André van Hoorn[1] and Wilhelm Hasselbring[2]

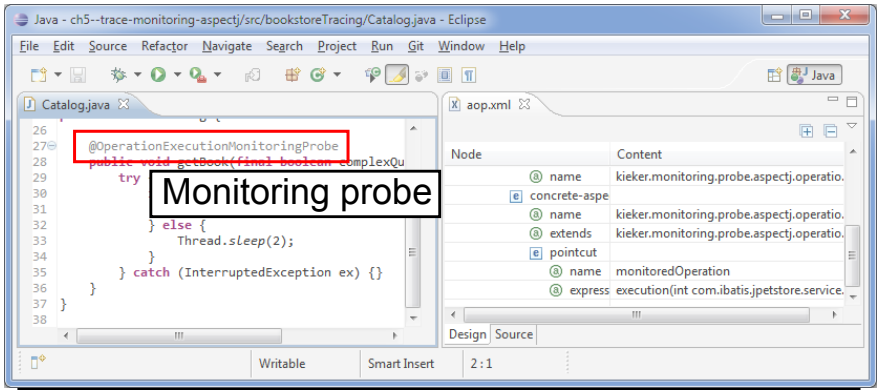
[1] *University of Stuttgart*
Institute of Software Technology
Reliable Software Systems Group

[2] *Kiel University*
Department of Computer Science
Software Engineering Group

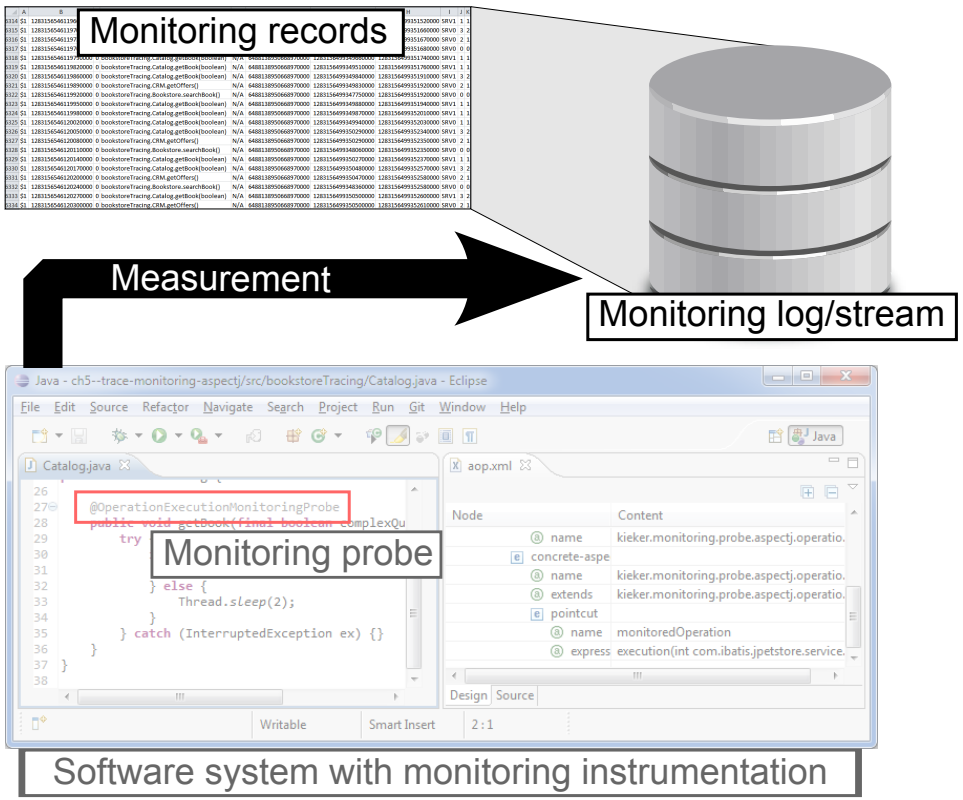
Joint Kieker/Palladio Days (KPDAYS '13)
November 27, 2013 @ Karlsruhe

Agenda

1. ~~Motivation: Performance, Monitoring, Dynamic Analysis, ...~~
2. What is Kieker?
3. What is Kieker Good For?
4. Who has been/is Working on Kieker?

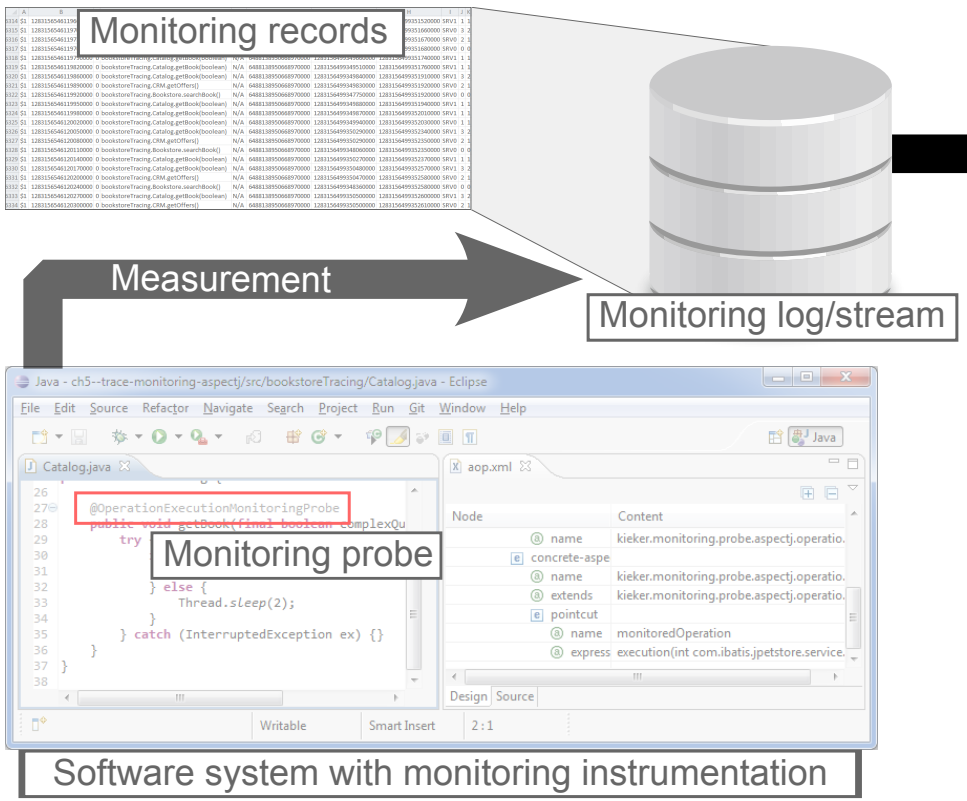


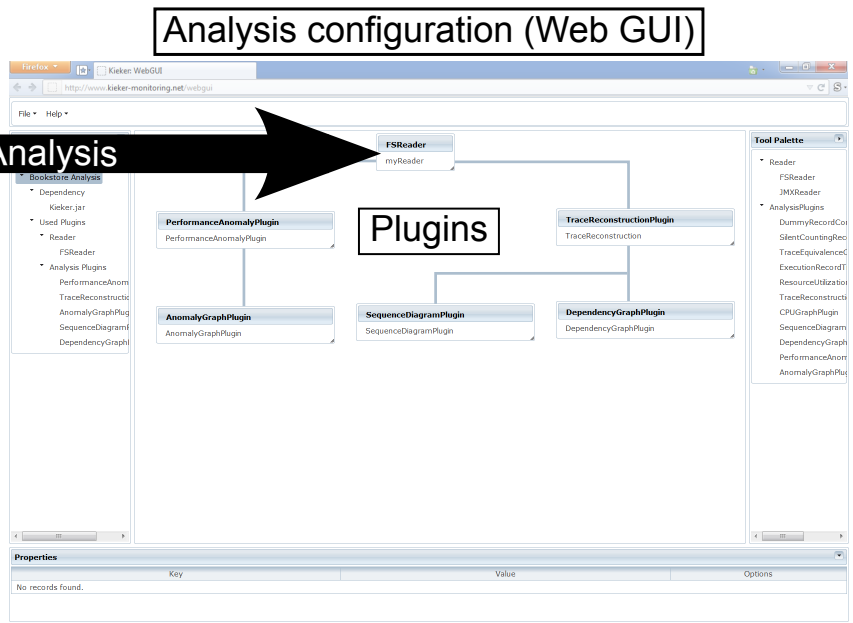
Software system with monitoring instrumentation


Monitoring records
Measurement
Monitoring log/stream
Monitoring probe
Software system with monitoring instrumentation

Dynamic Analysis with Kieker

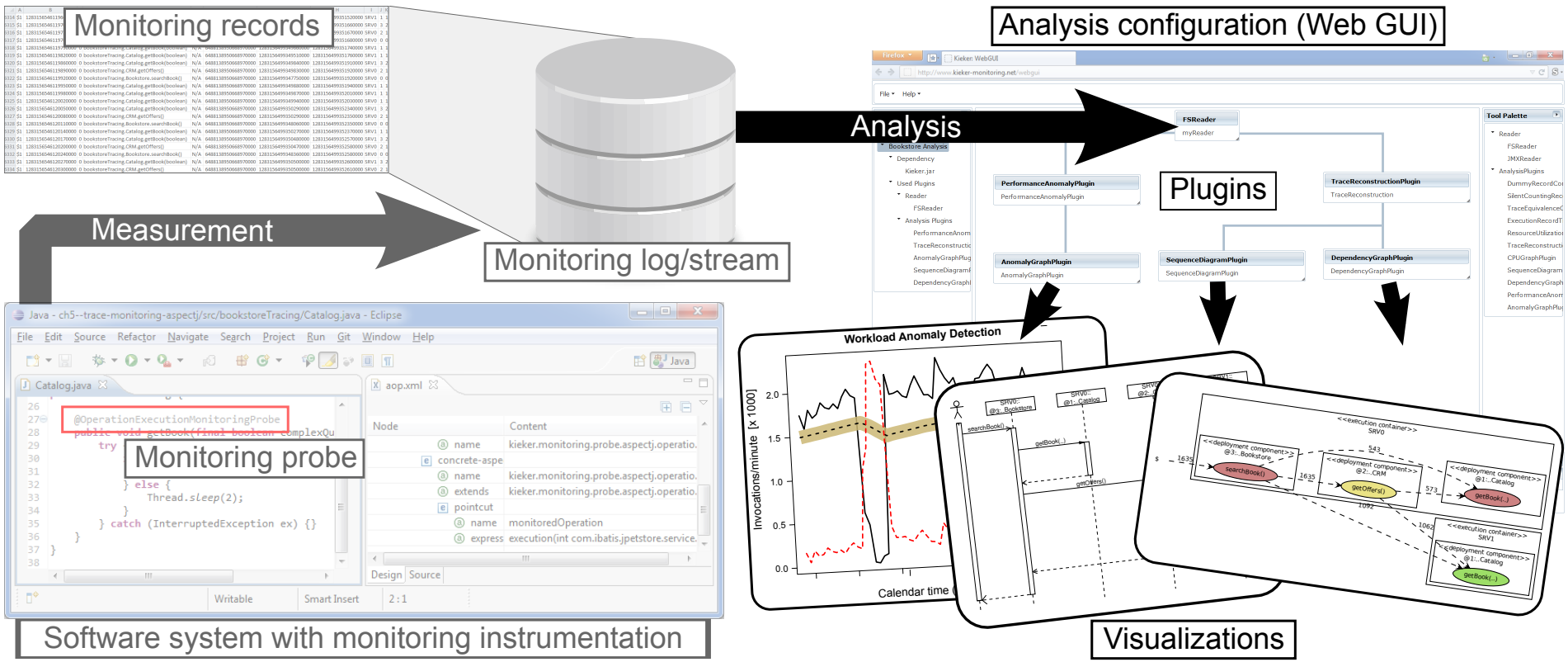
What is Kieker?



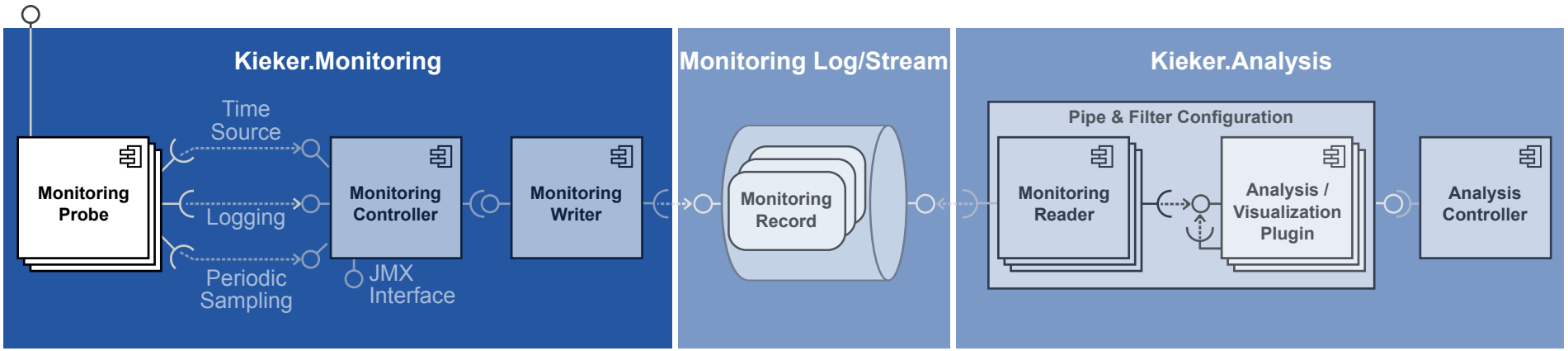


Dynamic Analysis with Kieker

What is Kieker?



Core Kieker Framework Components



Java probes/samplers:

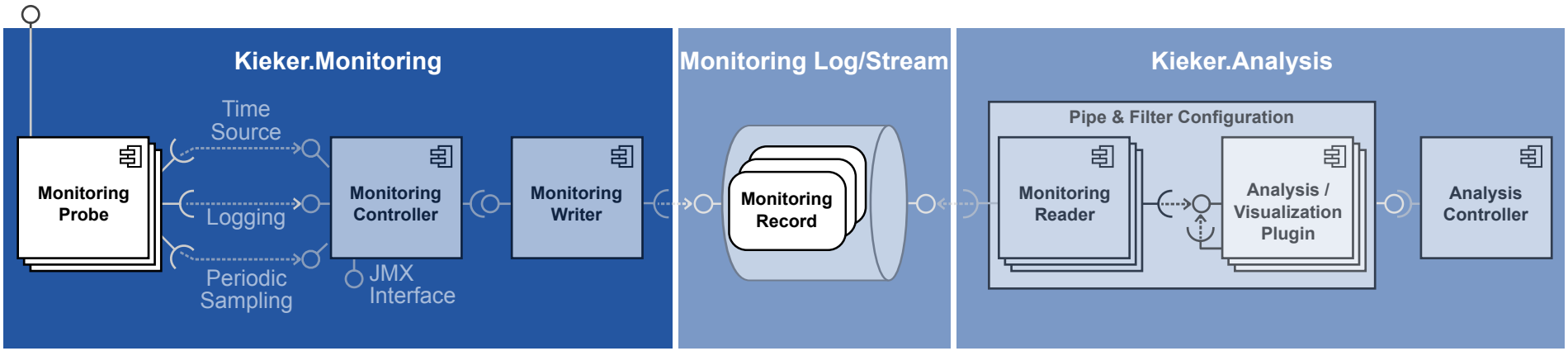
Monitoring Probes/Samplers

Control-flow tracing	Manual instrumentation		
	AspectJ	Spring	
	Servlet	CXF/SOAP	
	<your interception technology>		
Resource monitoring	Servlet / Sigar	CPU utilization	
		Memory usage	
	<your technology>		
<your monitoring probe>			

+ basic adapters for

- C#/.NET
- Visual Basic 6/COM
- COBOL

Core Kieker Framework Components



Java probes/samplers:

Monitoring Probes/Samplers

Control-flow tracing	Manual instrumentation		
	AspectJ	Spring	
	Servlet	CXF/SOAP	
	<your interception technology>		
Resource monitoring	Servlet / Sigar	CPU utilization	
		Memory usage	
	<your technology>		
<your monitoring probe>			

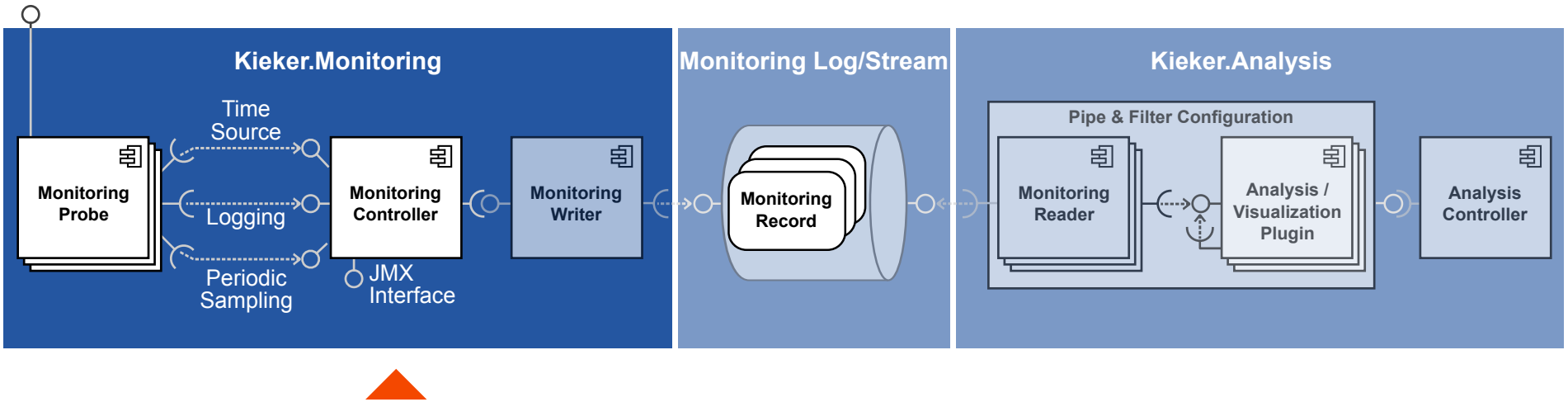
+ basic adapters for

- C#/.NET
- Visual Basic 6/COM
- COBOL

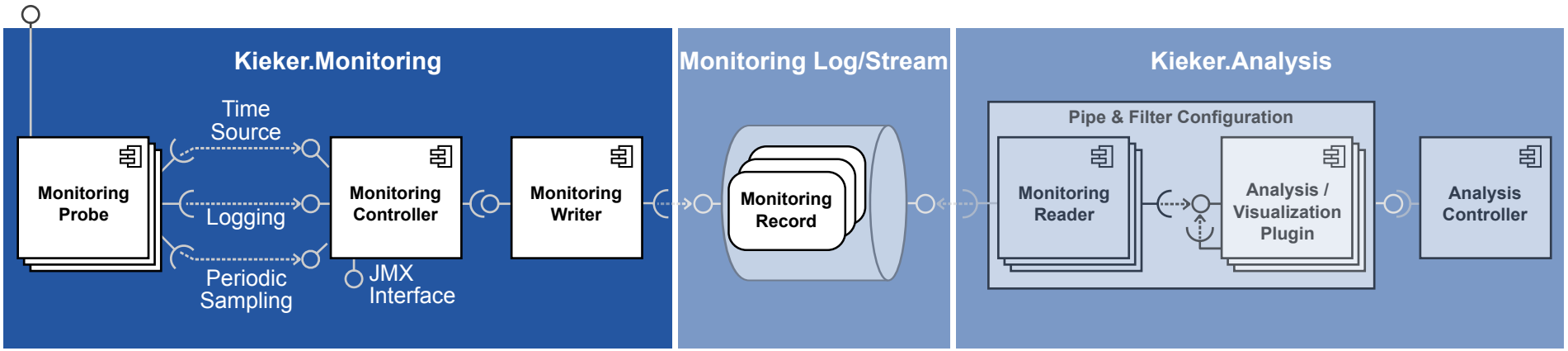
Monitoring Records

- Operation execution
- Control-flow events
- CPU utilization | Memory/swap usage
- Resource utilization
- Current time
- *<your monitoring record type>*

Core Kieker Framework Components

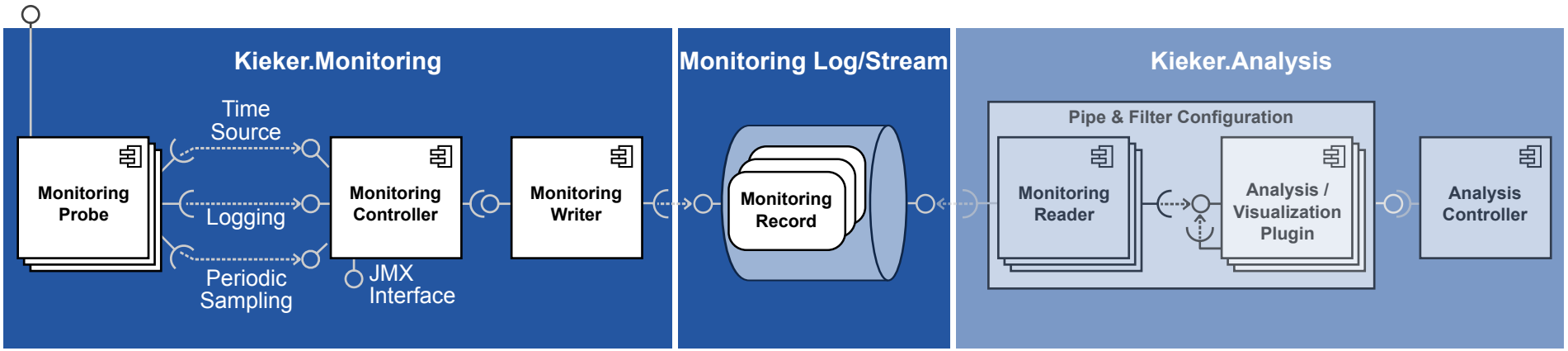


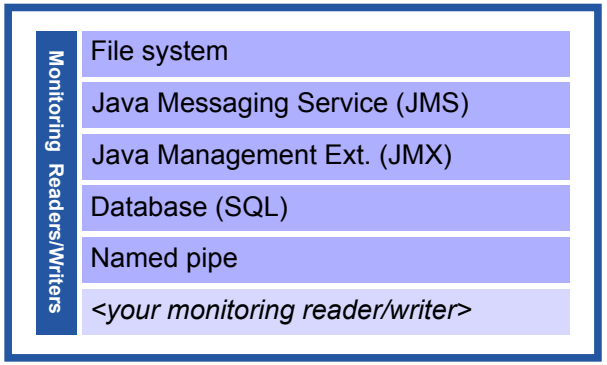
Core Kieker Framework Components



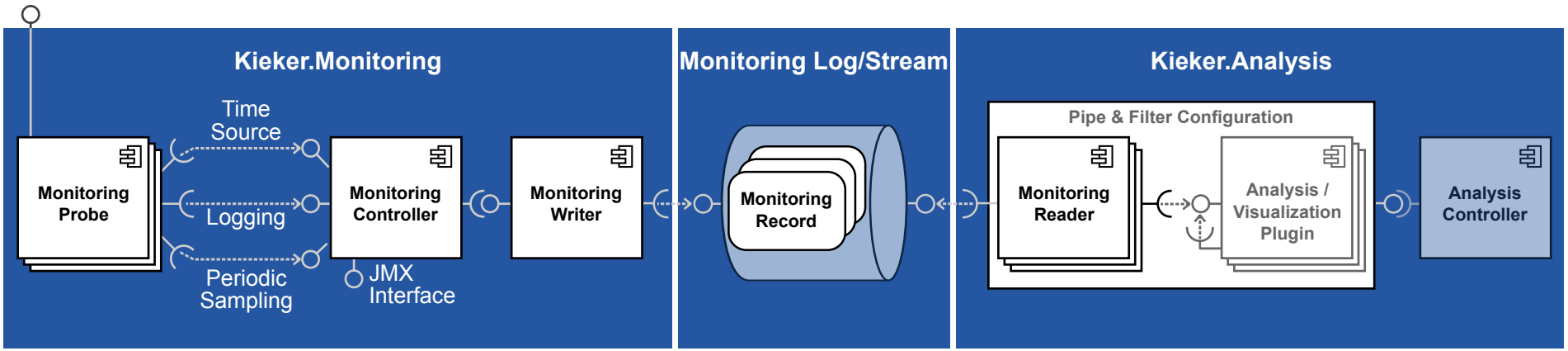
Monitoring Readers/Writers
File system
Java Messaging Service (JMS)
Java Management Ext. (JMX)
Database (SQL)
Named pipe
<your monitoring reader/writer>

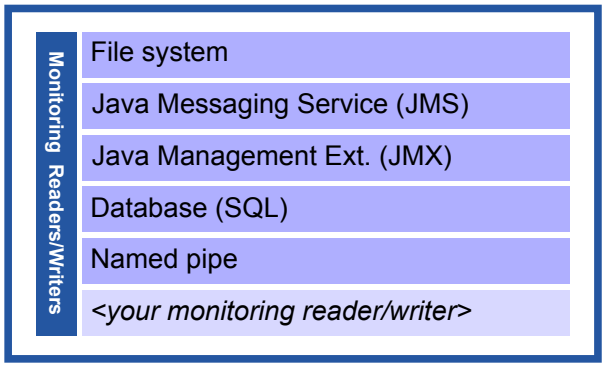
Core Kieker Framework Components



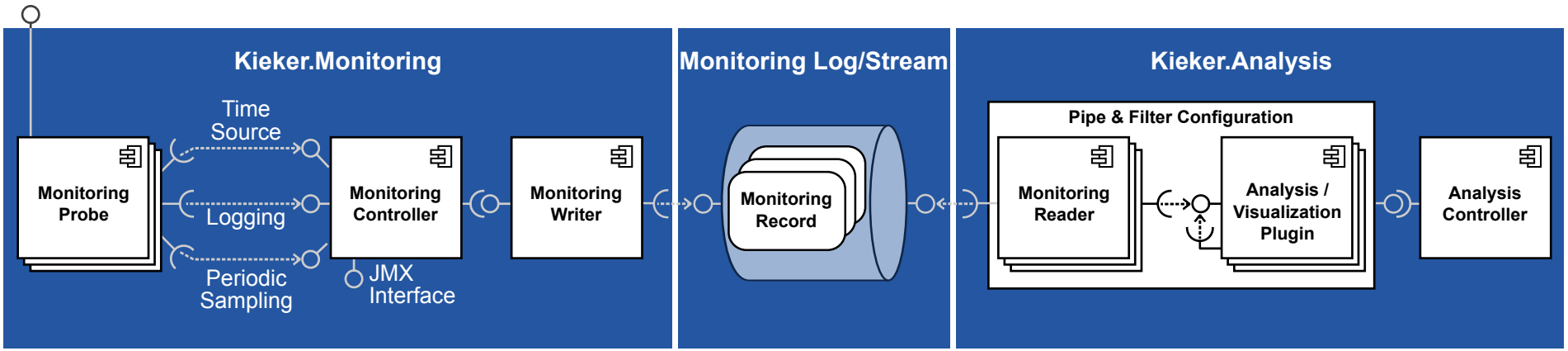


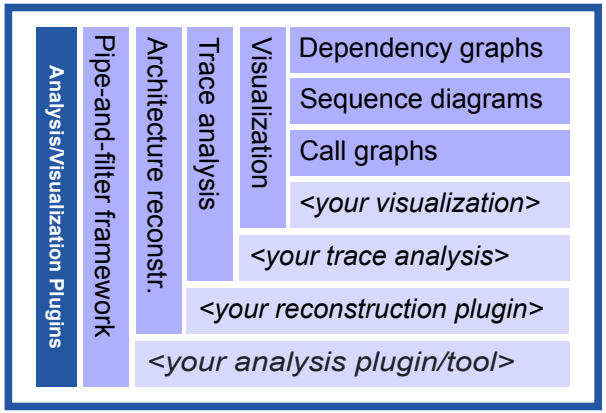
Core Kieker Framework Components



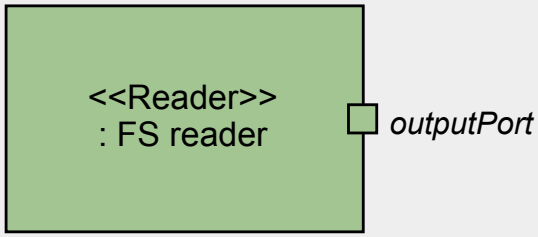


Core Kieker Framework Components

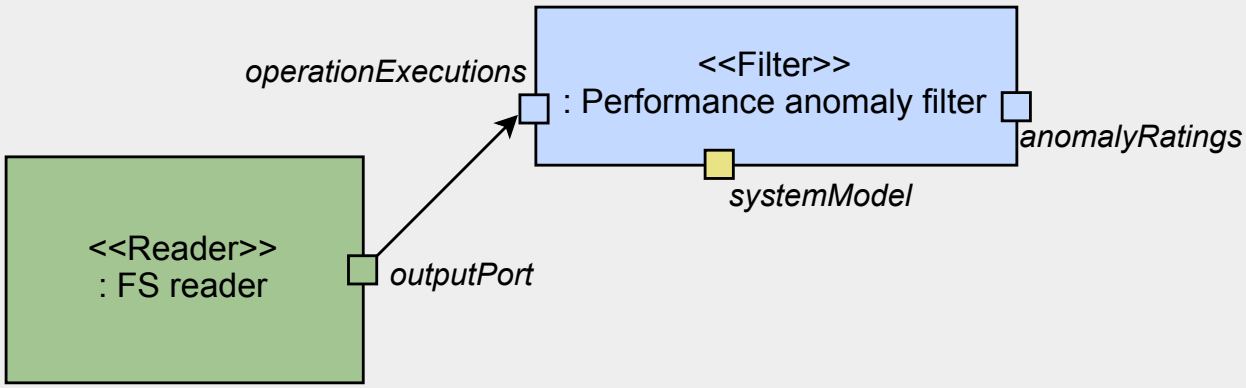




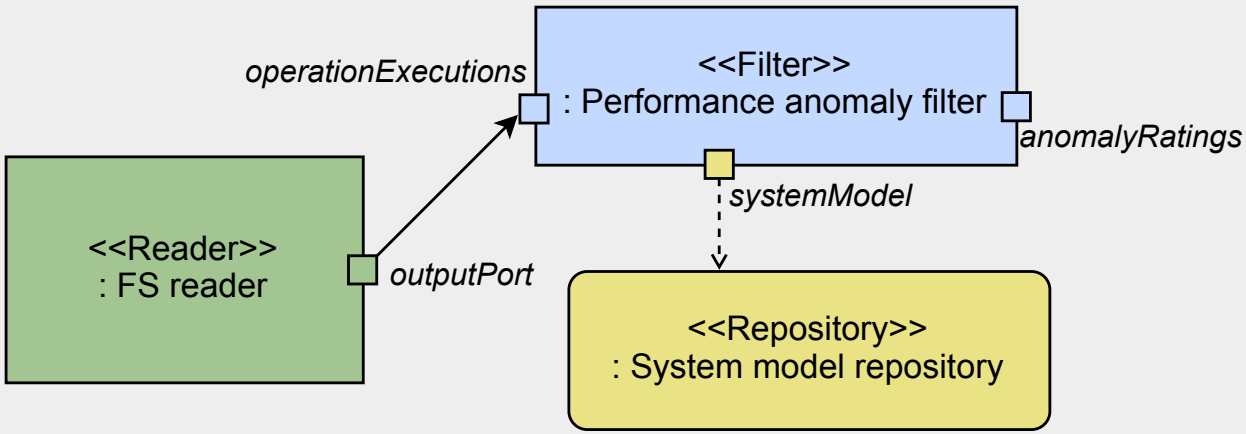
Example Pipe-and-Filter Configuration



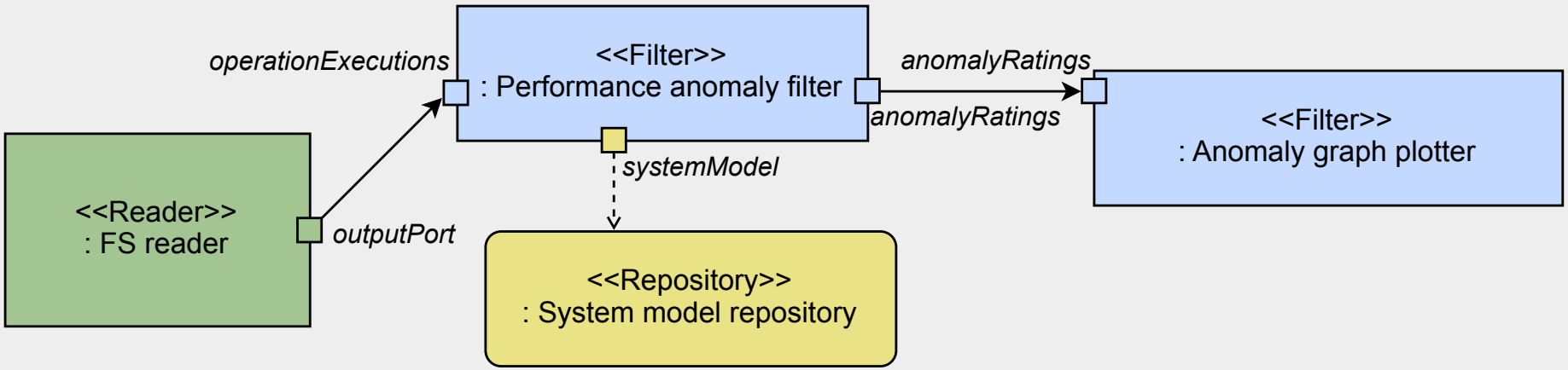
Example Pipe-and-Filter Configuration



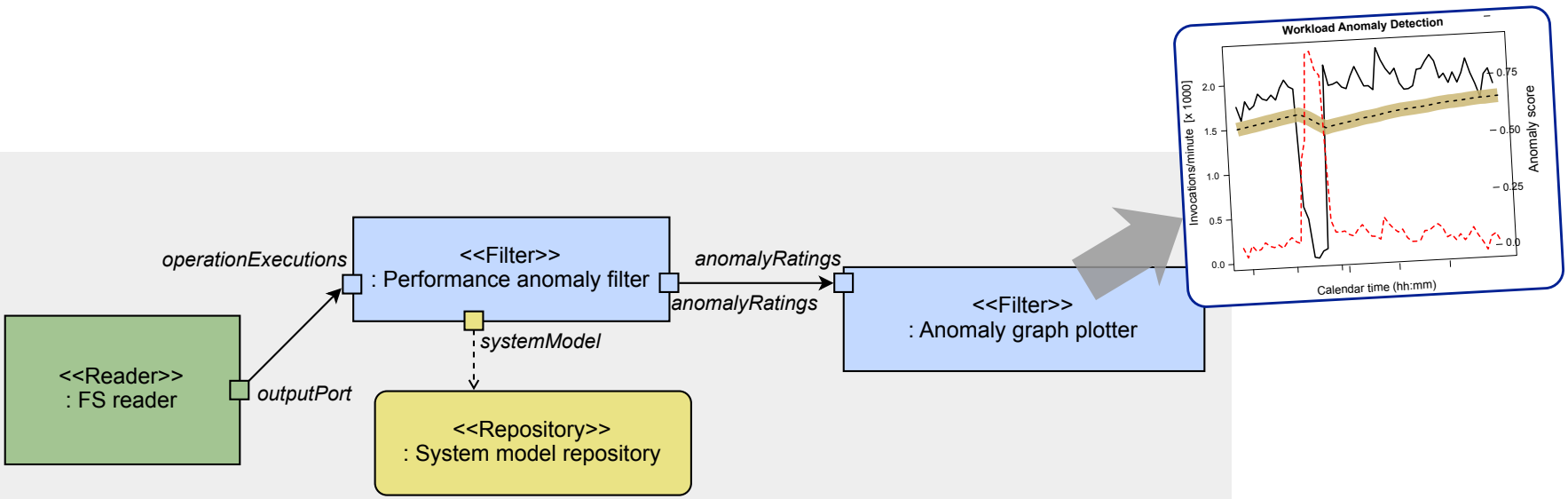
Example Pipe-and-Filter Configuration



Example Pipe-and-Filter Configuration



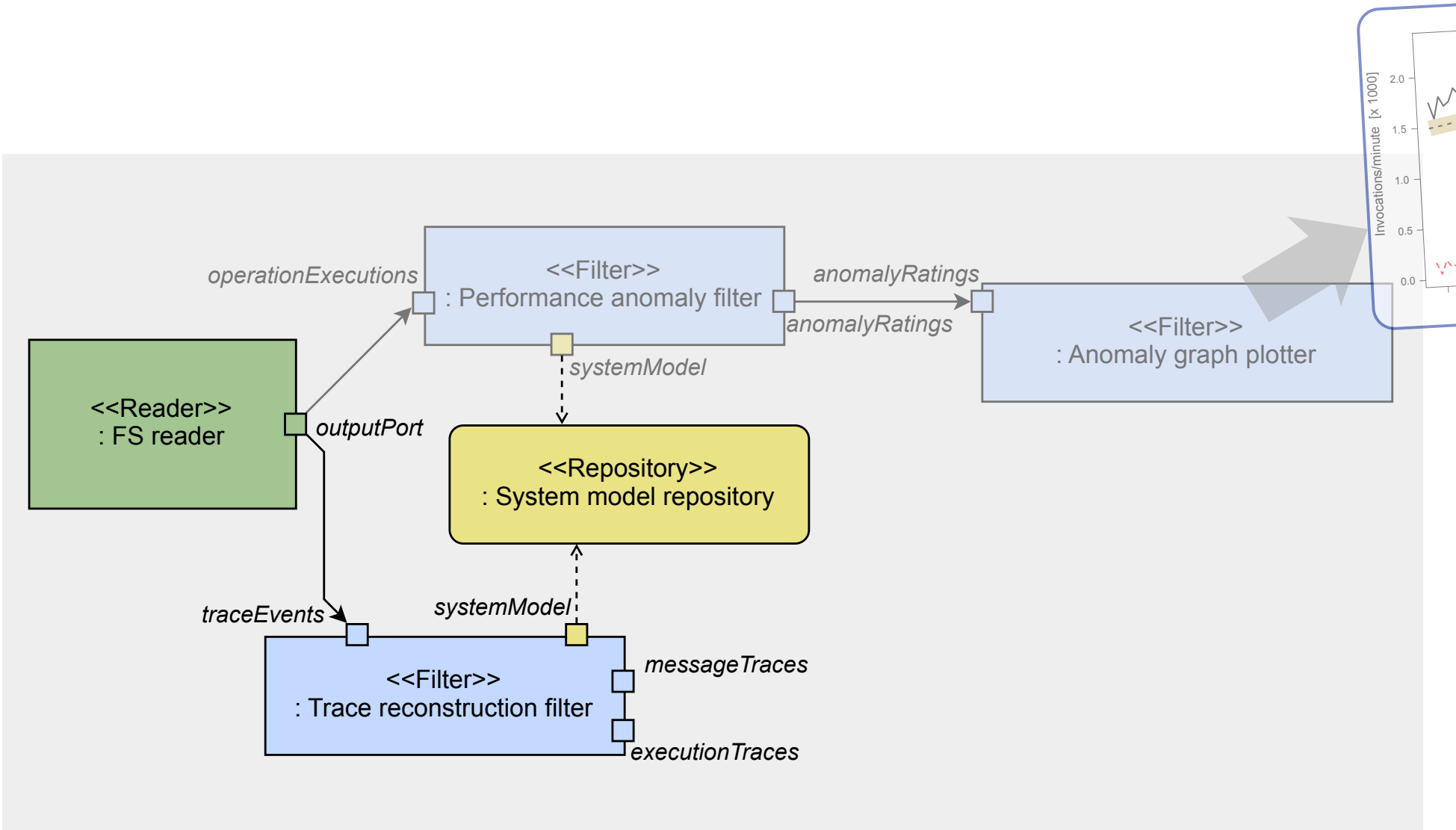
Example Pipe-and-Filter Configuration



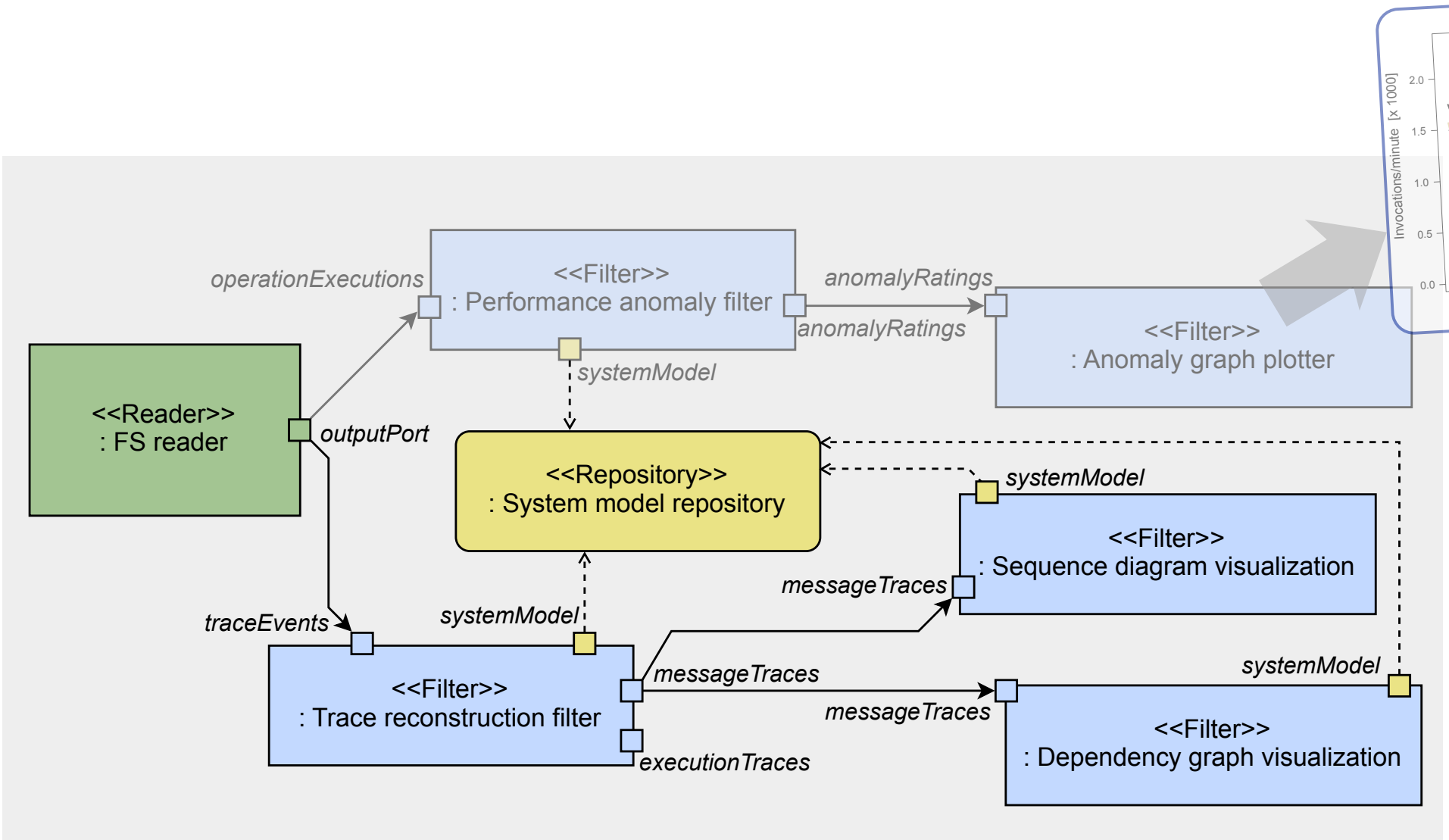
Kieker.Analysis example pipes-and-filters configuration

- *Performance anomaly detection and visualization*
- *Architecture and trace reconstruction/visualization*

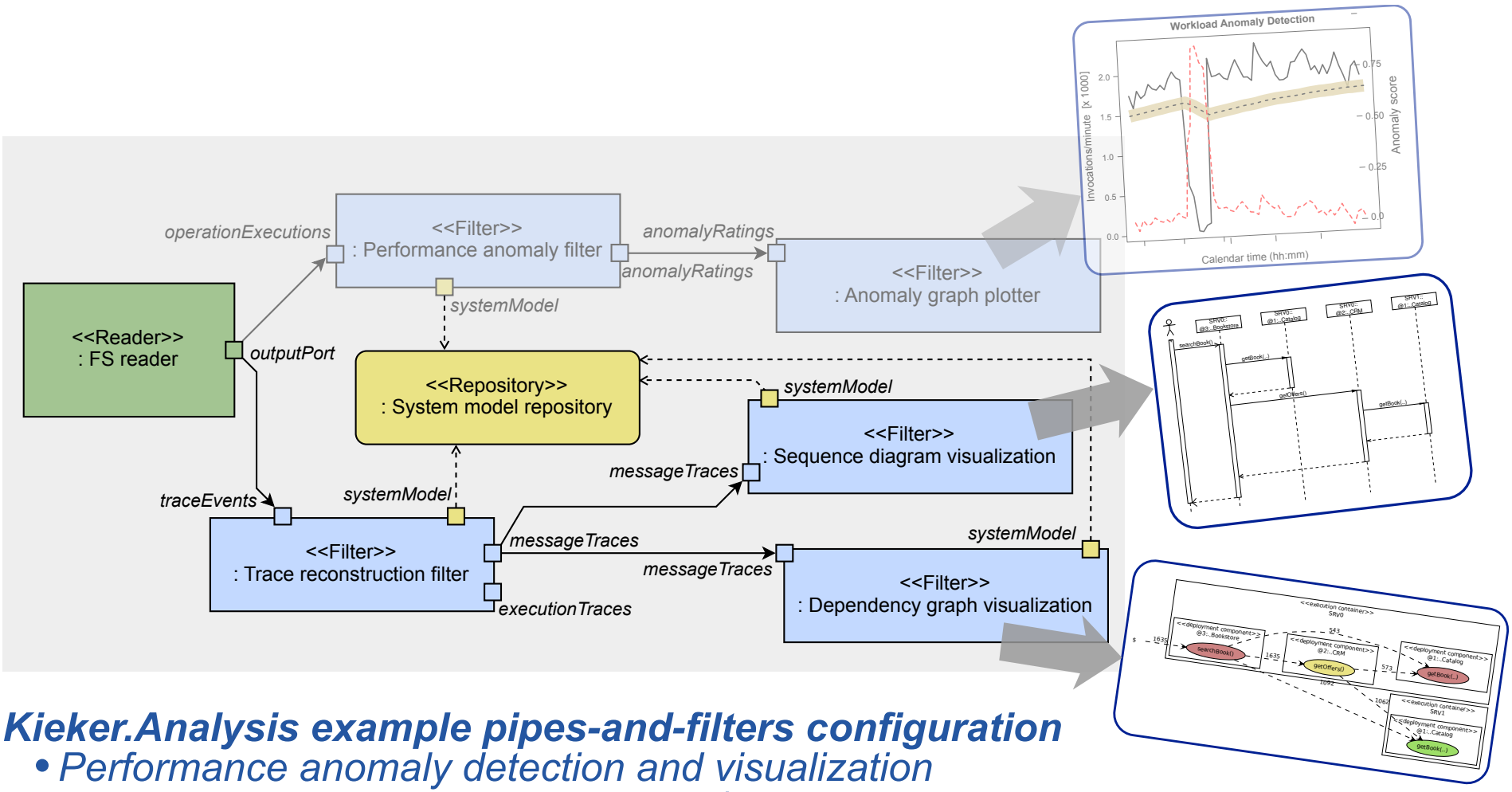
Example Pipe-and-Filter Configuration



Example Pipe-and-Filter Configuration

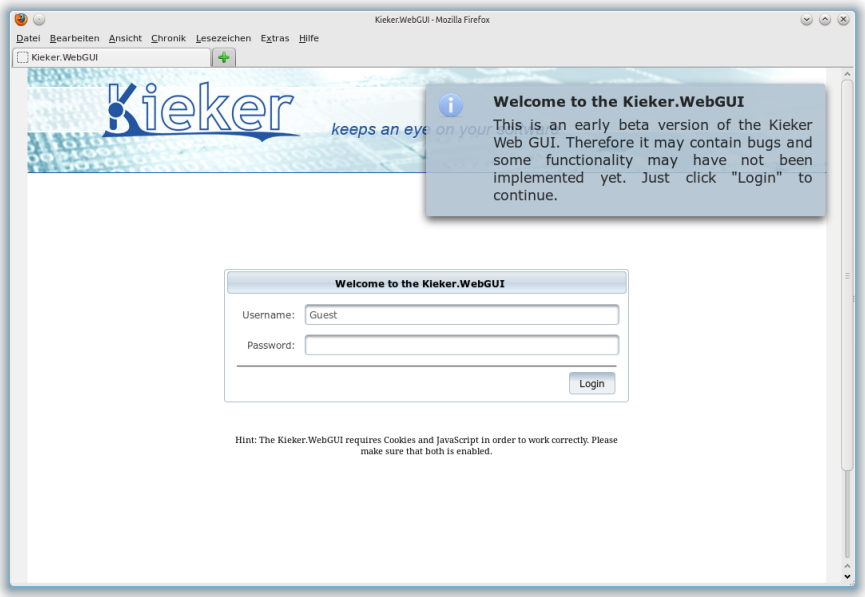


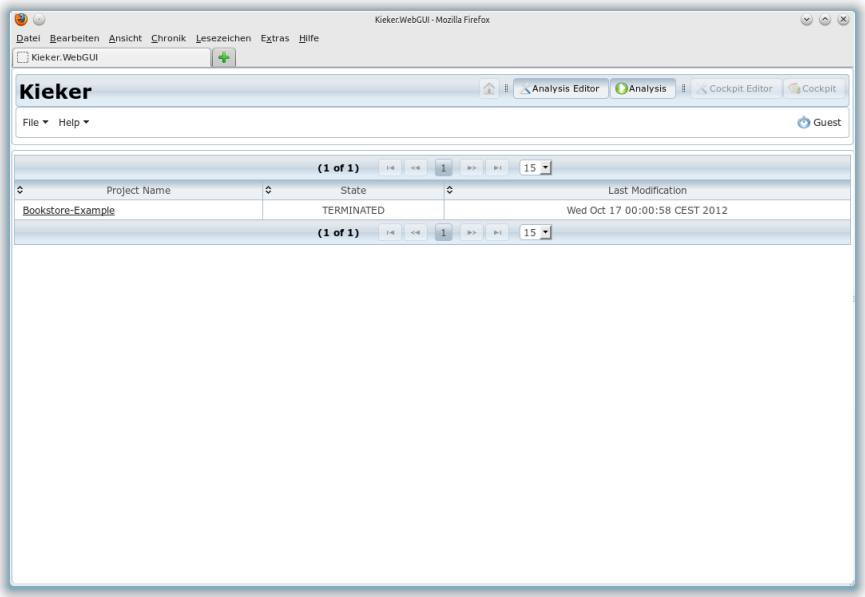
Example Pipe-and-Filter Configuration

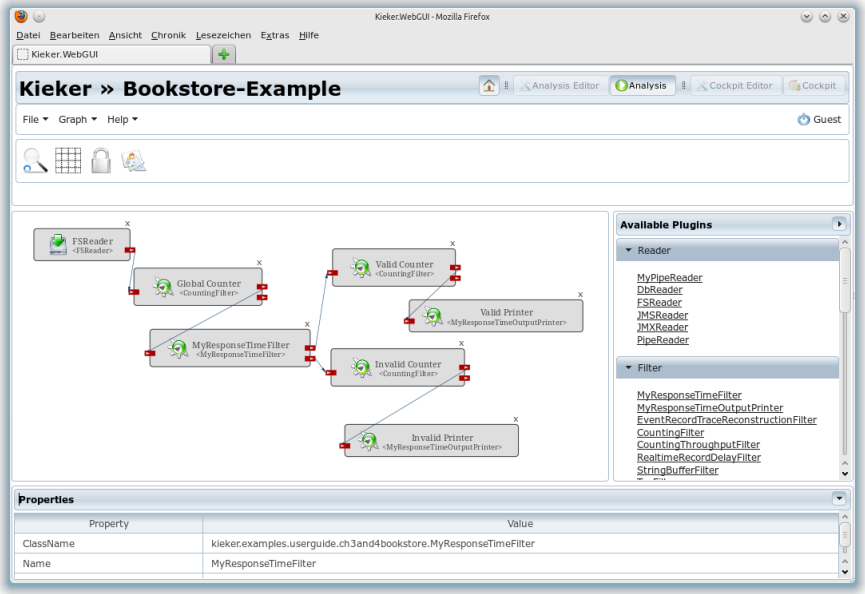


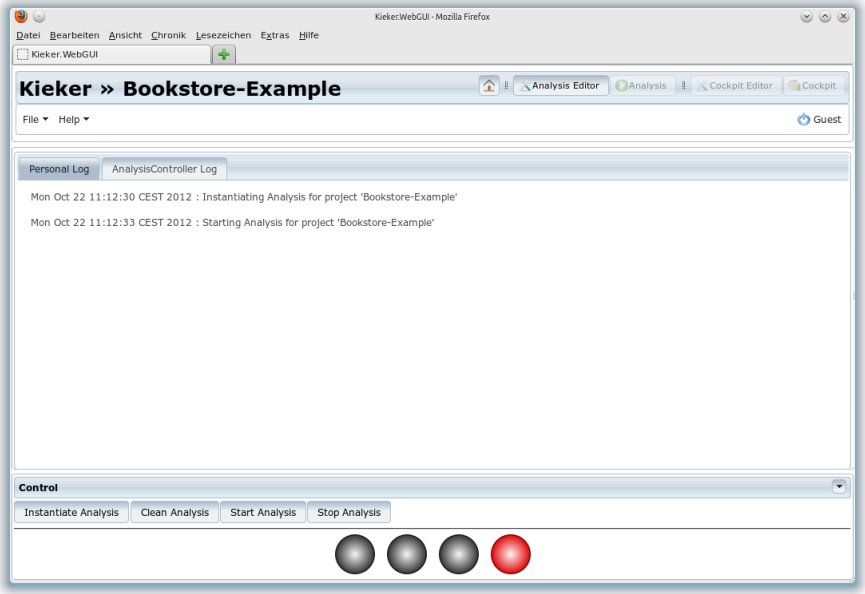
Kieker.Analysis example pipes-and-filters configuration

- *Performance anomaly detection and visualization*
- *Architecture and trace reconstruction/visualization*


Kieker.WebGUI - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
Kieker.WebGUI
Welcome to the Kieker.WebGUI
This is an early beta version of the Kieker Web GUI. Therefore it may contain bugs and some functionality may have not been implemented yet. Just click "Login" to continue.
Welcome to the Kieker.WebGUI
Username:
Guest
Password:
Login
Hint: The Kieker.WebGUI requires Cookies and JavaScript in order to work correctly. Please make sure that both is enabled.


Kieker.WebGUI - Mozilla Firefox
Kieker
Analysis Editor
Analysis
Cockpit Editor
Cockpit
File
Help
Guest
(1 of 1)
Project Name
State
Last Modification
Bookstore-Example
TERMINATED
Wed Oct 17 00:00:58 CEST 2012


Kieker.WebGUI - Mozilla Firefox
Kieker » Bookstore-Example
File
Graph
Help
Guest
FSReader
Global Counter
Valid Counter
Valid Printer
MyResponseTimeFilter
Invalid Counter
Invalid Printer
Available Plugins
Reader
MyPipeReader
DbReader
FSReader
JMSReader
JMXReader
PipeReader
Filter
MyResponseTimeFilter
MyResponseTimeOutputPrinter
EventRecordTraceReconstructionFilter
CountingFilter
CountingThroughputFilter
RealtimeRecordDelayFilter
StringBufferFilter
Properties
Property
Value
ClassName
kieker.examples.userguide.ch3and4bookstore.MyResponseTimeFilter
Name
MyResponseTimeFilter


Kieker.WebGUI - Mozilla Firefox
Kieker » Bookstore-Example
File
Help
Guest
Personal Log
AnalysisController Log
Mon Oct 22 11:12:30 CEST 2012 : Instantiating Analysis for project 'Bookstore-Example'
Mon Oct 22 11:12:33 CEST 2012 : Starting Analysis for project 'Bookstore-Example'
Control
Instantiate Analysis
Clean Analysis
Start Analysis
Stop Analysis

Agenda

1 ~~Motivation: Performance, Monitoring, Dynamic Analysis, . . .~~

2 What is Kieker?

3 What is Kieker Good For?

4 Who has been/is Working on Kieker?

Kieker Use Cases/Characteristics

1. Architecture Discovery (Dynamic/Hybrid Analysis)

2. Application Performance Management

Kieker Use Cases/Characteristics

1. Architecture Discovery (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)

2. Application Performance Management

Kieker Use Cases/Characteristics

1. Architecture Discovery (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems

2. Application Performance Management

1. Architecture Discovery (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems
   - Software visualization (2D/3D)

2. Application Performance Management

1. Architecture Discovery (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems
   - Software visualization (2D/3D)
   - Trace-based architecture analysis
2. Application Performance Management

Kieker Use Cases/Characteristics

1. Architecture Discovery (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems
   - Software visualization (2D/3D)
   - Trace-based architecture analysis
2. Application Performance Management
   - Continuous QoS monitoring + feedback (self-*)

Kieker Use Cases/Characteristics

1. Architecture Discovery (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems
   - Software visualization (2D/3D)
   - Trace-based architecture analysis
2. Application Performance Management
   - Continuous QoS monitoring + feedback (self-*)
   - Distributed tracing and trace-based analysis

Kieker Use Cases/Characteristics

1. Architecture Discovery (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems
   - Software visualization (2D/3D)
   - Trace-based architecture analysis
2. Application Performance Management
   - Continuous QoS monitoring + feedback (self-*)
   - Distributed tracing and trace-based analysis
   - Architecture-based performance analysis

1. Architecture Discovery (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems
   - Software visualization (2D/3D)
   - Trace-based architecture analysis
2. Application Performance Management
   - Continuous QoS monitoring + feedback (self-*)
   - Distributed tracing and trace-based analysis
   - Architecture-based performance analysis
   - Automatic problem detection and diagnosis

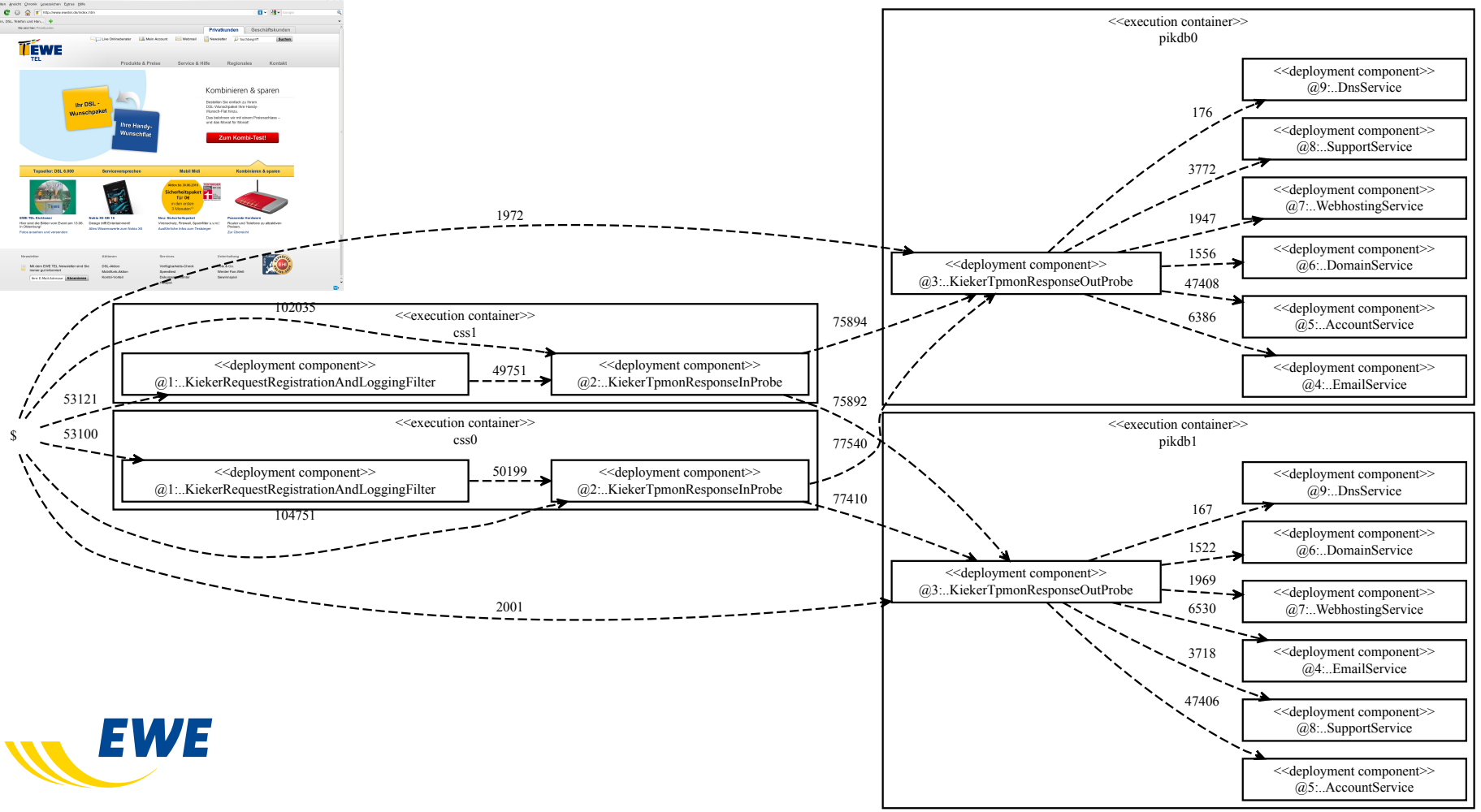
Kieker Use Cases/Characteristics

1. Architecture Discovery (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems
   - Software visualization (2D/3D)
   - Trace-based architecture analysis
2. Application Performance Management
   - Continuous QoS monitoring + feedback (self-*)
   - Distributed tracing and trace-based analysis
   - Architecture-based performance analysis
   - Automatic problem detection and diagnosis
   - Extraction of usage profiles (workload intensity, navigational patterns)

Kieker Use Cases/Characteristics

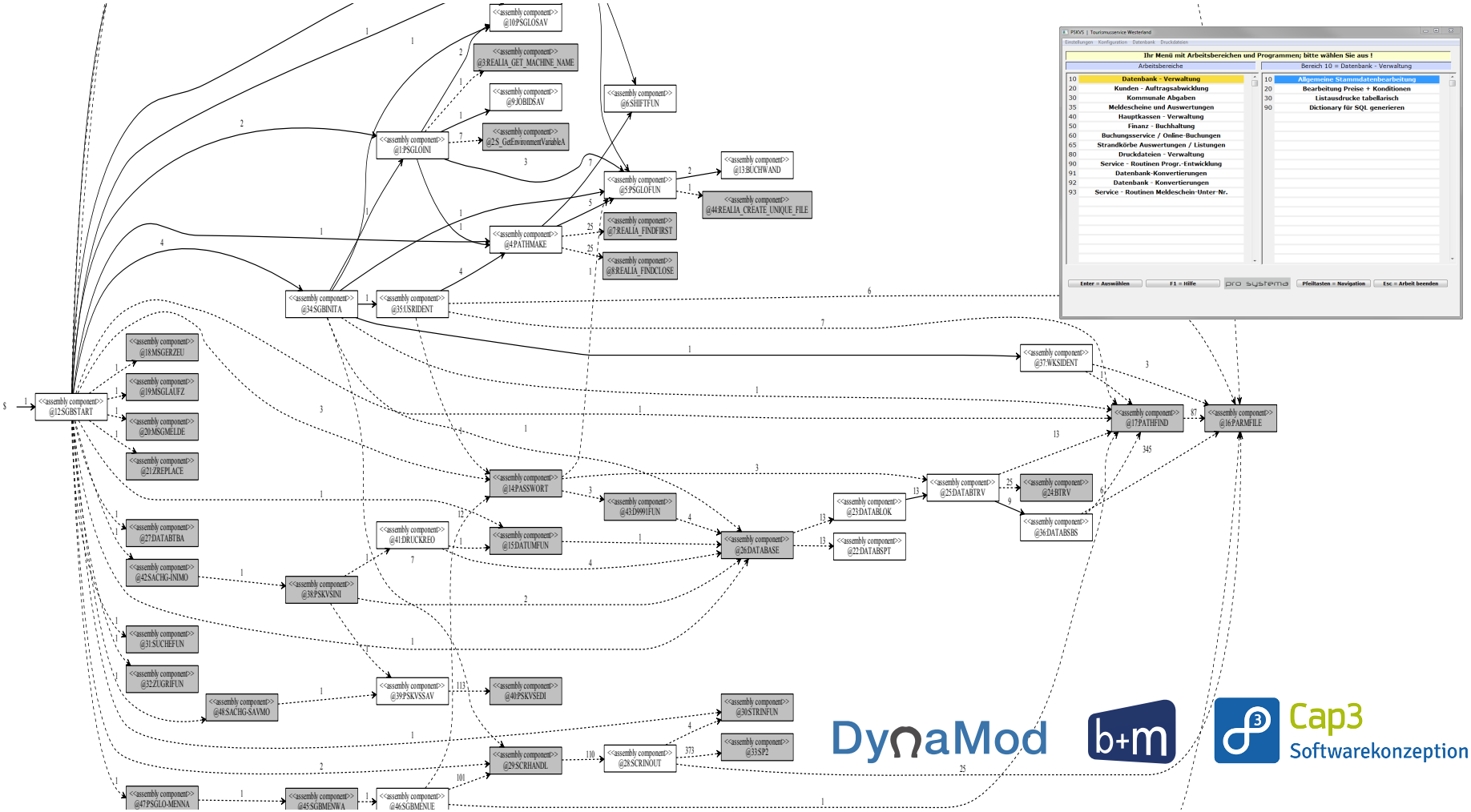
1. Architecture Discovery (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems
   - Software visualization (2D/3D)
   - Trace-based architecture analysis
2. Application Performance Management
   - Continuous QoS monitoring + feedback (self-*)
   - Distributed tracing and trace-based analysis
   - Architecture-based performance analysis
   - Automatic problem detection and diagnosis
   - Extraction of usage profiles (workload intensity, navigational patterns)
3. Characteristics (cross-cutting)
   - Modular, flexible, and extensible architecture
   - Non-intrusive instrumentation
   - Low performance overhead
   - Model-driven instrumentation and analysis
   - Evaluated in lab and industrial case studies

Architecture Discovery: Model Extraction + Visualization



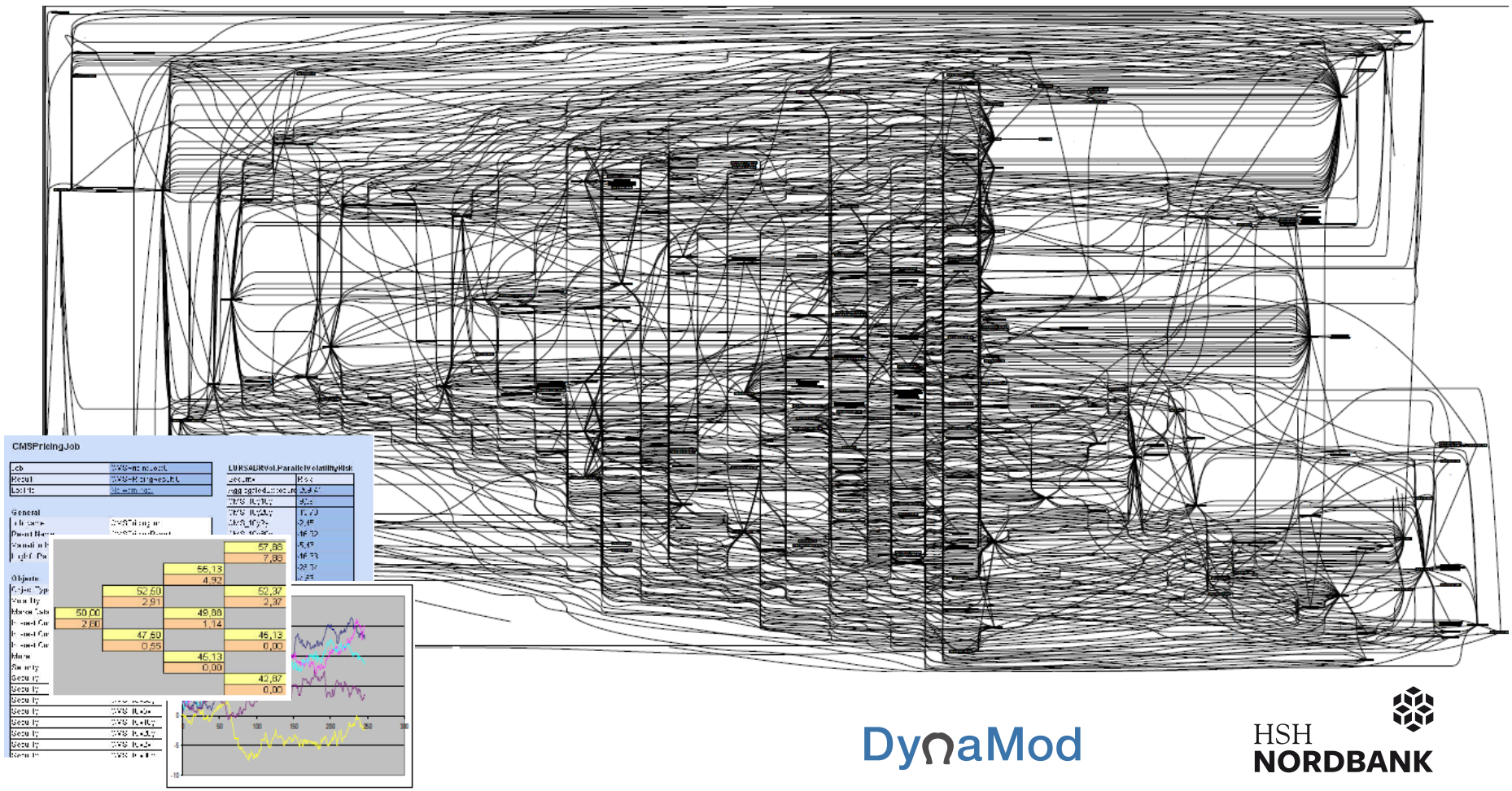
[van Hoorn et al. 2009]

Architecture Discovery: Model Extraction + Visualization (cont'd)



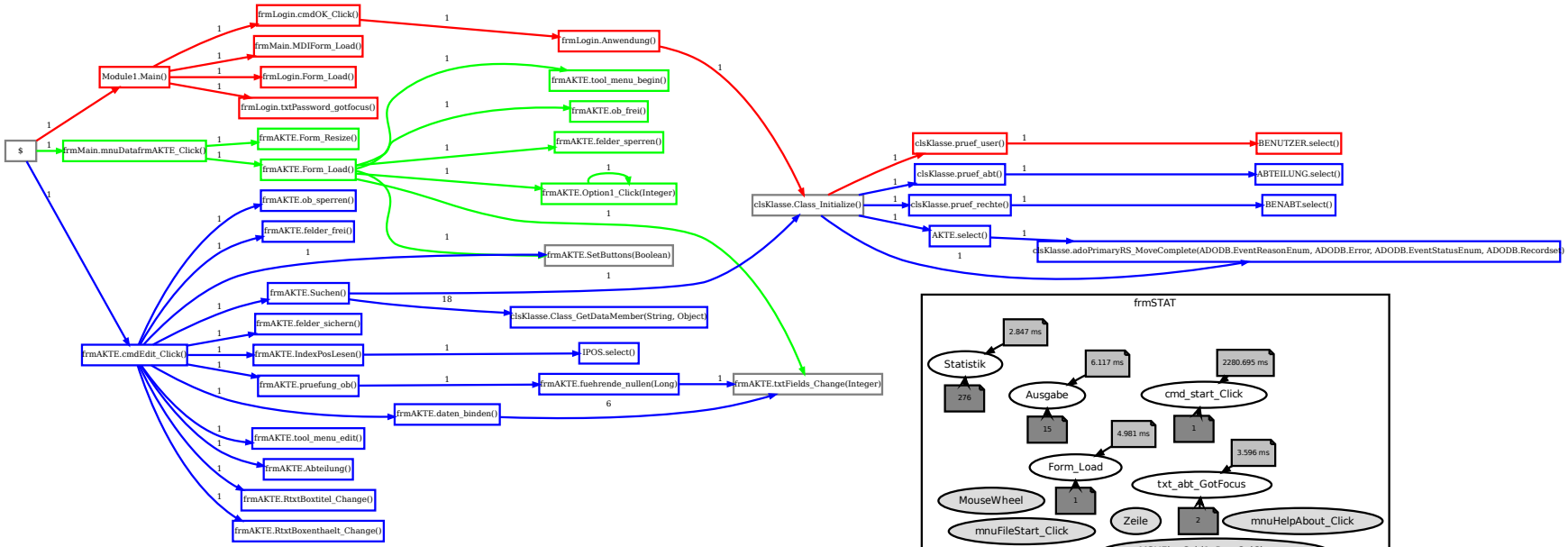
[Richter 2012]

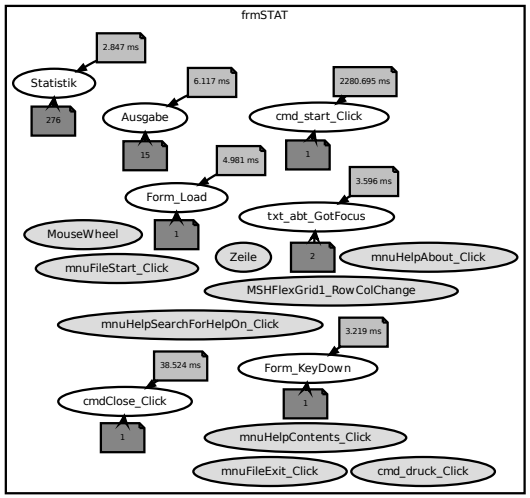
Architecture Discovery: Model Extraction + Visualization (cont'd)



[Magedanz 2011]

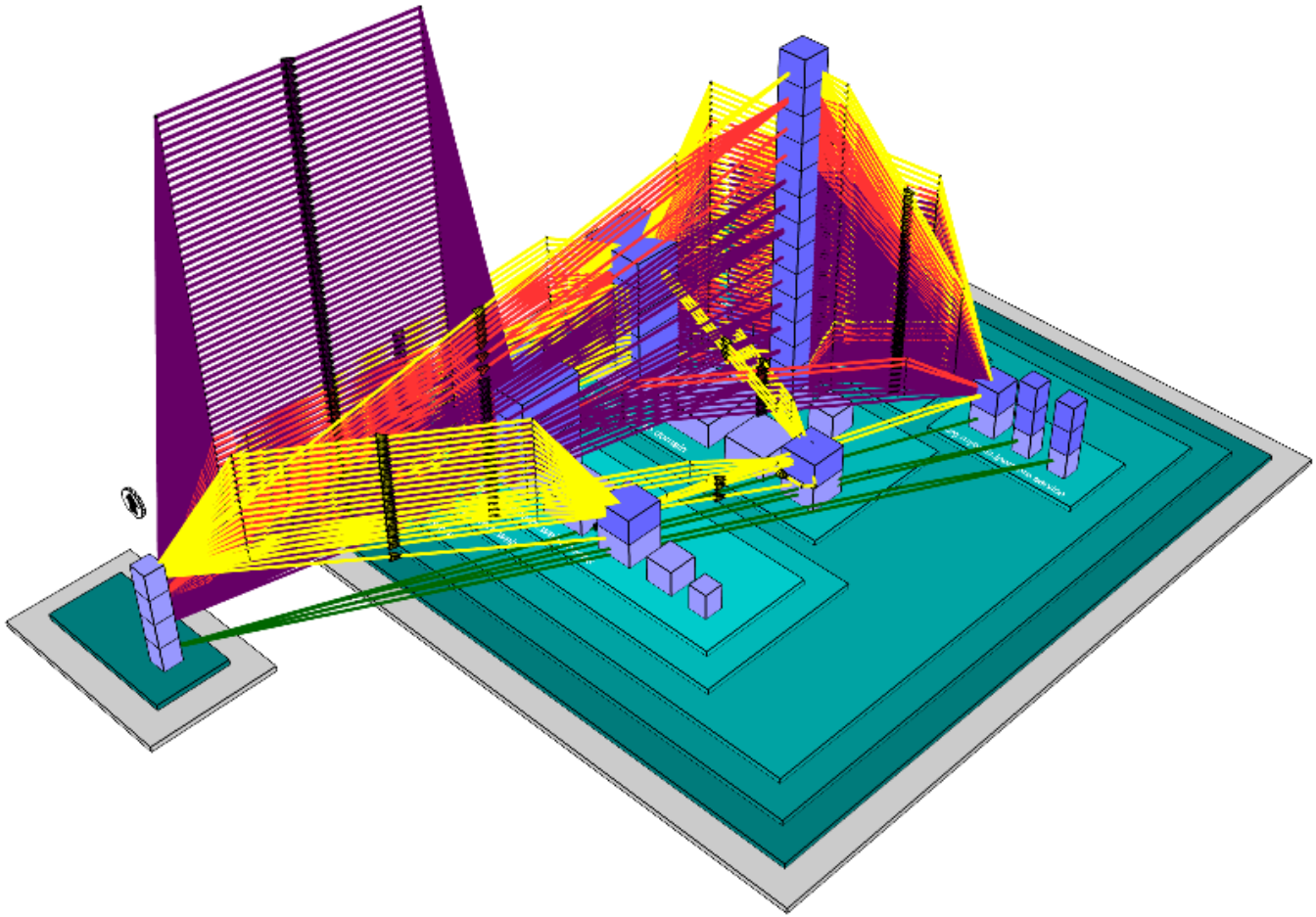
Architecture Discovery: Model Extraction + Visualization (cont'd)





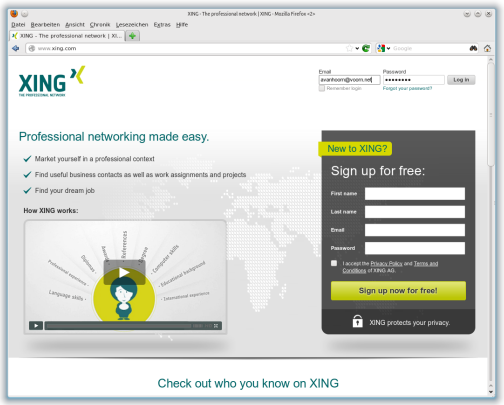
DynaMod b+m dataport

Architecture Discovery: Model Extraction + Visualization (cont'd)

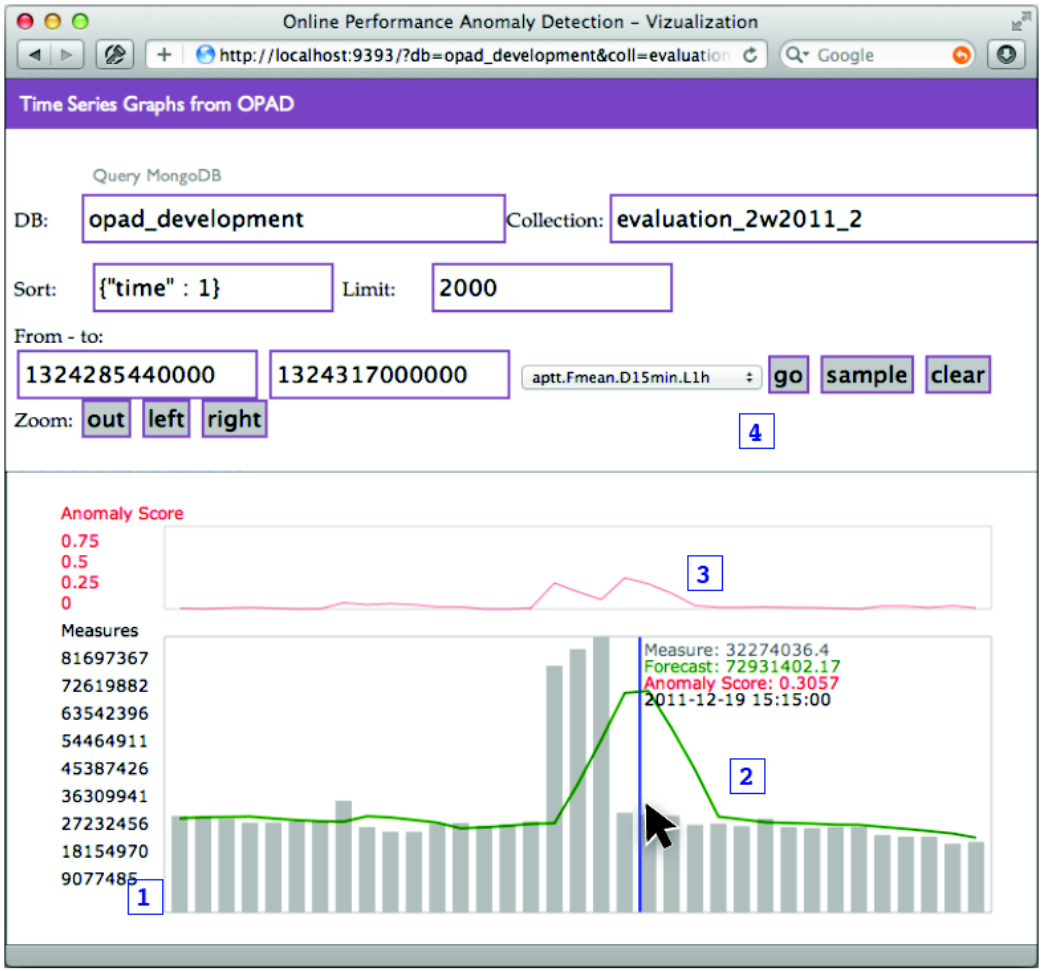


[Döhring 2012] (based on [Wulf 2010])

APM: Anomaly Detection + Diagnosis

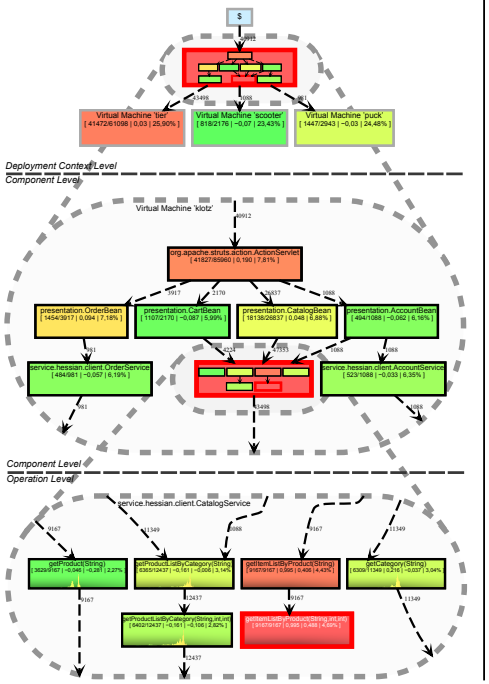


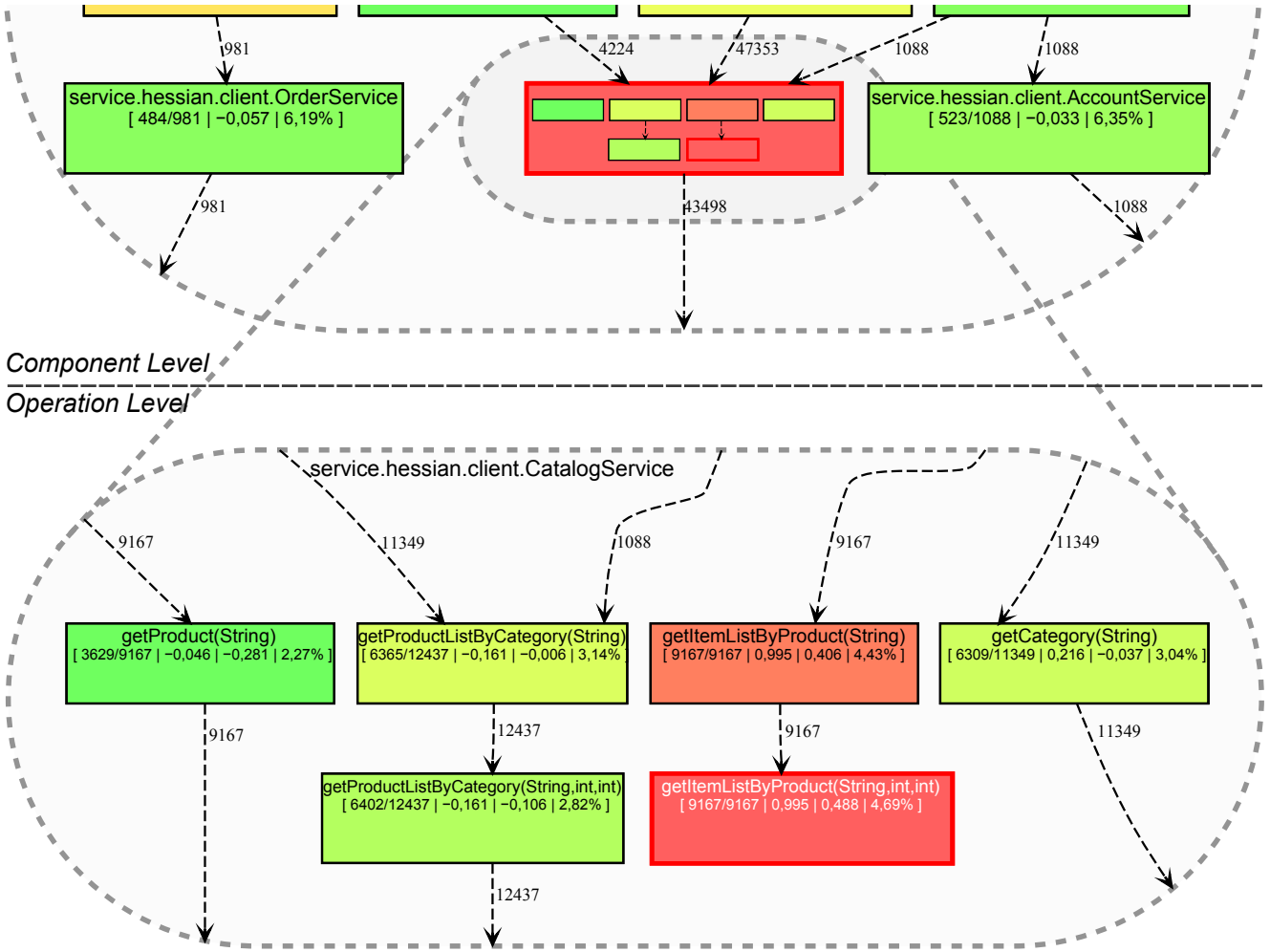
XING



[Bielefeld 2012, Frotscher 2013]

APM: Anomaly Detection + Diagnosis (cont'd)





[Marwede et al. 2009]

1 ~~Motivation: Performance, Monitoring, Dynamic Analysis, . . .~~

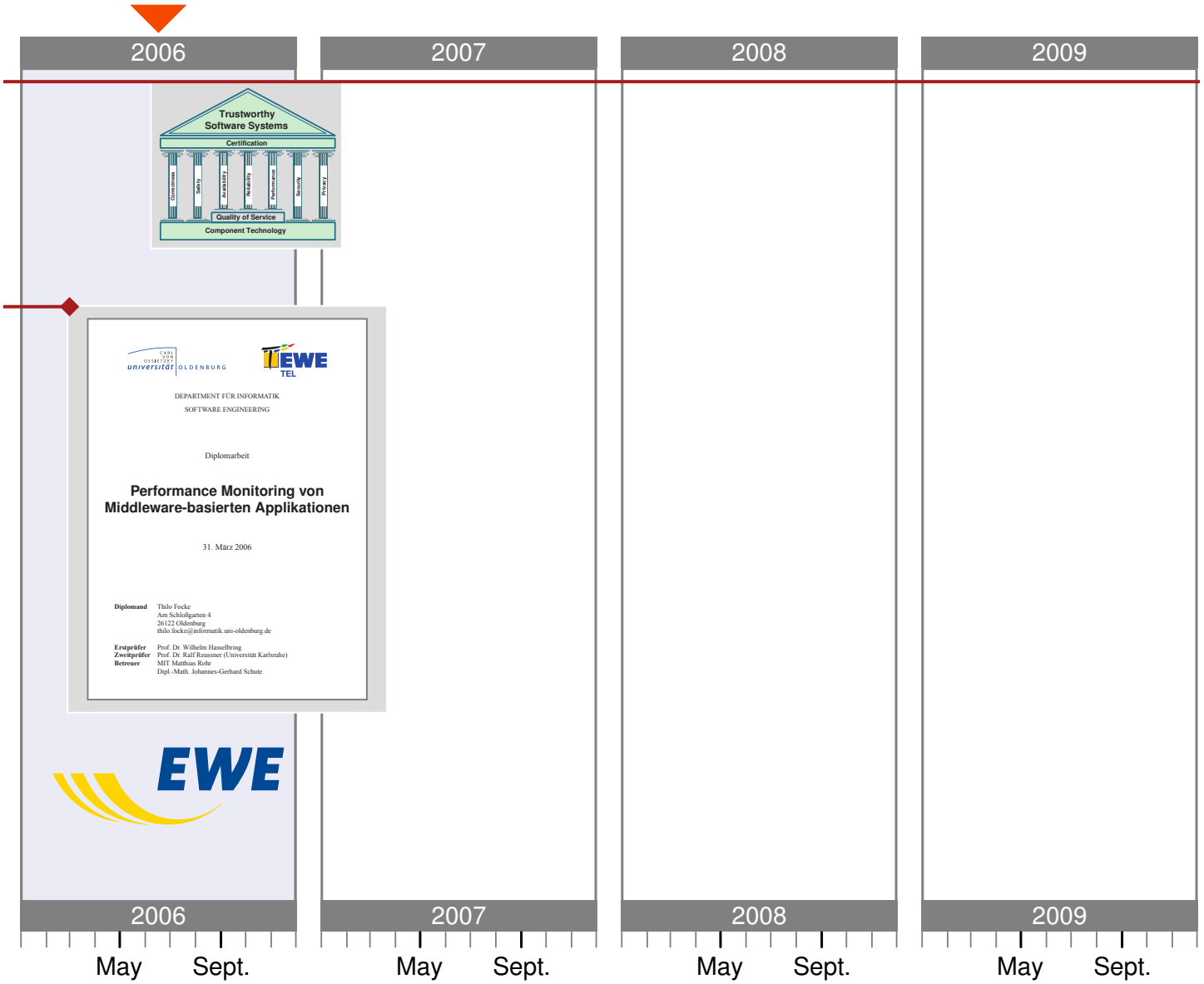
2 What is Kieker?

3 What is Kieker Good For?

4 Who has been/is Working on Kieker?

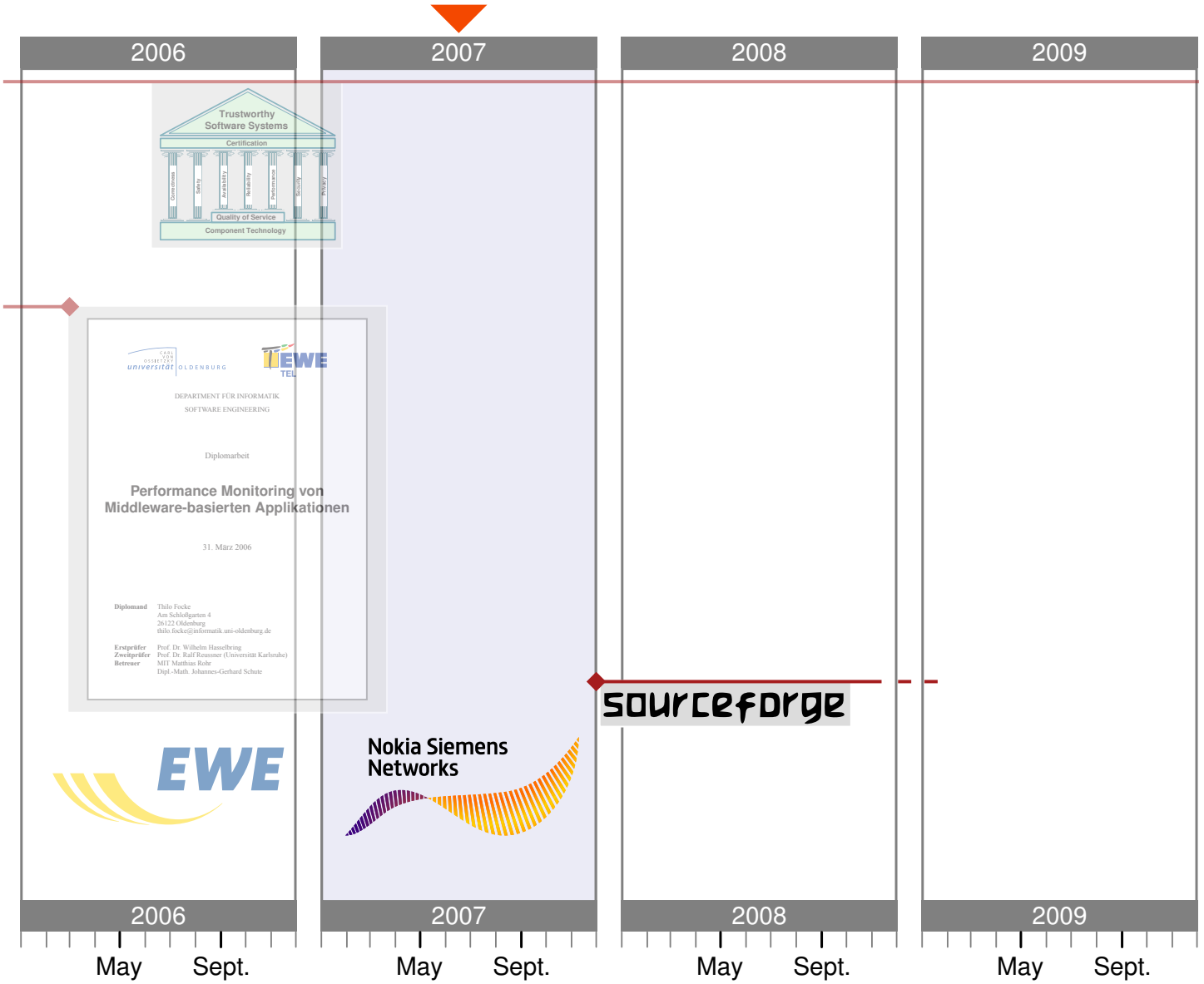
Looking back . . . 2006–2009

Who has been/is Working on Kieker?



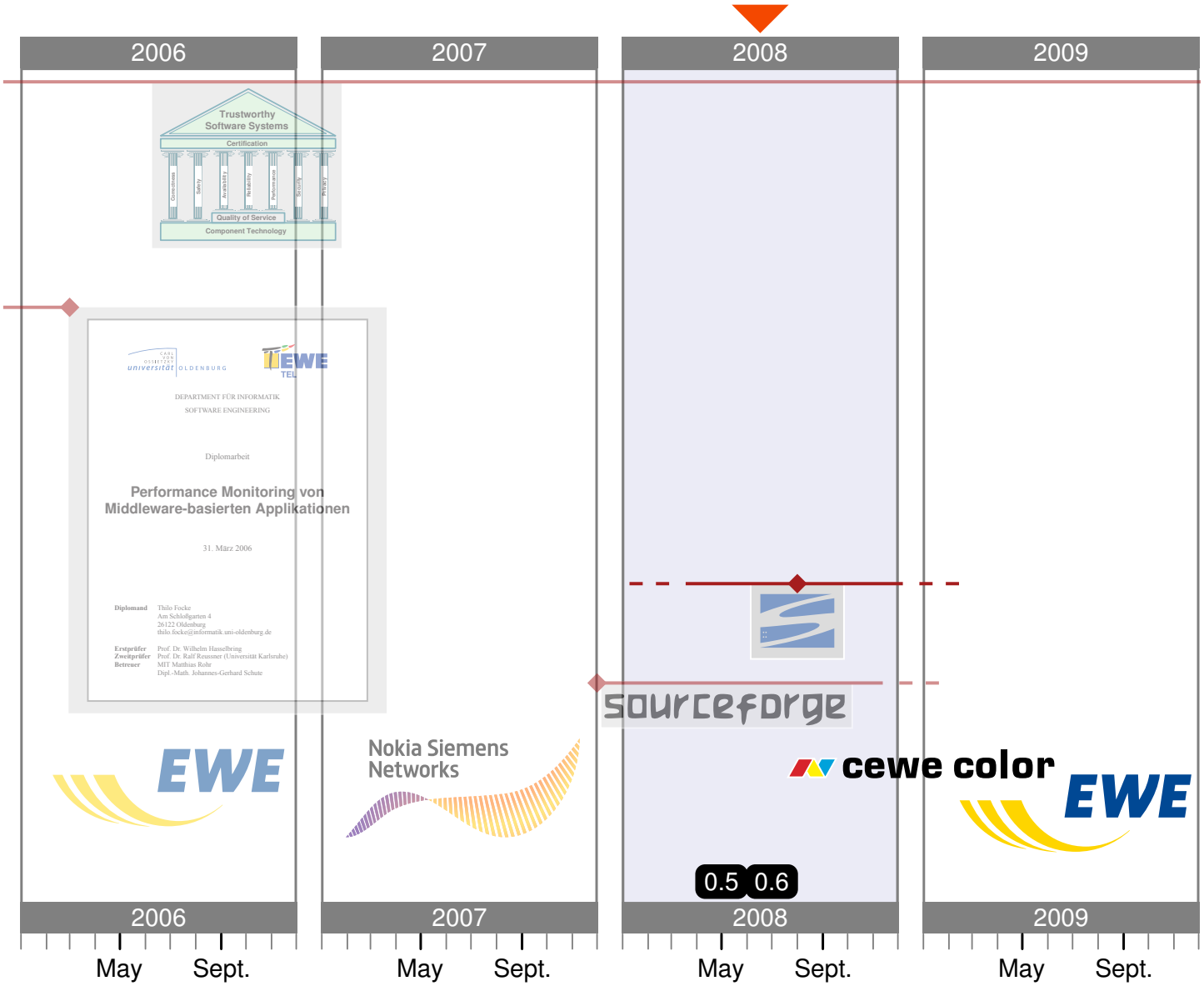
Looking back . . . 2006–2009

Who has been/is Working on Kieker?



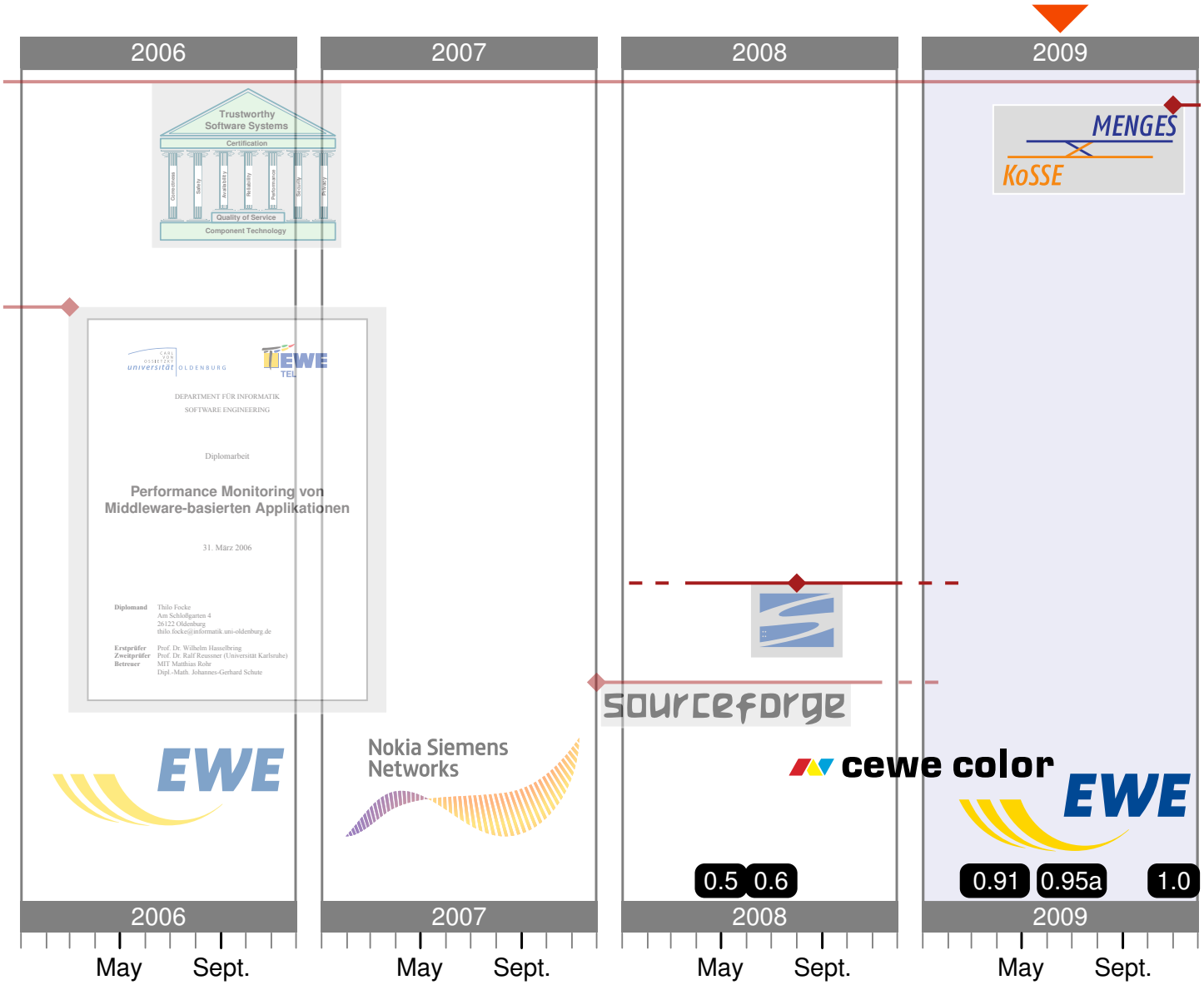
Looking back . . . 2006–2009

Who has been/is Working on Kieker?



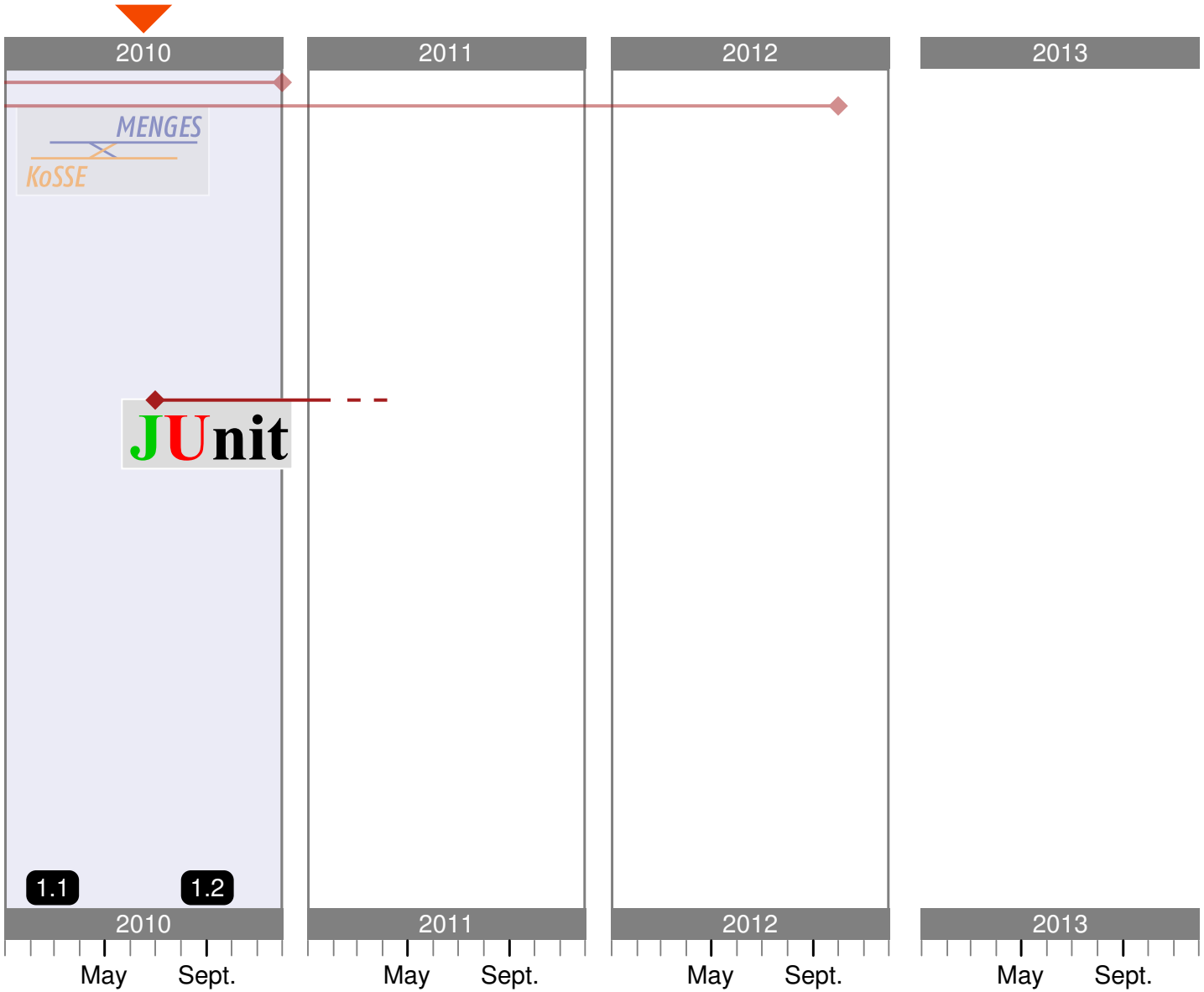
Looking back . . . 2006–2009

Who has been/is Working on Kieker?



Looking back . . . 2010–2013

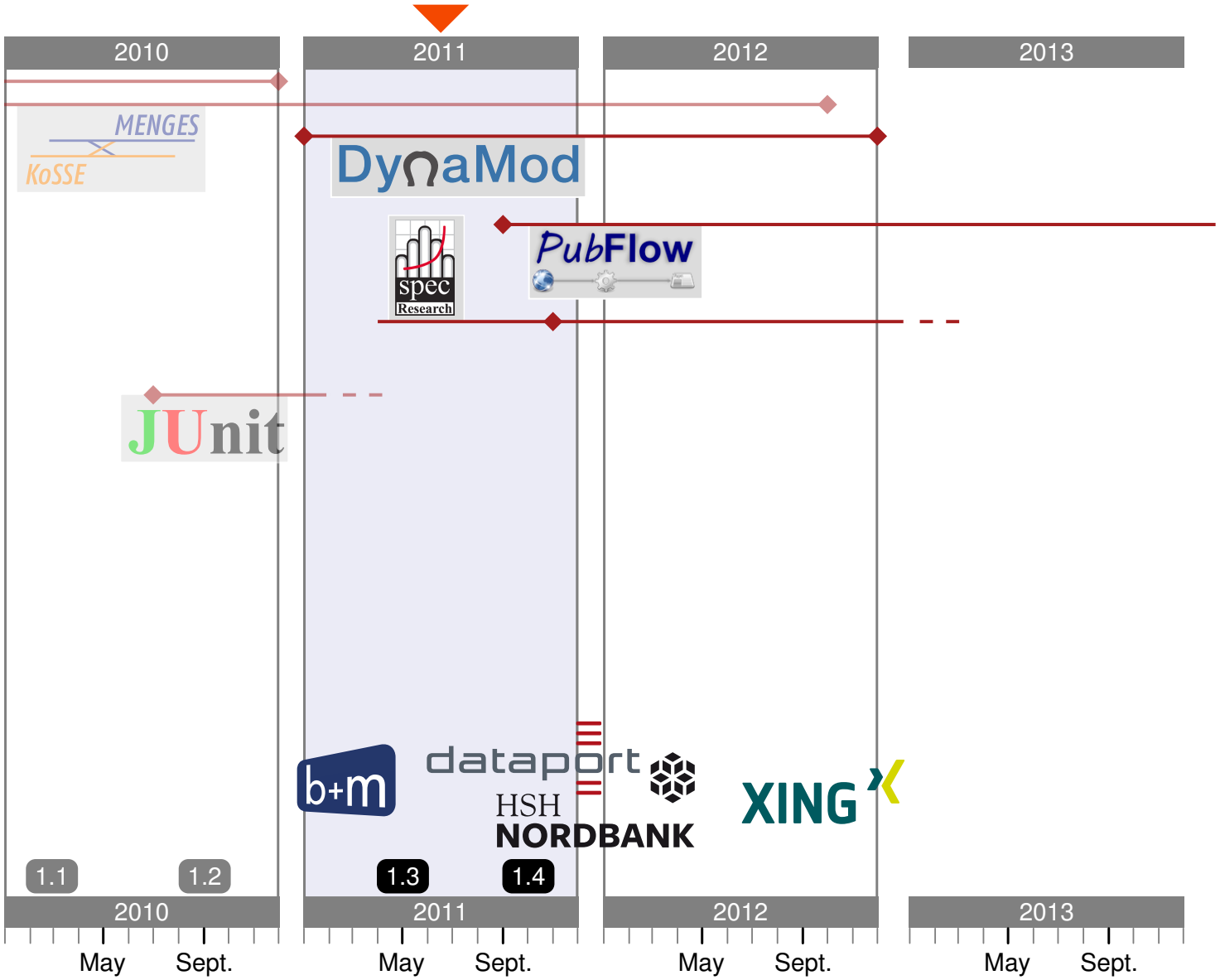
Who has been/is Working on Kieker?



Looking back . . . 2010–2013

Kieker

Who has been/is Working on Kieker?



Looking back . . . 2010–2013

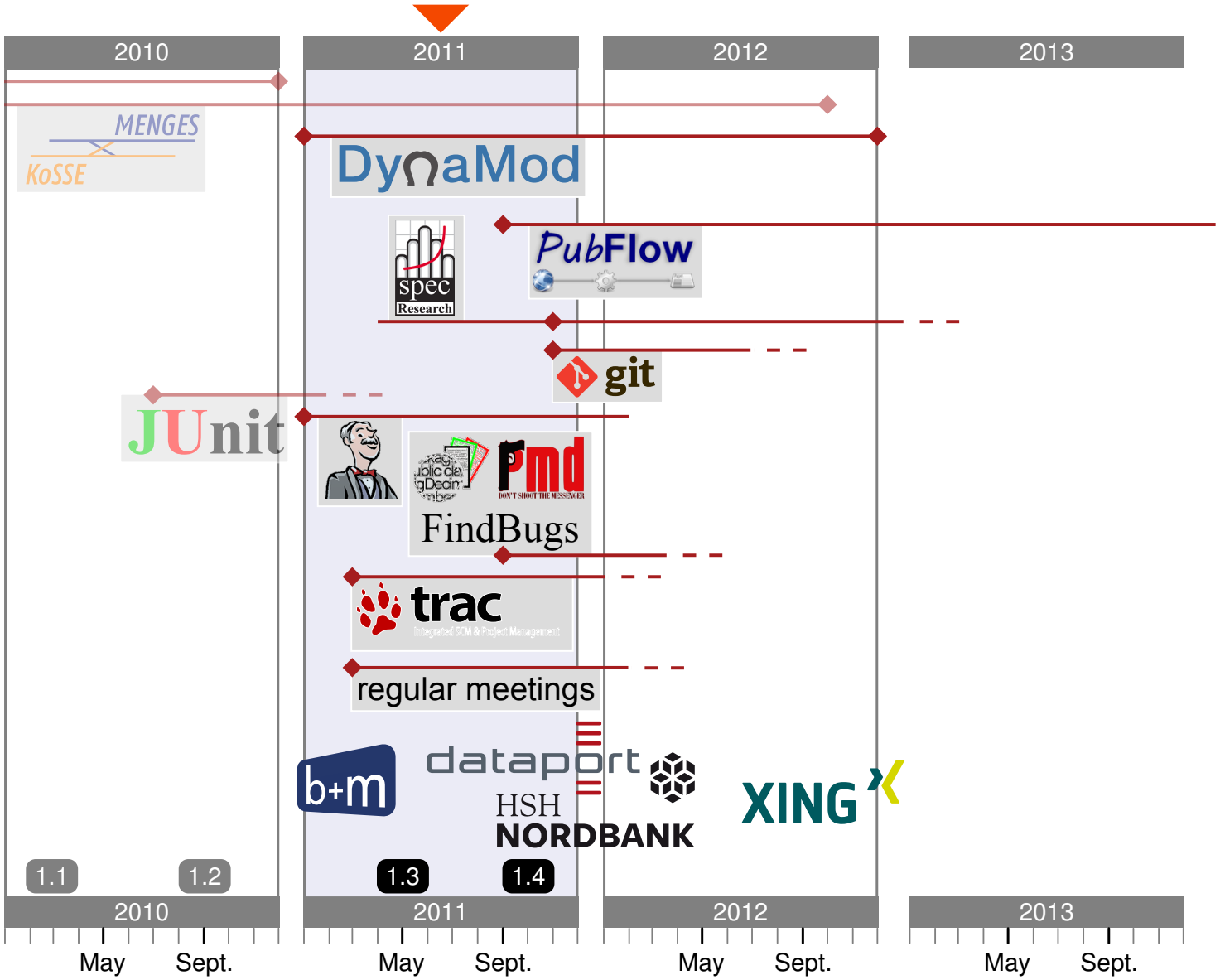
Kieker

Who has been/is Working on Kieker?

Looking back . . . 2010–2013

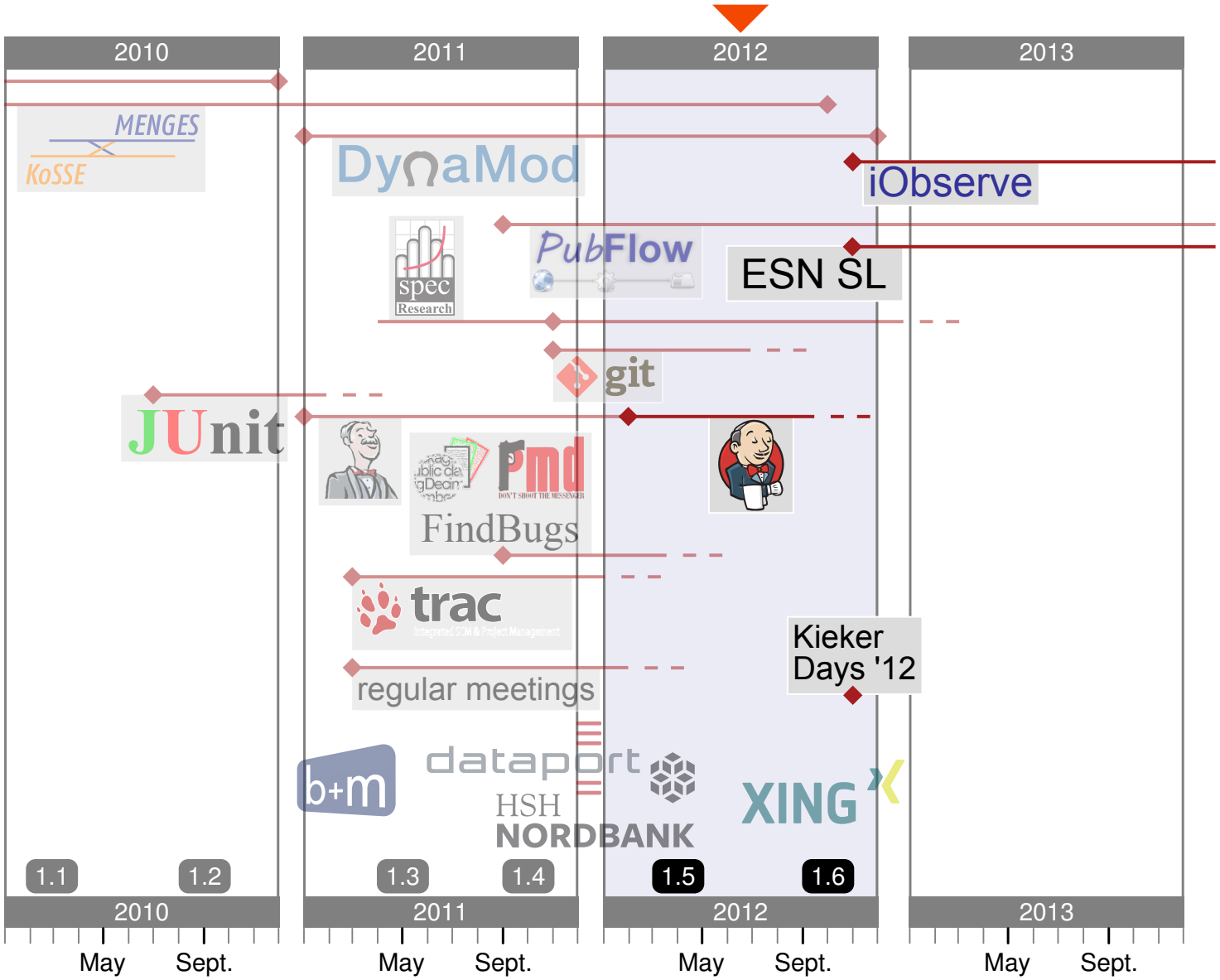
Kieker

Who has been/is Working on Kieker?



Looking back . . . 2010–2013

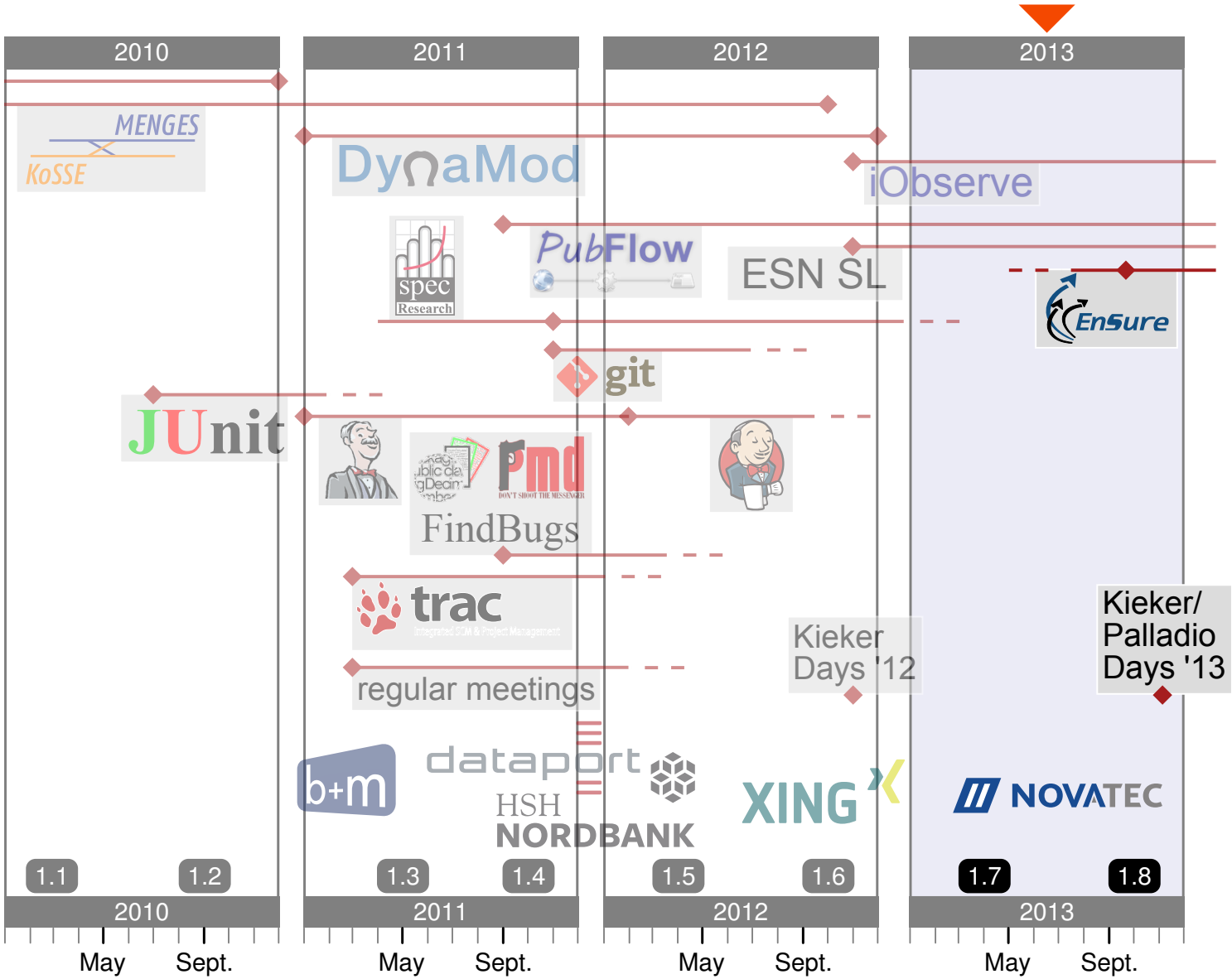
Who has been/is Working on Kieker?



Looking back . . . 2010–2013

Kieker

Who has been/is Working on Kieker?



For additional details on Kieker's evolution, see the Kieker Days '12 talk "Kieker: Overview, Review, and Outlook"

Thanks to All Contributors

Various people contributed to Kieker in the past years.

Tillmann (Till) Bielefeld, Peer Brauer, Philipp Döhring, Jens Ehlers, Nils Ehmke, Florian Fittkau, Thilo Focke, Sören Frey, Tom Frotscher, Wilhelm (Willi) Hasselbring, André van Hoorn, Reiner Jung, Benjamin Kiel, Dennis Kieselhorst, Holger Knoche, Arnd Lange, Marius Löwe, Marco Lübcke, Felix Magedanz, Nina Marwede, Robert von Massow, Jasminka Matevska, Oliver Preikszas, Sönke Reimer, Bettual Richter, Matthias Rohr, Nils Sommer, Lena Stöver, Jan Waller, Robin Weiß, Björn Weißenfels, Matthias Westphal, Christian Wulf

—Alphabetic list of people who contributed in different form (source code, bug reports, promotion, etc) and intensity

Visit *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`

Literature

T. C. Bielefeld. Online performance anomaly detection for large-scale software systems. http://eprints.uni-kiel.de/15488/, Mar. 2012. Diploma Thesis, Kiel University.

P. Döhring. Visualisierung von Synchronisationspunkten in Kombination mit der Statik und Dynamik eines Softwaresystems. Master's thesis, Kiel University, Oct. 2012. URL http://eprints.uni-kiel.de/15433/.

N. C. Ehmke. Everything in sight: Kieker's WebGUI in action (tutorial). In *Proceedings of the Symposium on Software Performance: Joint Kieker/Palladio Days (KPDAYS '13)*, 2013. Accepted for publication.

T. Focke. Performance Monitoring von Middleware-basierten Applikationen. Diplomarbeit, University Oldenburg, Mar. 2006.

T. Frotscher. Architecture-based multivariate anomaly detection for software systems, Oct. 2013. Master's Thesis, Kiel University.

Kieker Project. Kieker 1.8 user guide, Oct. 2013a. URL http://kieker-monitoring.net/documentation/.

Kieker Project. Kieker web site, 2013b. URL http://kieker-monitoring.net/.

F. Magedanz. Dynamic analysis of .NET applications for architecture-based model extraction and test generation. http://eprints.uni-kiel.de/15486/, Oct. 2011. Diploma Thesis, Kiel University.

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 A. Winter, R. Ferenc, and J. Knodel, editors, *Proceedings of the 13th European Conference on Software Maintenance and Reengineering (CSMR 2009)*, pages 47–57. IEEE Computer Society, Mar. 2009. ISBN 978-0-7695-3589-0. doi: 10.1109/CSMR.2009.15.

B. Richter. Dynamische Analyse von COBOL-Systemarchitekturen zum modellbasierten Testen ("Dynamic analysis of cobol system architectures for model-based testing", in German). http://eprints.uni-kiel.de/15489/, Aug. 2012. Diploma Thesis, Kiel University.

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. Technical Report TR-0921, Department of Computer Science, University of Kiel, Germany, Nov. 2009. URL http://www.informatik.uni-kiel.de/uploads/tx_publication/vanhoorn_tr0921.pdf.

A. van Hoorn, J. Waller, and W. Hasselbring. Kieker: A framework for application performance monitoring and dynamic software analysis. In *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering (ICPE 2012)*, pages 247–248. ACM, Apr. 2012. ISBN 978-1-4503-1202-8. doi: 10.1145/2188286.2188326.

C. Wulf. Runtime visualization of static and dynamic architectural views of a software system to identify performance problems, 2010.

For a comprehensive list of publications, talks, and theses about Kieker, visit:
http://kieker-monitoring.net/research/