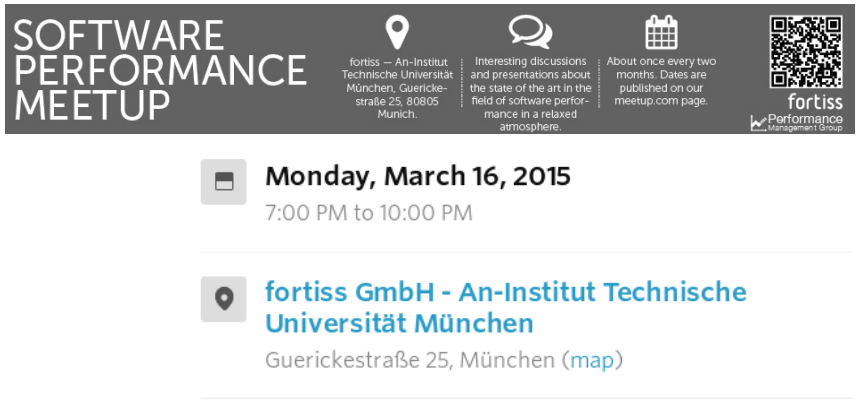
University of Stuttgart
Germany

Tool-Supported Application Performance Problem Detection and Diagnosis

André van Hoorn



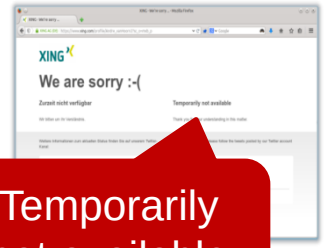
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

University of Stuttgart
Institute of Software Technology, Reliable Software Systems Group

http://www.iste.uni-stuttgart.de/rss/

Agenda

1. • Introduction – Performance Problems
2. • Kieker – Open Source APM Framework
3. • Performance Problem Detection and Diagnosis with Kieker
4. • Conclusions and Future Work

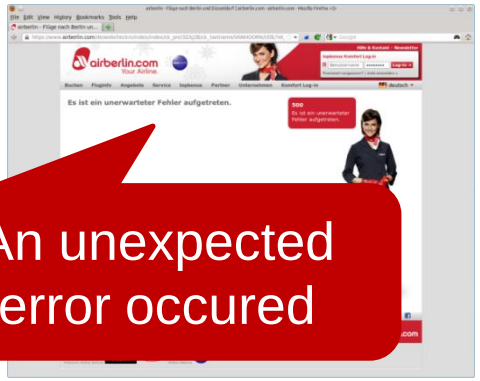


Kieker

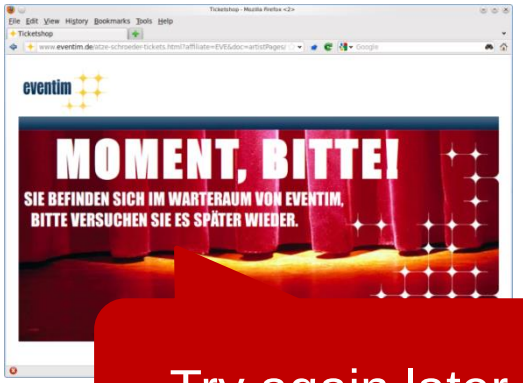
diagnoseIT

spec RG DevOps Performance

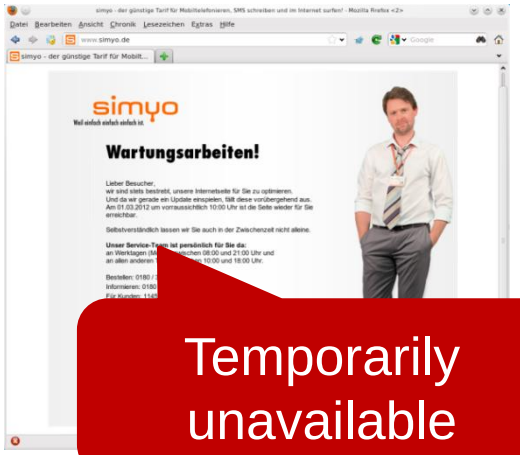
Performance (QoS) Problem Detection and Diagnosis



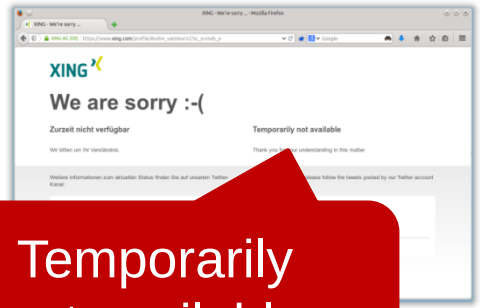
An unexpected error occured



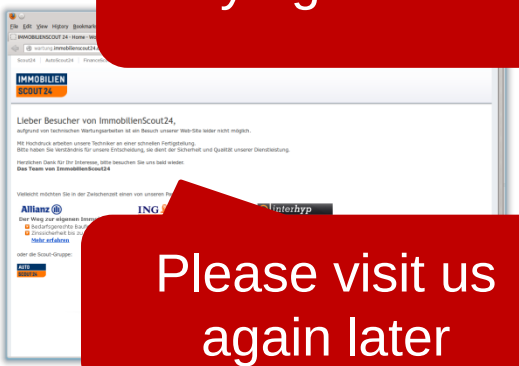
Try again later



Temporarily unavailable

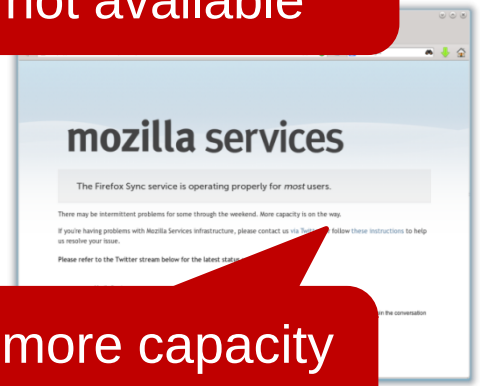


Temporarily not available

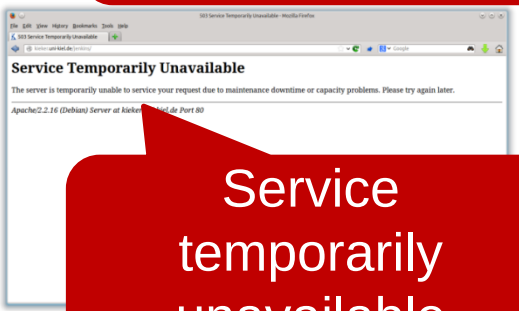


Please visit us again later

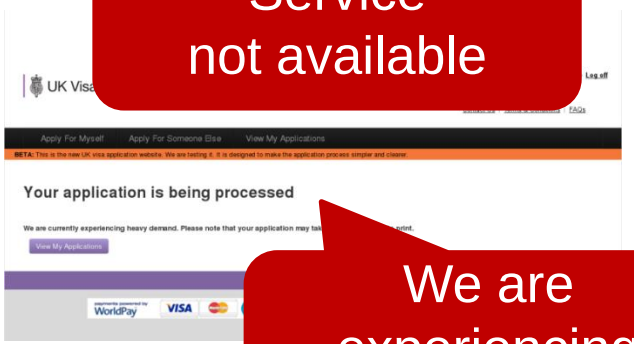
Service not available



… more capacity is on the way

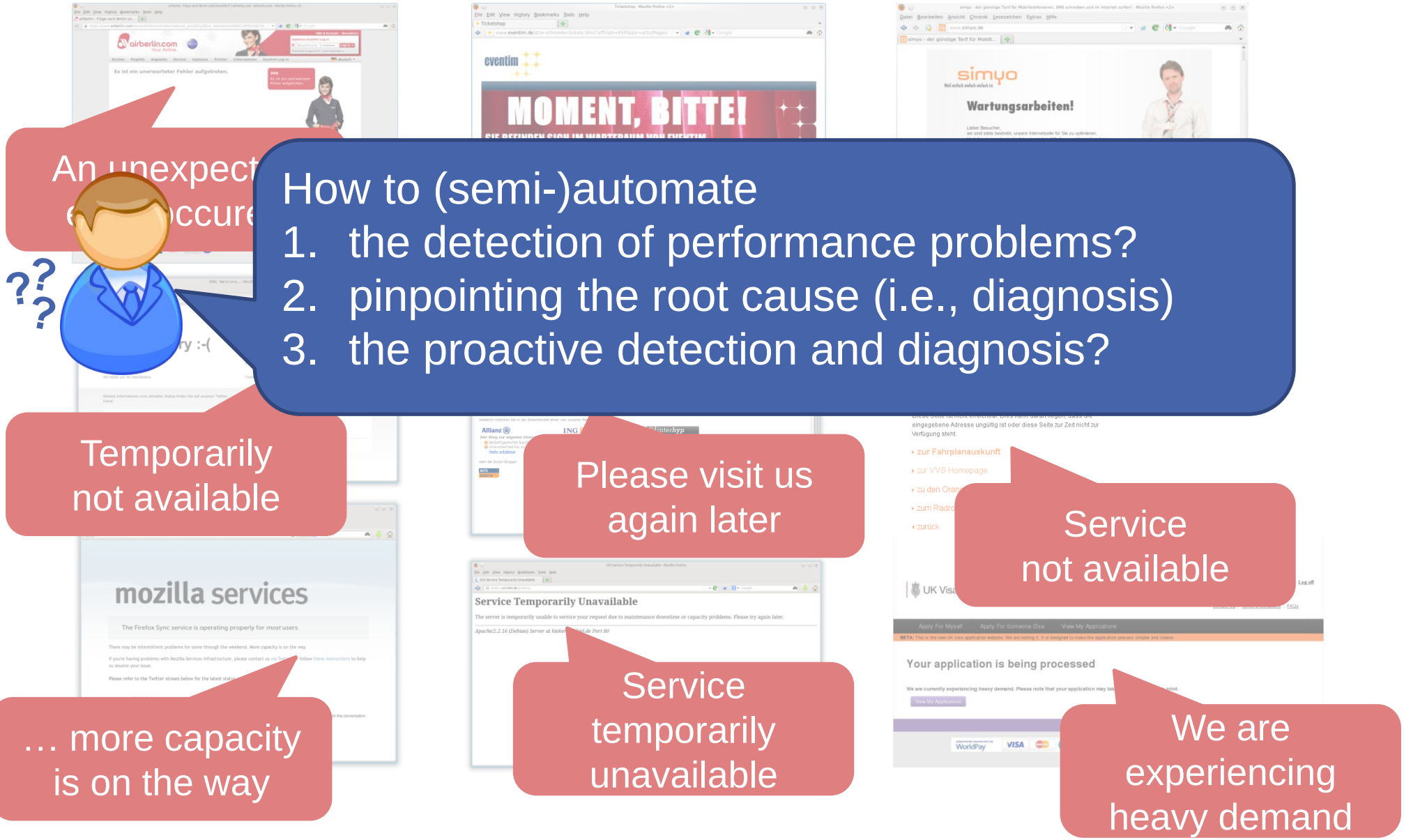


Service temporarily unavailable



We are experiencing heavy demand

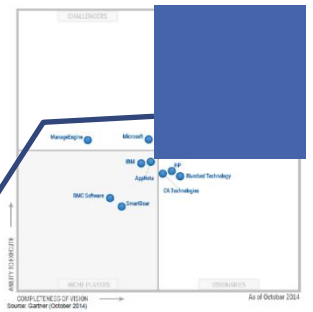
Performance (QoS) Problem Detection and Diagnosis



Application Performance Management

- APM dimensions according to Gartner (2014)
  1. End-user experience monitoring
  2. Application topology discovery and visualization
  3. User-defined transaction profiling
  4. Application component deep-dive
  5. IT operations analytics based on, e.g.,
     - Complex operations event processing
     - Statistical pattern discovery and recognition
     - Unstructured text indexing, search and inference
     - Multidimensional database search and analysis

https://www.gartner.com/doc/288942



- APM tools (selection)
  - Commercial:

    APPDYNAMICS dynatrace New Relic.
  - Free/open-source:

    inspectIT ...because performance matters! Kieker

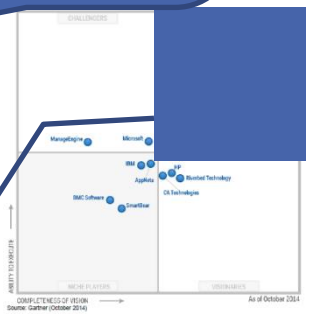
Application Performance Management

- APM dimensi
  1. End-user e
  2. Application
  3. User-define
  4. Application
  5. IT operations analytics based on, e.g.,
     - Complex operations event processing
     - Statistical pattern discovery and recognition
     - Unstructured text indexing, search and inference
     - Multidimensional database search and analysis
- APM tools (selection)
  - Commercial:

    APPDYNAMICS dynatrace New Relic.
  - Free/open-source:

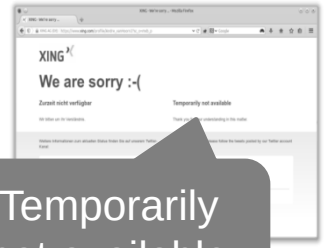
    inspectIT ...because performance matters! Kieker

Example ITOA activity:
Performance problem detection and diagnosis
- Reactive vs. proactive
- Manuel vs. automatic (incl. recommendations)
- State-based vs. transaction-based



Agenda

1. Introduction – Performance Problems
2. Kieker – Open Source APM Framework
3. Performance Problem Detection and Diagnosis with Kieker
4. Conclusions and Future Work

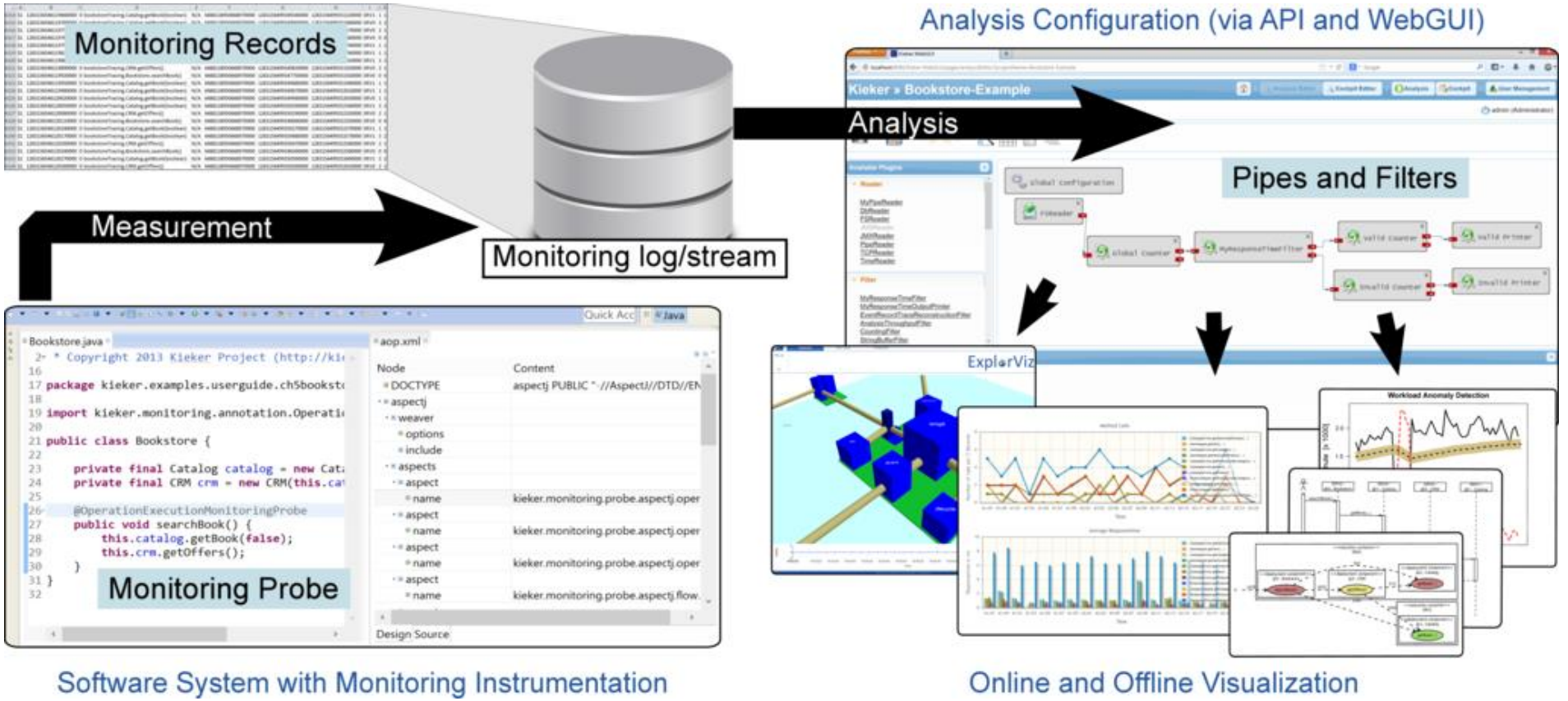


Kieker

diagnoseIT

spec RG DevOps Performance

Open Source APM Framework

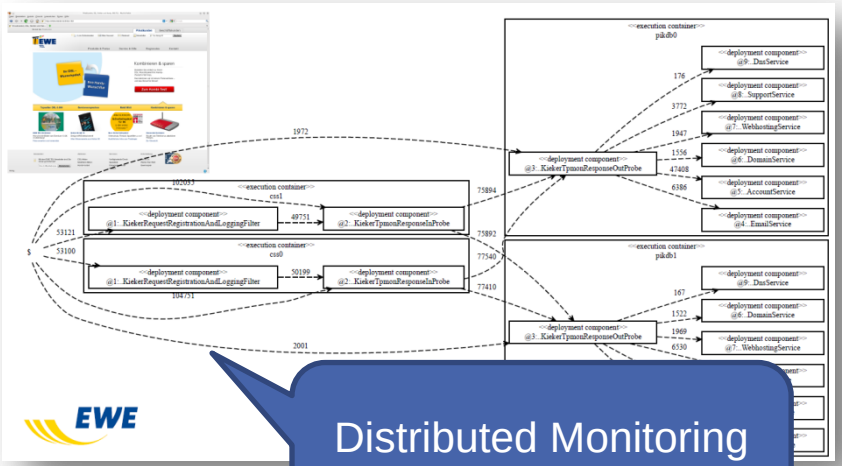


Download : http://kieker-monitoring.net/

Kieker is distributed as part of SPEC® RG's repository of peer-reviewed tools for quantitative system evaluation and analysis

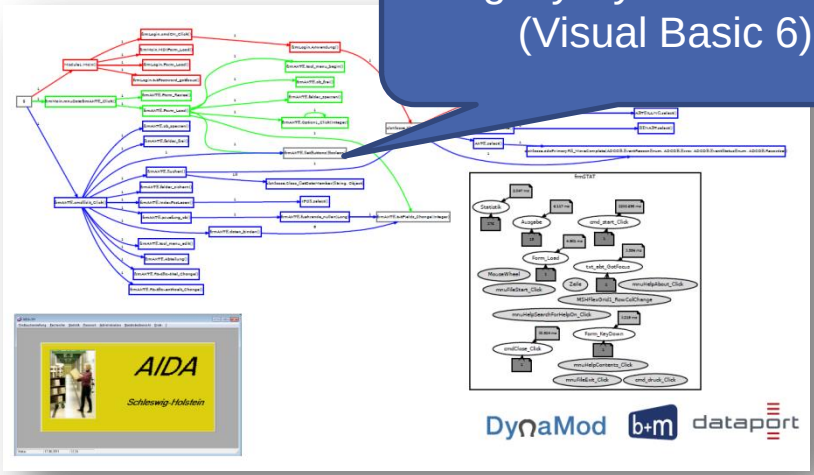
http://research.spec.org/projects/tools.html

Model Extraction and Visualization
(Selected Examples)

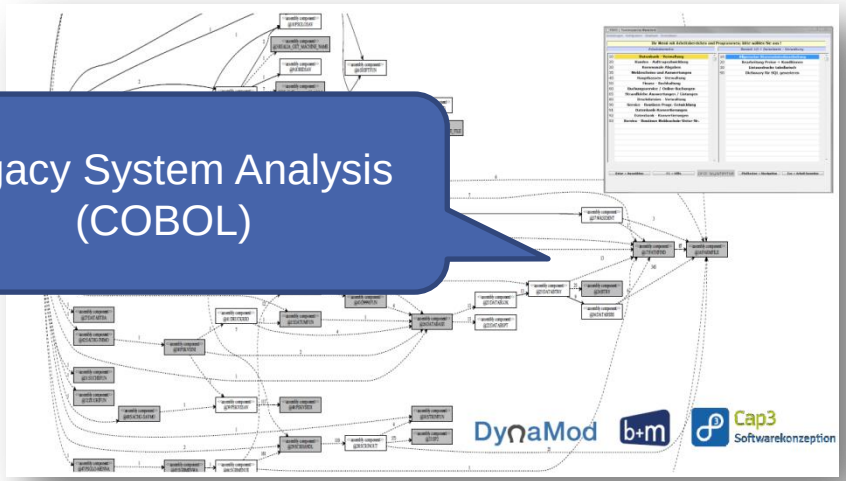


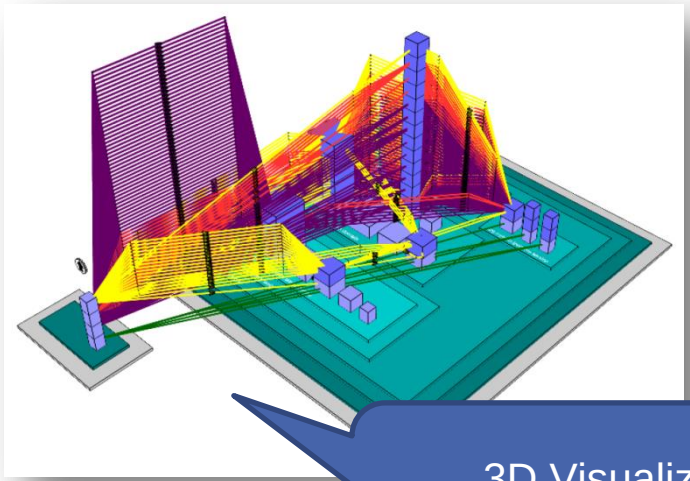
Distributed Monitoring
(Java EE/SOAP)

Legacy System Analysis
(Visual Basic 6)



Legacy System Analysis
(COBOL)





3D Visualization
of Concurrency

Visit http://kieker-monitoring.net/

KIEKER FRAMEWORK USE CASES RESEARCH AND CONSULTING

Projects, publications, talks, tutorials

About Kieker

analyzing a software system's runtime behavior — enabling Application Performance Monitoring and Architecture Discovery.

Download

User Guide

Issue Tracking

Search

QUICKLINKS

- Download
  - Release Notes
  - License
- Live Demo
- Features
- Documentation
  - Quickstart Guide
  - API Documentation
- Support
  - Contact
  - Trac / Wiki
- News & Blog

More than 60 Participants Attended SOSP '14 in Stuttgart

Posted on 14.12.2014 by André van Hoorn

References: Internal/External Researchers, Industry

Internal Researchers

Kieker is currently maintained jointly by the following research groups from Kiel University and the University of Stuttgart as part of their teaching and research activities, including collaborators from other academic or industrial institutions.

Kiel University, Kiel, Germany – Researchers from the Kiel University's Software Engineering Group investigate innovative techniques and methods for engineering, evolving, and operating continuously running software systems (research projects).

University of Stuttgart, Stuttgart, Germany – Researchers from the University of Stuttgart's Reliable Software Systems Group investigate innovative quantitative QoS analysis and forecasting methods for distributed software-intensive systems (research projects).

Feel free to contact us if you are interested in any aspect of the Kieker framework.

External Researchers

Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany – Researchers from the KIT's Software Design and Quality Group are using Kieker for different purposes, e.g., for detecting and diagnosing performance problems in systematic experiments. We are also collaborating with KIT researchers in the context of the iObserve research project.

RWTH Aachen University, Aachen, Germany – Researchers from the RTWH Aachen University's Software Construction Group are using Kieker for monitoring-based architecture reconstruction in their ARAMIS project on model-based software architecture evolution and analysis.

University of Novi Sad, Novi Sad, Serbia – Researchers from the University of Novi Sad were using Kieker for adaptive monitoring of software systems in the context of performance problem detection and diagnosis.

Warsaw University, Warsaw, Poland – Researchers from the Warsaw University employed Kieker for dynamic data acquisition of software architectures.

Xi'an Jiatong University, Xi'an, Shaanxi, China – Researchers from the Xi'an Jiatong University used Kieker for discovering architectural structures in software systems and to analyze software call graphs.

Industry

b+m Informatik AG, Melsdorf, Germany – With b+m, we collaborated in the context of the DynaMod and MENGES research projects. Moreover, Kieker is being used by b+m, e.g., for architecture discovery of large-scale COBOL mainframe systems. Contributions by b+m are part of the Kieker release.

CEWE COLOR AG & Co. OHG, Oldenburg, Germany – With CEWE COLOR, we collaborated in the context of the TrustSoft research project. CEWE COLOR provided a JavaEE-based web portal as a case study system for application performance monitoring. Contributions by CEWE COLOR are part of the Kieker release.

Dataport AöR, Altenholz, Germany – With Dataport, we collaborated in the context of the DynaMod research project. Dataport provided a VB6-based case study system for architecture discovery based on hybrid analysis with Kieker.

EPrints Services, Southampton, United Kingdom – With EPrints Services, we collaborated in the context of several thesis projects. We employ the EPrints system as a case study system for software performance analysis with Kieker for Perl-based systems. The Eprints team provides an integration of Kieker with EPrints as epkieker.

EWE TEL GmbH, Oldenburg, Germany – With EWE TEL, we collaborated in the context of the TrustSoft research project. EWE TEL provided a JavaEE-based web portal as a case study system for application performance monitoring. Contributions by EWE TEL are part of the Kieker release.

HSH Nordbank AG, Kiel, Germany – With HSH Nordbank, we collaborated in the context of the DynaMod research project. The HSH provided a C#-based function library for architecture discovery based on hybrid analysis with Kieker.

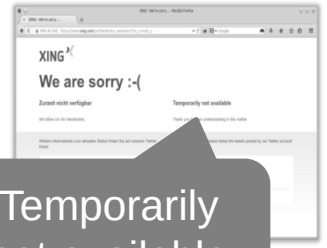
NovaTec GmbH, Leinfelden-Echterdingen, Germany – With NovaTec, we currently collaborate in the context of different teaching projects on application performance management. NovaTec published a nice blog article about their first experiences with Kieker.

SAP Research, Karlsruhe, Germany – Kieker is used as a tool to collect performance data for the Software Performance Cockpit. We are also collaborating with SAP Research in the context of the iObserve research project.

XING AG, Hamburg, Germany – With XING, we collaborated in the context of a Diploma thesis on online performance anomaly detection (OPAD). XING provided its core system xing.com for evaluating the Kieker-based OPAD implementation.

Agenda

1. • Introduction – Performance Problems
2. • Kieker – Open Source APM Framework
3. • Performance Problem Detection and Diagnosis (PPD&D) with Kieker
4. • Conclusions and Future Work

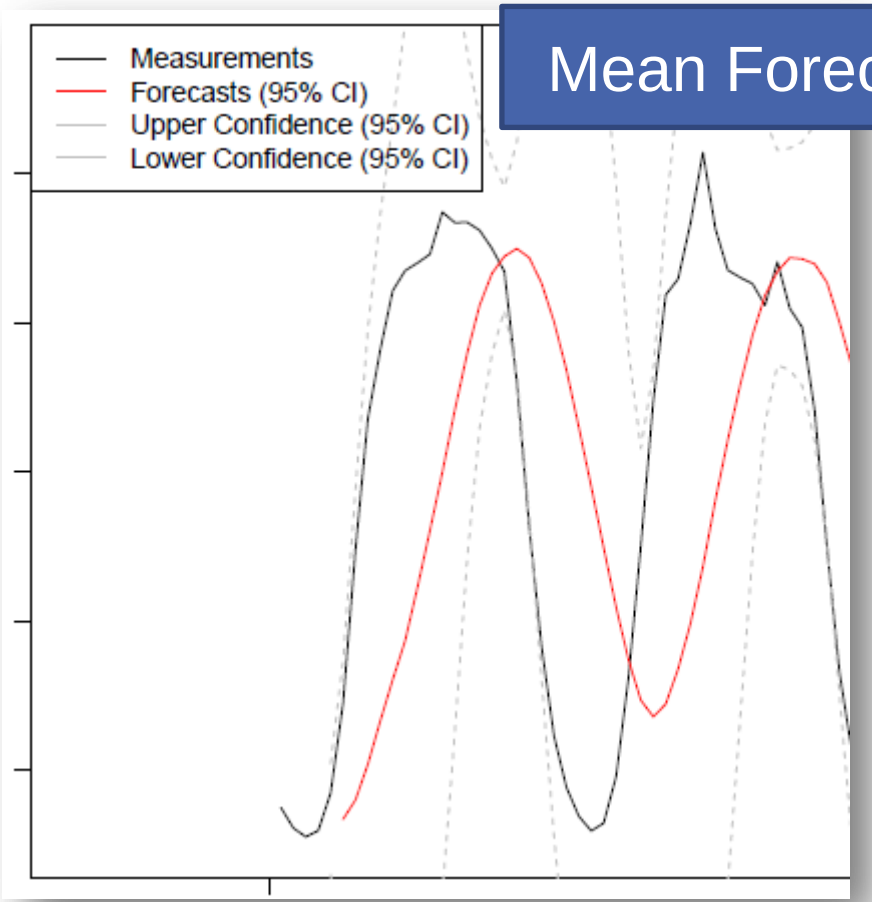


Kieker

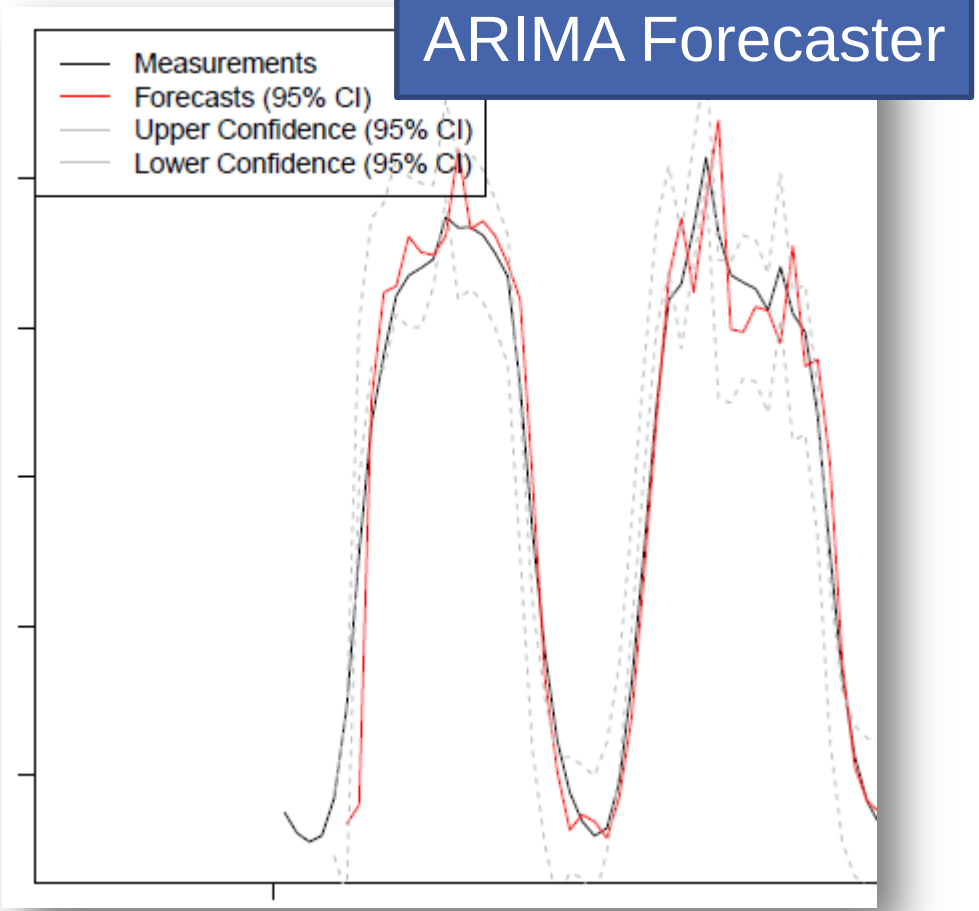
diagnoseIT

spec RG DevOps Performance

ΘPAD – Time Series Analysis Introduction



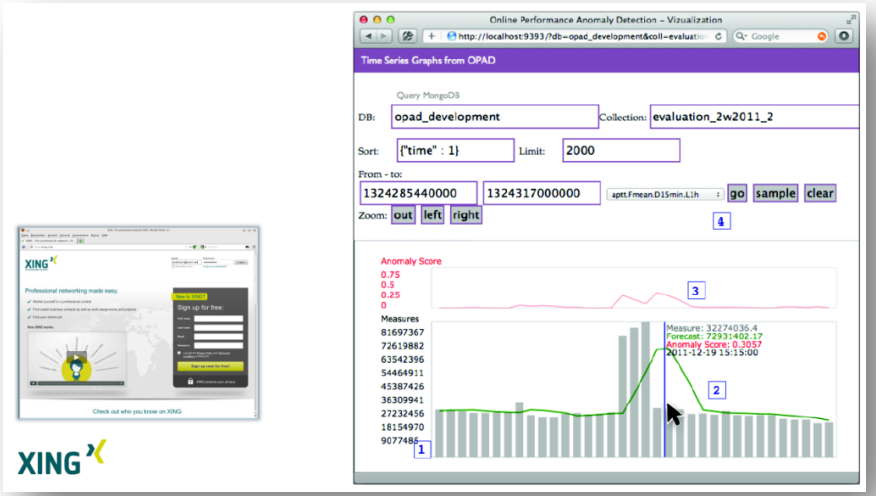
Mean Forecaster



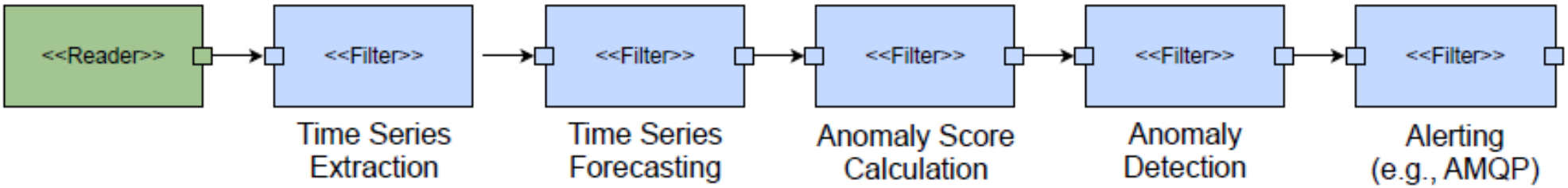
ARIMA Forecaster

Forecasts for Wikipedia data (Kieker ΘPAD example)

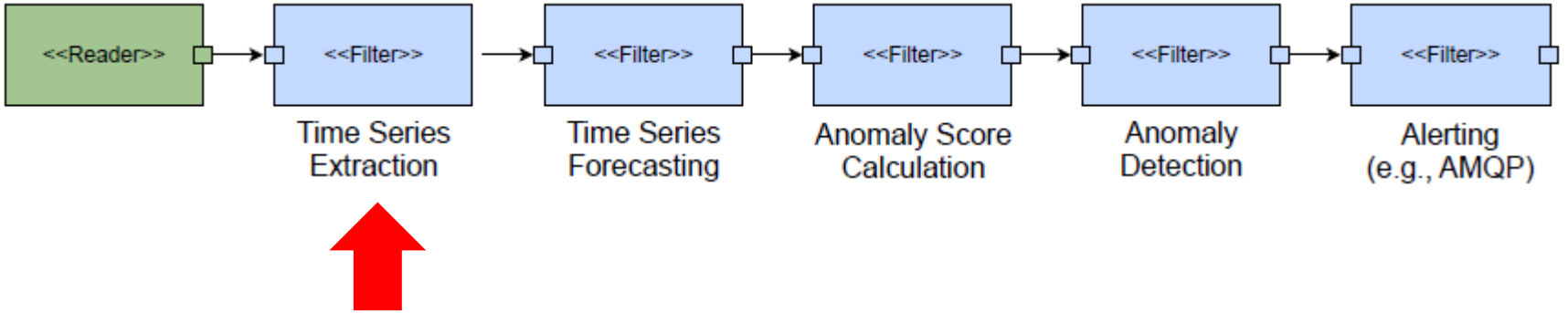
ΘPAD: Online Performance Anomaly Detection

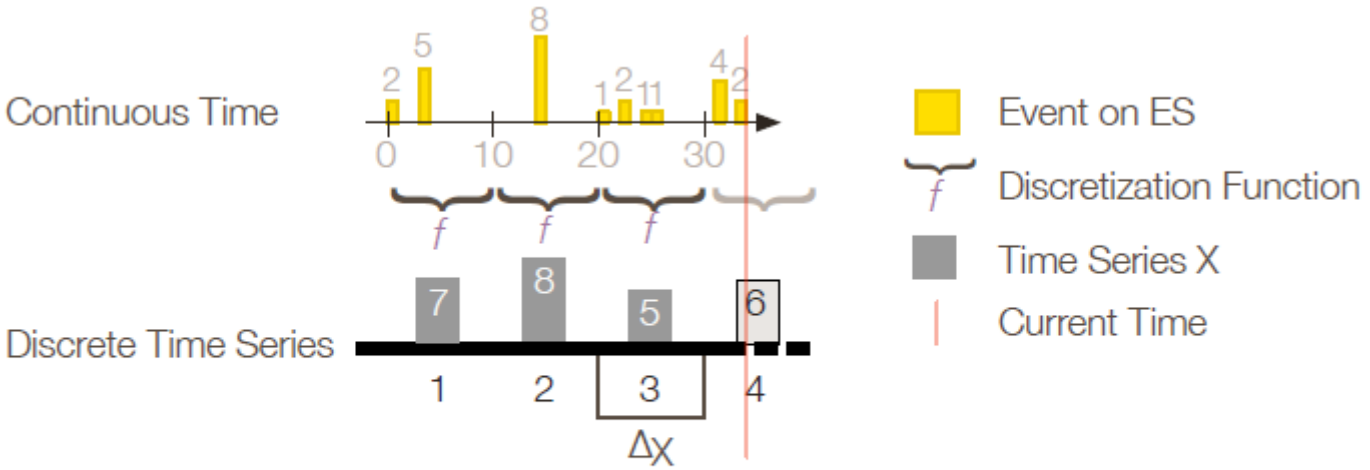


(Bielefeld, 2012), (Frotscher, 2013)



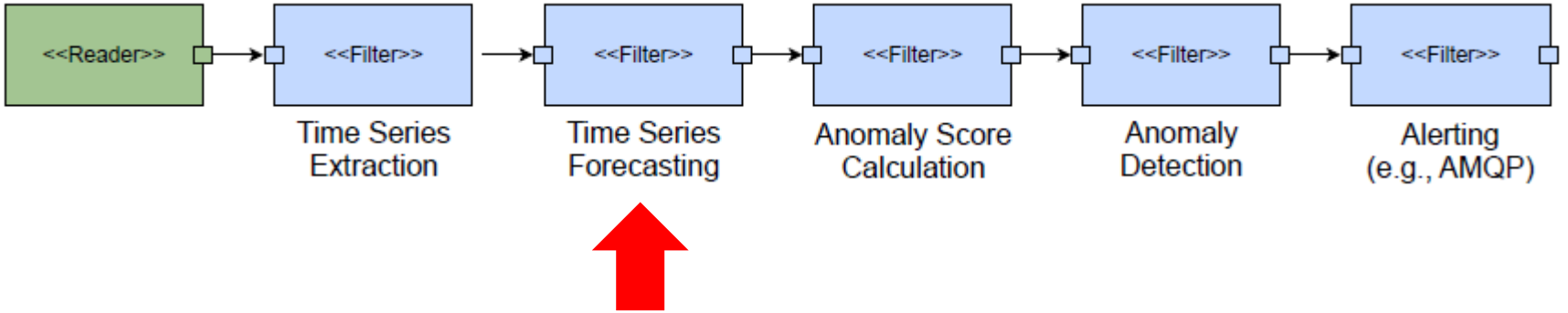
ΘPAD (cont'd) – Time Series Extraction

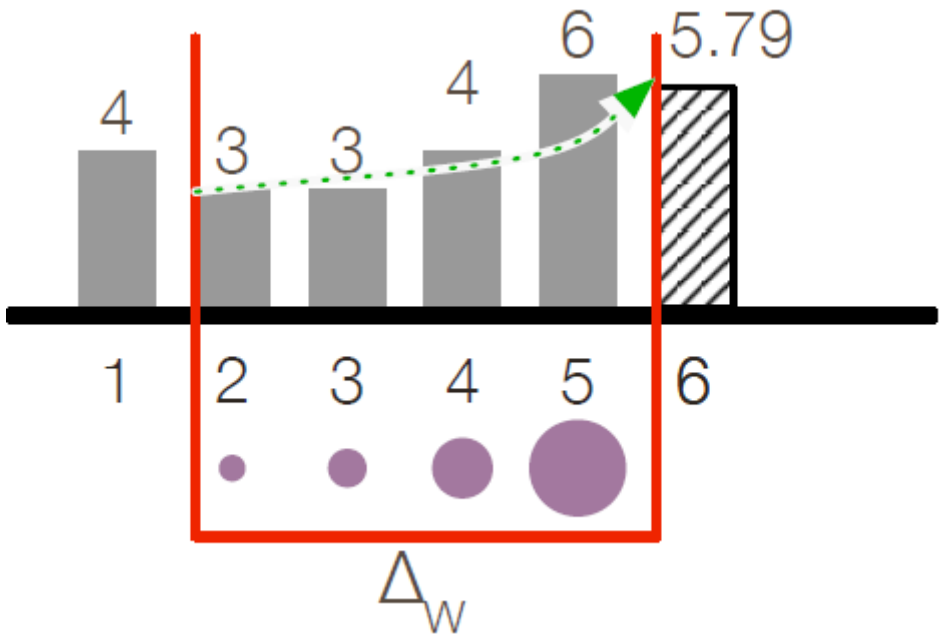




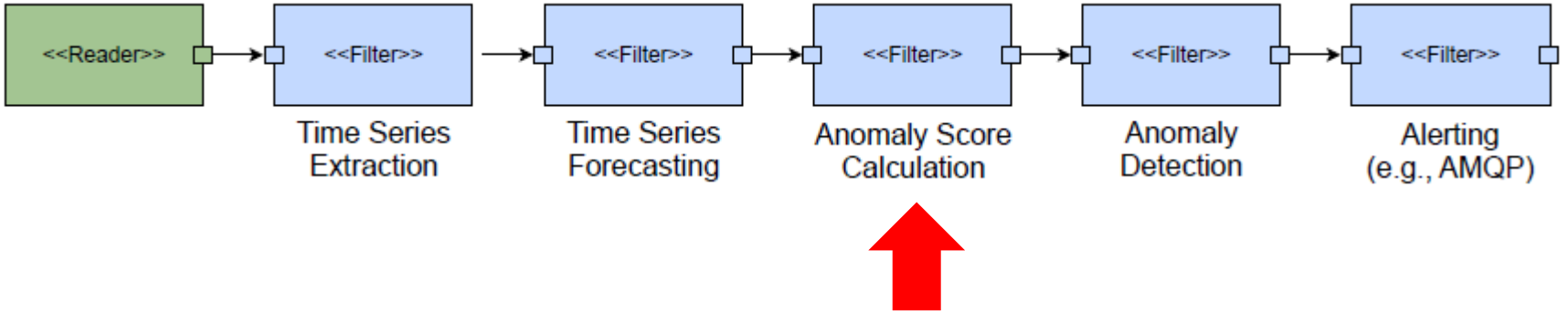

```
select sum(value) as aggregation
from MeasureEvent.win:time_batch( 1000 msec )
```

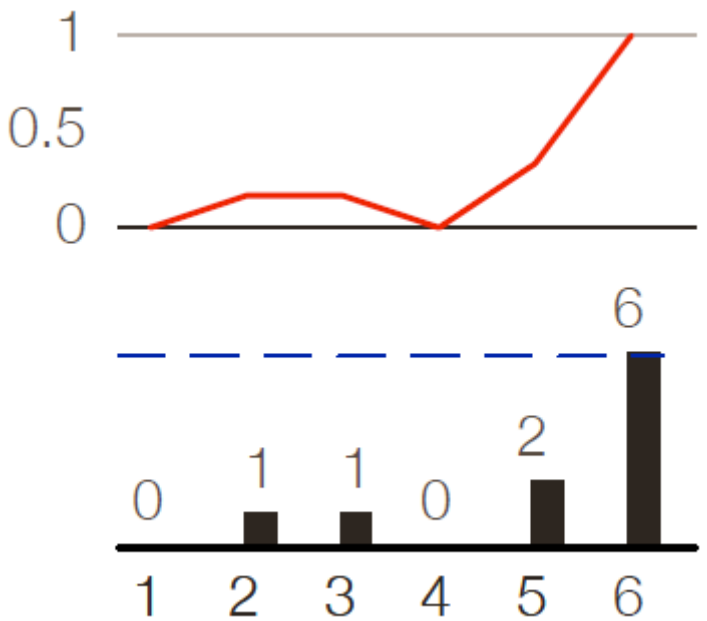
ΘPAD (cont'd) – Time Series Forecasting



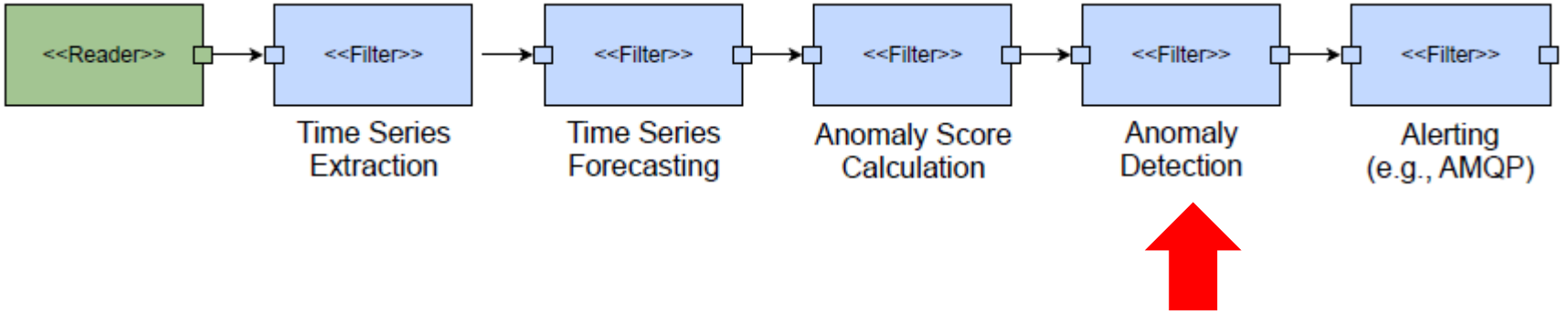


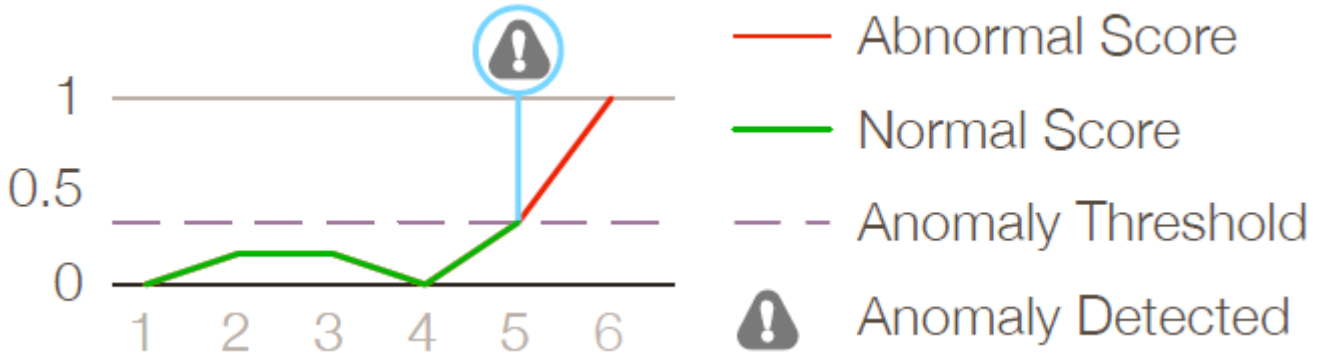
ΘPAD (cont‘d) – Anomaly Score Calculation



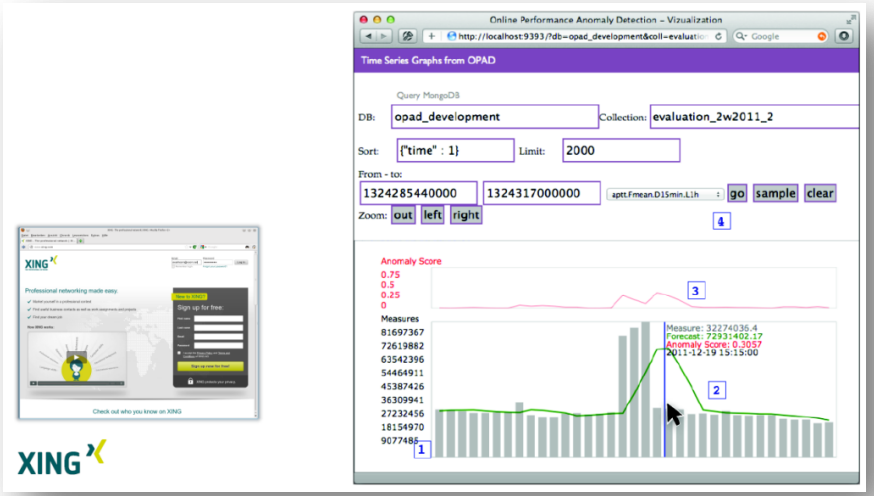


ΘPAD (cont‘d) – Anomaly Detection



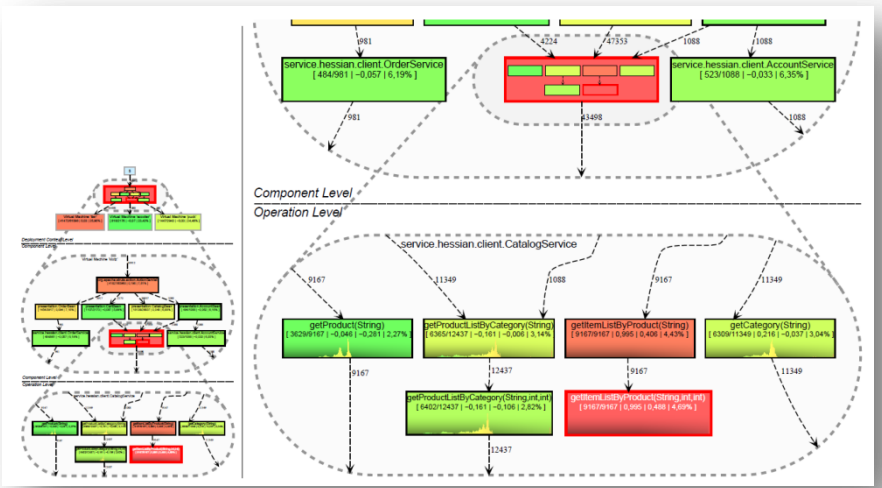


Kieker-based PPD&D Approaches



(Bielefeld, 2012), (Frotscher, 2013)

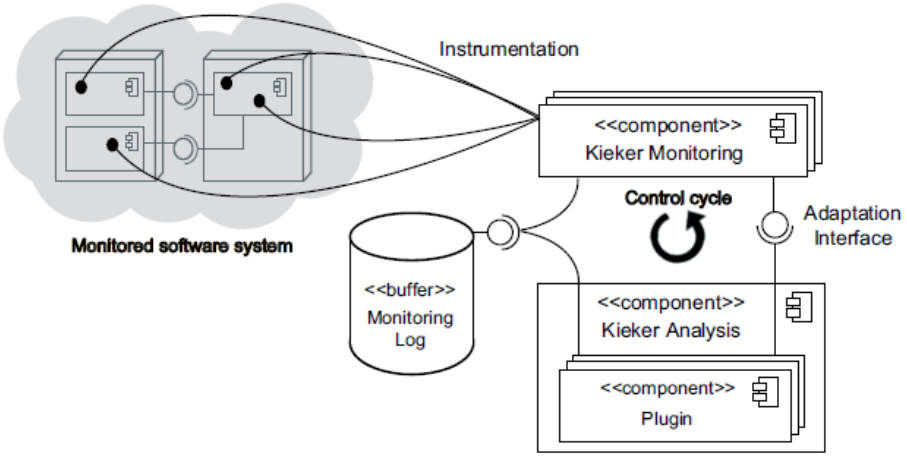
- Based on time series analysis (various algorithms)
- ΘPAD part of Kieker release
- Limited to problem detection
- No architecture consideration
- Case study at XING



(Marwede et al., 2009)

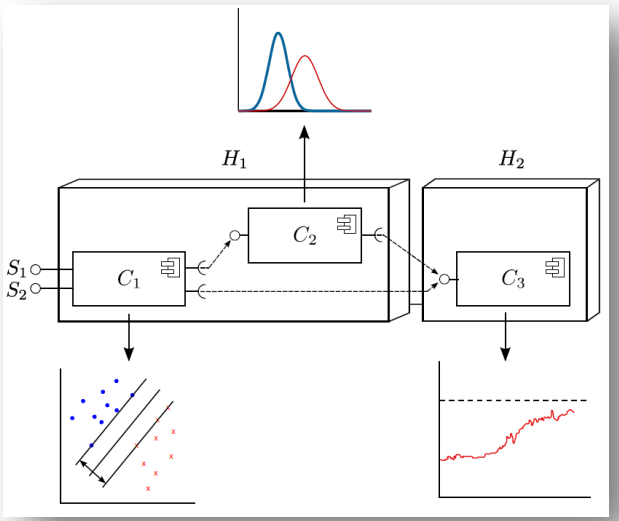
- Incorporates architectural knowledge (e.g., deployment, calling dependencies)
- Focusing on offline analysis
- Cf. Rohr (2015)

Kieker-based PPD&D Approaches



(Ehlers et al., 2011, 2012)

- Adaptive monitoring
- OCL-based decisions
- Cf. Okanovic et al. (2013)
- No dynamic (bytecode) instrumentation

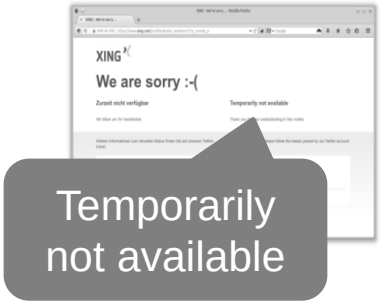


(Pitakrat et al., 2013, 2014)

- Proactive, hierarchical
- Inclusion of different statistical techniques (e.g., time series analysis, machine learning)
- Combination of multiple data sources (e.g., HDD SMART, log files) and architectural knowledge

Agenda

1. Introduction – Performance Problems
2. Kieker – Open Source APM Framework
3. Performance Problem Detection and Diagnosis with Kieker
4. Conclusions and Future Work



Summary

- APM is an increasingly important topic
- PPD&D is one APM activity

- Mature APM tools exist
  - Commercial tools
    - Support monitoring for various platforms and technologies
    - Only basic support for problem detection and diagnosis
  - Kieker presented as an extensible open-source example
    - Kieker-based approaches for problem detection and diagnosis

Challenges for APM and PPD&D (Selection)

1. Laborious and Continuous APM Configuration

- APM configuration time consuming and error prone
- Problem intensified due to faster development cycles

2. Problem Detection and Diagnosis

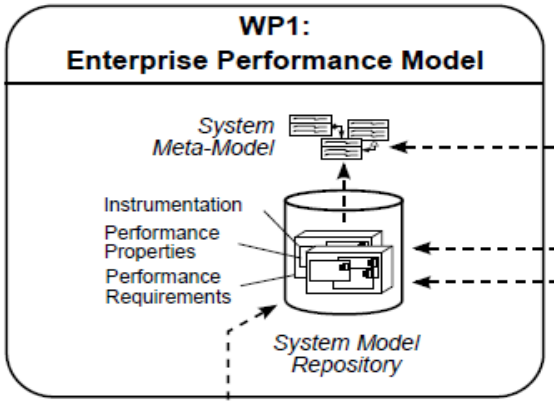
- Alerting thresholds hard to determine and to maintain
- Manual diagnosis requires extensive expert knowledge
- APM experts are scarce goods

3. Detachedness of APM processes

- APM processes often system-specific
- Few possibilities to deposit/reuse expert knowledge

Future Work: diagnose it

Expert-Guided Automatic Diagnosis of Performance Problems in Enterprise Applications



NOVATEC

Universität Stuttgart

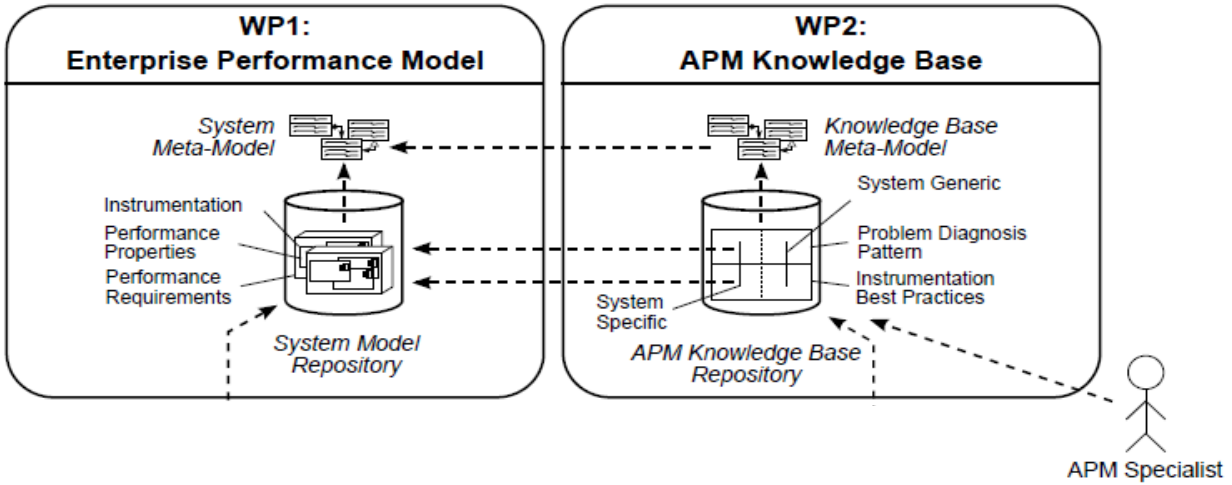
COMMERZBANK

ivv

SPONSORED BY THE
Federal Ministry
of Education
and Research

Future Work:

Expert-Guided Automatic Diagnosis of Performance Problems in Enterprise Applications



NOVATEC

Universität Stuttgart

COMMERZBANK

ivv

SPONSORED BY THE
Federal Ministry
of Education
and Research

Future Work:

Expert-Guided Automatic Diagnosis of Performance Problems in Enterprise Applications

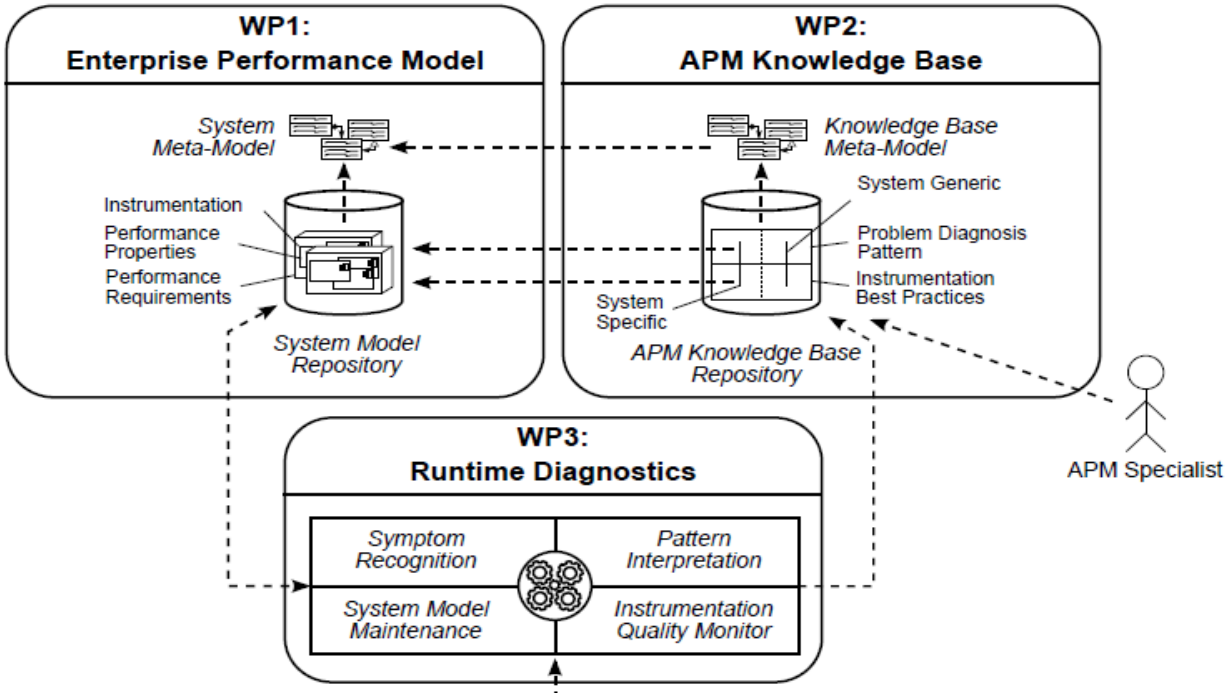
NOVATEC

Universität Stuttgart

COMMERZBANK

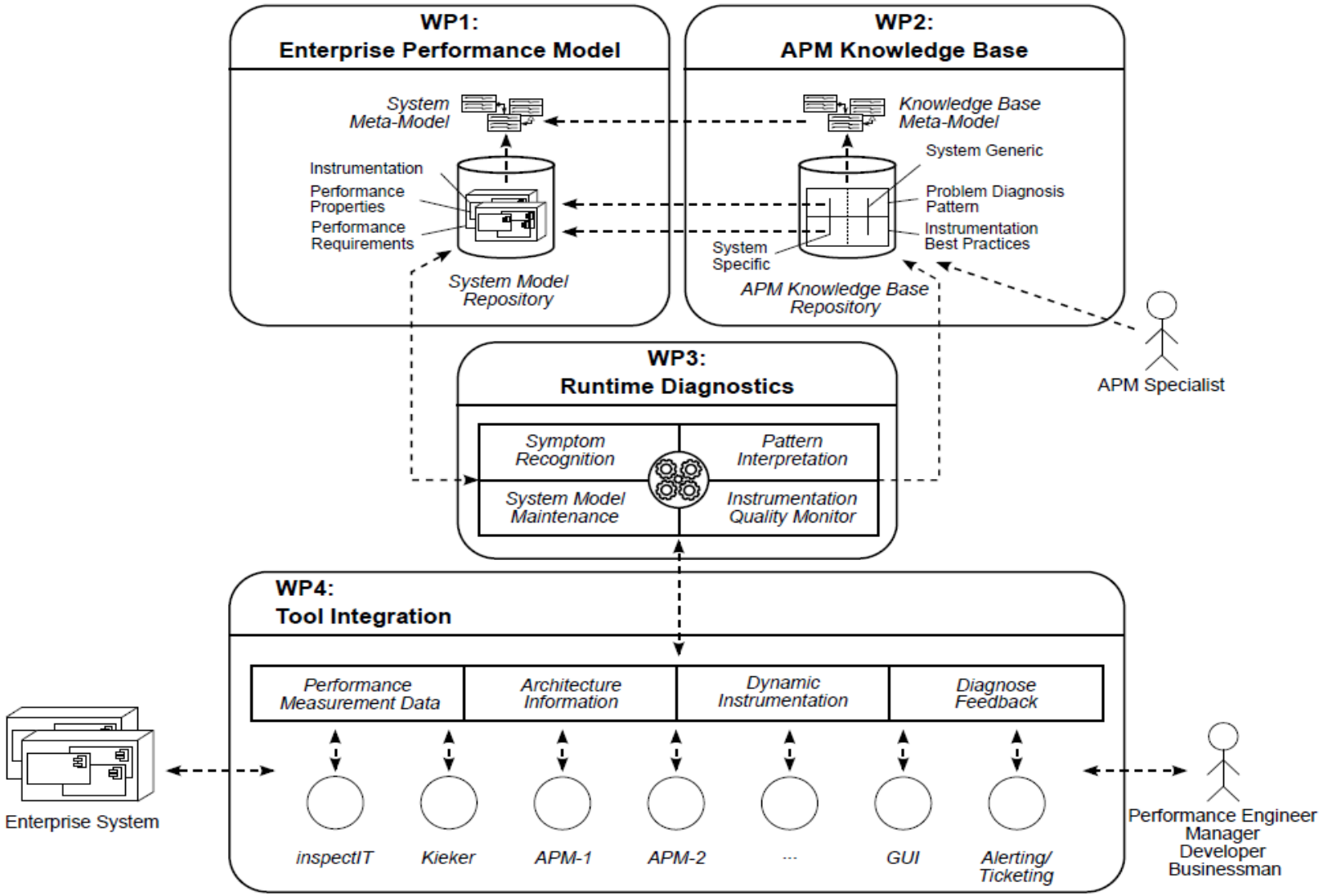
ivv

SPONSORED BY THE
Federal Ministry of Education and Research



Future Work:

Expert-Guided Automatic Diagnosis of Performance Problems in Enterprise Applications



SPONSORED BY THE
Federal Ministry of Education and Research

The End – My APM Wish List

- **Transparency, Openness, Technology Transfer**
  - e.g., sharing of best practices (libraries, white papers) for framework-specific instrumentation, problem detection and diagnosis
- **Interoperability**
  - e.g., common formats for instrumentation description, configuration, measurement data (traces)
- **Reproducibility, Comparibility**
  - e.g., sample/benchmark applications and datasets, case studies

➢ SPEC RG DevOps Performance Working Group
http://research.spec.org/devopswg/

➢ Kieker http://kieker-monitoring.net

References

- ***(Döhring, 2012)*** P. Döhring. Visualisierung von Synchronisationspunkten in Kombination mit der Statik und Dynamik eines Softwaresystems. Master's thesis, Kiel University, Oct. 2012.
- ***(Ehlers, 2012)*** J. Ehlers. Self-Adaptive Performance Monitoring for Component-Based Software Systems. PhD thesis, Department of Computer Science, Kiel University, Germany, 2012.
- ***(Ehlers et al., 2011)*** J. Ehlers, A. van Hoorn, J.Waller, and W. Hasselbring. Self-adaptive software system monitoring for performance anomaly localization. In Proceedings of the 8th ACM International Conference on Autonomic computing (ICAC'11). ACM, 2011.
- ***(Fittkau et al., 2014)*** F. Fittkau, A. van Hoorn, and W. Hasselbring. Towards a dependability control center for large software landscapes. In Proceedings of the 10th European Dependable Computing Conference (EDCC '14), IEEE, 2014.
- ***(Frotscher, 2013)*** T. Frotscher. Architecture-based multivariate anomaly detection for software systems, Master's Thesis, Kiel University, 2013.
- **(Gartner, 2014)** J. Kowall and W. Cappelli. Gartner's Magic Quadrant for Application Performance Monitoring 2014
- ***(Marwede et al., 2009)*** N. S. Marwede, M. Rohr, A. van Hoorn, and W. Hasselbring. Automatic failure diagnosis support in distributed large-scale software systems based on timing behavior anomaly correlation. In Proc. CSMR '09. IEEE, 2009.

References (cont‘d)

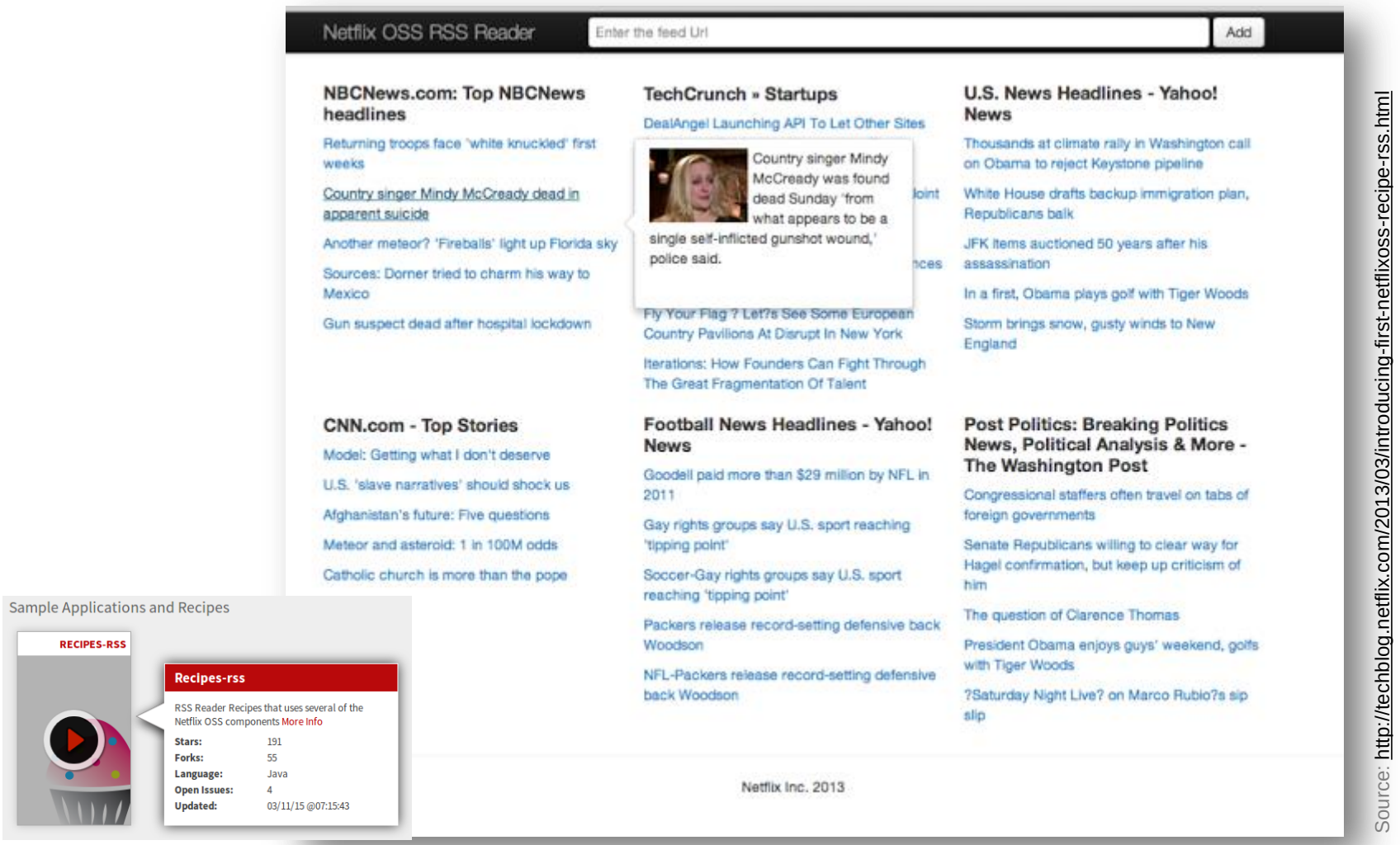
- ***(Okanovic et al., 2013)*** D. Okanovic, A. van Hoorn, Z. Konjovic, and M. Vidakovic. SLA-driven adaptive monitoring of distributed applications for performance problem localization. Computer Science and Information Systems (ComSIS), 10(10), 2013.
- ***(Pitakrat, 2013)*** T. Pitakrat. Hora: Online failure prediction framework for component-based software systems based on kieker and palladio. In Proc. SOSP 2013. CEUR-WS.org, Nov. 2013.
- ***(Pitakrat et al., 2014)*** T. Pitakrat, A. van Hoorn, and L. Grunske. Increasing dependability of component-based software systems by online failure prediction. In Proc. EDCC’14. IEEE, 2014.
- ***(Richter, 2012)*** B. Richter. Dynamische Analyse von COBOL-Systemarchitekturen zum modellbasierten Testen ("Dynamic analysis of cobol system architectures for model-based testing", in German). Diploma Thesis, Kiel University. 2012.
- ***(Rohr, 2015)*** M. Rohr. Workload-sensitive Timing Behavior Analysis for Fault Localization in Software Systems. PhD thesis, Department of Computer Science, Kiel University, Germany, 2015.
- ***(Rohr et al., 2010)*** M. Rohr, A. van Hoorn, W. Hasselbring, M. Lübcke, and S. Alekseev. Workload-intensity-sensitive timing behavior analysis for distributed multi-user software systems. In Proc. WOSP/SIPEW ’10. ACM, 2010.

References (cont‘d)

- ***(van Hoorn, 2014)*** A. van Hoorn. Model-Driven Online Capacity Management for Component-Based Software Systems. PhD thesis, Department of Computer Science, Kiel University, Germany, 2014
- ***(van Hoorn et al., 2009)*** A. van Hoorn, M. Rohr, W. Hasselbring, J. Waller, J. Ehlers, S. Frey, and D. Kieselhorst. Continuous monitoring of software services: Design and application of the Kieker framework. TR-0921, Department of Computer Science, University of Kiel, Germany, 2009.
- ***(van Hoorn et al., 2012)*** A. van Hoorn, J.Waller, and W. Hasselbring. Kieker: A framework for application performance monitoring and dynamic software analysis. In Proc. ACM/SPEC ICPE ’12. ACM, 2012.
- **(Wulf, 2012)** C. Wulf. Runtime visualization of static and dynamic architectural views of a software system to identify performance problems. B.Sc. Thesis, University of Kiel, Germany, 2010.

BONUS/BACK-UP

Netflix OSS Recipes Application (Motivating Example)

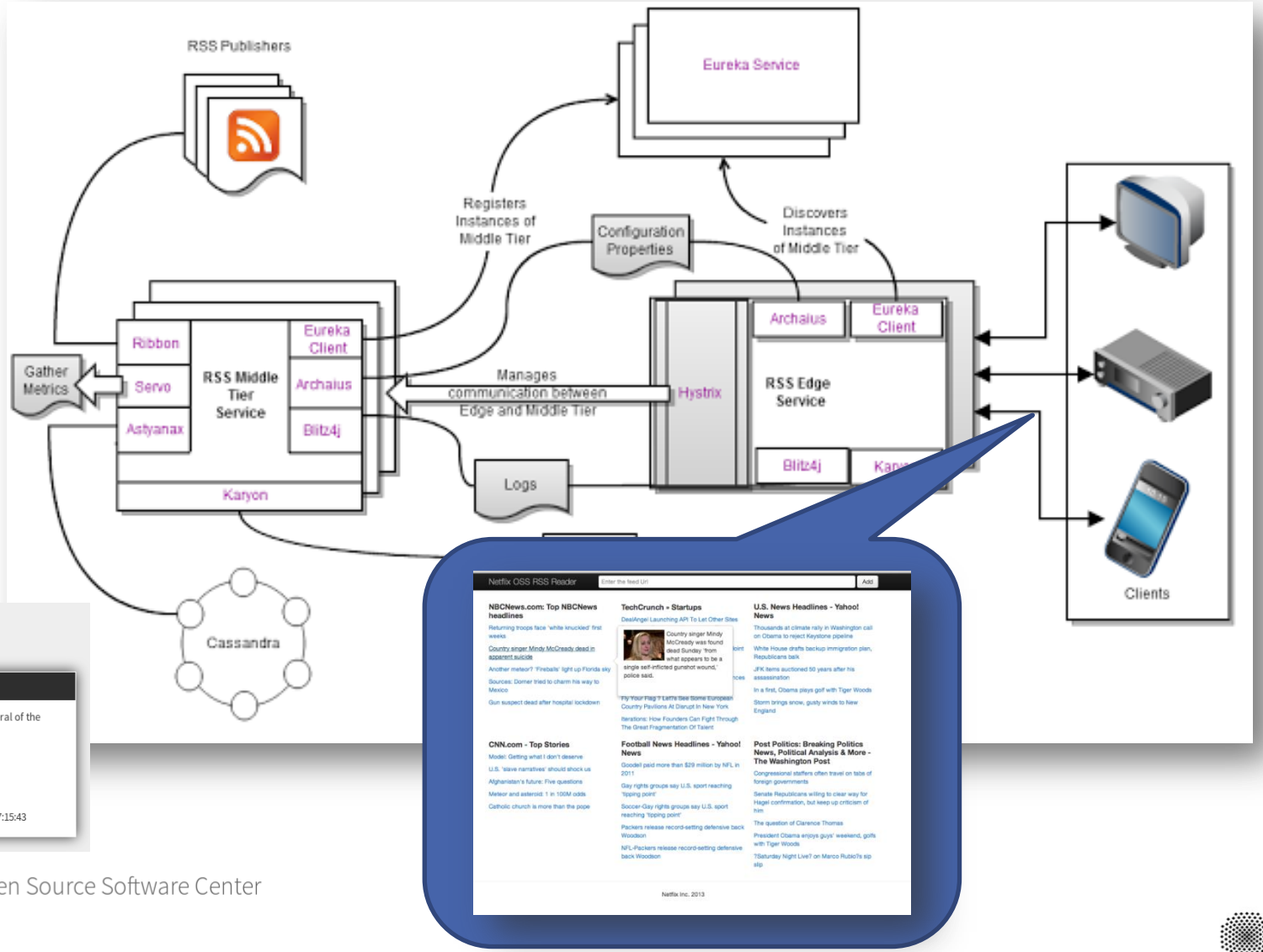


Source: http://techblog.netflix.com/2013/03/introducing-first-netflixoss-recipe-rss.html

NETFLIX | OSS Netflix Open Source Software Center

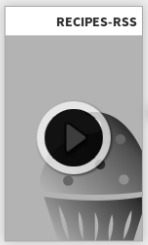
Download: https://github.com/Netflix/recipes-rss

Netflix OSS Recipes Application (cont‘d)



Source: https://github.com/Netflix/recipes-rss/wiki/Architecture

Sample Applications and Recipes



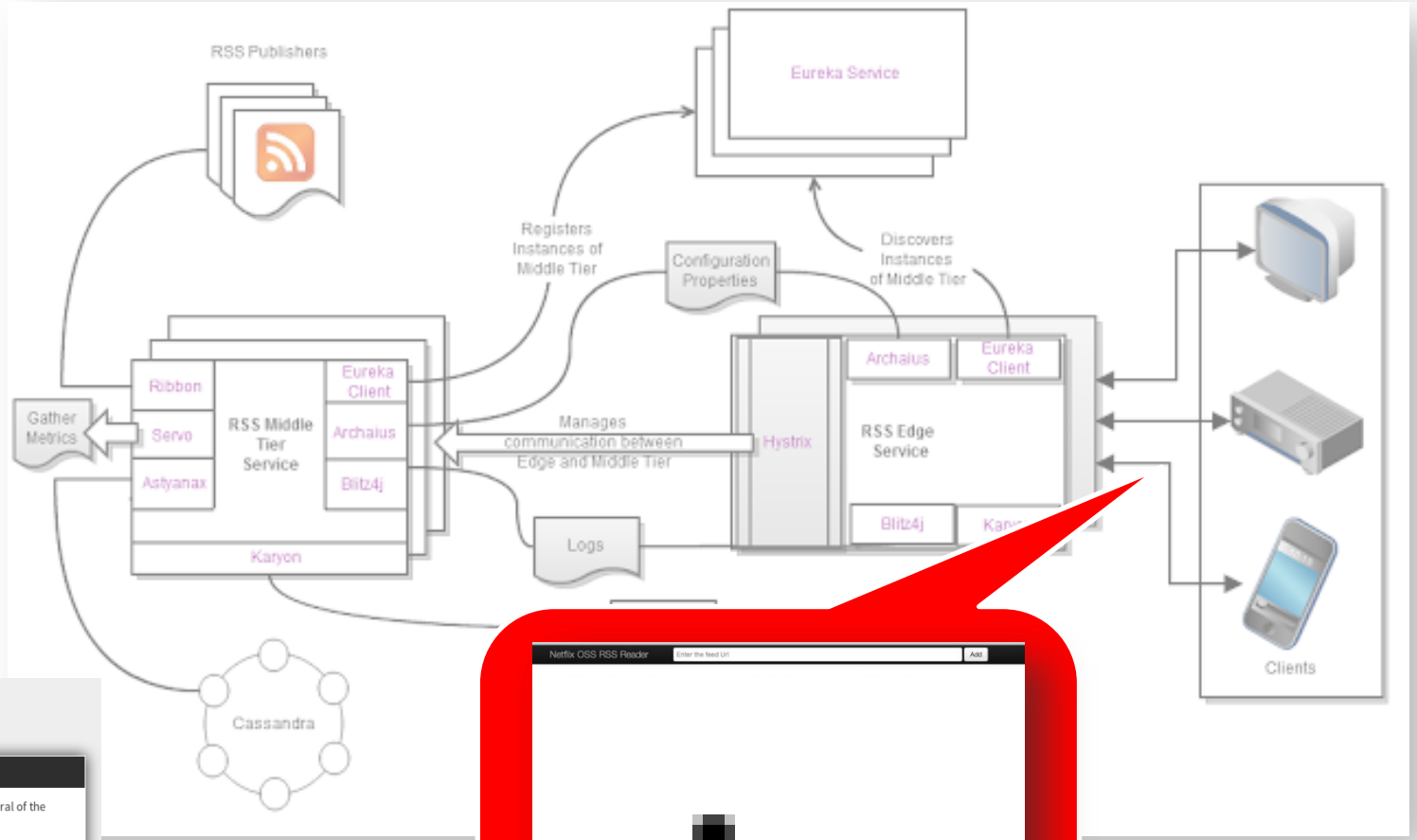
Recipes-rss

RSS Reader Recipes that uses several of the Netflix OSS components More Info

Stars:	191
Forks:	55
Language:	Java
Open Issues:	4
Updated:	03/11/15 @07:15:43

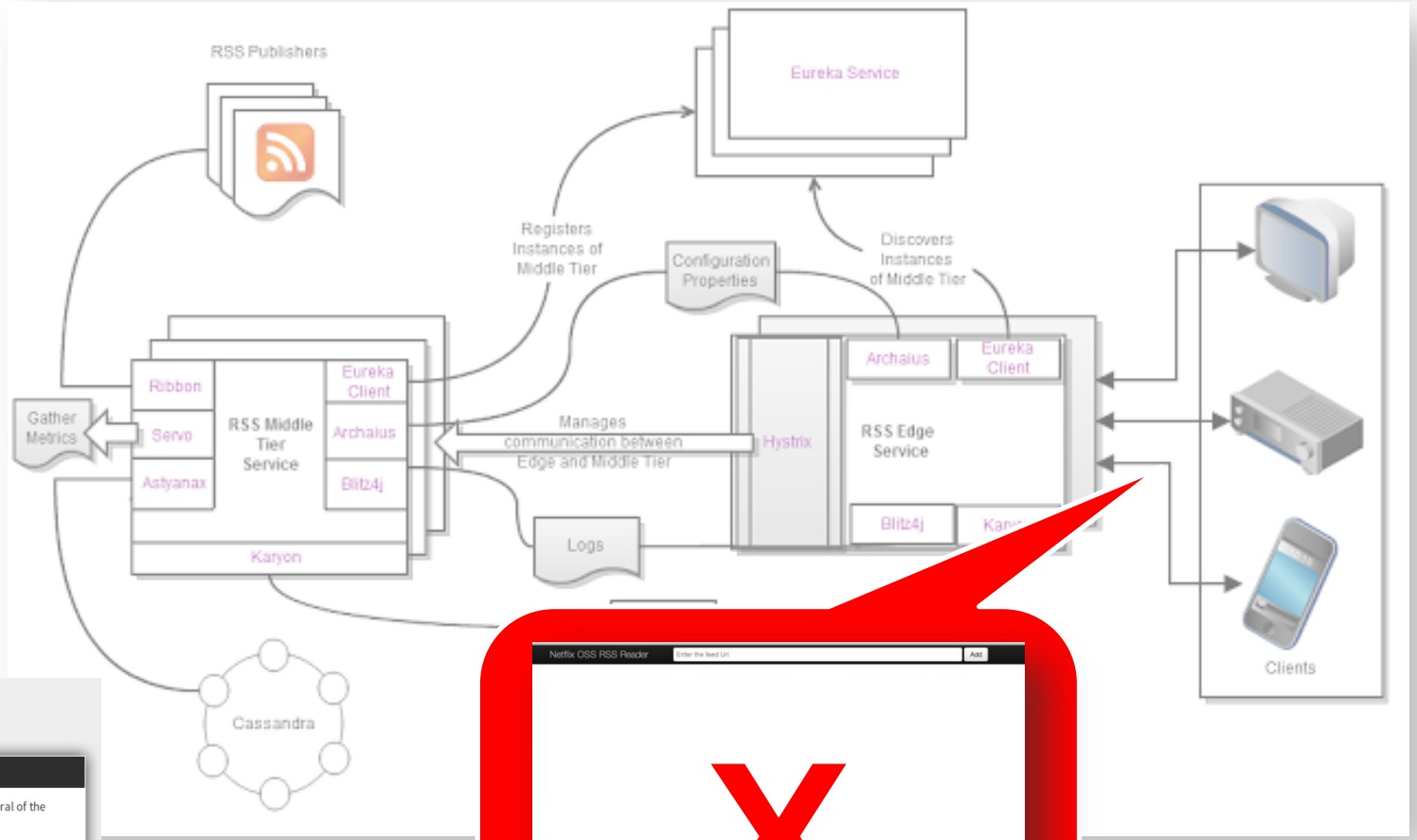
NETFLIX | OSS Netflix Open Source Software Center

Problem Symptom 1: Increased Response Times



NETFLIX | OSS Netflix Open Source Software Center

Problem Symptom 2: Service Unavailability



NETFLIX | OSS Netflix Open Source Software Center

Problem Detection and Diagnosis Approaches

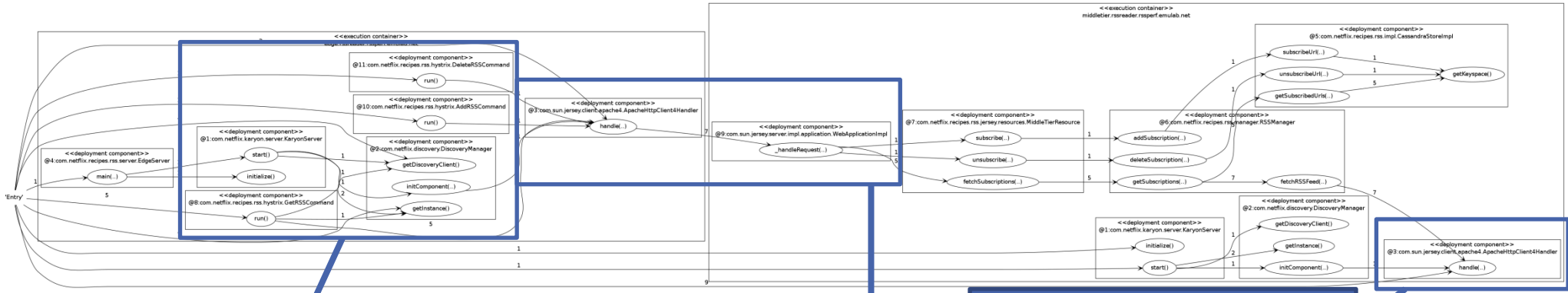
- Features
  - Reactive vs. proactive
  - Manual vs. automatic
  - State-based vs. transaction-based etc.

- Statistical techniques
  - Time series analysis
  - Anomaly detection (incl. change detection)
  - Machine learning etc.

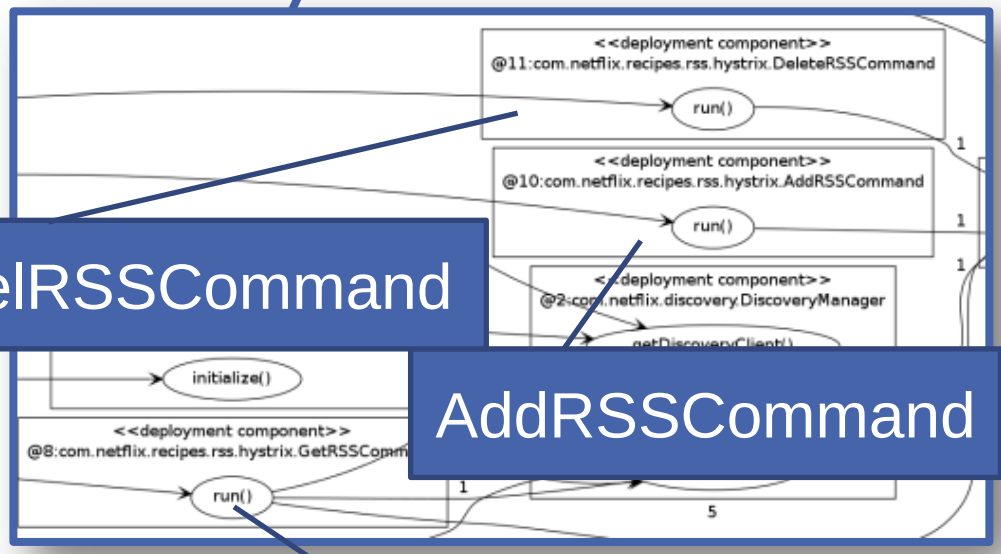
Example Kieker Plots for Netflix OSS Recipes Application

Edge

Middletier



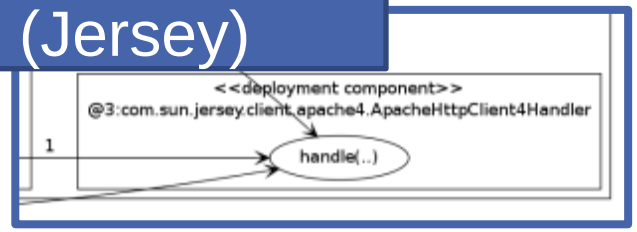
Backend call (Jersey)



DelRSSCommand

AddRSSCommand

GetRSSCommand



Outgoing/incoming REST calls (Jersey)