

Engineering Scalable Cloud Systems

Steffen Becker, Uni Stuttgart
`steffen.becker@informatik.uni-stuttgart.de`

Available Resources for Driving Cars



[birgitH / pixelio.de]

(Regulated) Load



[http://www.general-anzeiger-bonn.de/incoming/861098-1.jpg-article299521.html/ALTERNATES/v4_3_w1240/861098-1.jpg]

Pay per use



[espana-elke / pixelio.de]

Limited Processing Rate



[SCHAU.MEDIA / pixelio.de]

Capacity Exceeded



[Rainer Sturm / pixelio.de]

Elastic Capacity Increase Enabled through Scalability



[<https://data.motor-talk.de/data/galleries/0/6/7618/35559068/2011-02-14-anti-stau-programm-6205363279726323242-5062351213067421571.jpg>]

Legacy System



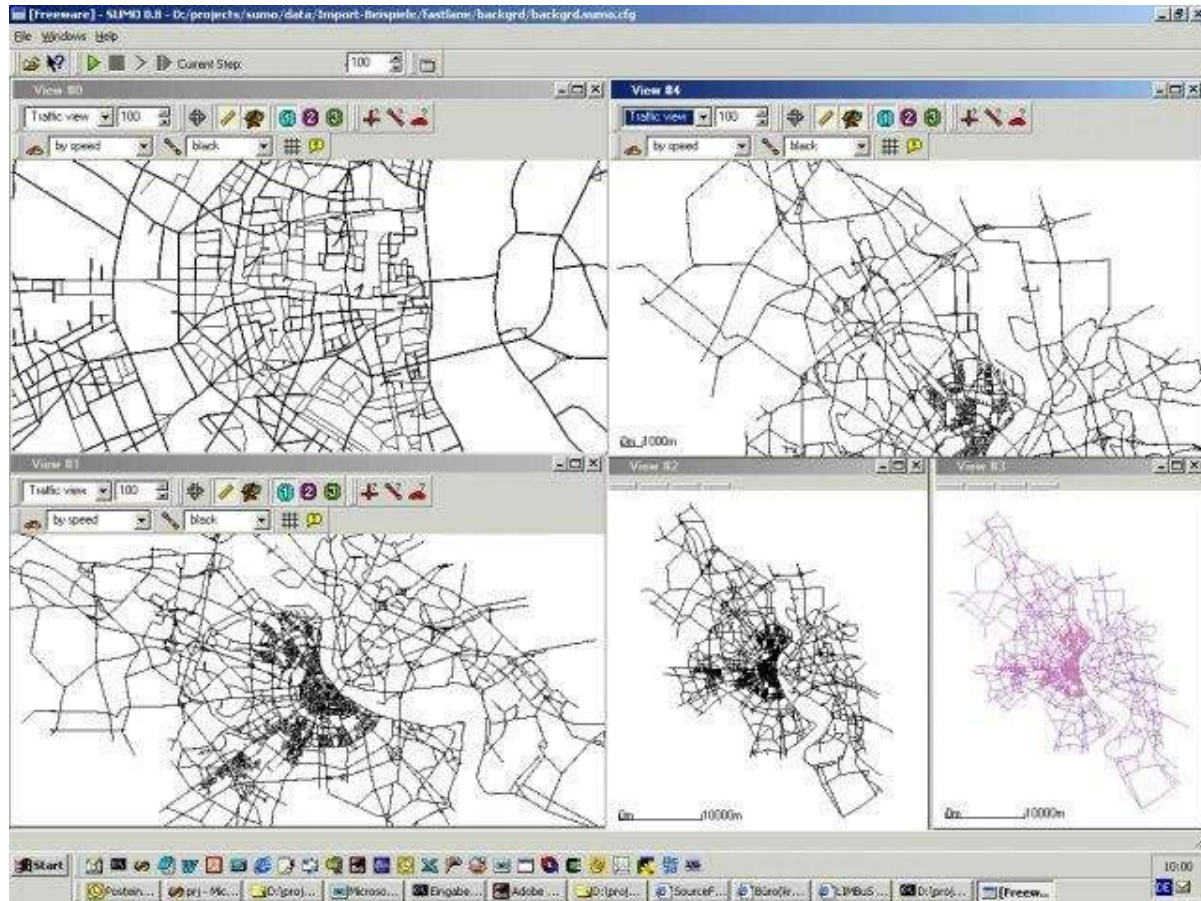
[Likretia / pixelio.de]

Modernization



[<http://www.tagesspiegel.de/politik/verkehrspolitik-streit-um-den-autobahnausbau/12770952.html>]

Scalability Analysis via Simulation



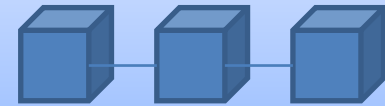
[<https://sourceforge.net/projects/sumo/>]

And IT?

15-Year old
JSP Web Shop

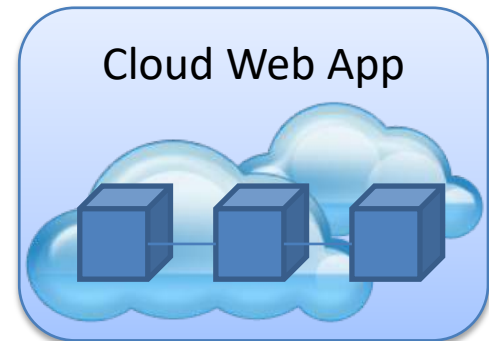
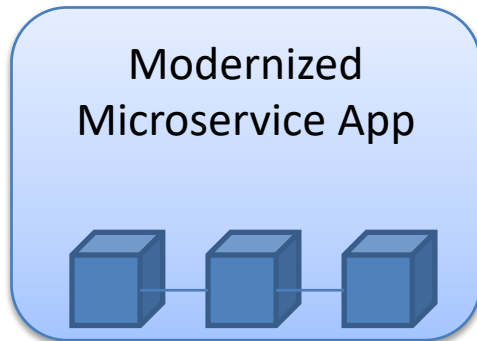


Modernized
Microservice App



“Finally, more Users
are coming!
Put it in the Cloud!”

And IT?



“But will it scale now?
Up to which load?
What do we have to
pay for the Cloud?”



Remaining Talk Outline

Introduction

- Cloud Computing
 - Running Example
 - New Quality Properties to consider
- 

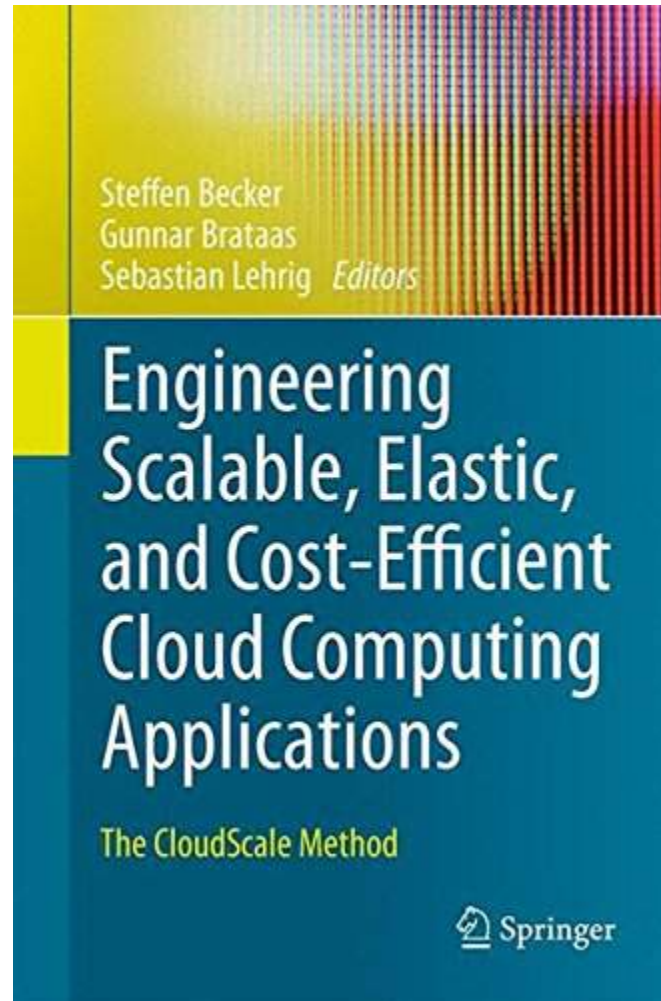
Cloud Scale

- Method Overview
 - Forward Engineering
 - Reengineering and Migration Support
- 

Concluding Remarks

- Summary
- Future Work

CloudScale book



Remaining Talk Outline

Introduction

- Cloud Computing
- Running Example
- New Quality Properties to consider

Cloud Scale

- Method Overview
- Forward Engineering
- Reengineering and Migration Support

Concluding Remarks

- Summary
- Future Work

Cloud Computing: NIST Definition Characteristics

1.

- On demand self-service

2.

- Broad network access

3.

- Resource pooling

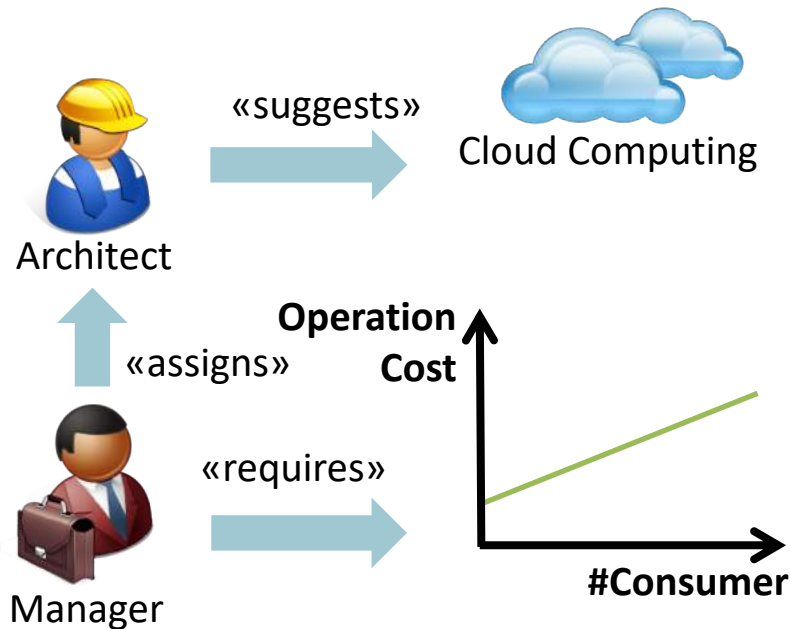
4.

- Rapid elasticity

5.

- Measured service

Refined Motivation Example



Management concerns

- Avoid disappointments
 - Response times
 - Operation costs
- But how can we measure and analyse this?

Definitions

Scalability

“ability of a cloud layer to increase the maximum *workload* it can handle as bound by its *SLOs* by expanding its quantity of consumed *services*”

Elasticity

“the degree a cloud layer autonomously adapts the maximum *workload* it can handle as bound by its *SLOs* to workload over *time*”

Efficiency

“a measure that relates the demanded maximum *workload* a cloud layer can handle as bound by its *SLOs* to consumed *services* over time”

Definitions

Scalability

“ability of a cloud layer to increase the **maximum workload it can handle as bound by its SLOs** by expanding its quantity of consumed *services*”

Elasticity

“the degree a cloud layer autonomously adapts the **maximum workload it can handle as bound by its SLOs** to workload over *time*”

Efficiency

“a measure that relates the demanded **maximum workload a cloud layer can handle as bound by its SLOs** to consumed *services* over time”

Metrics

Capacity

“maximum workload a cloud layer can handle as bound by its Service Level Objectives (SLOs)”

- **E.g. arrival rate capacity:** The cloud layer scales up to 112 consumers/minute one a defined static work situation

Scalability

“ability of a cloud layer to increase its capacity by expanding its quantity of consumed services”

- **Scalability range:** The cloud layer can deal with up to an additional 100 consumers/minute when scaling-out its database

Metrics

Elasticity

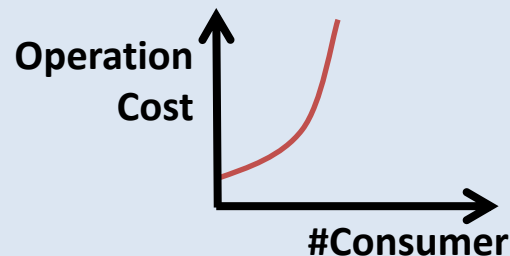
“the degree a cloud layer autonomously adapts capacity to workload over time”

- **Time to Quality Repair:** 30 seconds for an additional 10 requests/hour
- **Number of SLO Violations:** 42 SLO (response time) violations in 1 hour

Efficiency

“a measure that relates demanded capacity to consumed services over time”

- **Consumer Number Cost:**



Talk Outline

Introduction

- Cloud Computing
- Running Example
- New Quality Properties to consider

Cloud Scale

- Method Overview
- Forward Engineering
- Reengineering and Migration Support

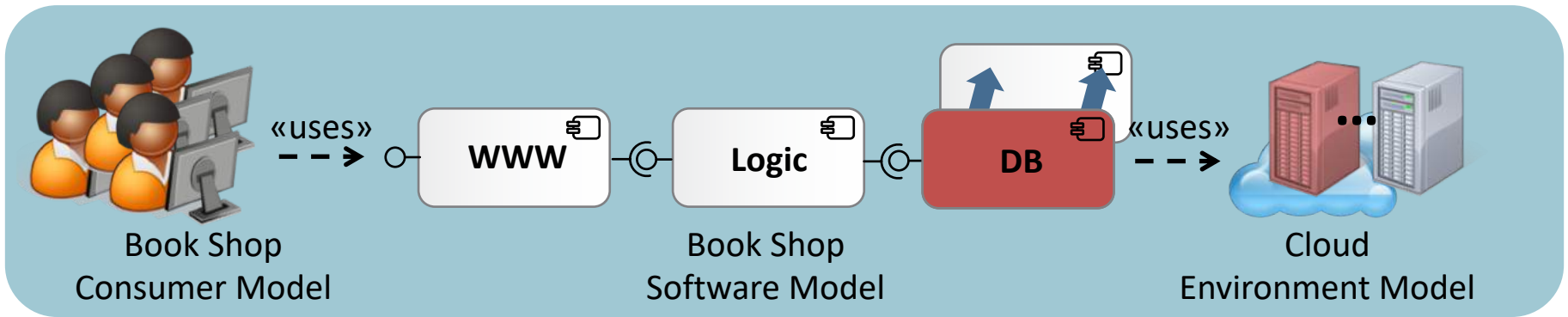
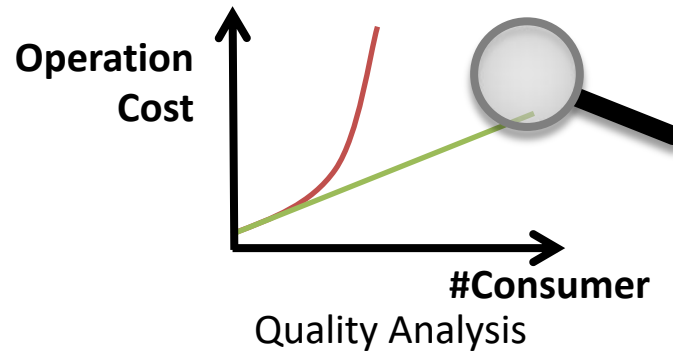
Concluding Remarks

- Summary
- Future Work

“But will it scale now?
Up to which load?
What do we have to
pay for the Cloud?”

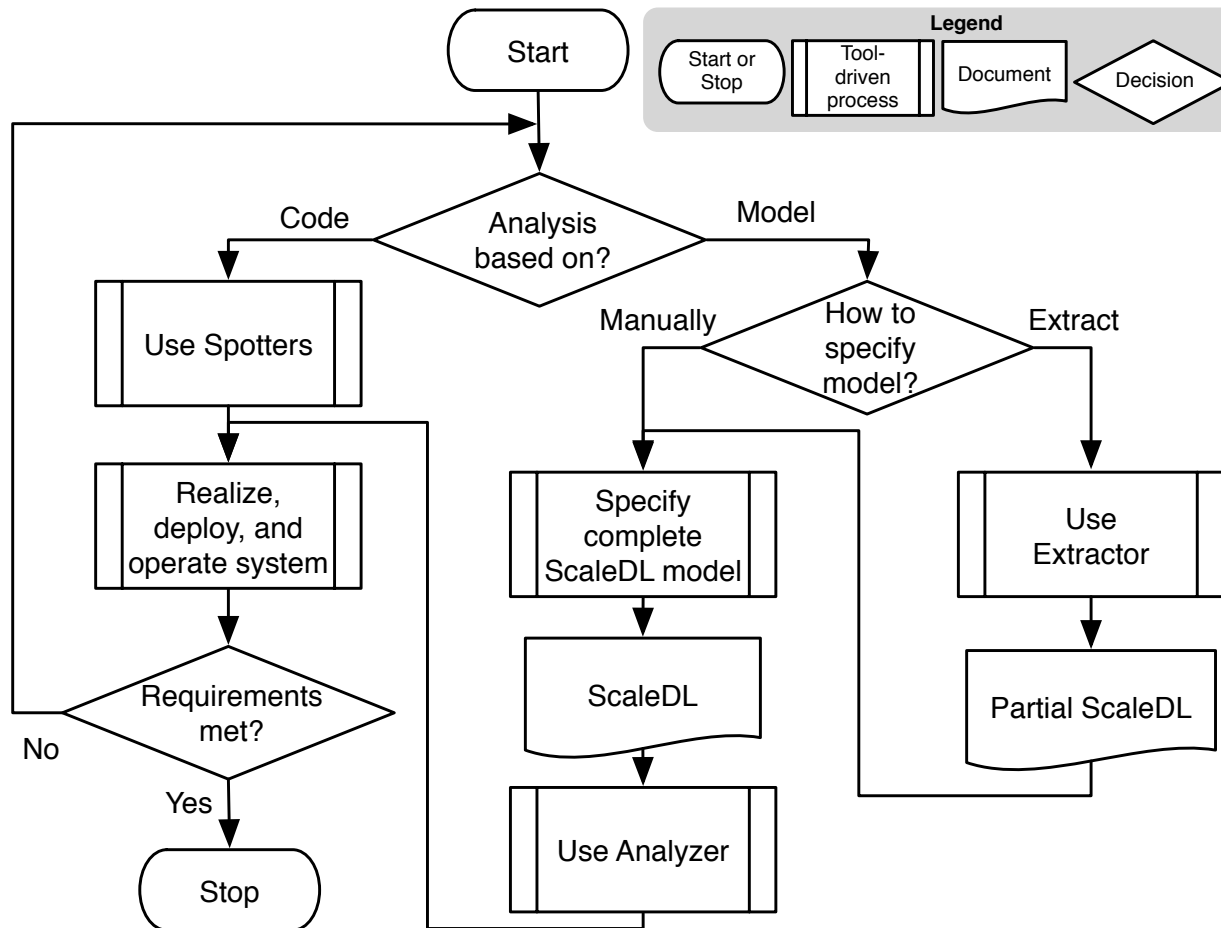


CLOUD SCALE METHOD



ScaleDL Model

CloudScale Method



CloudScale IDE

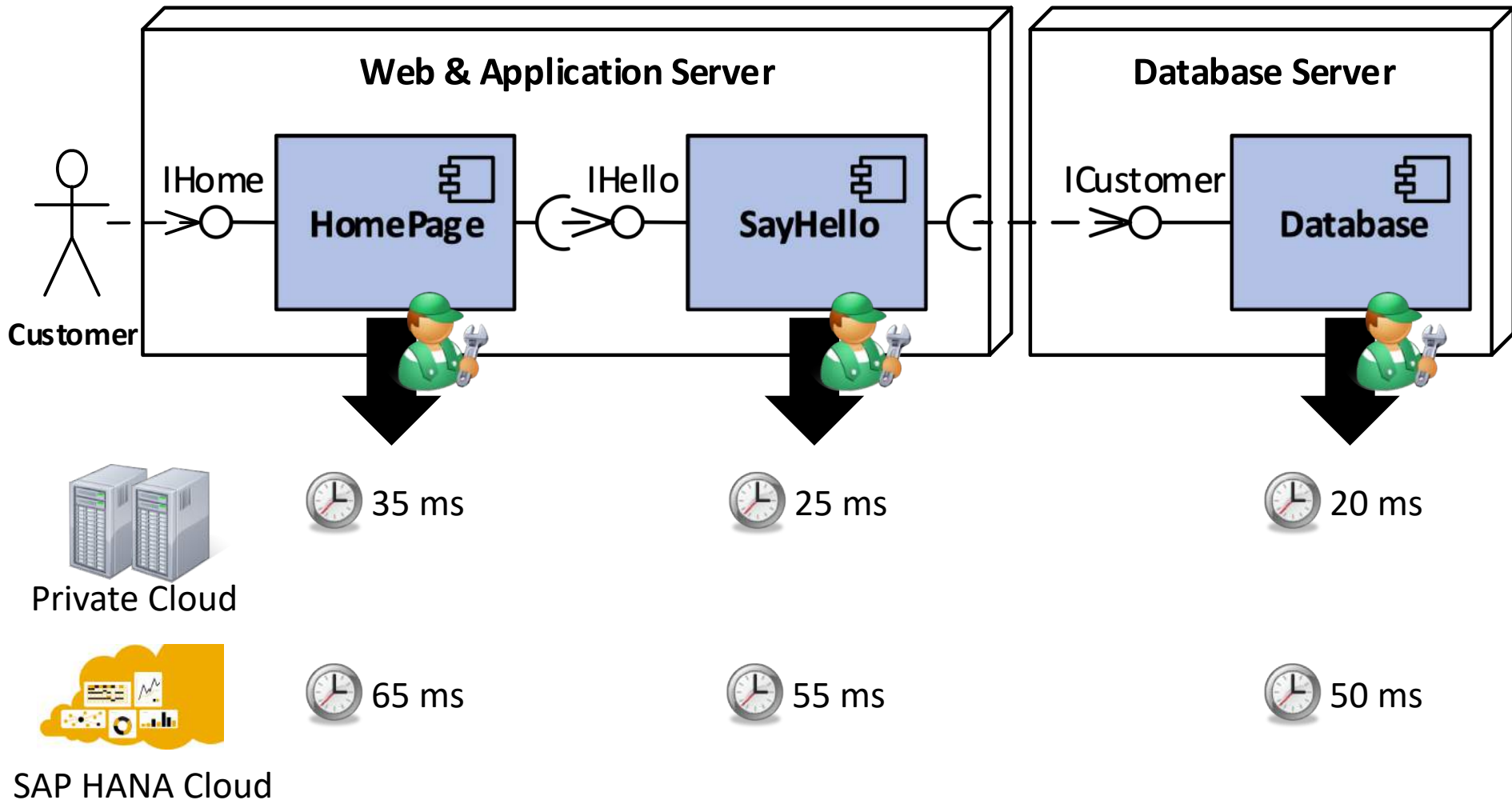
The screenshot displays the CloudScale IDE interface, which is divided into several panes. The top-left pane shows a project explorer with a tree structure under 'Minimal-Example', including 'Analysier', 'Fibonacci (Overview) sadf [Alternative]', 'Input', 'Configurations', 'Results', 'alternative [Alternative]', 'Extractor', 'Fibonacci [Alternative]', 'Static Spotter', 'Dynamic Spotter', 'Server connection', and 'ScaleDL models'. The bottom-left pane shows a workflow diagram with various components like 'Input', 'Transform', 'Simulation', and 'Output' connected by arrows. The right pane is titled 'Analysier input [Fibonacci (Overview) sadf]' and 'Analysier config [Basic normal]'. It contains a 'Basic normal [NORMAL]' configuration section with tabs for 'Input', 'Results', and 'Details'. The 'Details' tab shows 'Input: Fibonacci (Overview) sadf', '#Results: 1', 'Last change: 18/11, 11:53:37', and 'Last result: 26/10, 11:27:15'. Below this is a 'Basic settings' section with tabs for 'Measuring points', 'Monitors', 'SLO', and 'Advanced editor'. The 'Advanced editor' tab shows a list of model repositories and their configurations. At the bottom right, there is a 'Property' table and a 'Run' button.

Property	Value
Description	Alternative experiment
Id	171c34c9-e9f4-43e2-a75b-d82caa3da9cc
Name	CloudScale experiments model
Repetitions	1
Tool Configuration	Simu Lizar Configuration SimuLizar default configuration

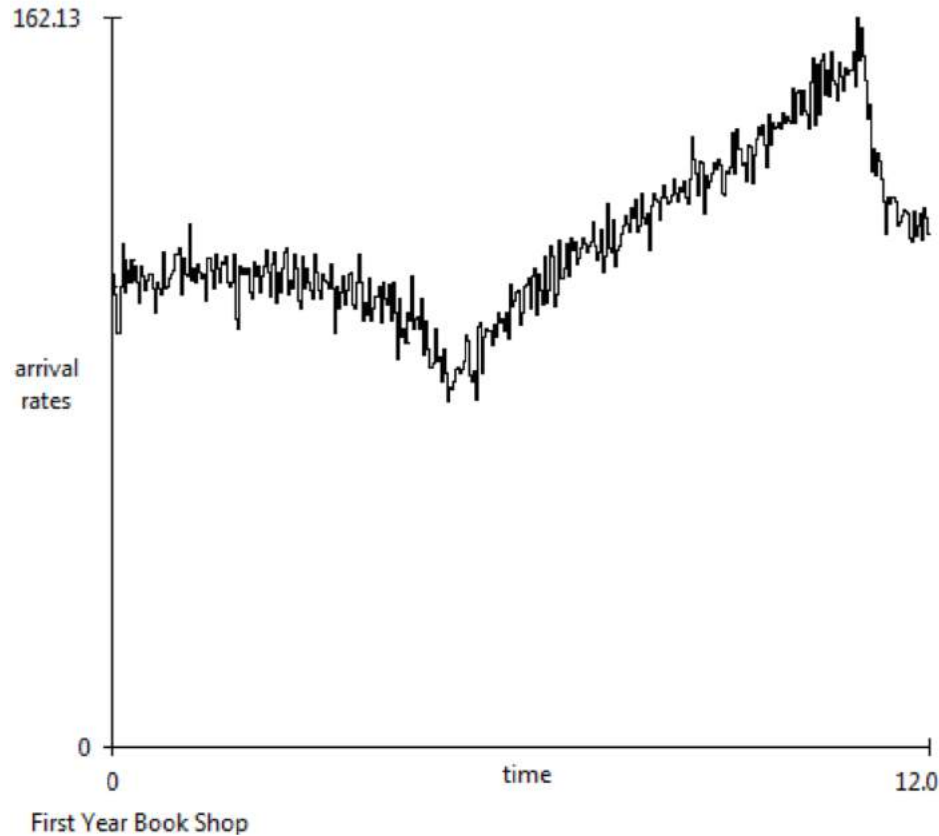
Alternative is valid. Run

FORWARD ENGINEERING

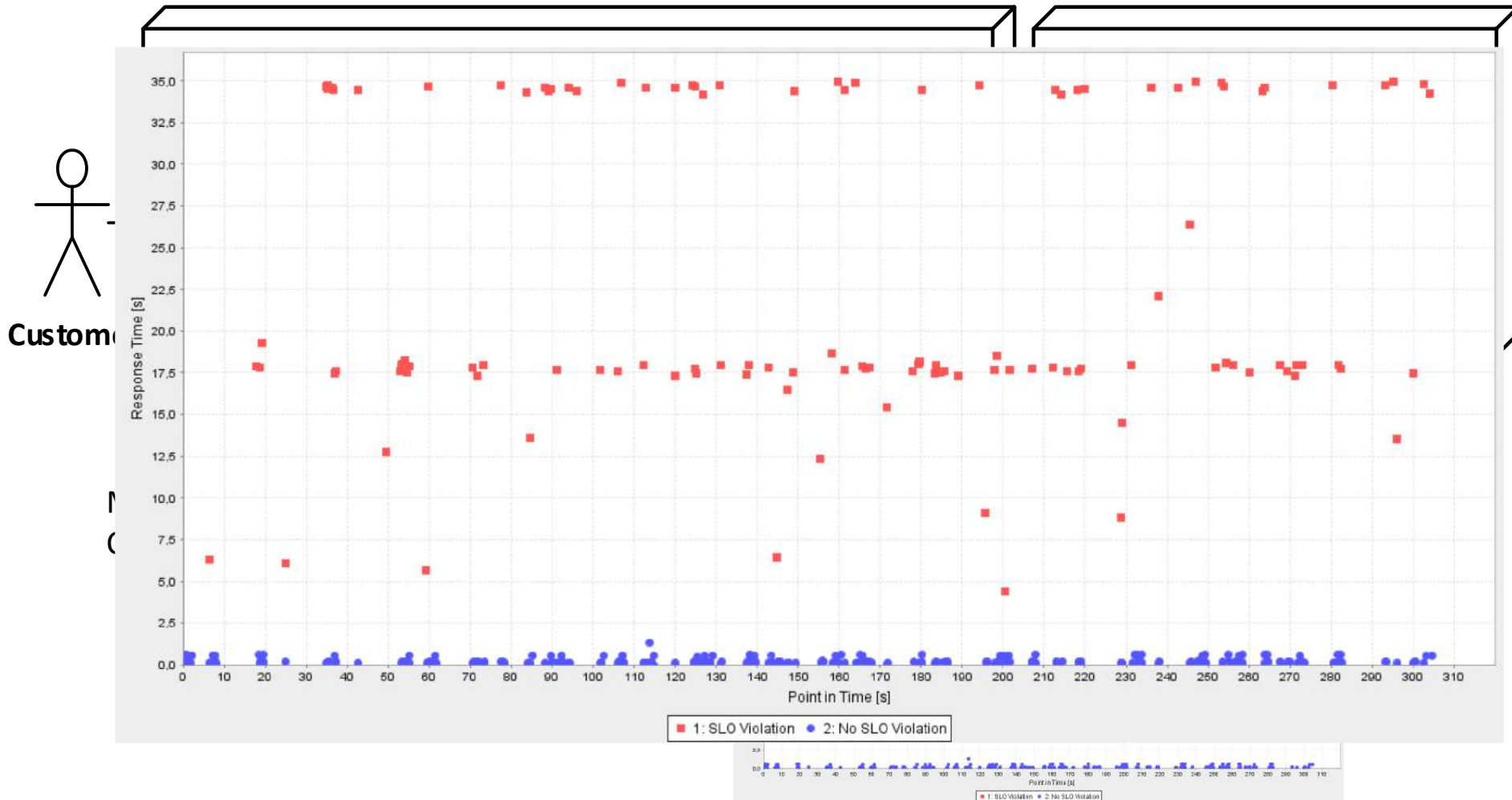
Analyse: Calibrate Model



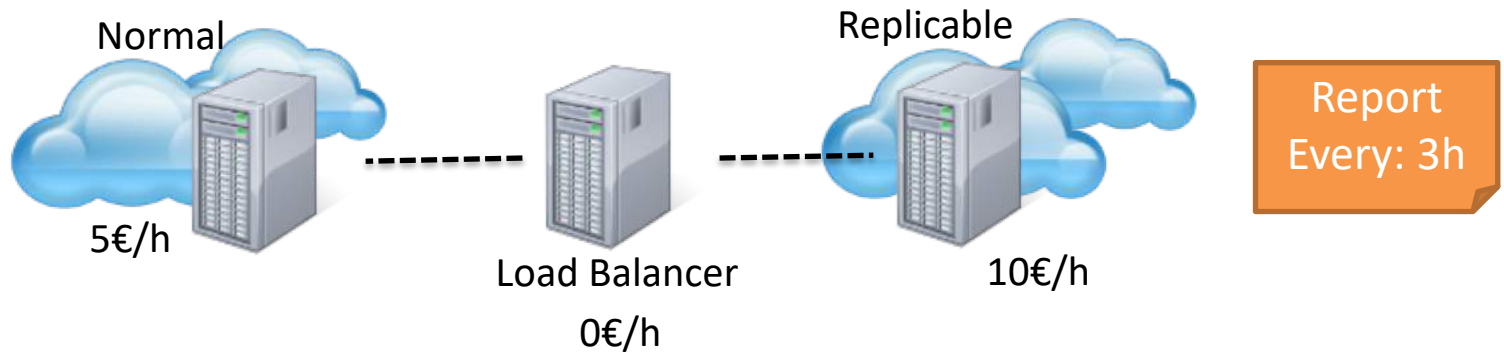
Analyse: Model Workload Evolution



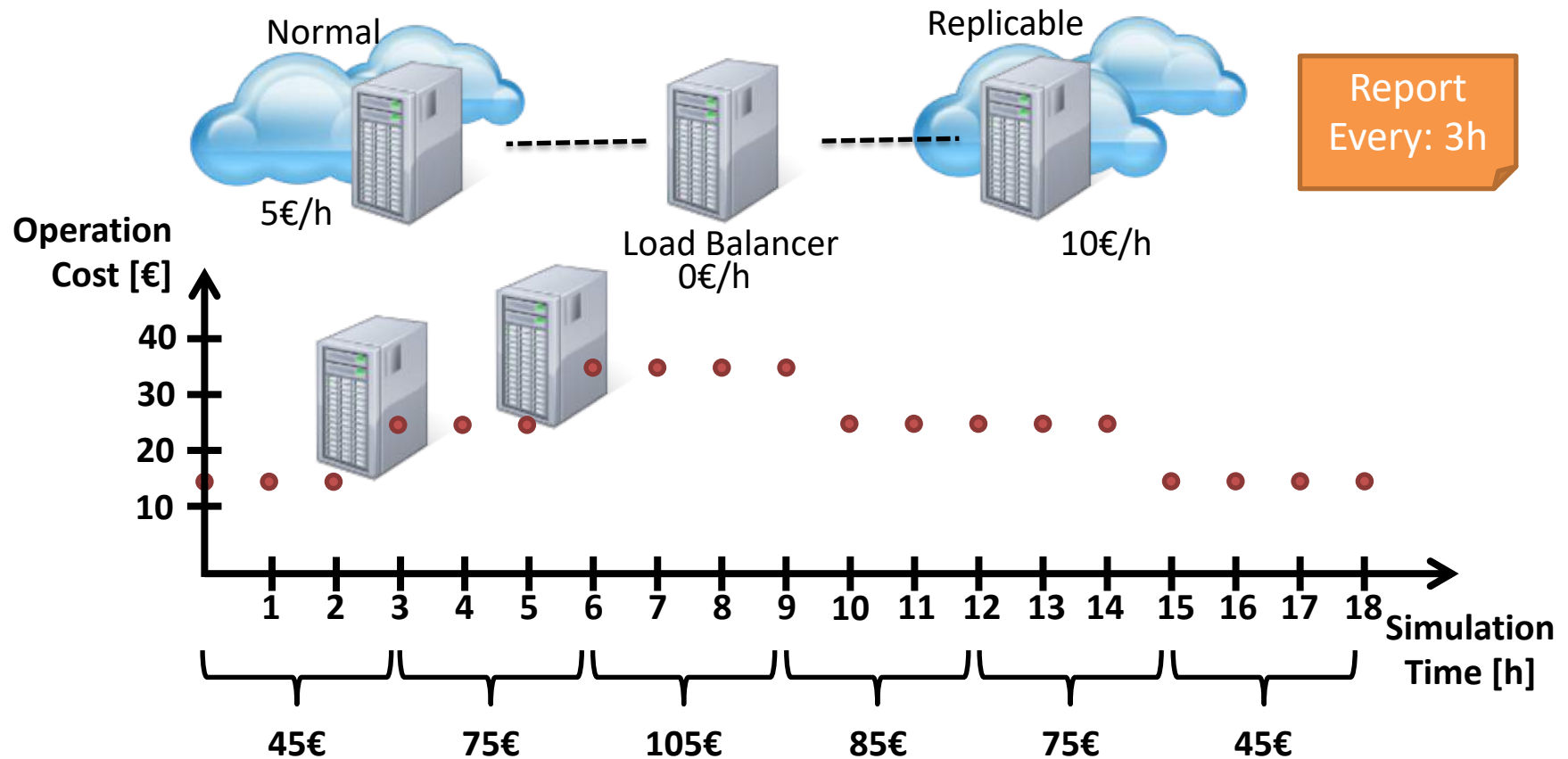
Analyse: Running the Analyser



Measuring Cost



Measuring Cost



15-Year old
JSP Web Shop



REENGINEERING AND MIGRATION SUPPORT

Sketching the General Idea

15-Year old
JSP Web Shop

```
do  
for  
i=1..N  
print
```



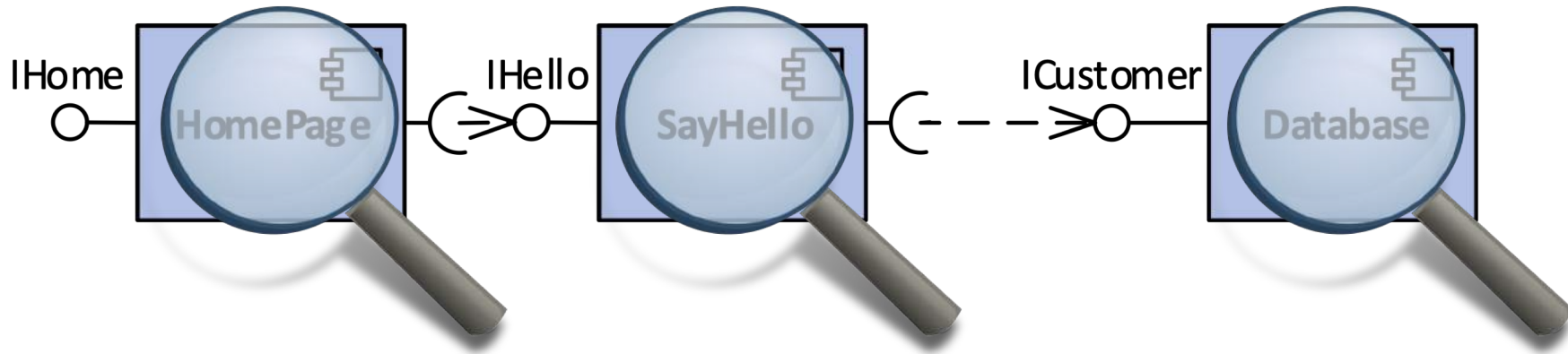
Any Symptoms?

- Traffic Jams?
- Hick-Ups?
- Ramp Effects?

Then I know what's your
problem, my friend! You've
got DB Congestion issues!



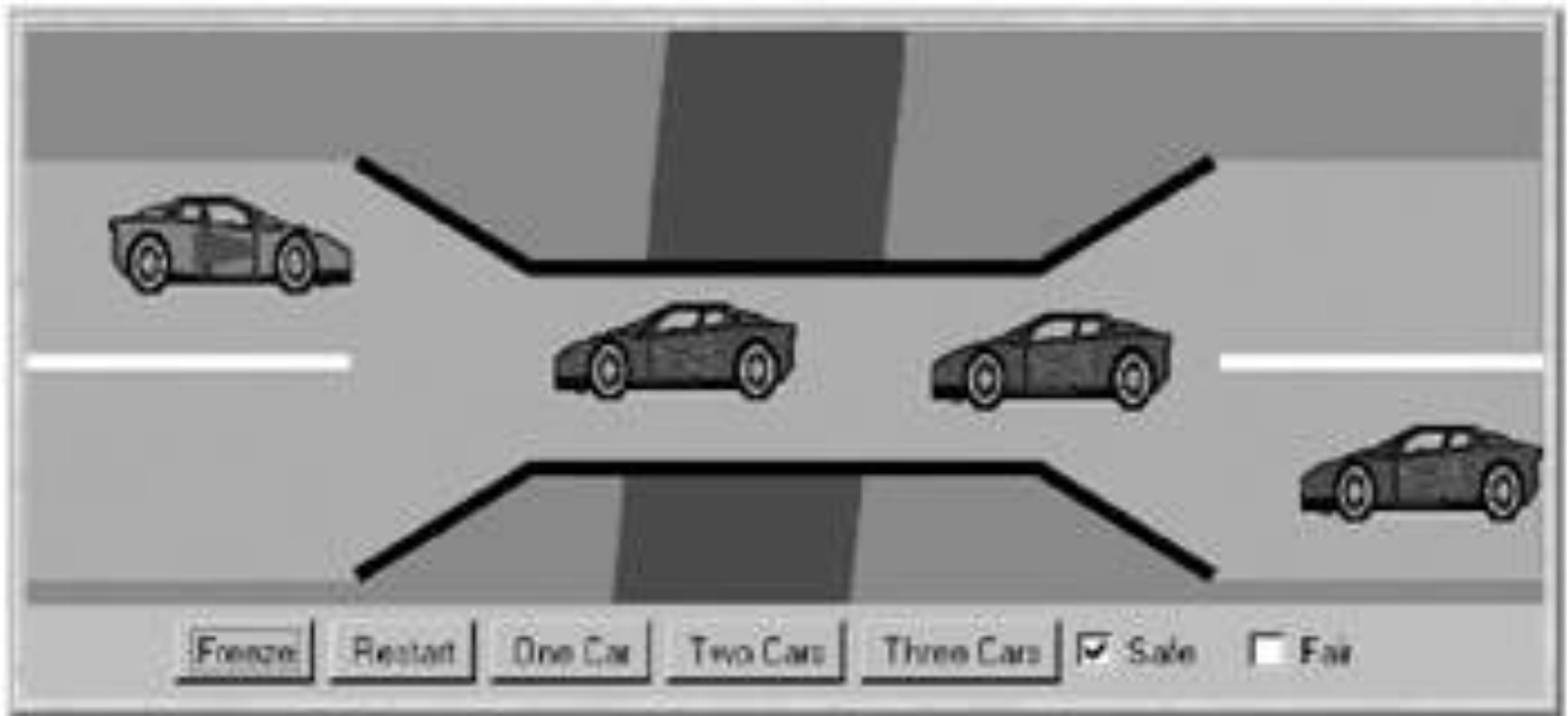
SpotHowNotTos: Static Spotter



First, let's see how the code does...

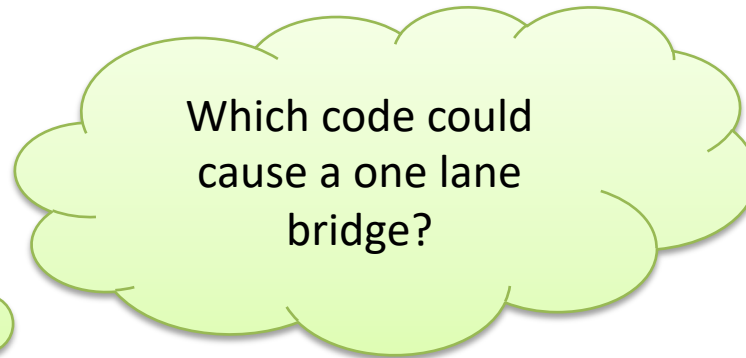


HowNotTo Example: One Lane Bridge



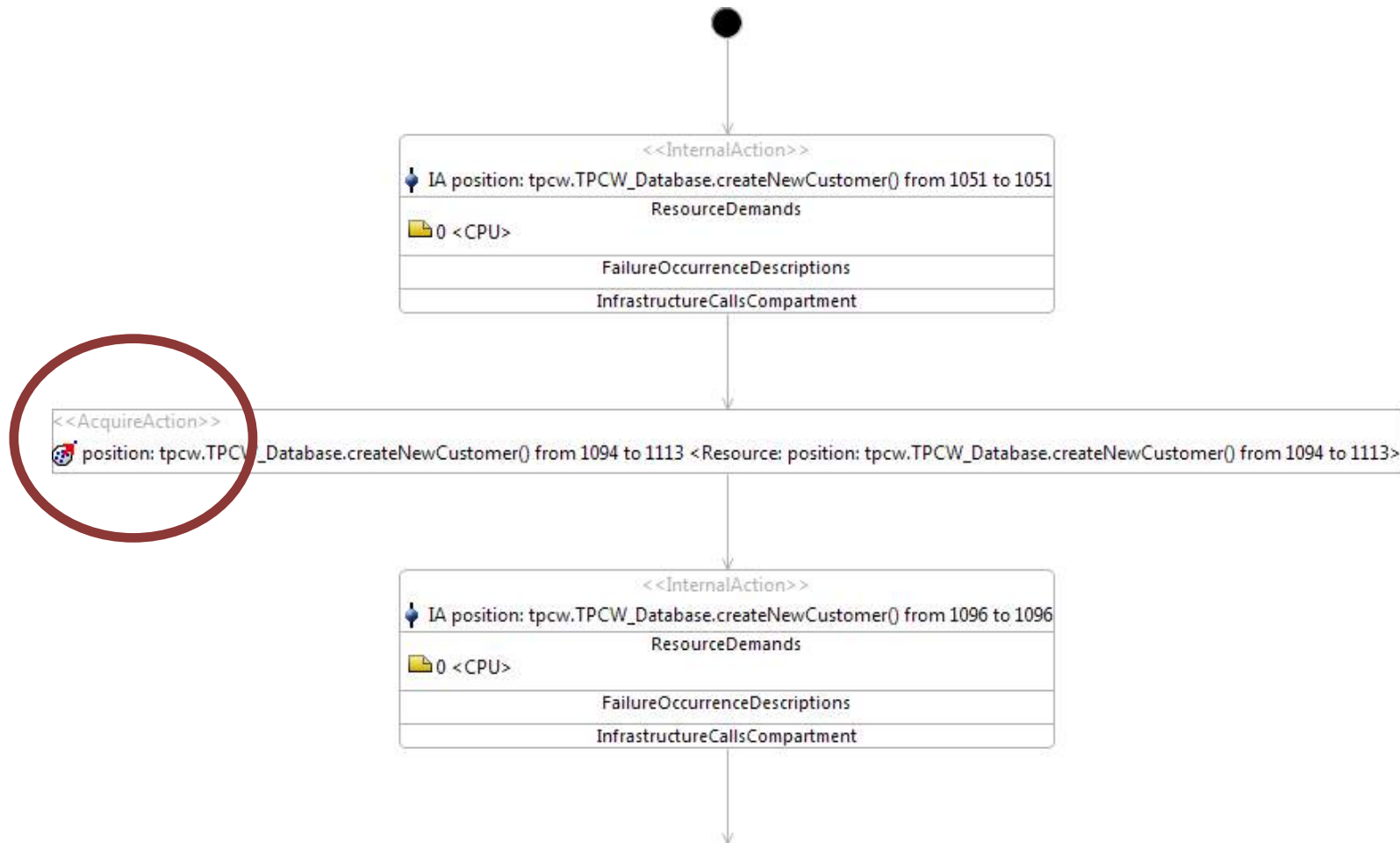
[J. Magee, J. Kramer (2006): *Concurrency: State Models and Java Programs*]

Code to identify...

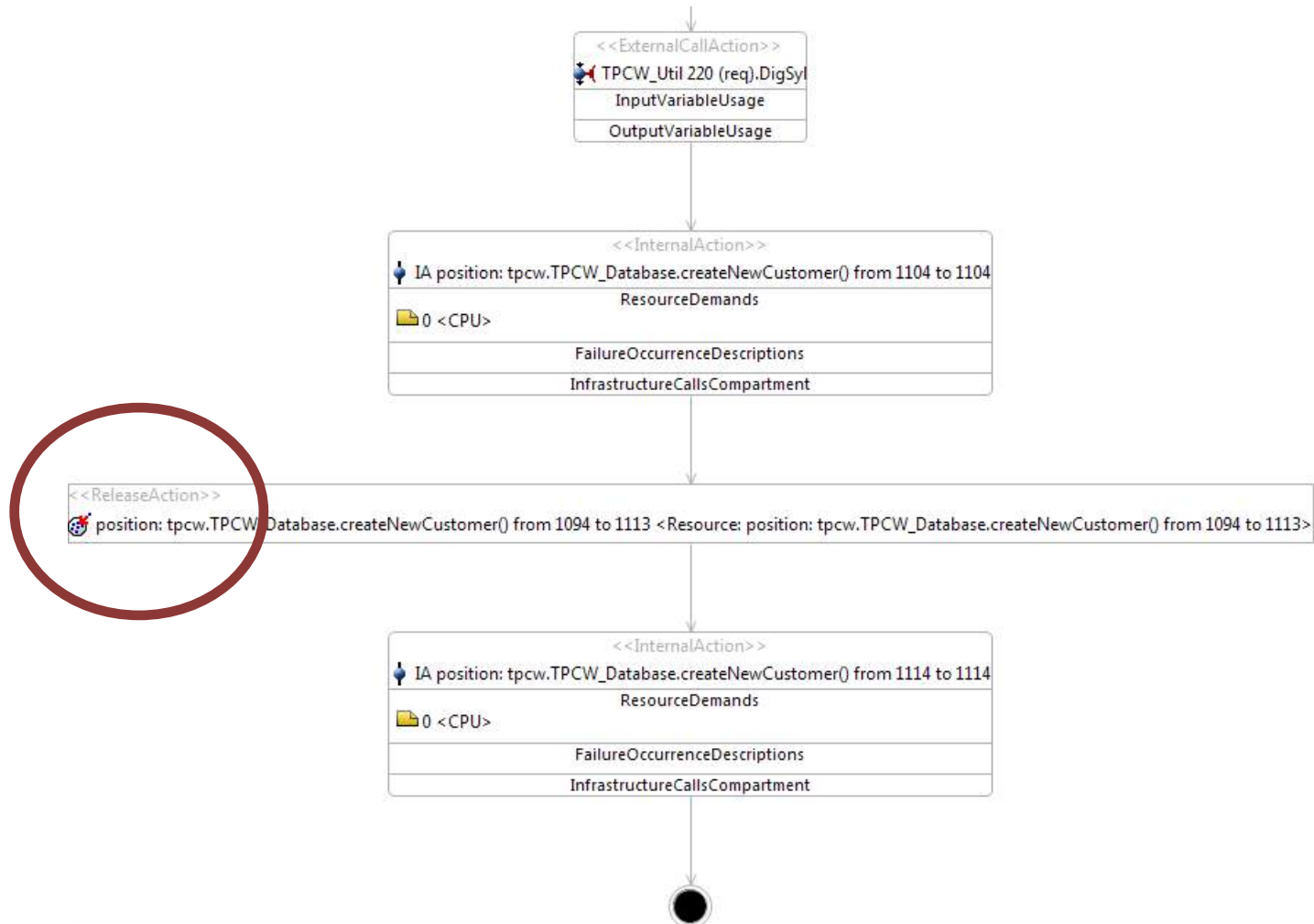


```
public void methodA() {  
    [...]  
    synchronized(this.class) {  
        doProcess(data);  
    }  
    [...]  
}
```












One AcquireReleasePair found: tpcw.TPCW_Database.createNewCustomer()



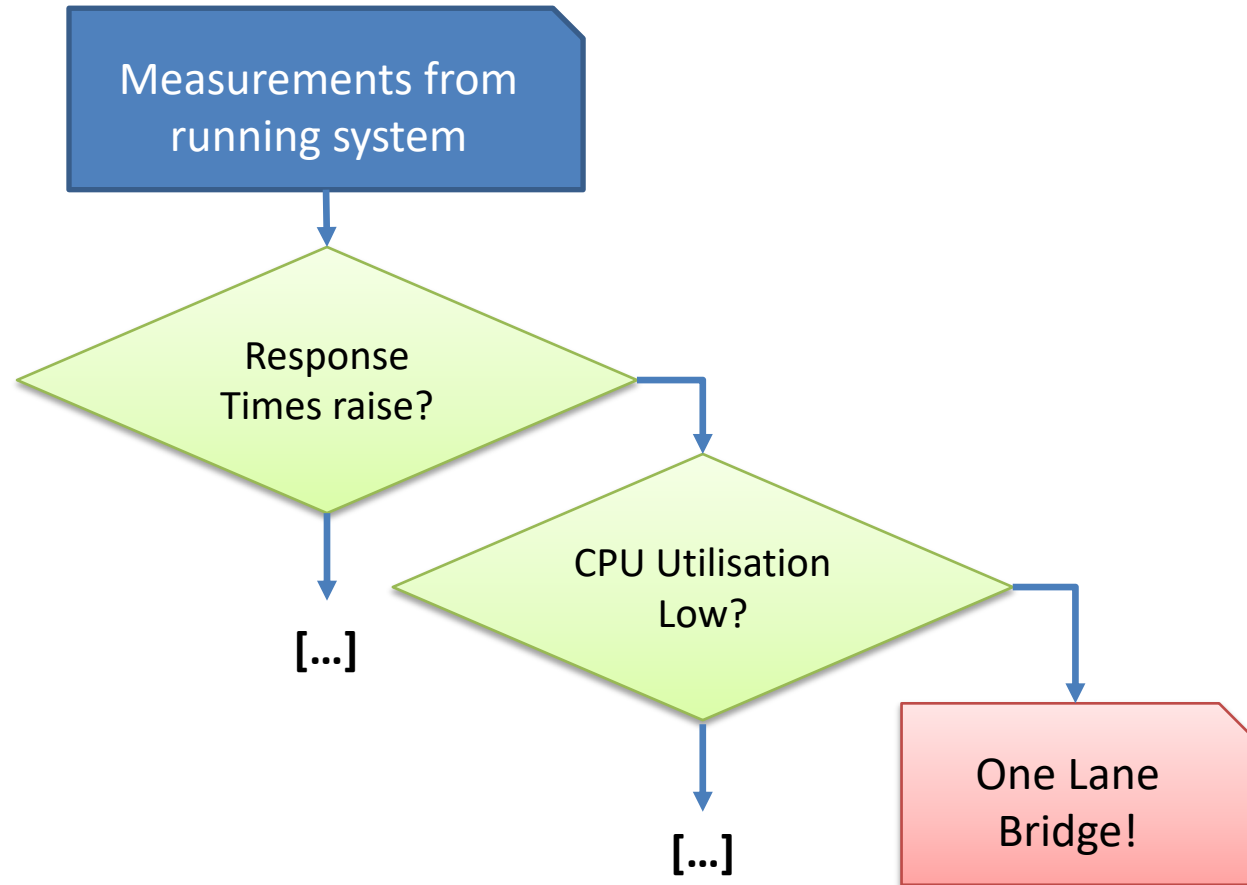
One AcquireReleasePair found: tpcw.TPCW_Database.createNewCustomer()



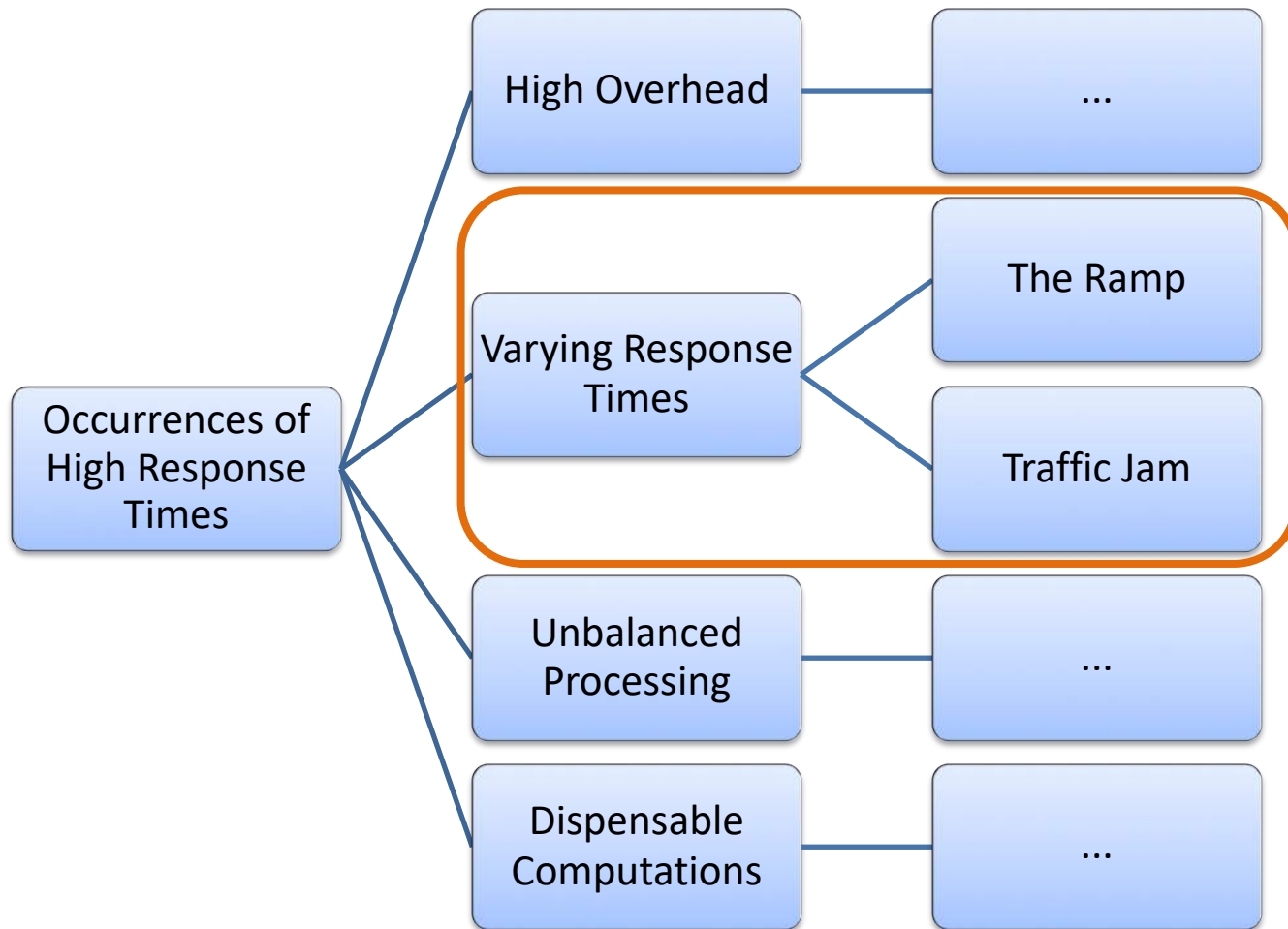
Spot: Running the Static Spotter

Annotation	Rating	Annotated Elements
▲  AcquireReleasePair (2 annotations)		
▶  AcquireReleasePair	100,00%	acquire=de.uka.ipd.sdq.pcm.seff.imp
▶  AcquireReleasePair	100,00%	acquire=de.uka.ipd.sdq.pcm.seff.imp
▶  FindMethods (208 annotations)		
▶  FindPrimitiveComponents (72 annotations)		
▲  SynchronizedMethod (5 annotations)		
▶  SynchronizedMethod	100,00%	synchronizedMethods=org.eclipse.gi
▶  SynchronizedMethod	100,00%	synchronizedMethods=org.eclipse.gi
▶  SynchronizedMethod	100,00%	synchronizedMethods=org.eclipse.gi
▶  SynchronizedMethod	100,00%	synchronizedMethods=org.eclipse.gi
▶  SynchronizedMethod	100,00%	synchronizedMethods=org.eclipse.gi

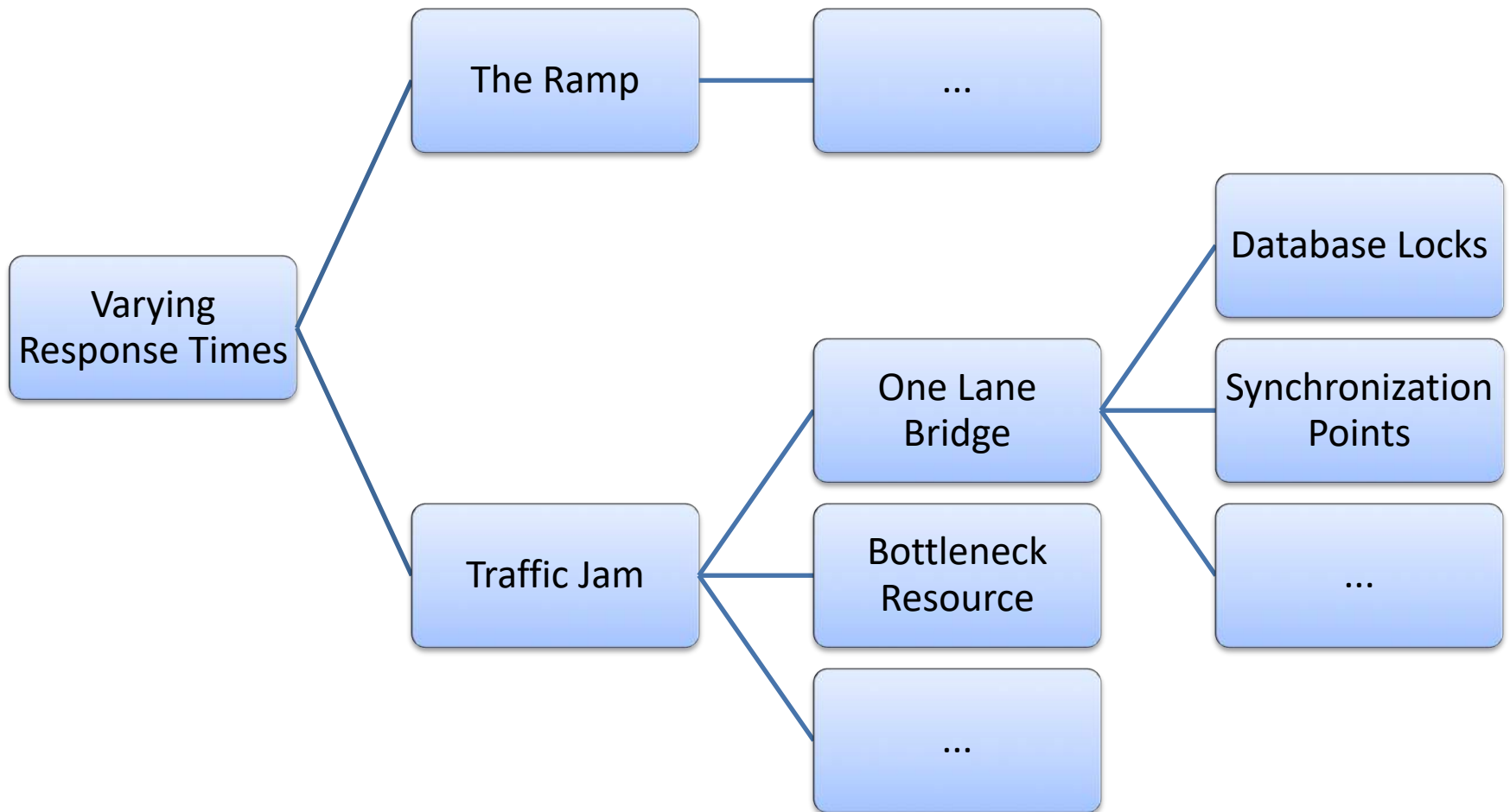
Dynamic Spotter: Problem Hierachy



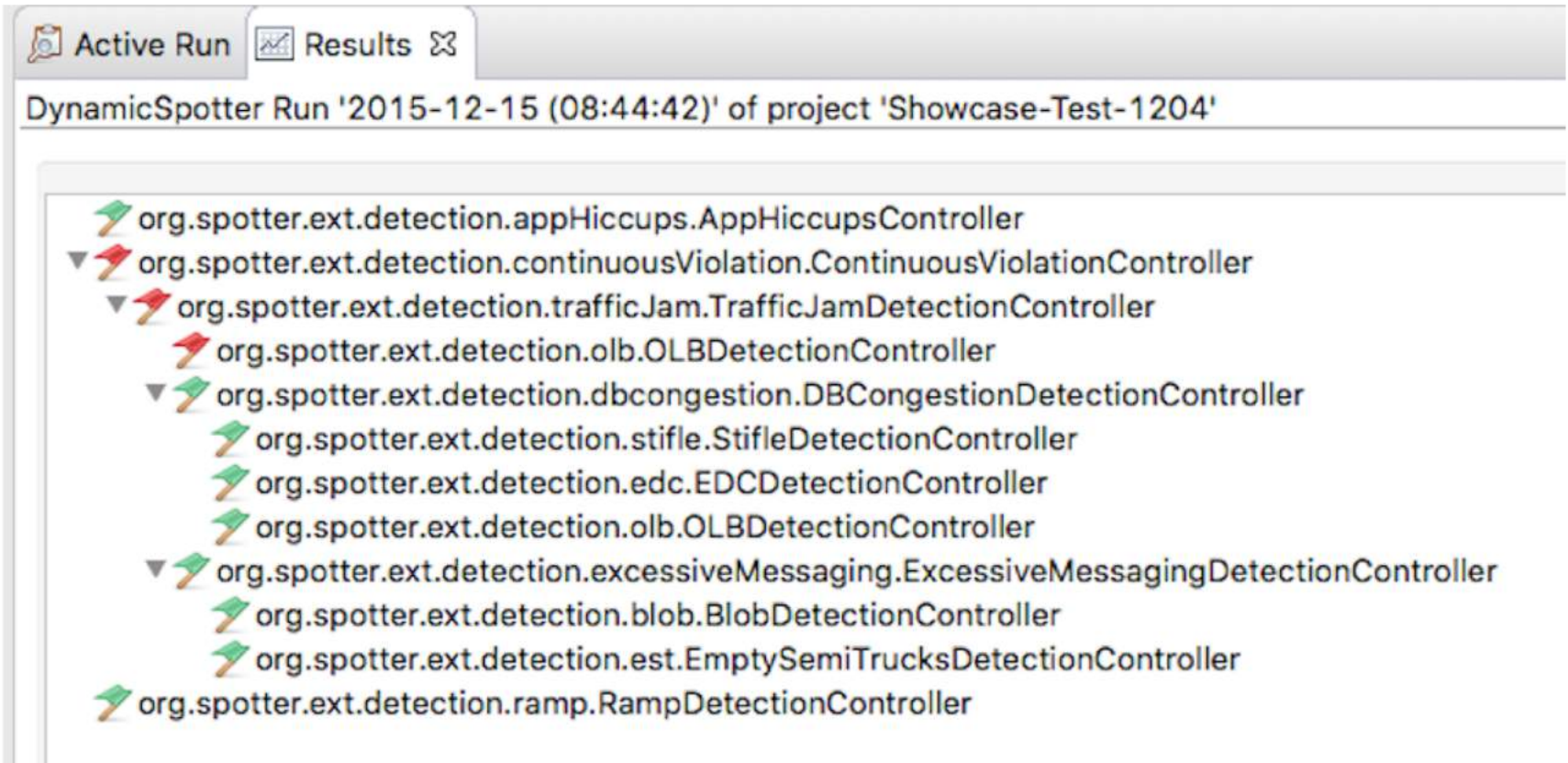
Problem Hierarchy: Excerpt



Problem Hierarchy: Excerpt cont.



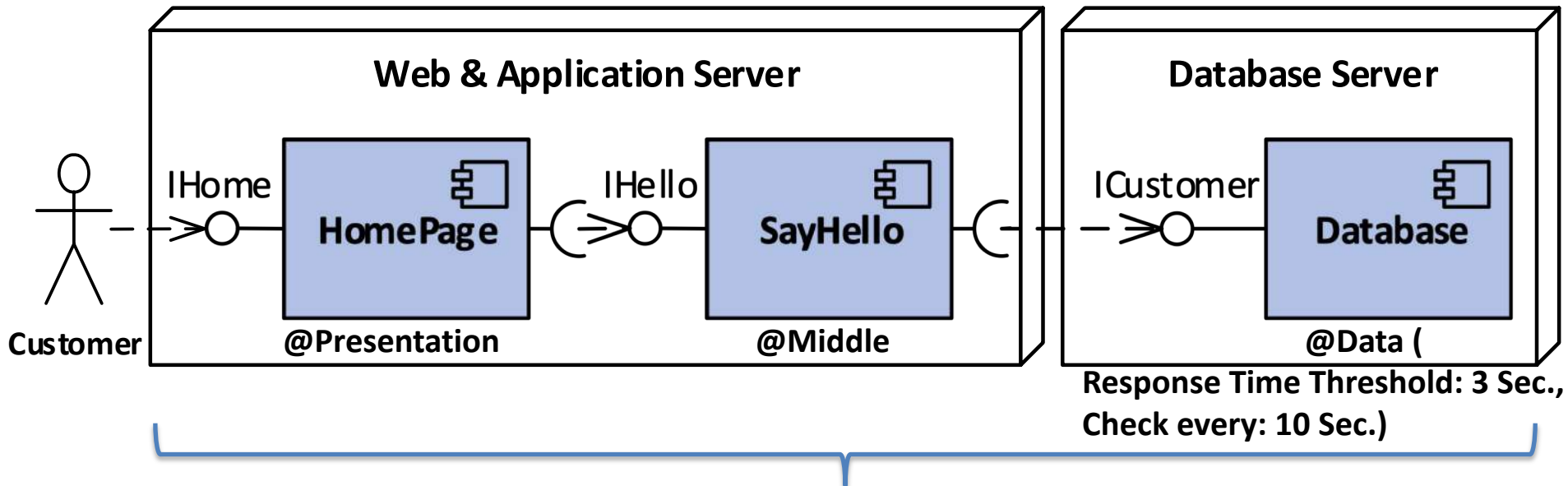
Running Dynamic Spotter



The screenshot shows the 'Active Run' tab of the DynamicSpotter interface. The title bar indicates the run is for 'DynamicSpotter Run '2015-12-15 (08:44:42)' of project 'Showcase-Test-1204''. The main area displays a tree of controllers, each preceded by a green flag icon. The controllers are listed as follows:

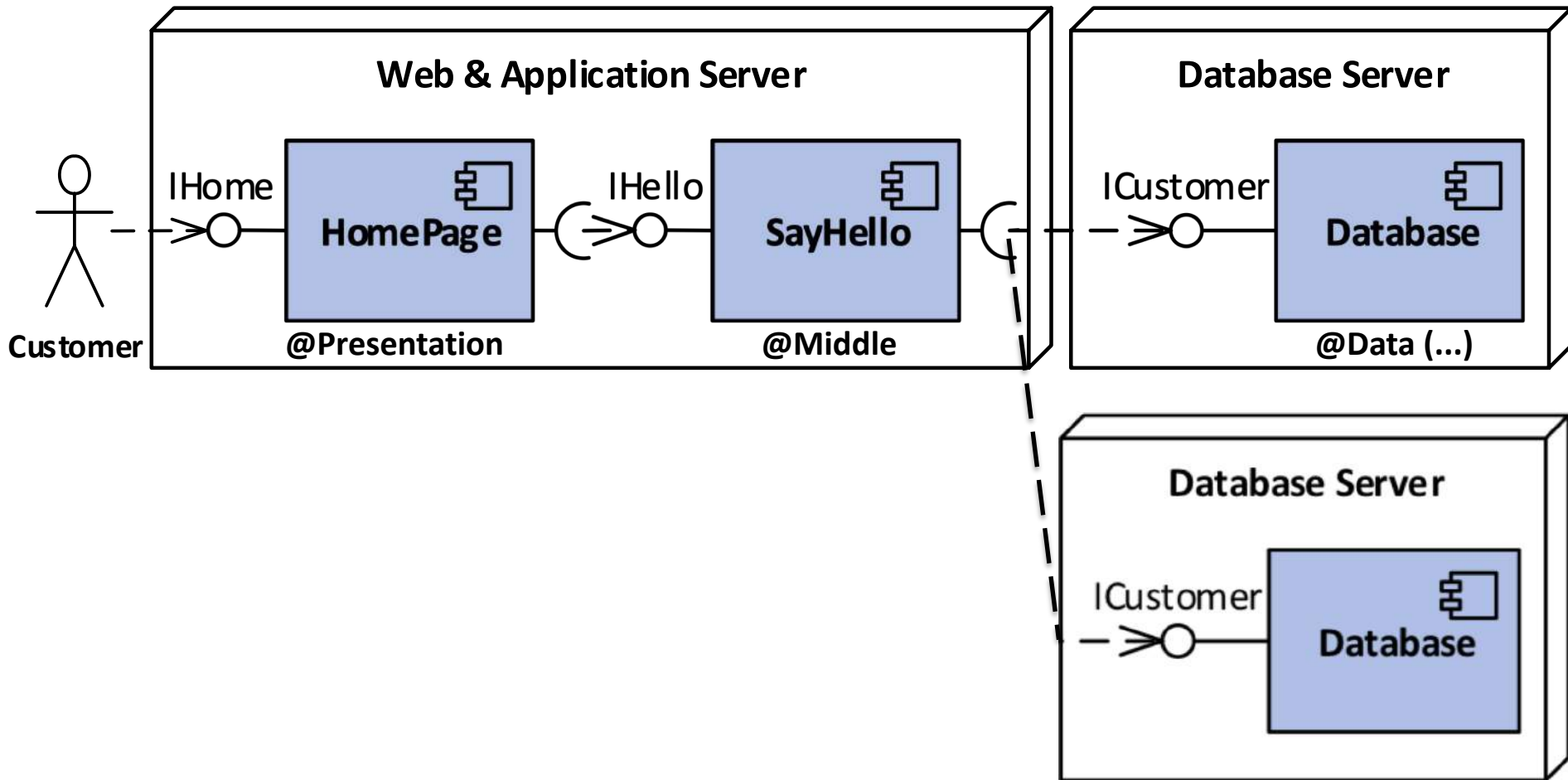
- org.spotter.ext.detection.appHiccups.AppHiccupsController
- ▼ org.spotter.ext.detection.continuousViolation.ContinuousViolationController
 - ▼ org.spotter.ext.detection.trafficJam.TrafficJamDetectionController
 - org.spotter.ext.detection.olb.OLBDetectionController
 - ▼ org.spotter.ext.detection.dbcongestion.DBCongestionDetectionController
 - org.spotter.ext.detection.stifle.StifleDetectionController
 - org.spotter.ext.detection.edc.EDCDetectionController
 - org.spotter.ext.detection.olb.OLBDetectionController
 - ▼ org.spotter.ext.detection.excessiveMessaging.ExcessiveMessagingDetectionController
 - org.spotter.ext.detection.blob.BlobDetectionController
 - org.spotter.ext.detection.est.EmptySemiTrucksDetectionController
- org.spotter.ext.detection.ramp.RampDetectionController

Reanalyse: Apply HowTo using AT

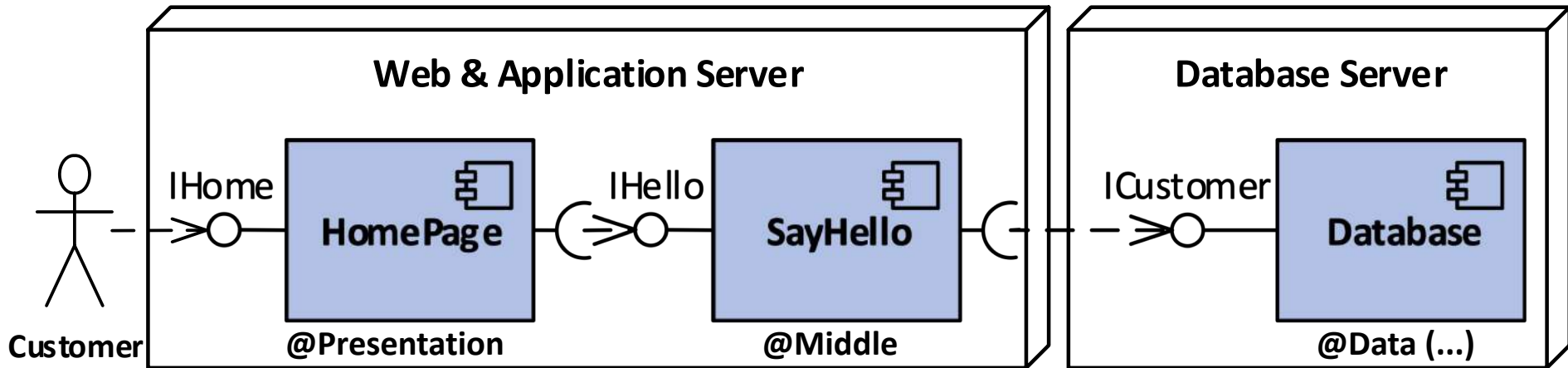


Architectural Template Application: „3-Layer Architecture with Replicable Database“

Reanalyse: Analysis Time Adaptations

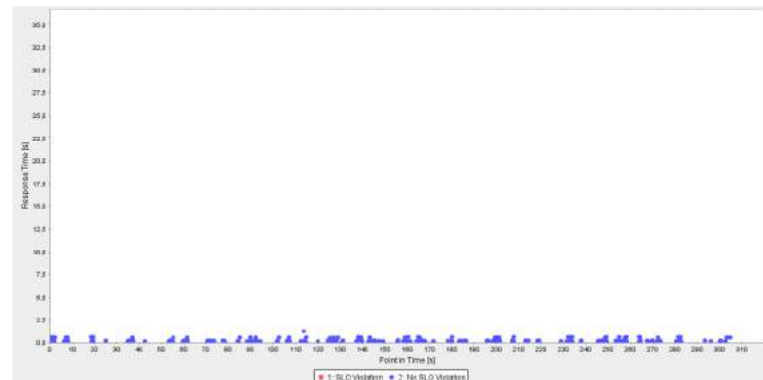


Reanalyse: Running the Analyser



Max. Customer
Capacity: 700

SLO Violations
over Time:



Talk Outline

Introduction

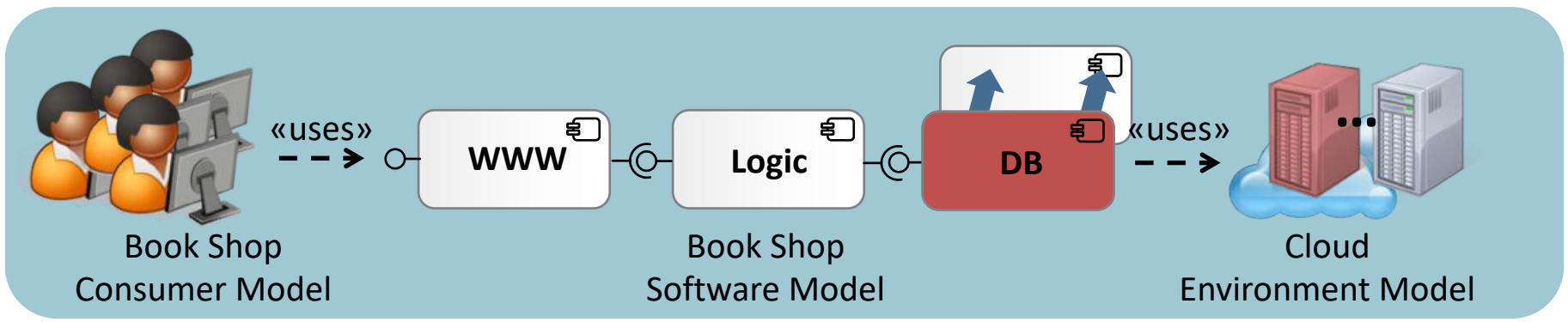
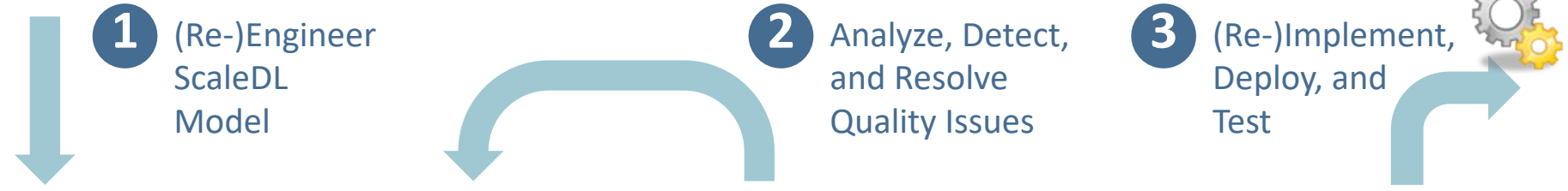
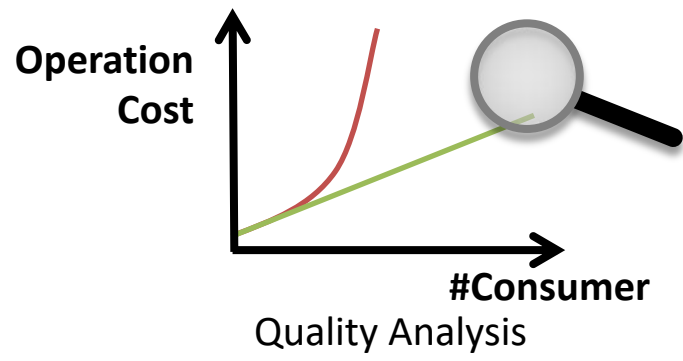
- Cloud Computing
- Running Example
- New Quality Properties to consider

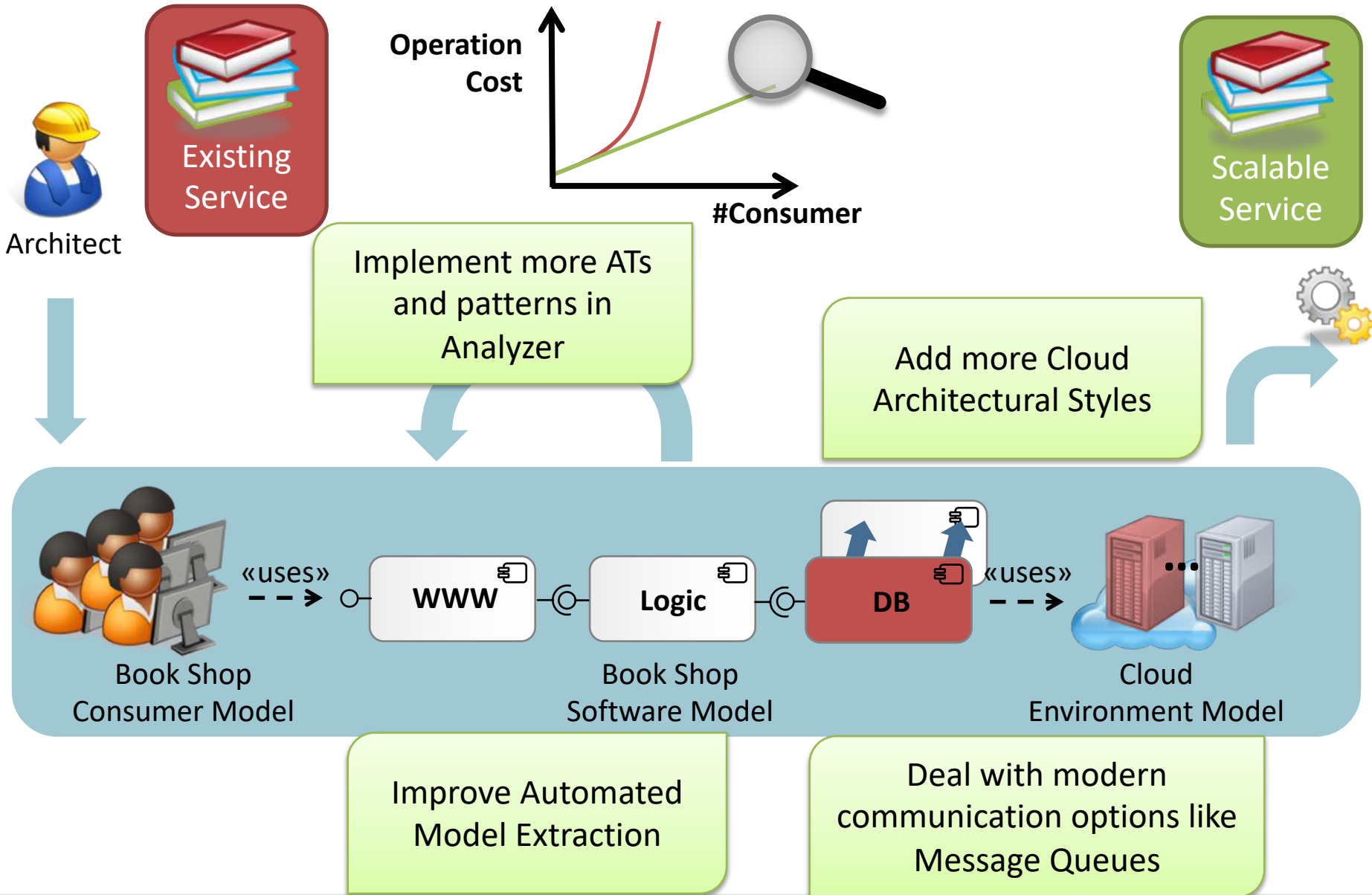
Cloud Scale

- Method Overview
- Forward Engineering
- Reengineering and Migration Support

Concluding Remarks

- Summary
- Future Work

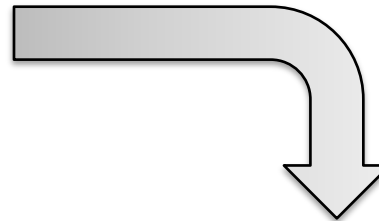




No more Traffic Jams!



[Rainer Sturm / pixelio.de]



www.iste.uni-stuttgart.de/rss
palladio-simulator.org



[birgitH / pixelio.de]