HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

Model Transformations in Non-Functional Analysis

Assistant Prof. Dr.-Ing. Steffen Becker

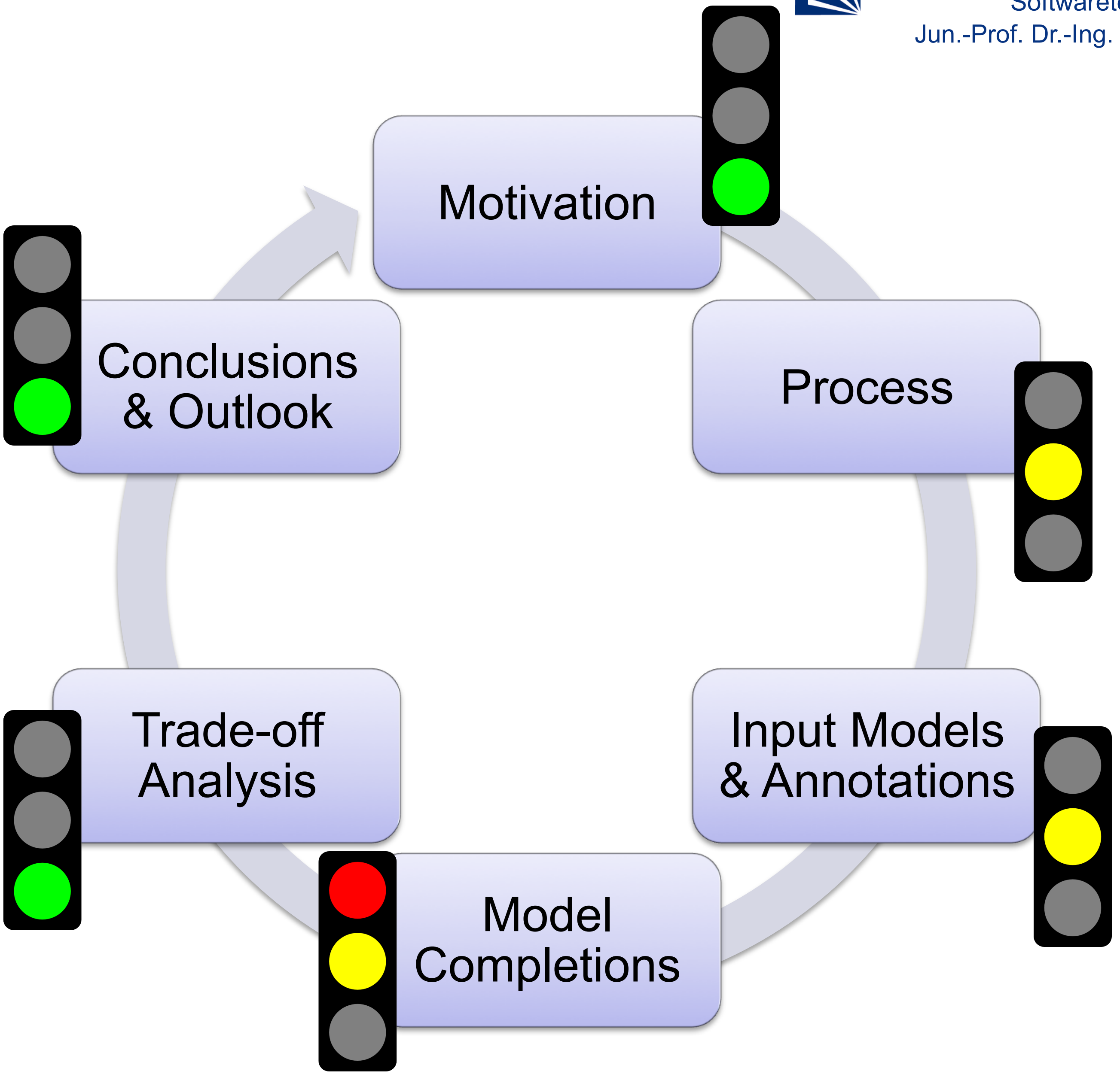
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Personal Background

- Studied Wirtschaftsinformatik in Darmstadt
  - Meta-Modelling Focus
  - Component Meta-Models and Marketplaces
- Research Assistant in Oldenburg
  - Palladio Component (Meta-)Model
  - Model-Driven Architecture-Based Quality Prediction
- Research Assistant in Karlsruhe
- Department Manager at FZI
- Disputation in Oldenburg to become
  Dr.-Ing. in a performance engineering context
- Since April 2010 Juniorprofessor at U Paderborn

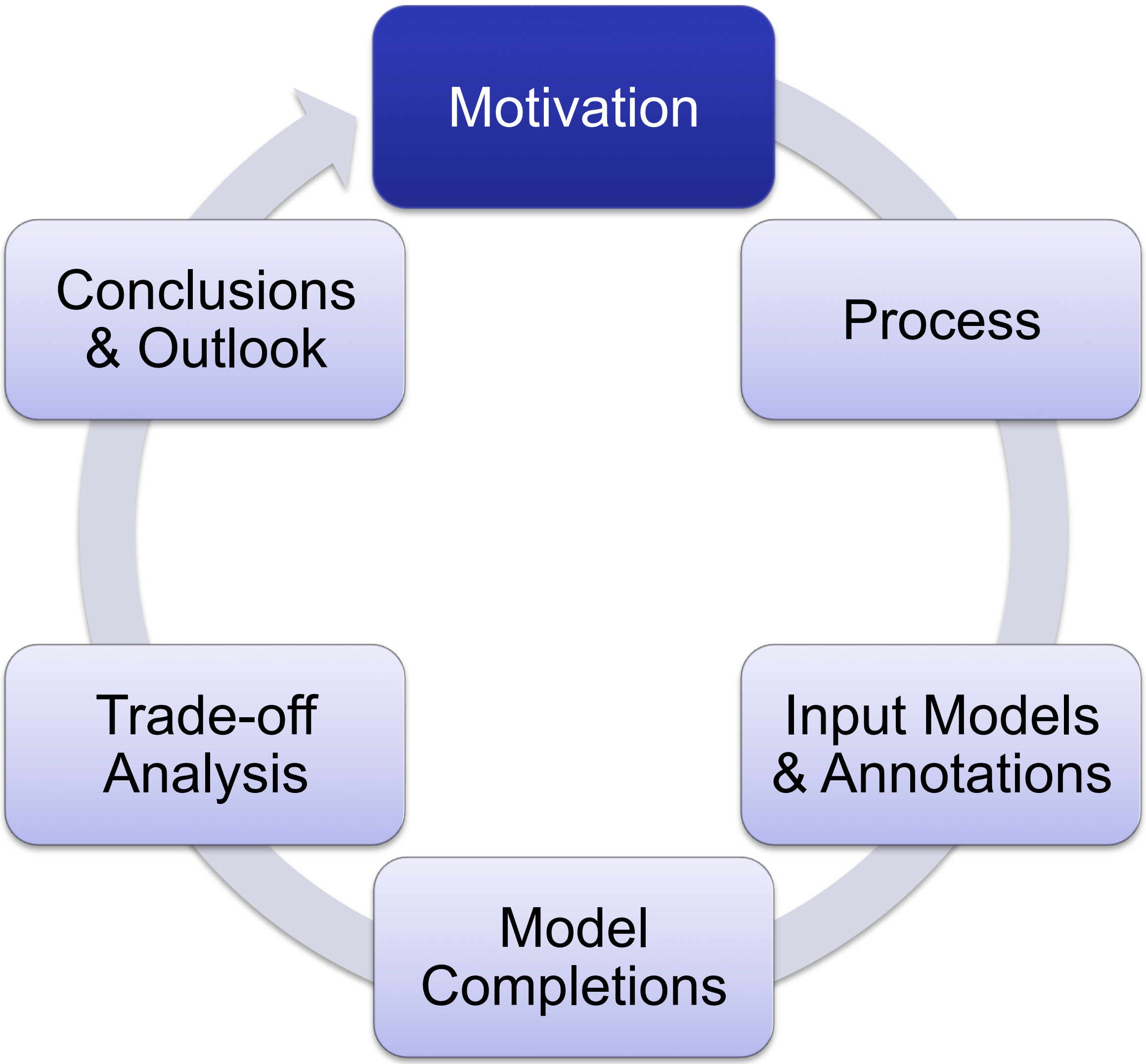
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Outline



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Outline



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Example: SAP Business By Desgin

leidet offenbar noch unter Kinderkrankheiten. Die Wirtschaftszeitung *Euro am Sonntag* berichtet in ihrer aktuellen Ausgabe von Performance-Problemen sowie einer drohenden Verschiebung des Starttermins und beruft sich dabei auf Insider. So laufe die Software derzeit noch deutlich langsamer als die Standardanwendung R/3 und verkrafte noch nicht einmal 10 gleichzeitige Nutzer. Von der Zielvorgabe, 1000 parallele Nutzer zu bedienen, sei das Produkt noch "Lichtjahre entfernt". Hauptproblem sei das Design des Systems, das Servern den Zugriff auf brachliegende Kapazitäten anderer Rechner nicht ermögliche. Dem Insider zufolge werde das zu einer Verzögerung führen, das Produkt sei frühestens in einem Jahr marktreif.

- Development costs >300 Mio. €, est. TTM + 1 year (was 3 finally)
- Handles 10 users instead of the planned 1000
- Cite: „Main issue is the *system's design*, which disallows server to access free capacities on other machines…“

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Context?

System Type

- Web-based system
- Three-tier business information system

Characteristics

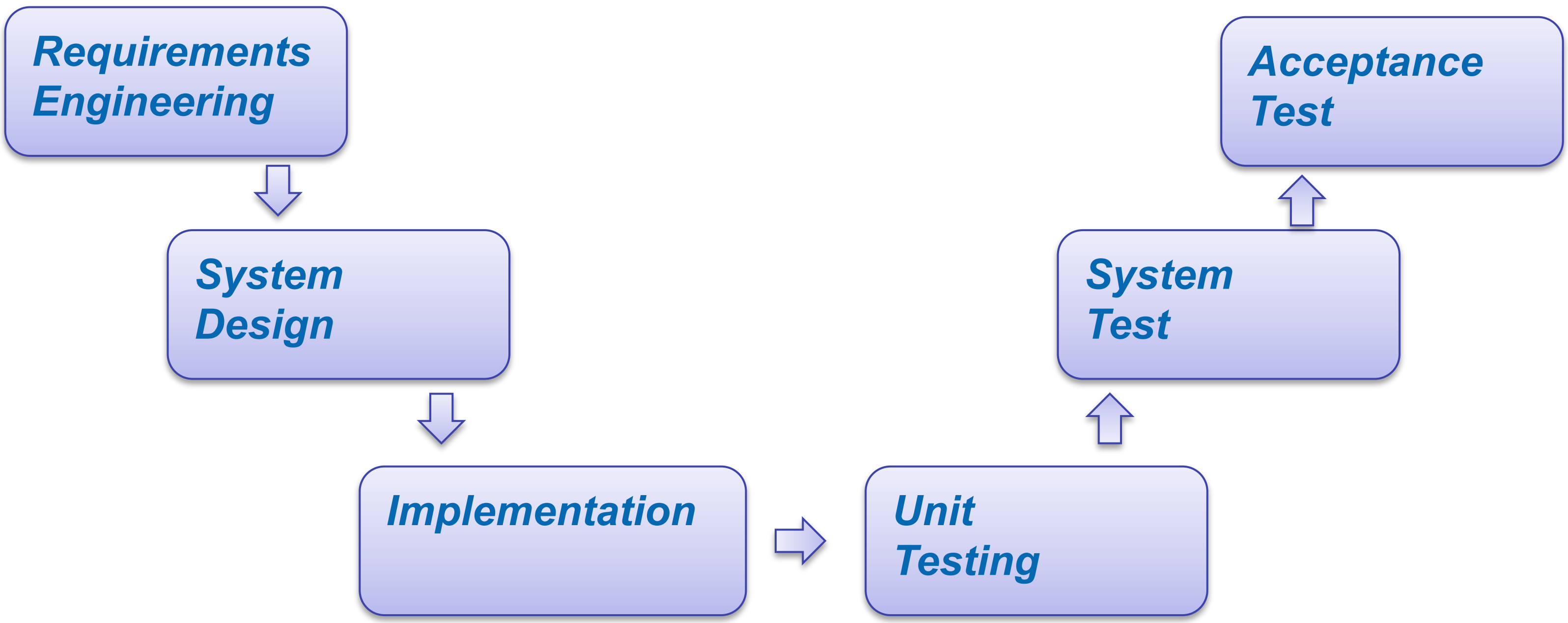
- Request-Response communication, often blocking
- Open world assumptions, users come whenever they like

Metrics of Interest

- Response time for requests, Utilization, Throughput
- PROFOD
- Costs

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

What might have happened?

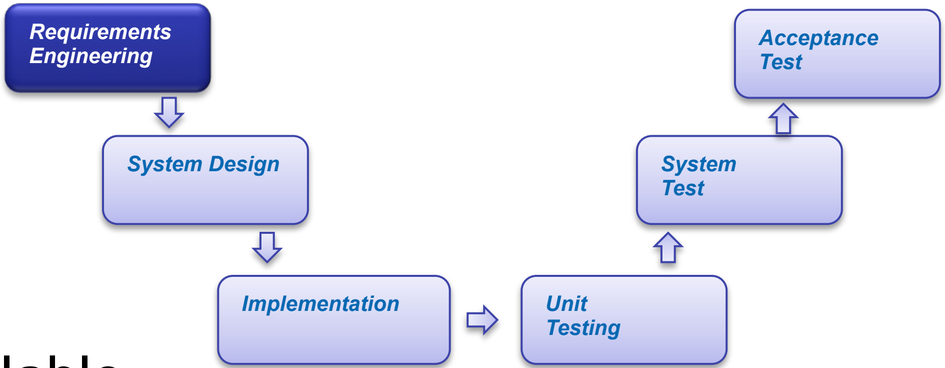


© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

What might have happened?

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

- Capture non-functional properties
- Typical issues
  - Imprecise
    “The system has to be scalable, reliable, performant, … and should not cost anything”
  - Ignore context
    “The system should be unavailable only 3 min / year”
  - Overly strict
    “The system should always react in 1sec”



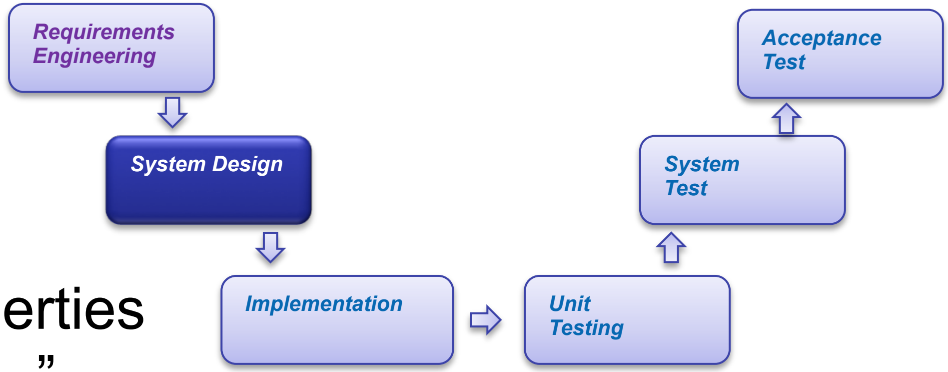
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Examples: HowTo do it better

- Response time of service bookFlight is 1ms
- Service fails 1h per year in the mean
- Development of the system cost 10.000€
- Average utilisation of server CPU is 80%
- Mean repair time of the server is 1h
- Throughput must at least be 5M transactions per hour

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

What might have happened?



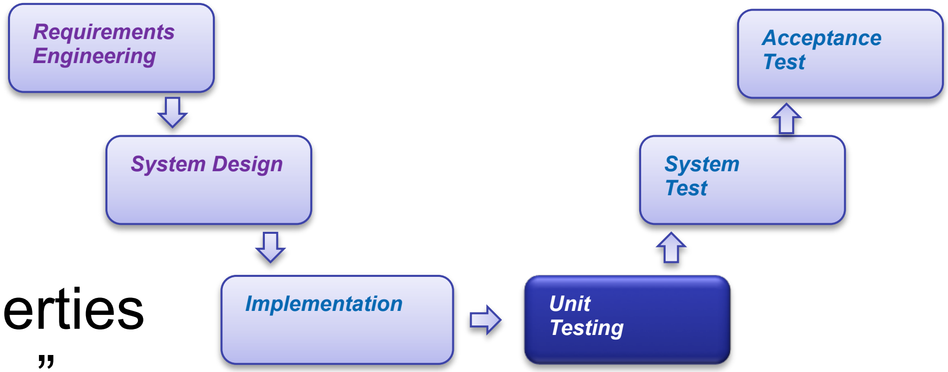
- Design architecture, classes, components, …
- Typical issues
  - Ignore non-functional properties
    “We can fix anything later…”
  - Reason on them
    Informal (SAAM, ATAM), based on experience
  - Document non-functional rationales for decisions
    Implicit in patterns or class designs
  - No context again
    “A cache in front of a data access is always good…”

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

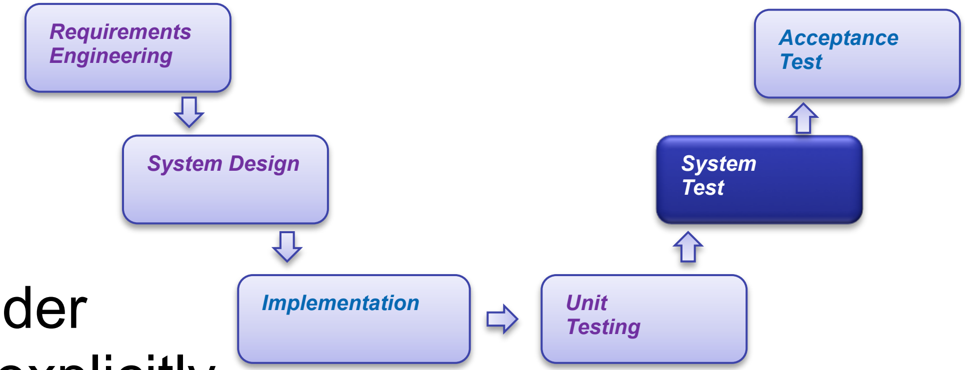
What might have happened?

- Test single entities
- Typical issues
  - Ignore non-functional properties
    "We can fix anything later…"
  - No context
    "I do not know the context of my module, and here it performs fine!"
  - No explicit non-functional tests
    Mainly test of module functionality



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

What might have happened?



- Test drive composed system
- Typical issues
  - For the first time we consider non-functional properties explicitly
  - Create context
    “Target is one million users, can we test that in our lab? What will it cost?”
- If issues arise
  - Re-Implement, Re-Design, Re-Negotiate
  - Project failures are likely
  - Most kinds of architecture rework is very costly

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Compared to: What is Engineering?

"[..] creative application of scientific principles to design or develop structures, machines, apparatus, [..] to forecast their behavior under specific operating conditions; all as respects an intended function, economics of operation and safety to life and property."

Systematic design of system	System model
	Environment model
	Requirements model
Forecast of quality aspects	Requirement fulfilment
	Economic operation
	Safety properties

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Engineering history

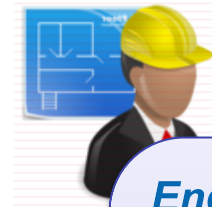
HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

Craftsmanship

- Customer and developer often identical
- Talent and experience guide development

Industrial production

- Work division
- Specialised education
- Tools developed by third-parties



Engineering

Optimized Quality of products and processes

requires

- Scientific foundations / theory
- Guidelines to apply methods to real world problems

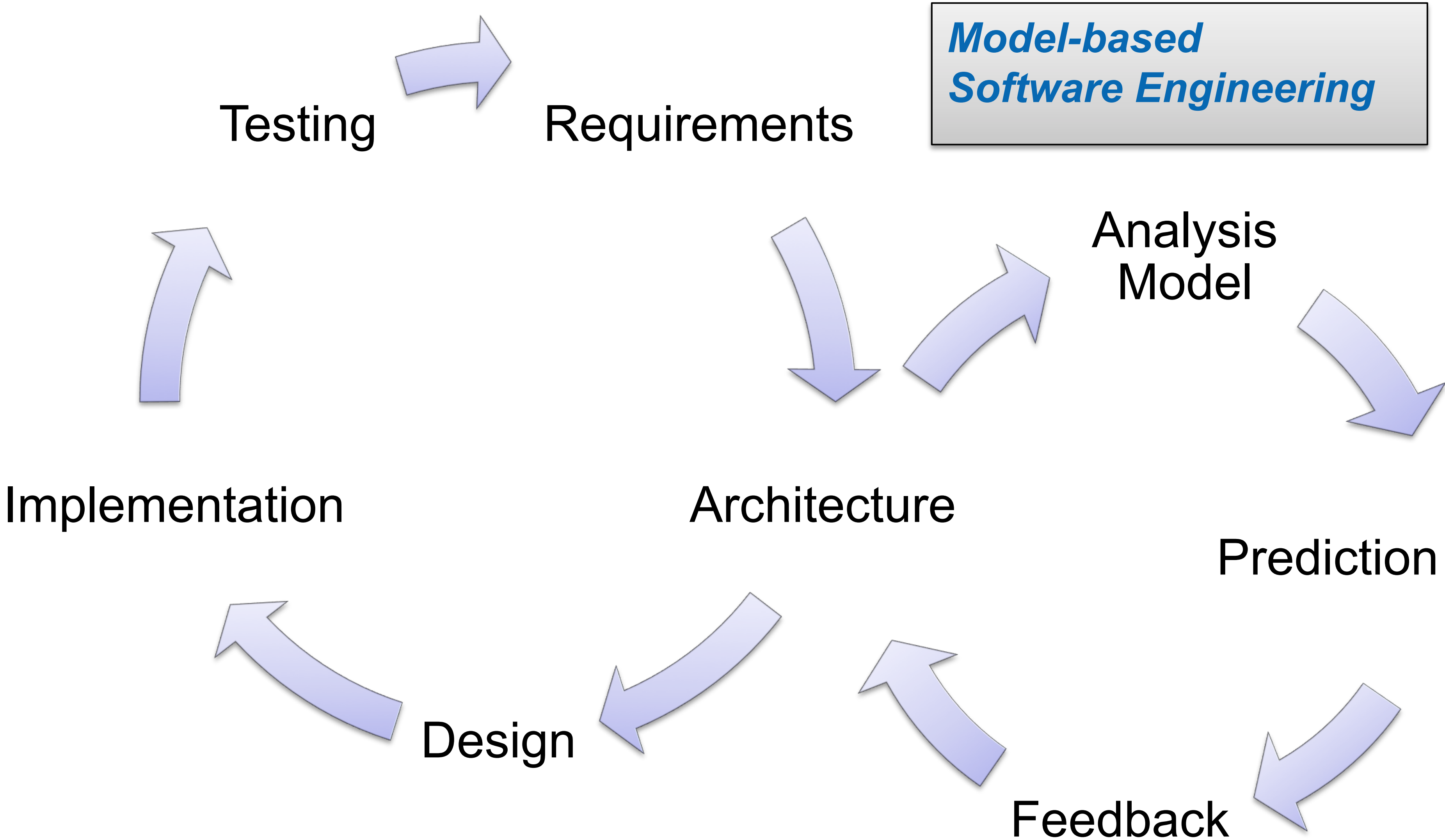
based on [Shaw&Garlan95]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

Development Cycles

Model-based Software Engineering

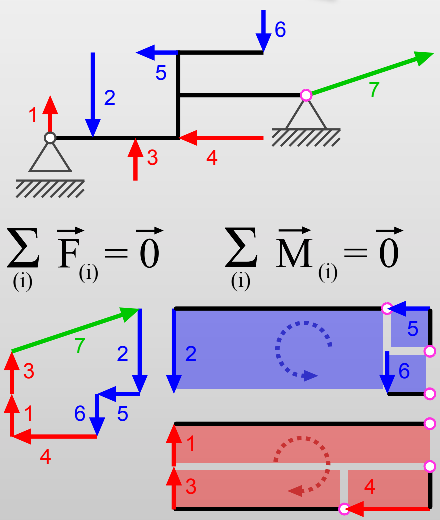


© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Civil Engineering Analogy

Model

Theory



$$\sum_{(i)} \vec{F}_{(i)} = \vec{0} \qquad \sum_{(i)} \vec{M}_{(i)} = \vec{0}$$

[http://de.wikipedia.org/wiki/Statik_%28Physik%29]

Analysis Results

3.7 Spannungsnachweis

$I_p = 0{,}25$

$\mu = 0{,}80$

$\eta = 2{,}0$

$Q_{zul.} = 0{,}7178\ \text{MN} \cdot 0{,}8 \cdot 0{,}5$

$= 0{,}287\ \text{MN}$

$= 287\ \text{kN}$

$N = 79{,}7 + 1{,}30^2 \cdot 1{,}30 \cdot 25$

$= 134{,}6\ \text{kN}$

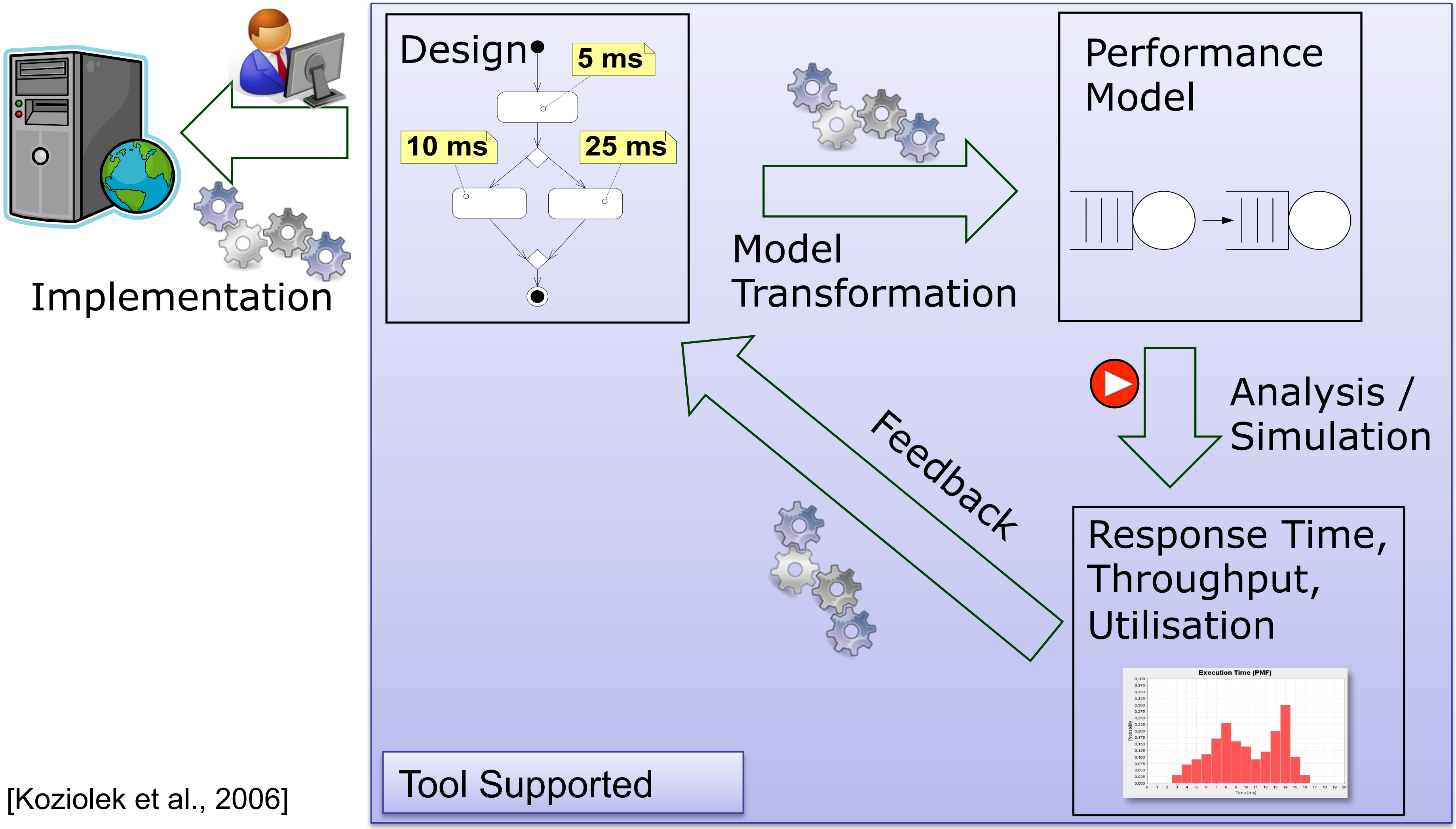
$$\boxed{V_{max} = N = 134{,}6\ \text{kN} < Q_{zul.} = 287\ \text{kN}}$$

Die vertikale Tragfähigkeit ist gewährleistet

[http://de.wikipedia.org/wiki/Baustatik]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

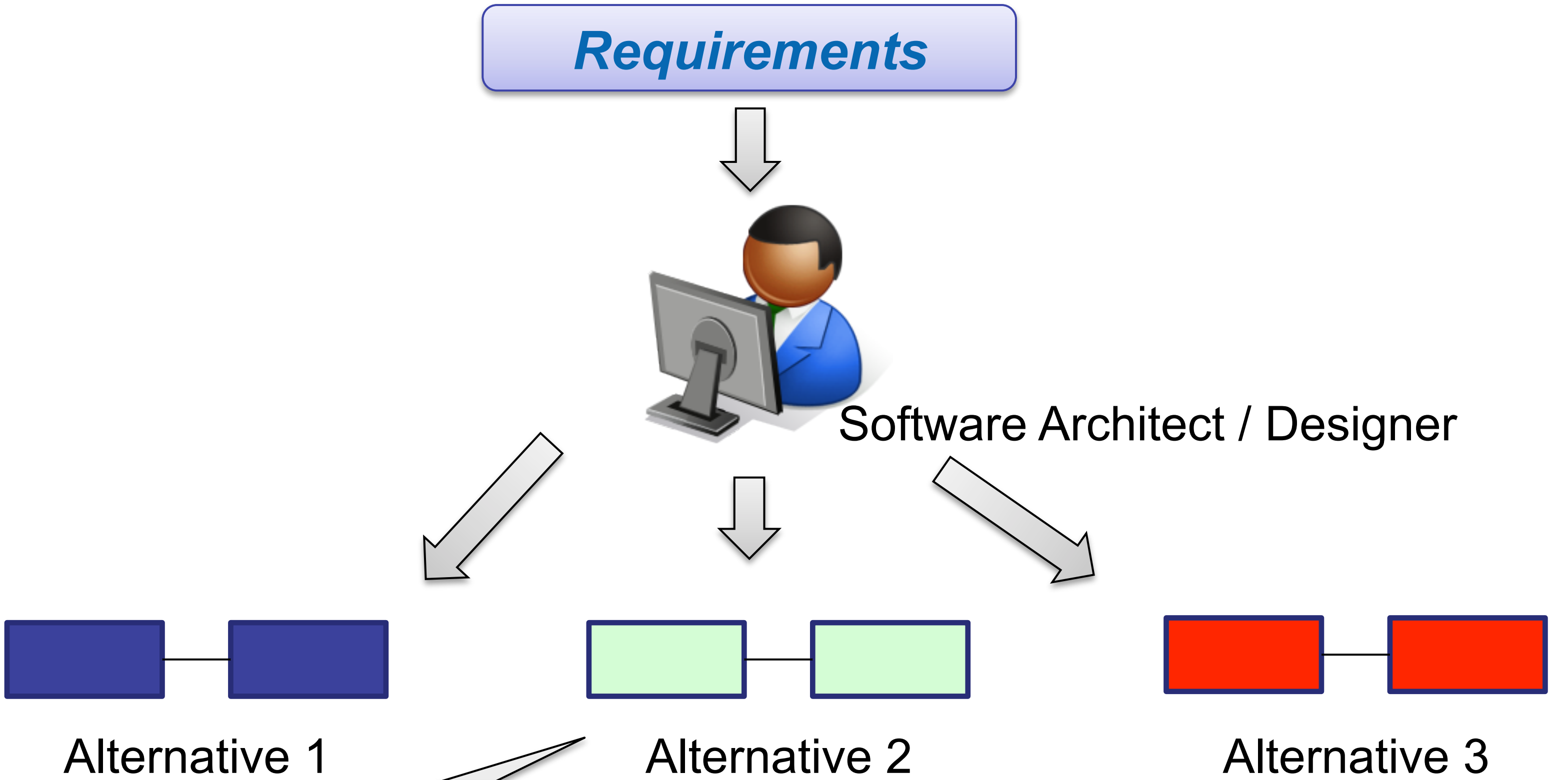
An Instance is Model-Driven Performance Engineering



[Koziolek et al., 2006]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Different alternative designs...



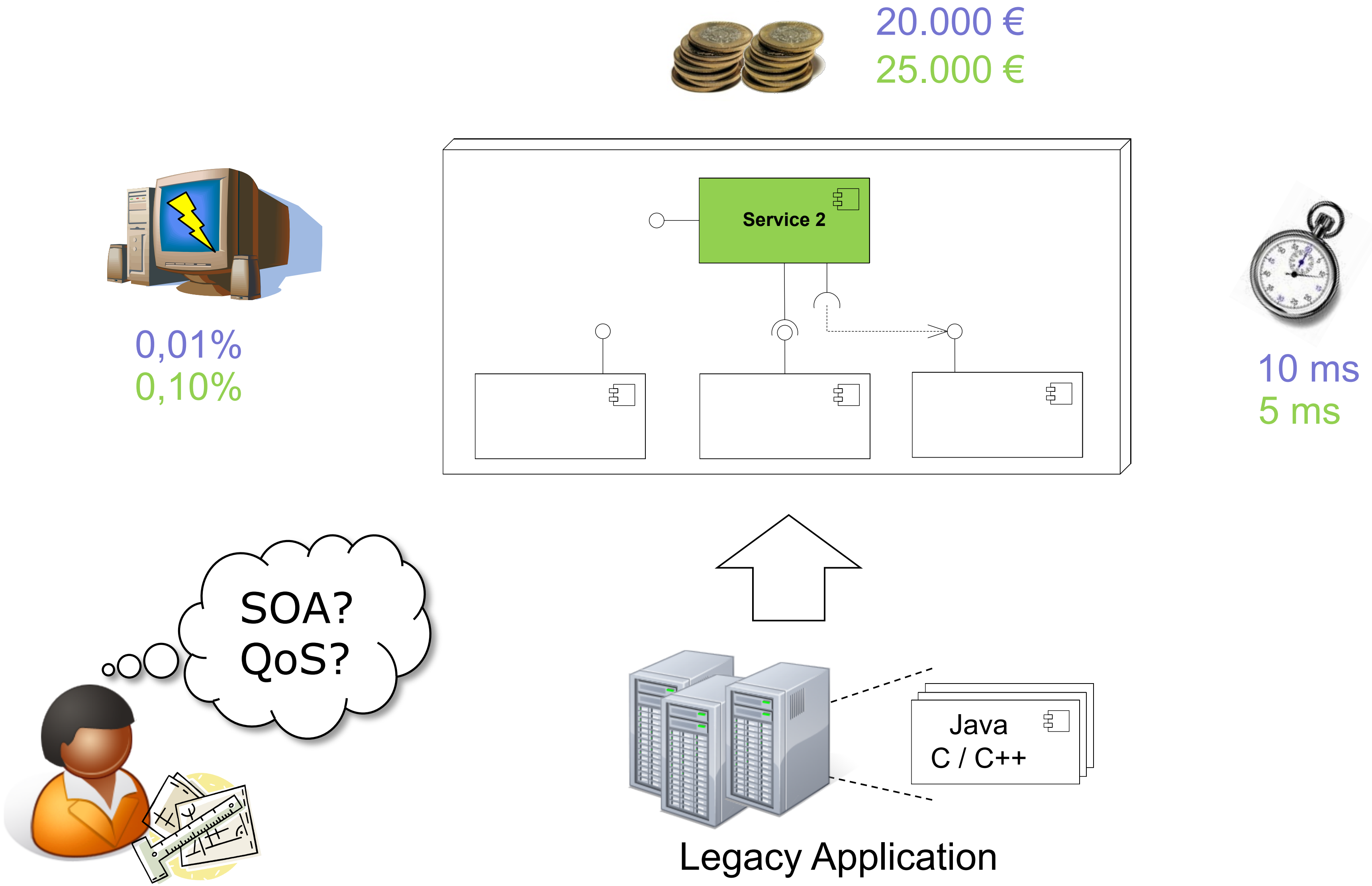
Assume: Same functionality, different QoS

Which alternative do we implement?

[SWA09]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Example…



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

What is evaluated?

Examples

- Encryption component: Length of the key
  - Security vs. performance
- Message formats: priority of message processing
  - Expressiveness / transmission security vs. performance (of the whole system)
- After such design alternatives are identified, they can be evaluated:
  - Detailed design
  - Application of analysis methods
  - Prototyping

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

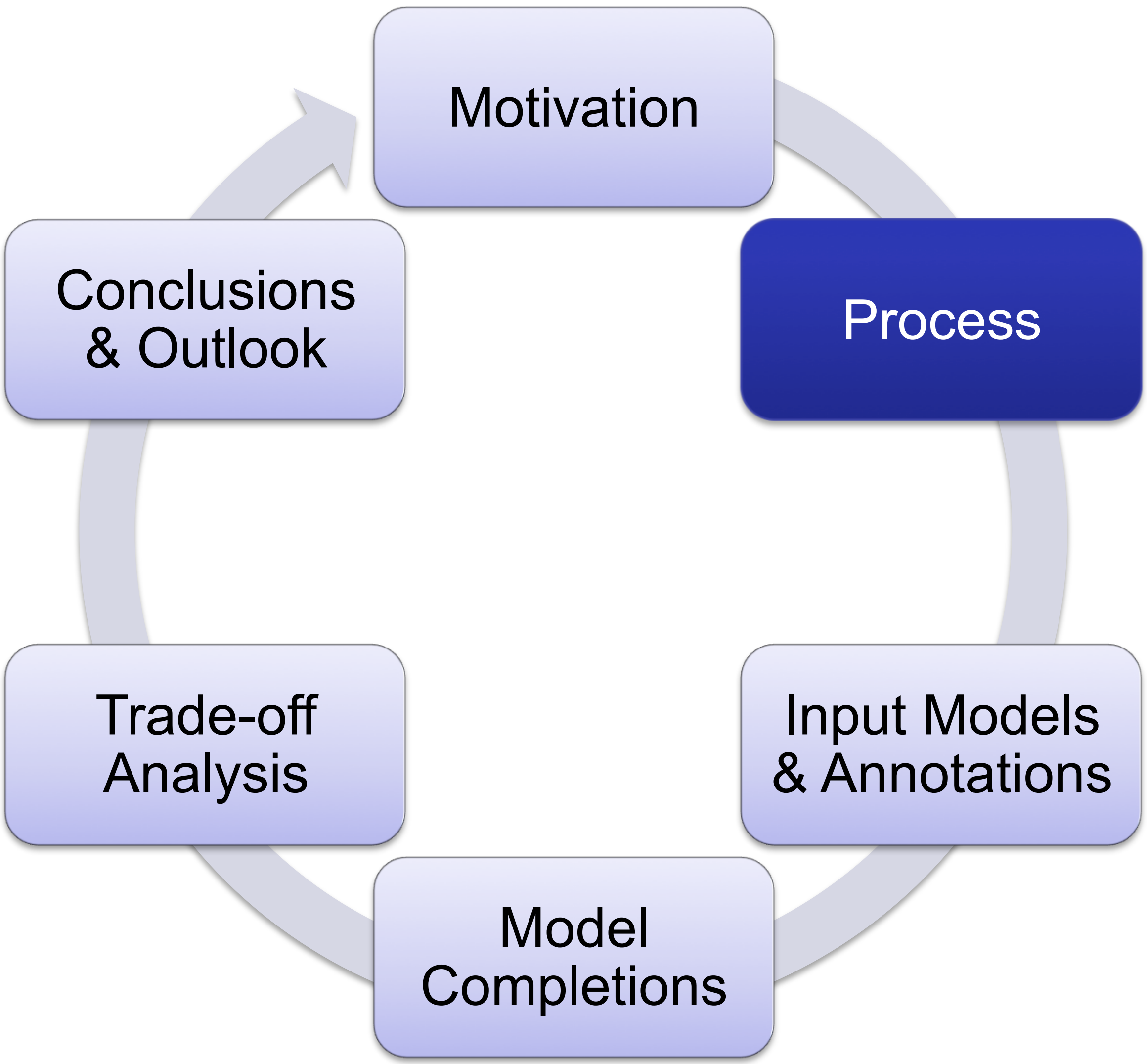
Further examples?

- On site assignment:
  - Collect some alternative design or architectural patterns
  - List their variants?
  - What is their expected impact wrt. quality properties?

- Examples
  - MVC vs. its variants?
  - MVC vs PAC?
  - Cache
  - Pooling
  - ...

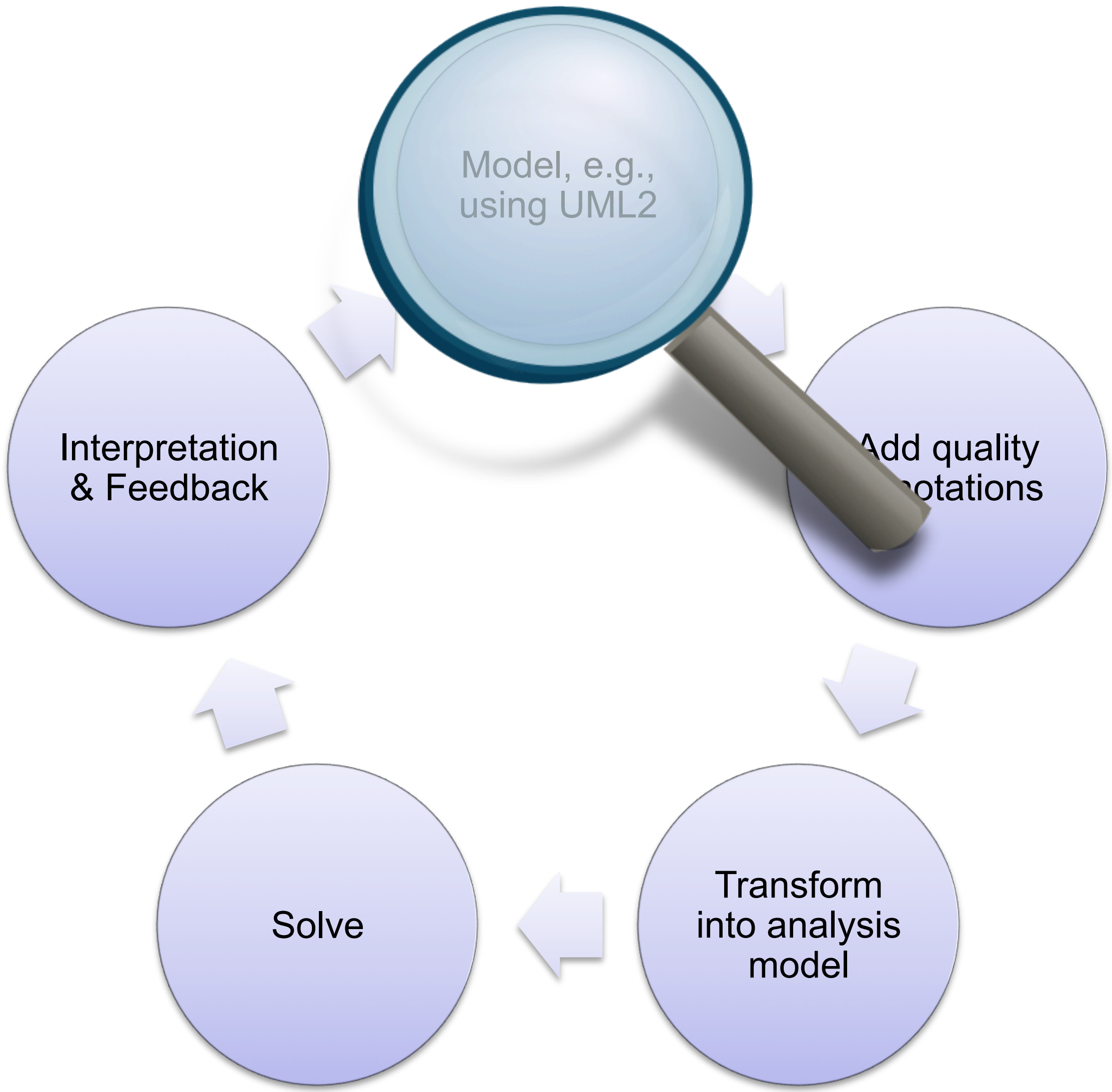
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Outline



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Process



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Model

Software Architects create a model of the SuS

- SW Models primarily focus on software & hardware – not QoS issues
- Software architects create and maintain models
  - Can they do this?
  - A question for empirical Software Engineering

Many languages exist, e.g.,

- UML2 + MARTE
- PalladioCM
- PUMA
- ProgressCM

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Annotations

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

Models need QoS annotations, e.g.,

- Resource demands
- Failure rates

Requirements?

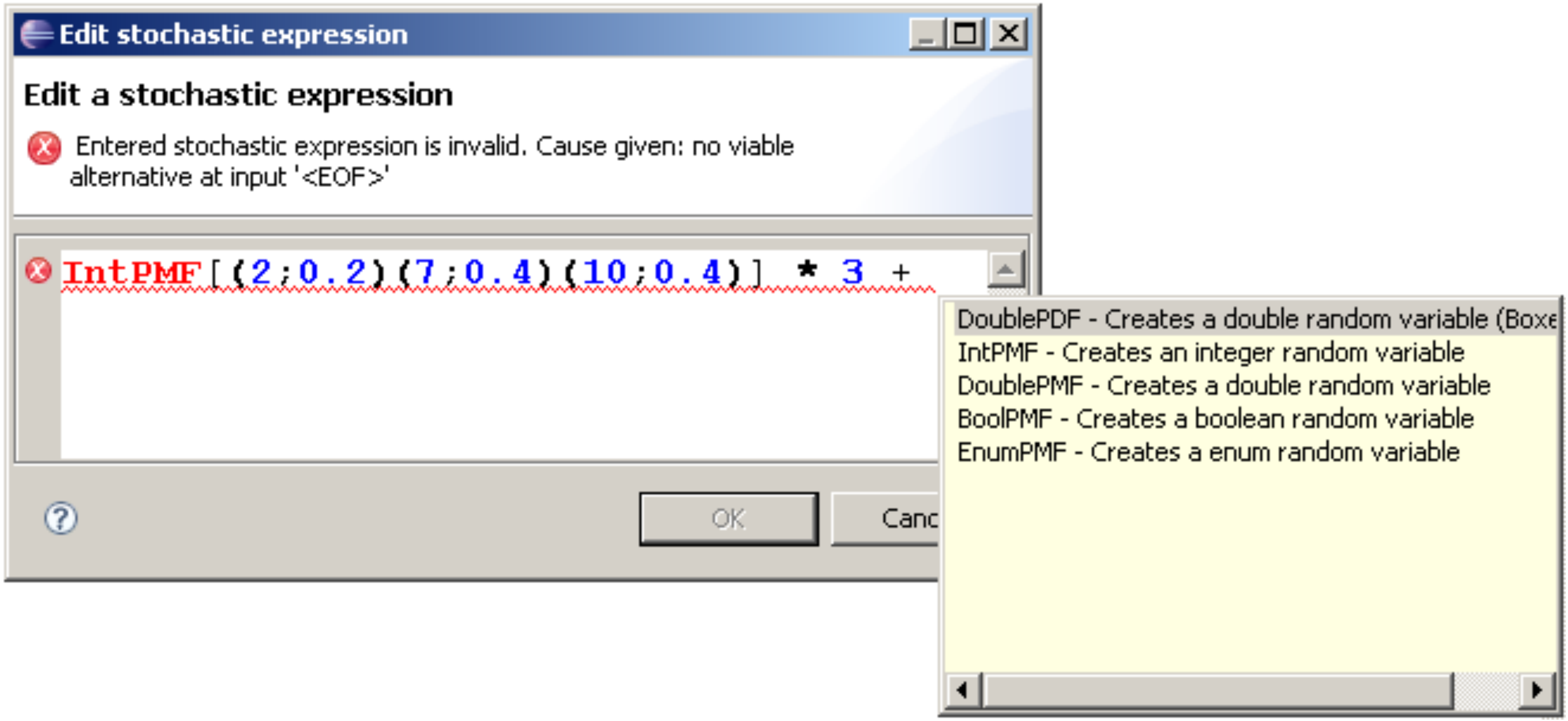
- Easy to specify
- Tool support to prevent mistakes
- Easy access from model transformations

State of the Art?

- MARTE
  - Extensive & complex standard
  - Parsers not well standardized
- Specialised languages
  - Commonly have precise syntax & semantics
  - Some accessible by model-transformations

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Screenshot
PCM StoEx language



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Transform

Generates analysis model

- (L)QN
- Markov Chain

Requirements?

- Should be fully automated and easily accessible by software architects
  - Again, a question for empirical software engineering
  - In practice experiments not done often enough!
- Full automation implies all information is in the model

Development?

- Needs expert both for software models and analysis theory
- Additionally, expertise in MDD technologies required

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Solve

Automated step to generate analysis results

- Throughput
- PROFOD

Usually done by running external solvers like

- PRISM
- Java Modelling Tools (QN tool)
- SimQPN (Petri Net based)
- CSP Checker (Process algebra)
- …

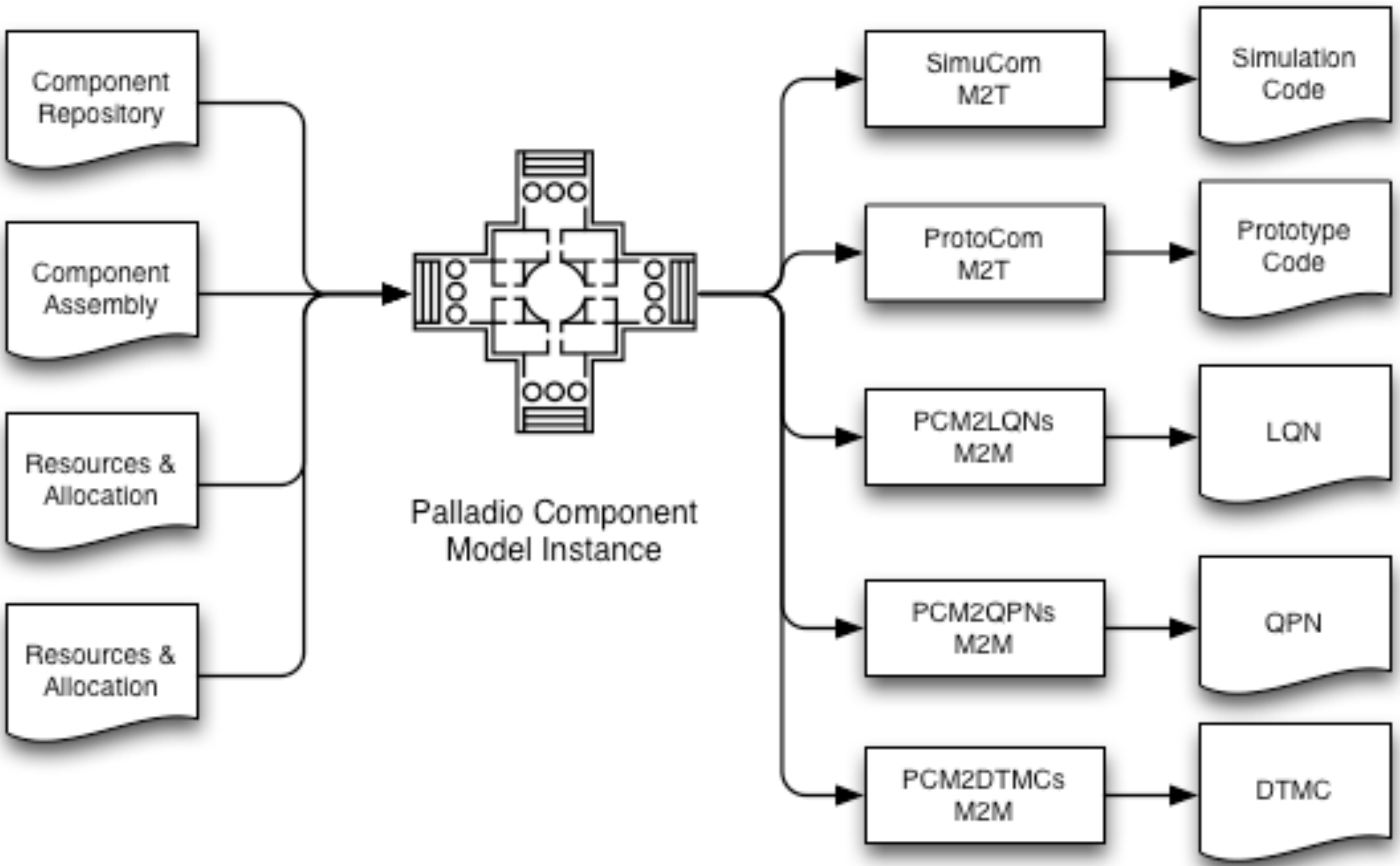
Issues?

- Typical issue: Models and tools do not scale for arbitrary inputs
- → Transformations often need to abstract!

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

PCM:
Example transformations & solvers…

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Why so many solvers…?

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Disclaimer: This is joint work

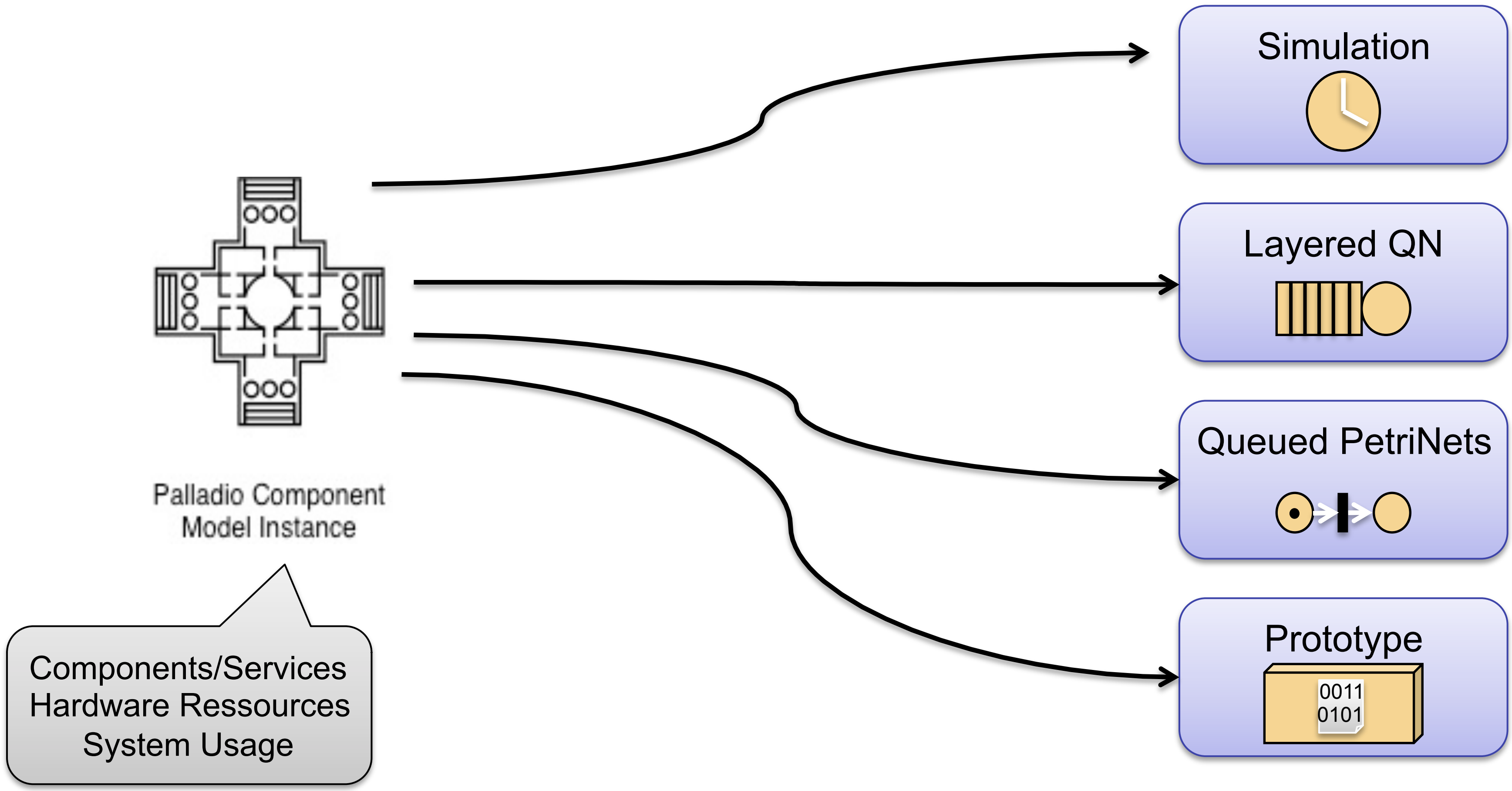
- Presentation of the results of a series of experiments
- Performed over 1 year duration
- Submitted to Transactions of Software Engineering (TSE), currently under review
- Results of joint work
  - Karlsruhe Institute of Technology
  - University of Zurich
  - University of Paderborn
  - Gameforge

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Experiment Design: Research Questions

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

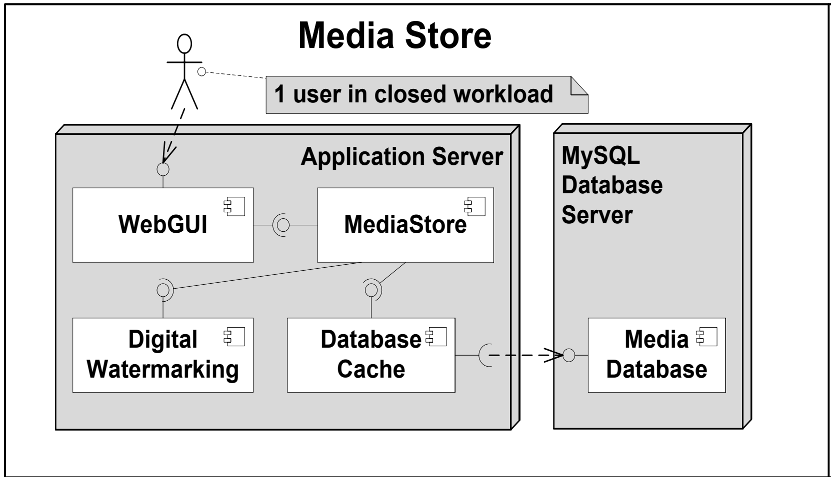
R2: What is the runtime of each solver?



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

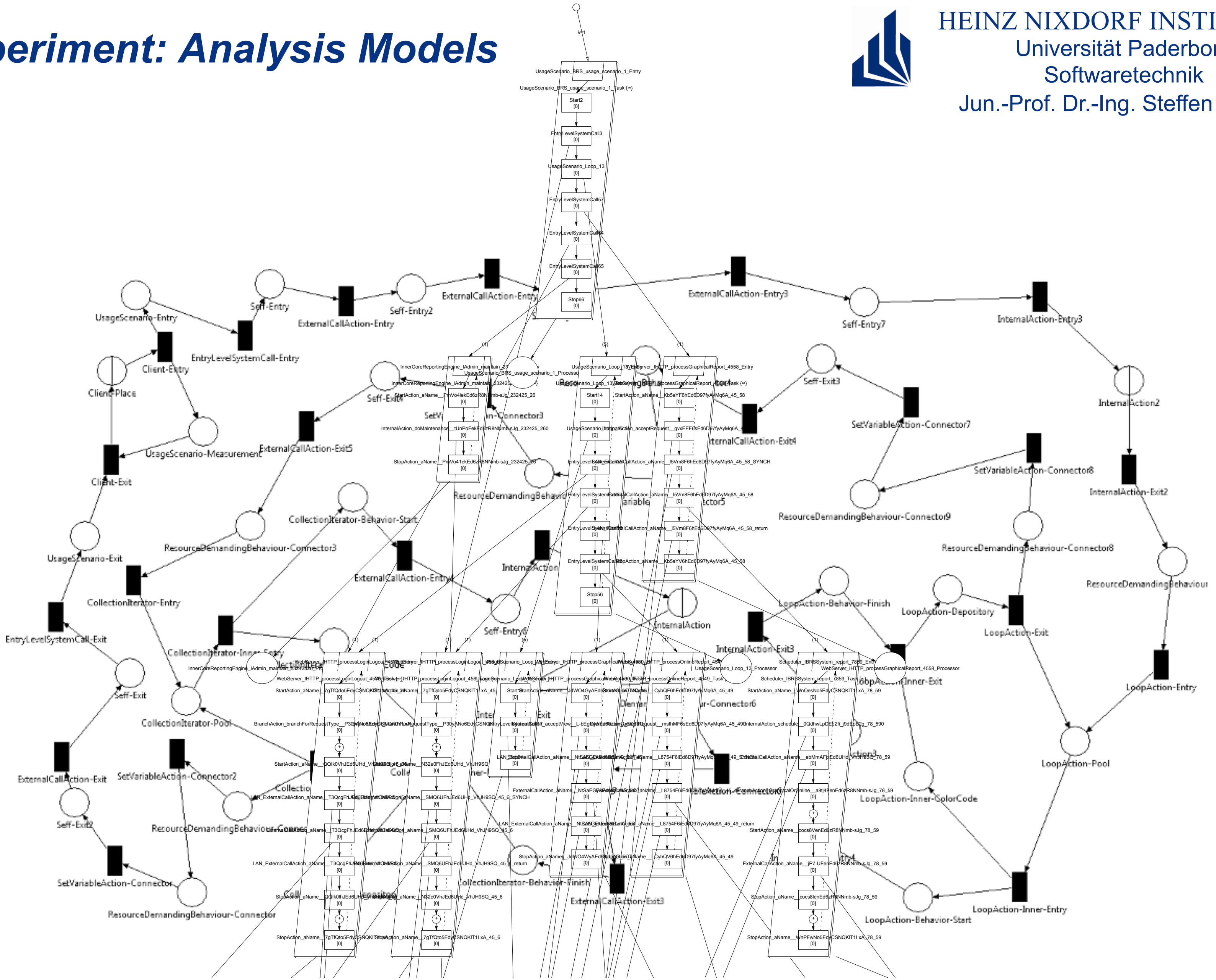
Experiment: Test Models

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker



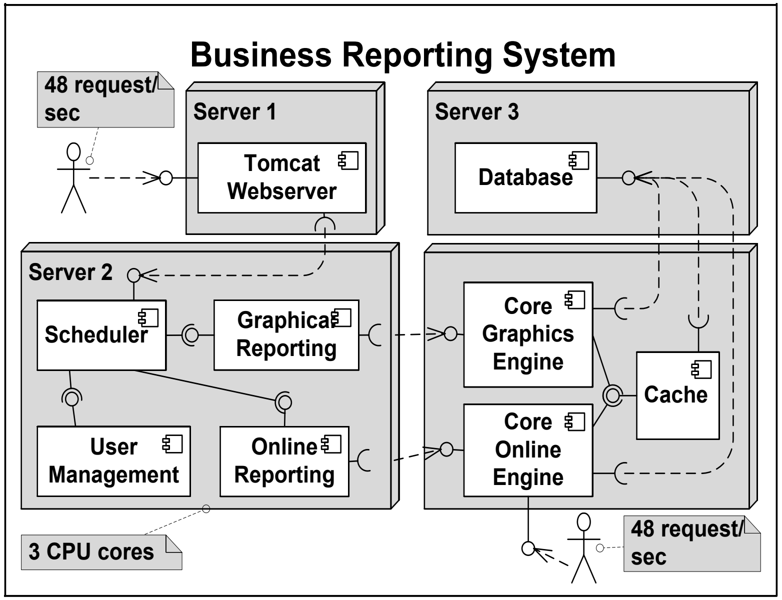
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Experiment: Analysis Models



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

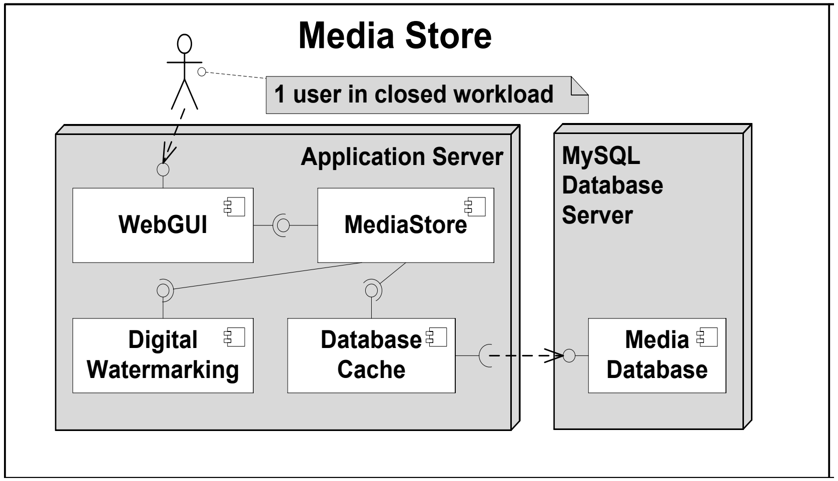
Experiment: Results excerpt



Solver	Mean RT pred.	Diff in %	Solver RT
Simulation	5,22 s	N/A	591,5 s
SimQPN	5,92 s	13,3%	74,5 s
LQN	7,39 s	41,5%	4,0 s
Prototype	6,53 s	25,0%	972,9 s

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Experiment: Results excerpt



Solver	Mean RT pred.	Diff in %	Solver RT
Simulation	1,33 s	N/A	31,4 s
SimQPN	1,33 s	0%	6,4 s
LQN	1,28 s	-3,3%	1,3 s
Prototype	1,02 s	-22,8%	1034,7 s

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Interpretation

Performing role: Skilled analysts, …

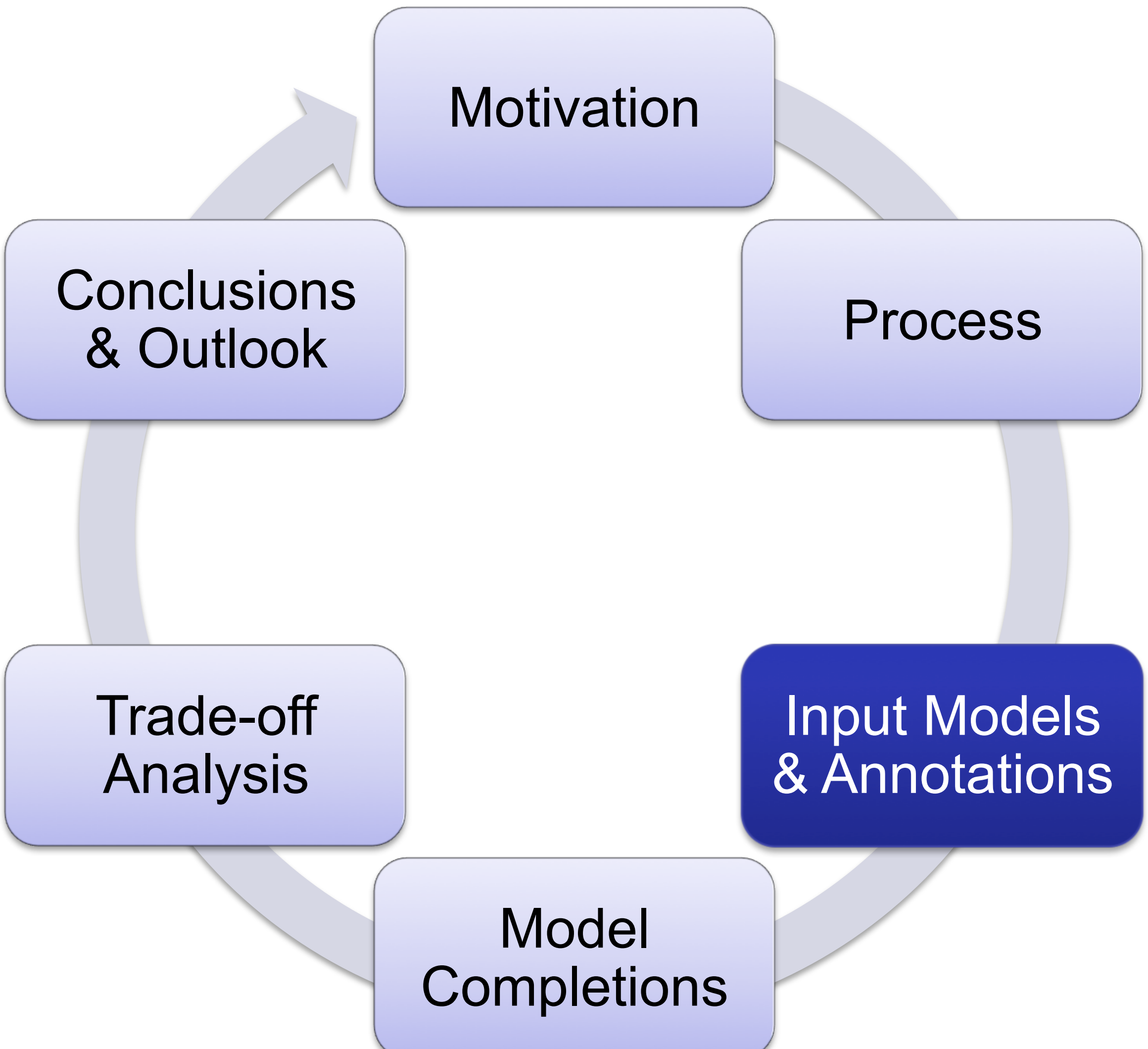
- Take analysis model results
- Translate them into statements about the SuS
- Suggest improved designs

Example:

- Analysis model: Service failure probability > 10%
- Software model: Analyse single failure probabilities
- Countermeasures
  - Introduce redundancy
  - Replication
  - Voting algorithms
  - …

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

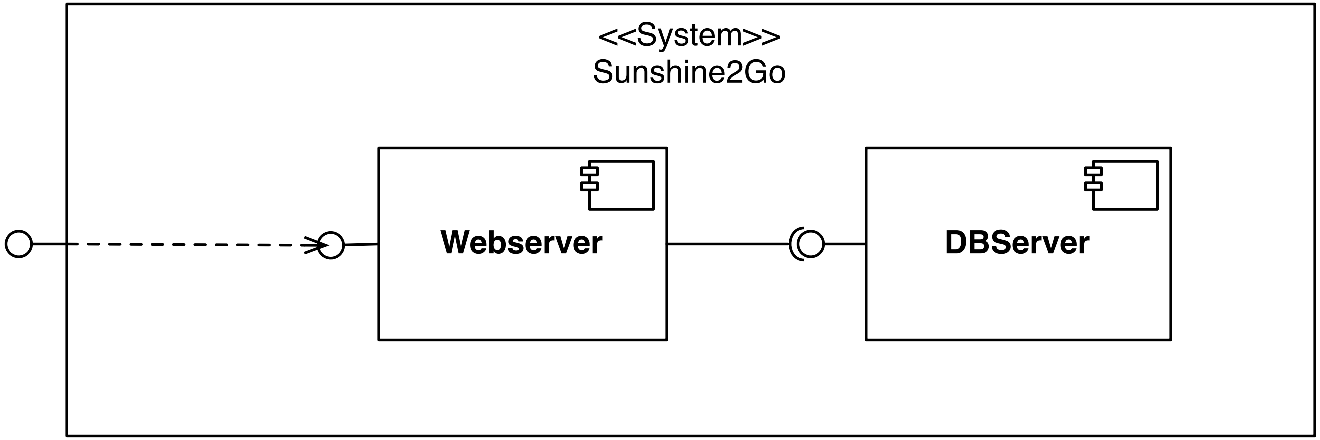
HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

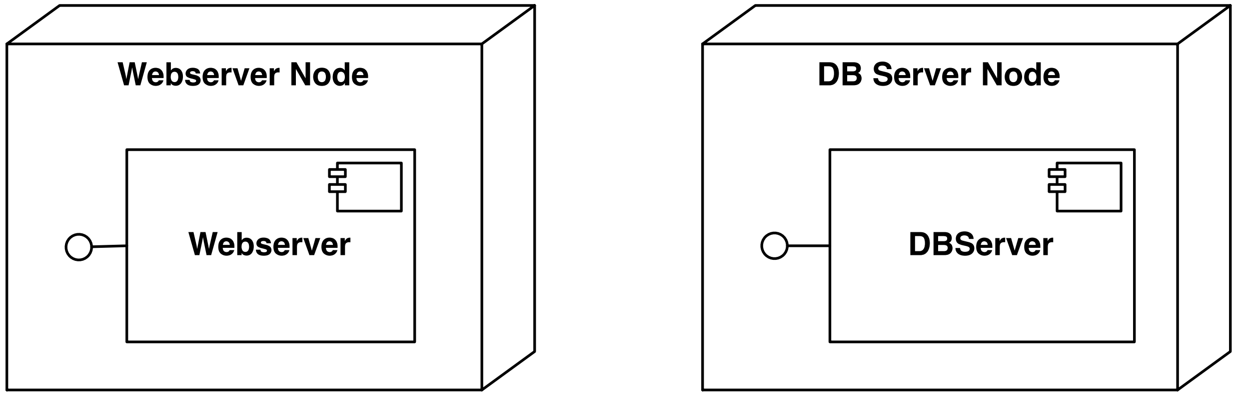
- Dynamic web server example
  - Dynamic web service for querying the weather forecast of the next 7 days “Sunshine2Go”
  - Reports weather conditions of next 7 days for given location in the world



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Allocation of “Sunshine2Go”

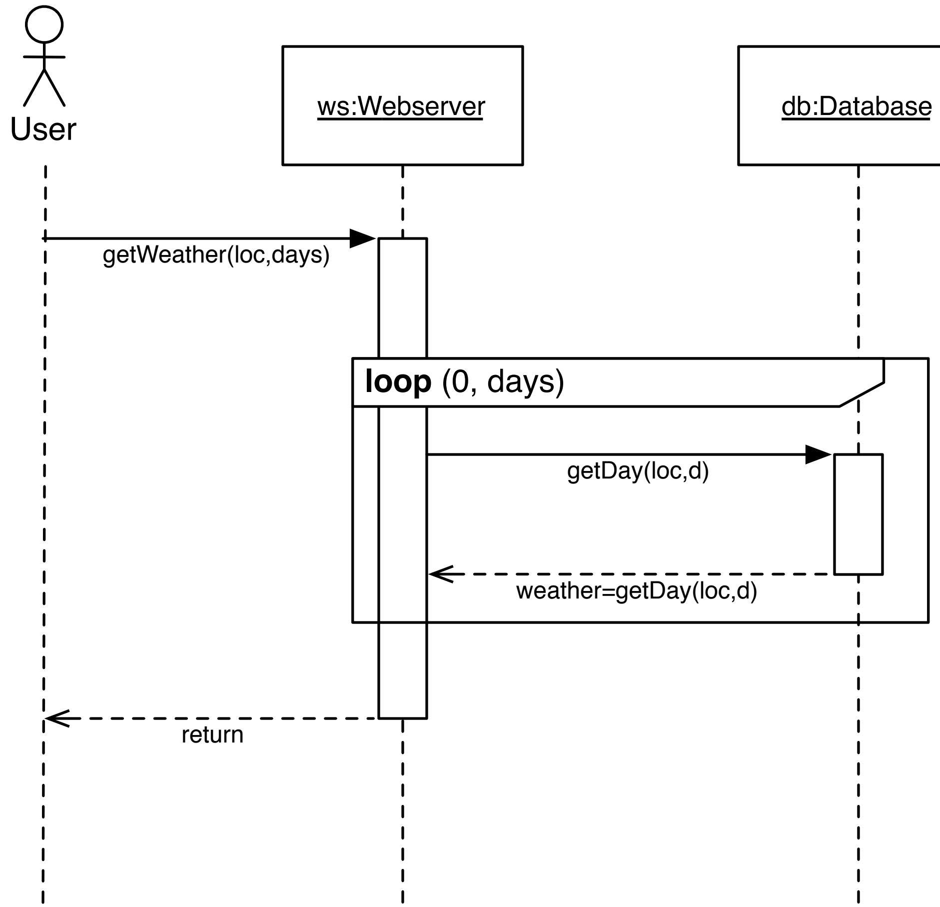
- Deployment is on local servers per component
- Web server runs the web server component, e.g., Apache
- Database server runs the DB, e.g., MySQL



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Main usage scenario

- Users query for the weather of a certain location and the amount of days in the future



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Requirements

- Initial use case
  - At the main business hours, users arrive with an mean interarrival time of 3 sec and exponential distribution (Poisson arrival process)
  - The mean response time of each call should not exceed 1sec
  - The utilisation of all resources should be below 80%
- Additional use cases are interesting, but skipped here for time reasons

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Measurements…

- Service time at webserver CPU is exponentially distributed with mean of 1WU
- At the database for each request the CPU has to perform 2WU and the HDD has to execute 1Read
- 1WU of Webserver CPU = 0.01 sec
- 1WU of DB Server CPU = 0.01 sec
- 1WU of DB HDD = 0.03 sec
- All CPU act approx. with processor sharing
- All HDDs approx. with FCFS scheduling
- Measurements indicate each step can fail with a probability of failure of 0.001

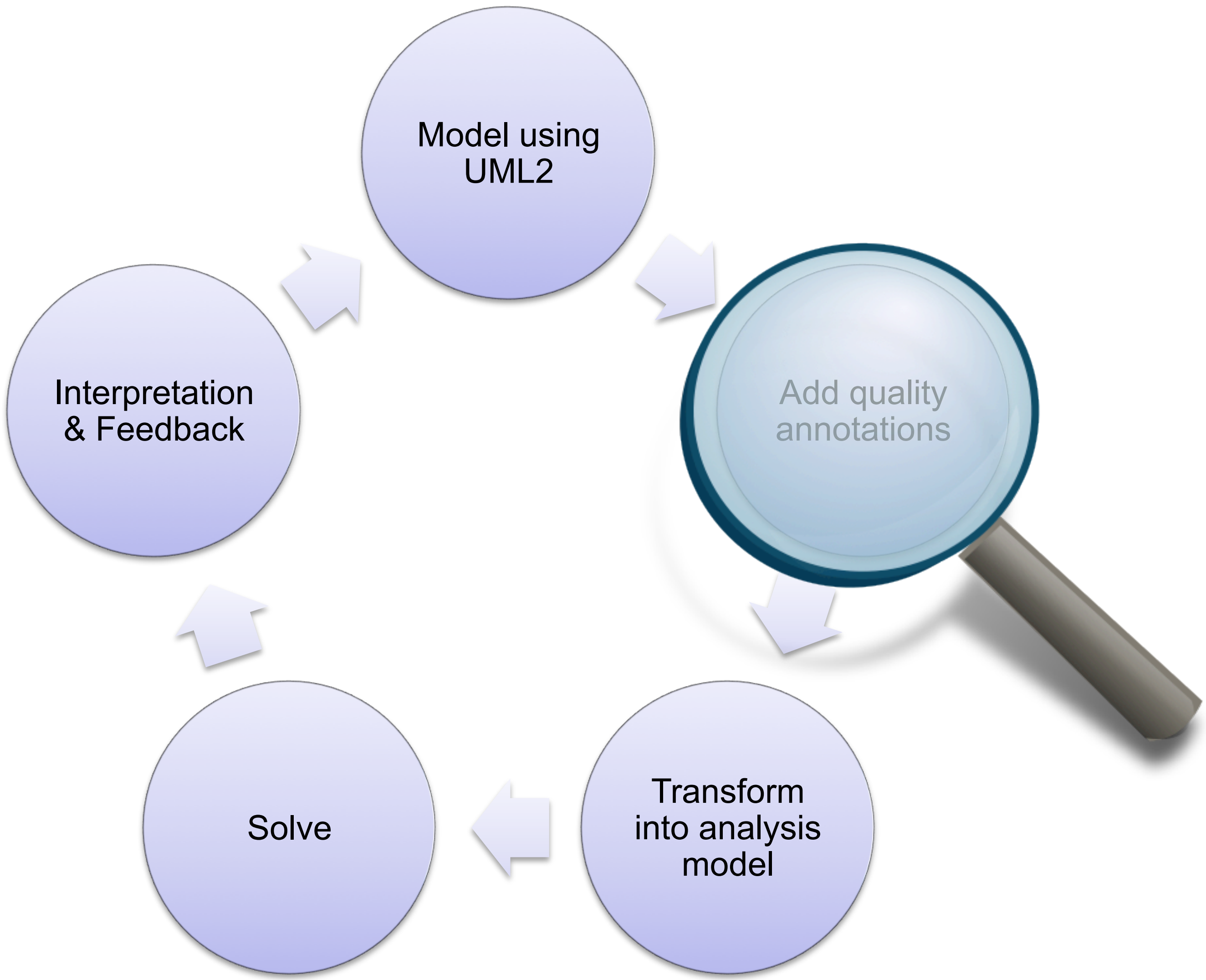
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Enhancing the model using UML MARTE

Using UML MARTE

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Process



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

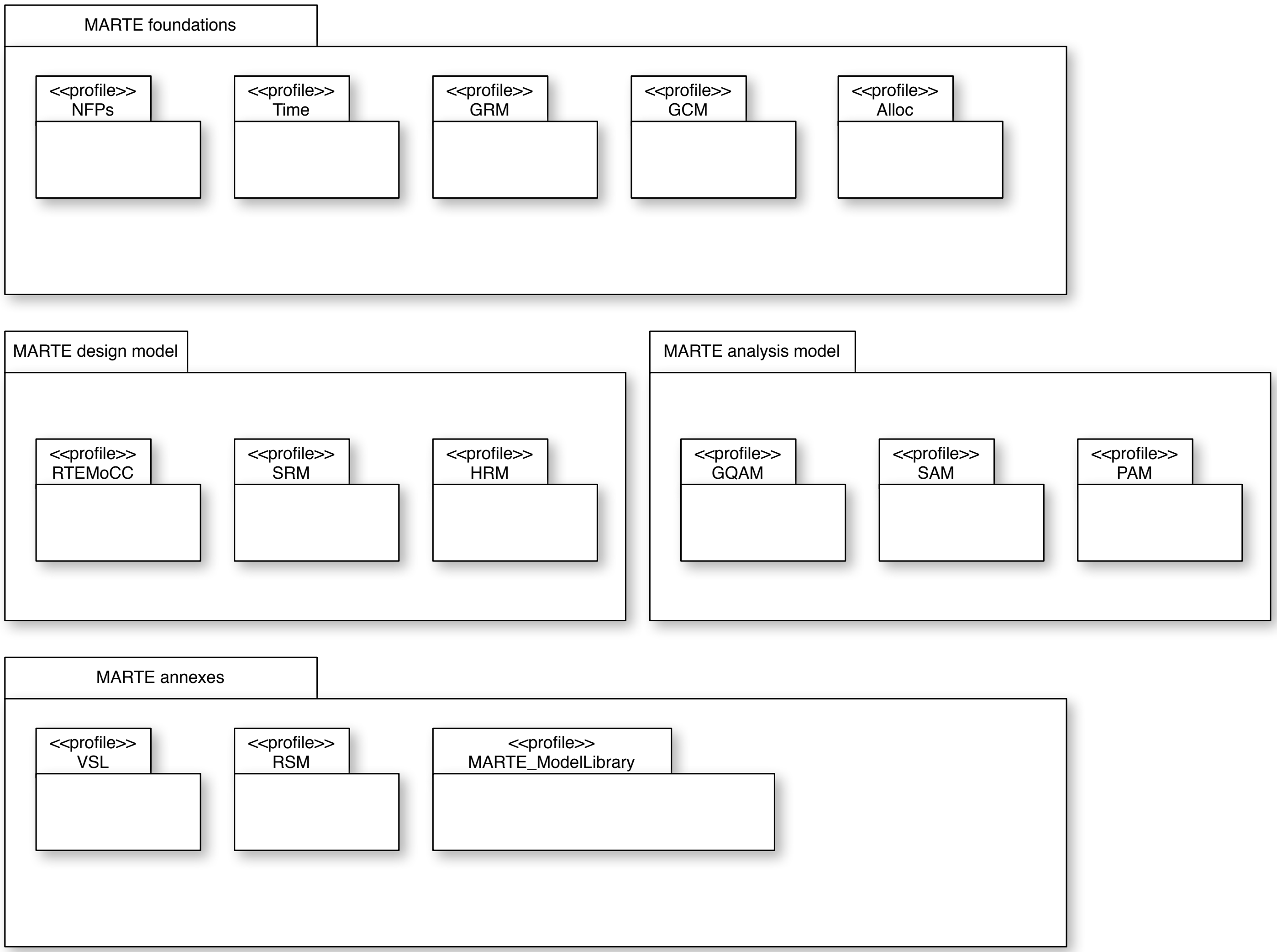
UML MARTE

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

- UML Profile for MARTE
  Modelling and Analysis of RT Embedded Systems
- Current Version: 1.0
- Released in 2009
- Maintained by OMG
- Successor of UML SPT which it is supposed to replace
- Uses UML QoS & FT to define quality characteristics
- Slight overlap with SysML, supposed to be aligned
- Extensive Standard, >700 pages
- Defines several profile packages

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

MARTE Layout

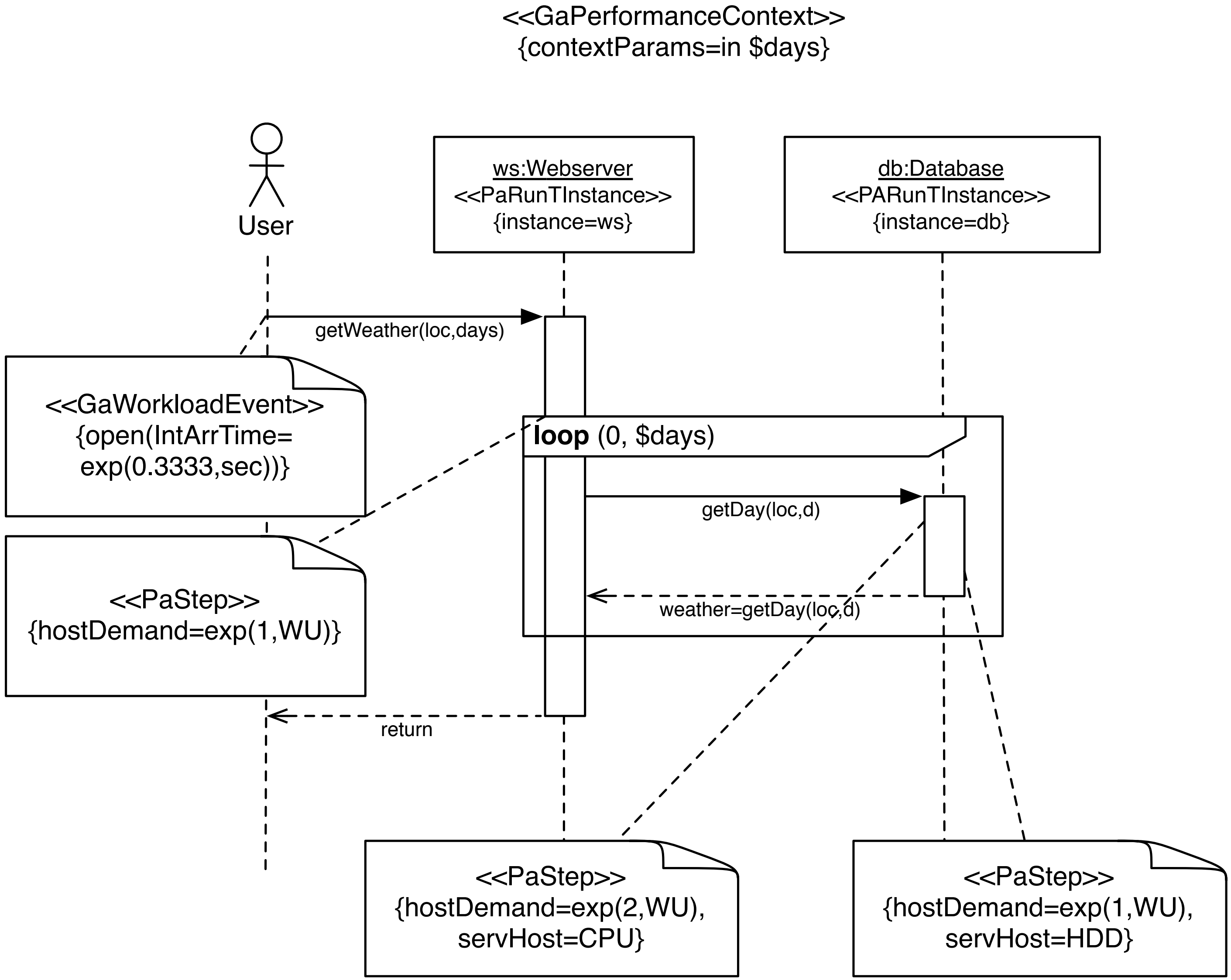


[http://www.omg.org/omgmarte/]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Main Usecase

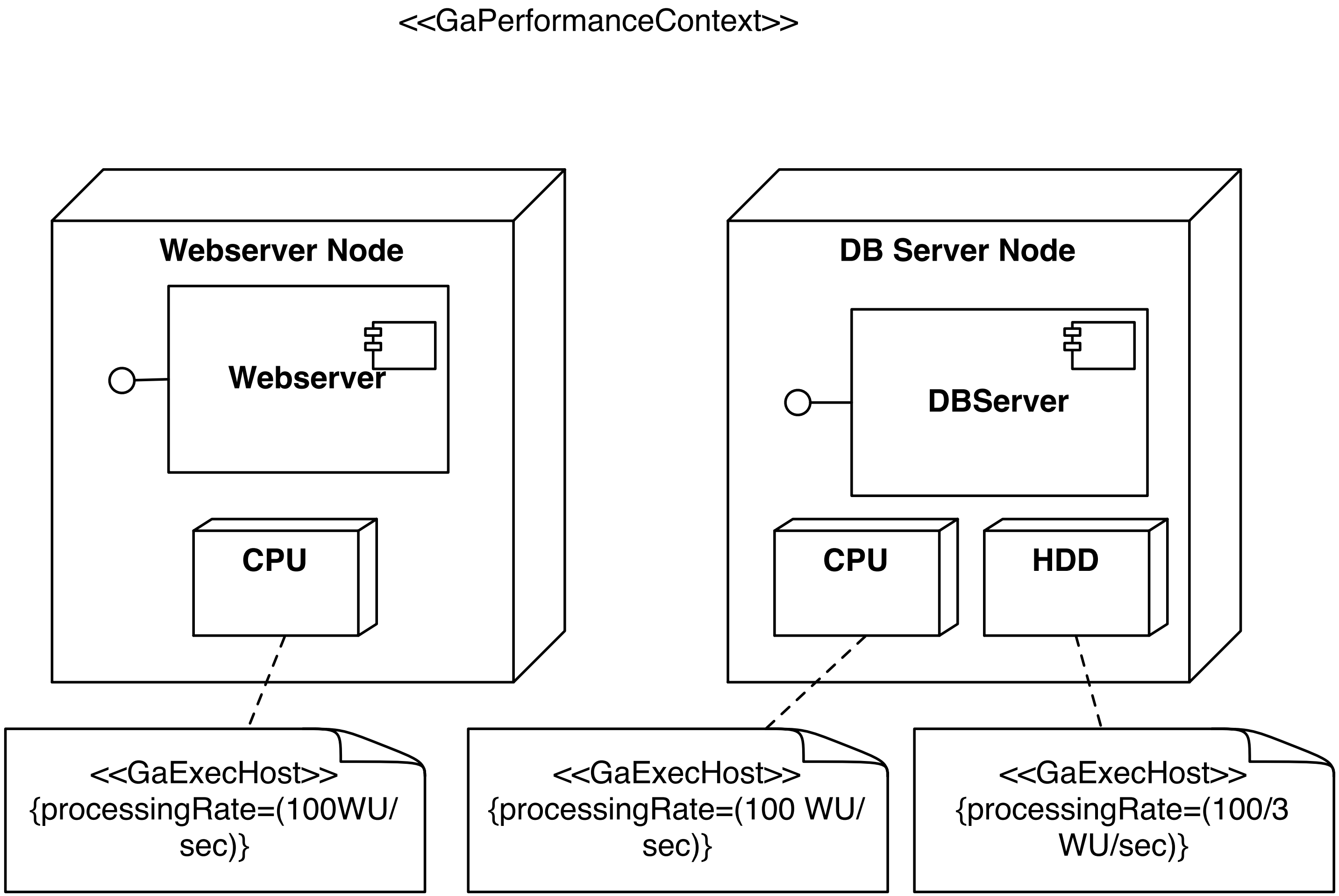
HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Annotated Deployment

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker



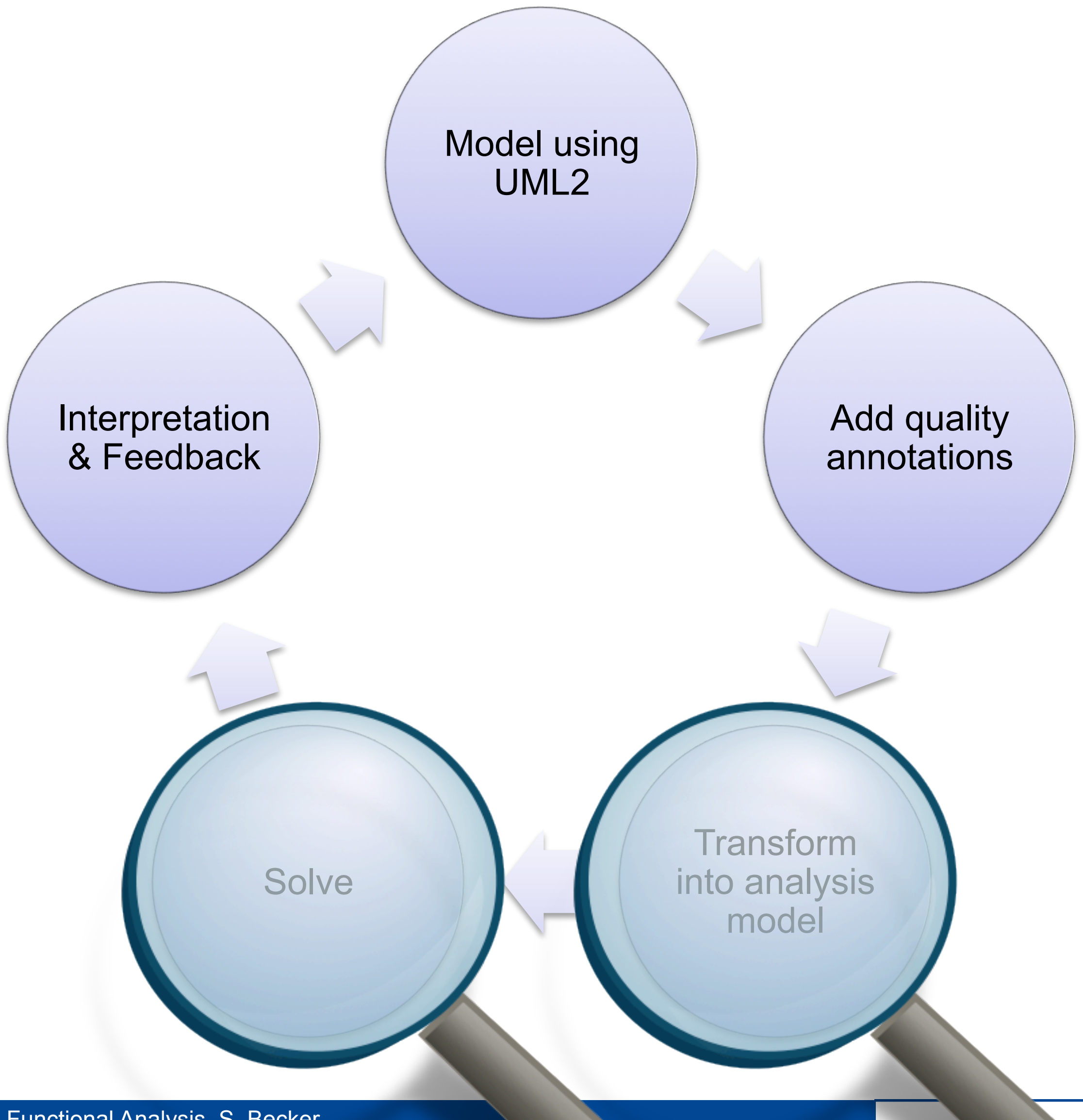
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Analysis Model

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Process

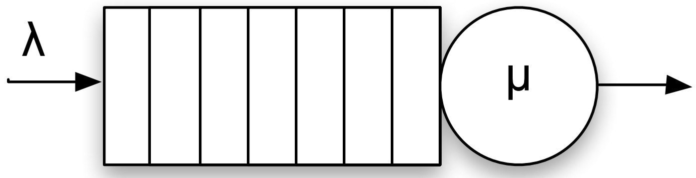
HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Queuing systems

- Mature analysis model for computer system performance
- Idea: Represent **active resources** as **waiting queues**
- Graphical notation:



E.g. queues for
  - CPUs
  - Disks
  - Network
- Reflects waiting times due to **contention**
- Underlying formalism is **Continuous Time Markov Chains (CTMCs)**

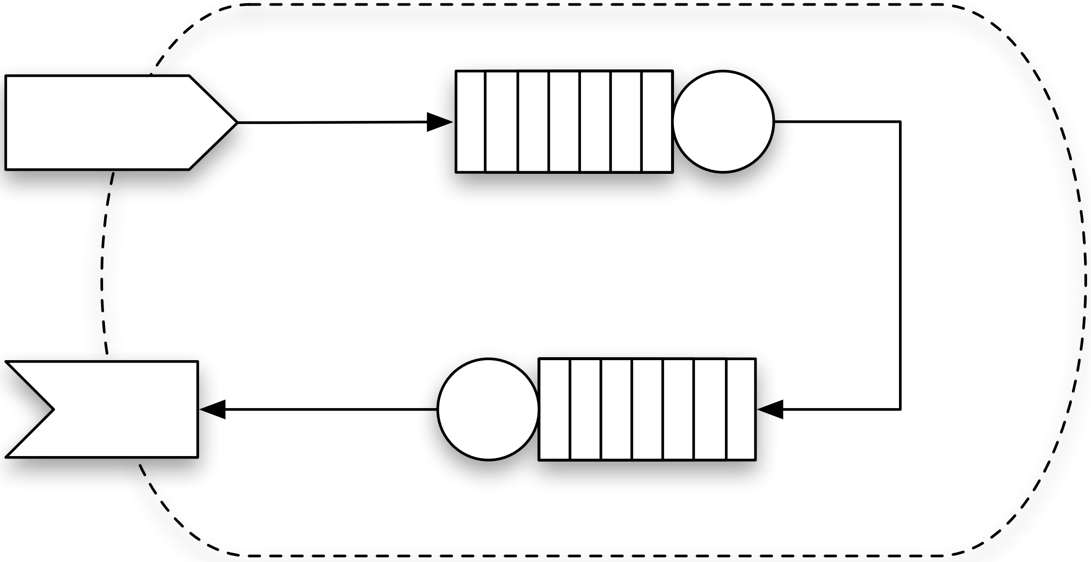
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Queuing network

- Queuing network
  - Directed graph
  - Nodes are service centre
  - Edges represent potential paths of jobs
  - Multiple job classes
    - Each having homogeneous service demands
    - Homogeneous routing probabilities
    - I.e., jobs in a class cannot be distinguished
- Widely used
  - Offer decent performance prediction accuracy
  - Good modelling paradigm and usability

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Open network

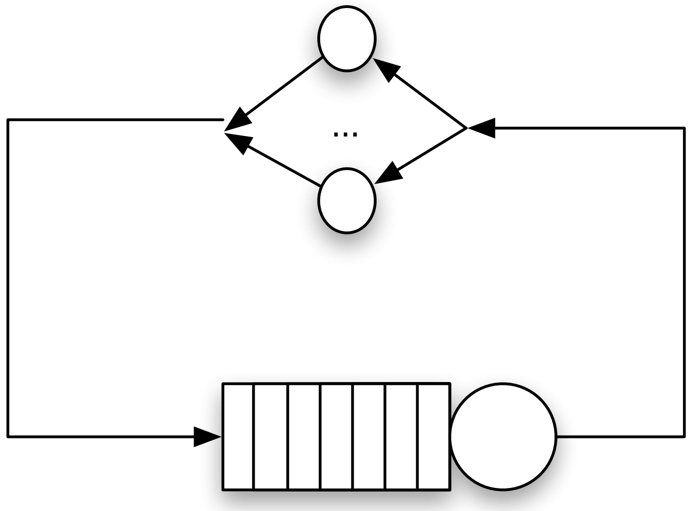


- Jobs enter net at source nodes
- Jobs leave net at sink nodes
- I.e., number of jobs not defined upfront
- Arrival process characterised by inter-arrival time
- Each queue is characterised in Kendall notation

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Closed network

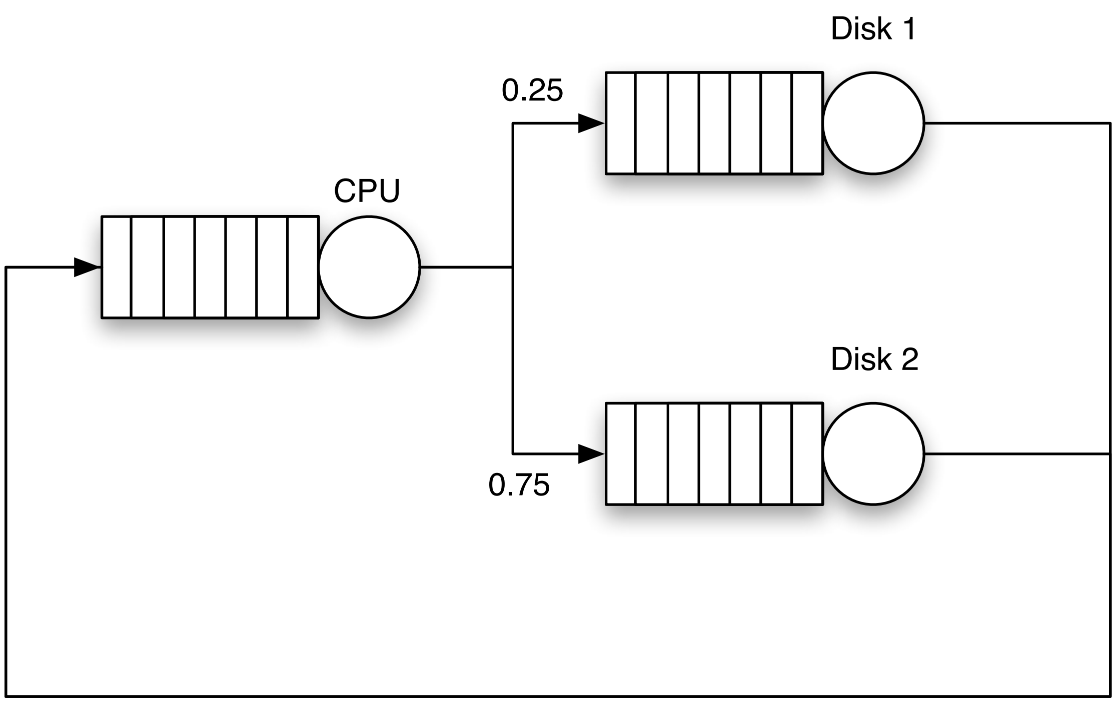
- Network of a fixed number of jobs
- Contains a set of terminals
- Jobs circulate in network
- Jobs delay in terminals for a given think-time interval
- Sometimes also modelled by IN and OUT ports which are connected in a loop



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Routing in QNs

- Jobs take probabilistic routes
- Route specified via transitions probabilities
- Given per class of jobs for multiclass networks



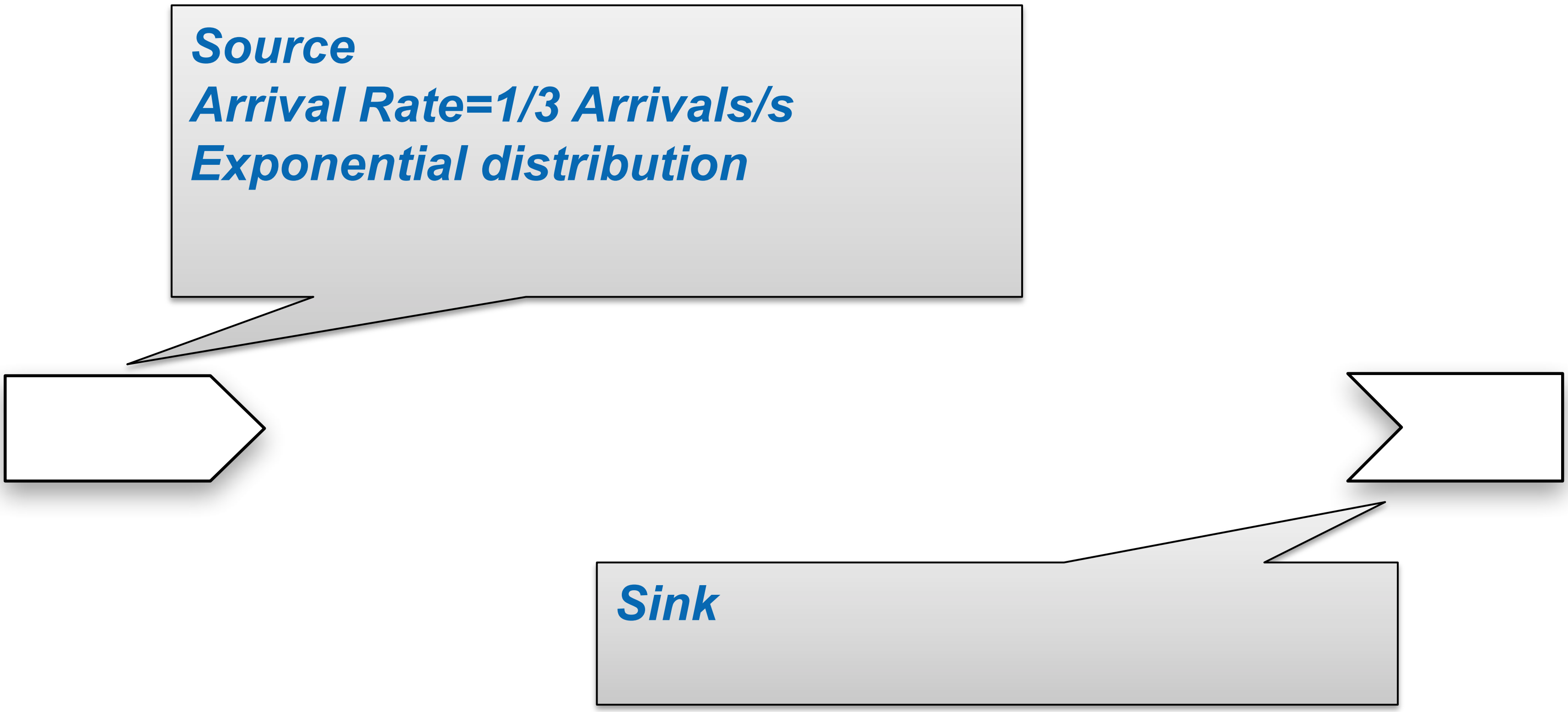
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

QN for our example…

Highlighting the core ideas of the transformation step

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

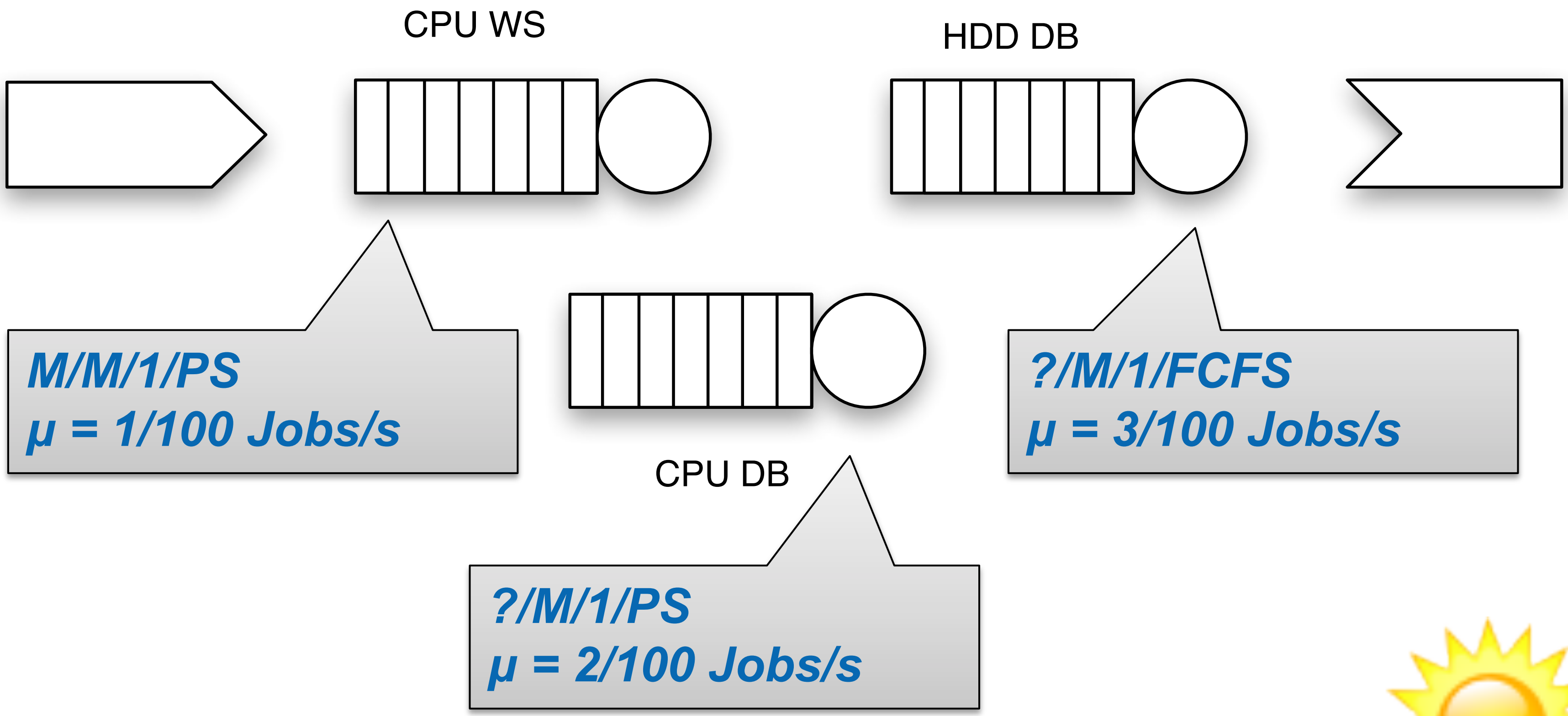
Performance: Derive QN



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Derive QN

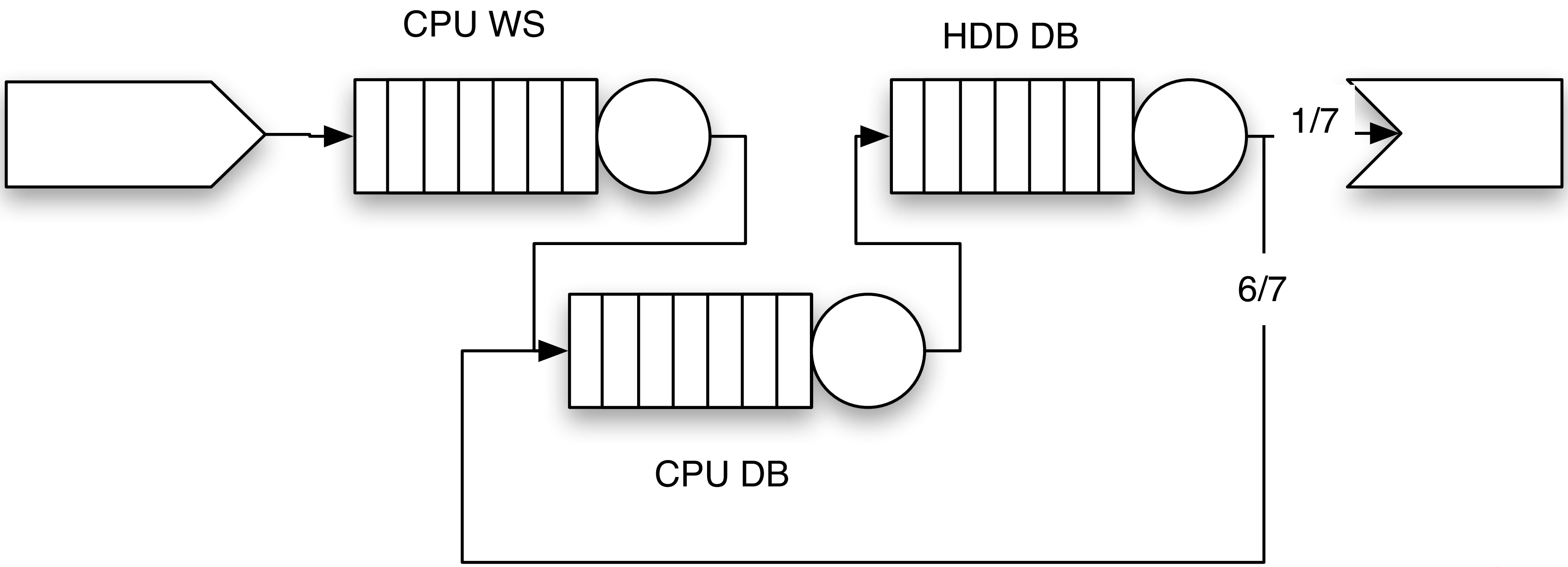
Add a queue per GaExecHost and their parameters



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Derive QN

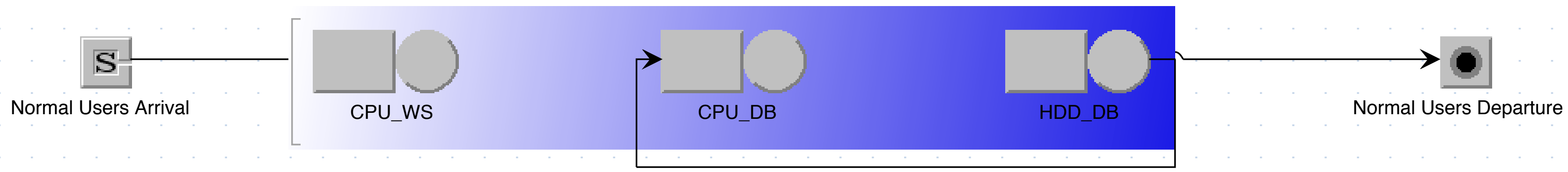
Add routes and their probabilities



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Put it in a Solver

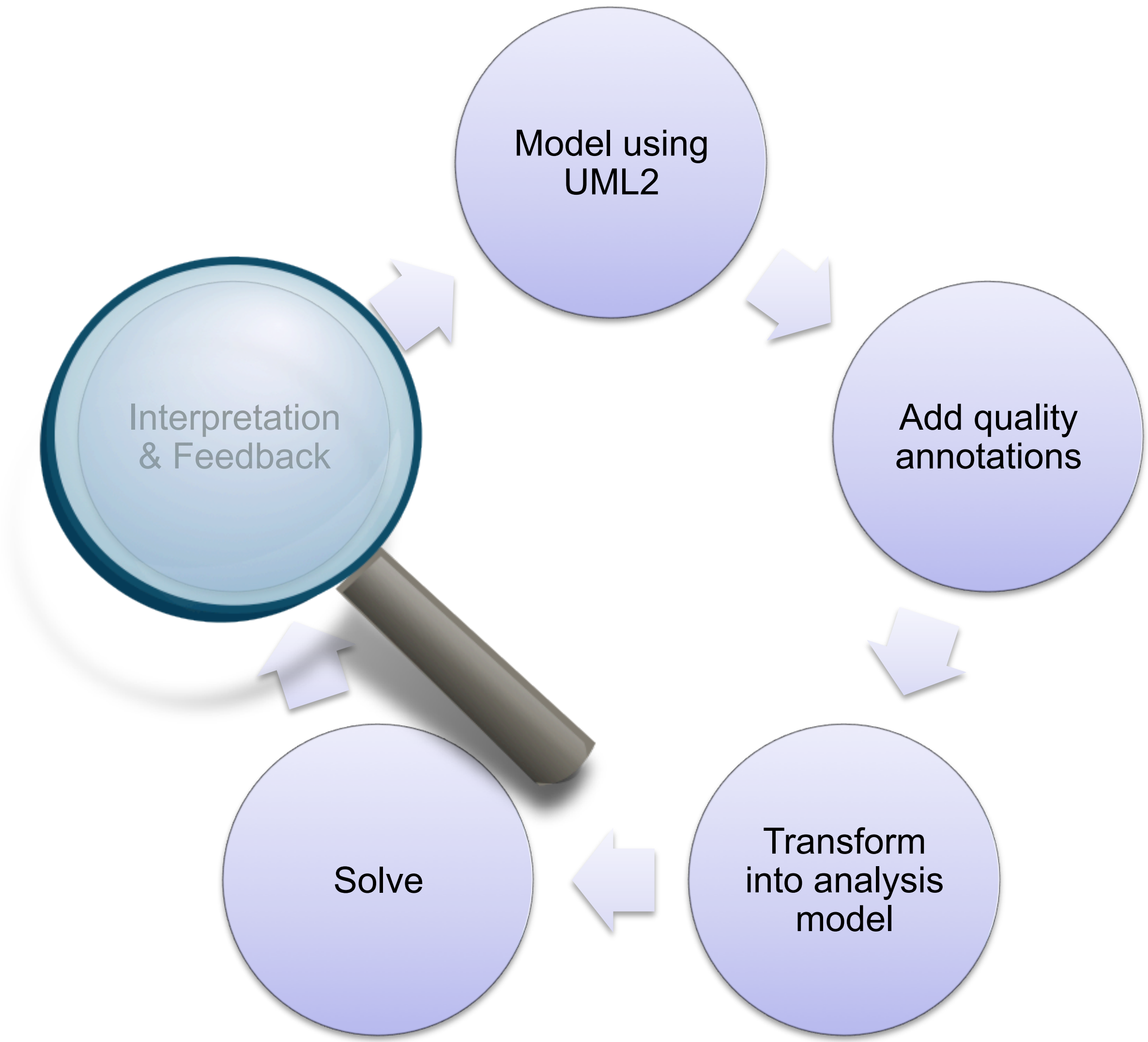
JSimGraph Model



- Results (according to jMVA)
  - Mean System RT = 0.5502 sec
  - Util CPU_WS = 0.33%
  - Util CPU_DB = 2.33 %
  - Util HDD_DB = 6.99 %

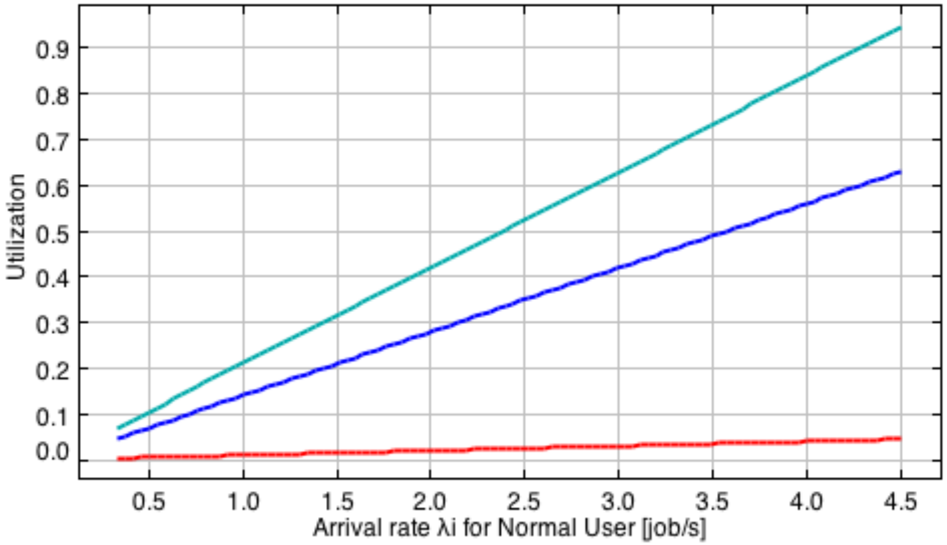
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Process



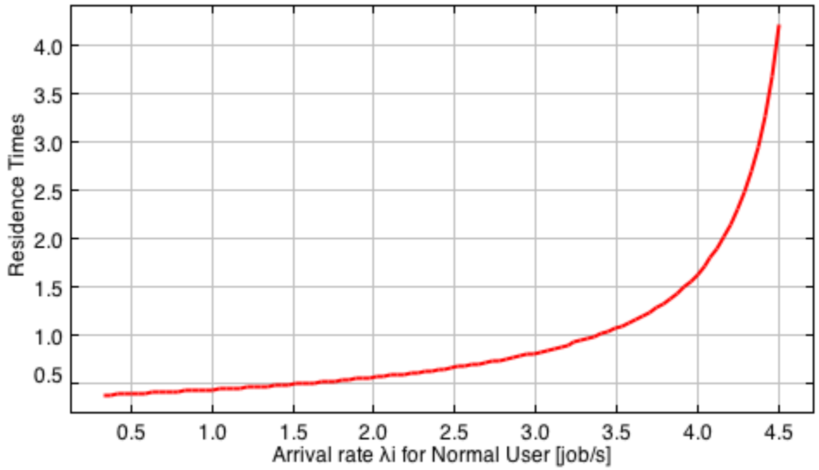
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

What-if analysis for arrival rate



- At arrival rate 4.5 the DB HDD (cyan) becomes the bottleneck
- The HDD queue starts to overload

- RT constraint from our requirements is violated at arrival rate of > 3.5 job/s
- Starting from a rate of 2.5 jobs/s the system shows overload effects



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

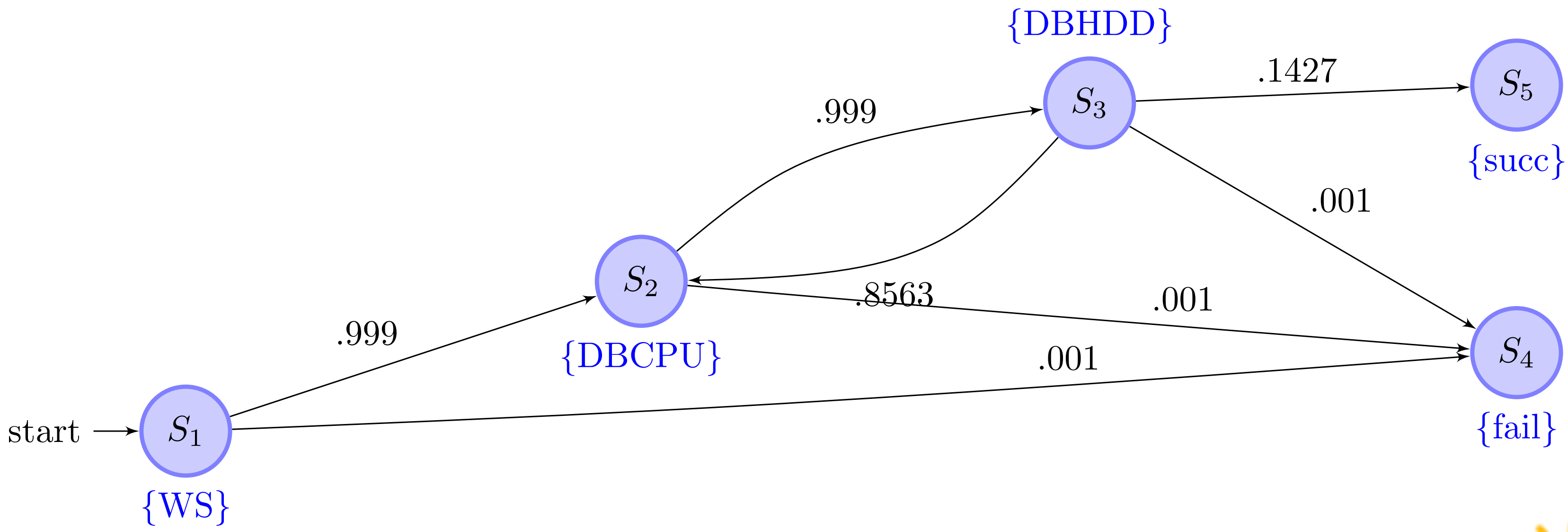
Reliability

- Common method is to create a Discrete Time Markov Chain (DTMC), i.e., disregard time
- Each step of the system becomes a state of the DTMC
- In each state, system can fail according to a failure probability. It reaches an absorbing failure state in case of failure
- Loop can be either unrolled or approximated by a geometric distribution
- PROFOD is probability of reaching the success state, i.e., the state after the last system action

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

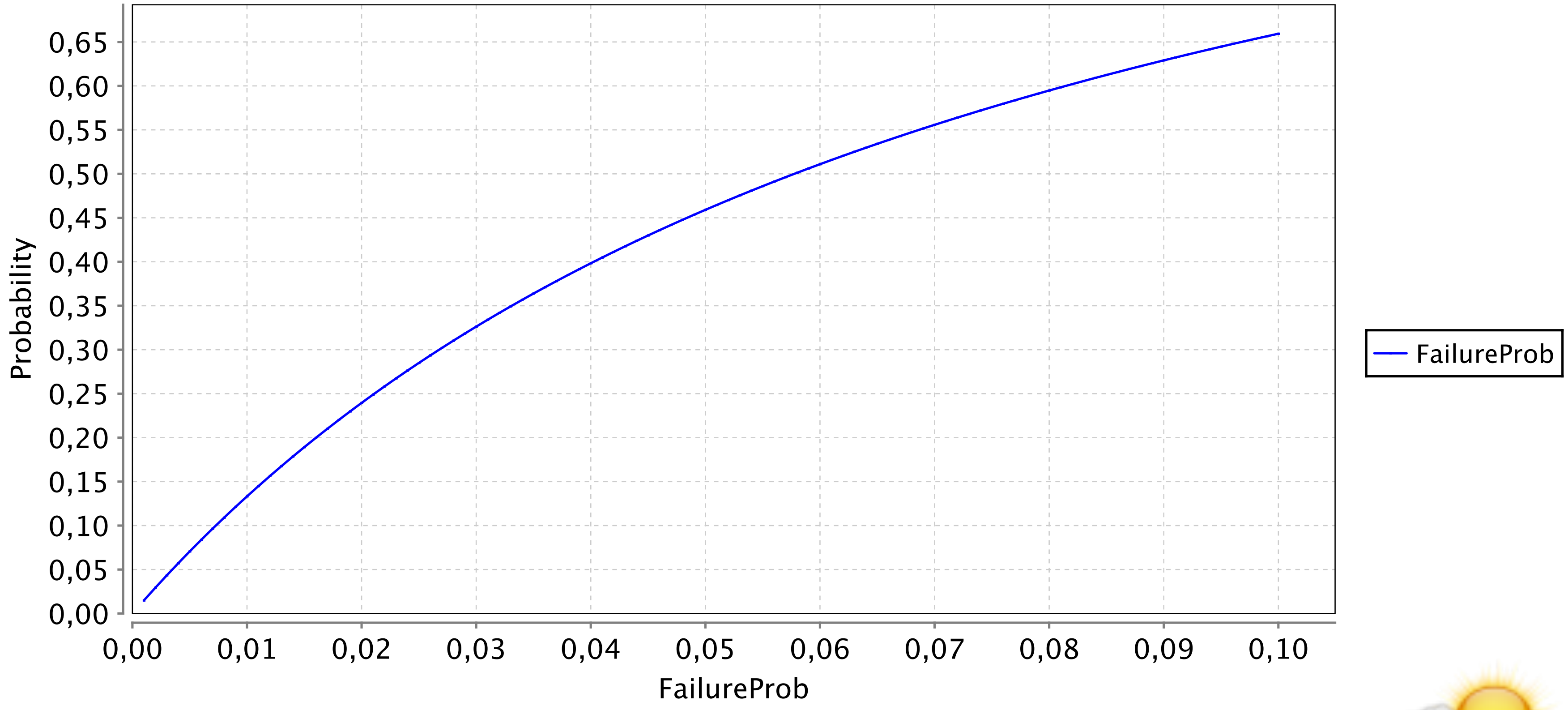
HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

- Create a DTMC for our SuS. Here approx. of loop counts
- PROFOD approx. 1,48% according to PRISM



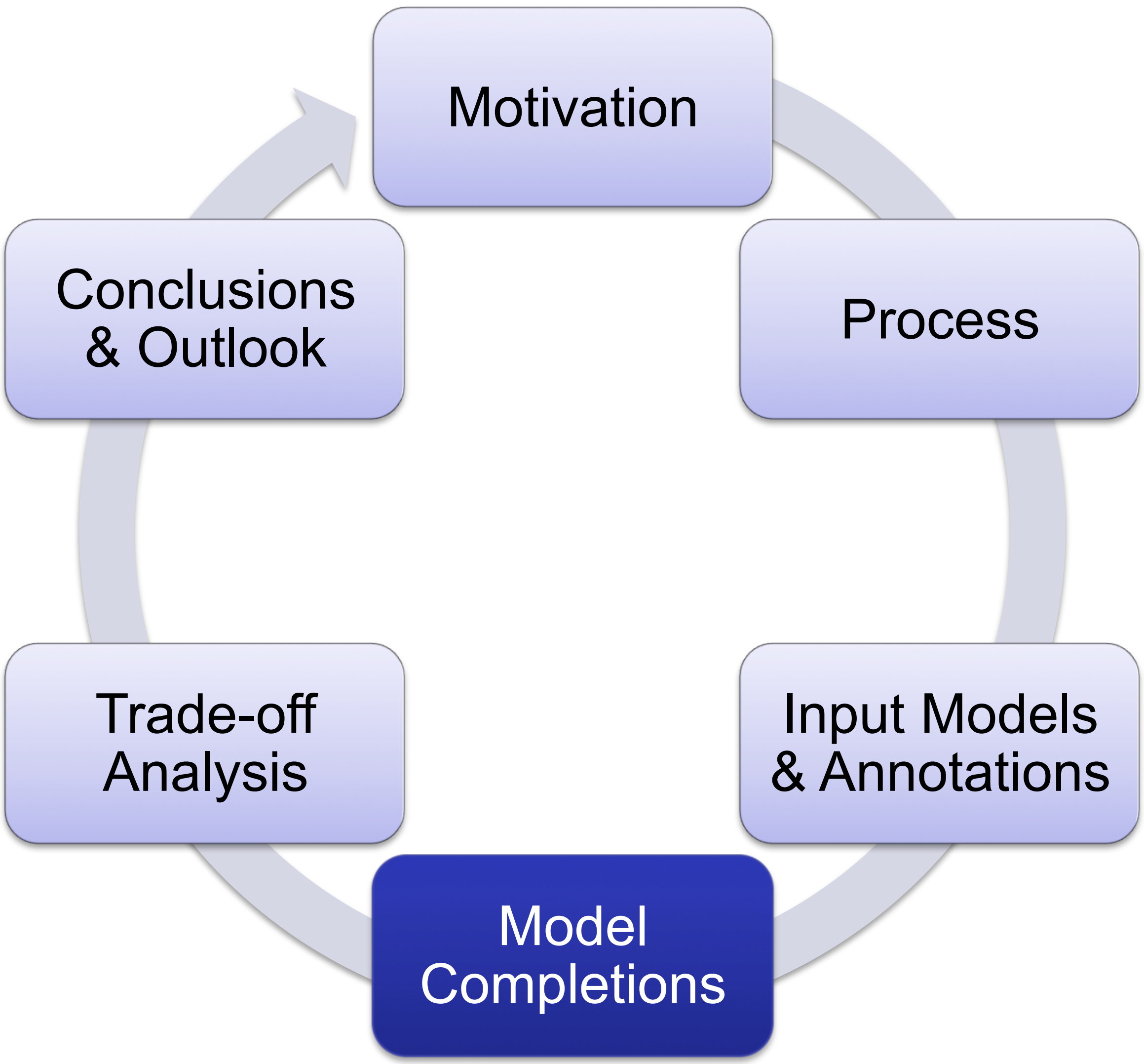
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

What if analysis



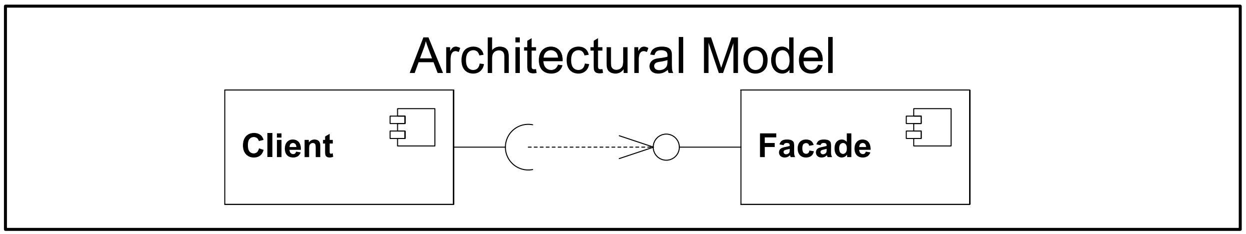
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Outline



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

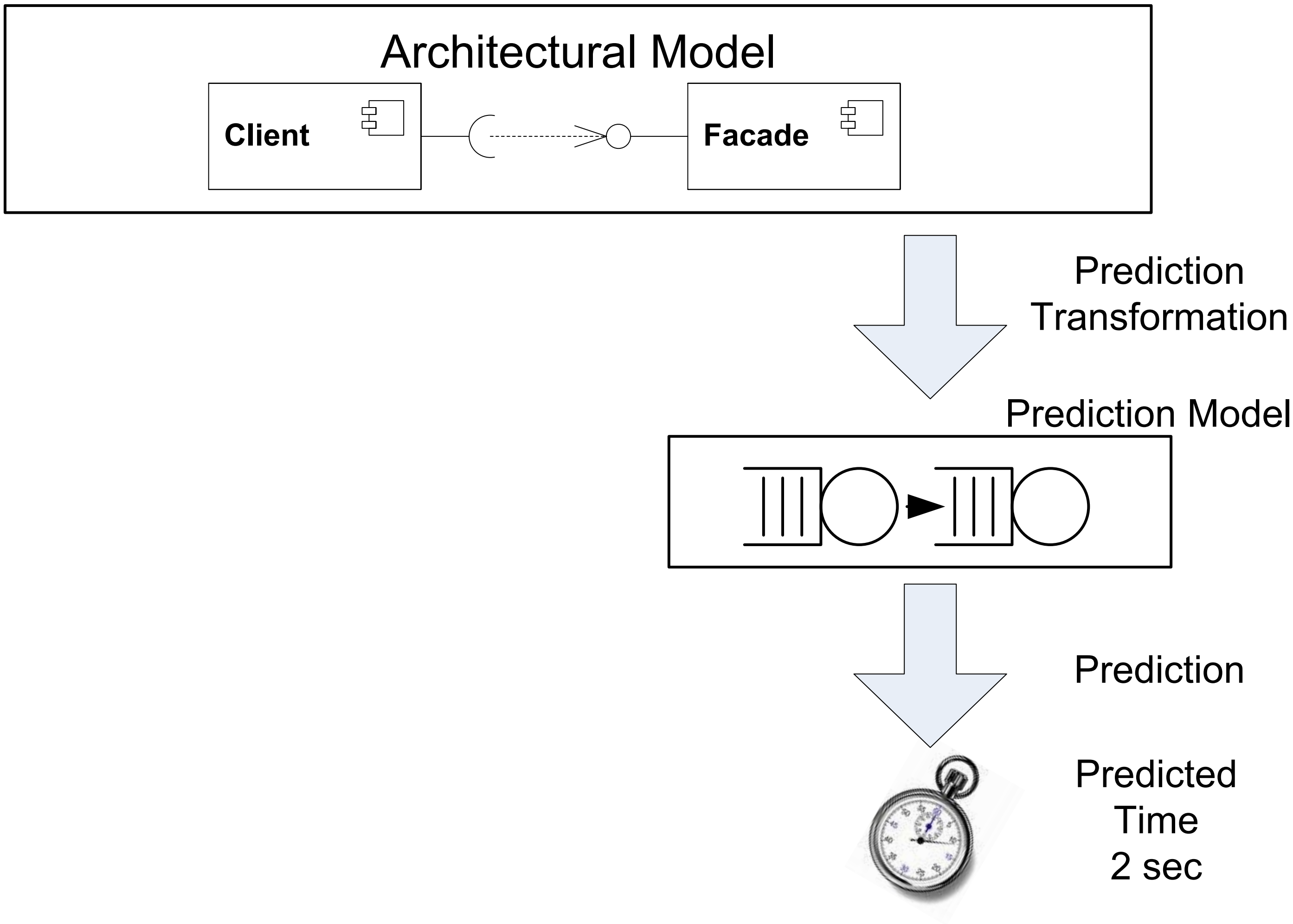
Motivation



[Becker, WOSP, 2008]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Motivation

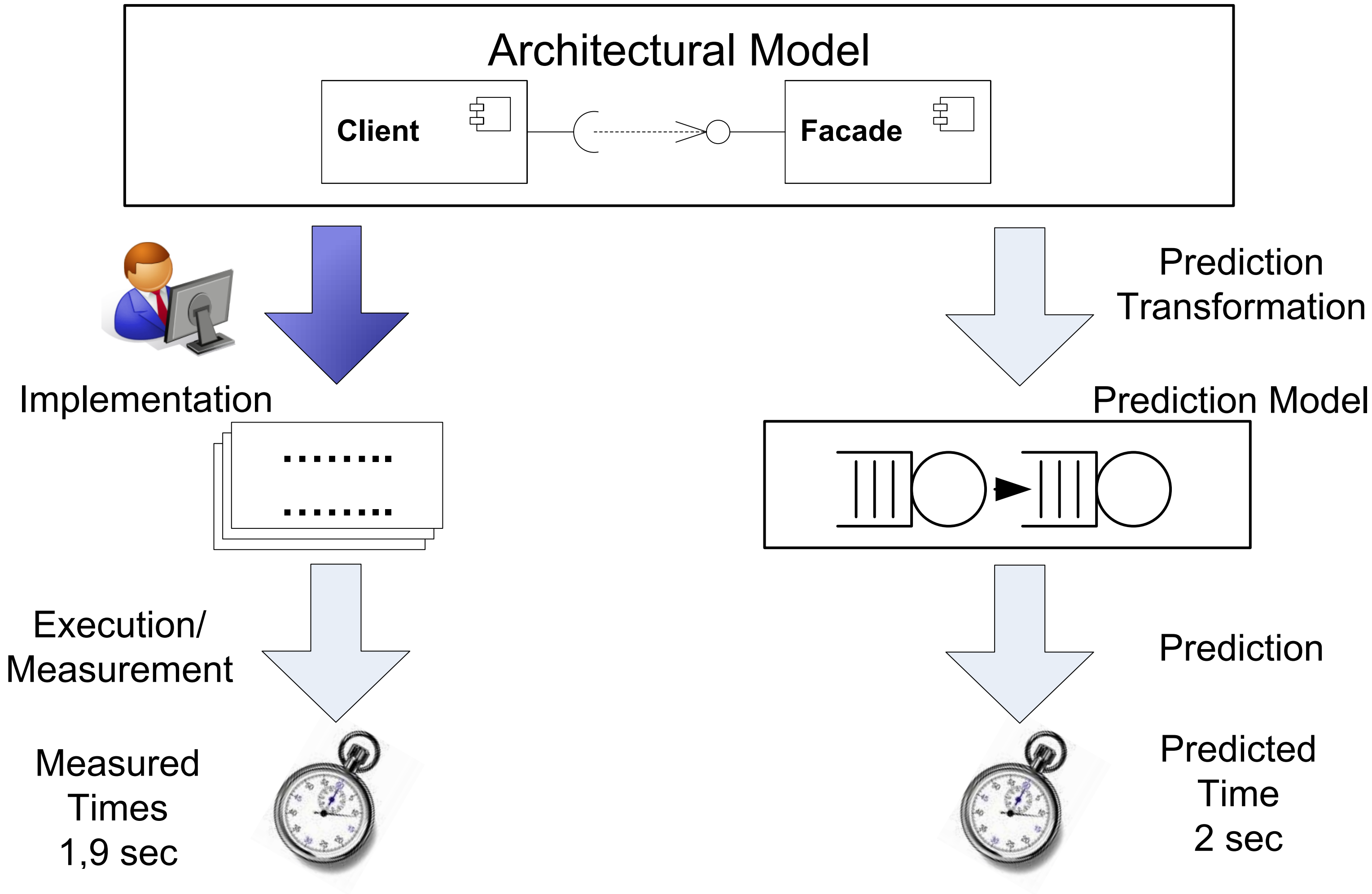


[Becker, WOSP, 2008]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Motivation

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

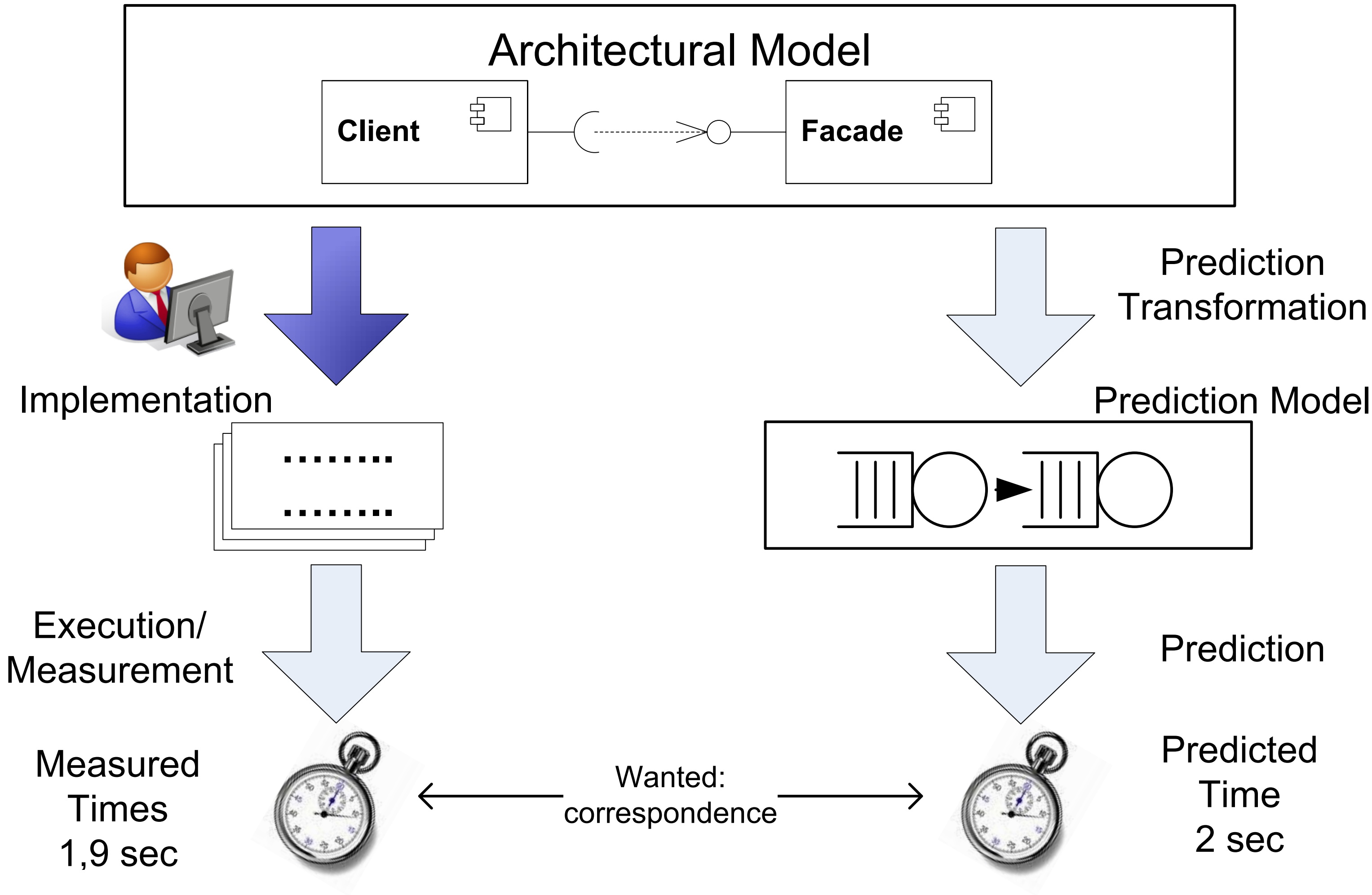


[Becker, WOSP, 2008]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Motivation

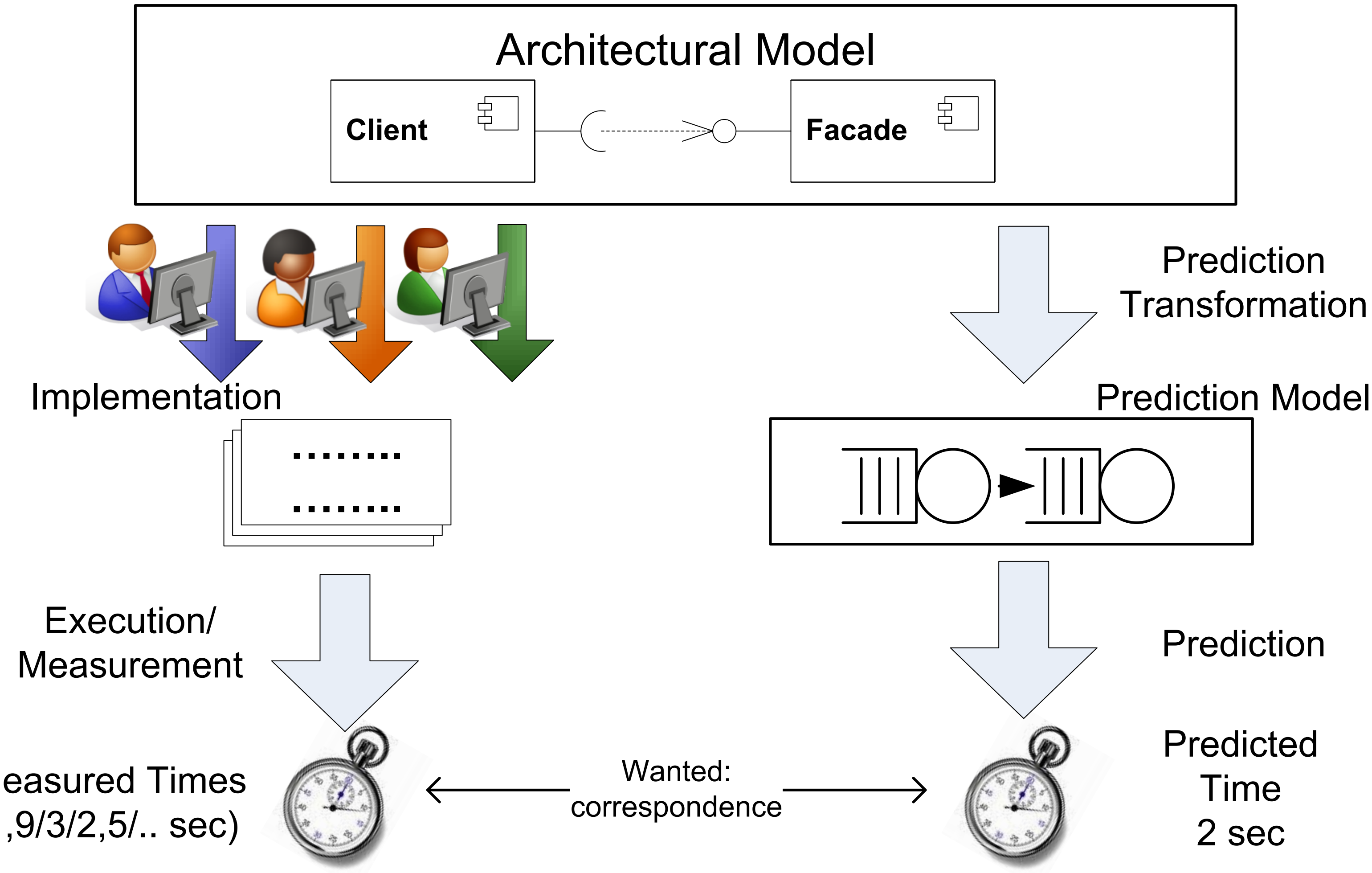
HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker



[Becker, WOSP, 2008]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

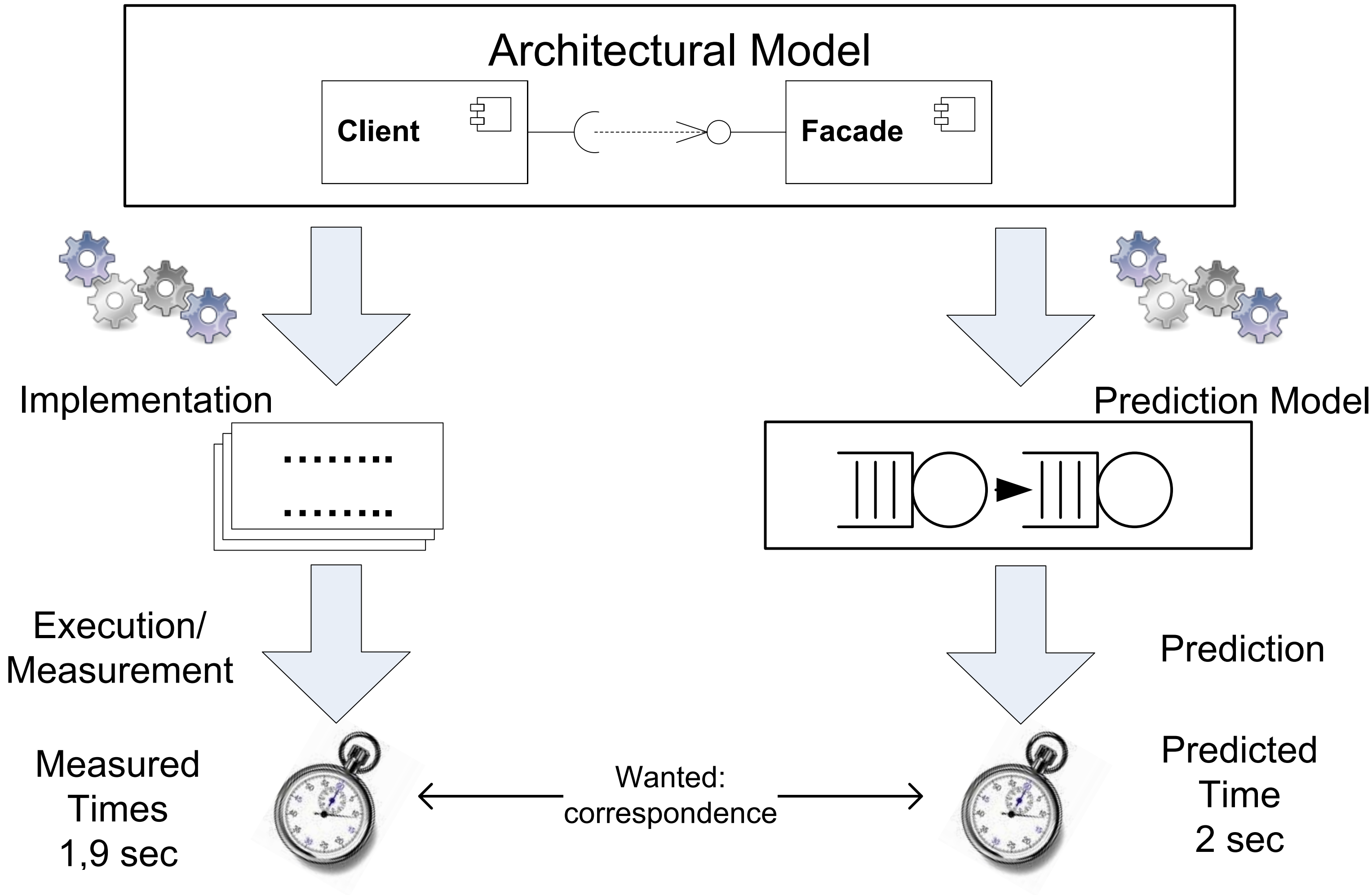
Motivation



[Becker, WOSP, 2008]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

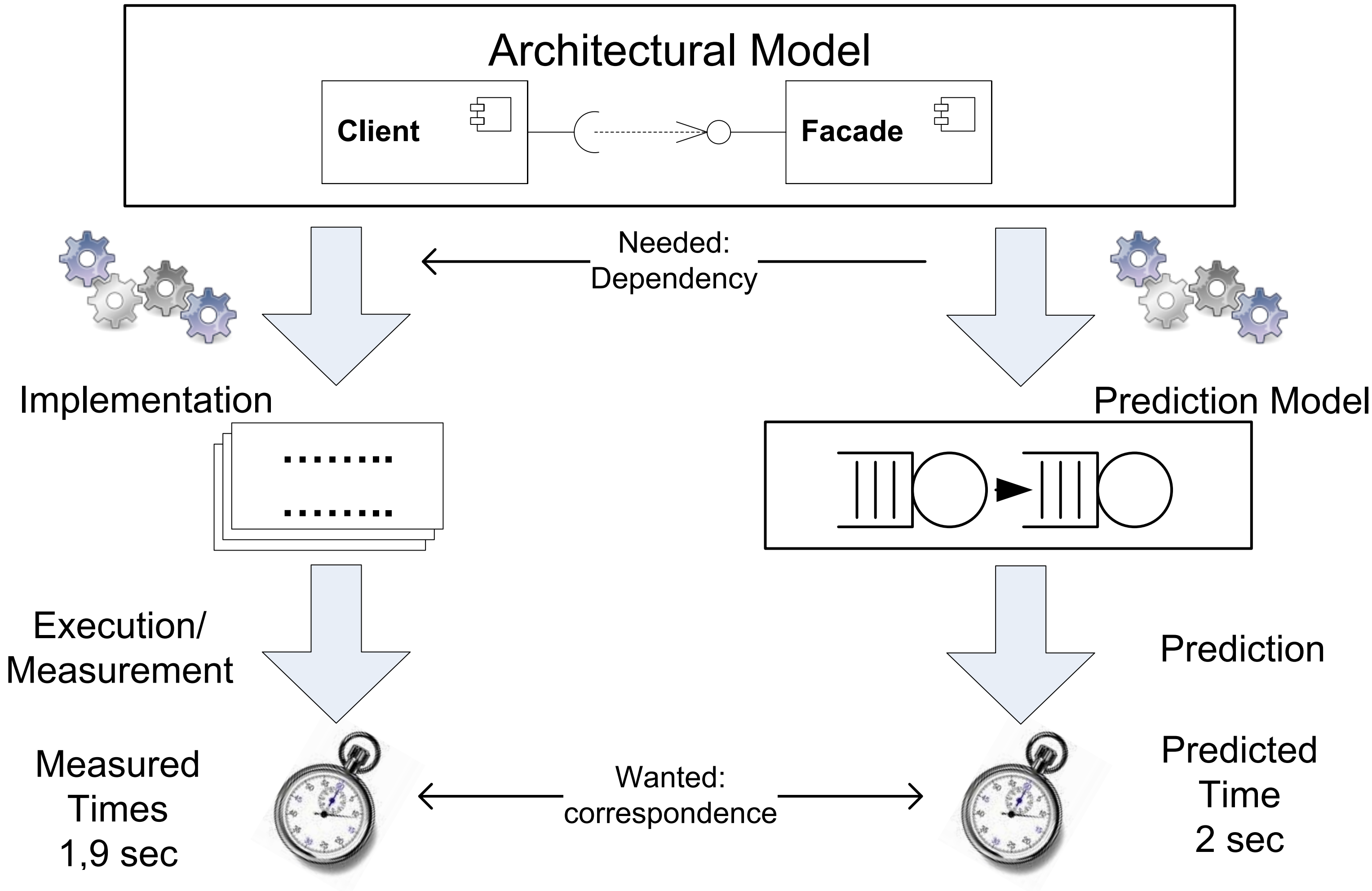
Motivation



[Becker, WOSP, 2008]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

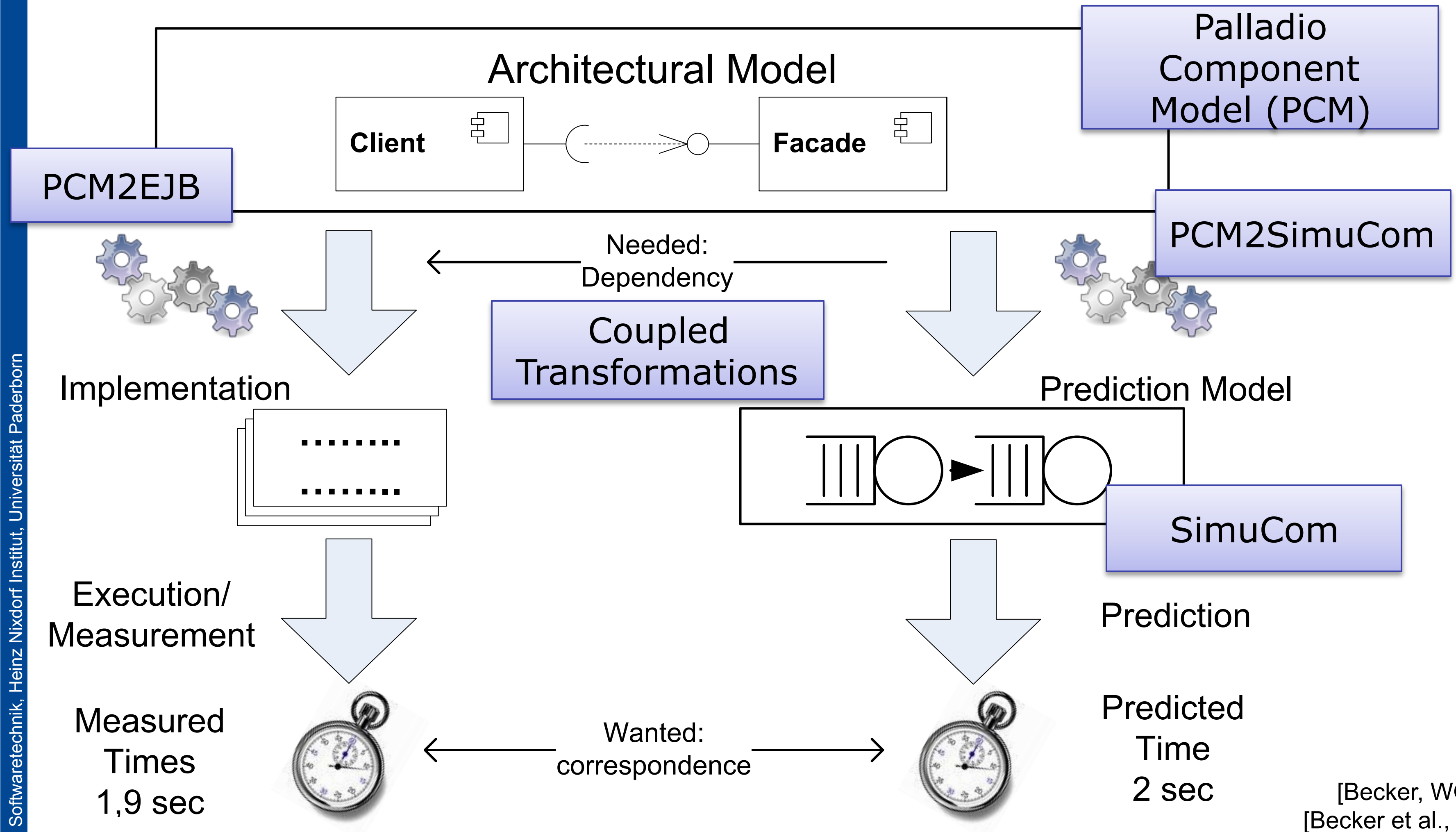
Motivation



[Becker, WOSP, 2008]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Motivation

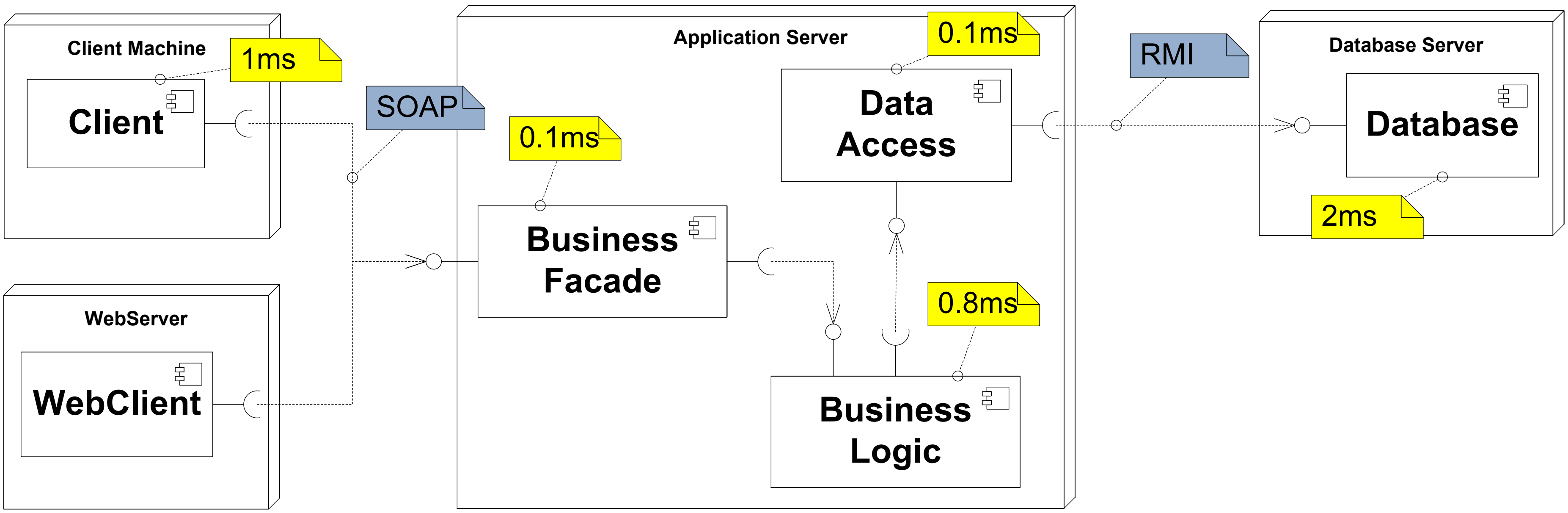


[Becker, WOSP, 2008]
[Becker et al., JSS, 2008]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

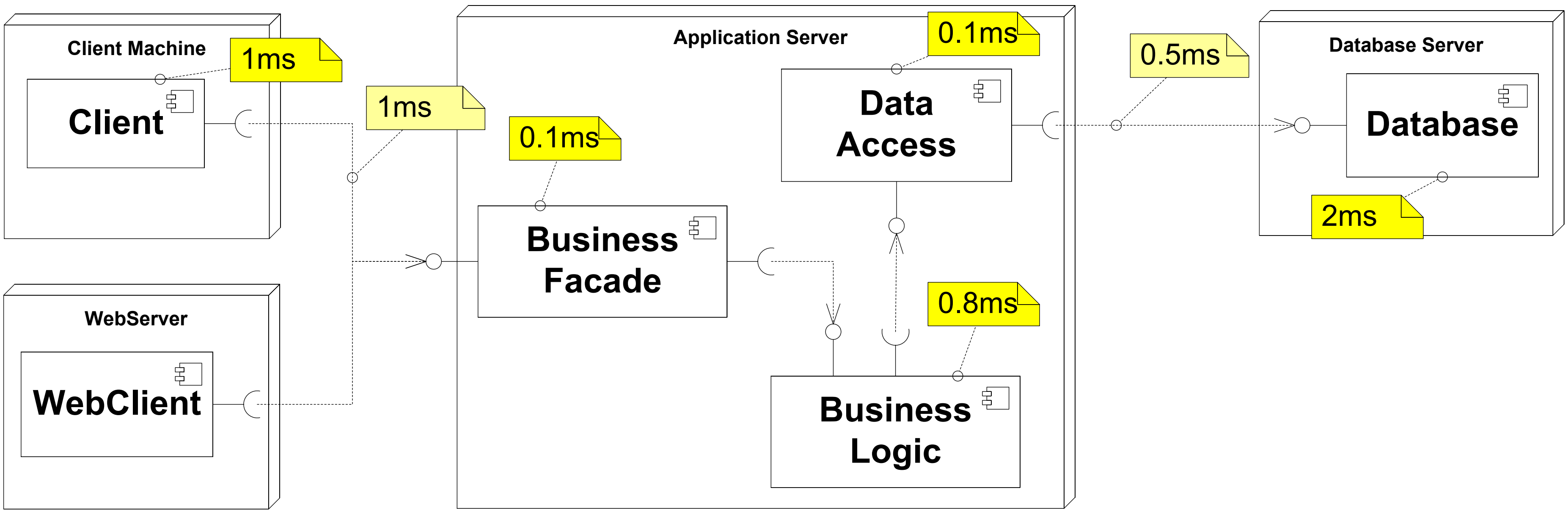
Example cont.



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Example cont.

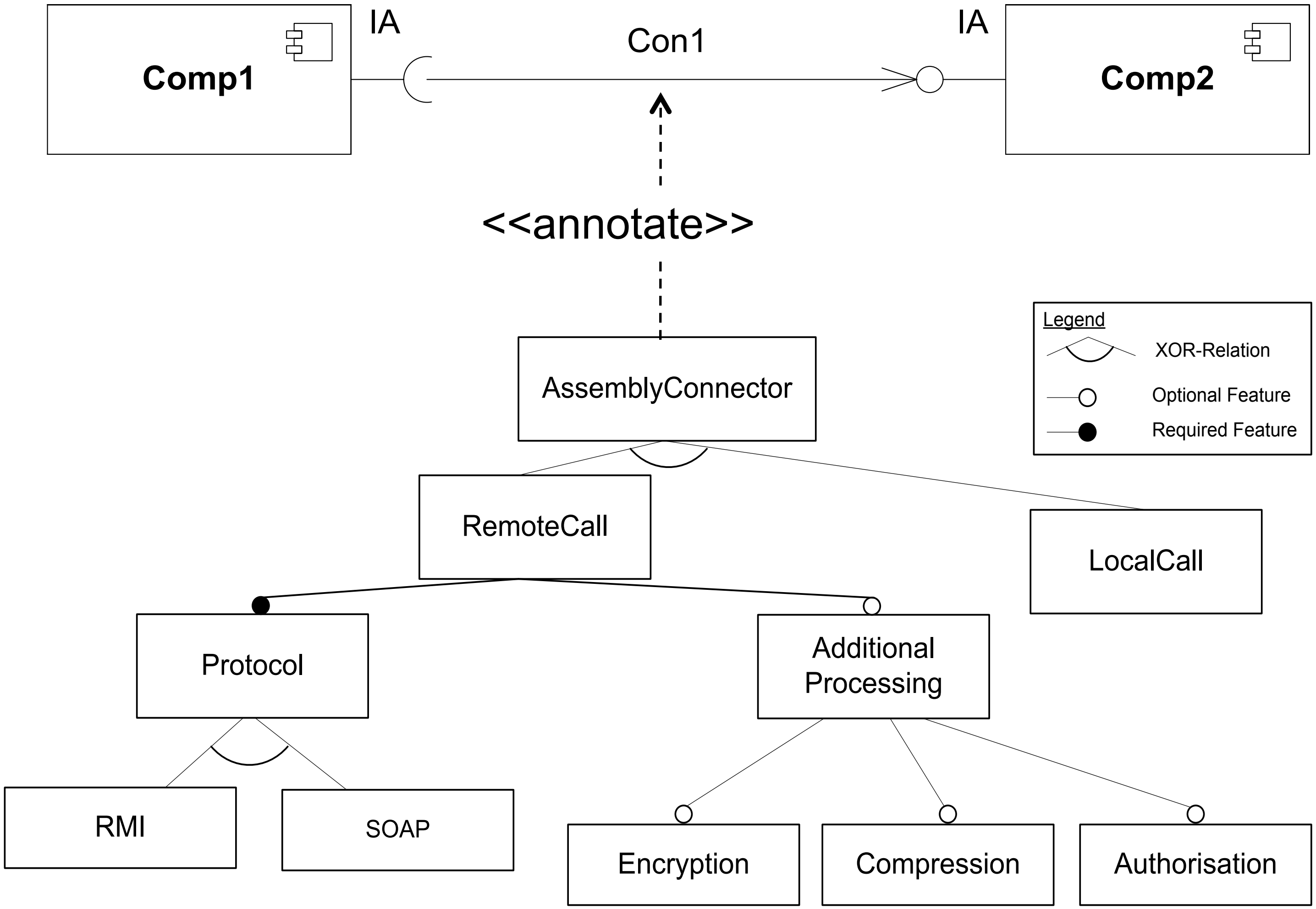
HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Connector Completion

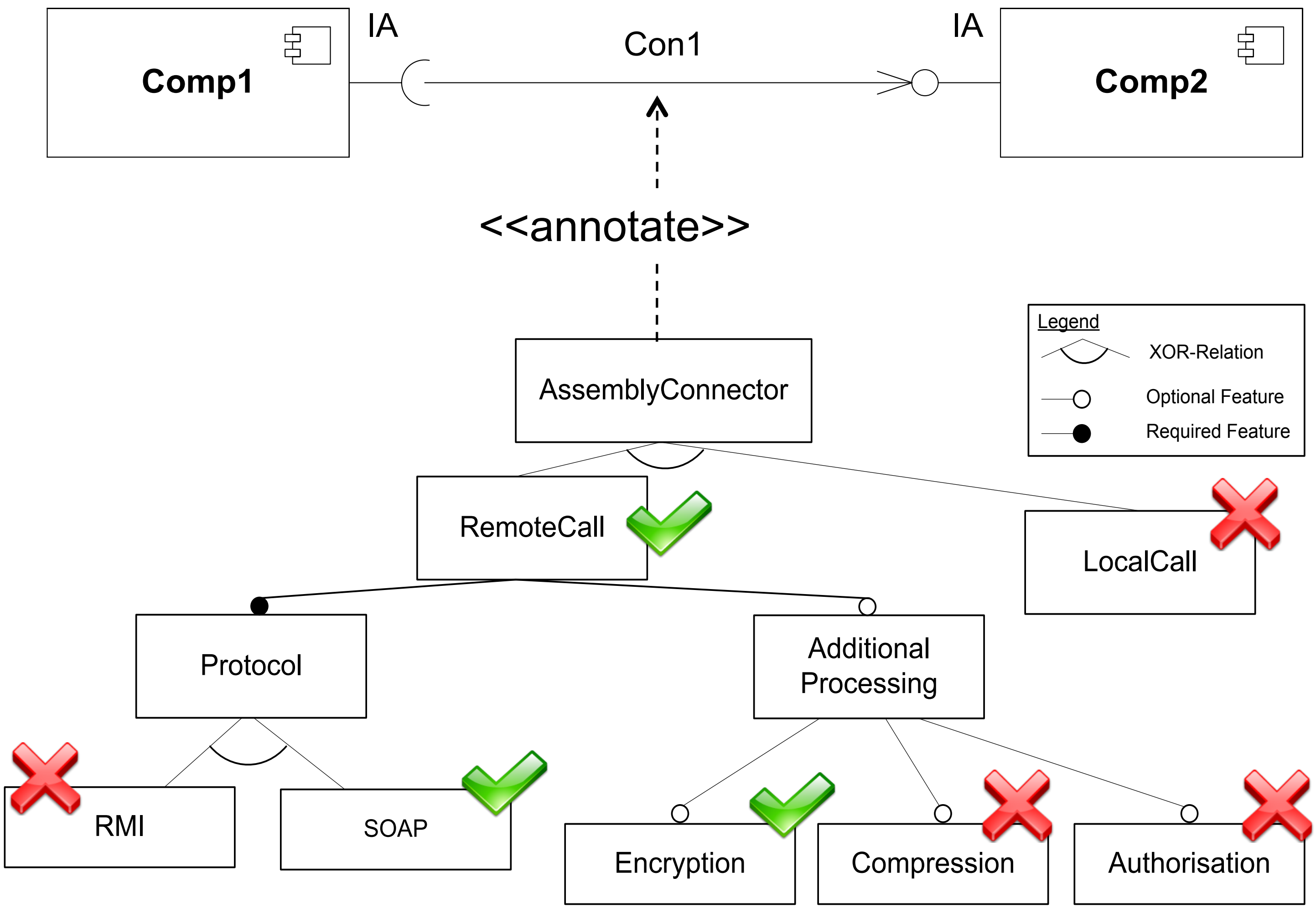
HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

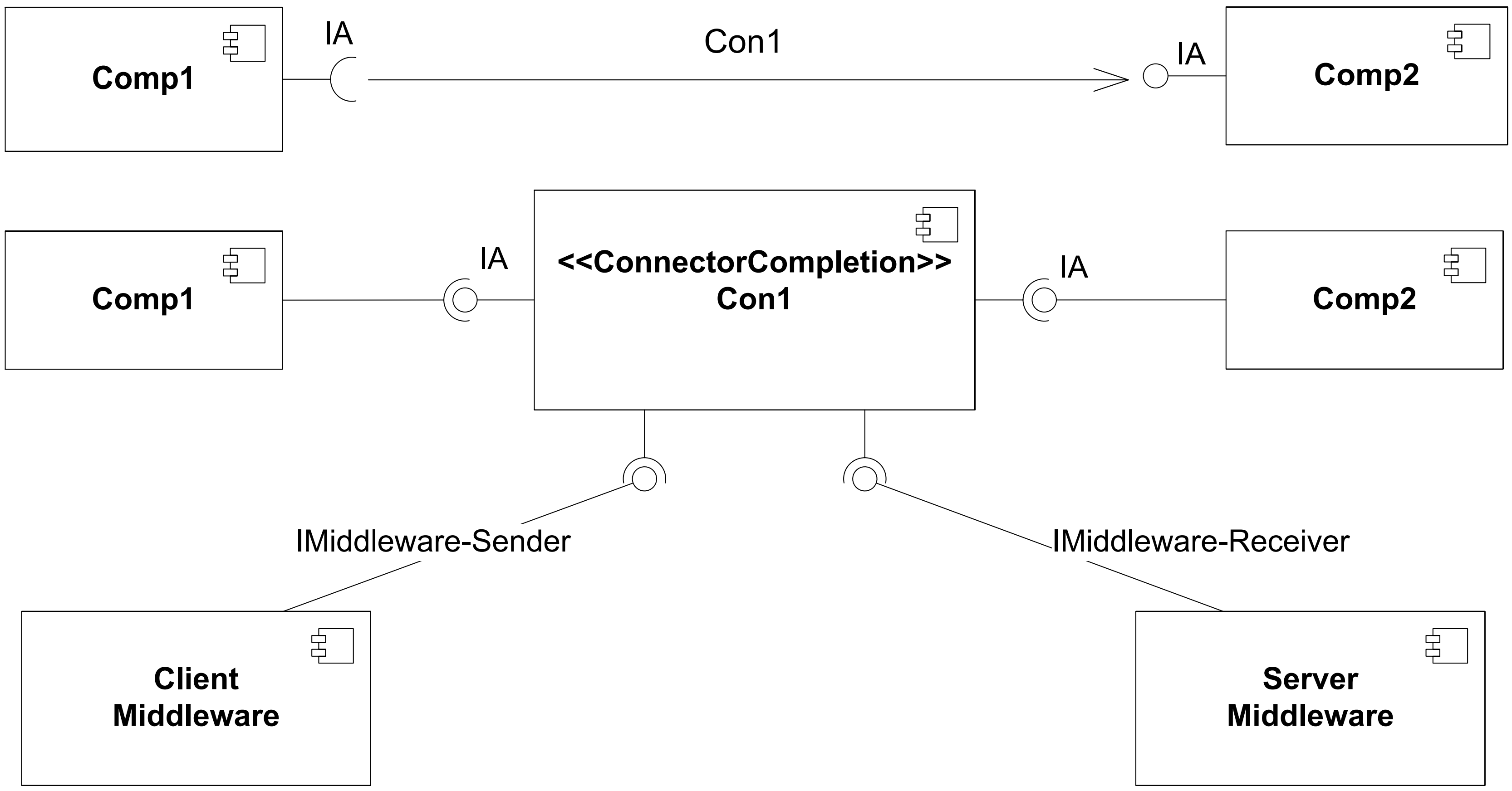
Connector Completion

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker



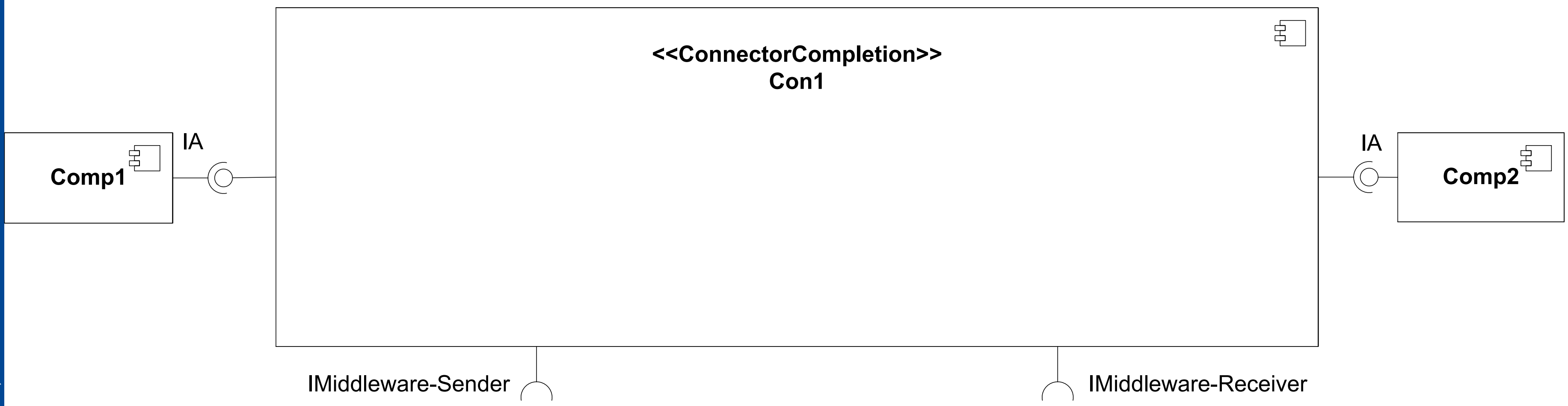
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Connector Completion



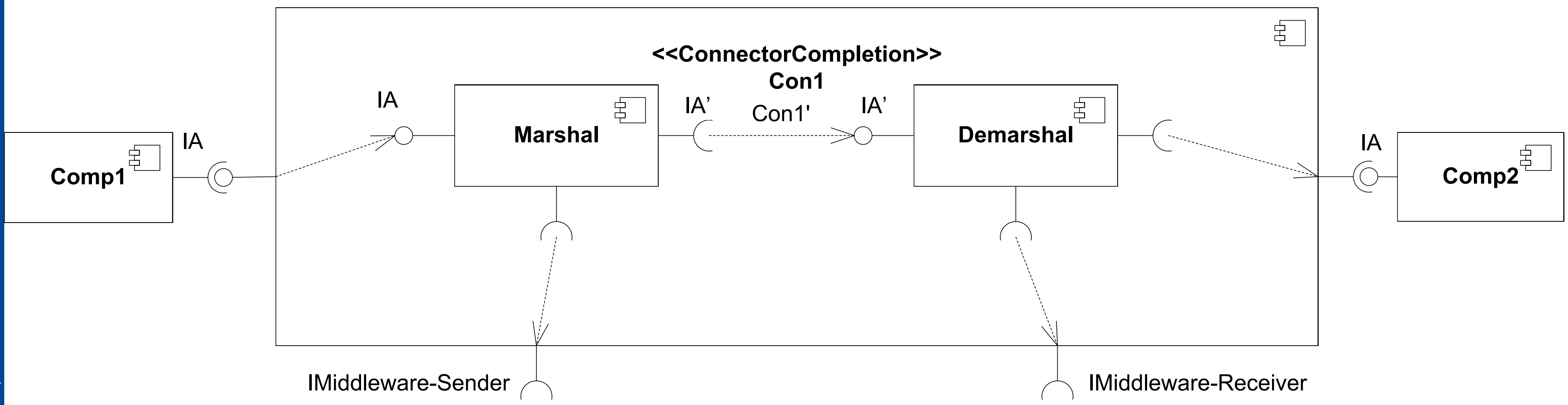
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Connector Completion



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

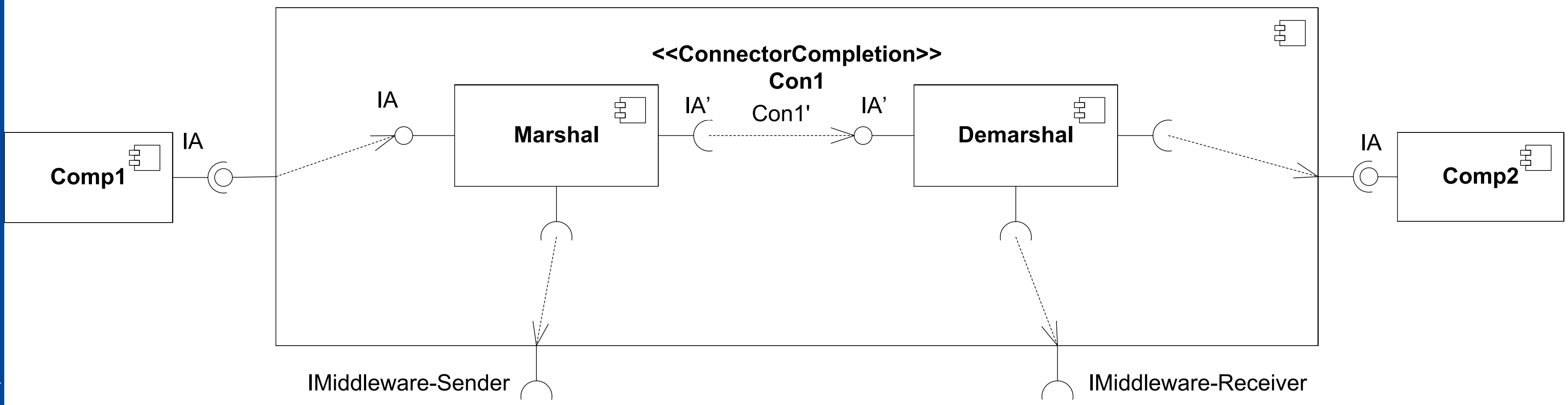
Connector Completion



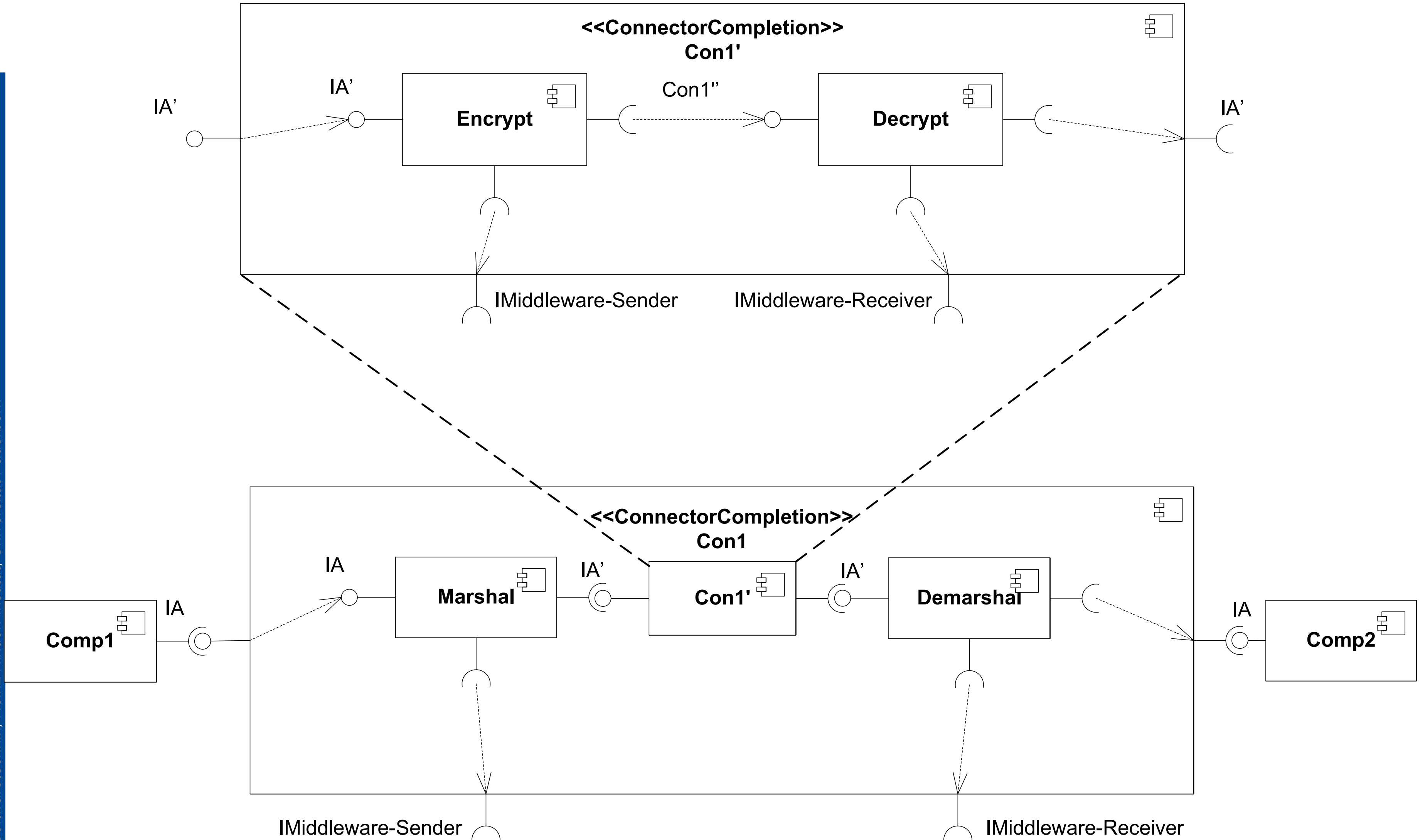
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Connector Completion

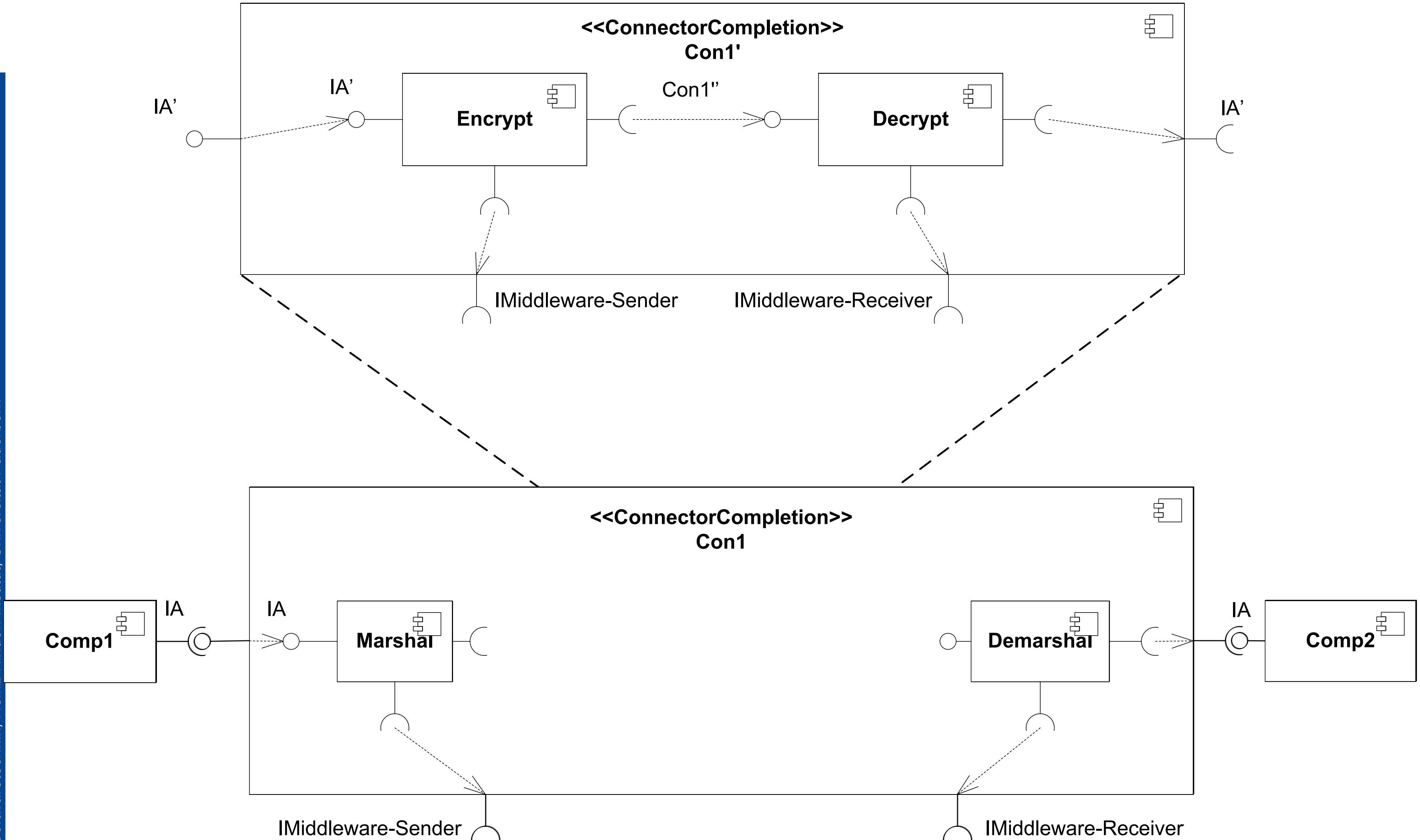
HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker



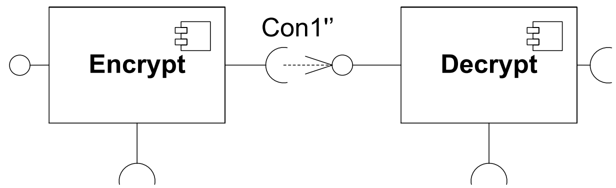
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

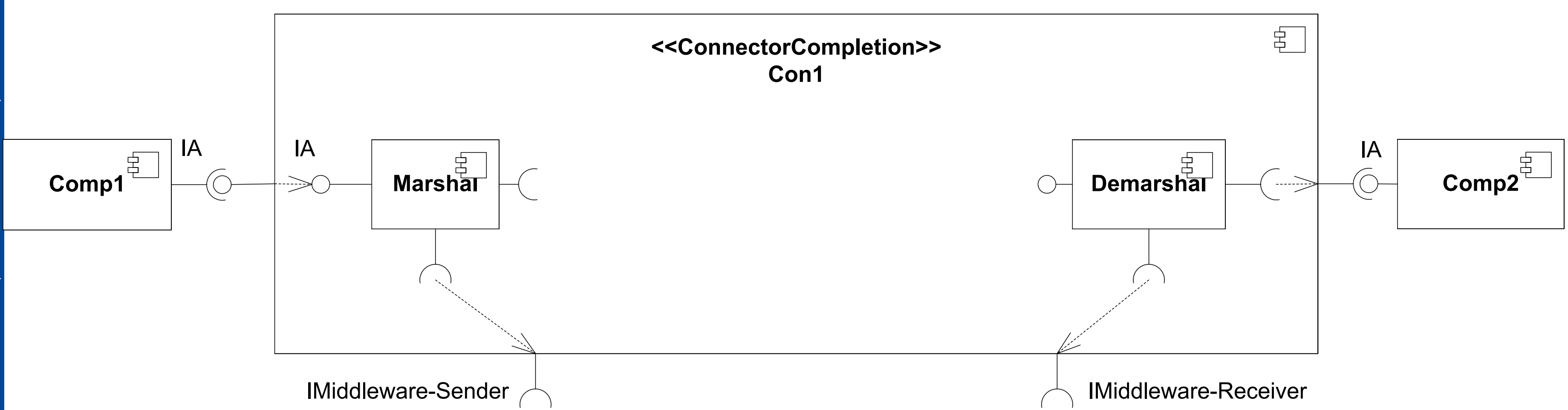


© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

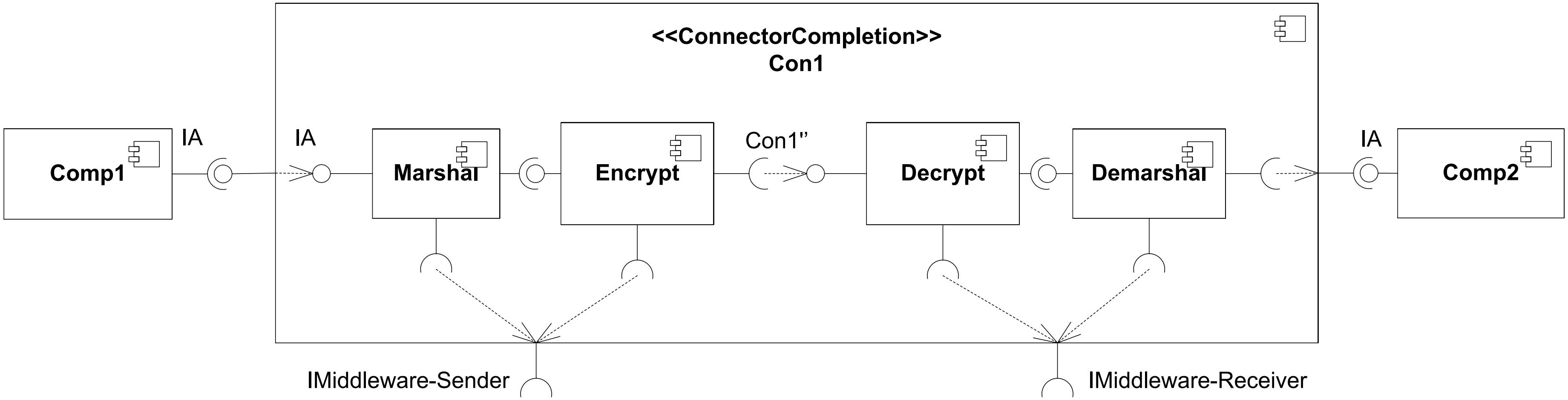


© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn





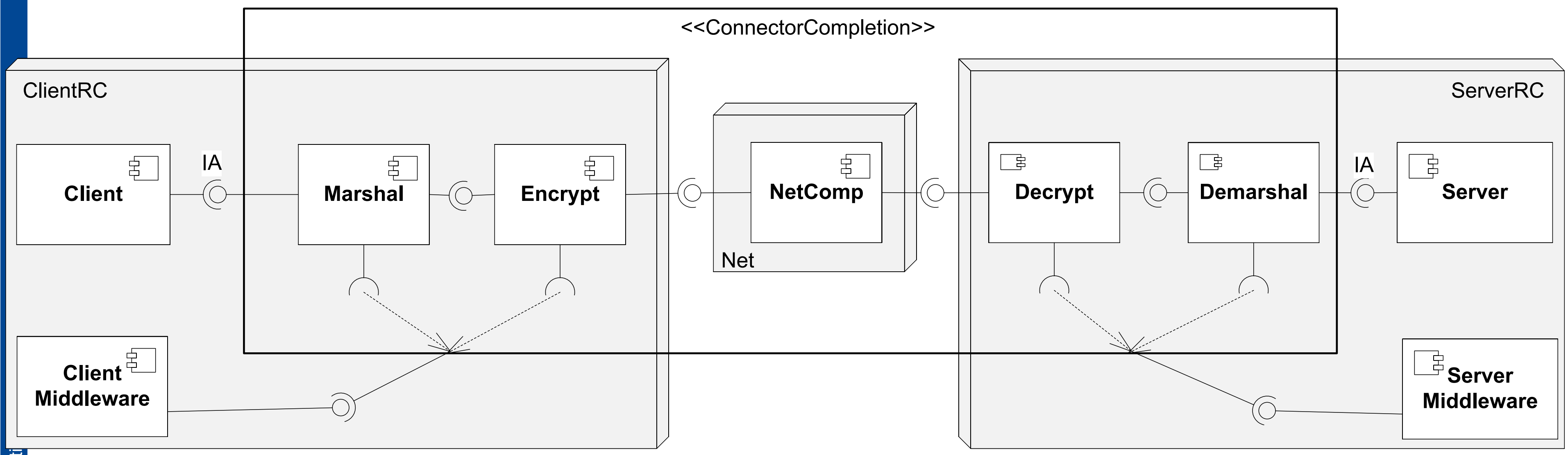
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

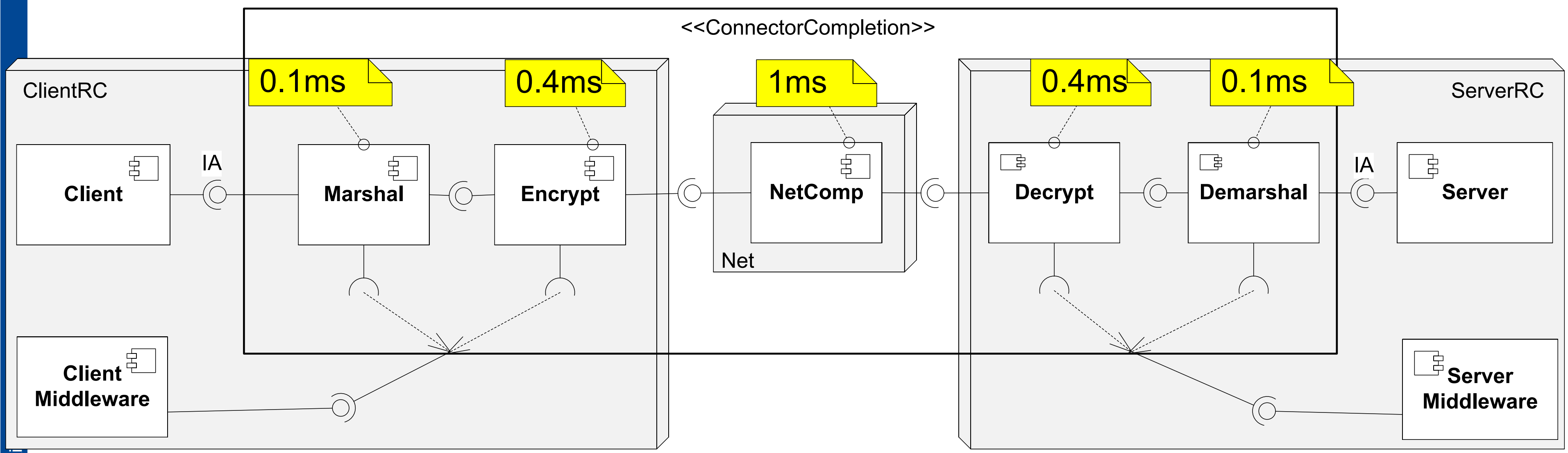
Deployed Completion



© Fachgebiet Softwaretechnik, Heinz Nixdorf Insti

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

Deployed Completion

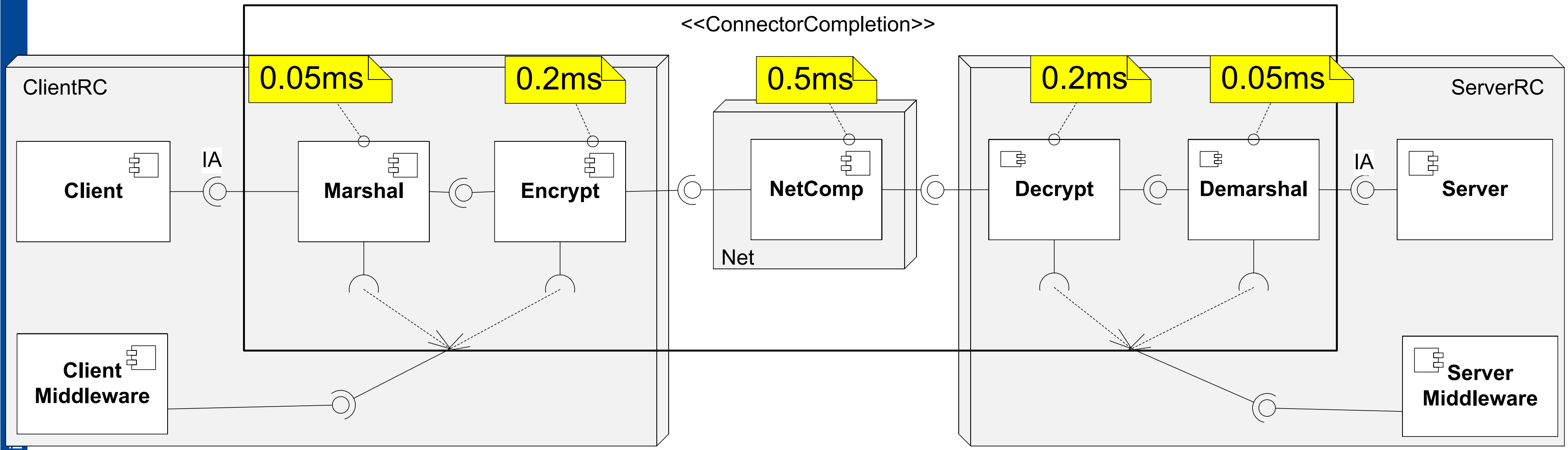


SOAP & Encryption

© Fachgebiet Softwaretechnik, Heinz Nixdorf Insti

Deployed Completion

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker



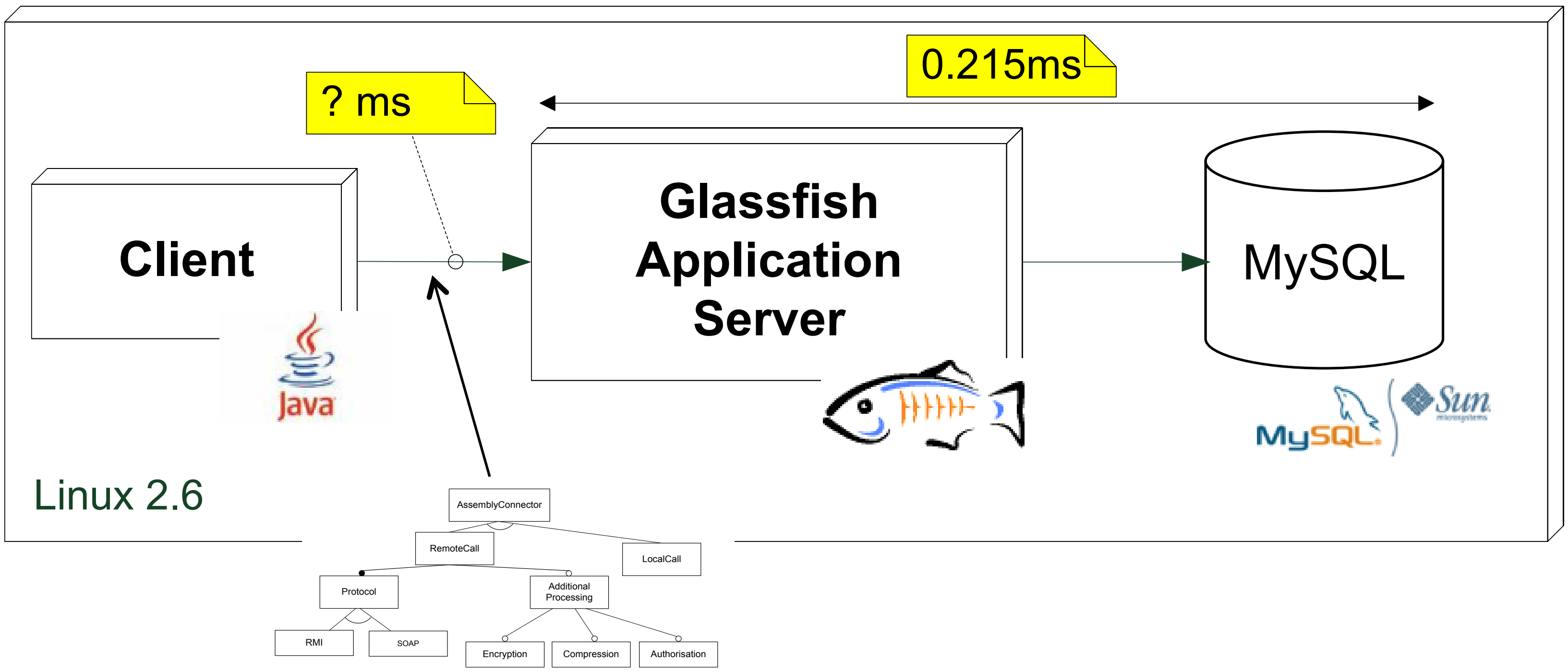
RMI & Encryption

© Fachgebiet Softwaretechnik, Heinz Nixdorf Insti

Model-Driven Completions: Evaluation

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Measurement Setting



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Results

Protocol	Auth	Enc	E(Meas) [ms]	E(Pred) [ms]	Diff. [ms]	Diff. [%]
None	✗	✗	n/a	0.215	n/a	n/a

Prediction Error > 400%

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

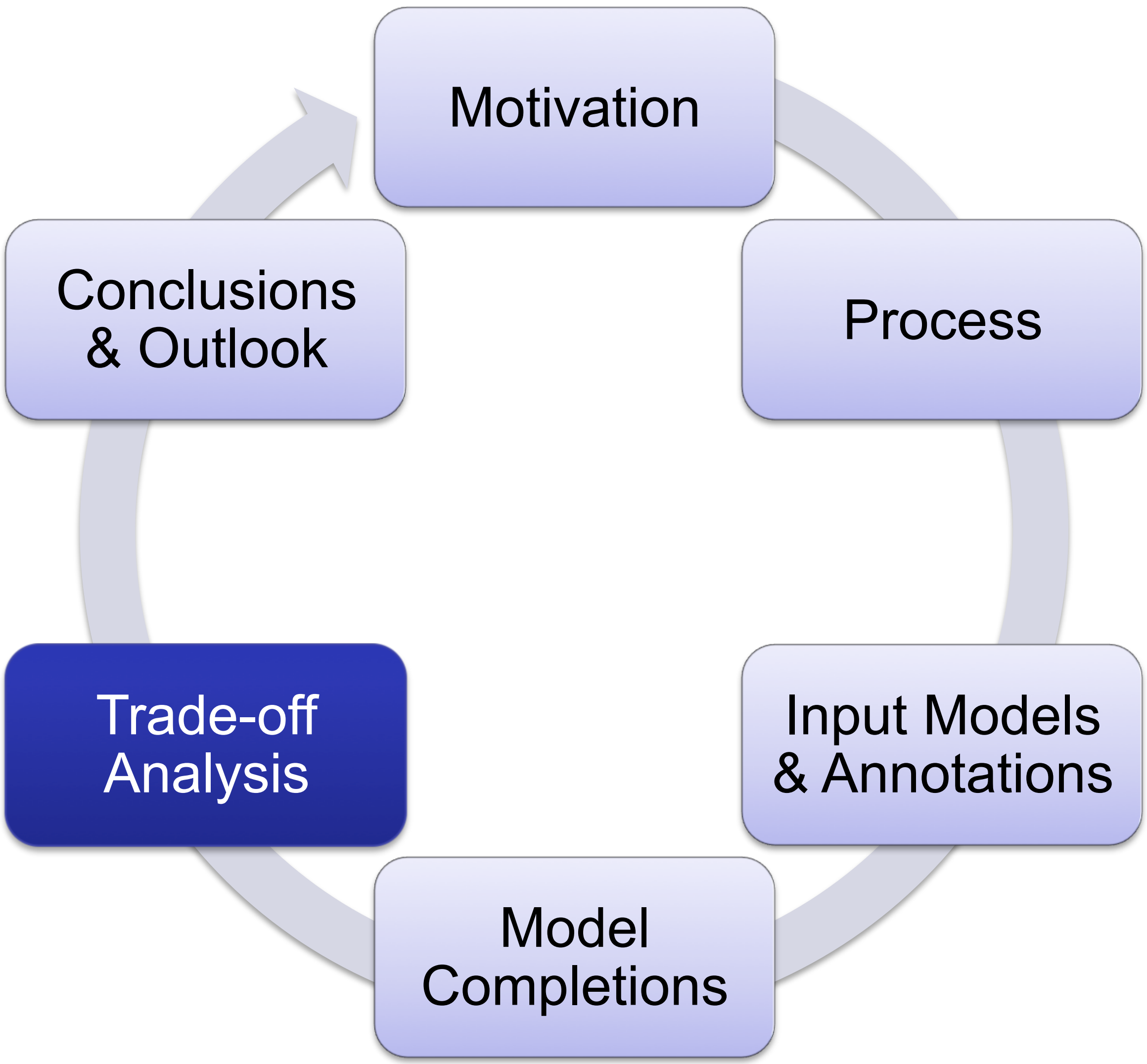
Results

Protocol	Auth	Enc	E(Meas) [ms]	E(Pred) [ms]	Diff. [ms]	Diff. [%]
None	✗	✗	n/a	0.215	n/a	n/a
RMI	✗	✗	0.988	0.971	0.017	1.7
RMI	✓	✗	1.941	1.871	0.070	3.6
RMI	✗	✓	1.113	1.111	0.002	0.2
RMI	✓	✓	2.060	2.011	0.049	2.4
SOAP	✗	✗	1.329	1.283	0.046	3.5
SOAP	✗	✓	1.498	1.463	0.035	2.3

Prediction Error < 3.6%

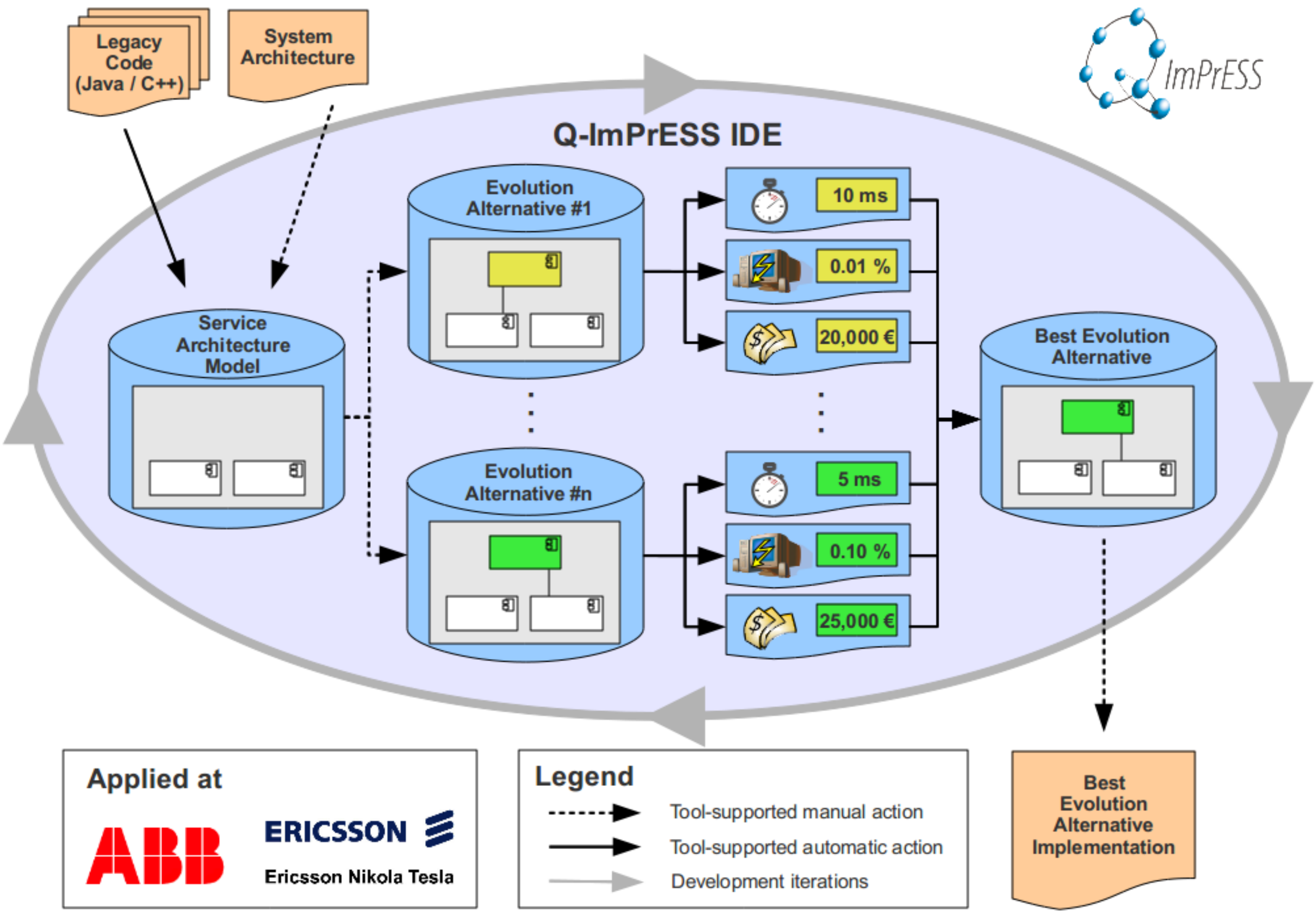
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Outline



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Trade-off Analysis
Example: Q-ImPrESS



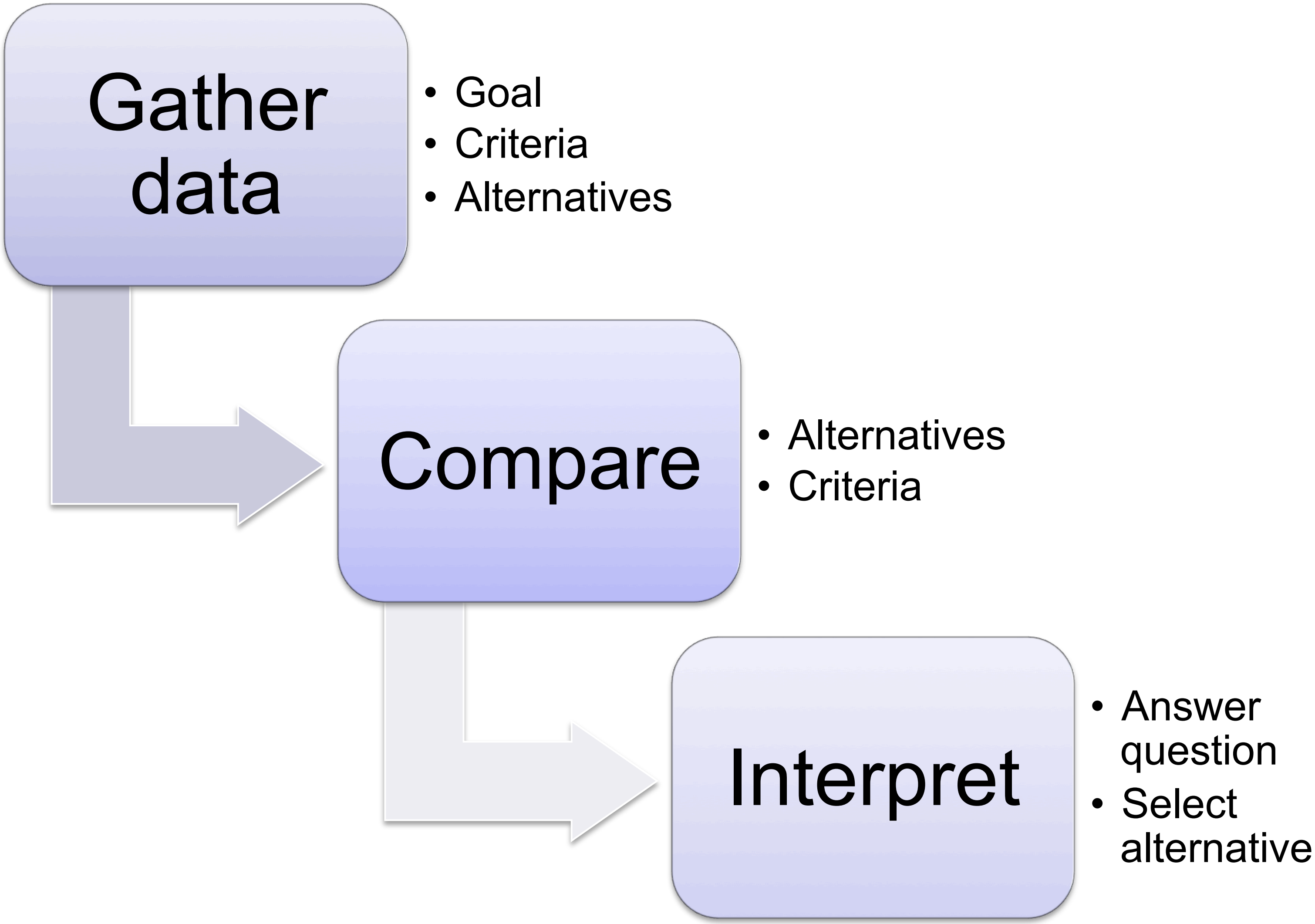
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Analytic Hierarchy Process

- Decision support method developed by the mathematician Thomas Saaty in the 1970s
- General method for multi-criteria decision making (not only software, but, e.g., also politics)
- Works quantitatively with qualitative inputs
- Goals
  - Make decisions rationally
  - Make decision preferences explicit and understandable
  - Support group/team decisions
  - Discover inconsistent decisions/preferences

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Process Overview



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

1st Step: Gather Data

- Goal (here): „Which architectural alternative offers the best trade-off among the quantitative properties?“
- Criteria
  - Quantitative measurements/predictions on architecture metrics
  - We can also include non-quantitative criteria if we like
  - E.g., response time, PROFOD, etc.
  - Exclude direct alternative costs, they are normally included in step 3
- Alternatives
  - Alternative architectures

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Alternatives and criteria

	Avg. Job / s = 1 / Avg. RT	1 - PROFOD	Ease of impl. [5 = easy, 1 = hard]
Alternative 1	1 / 1s	0.99	3
Alternative 2	1 / 2s	0.9	2
Alternative 3	2 / 3s	0.9	5
Alternative 4	1 / 3s	0.9999	1

Values e.g. from a QN

Values e.g. from a DTMC

Manual input

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

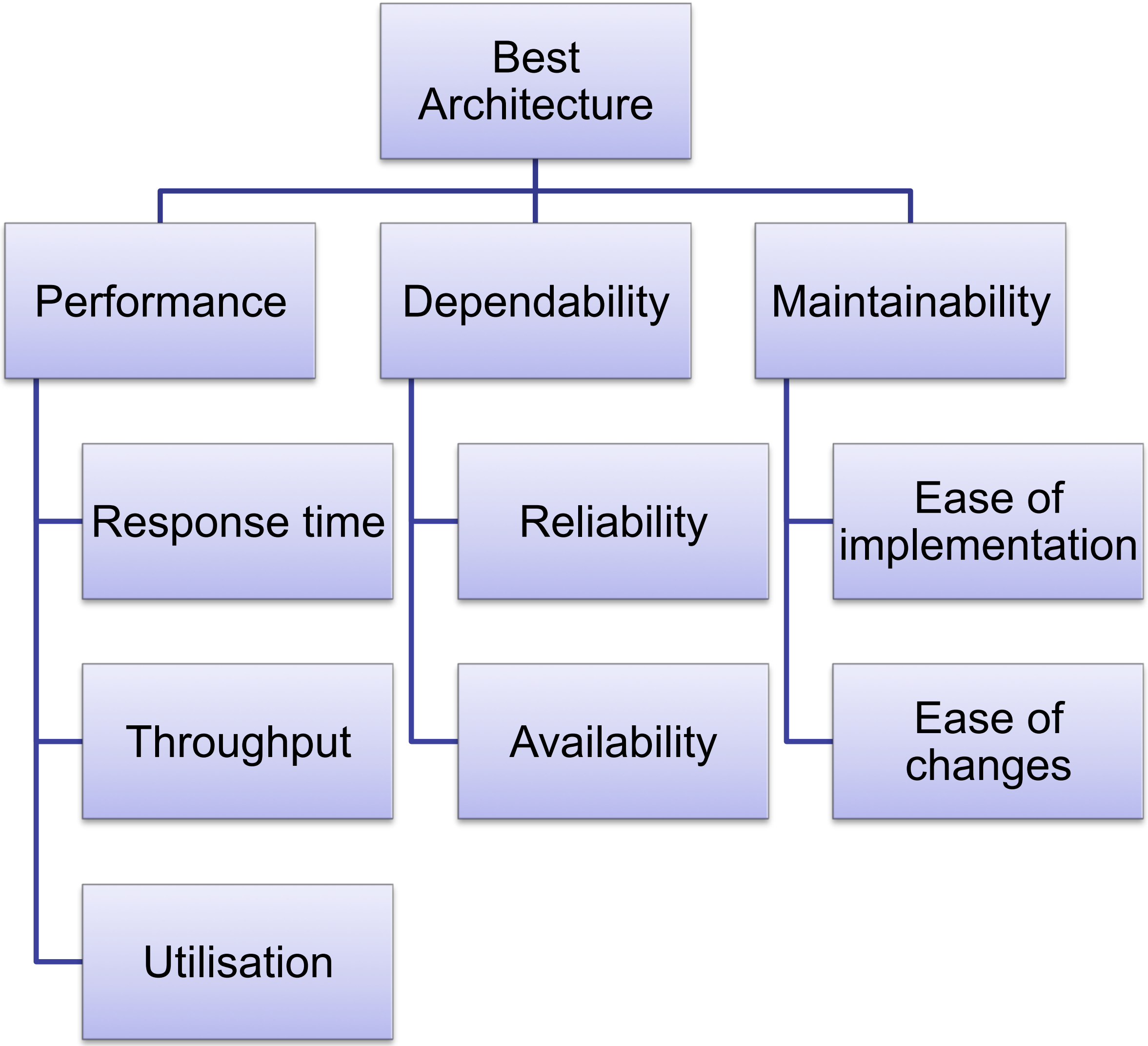
Dominance & Pareto front

- If an alternative A is better in *all* criteria values than an alternative B, A is said to dominate B → B can be removed from the list of alternatives
  - E.g. Alternative 1 dominates Alternative 2
- The set of alternative which are not dominated by any other alternative is called the Pareto-optimal set or (if shown in figures) the Pareto-front

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

1st Step: Create a Hierarchy



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

2nd Step: Priorities

- How to formalise priorities?
- Naive approach: Let stakeholders assign direct weights
- Problem
  - Decision theory research shows stakeholders give bad weights when asked to do it directly
  - Test yourself: Which weight is your preference for performance vs. reliability?
- Solution
  - AHP uses pair-comparisons
  - Compare two sub-goals of a given parent goal
  - Judge it quantitatively

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Priorities (cont.)

- Use Scale from 1 to 9 (1-5 or 1-15 also used)
  - 1 = equally important
  - 3 = a bit more important
  - 5 = more important
  - 7 = much more important
  - 9 = extremely more important
  - 2,4,6,8 = intermediate values
  - Use 1/5 for „less important“ and so on
- Advantages
  - Easier to judge and express
  - Pairwise comparison limits scope of comparison

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Priorities example

Derive a matrix from answers

	Performance	Reliability	Ease of Impl.
Performance	1	3	1/3
Reliability	1/3	1	1/9
Ease of Impl.	3	9	1

Matrix has "symmetry" if answers are consistent, AHP can also deal with some inconsistency

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Consistency

- Can be used to check consistency in the answers of the decision makers
- Let v^h_{ij} denote matrix entry ij in a matrix of the parent goal h
- The comparison matrix is consistent if

$$v^h_{ik} = v^h_{ij} \cdot v^h_{jk} \qquad \forall i, j, k \in U(h)$$

  holds.
  (with U(h) contains the indices of the criteria or alternatives)
- For consistent matrices it is sufficient to determine the first row, i.e., we have to determine $|U(h)| - 1$ pair comparisons

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Determining weights

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

- Let $\vec{w}^h$ denote the weights of subgoals used to determine the value of the parent goal
- $\vec{w}^h$ should reflect the preferences of the decision maker as given in the matrix
- That is

  $$v_{ij} \approx \frac{w_i}{w_j}$$

  i.e. the weights should reflect the relative preferences
- To ensure comparability with other goals normalise $\vec{w}^h$

  $$\sum_{i=1}^{|U(h)|} w_i = 1$$

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Determining weights cont.

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

- If matrix is consistent, take arbitrary column vector and normalise

$$w_i = \frac{v_{ij}}{\sum_{k=1}^{|U(h)|} v_{ki}}$$

- Example: 2nd column = (3,1,9), i.e., it follows

$$\vec{w} = (\frac{3}{13}, \frac{1}{13}, \frac{9}{13})$$

- Otherwise find a solution with as little deviation as possible
  → Heuristics exists

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Heuristic Solutions

- Determine eigenvector belonging to the maximum eigenvalue showed to be a good heuristic
- Process
  - Solve characteristic equation $det(V - I\mu) = 0$
  - Pick maximum eigenvalue μ_{max}
  - Determine the matching eigenvector by solving

    $$(V - \mu_{max} I)x(\mu_{max}) = 0$$

  - Let $\vec{w} = x(\mu_{max})$

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Quantify inconsistency

- There is a inconsistency index telling us the inconsistency of the matrix

$$IK(V) = (\mu_{max} - |U(h)|)/(|U(h)| - 1)$$

- Problem IK(V) depends on the size of V
- The larger V the more likely are violations of the consistency constraint
- Solution: Use randomly generated V's as reference to normalise the metric. This is called ZI(|U(h)|)
- Heuristic for too inconsistent matrices: If

$$\frac{IK(V)}{ZI(|U(h)|)} > 0.1$$ then redo pair comparisons

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Determine utility values

- Numeric values (1 / RT, 1 – PROFOD)
  - Normalise values
  - Multiply normalised vector with weight
  - Ensure larger values correspond to more utility
- Qualitative values (e.g. ease of implementation)
  - Again use pair comparisons
  - Determine utility in the same way as weights
  - Use normalised vector as utility

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Utility of our alternatives

	Avg. Job / s = 1 / Avg. RT	1 - PROFOD	Ease of impl. [5 = easy, 1 = hard]
Alternative 1	0.5	0.34	3/9
Alternative 3	0.33333	0.311	5/9
Alternative 4	0.16666	0.345	1/9

Maybe a better scaling is needed here??

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Solution for our simple example

- Solve (for Alternatives 1,3,4, i.e., removing the dominated)

$$\begin{pmatrix} 0.5 & 0.34 & 0.333 \\ 0.333 & 0.311 & 0.555 \\ 0.166 & 0.345 & 0.111 \end{pmatrix} \begin{pmatrix} 0.23 \\ 0.076 \\ 0.694 \end{pmatrix} = \begin{pmatrix} 0.372 \\ 0.485 \\ 0.141 \end{pmatrix}$$

- Select Alternative with maximum weighted value
- In this case, it is Alternative 3 with a value of 0.485

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Discussion

- Rank inversion
  - Adding alternatives may alter alternative ranking
  - Even for irrelevant alternatives
  - One reason is normalisation over the range of values
- Number of pair comparisons
  - Min |U(h)|-1, Max |U(h)|(|U(h)|-1)/2
  - More comparisons allow to judge inconsistency
  - However, people get tired answering pair comparisons
- Scale
  - Good scale for AHP determined in practice
  - However, no rationale for the scales used
- Nevertheless, helpful in practice for small numbers of alternatives

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

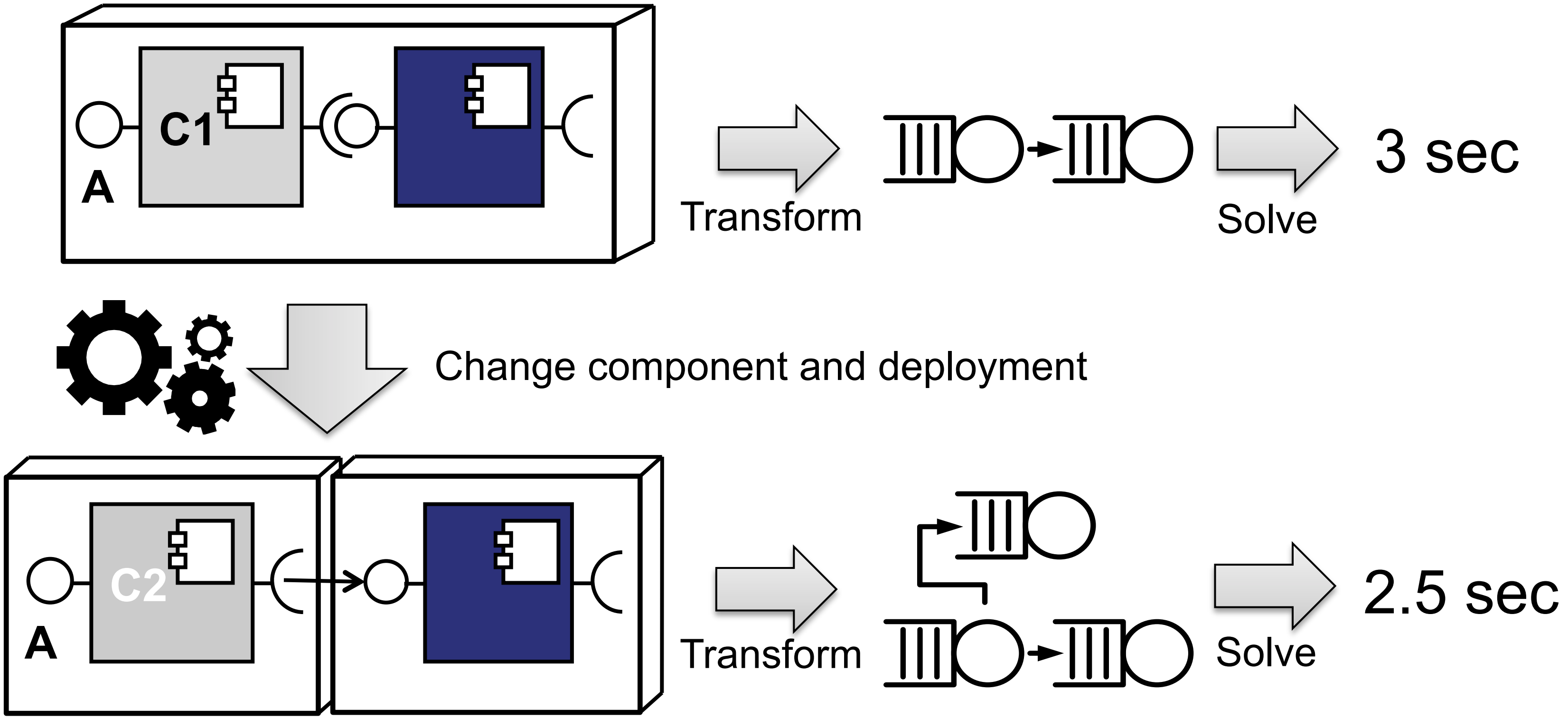
Automated Quantitative Approach

Using Meta-Heuristics

[Slides with kind permission from Anne Koziolek]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

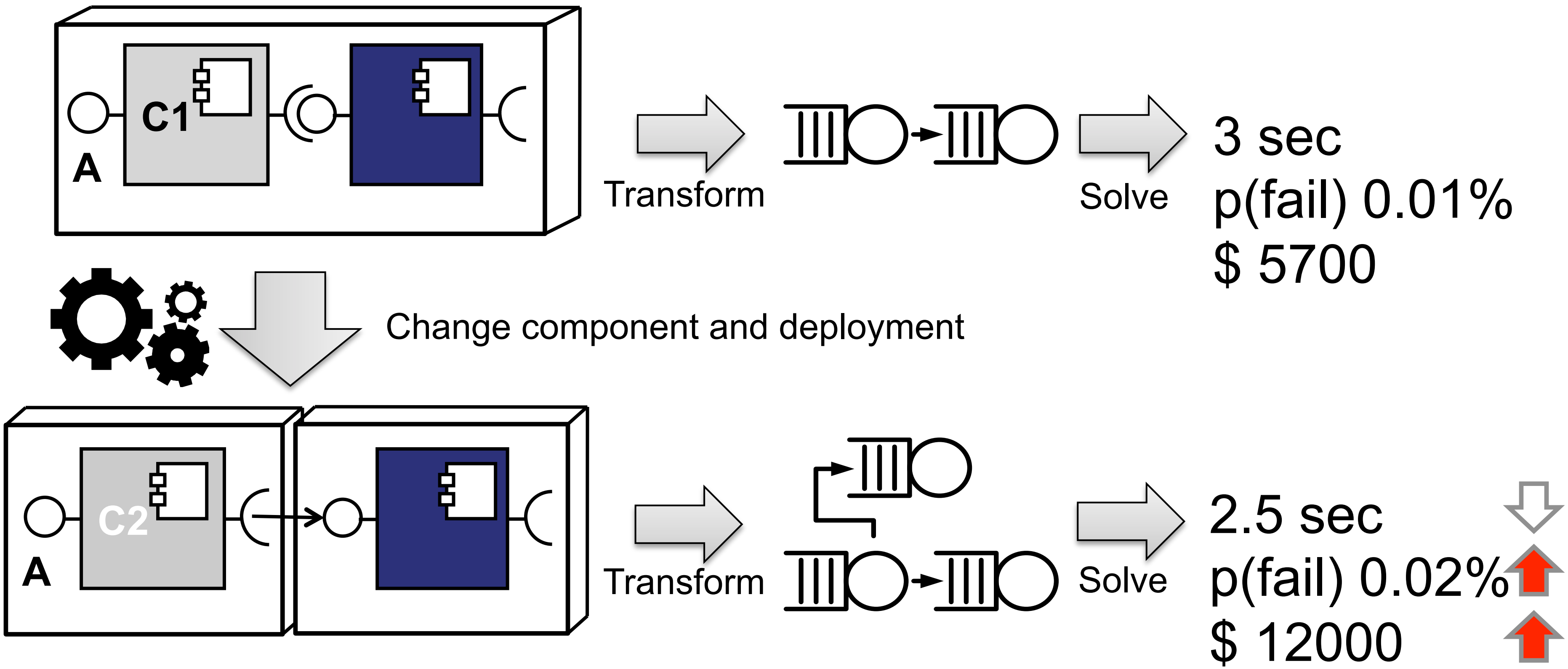
Idea



[MKB10]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Idea

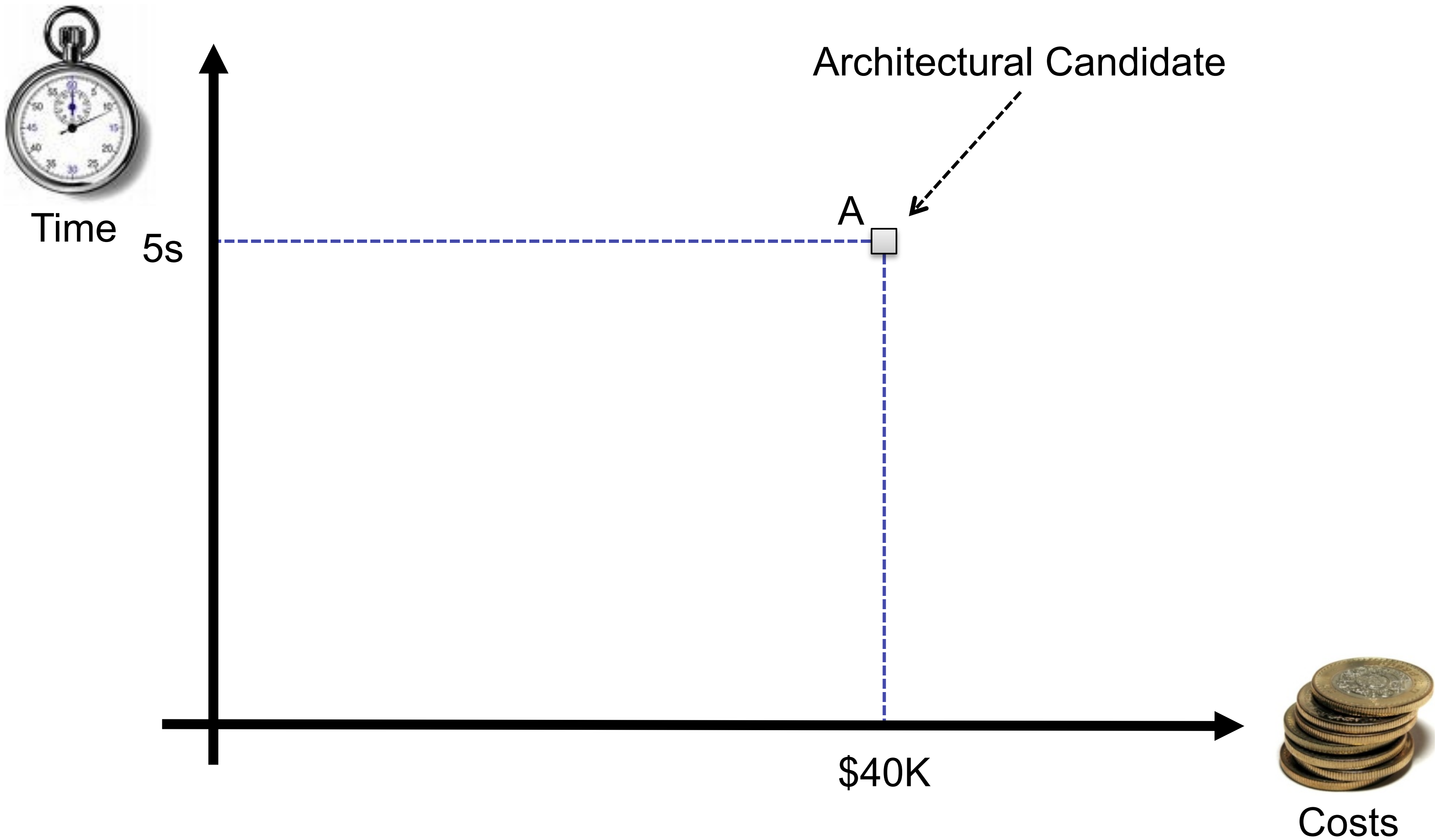


➔ Optimise multiple criteria at once

[MKB10]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

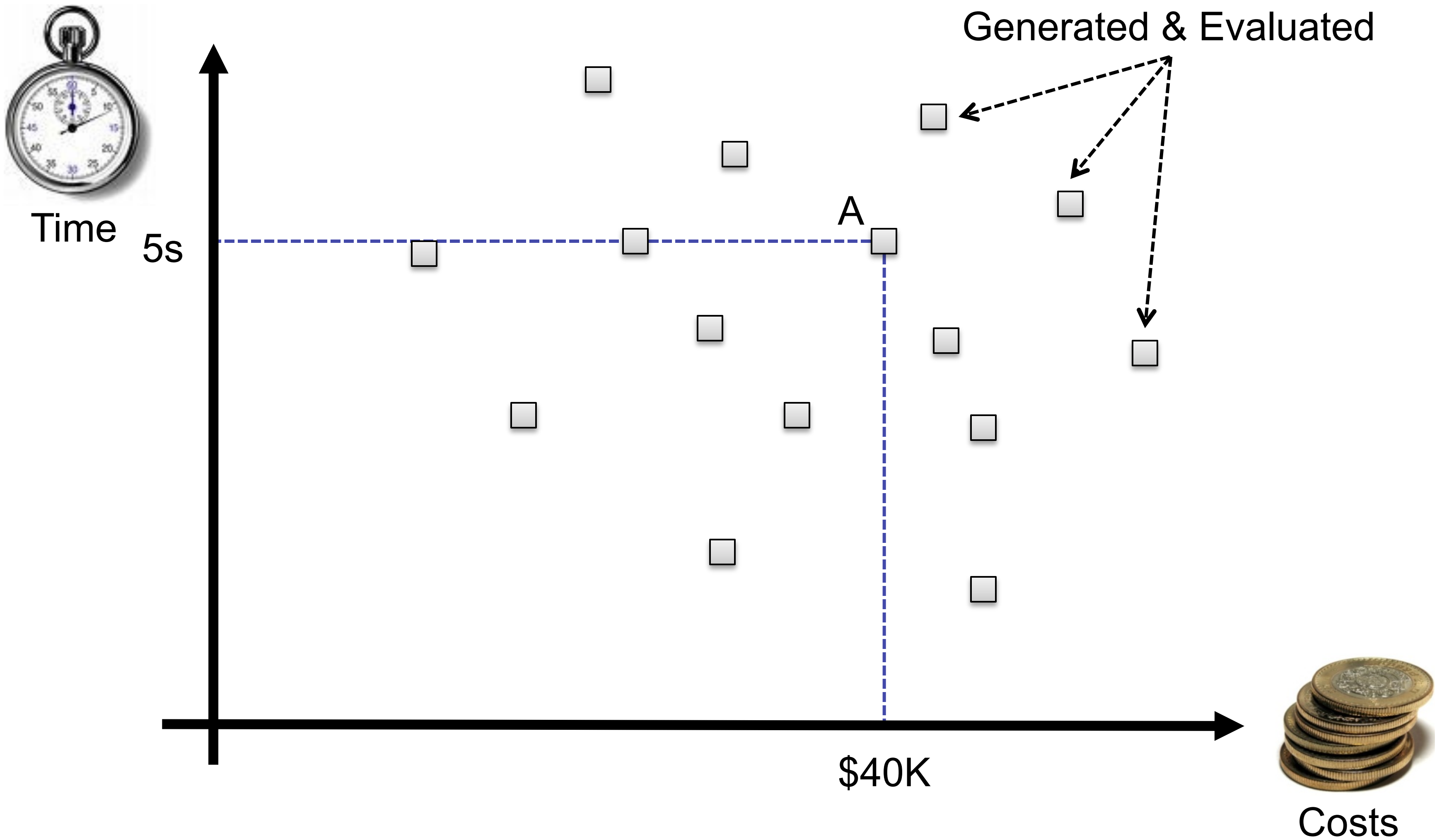
Multi-criteria optimisation



[MKB10]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

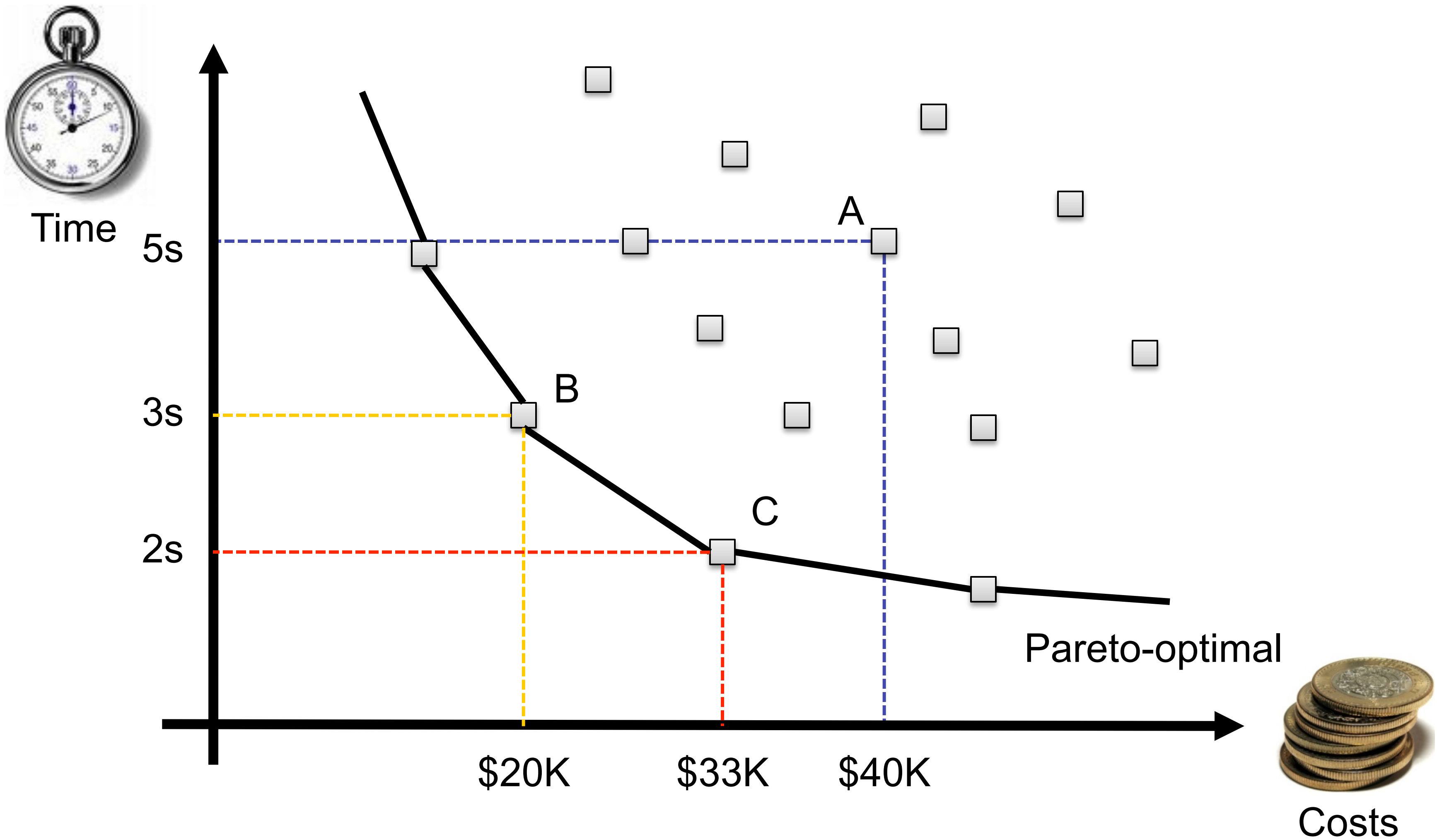
Multi-criteria optimisation



[MKB10]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Pareto front



[MKB10]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Process

Initial candidate

Generic degrees of freedom

1. Search problem instantiation

[MKB10]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Degrees of Freedom

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

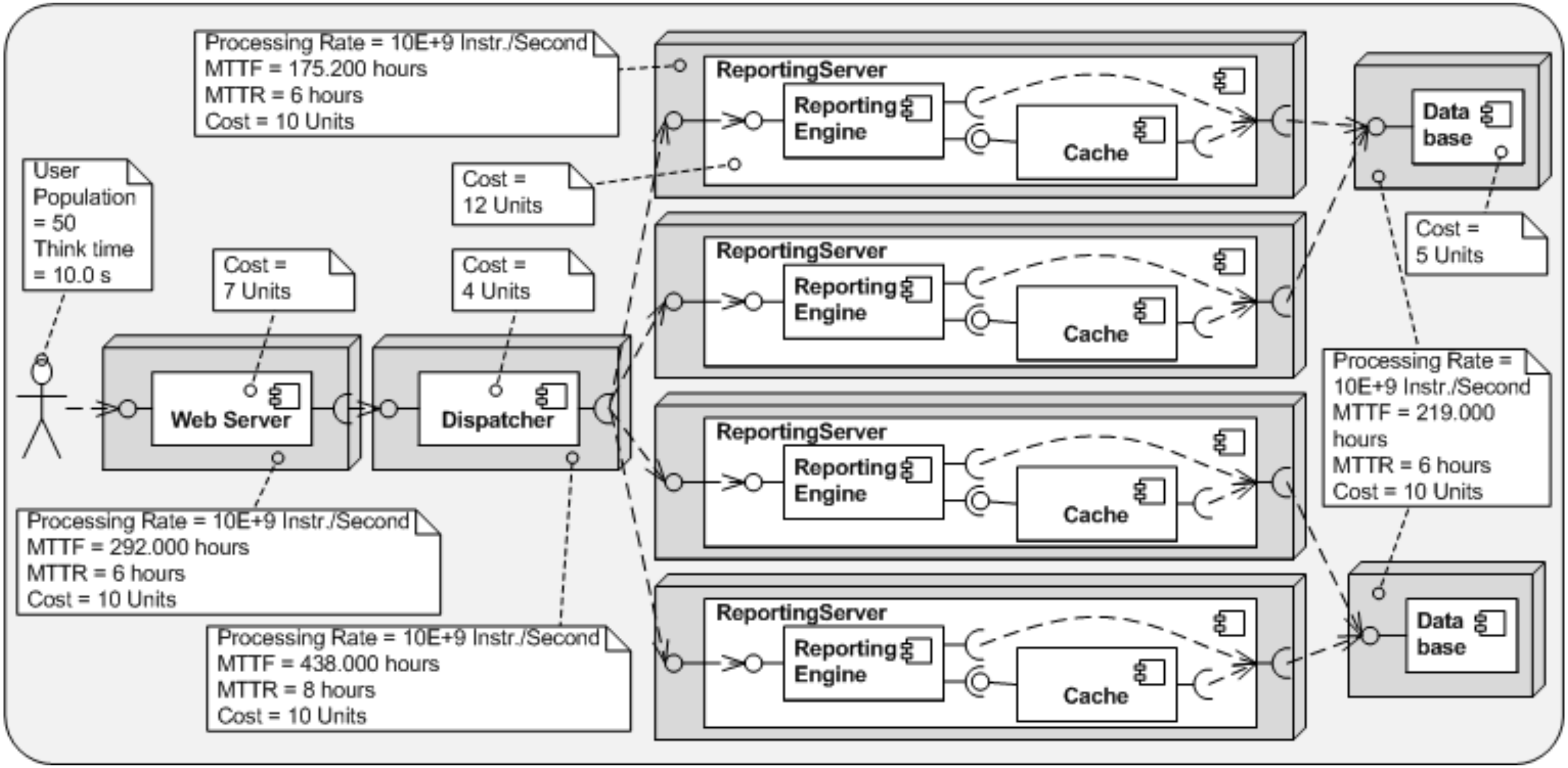
Software
Component selection
Middleware selection
Component replication
Software configuration

Deployment
Allocation
Processing Rate
Number of Servers

[MKB10]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

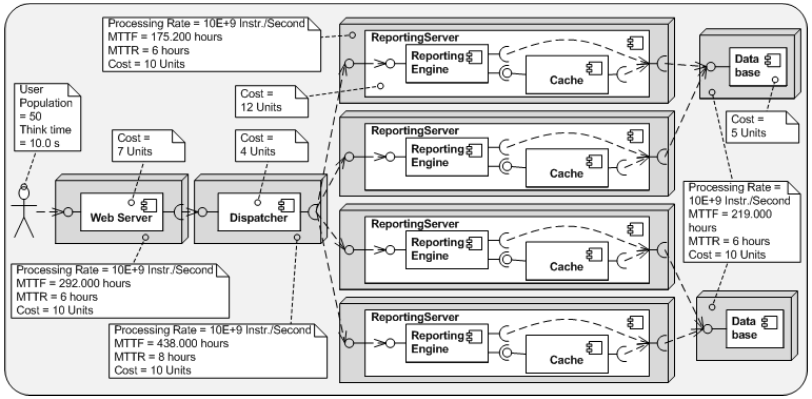
Case Study



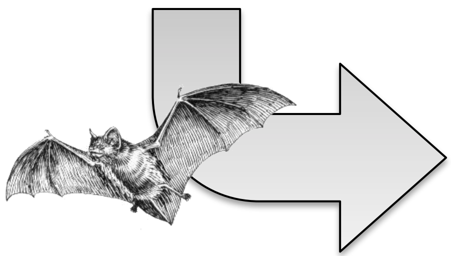
[MKB10]

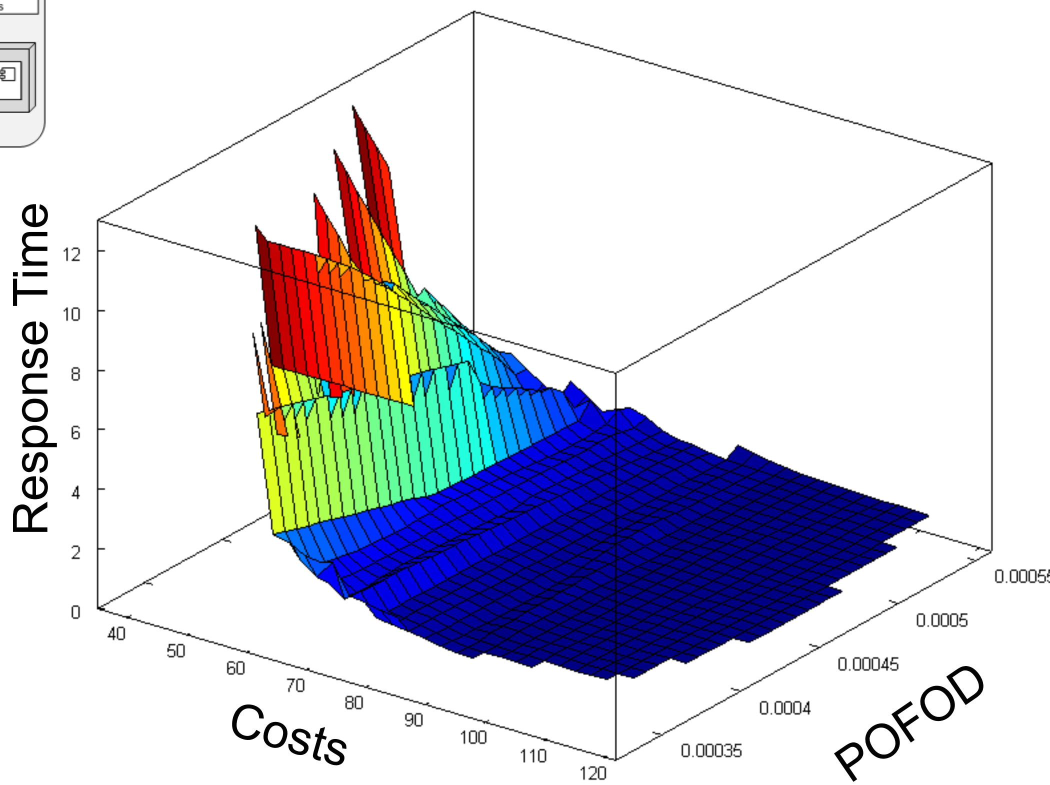
© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Results



- Component allocation
- Processing rates
- Component selection



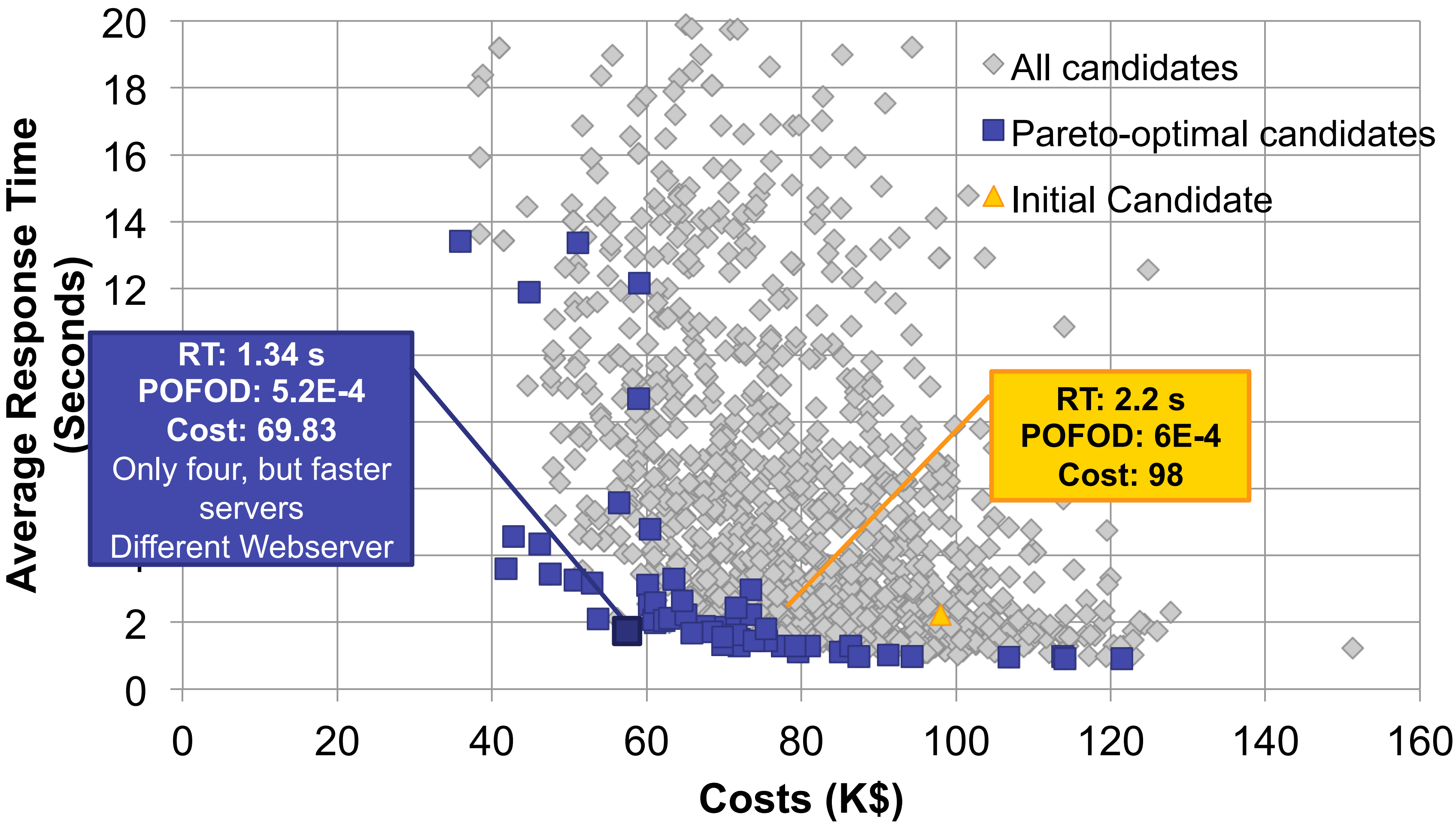


- 1235 candidates
- 58 Pareto optimal
- 8h running time

[MKB10]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Results



[MKB10]

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Other Approaches

ArcheOpteryx

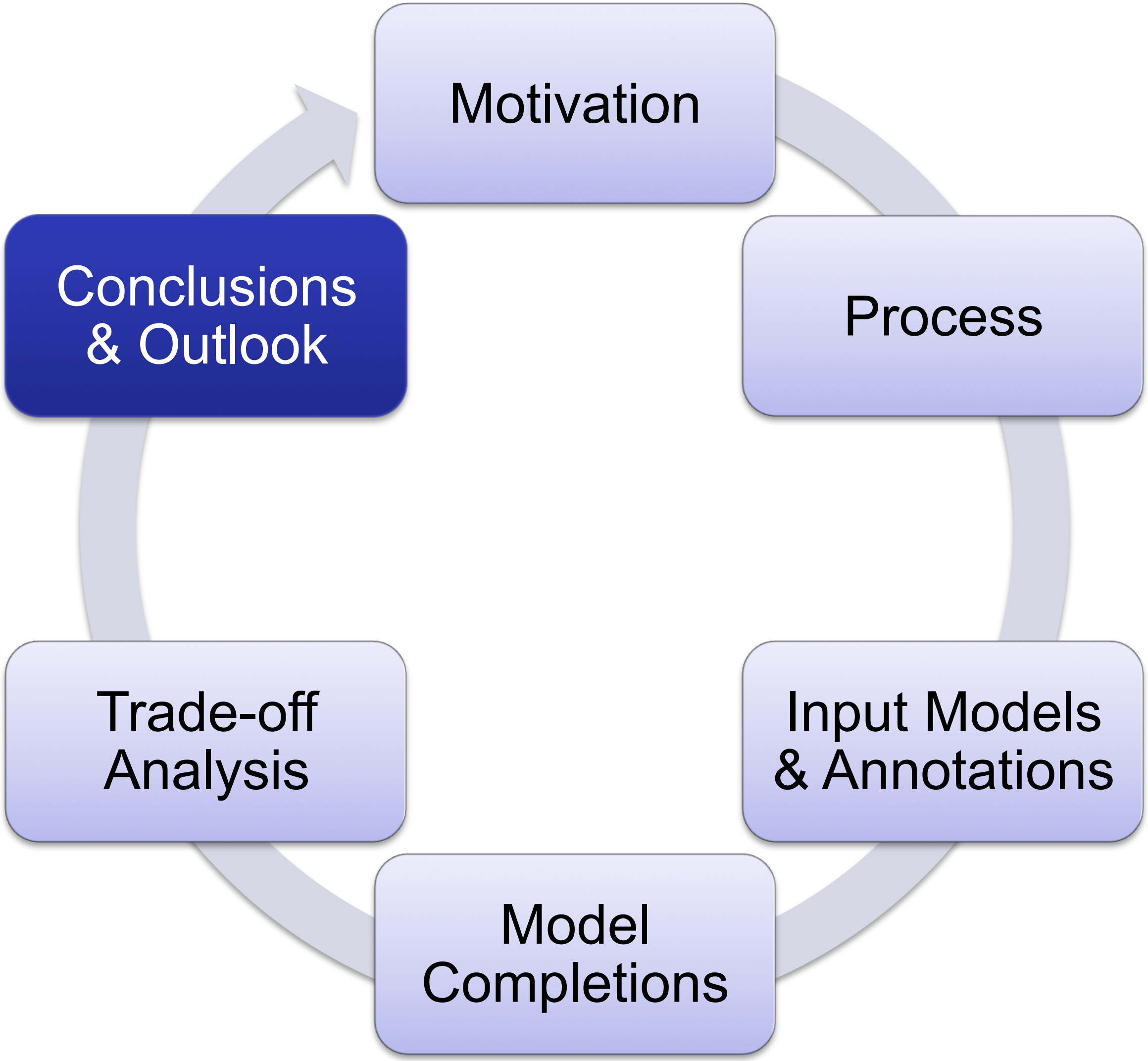
- E.g., Allocation of Embedded Components
- Reliability and Cost

Deep Compass

- Robocop Component
- Schedulability and costs

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Outline



© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

HEINZ NIXDORF INSTITUT
Universität Paderborn
Softwaretechnik
Jun.-Prof. Dr.-Ing. Steffen Becker

Lessons Learned

- Motivation for Model-Driven QoS analysis
- Process for model-driven analysis
- Transformations & Refinements
- Trade-off Analysis

Outlook

- New input models and system types
  - Mechatronic systems
  - Self-adaptive systems
- Automated selection of right abstraction level

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Questions, Comments, Feedback?

Palladio
Component Model

SFB 901
ON - THE - FLY COMPUTING

Jun.-Prof. Dr.-Ing. Steffen Becker
Heinz-Nixdorf-Institute, University of Paderborn, s-lab
Zukunftsmeile 1
33102 Paderborn, Germany
steffen.becker@upb.de

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn

Further Reading

- Model-Based Software Performance Analysis, Vittorio Cortellessa, Antinisca Di Marco, Paola Inverardi
- Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software, Connie Smith, Lloyd G. Williams
- The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modelling: Techniques for Experimental Design, Measurement, Simulation and Modelling, Raj Jain
- Queuing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications, Gunter Bolch, Stefan Greiner, Hermann de Meer

© Fachgebiet Softwaretechnik, Heinz Nixdorf Institut, Universität Paderborn