Engineering Scalable Cloud Systems

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

CloudScale

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]

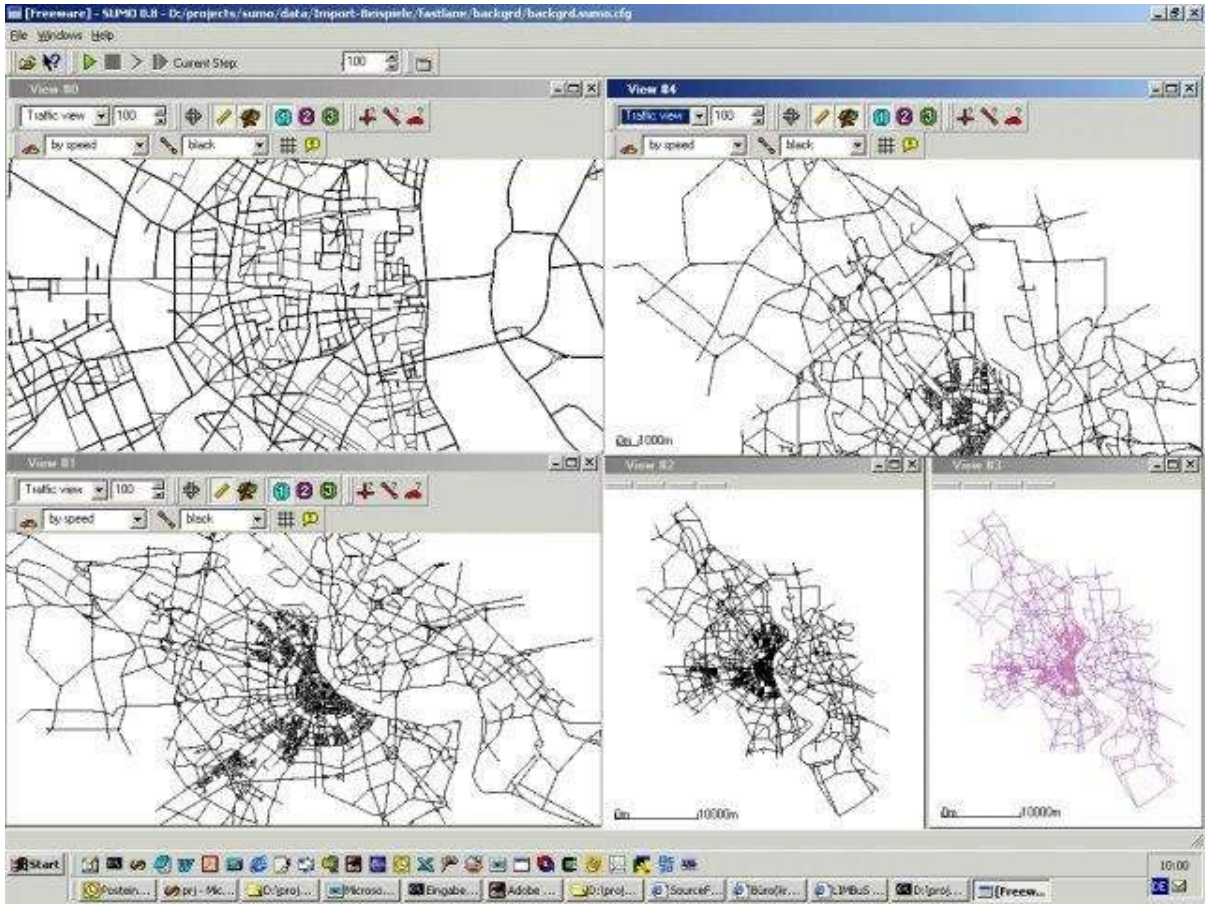
Legacy System

[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/]

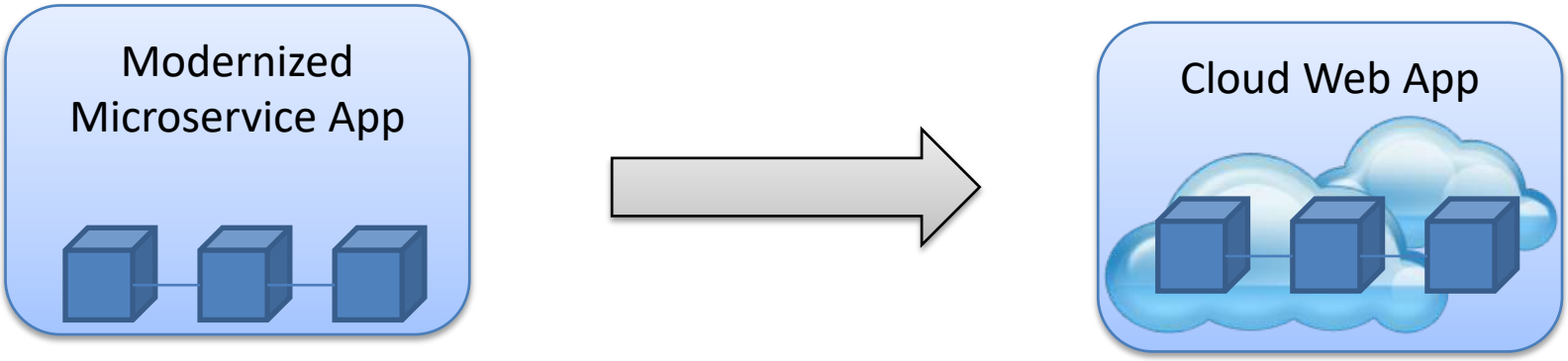
And IT?

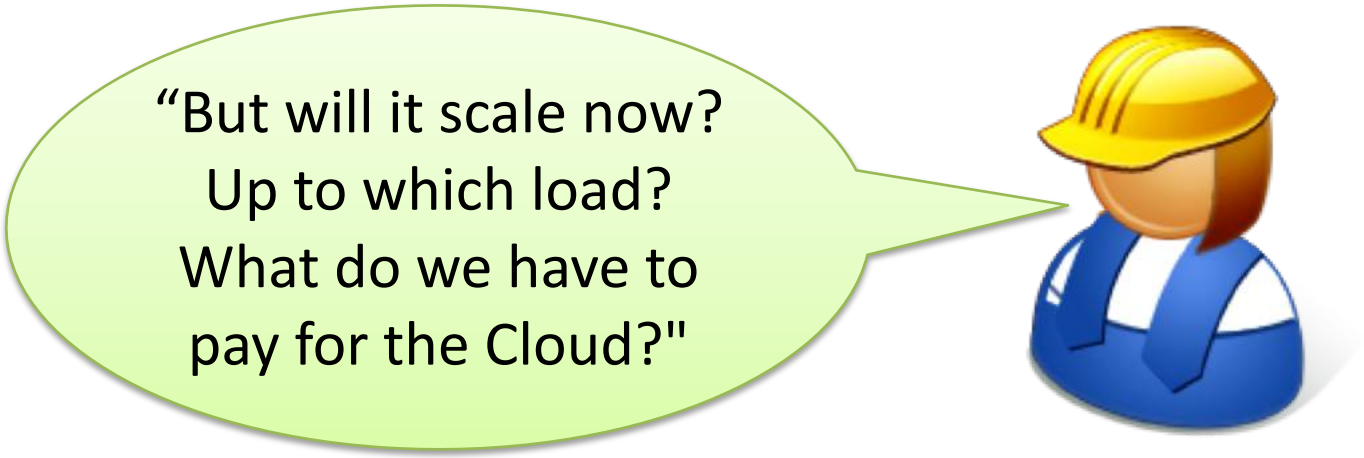
15-Year old
JSP Web Shop

Modernized
Microservice App

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

And IT?


Modernized
Microservice App
Cloud Web App


“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

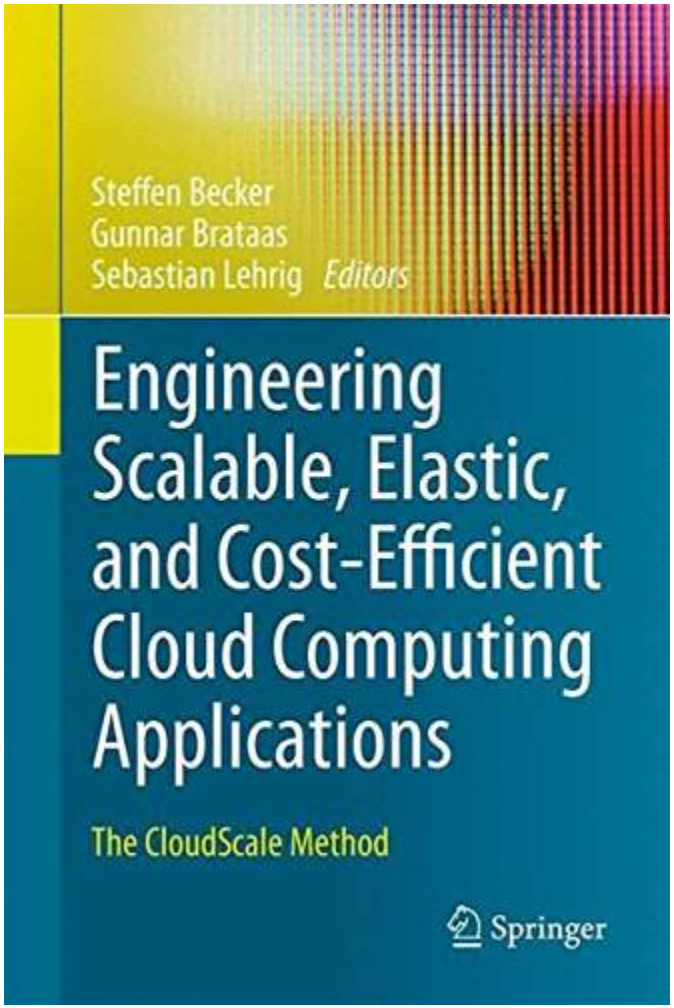
Cloud Scale

- 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

Consumer

«uses»

Online Book Shop

«uses»

Cloud Computing Environment

Architect

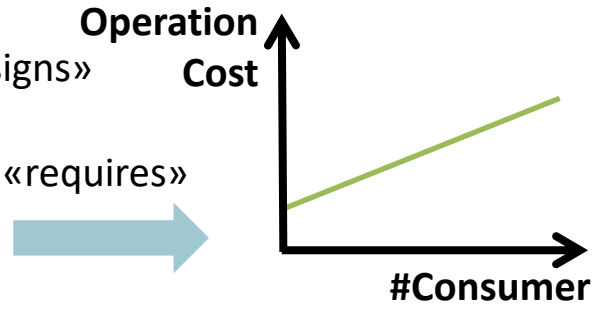
«suggests»

Cloud Computing

«assigns»

Manager

«requires»



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 situtation

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

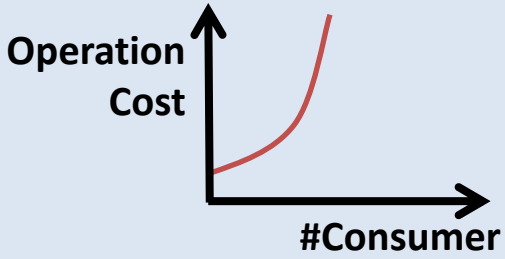
"the degree a cloud layer autonomously adapts **capacity** to workload over time"

- **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

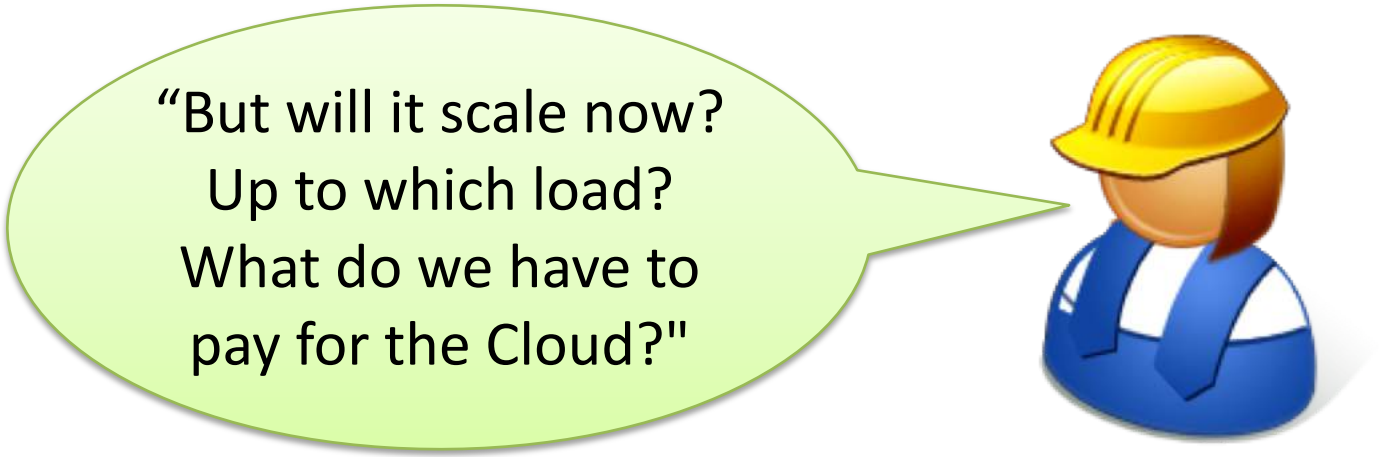
- Cloud Computing
- Running Example
- New Quality Properties to consider

Cloud Scale

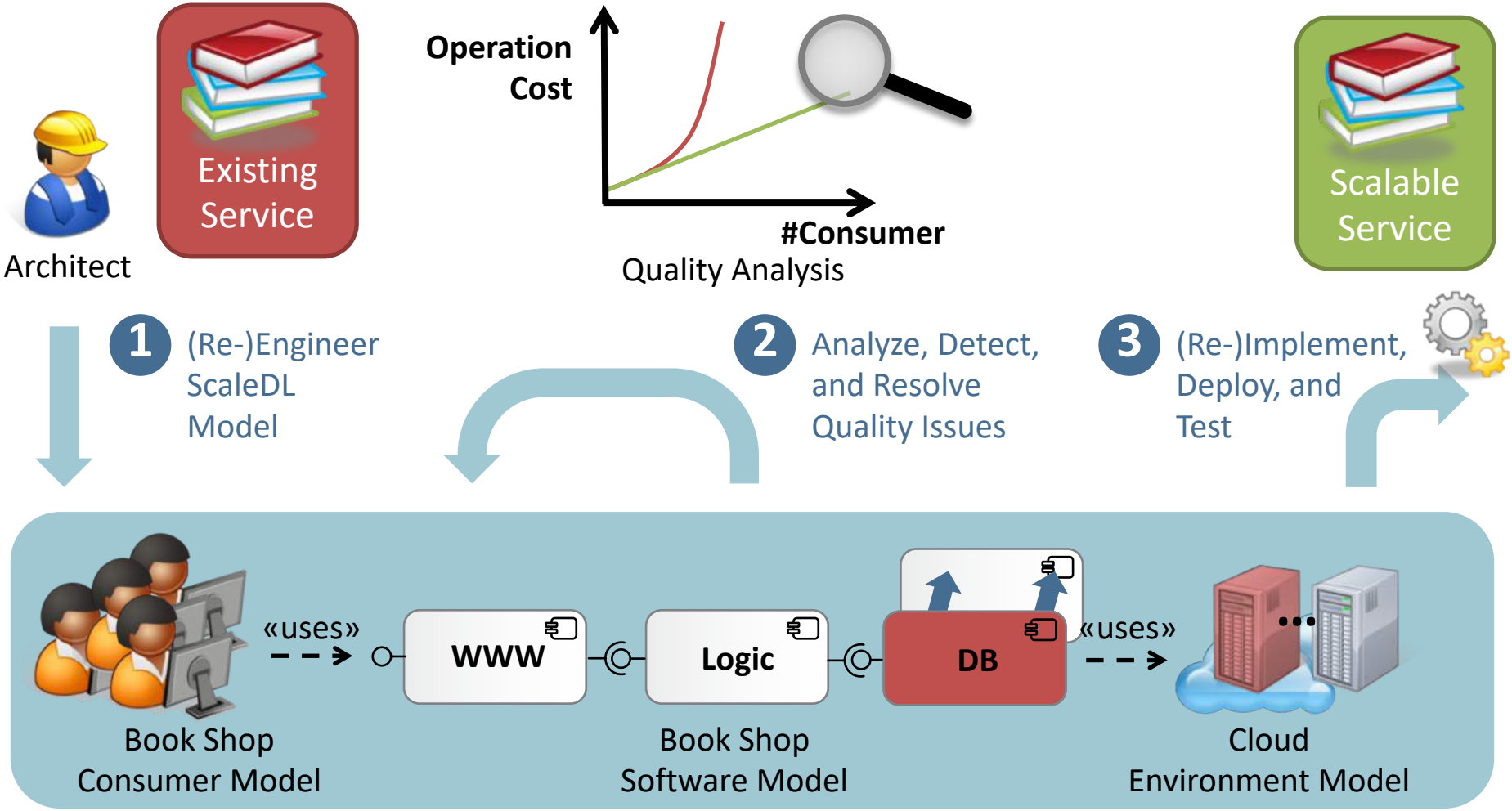
- Method Overview
- Forward Engineering
- Reengineering and Migration Support

Concluding Remarks

- Summary
- Future Work

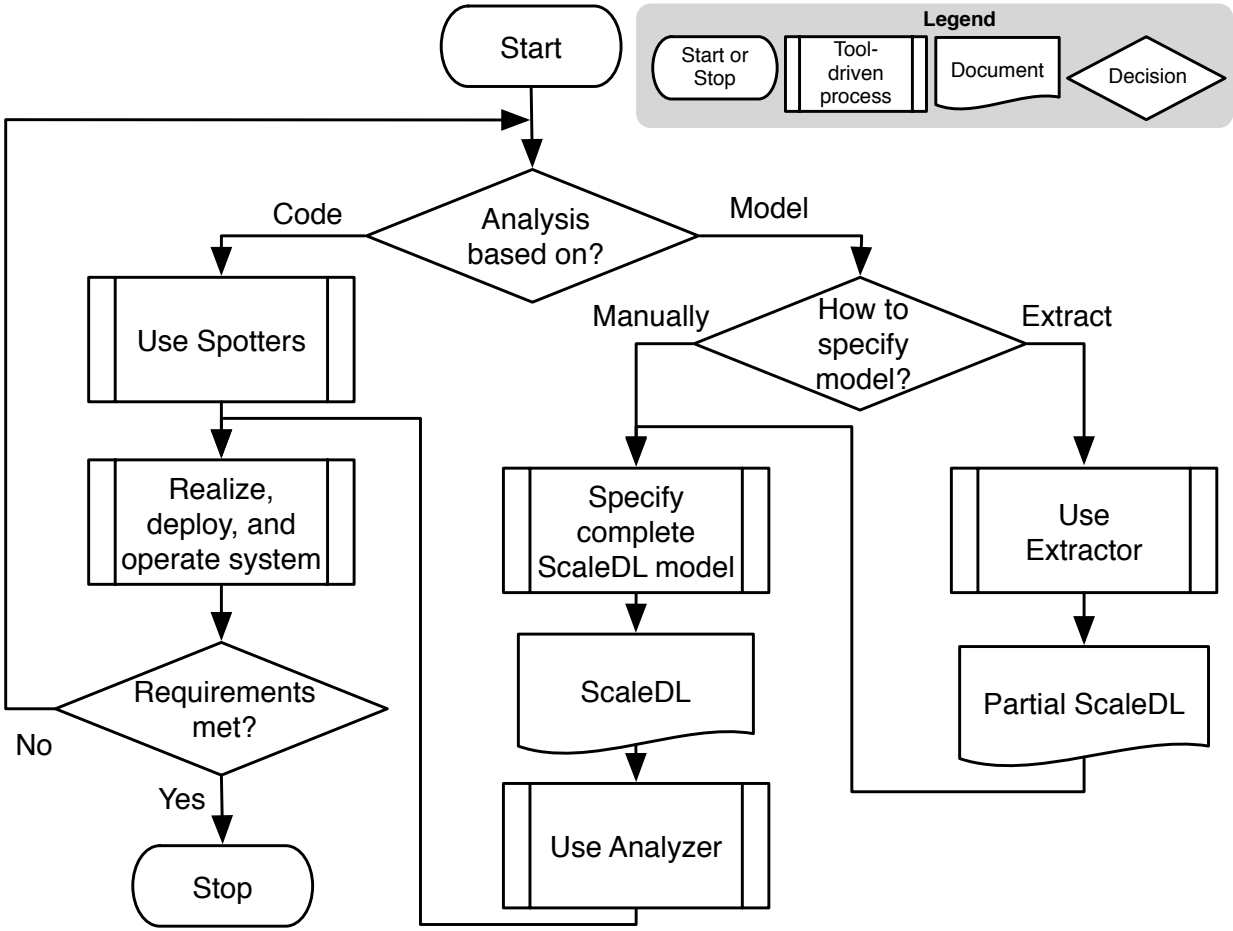


CLOUD SCALE METHOD



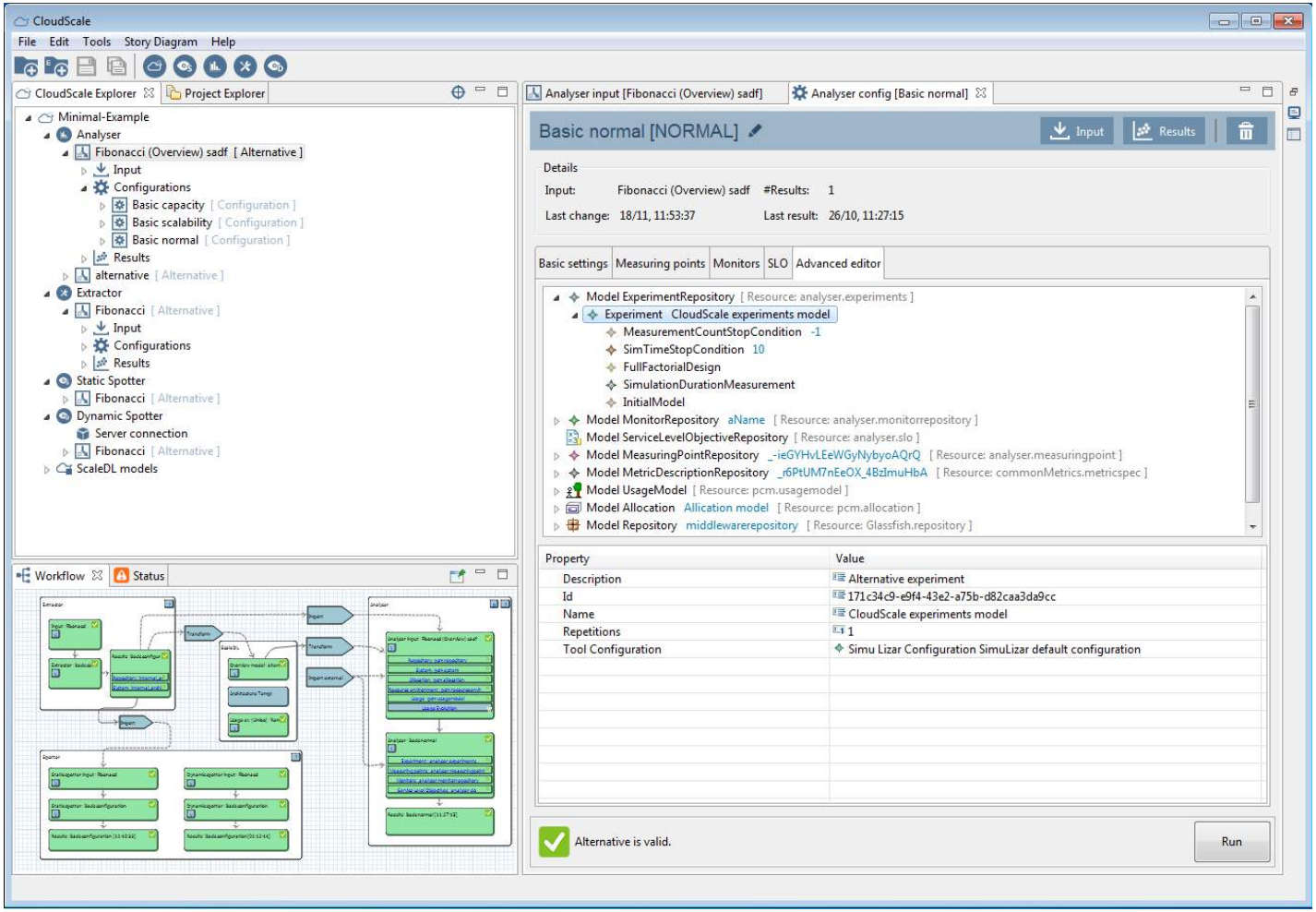
ScaleDL Model

CloudScale Method



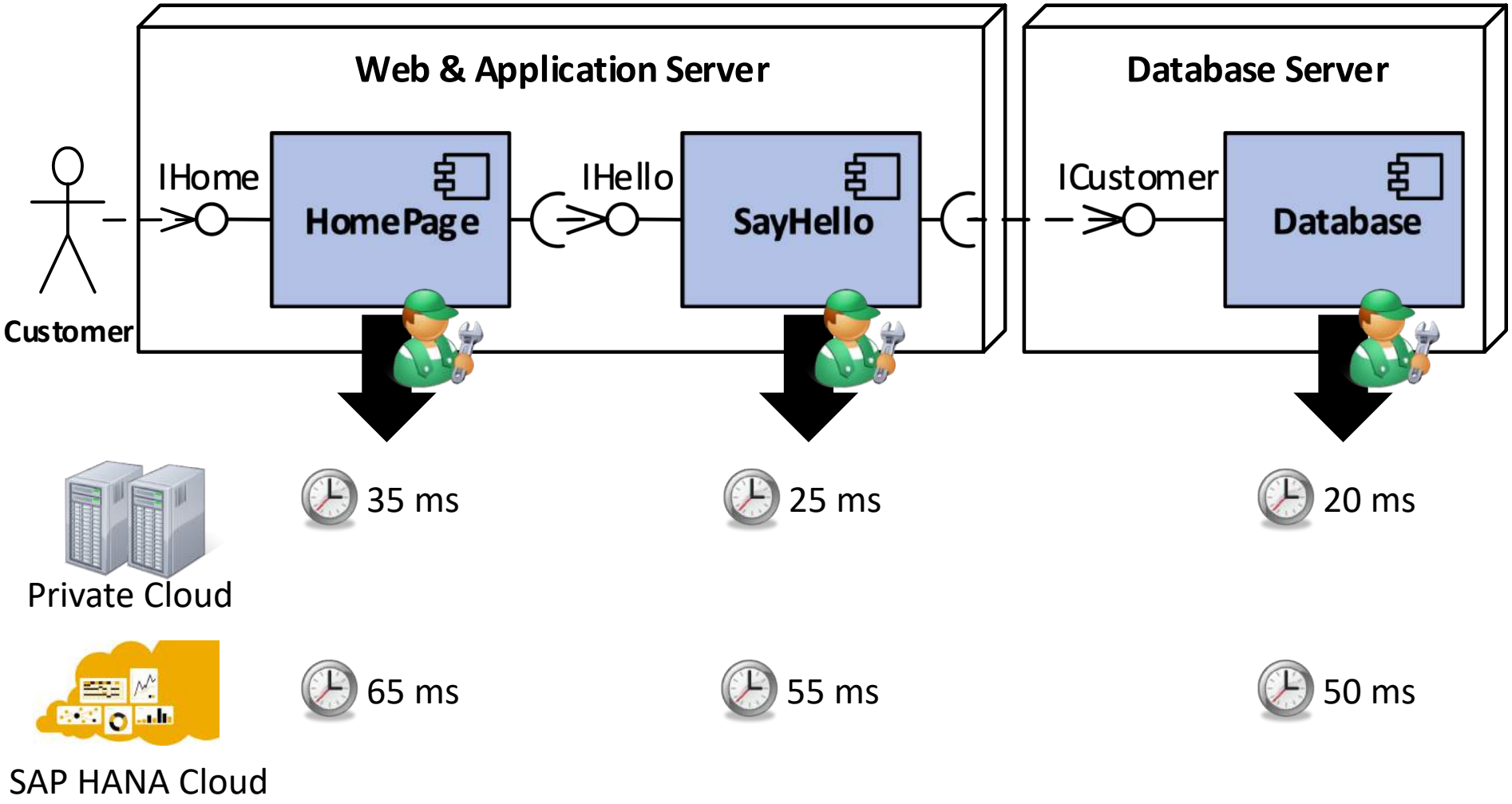
CloudScale

CloudScale IDE

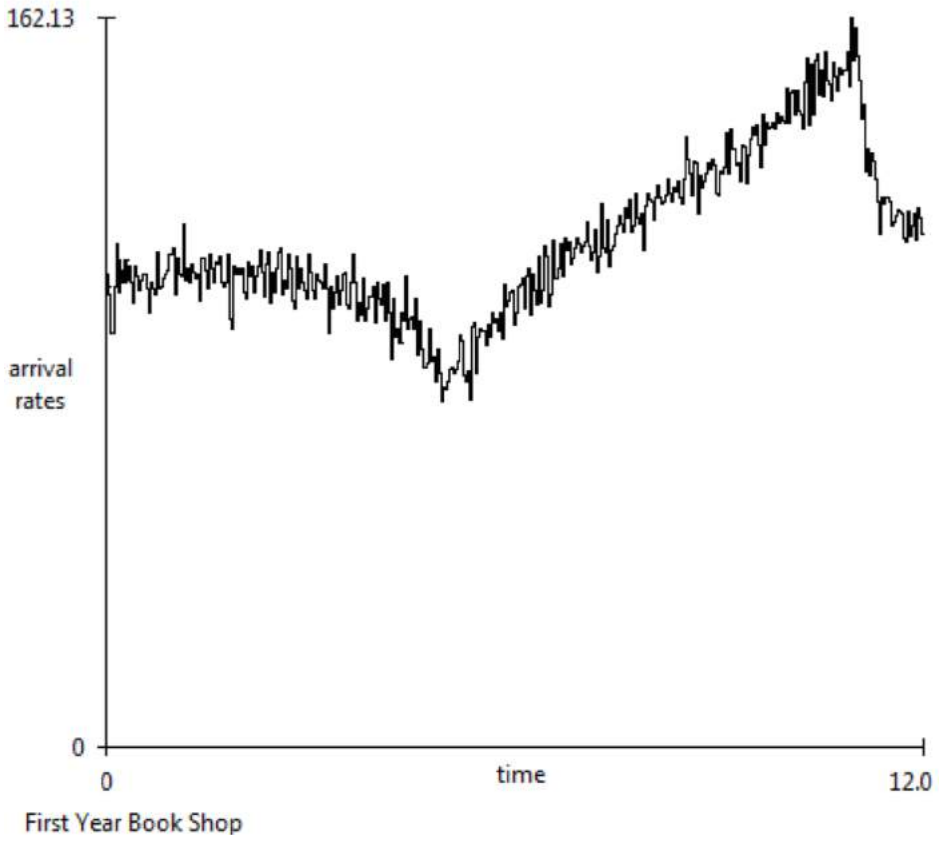


FORWARD ENGINEERING

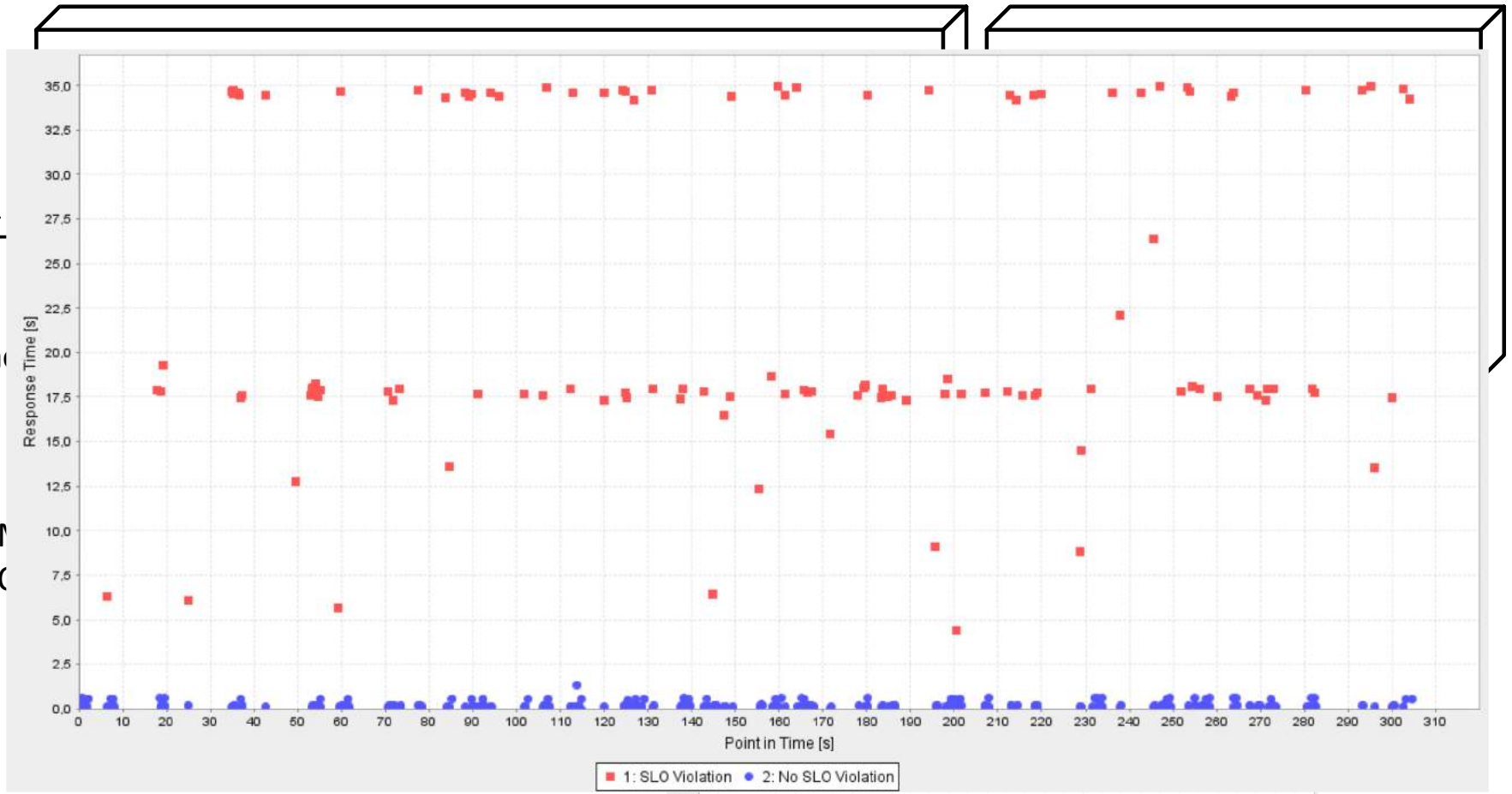
Analyse: Calibrate Model



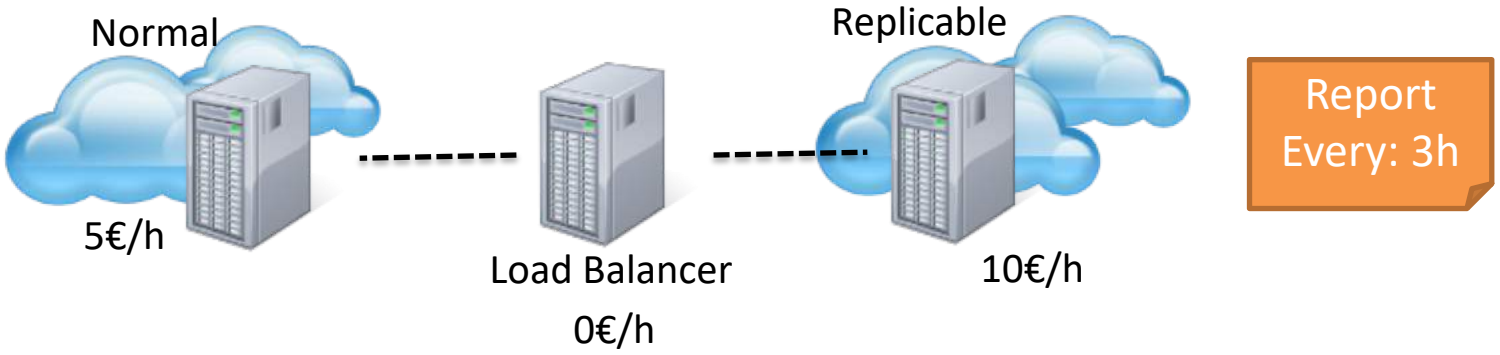
Analyse: Model Workload Evolution



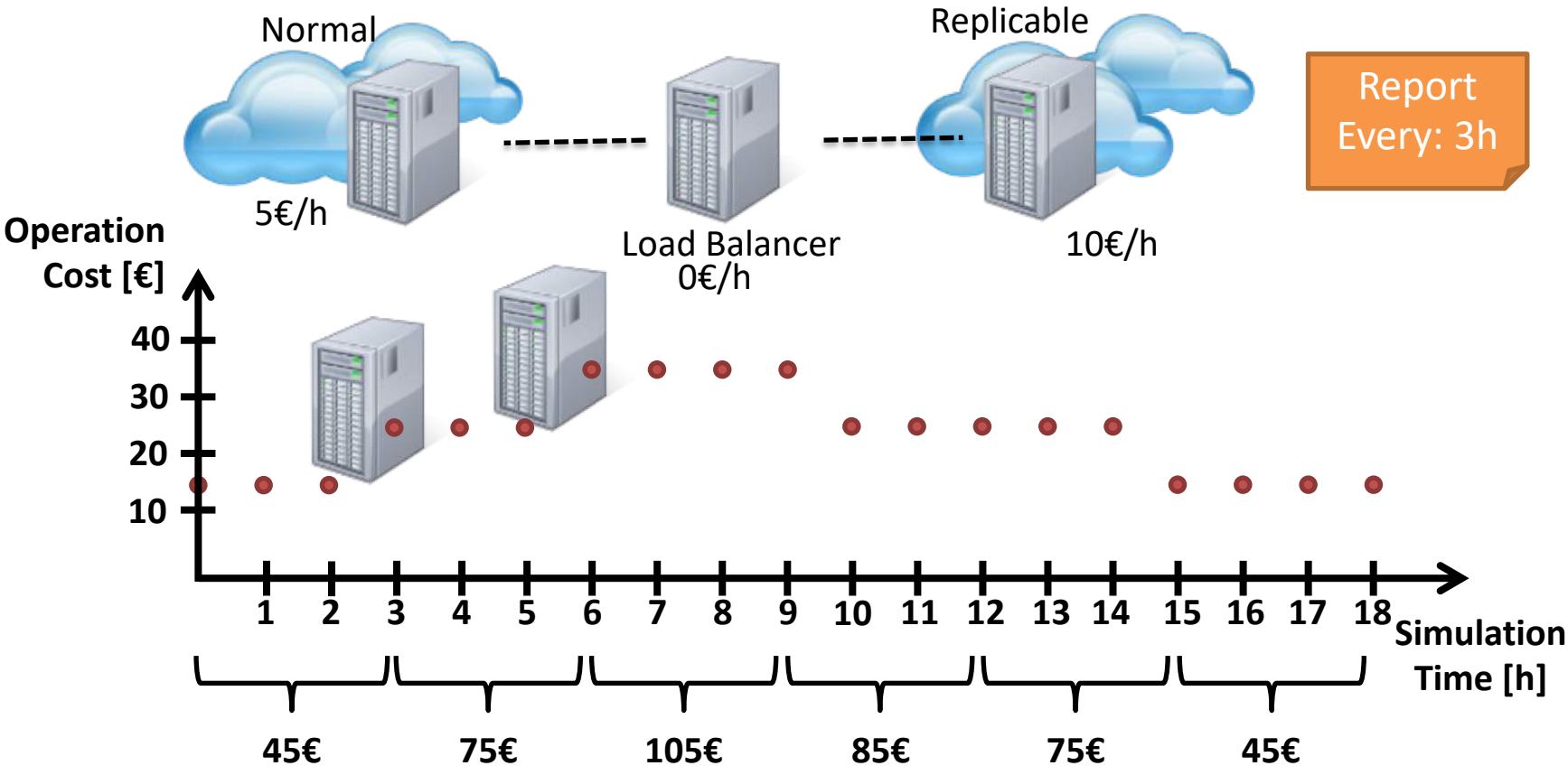
Analyse: Running the Analyser



Measuring Cost



Measuring Cost



CloudScale

REENGINEERING AND MIGRATION SUPPORT

Sketching the General Idea

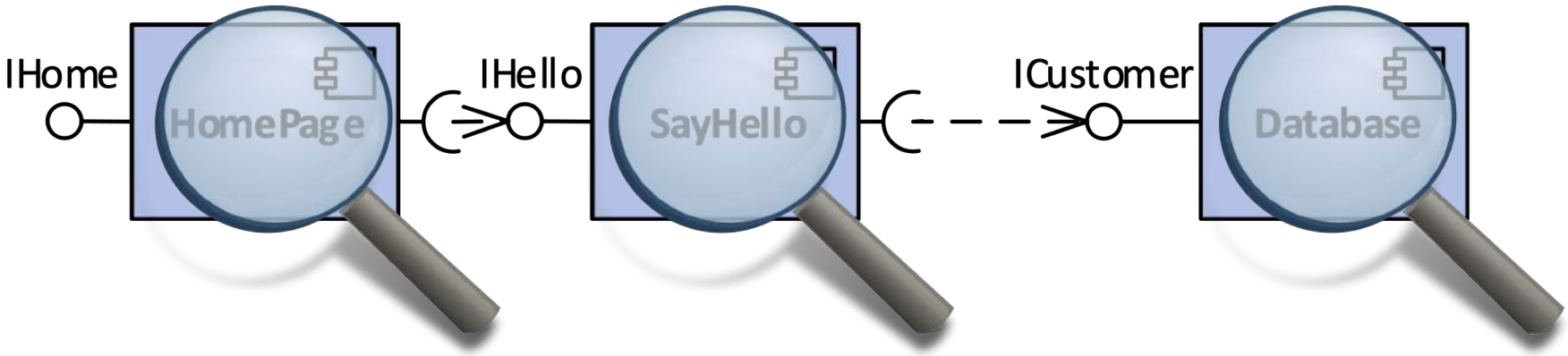
15-Year old
JSP Web Shop

Any Symptoms?

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

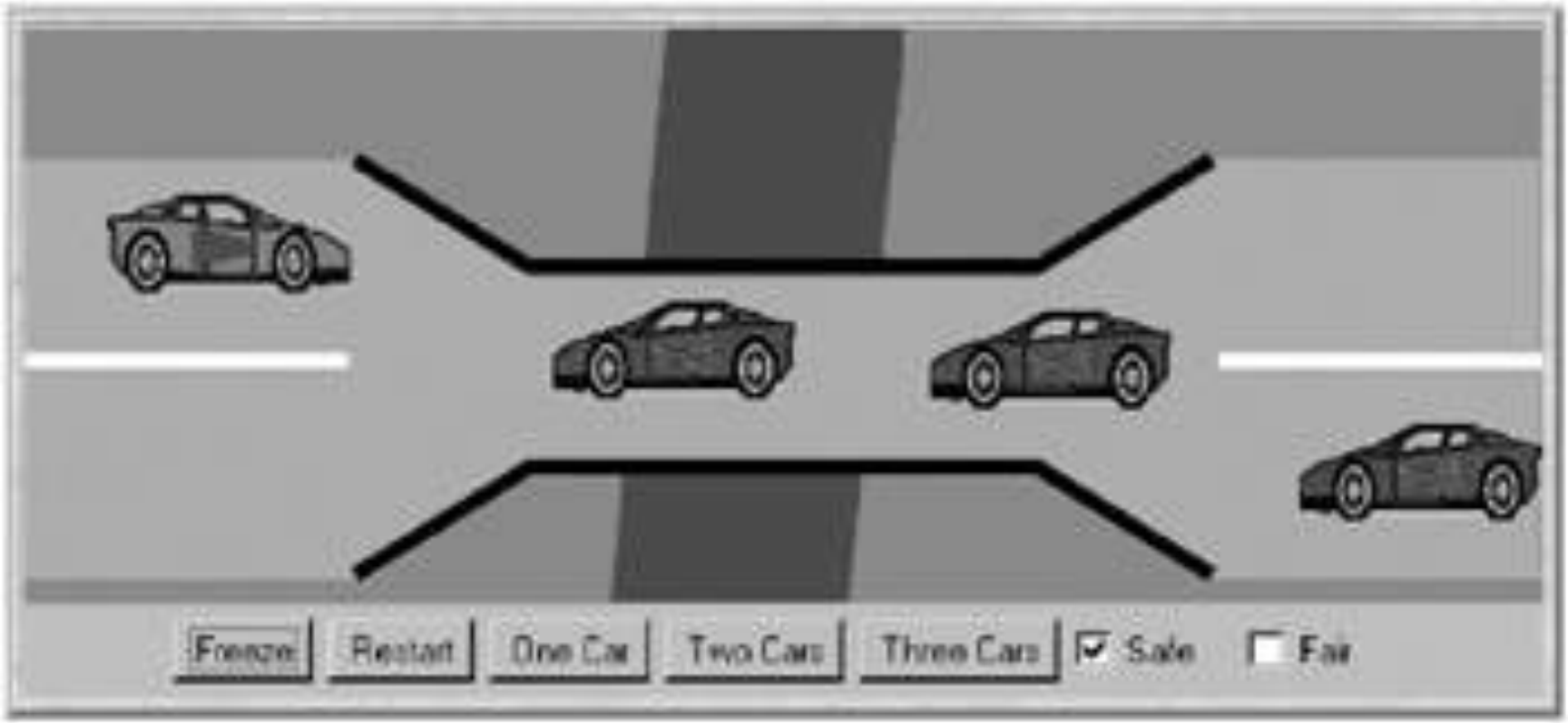
Then I know what's your problem, my friend! You've got DB Congestion issues!

SpotHowNotTos: Static Spotter



First, let's see how the code does…

HowNotTo Example: One Lane Bridge



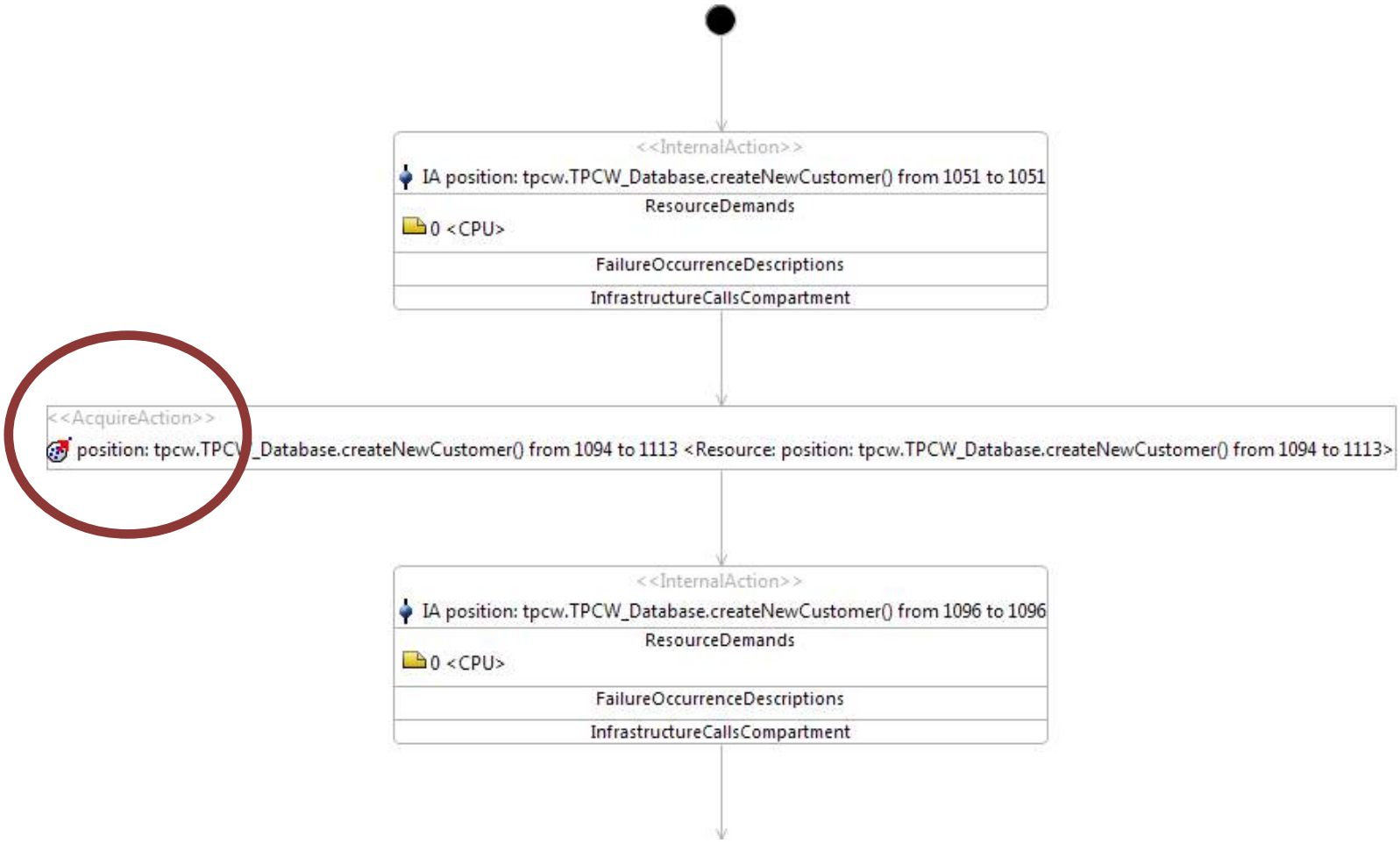
[J. Magee, J. Kramer (2006): *Concurrency: State Models and Java Programs*]

Code to identify…

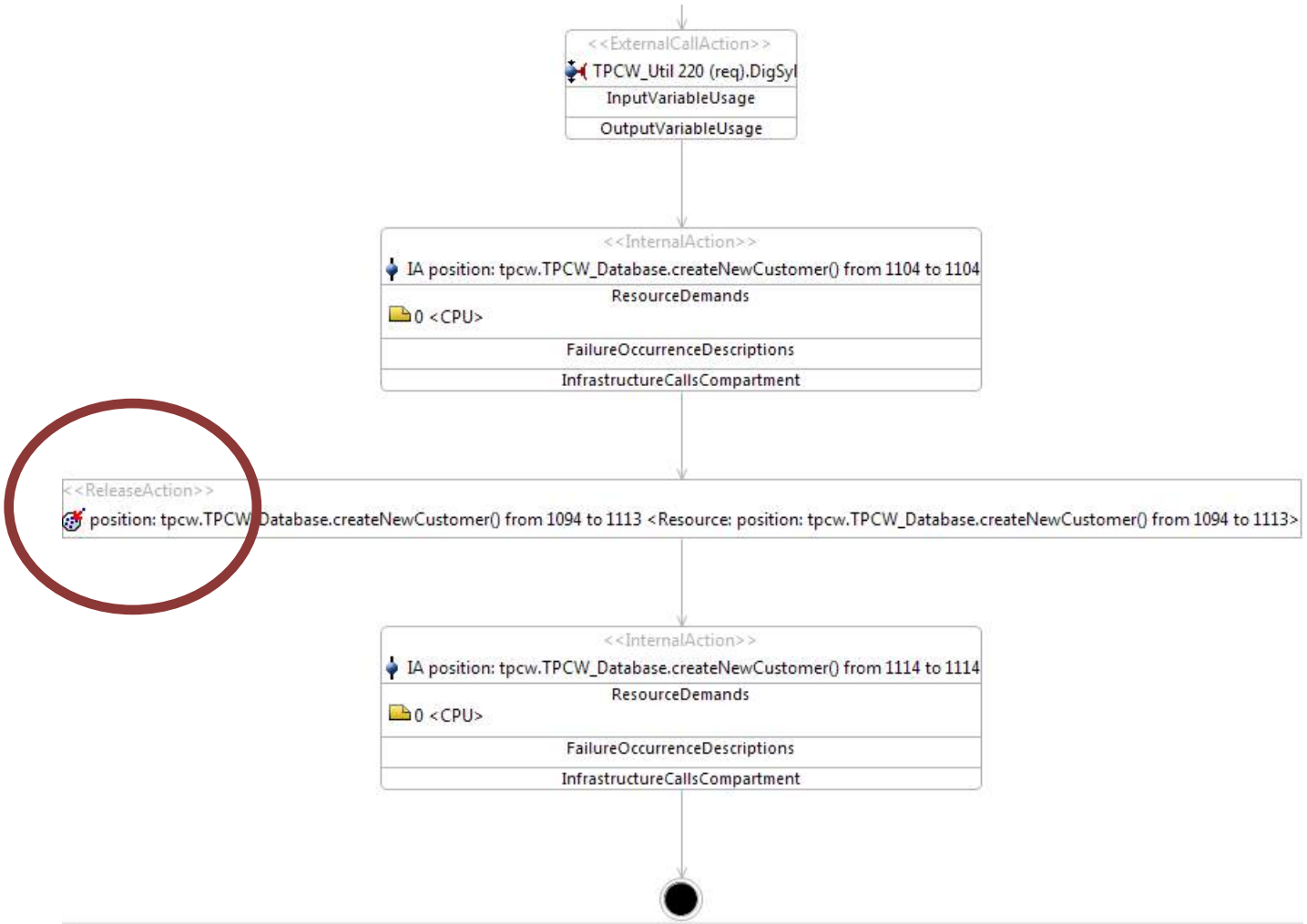
Which code could cause a one lane bridge?

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

One AcquireReleasePair found: tpcw.TPCW_Database.createNewCustomer()



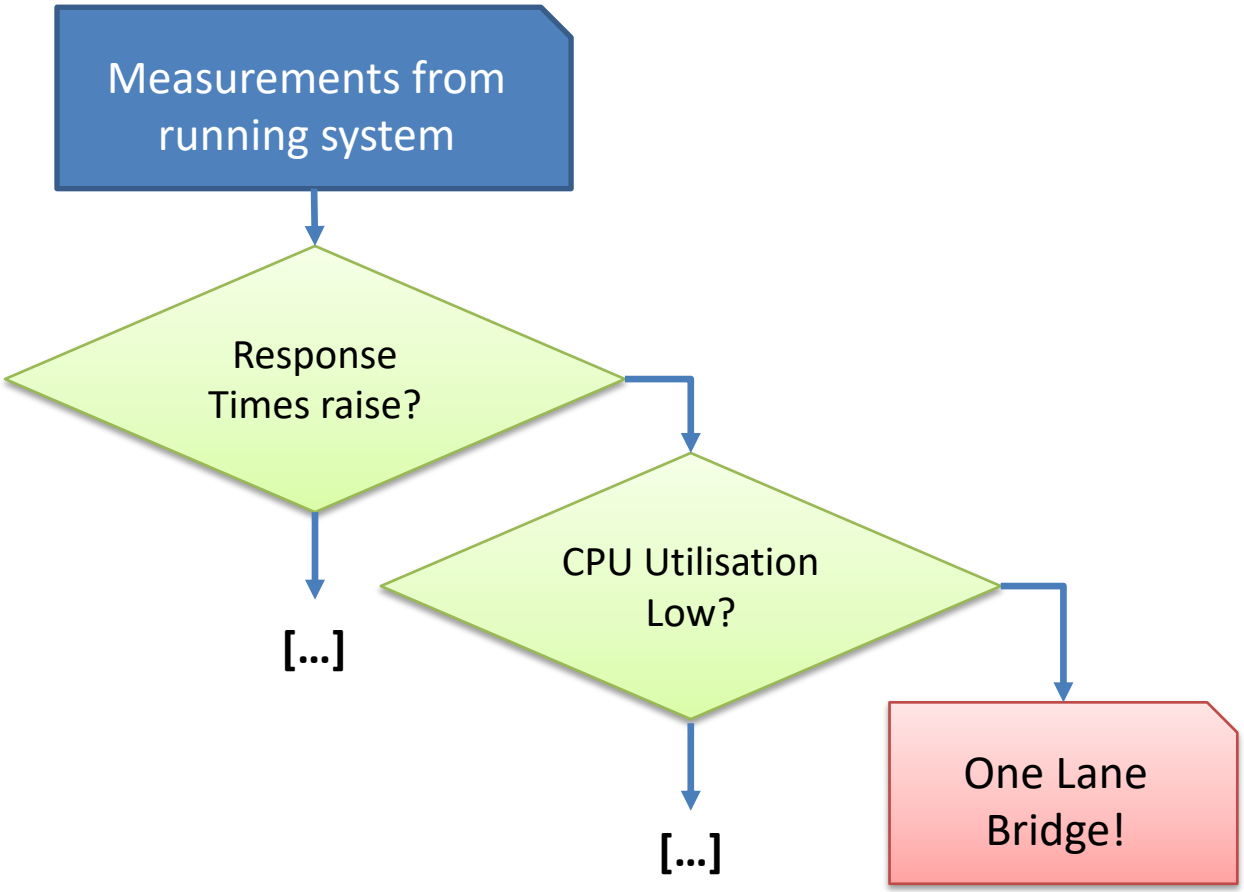
One AcquireReleasePair found: tpcw.TPCW_Database.createNewCustomer()



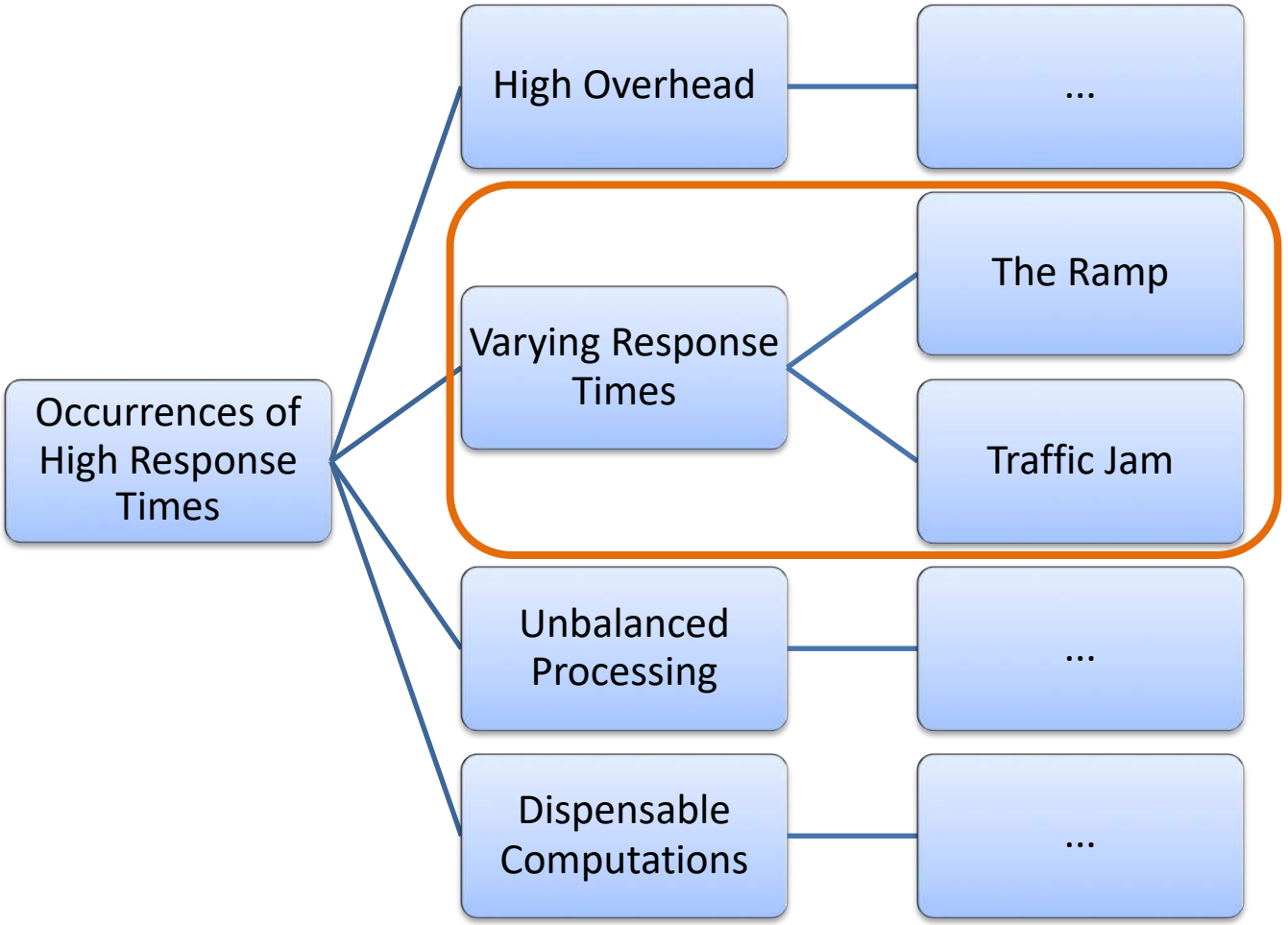
Spot: Running the Static Spotter

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.g
SynchronizedMethod	100,00%	synchronizedMethods=org.eclipse.g
SynchronizedMethod	100,00%	synchronizedMethods=org.eclipse.g
SynchronizedMethod	100,00%	synchronizedMethods=org.eclipse.g
SynchronizedMethod	100,00%	synchronizedMethods=org.eclipse.g

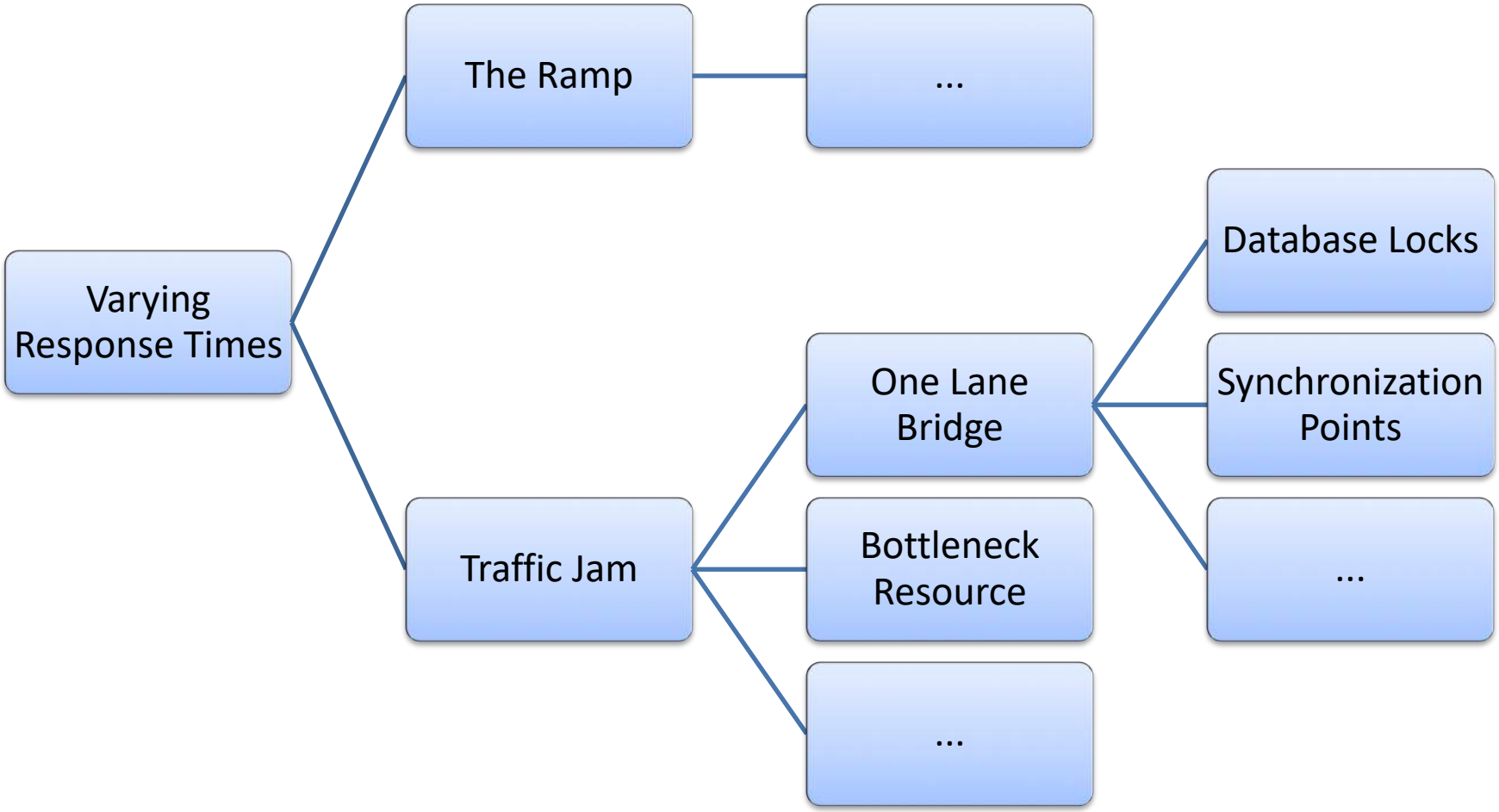
Dynamic Spotter: Problem Hierachy



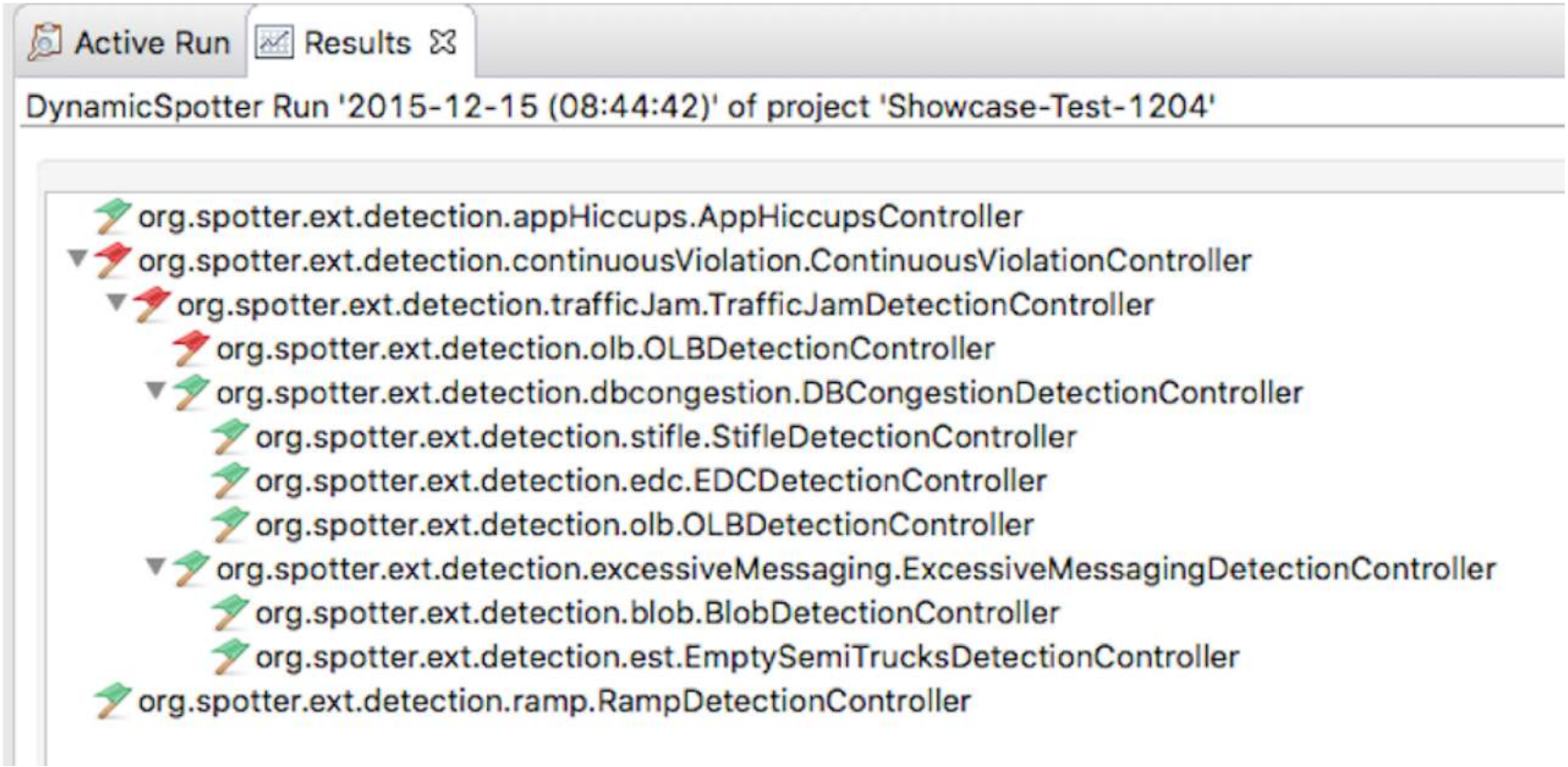
Problem Hierarchy: Excerpt



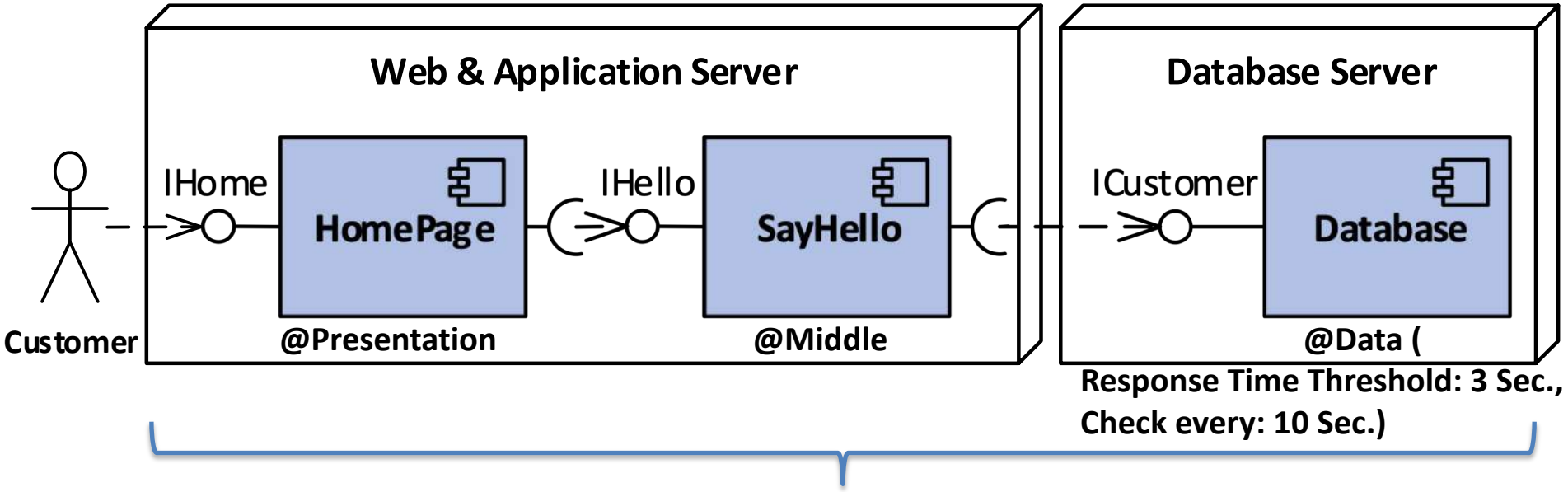
Problem Hierarchy: Excerpt cont.



Running Dynamic Spotter

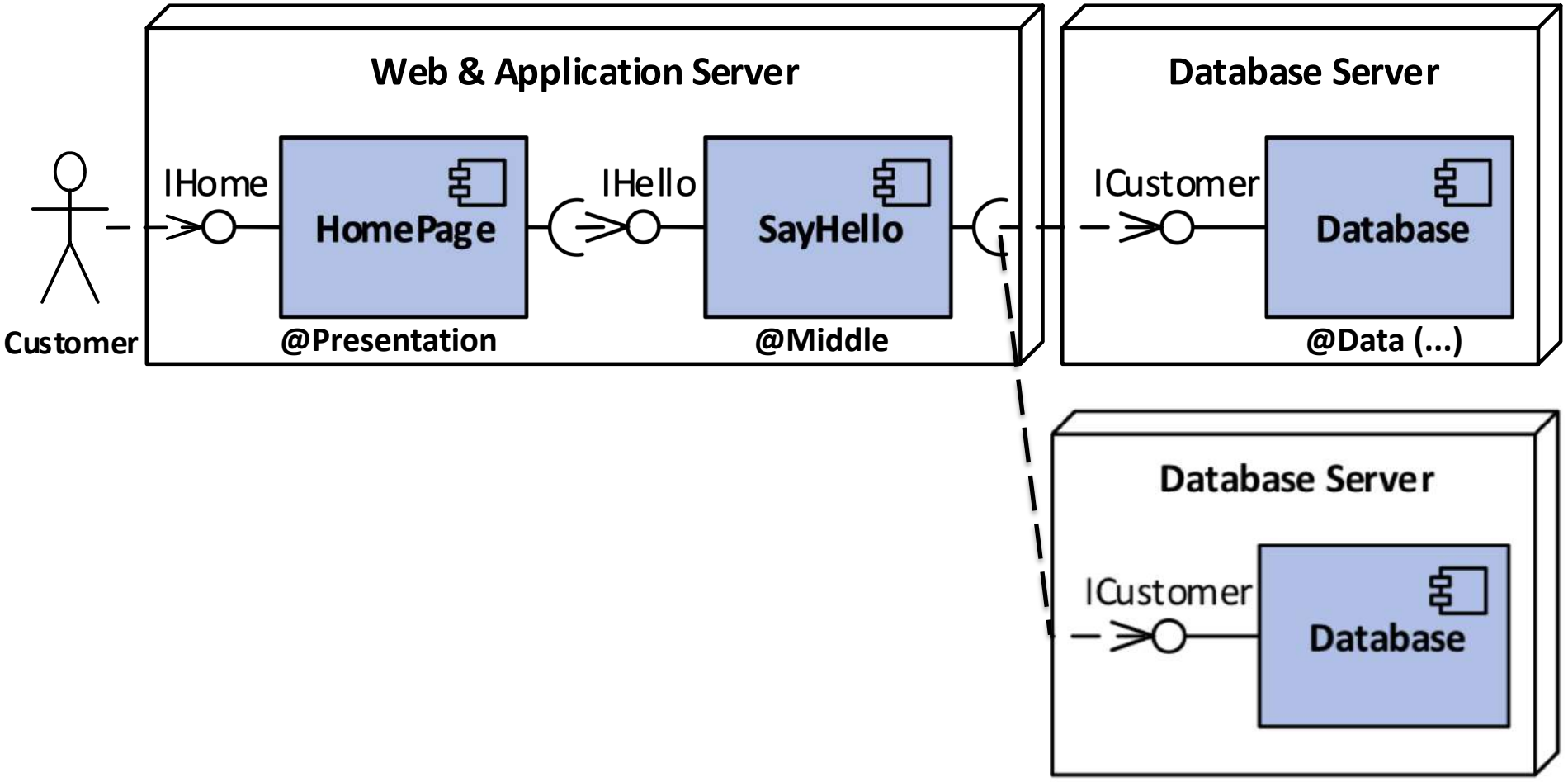


Reanalyse: Apply HowTo using AT

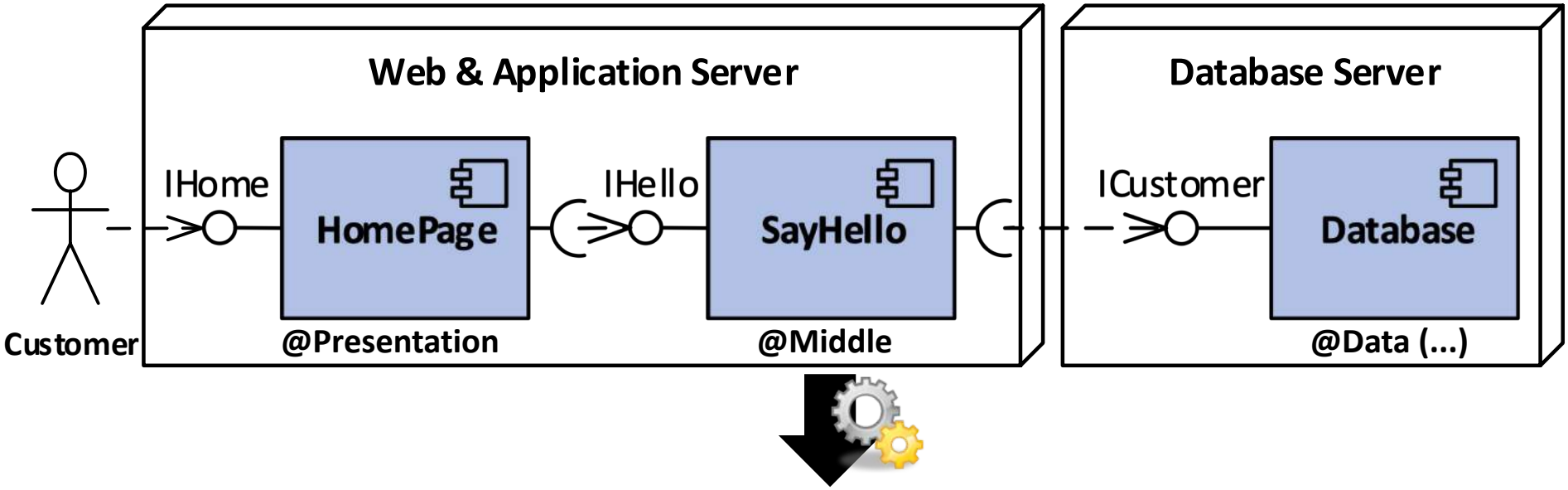


Architectural Template Application: „3-Layer Architecture with Replicable Database“

Reanalyse: Analysis Time Adaptations

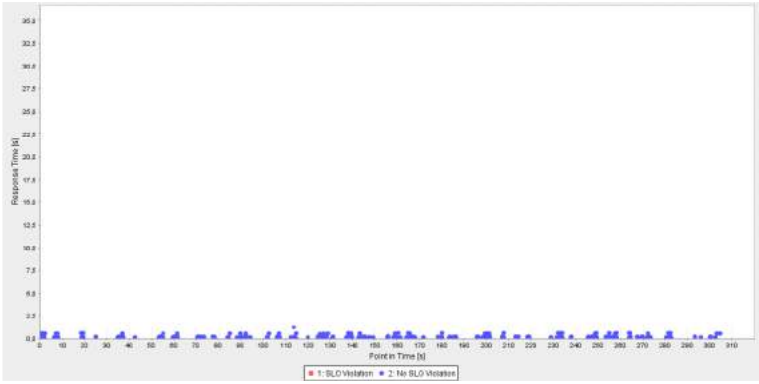


Reanalyse: Running the Analyser



Max. Customer Capacity: 700

SLO Violations over Time:



Talk Outline

Introduction

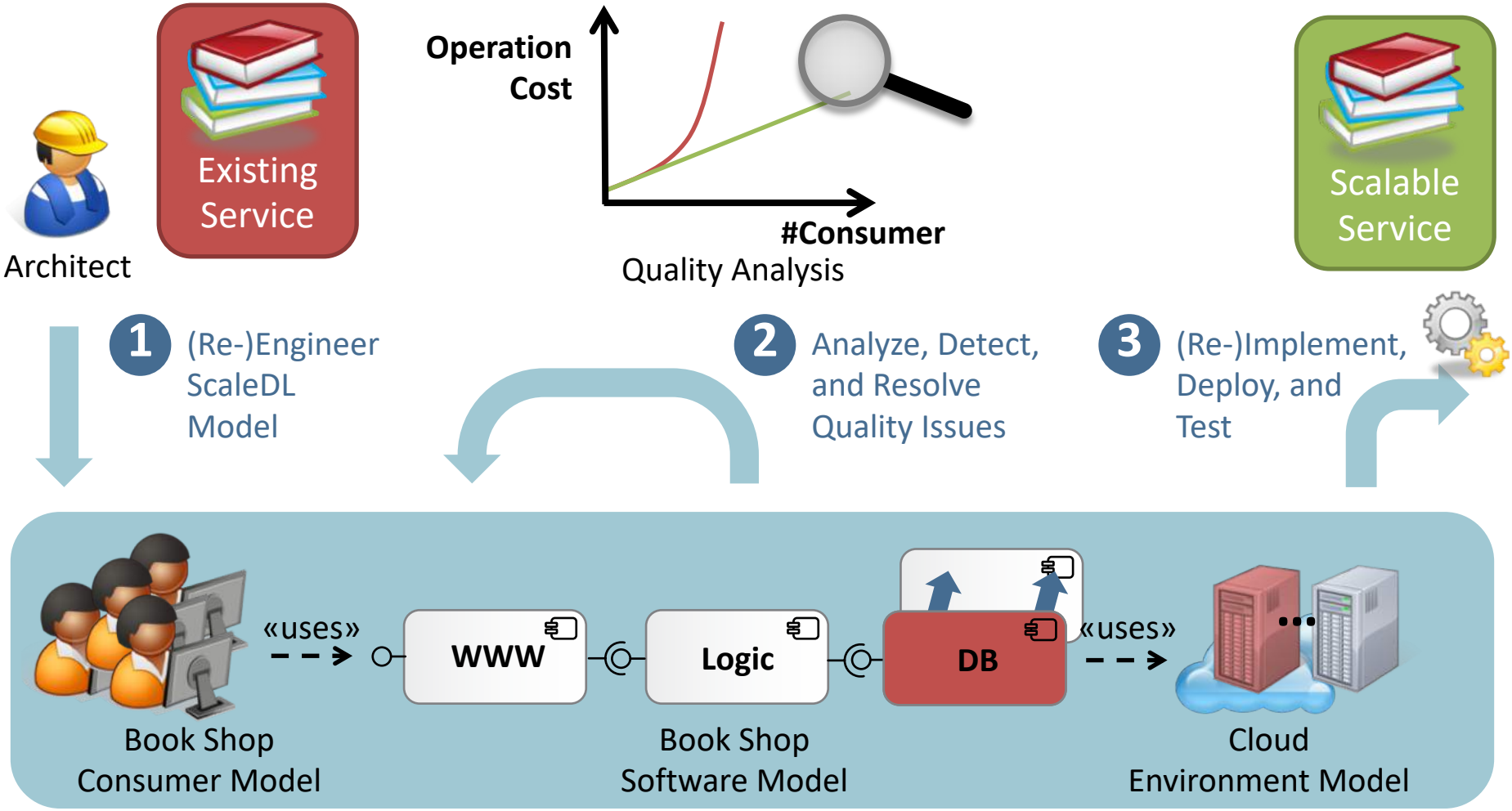
- Cloud Computing
- Running Example
- New Quality Properties to consider

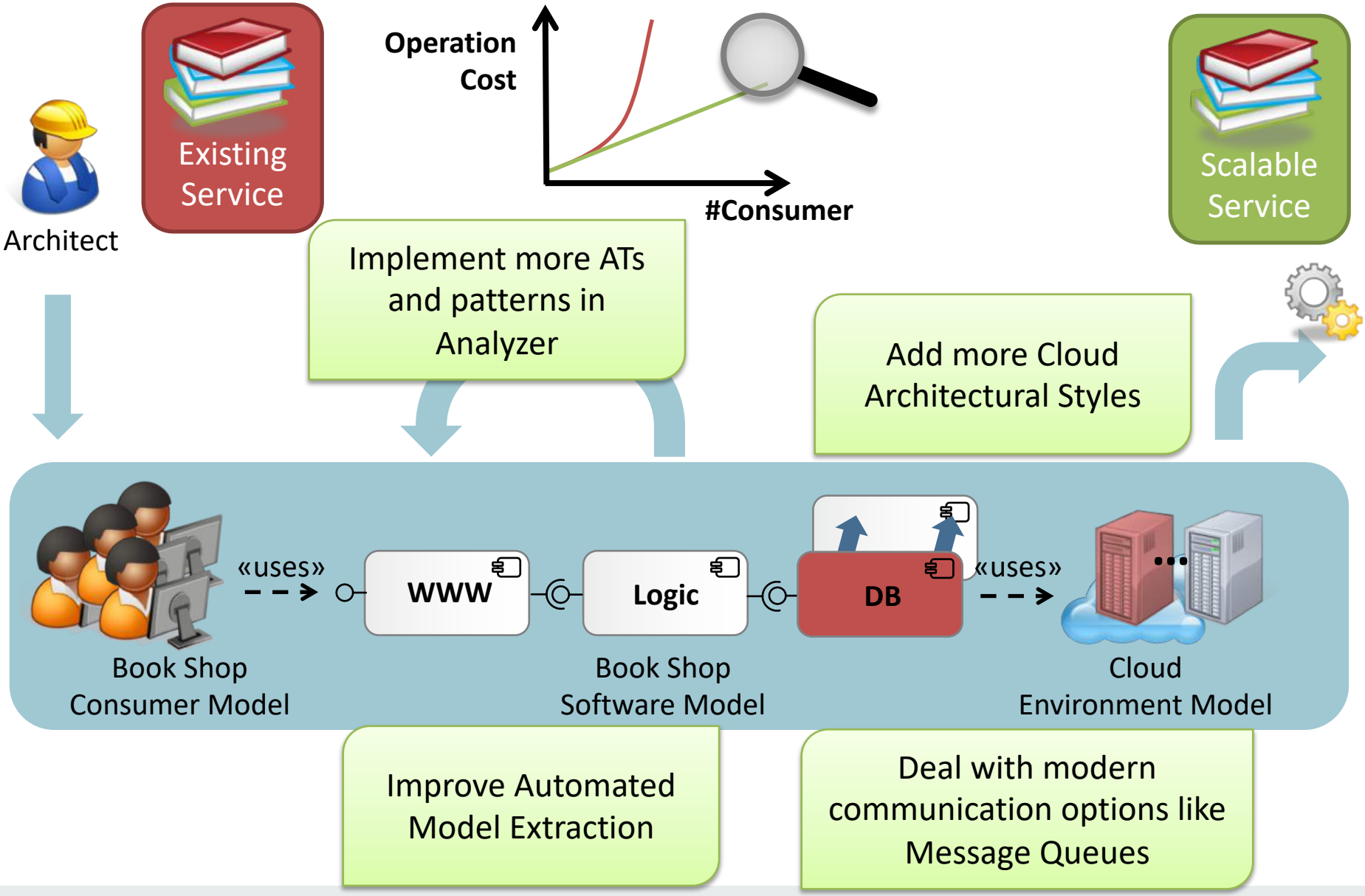
Cloud Scale

- Method Overview
- Forward Engineering
- Reengineering and Migration Support

Concluding Remarks

- Summary
- Future Work


Existing Service
Architect
Operation Cost
#Consumer
Quality Analysis
Scalable Service
1 (Re-)Engineer ScaleDL Model
2 Analyze, Detect, and Resolve Quality Issues
3 (Re-)Implement, Deploy, and Test
«uses»
WWW
Logic
DB
«uses»
Book Shop Consumer Model
Book Shop Software Model
Cloud Environment Model


Existing Service
Operation Cost
#Consumer
Scalable Service
Architect
Implement more ATs and patterns in Analyzer
Add more Cloud Architectural Styles
«uses»
WWW
Logic
DB
«uses»
Book Shop Consumer Model
Book Shop Software Model
Cloud Environment Model
Improve Automated Model Extraction
Deal with modern communication options like Message Queues

No more Traffic Jams!

[Rainer Sturm / pixelio.de]

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

[birgitH / pixelio.de]