Karlsruhe Institute of Technology

Descartes research

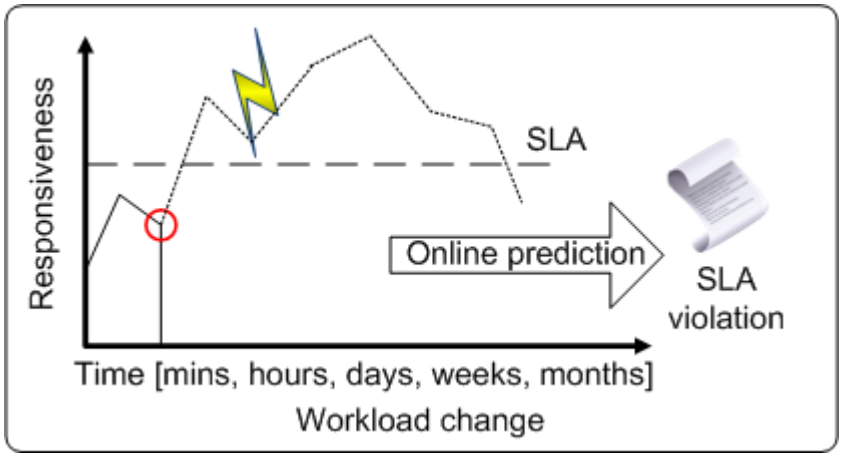
New Concepts for Online Architecture-Level Performance Models

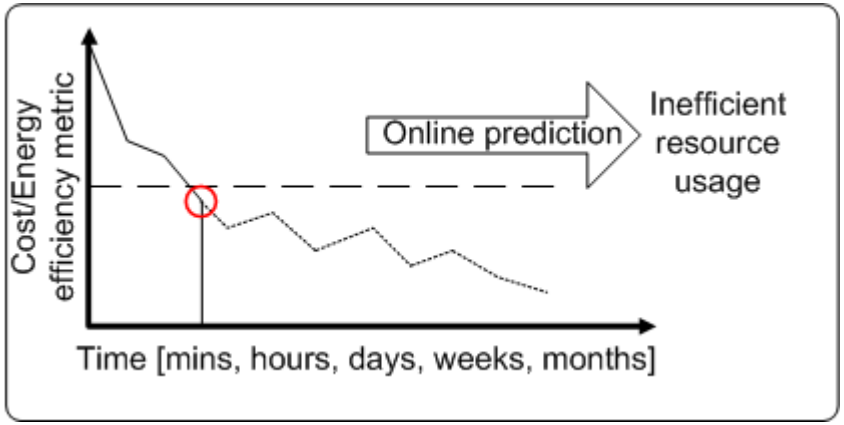
Fabian Brosig (fabian.brosig@kit.edu)

Open Excellence Workshop, December 4, 2012, Karlsruhe

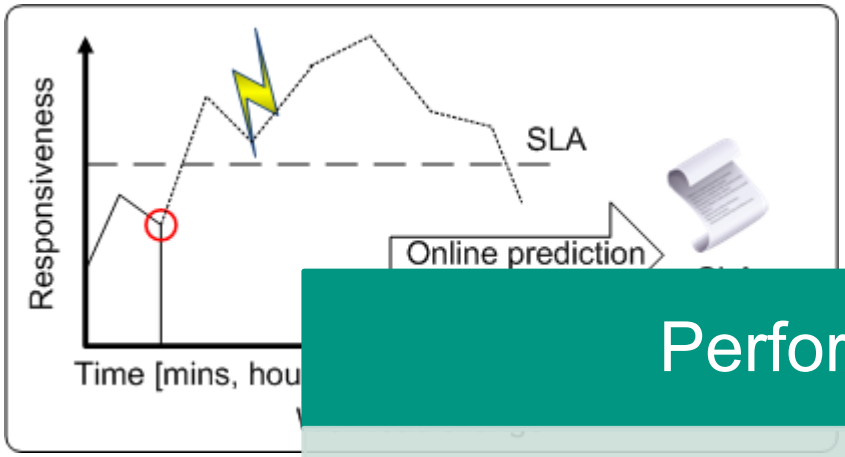
DESCARTES RESEARCH GROUP
INSTITUTE FOR PROGRAM STRUCTURES AND DATA ORGANIZATION, FACULTY OF INFORMATICS

Motivation:
Online Performance Prediction



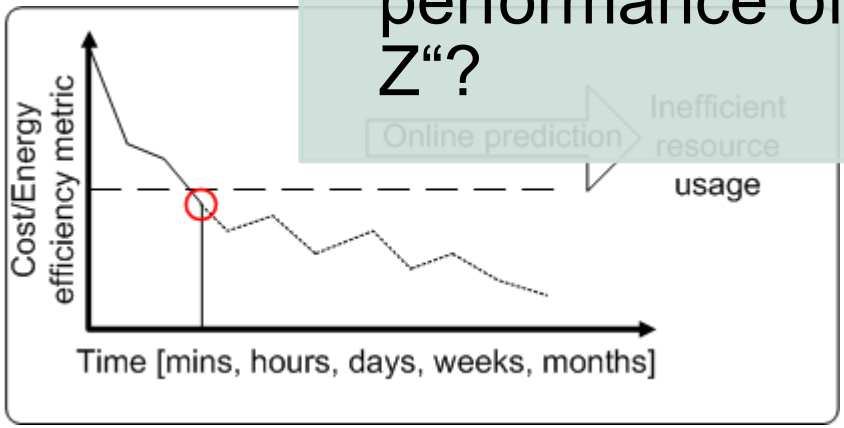


Motivation:
Online Performance Prediction

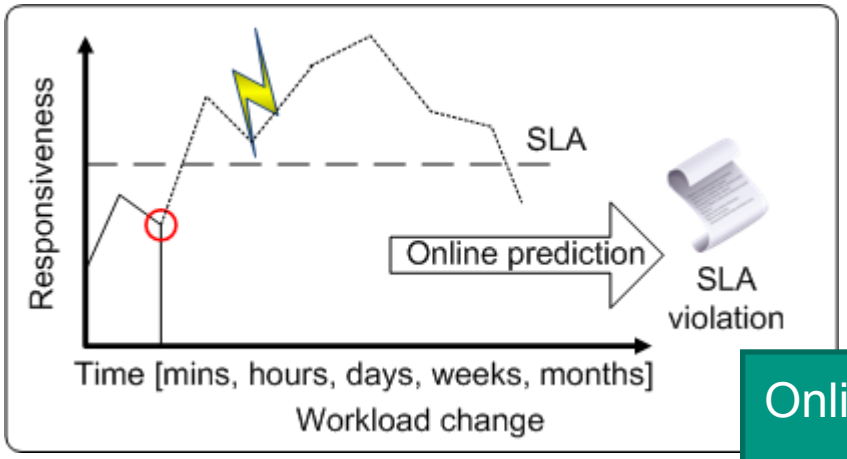


Performance Query

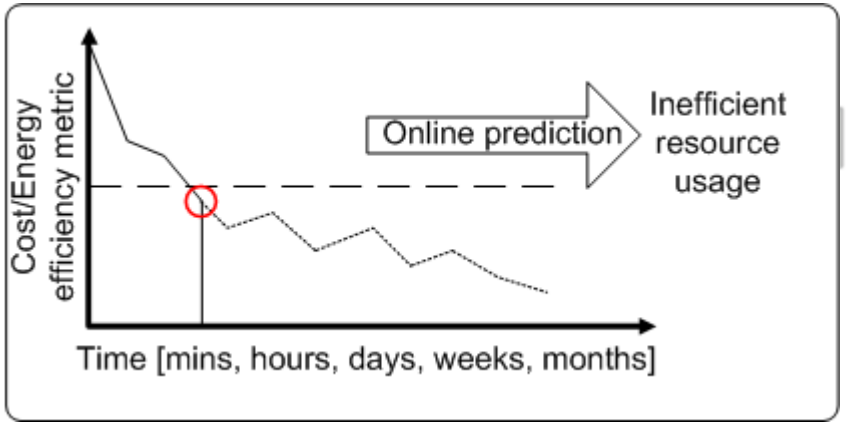
- Question:
  How does Change X affect the performance of Y in terms of metric Z“?



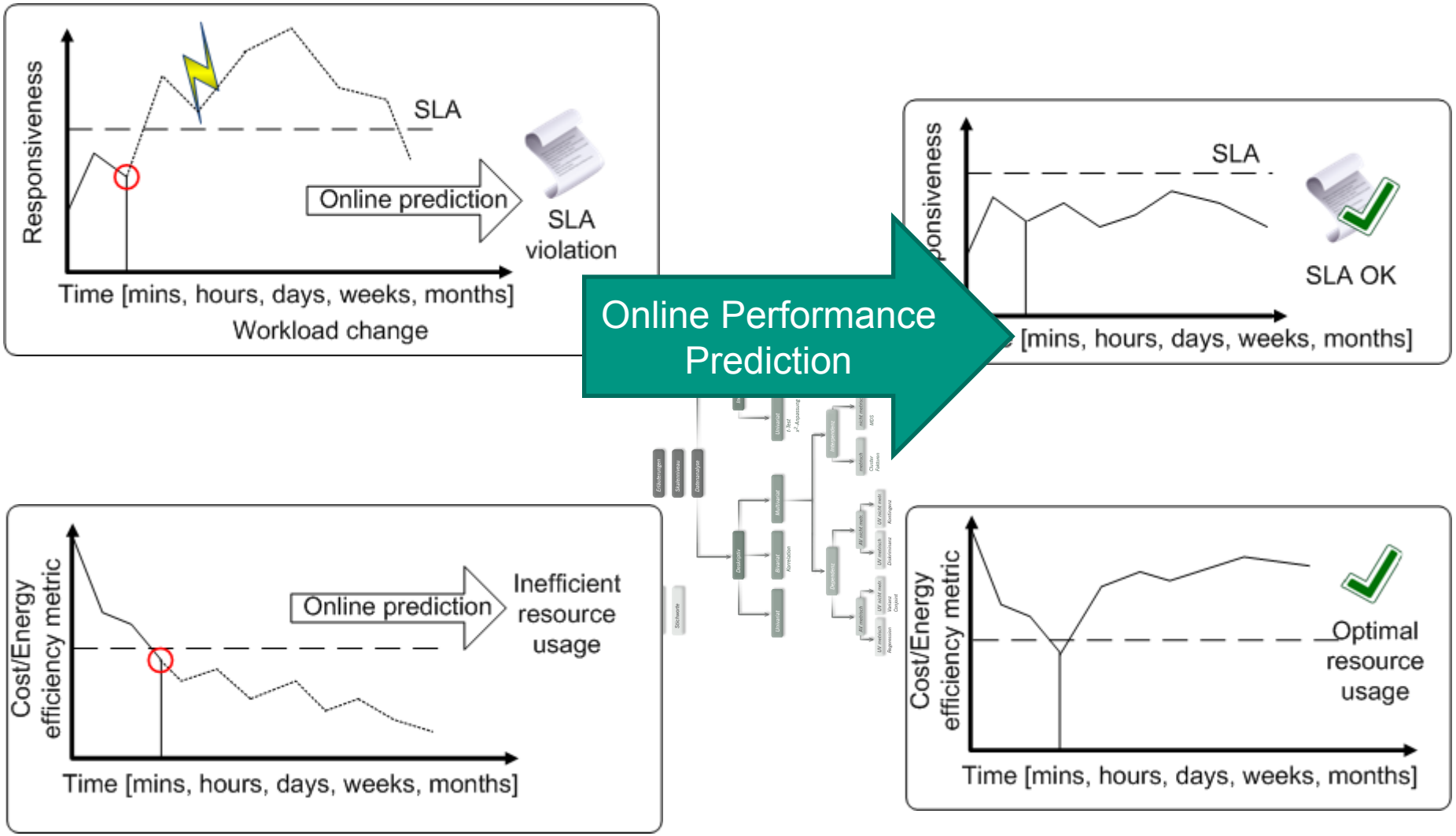
Motivation:
Online Performance Prediction



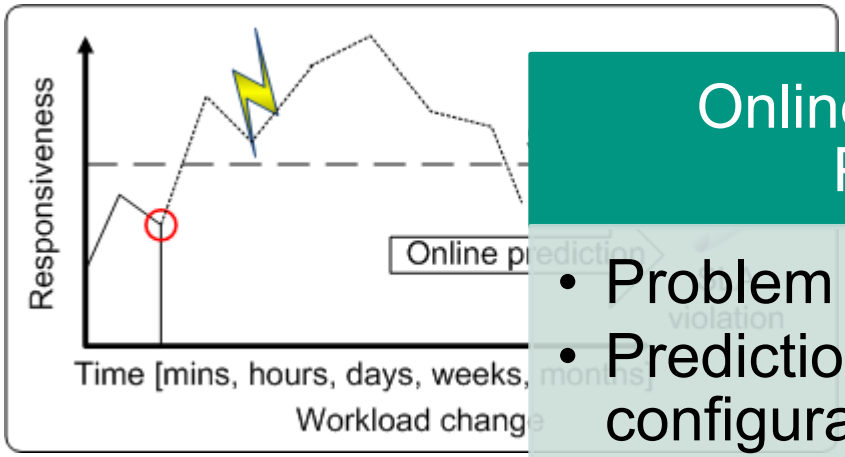
Online Performance Prediction



Motivation:
Online Performance Prediction



Motivation: Online Performance Prediction

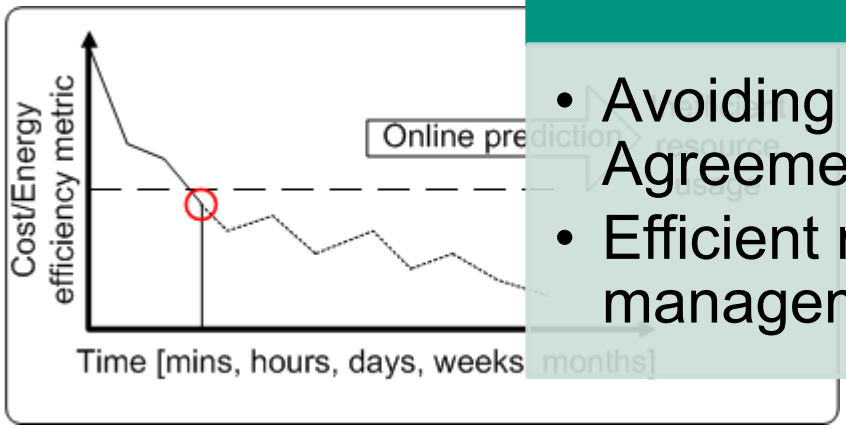


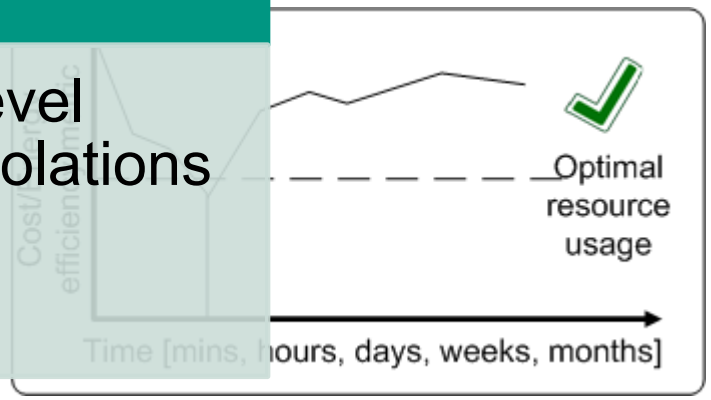
Online Performance Prediction

- Problem anticipation
- Predictions for different configurations

Precondition for

- Avoiding Service Level Agreement (SLA) violations
- Efficient resource management





State-of-the-Art: Models for performance prediction at run-time

- Based on, e.g., (layered) queueing networks or queueing petri nets (*predictive performance models*)
- Simple models that abstract the system at a very high level
- Services modeled as black boxes
- Software architecture and configuration not modeled
- Many restrictive assumptions imposed

[G. Pacifici et al], [A. D'Ambrogio et al], [G. Tesauro et al], [D. Menasce et al], [C. Adam et al], [I. Foster er al], [A. Othman et al], [P. Shivam et al], …

State-of-the-Art: Models for design&deployment time

- Software architecture models
  - annotated with descriptions of system's performance-relevant behavior
- Used at design time
  - Evaluate alternative system designs
- Used at deployment time
  - Predict the system performance for capacity planning purposes

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

Approach

Use architecture-level performance models for online performance prediction

Differences in the type of data available for model parameterization

Current approaches to model component execution context not suitable for use at run-time

Overhead in analyzing models

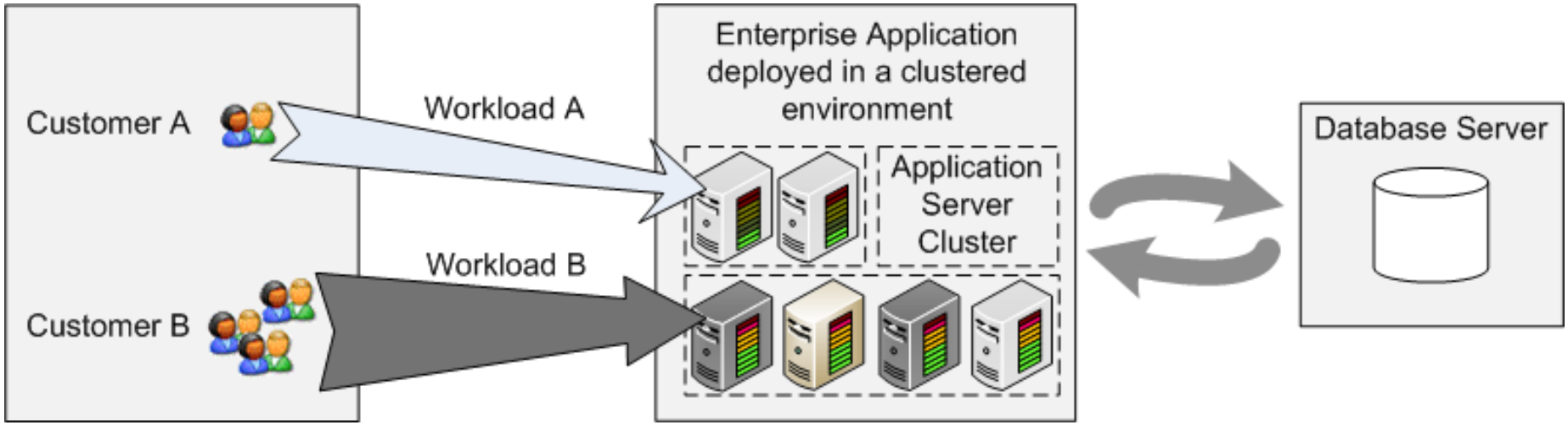
Outline

- Online performance prediction scenario

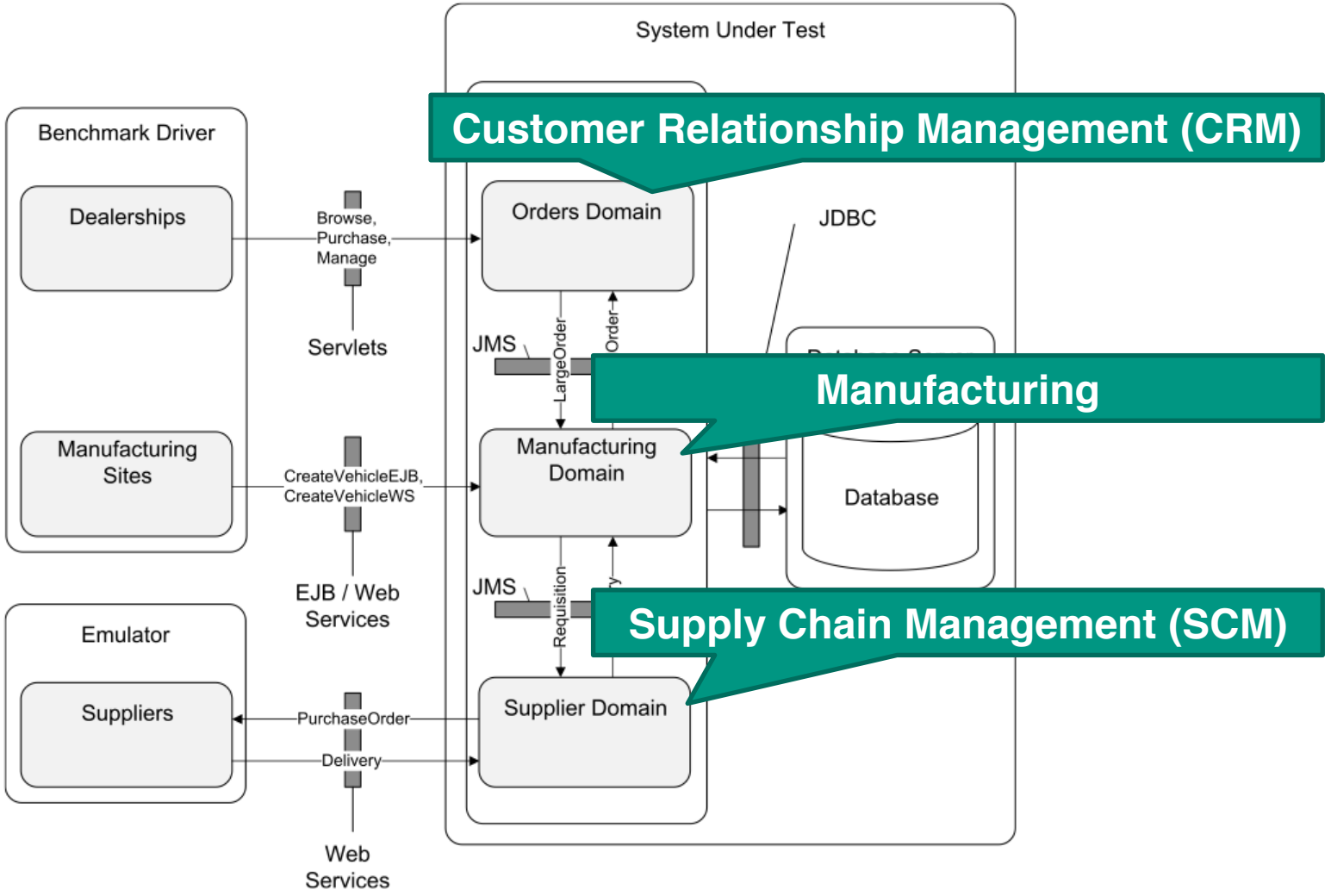
- Application Architecture Model
  - Abstractions to model service behavior
  - New modeling concepts for expressing parameter and context dependencies

ONLINE PREDICTION SCENARIO

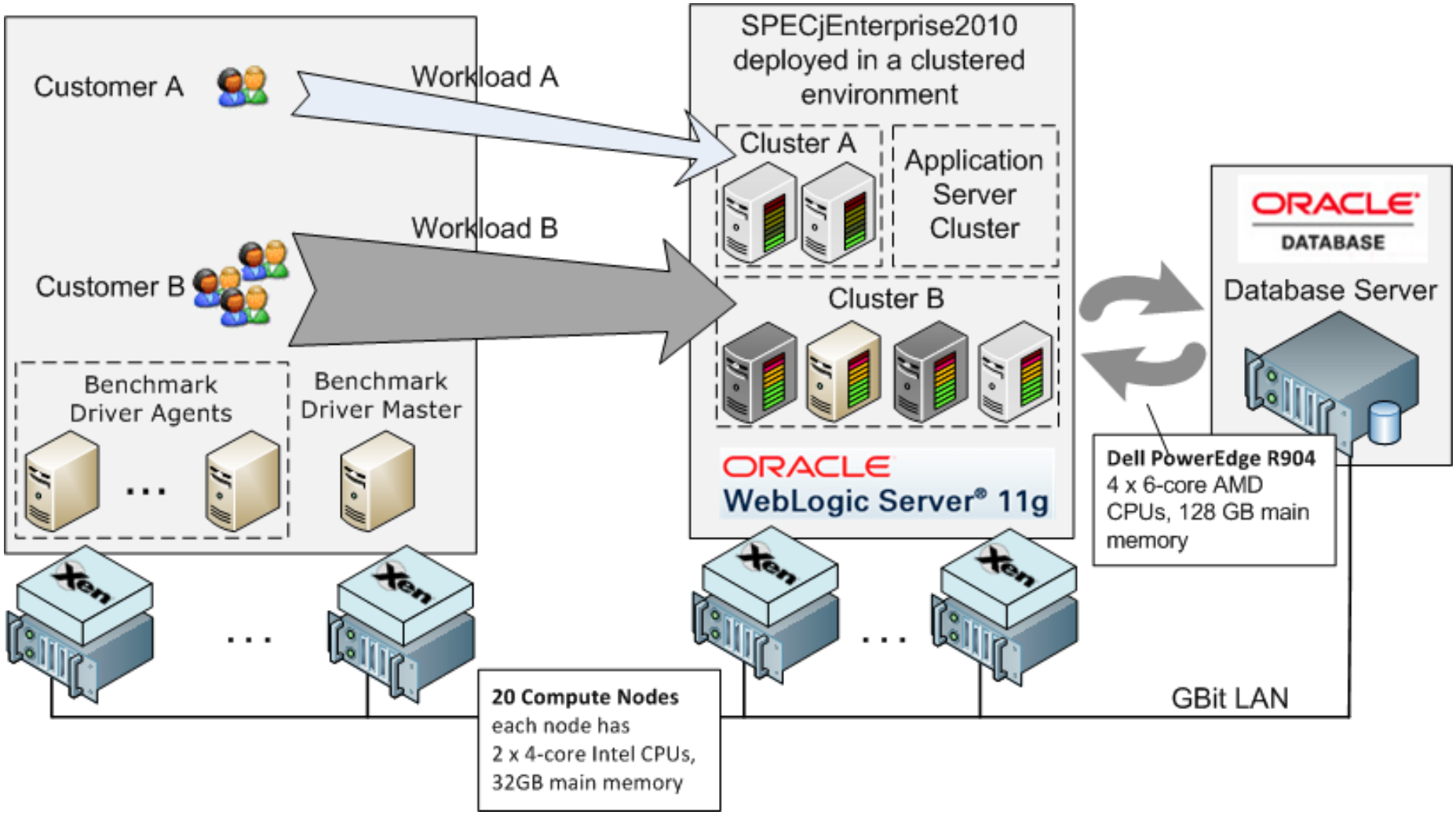
Online Performance Prediction Scenario



SPECjEnterprise2010

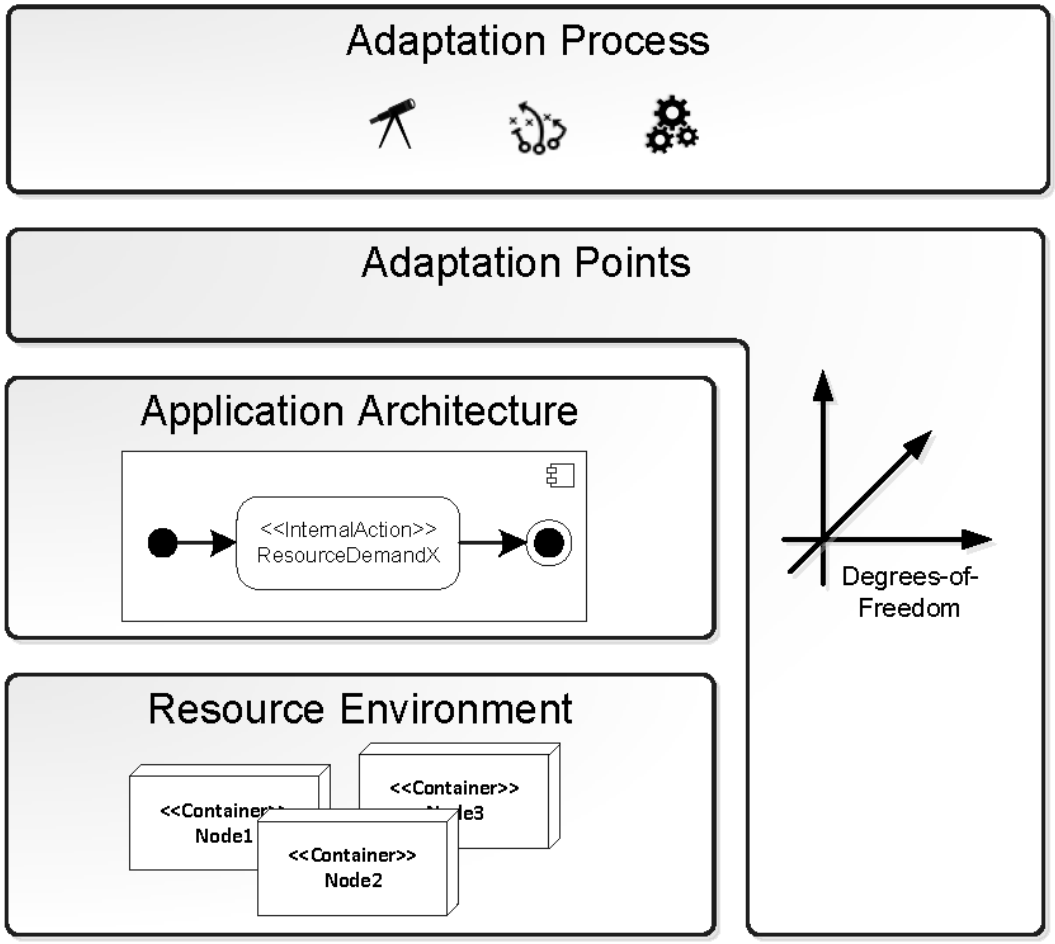


Online Performance Prediction Scenario

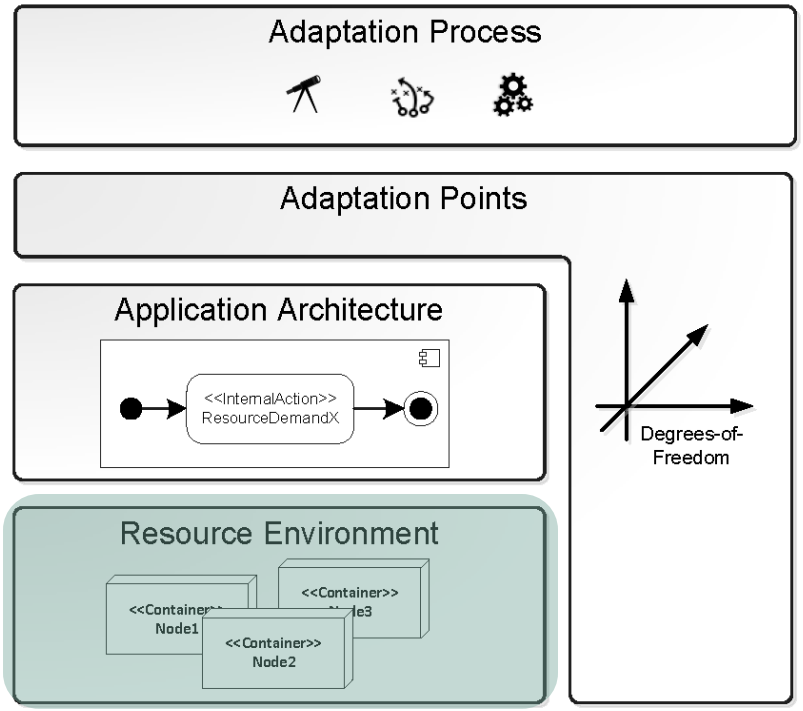


APPLICATION ARCHITECTURE MODEL

Descartes Meta-Model



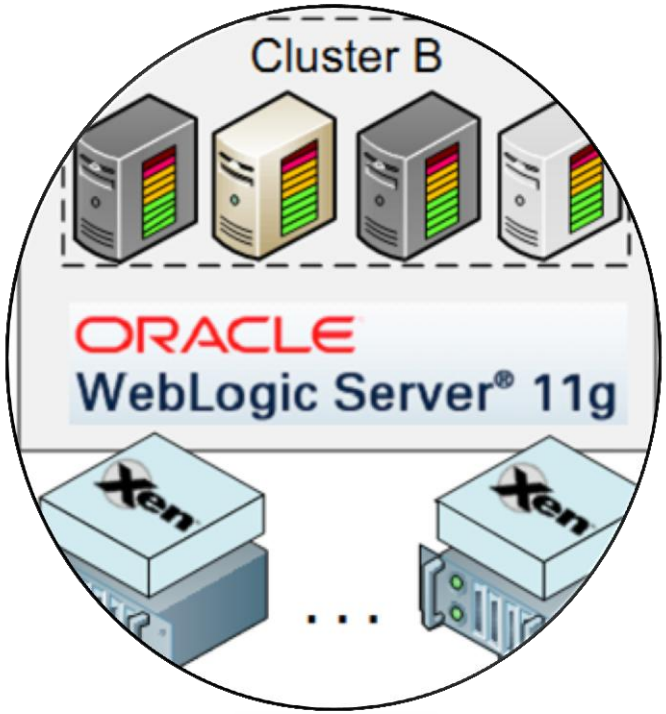
Descartes Meta-Model: Resource Environment

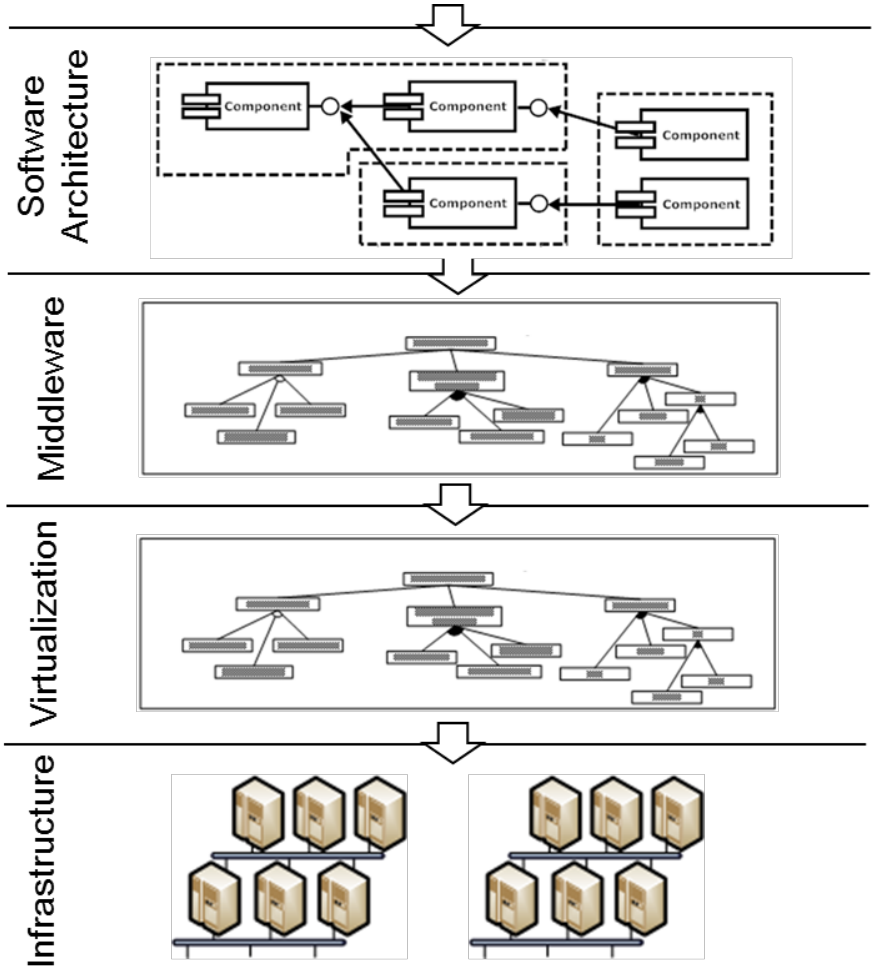


Further details in:

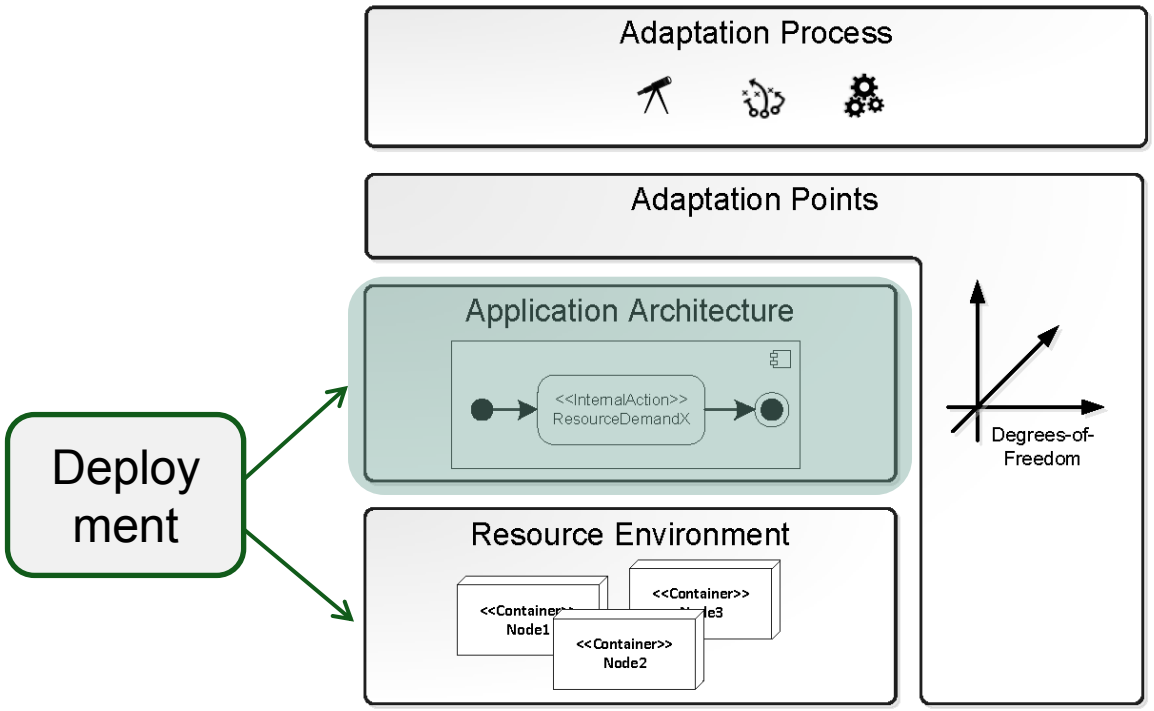
- N. Huber, F. Brosig and S. Kounev. **Modeling Dynamic Virtualized Resource Landscapes**. In Proceedings of the 8th ACM SIGSOFT International Conference on the Quality of Software Architectures (QoSA 2012), Bertinoro, Italy, June 25-28, 2012.

Resource Landscape: Example





DMM: Application Architecture



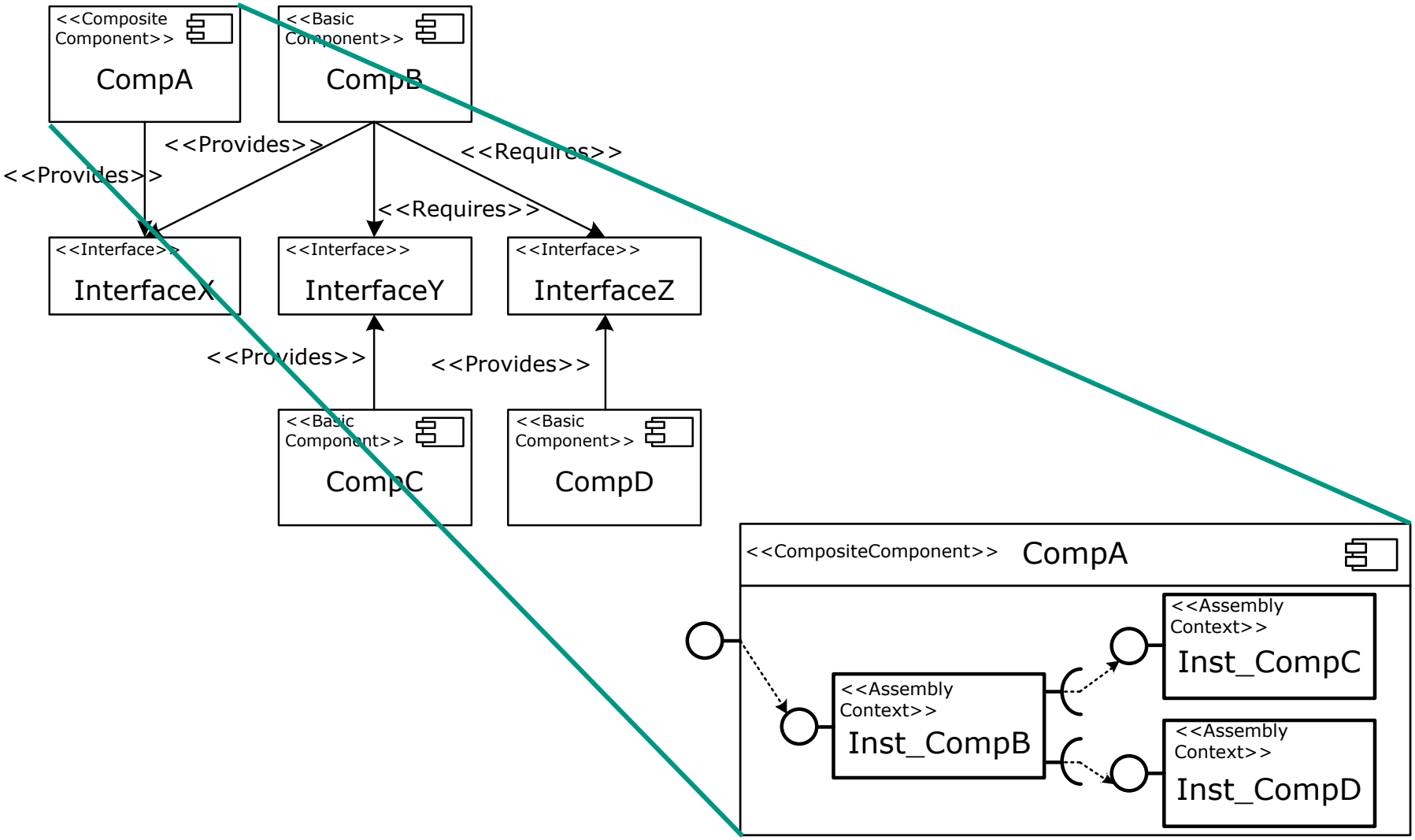
Further details in, e.g.:

- F. Brosig, N. Huber, and S. Kounev. **Modeling Parameter and Context Dependencies in Online Architecture-Level Performance Models**. In *Proceedings of the 15th ACM SIGSOFT International Symposium on Component Based Software Engineering (CBSE 2012), Bertinoro, Italy*, June 26-28, 2012.

Application Architecture Model

- Service Behavior Abstractions for Different Levels of Granularity
- Probabilistic Parameter Dependencies
- Deployment-Specific Resource Demands / Response Time

„Recap“ – (Composite) Components and Interfaces

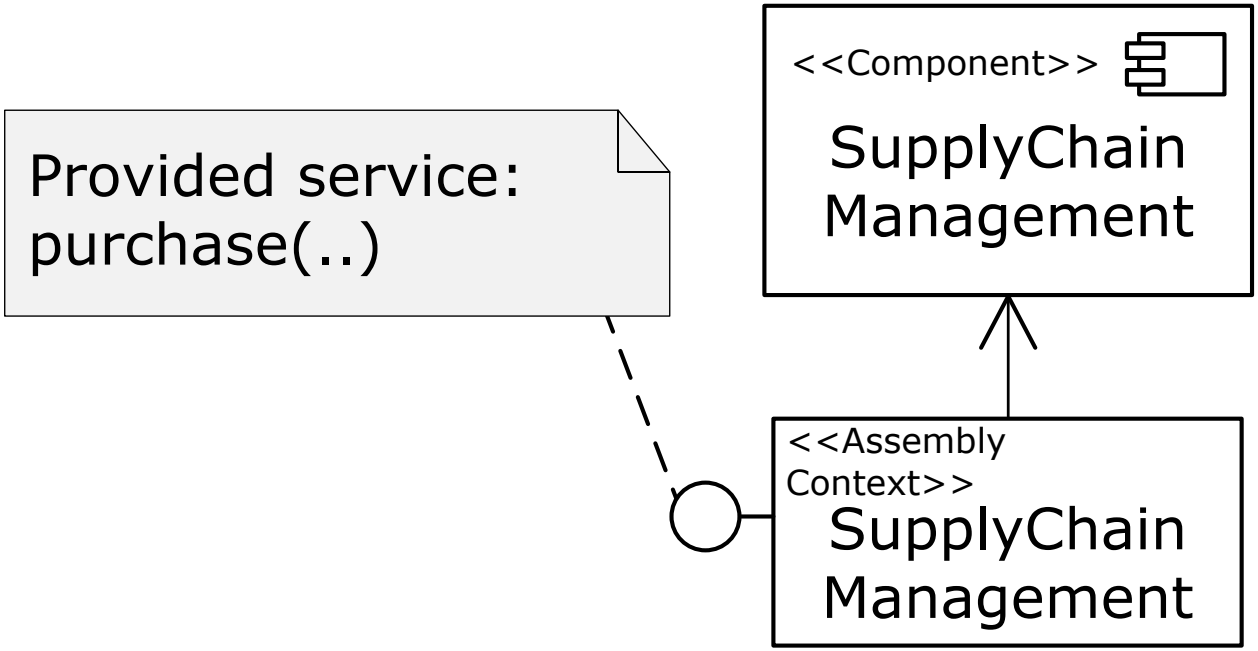


Application Architecture Model

- Service Behavior Abstractions for Different Levels of Granularity
- Probabilistic Parameter Dependencies
- Deployment-Specific Resource Demands / Response Time

Service Behavior Abstractions

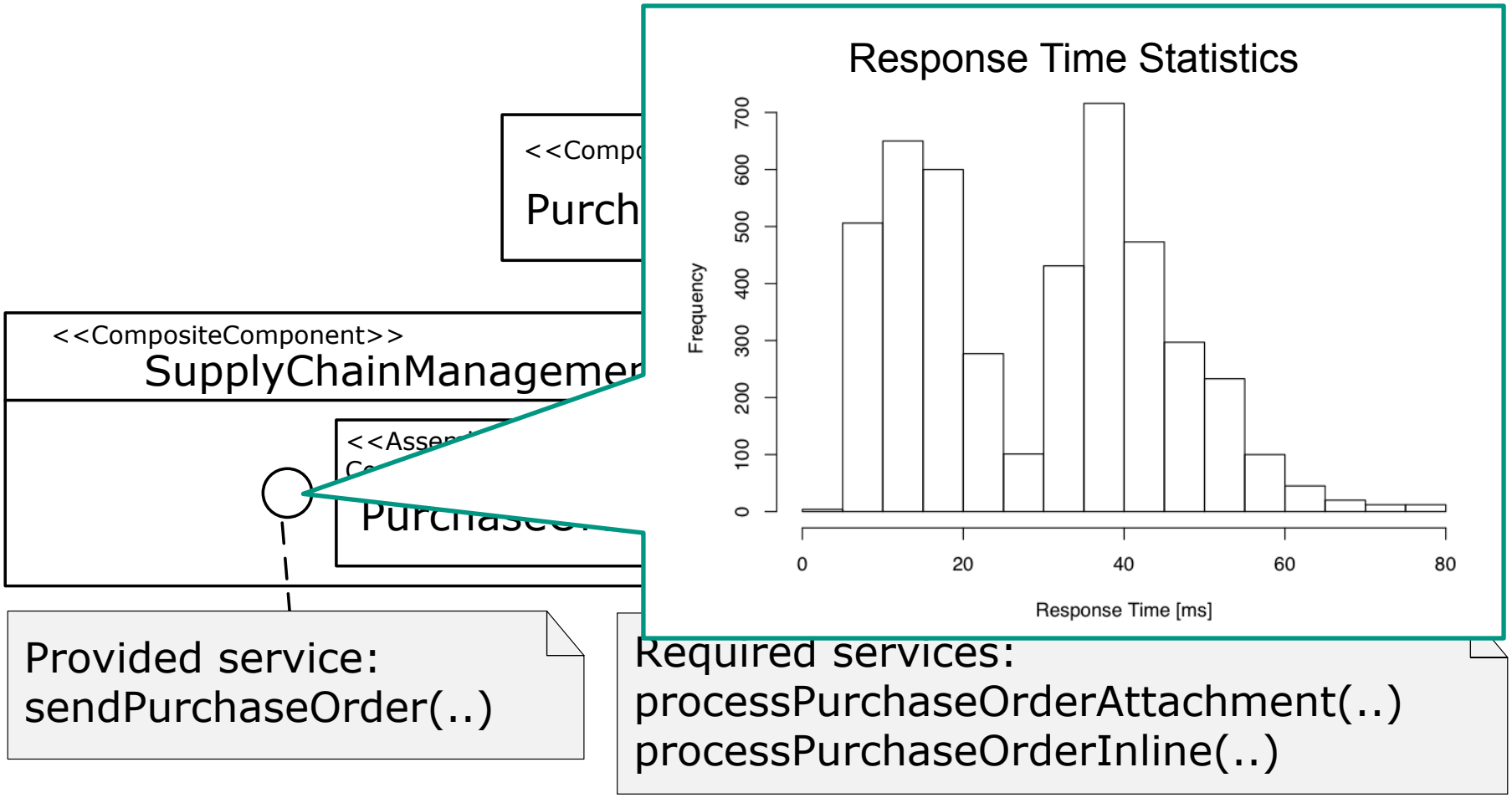
- What do we want to predict?
- What performance-relevant data is available?



Outsourced Service?

3rd Party Component?

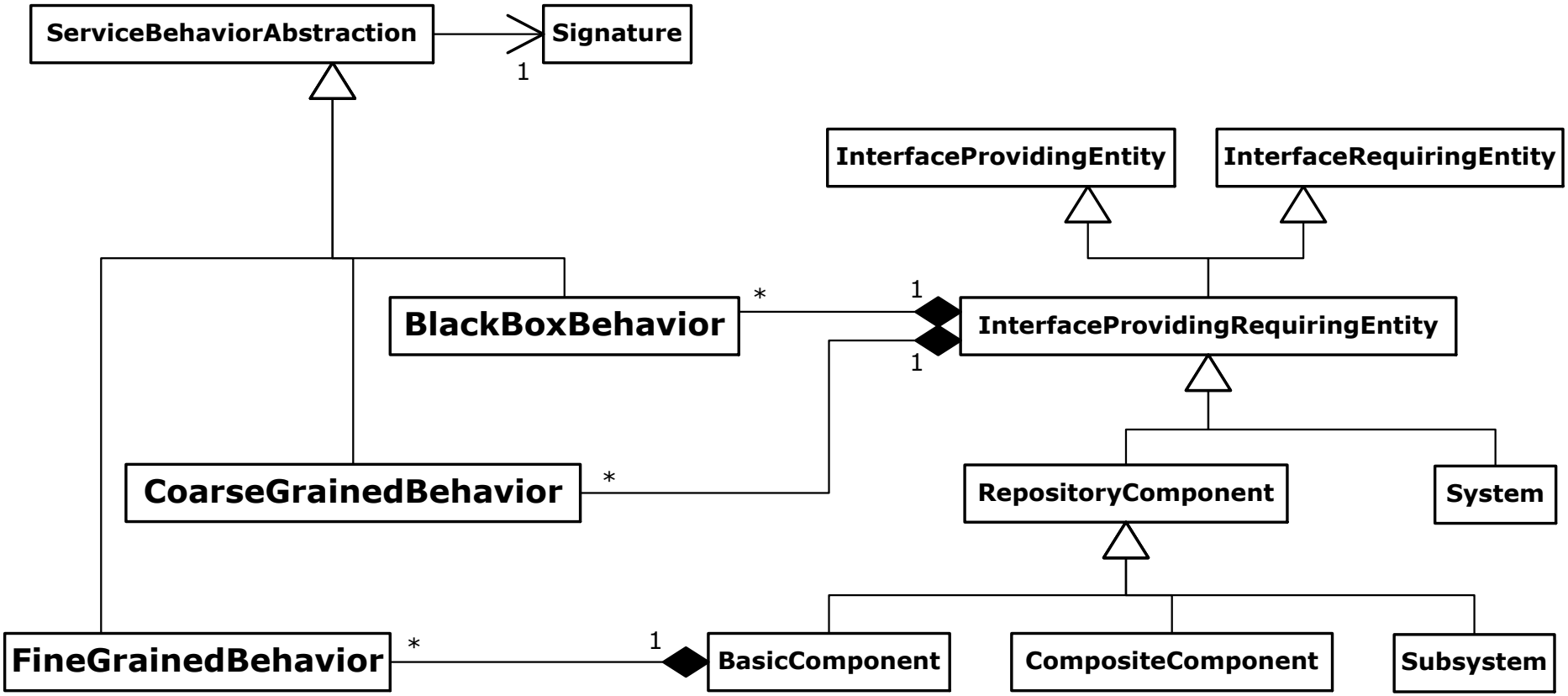
Service Behavior Abstractions



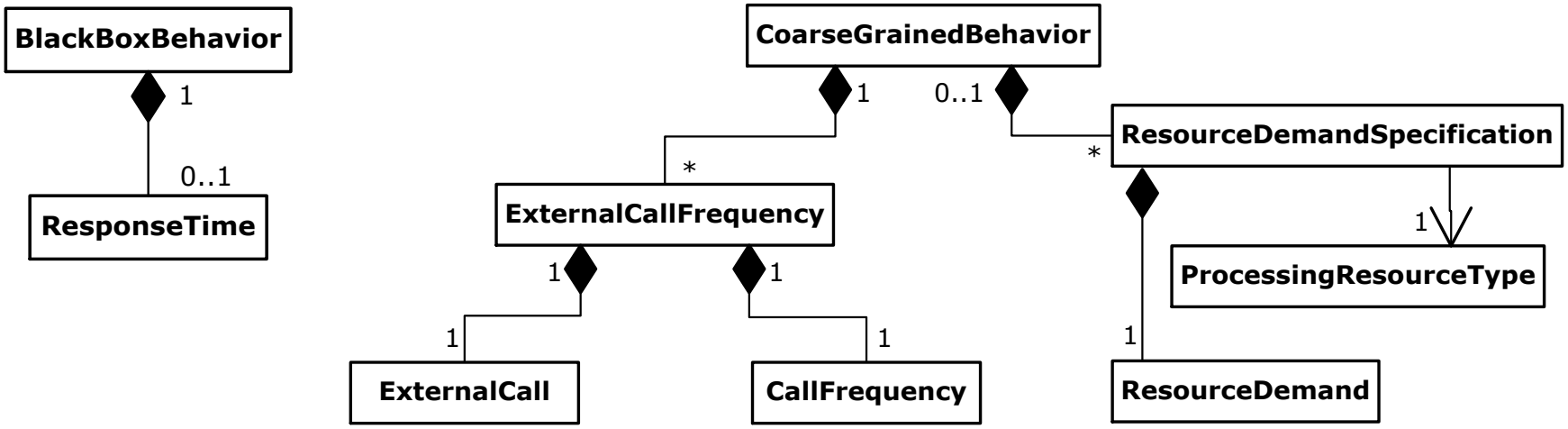
Service Behavior Abstractions

- BlackBoxBehavior
  - No information about resources, resource demands, control flow, call frequencies,…
- CoarseGrainedBehavior
  - Information at component boundary level (external services, resource consumption,...)
- FineGrainedBehavior
  - Information about component-internals (control flow, resource demands, parametric dependencies,...)

Service Behavior Abstractions - Metamodel

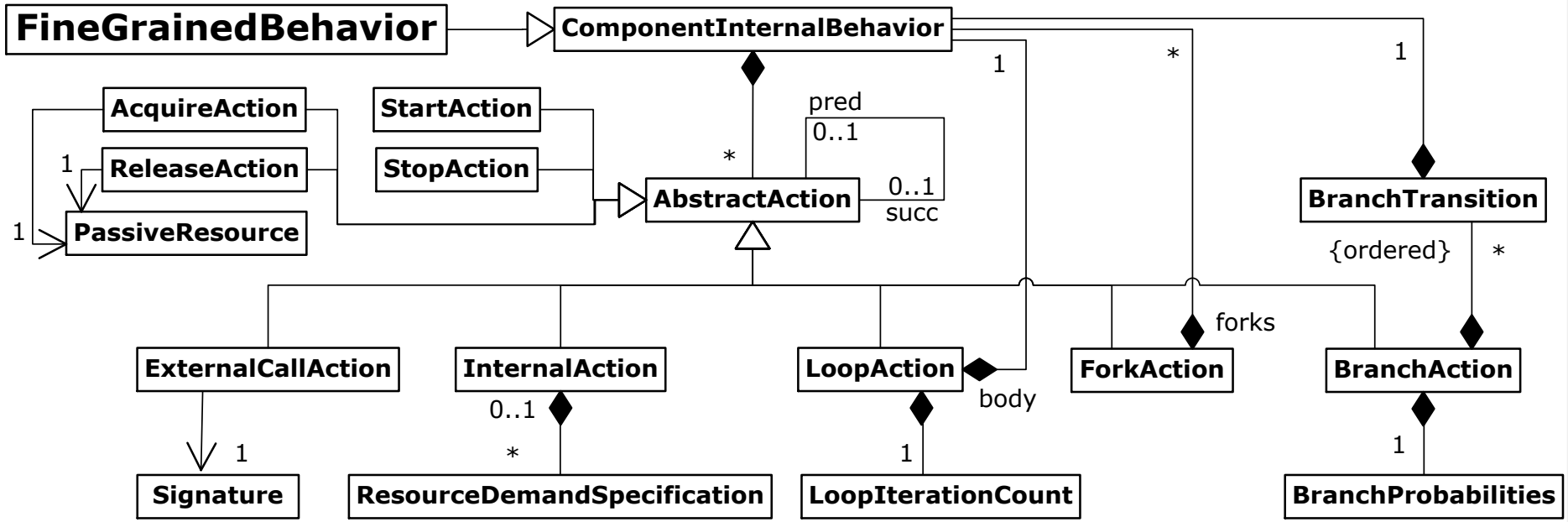


Service Behavior Abstractions - Metamodel



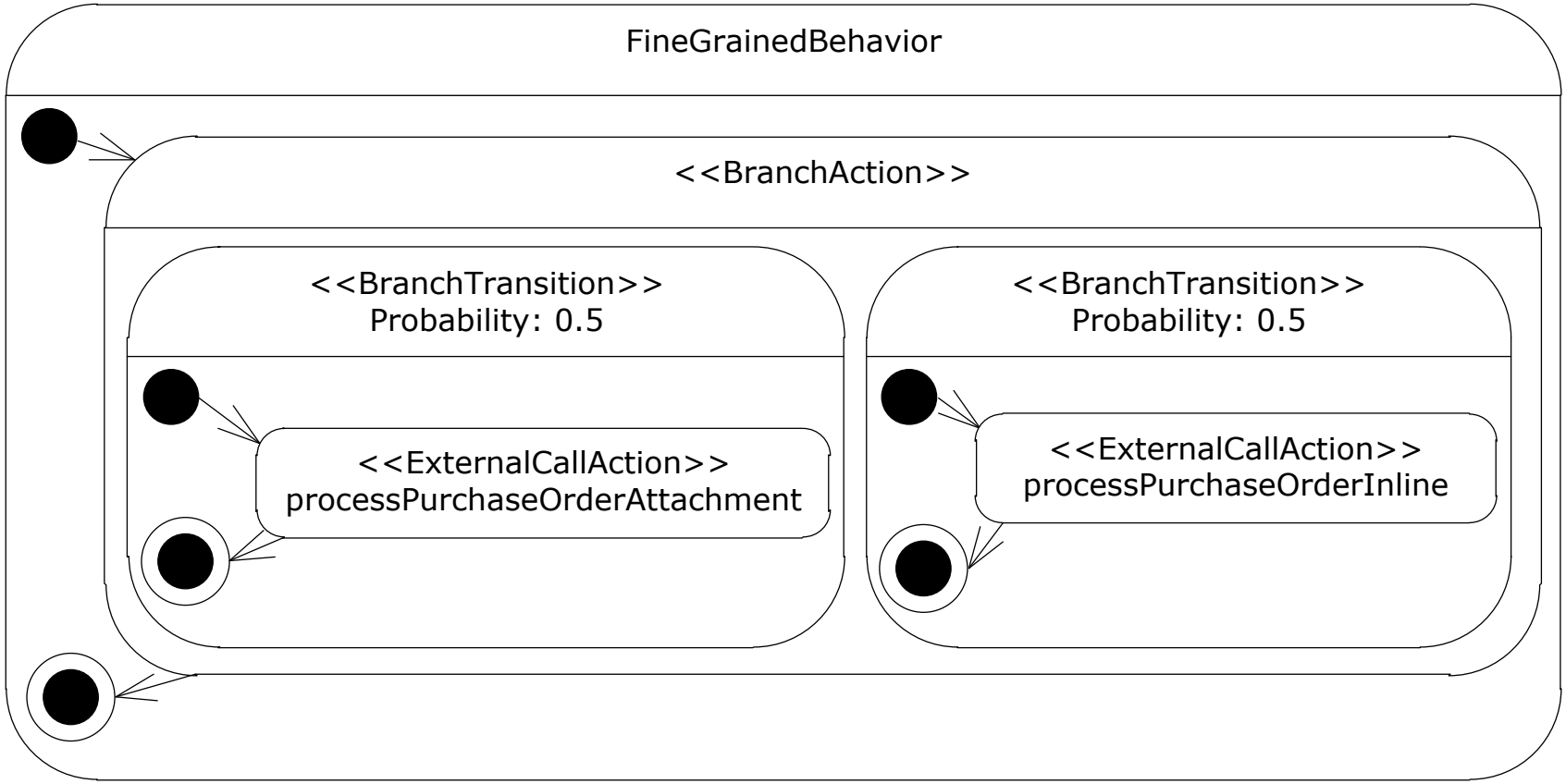
Service Behavior Abstractions - Metamodel

- Control flow abstraction



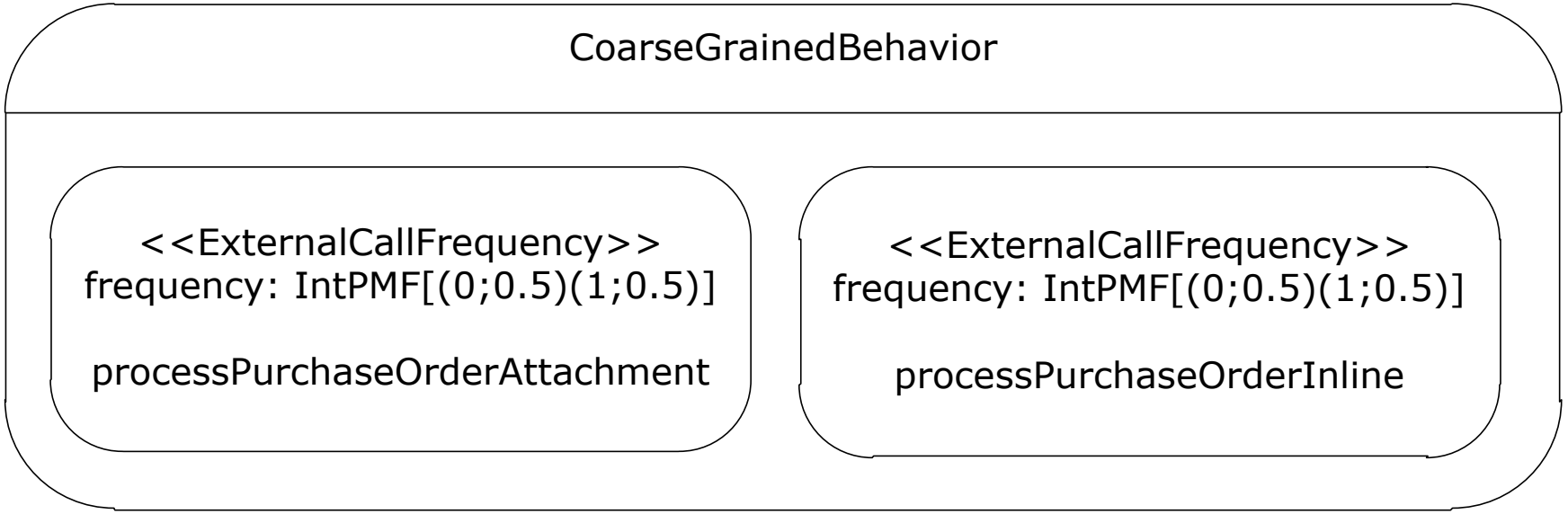
Service Behavior Abstraction - Example

- sendPurchaseOrder: FineGrainedBehavior

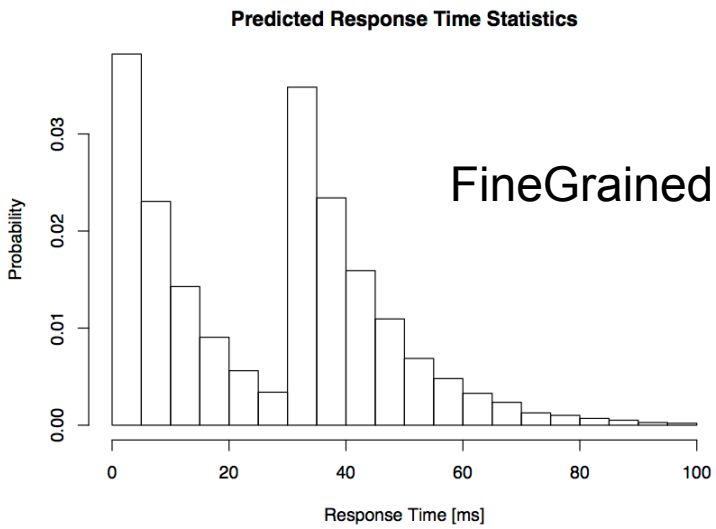


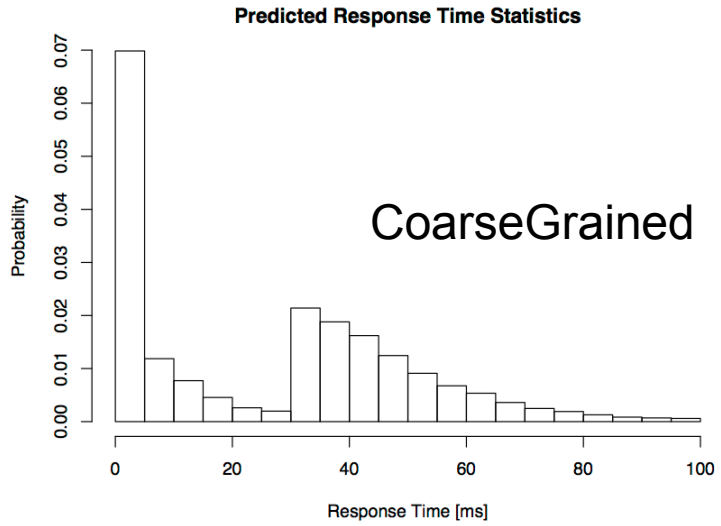
Service Behavior Abstraction - Example

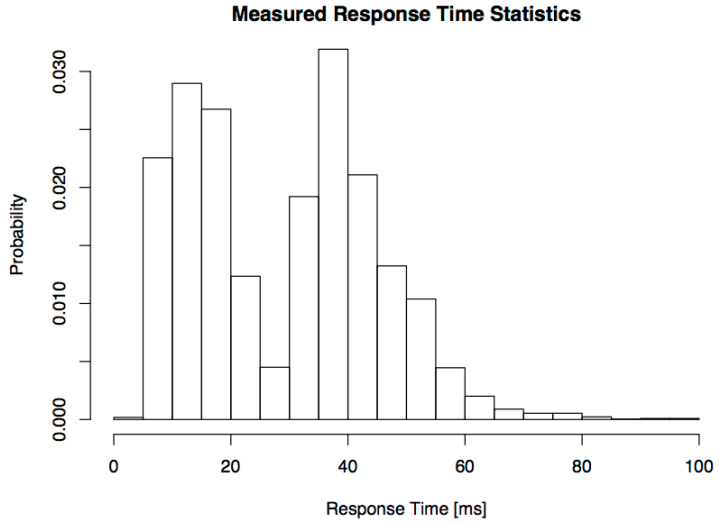
- sendPurchaseOrder: CoarseGrainedBehavior



Service Behavior Abstraction - Example

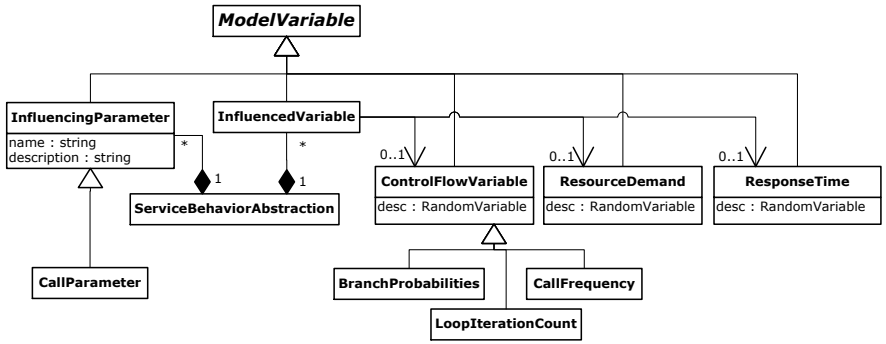




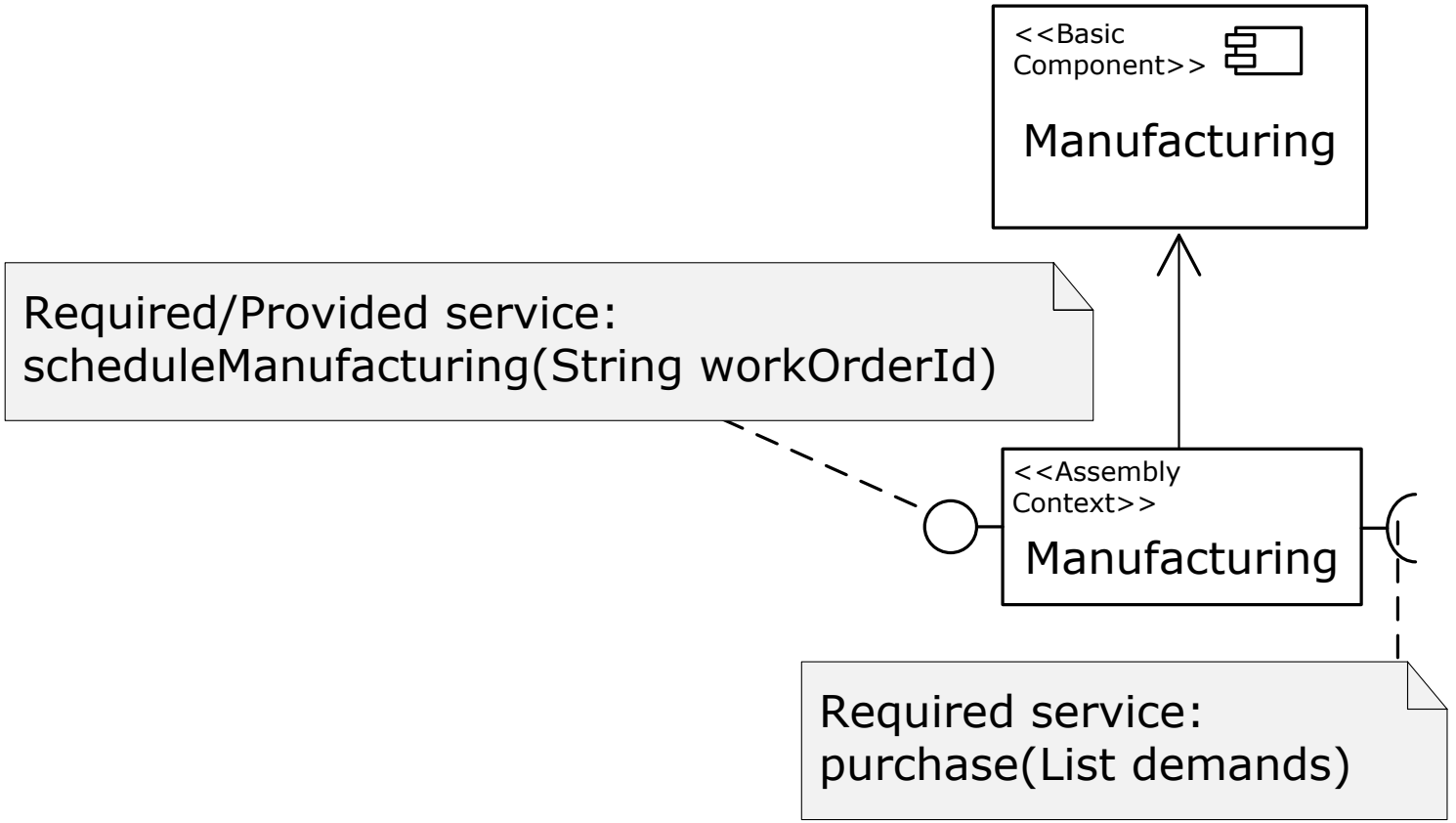


Parameter and Context Dependencies

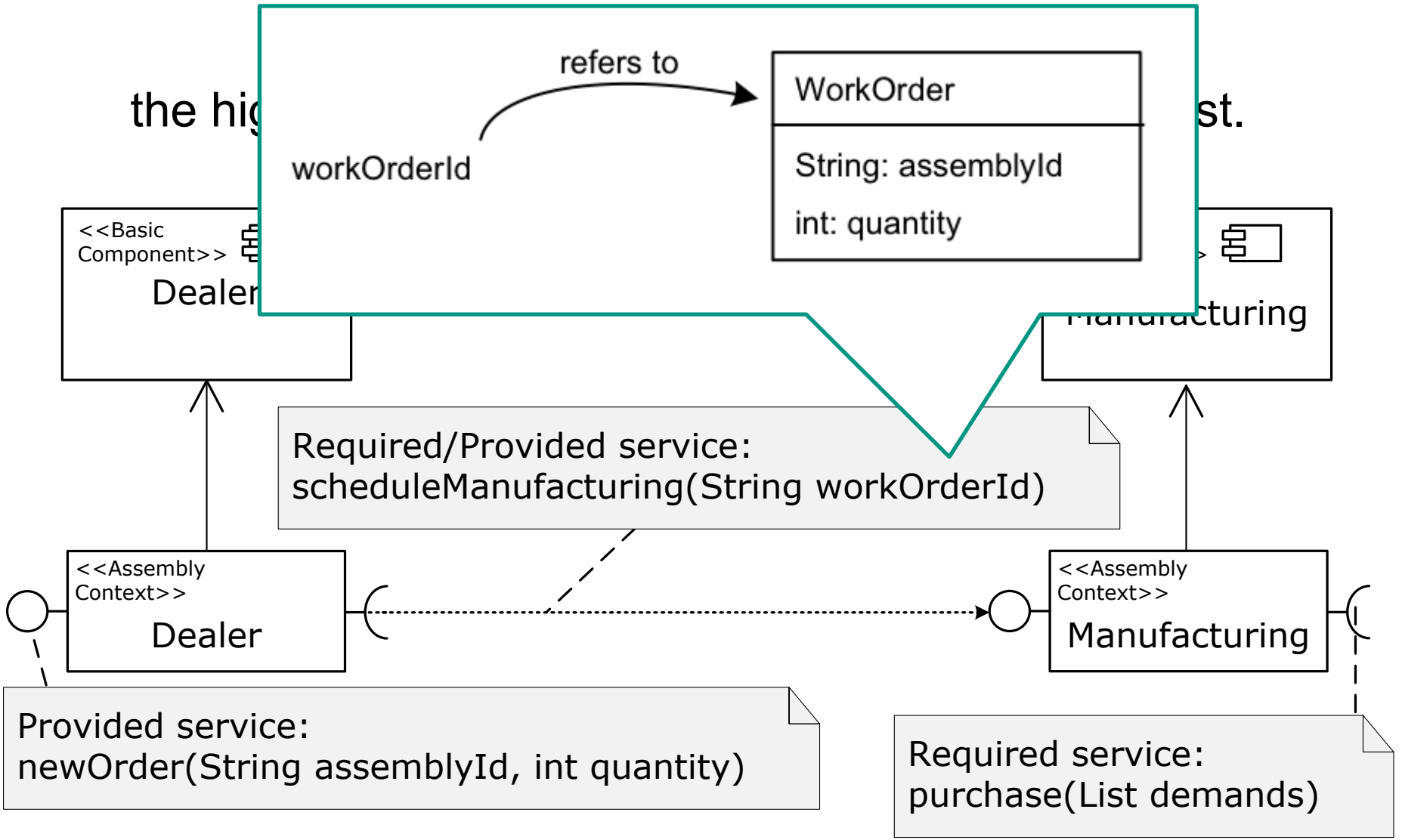
- Characterize dependencies probabilistically
- Influencing parameters
  - Service input parameters
- Influenced variables
  - Loop iteration numbers (FineGrainedBehavior)
  - Branch probabilities (FineGrainedBehavior)
  - Call frequencies (CoarseGrainedBehavior)
  - Resource demands (FineGrainedBehavior, CoarseGrainedBehavior)
  - Response times (BlackBoxBehavior)



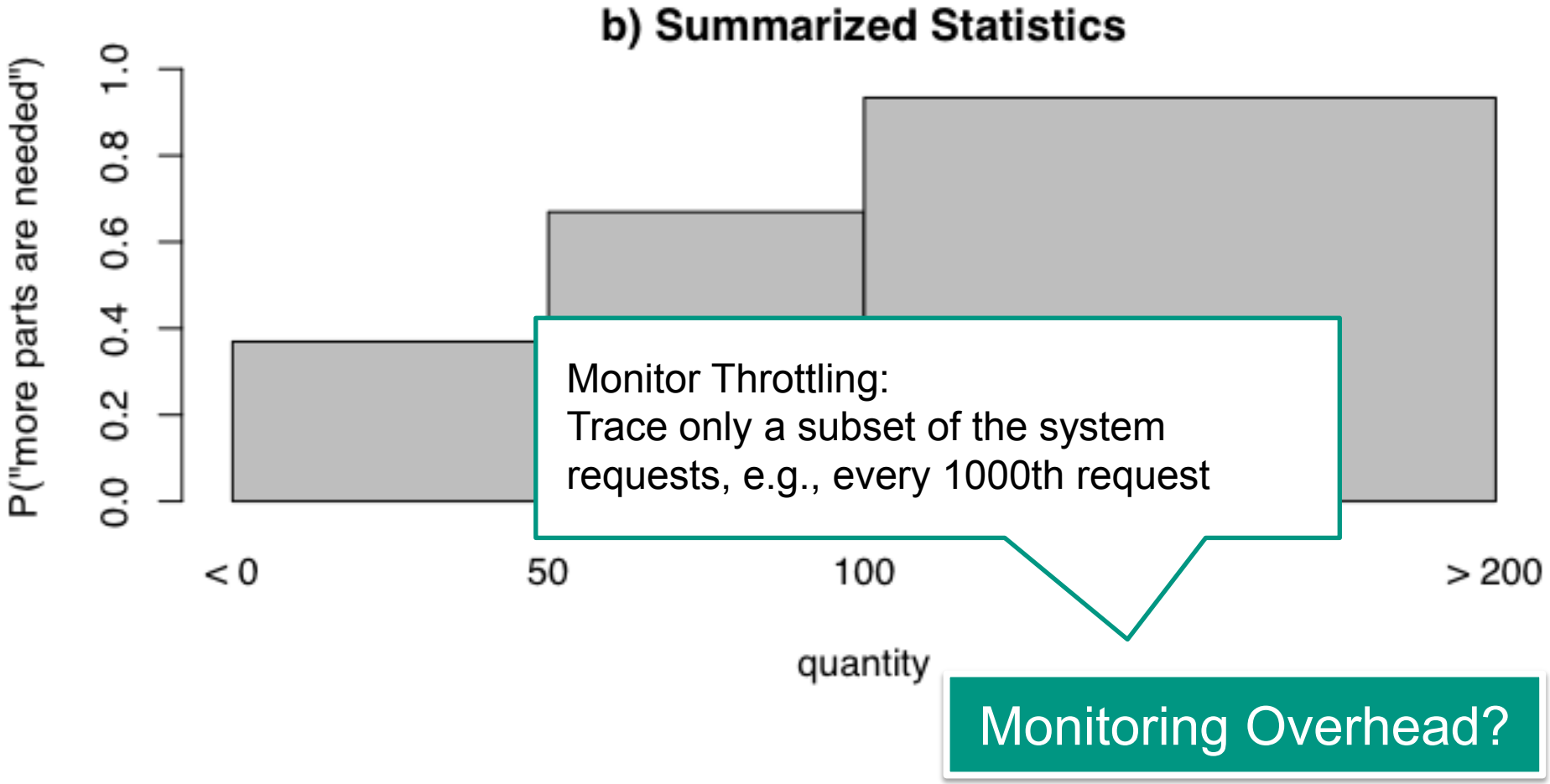
Parameter Dependencies



Parameter Dependencies



Parameter Dependencies

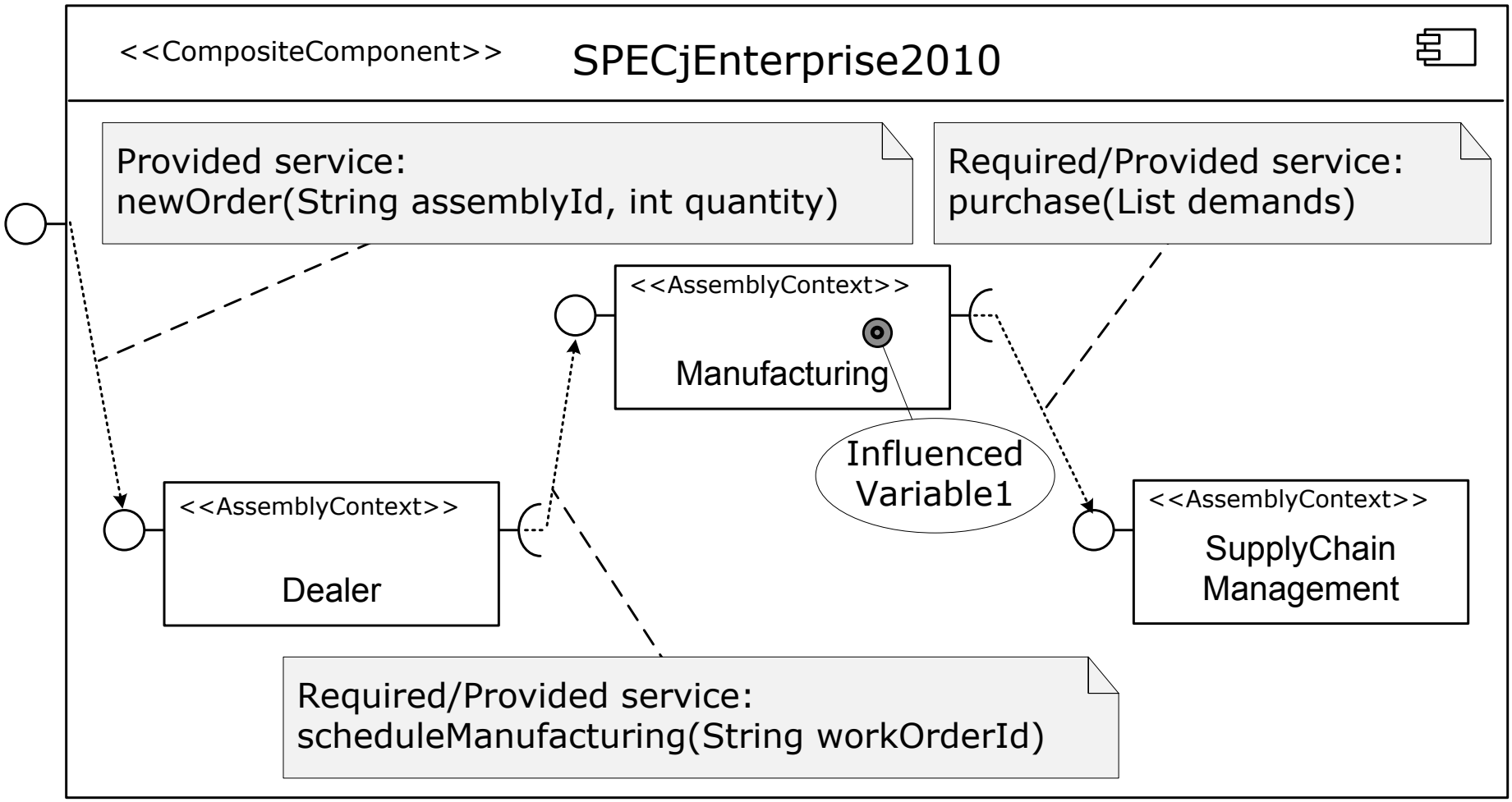


Parameter Dependencies

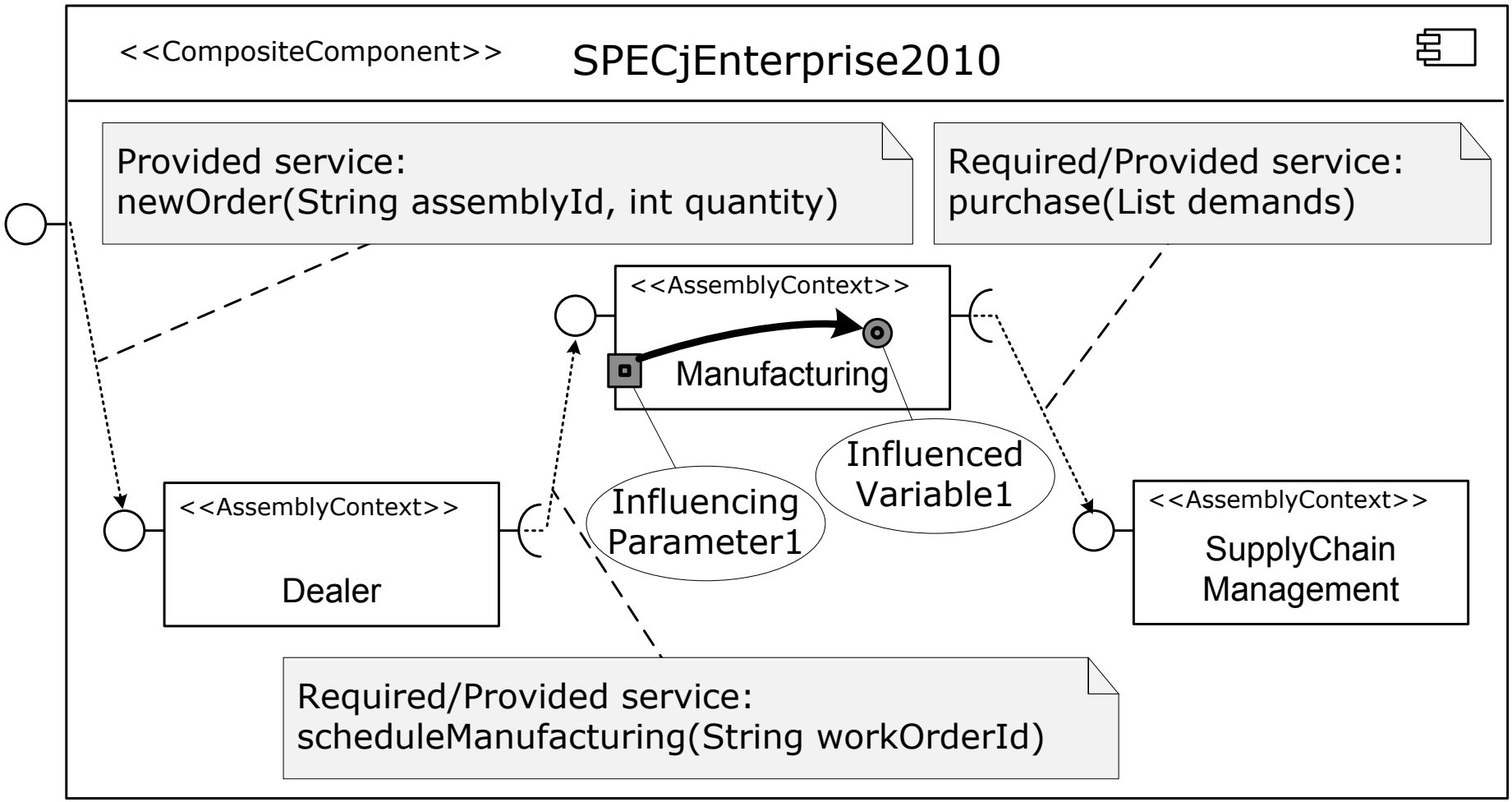
Workload:	Measurements:		Predictions:	
scheduleManufacturing	U_WLS	R_avg [ms]	U_WLS	R_avg [ms]
quantity in [0..10]	0.166	17.2	**0.365**	19.4
quantity in [0..200]	0.627	32.9	[illegible]07	28.3

With P(„more parts are needed“) = 0.10:
U_WLS = **0.196**, R_avg = 16.8 ms

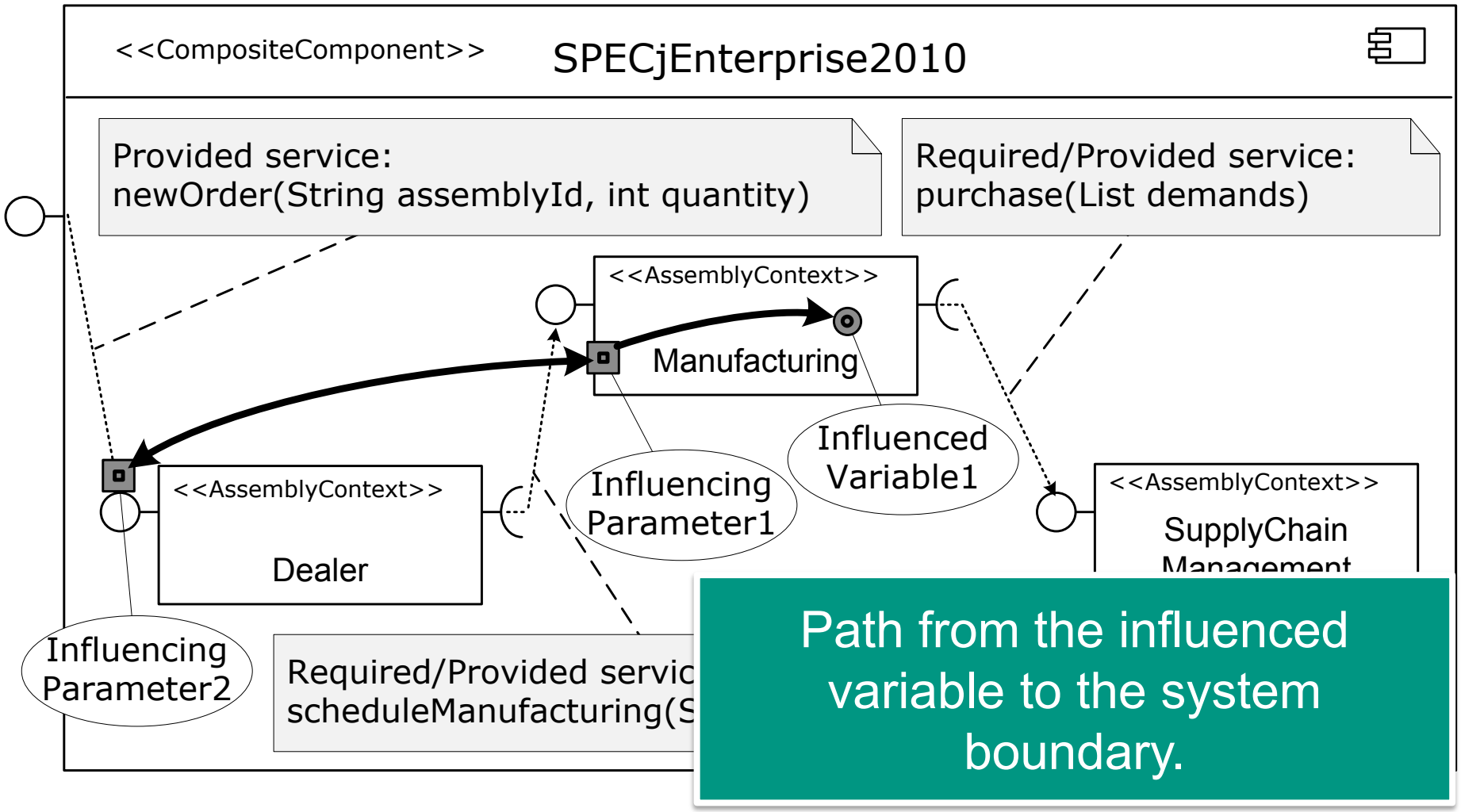
Parameter Dependencies - Approach



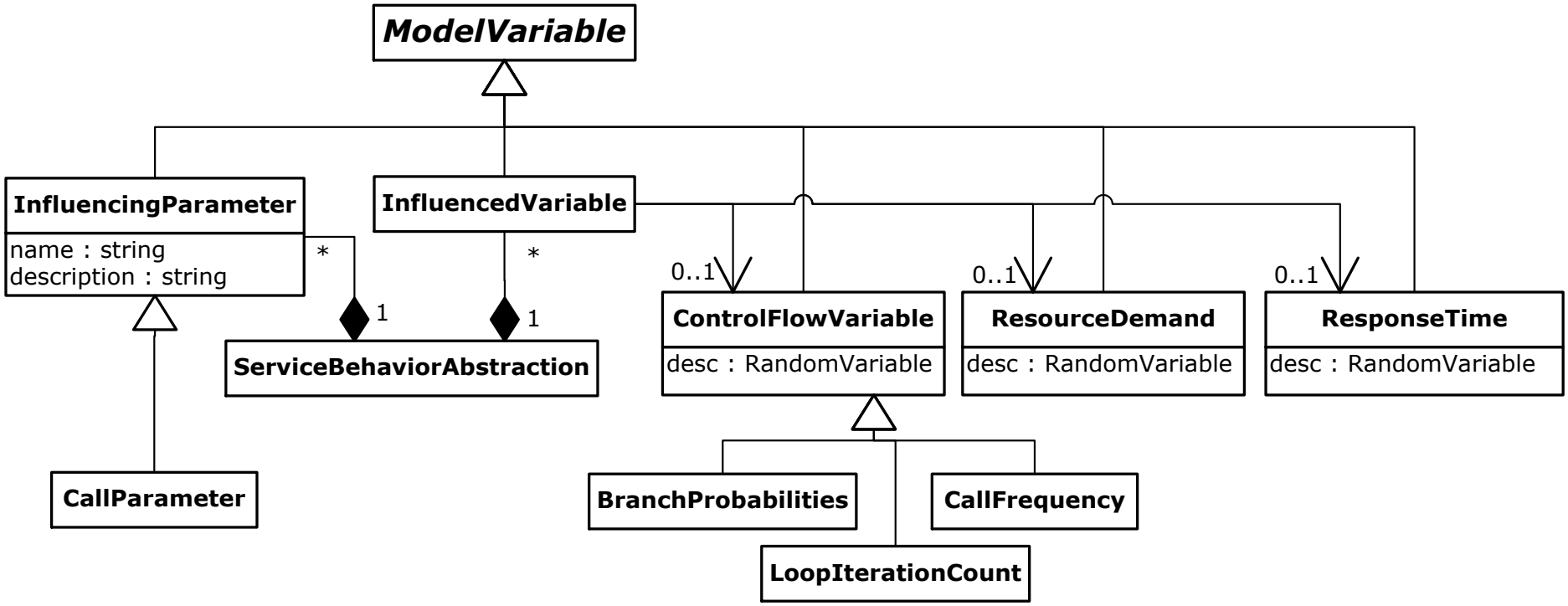
Parameter Dependencies - Approach



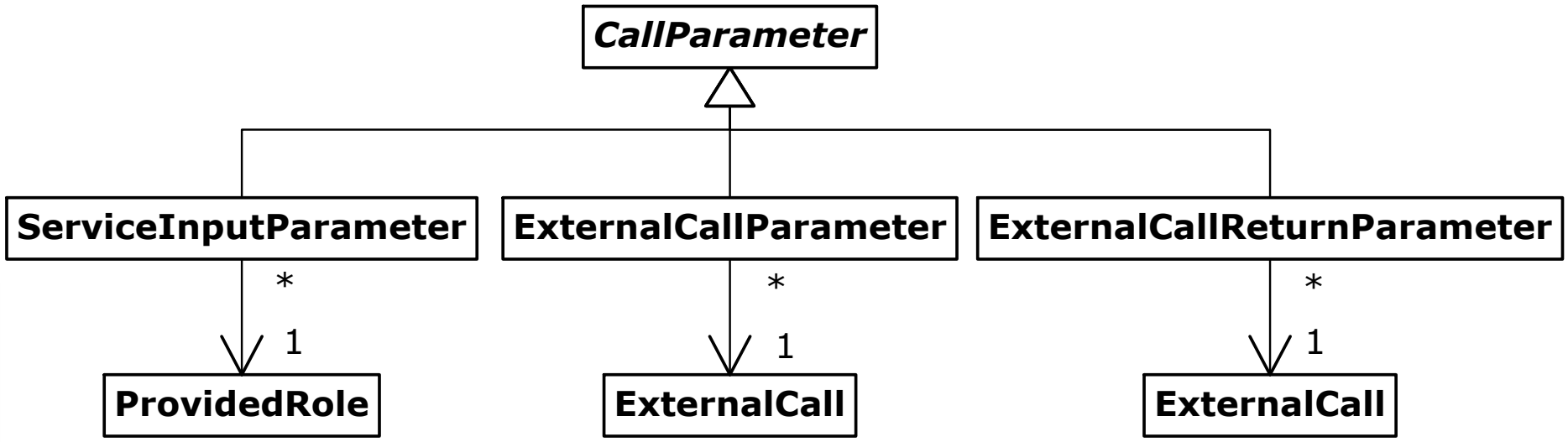
Parameter Dependencies - Approach



Model Variables - Metamodel

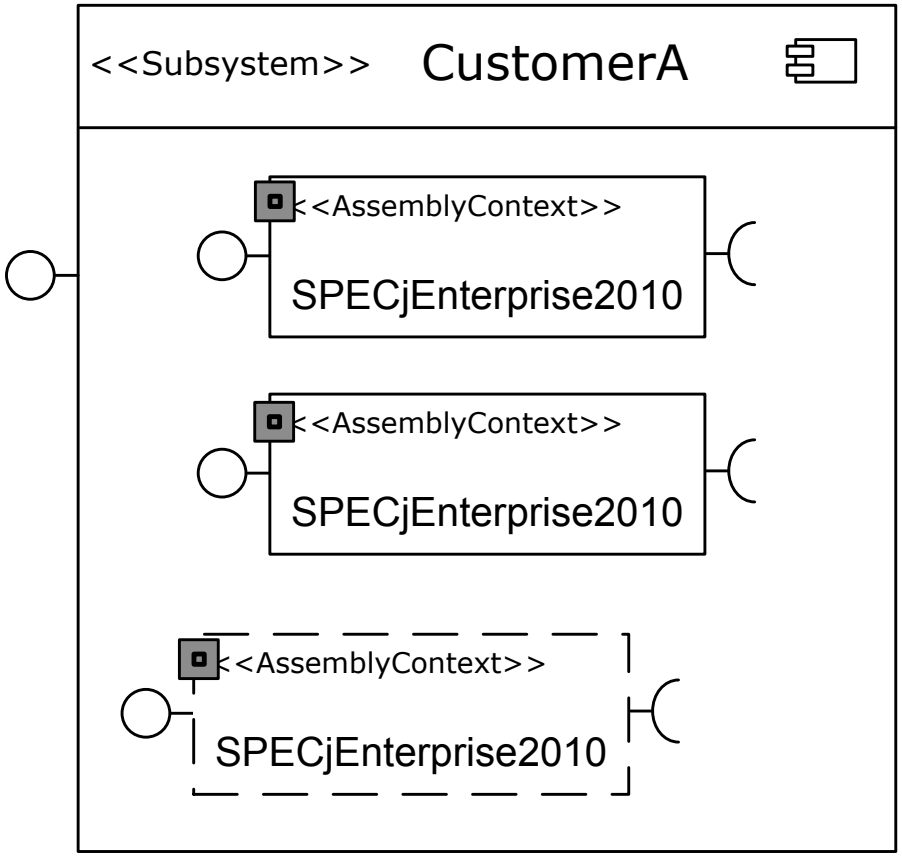


Call Parameter Hierarchy - Metamodel



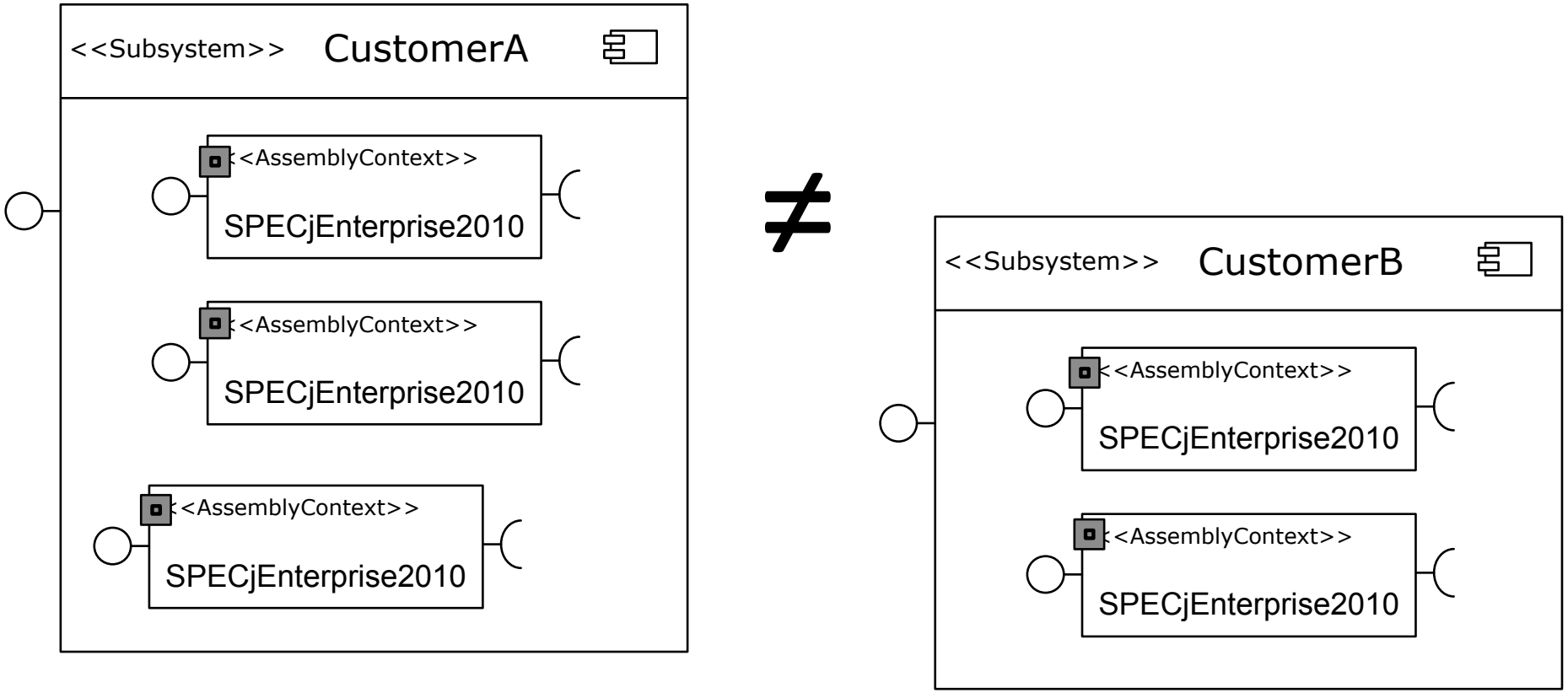
Validity of Measurements – „Scope“

- Customer-specific application server cluster
  - Scenario: Replicate Server Instance

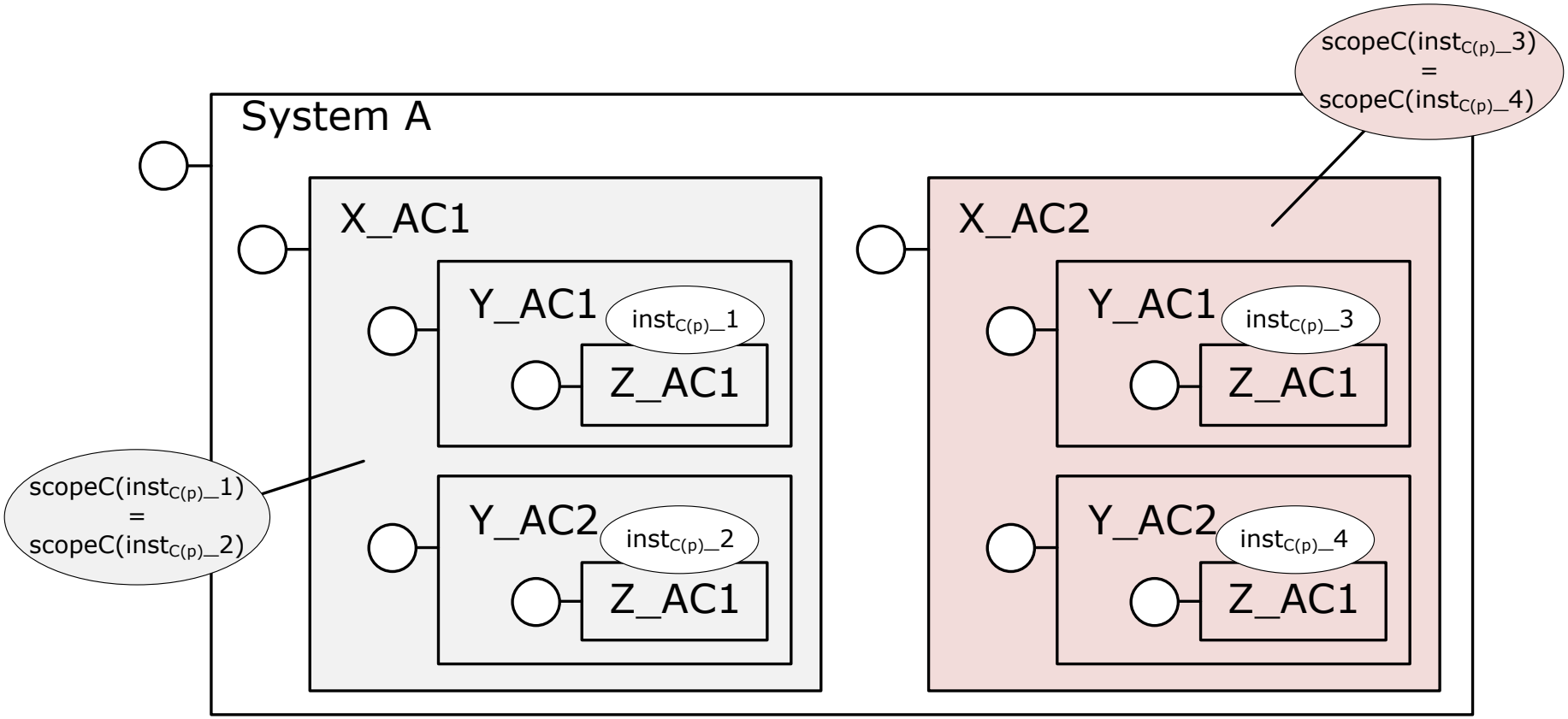


Validity of Measurements – „Scope“

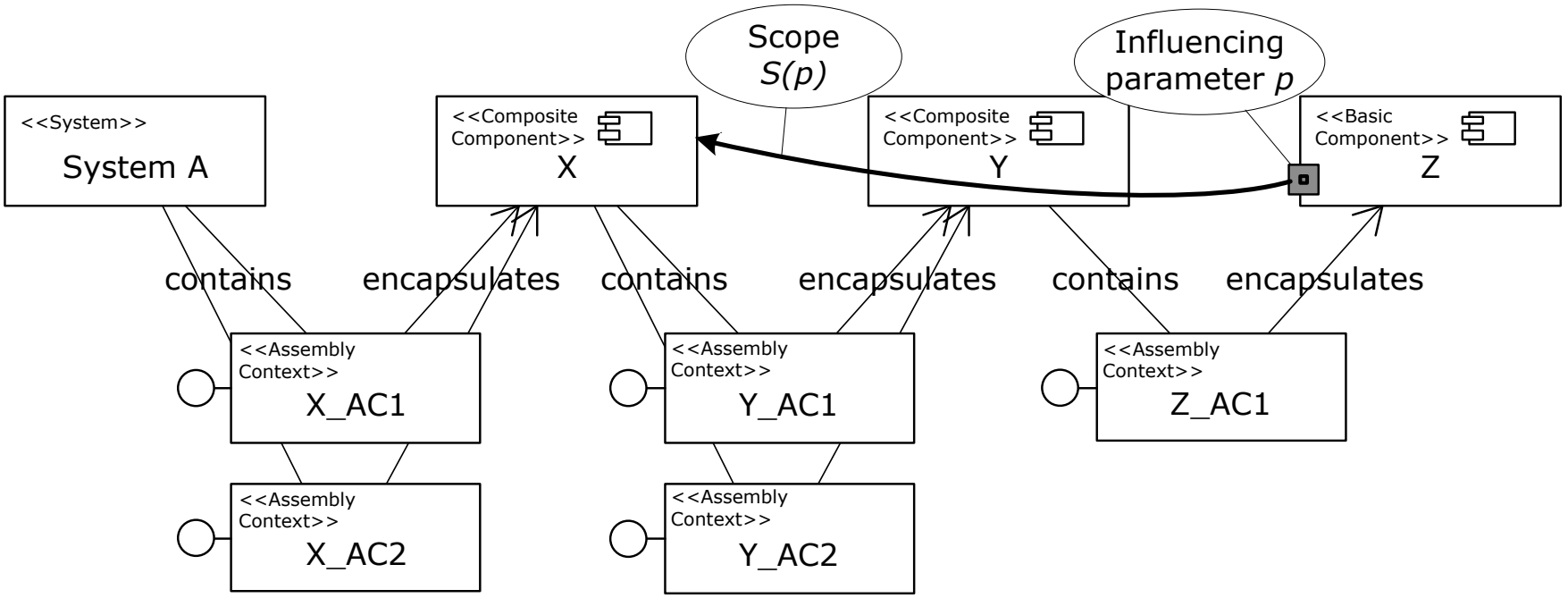
- Customer-specific application server cluster
  - Scenario: Replicate Server Instance



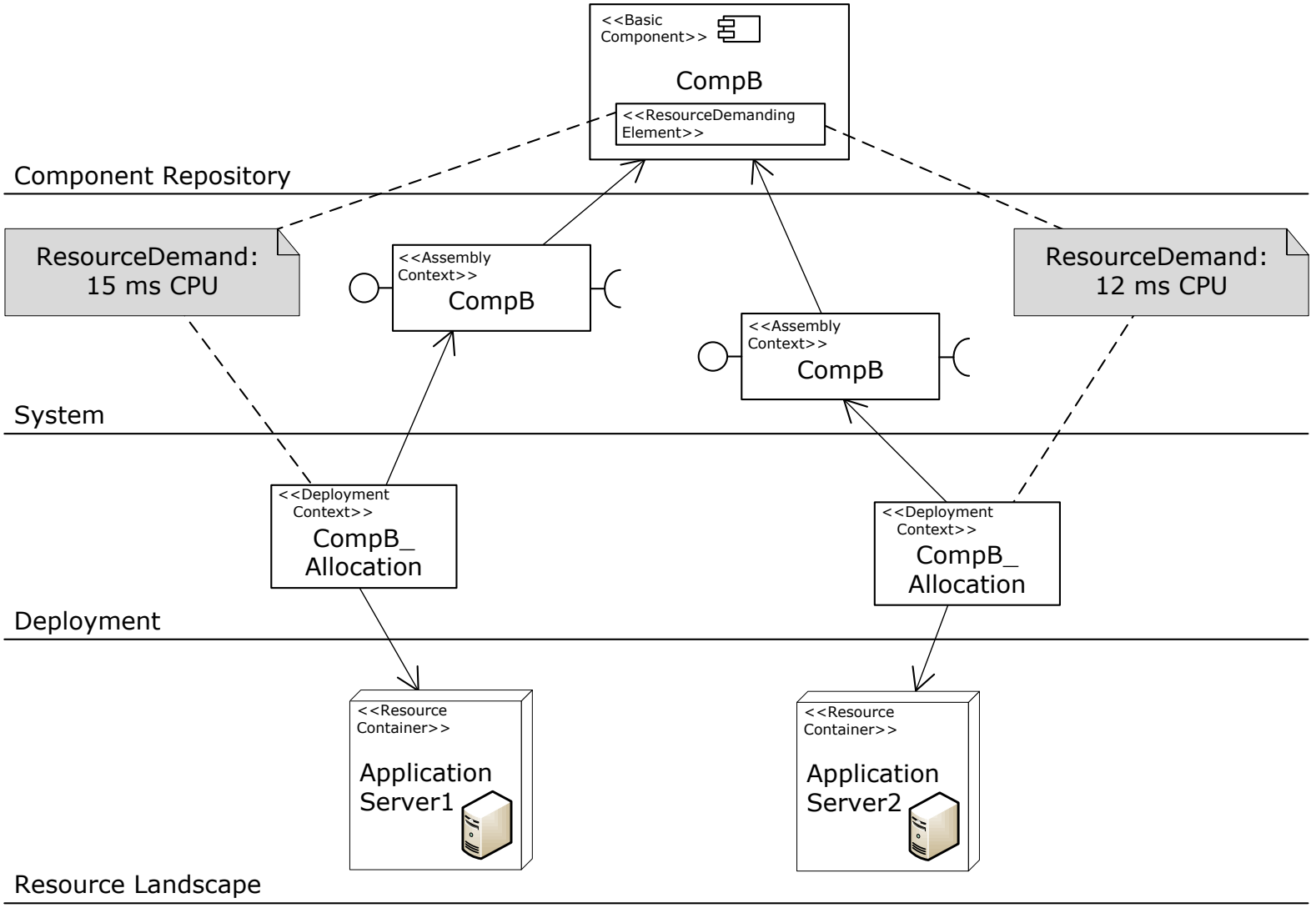
Modeling Scopes - Example



Modeling Scopes - Example



Deployment-Specific Resource Demands



Summary

- Use architecture-level performance models for online performance prediction
- Application Architecture Model
  - Different abstractions to model service behavior
  - New modeling concepts for expressing parameter and context dependencies
- Descartes Meta-Model (DMM)