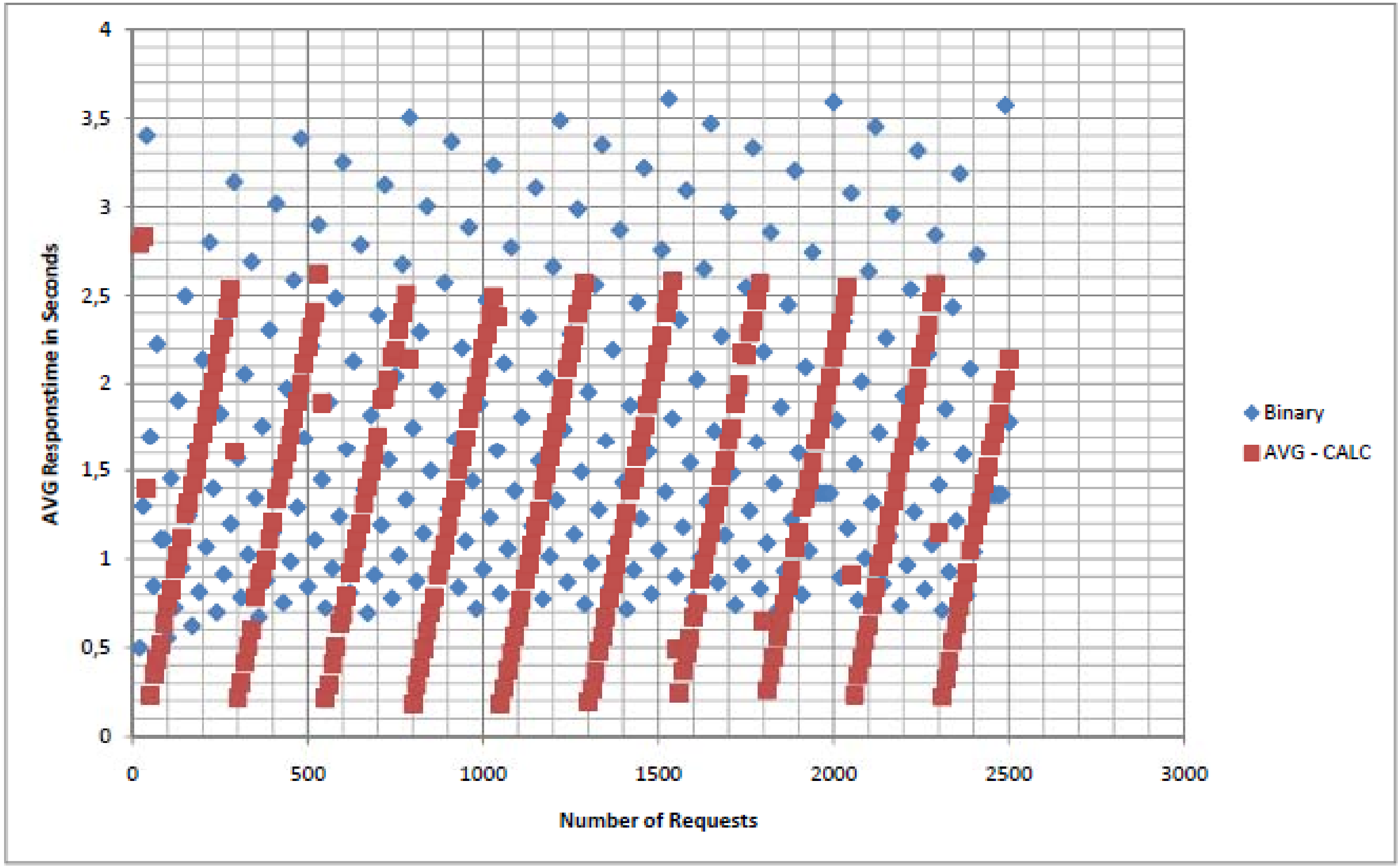
Case Study 3/5 15 Minutes Sawtooth 3/3

Automatic Simulation Model Parameter Adjustment



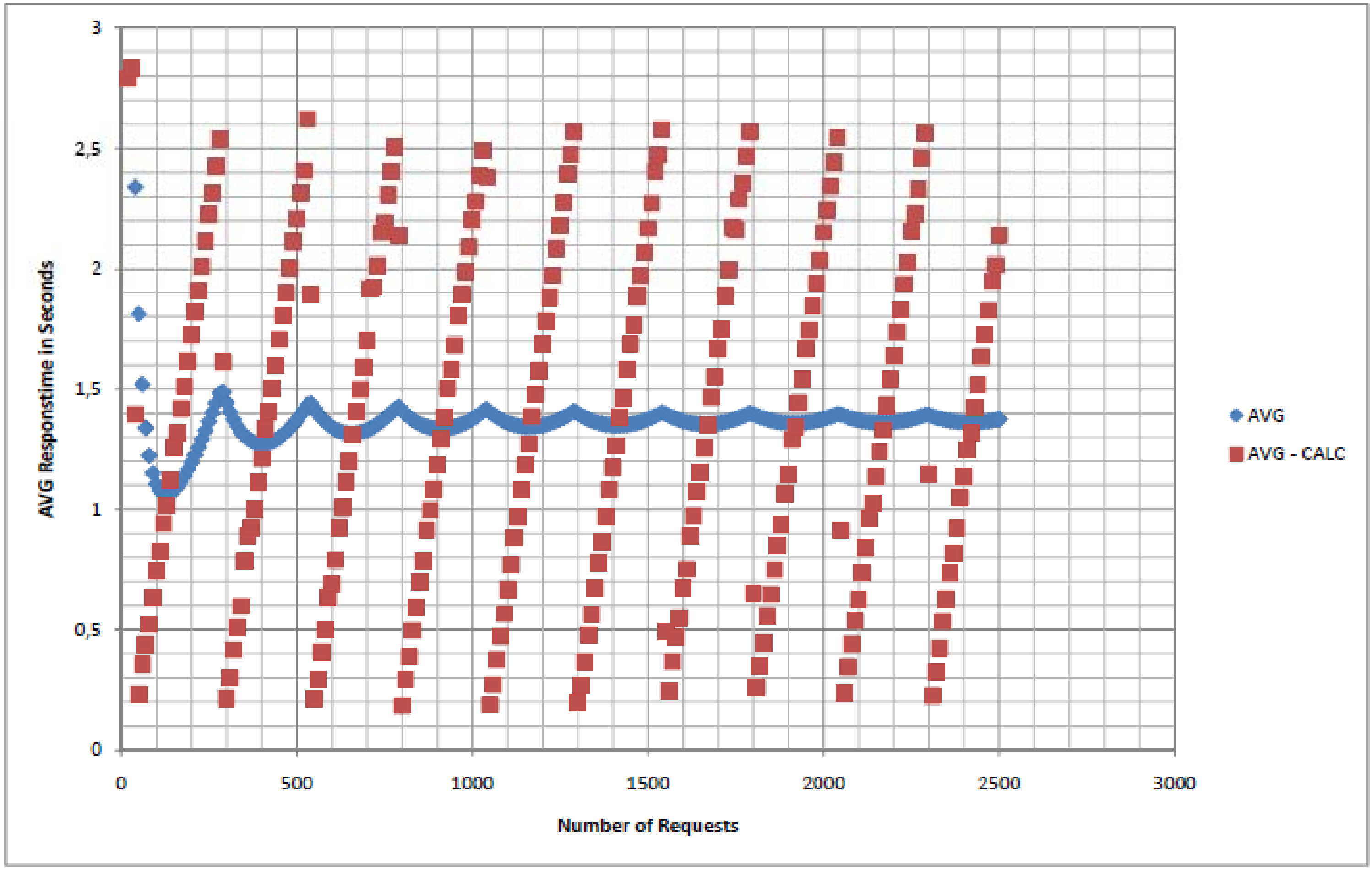
Case Study 4/5 60 Minutes Sawtooth 1/3

Automatic Simulation Model Parameter Adjustment



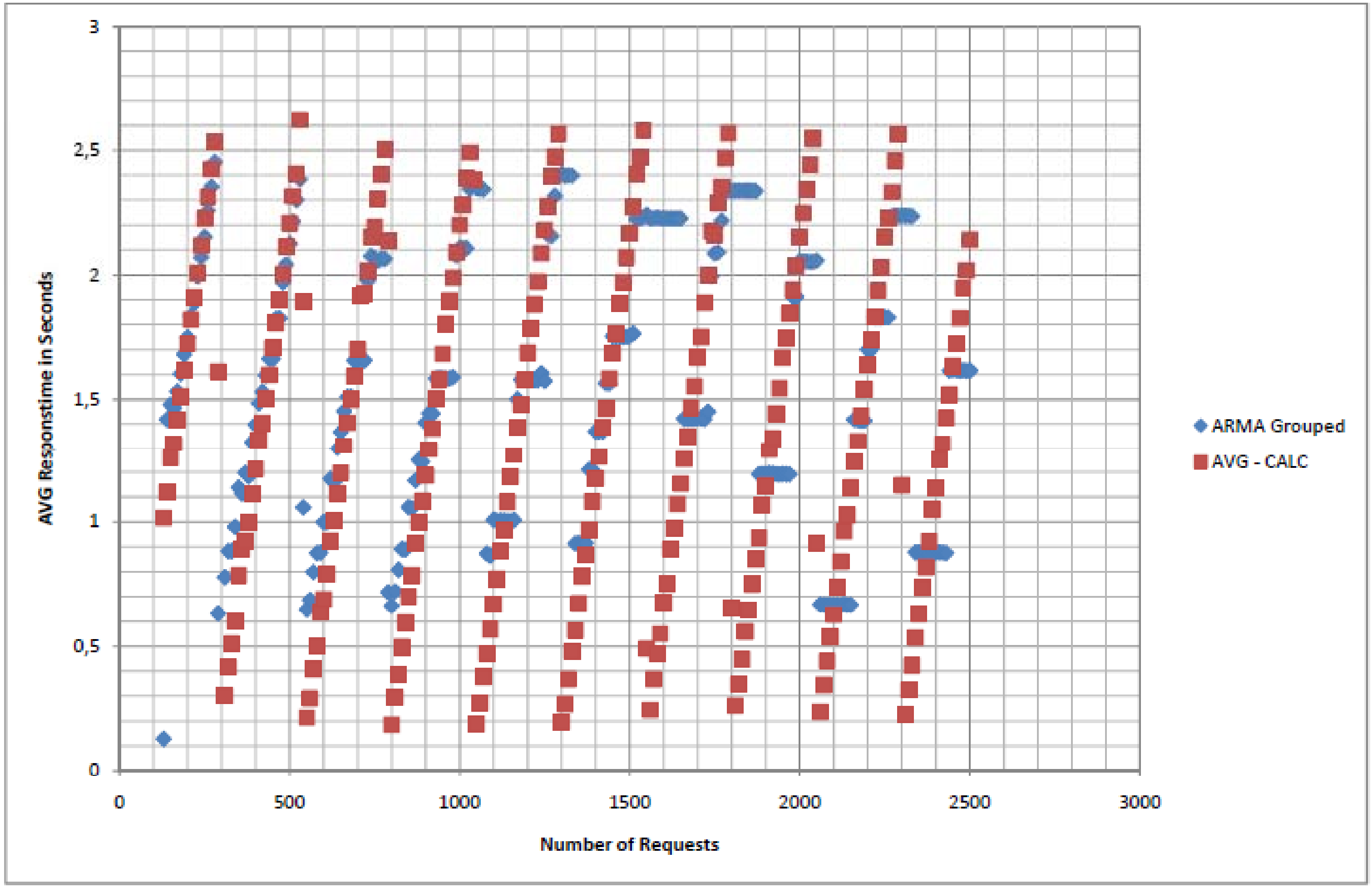
Case Study 4/5 60 Minutes Sawtooth 2/3

Automatic Simulation Model Parameter Adjustment



Case Study 4/5 60 Minutes Sawtooth 3/3

Automatic Simulation Model Parameter Adjustment



Case Study 5/5

Automatic Simulation Model Parameter Adjustment

- Which approach performs best for varying sample size?
  - 1 Minute
    - (1) ARMA; (2) AVG
  - 15 Minutes
    - (1) ARMA and ARMA G.; (2) AVG
  - 60 Minutes
    - (1) AVG; (2) ARMA G.
  - Global Ranking
    - ARMA G. (1,500)
    - AVG (1,555)
    - ARMA (1,666)
    - Binary (3,111)