Review and Update on Kieker

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

[1] Kiel University (CAU)
Software Engineering Group
&
[2] University of Stuttgart
Reliable Software Systems Group

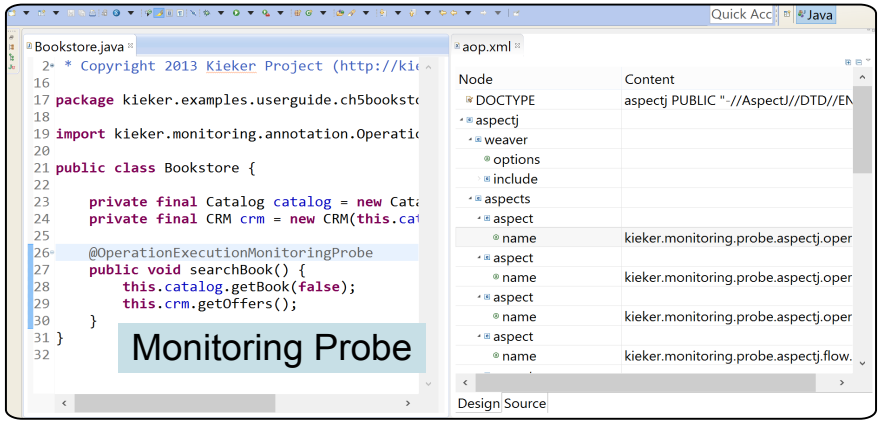
SSP 2015

05. November 2015 @ Munich

Dynamic Analysis with Kieker

[van Hoorn et al. 2012]

Overview

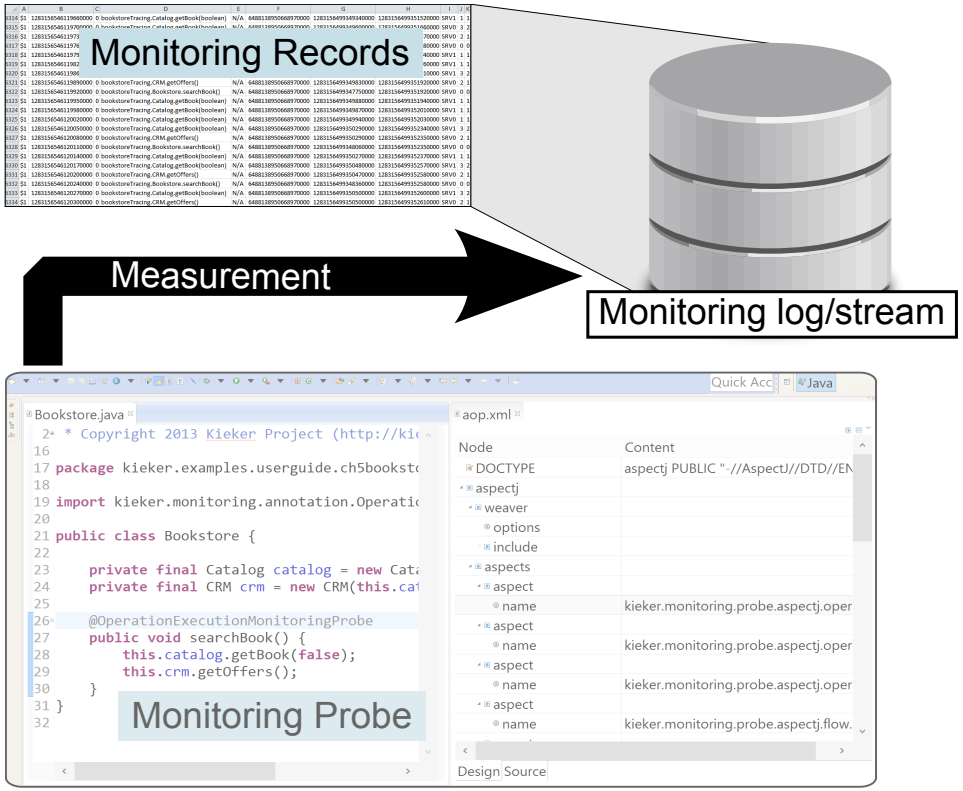


Software System with Monitoring Instrumentation

Dynamic Analysis with Kieker

[van Hoorn et al. 2012]

Overview

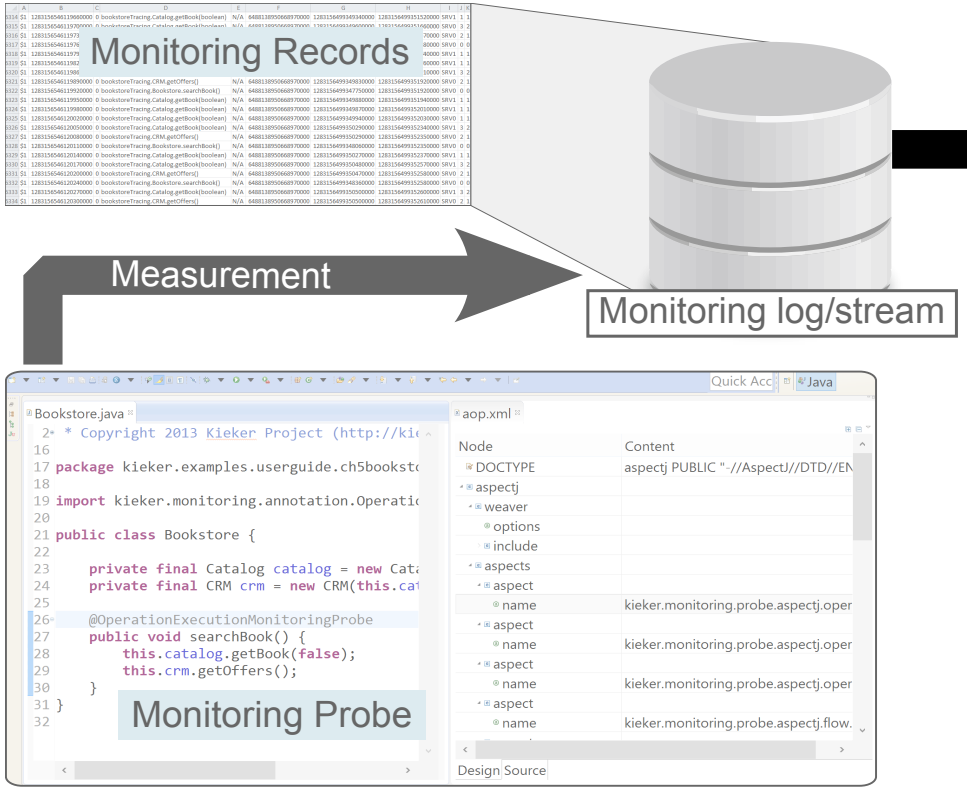


Software System with Monitoring Instrumentation

Dynamic Analysis with Kieker

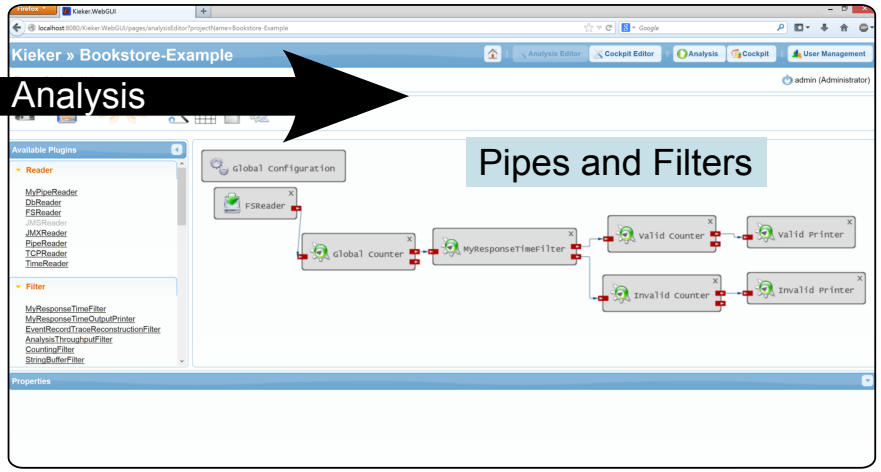
[van Hoorn et al. 2012]

Overview



Software System with Monitoring Instrumentation

Analysis Configuration (via API and WebGUI)

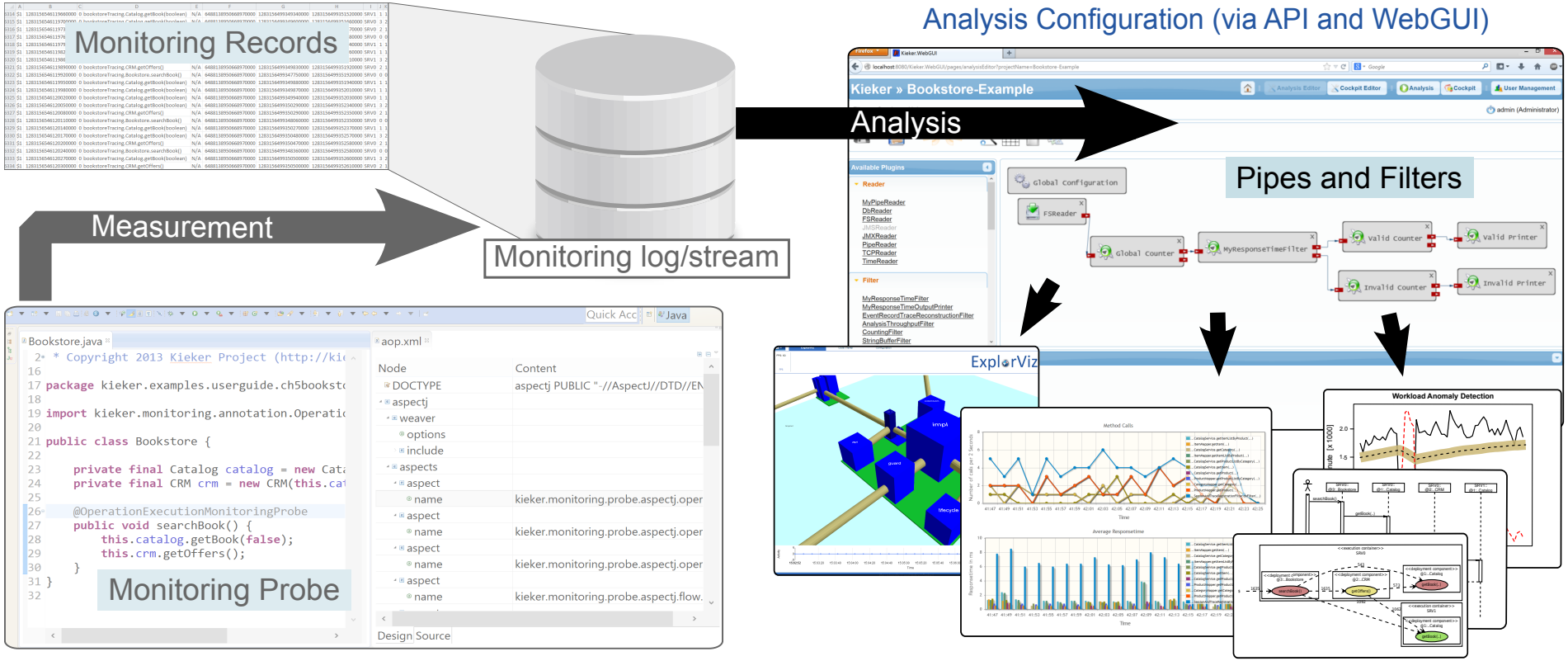


Dynamic Analysis with Kieker

[van Hoorn et al. 2012]

Overview

Christian-Albrechts-Universität zu Kiel



Kieker

- Modular, flexible, and extensible architecture (Probes, records, readers, writers, filters etc.)
- Pipes-and-filters framework for analysis configuration
- Distributed tracing (logging, reconstruction, visualization)
- Low overhead (designed for continuous operation)
- Evaluated in lab and industrial case studies

b+m business IT management · cewe color · dataport · EWE · HSH NORDBANK · XING

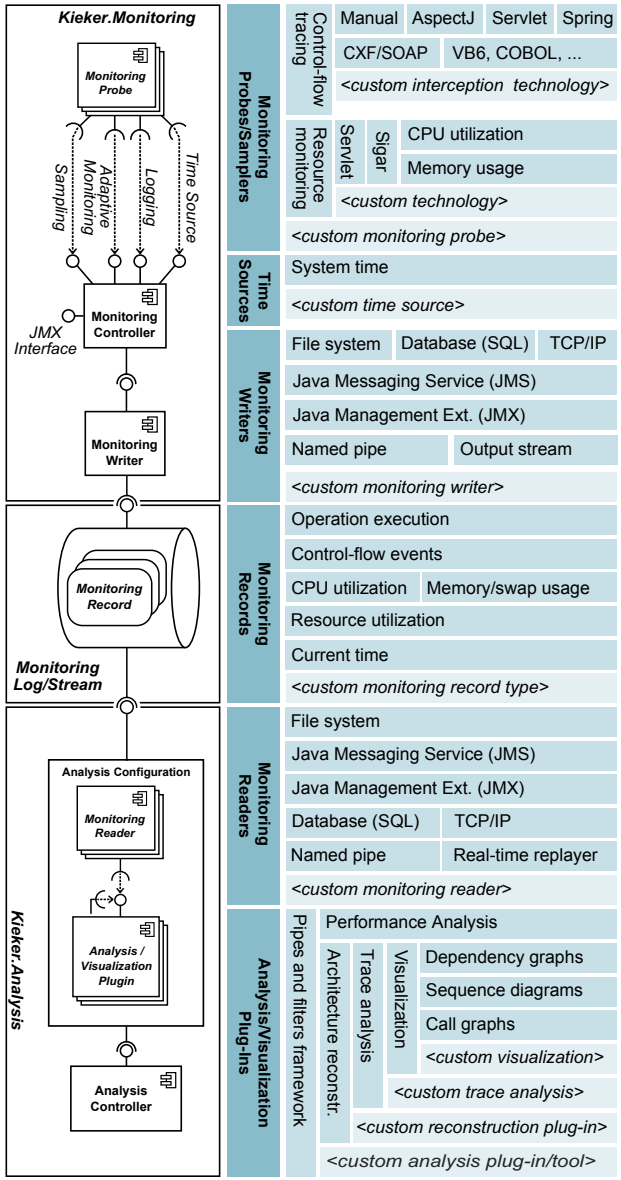
Kieker is open-source software (Apache License, V. 2.0)

`http://kieker-monitoring.net`

Recommended Tool of the SPEC Research Group

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`

spec Research



Agenda

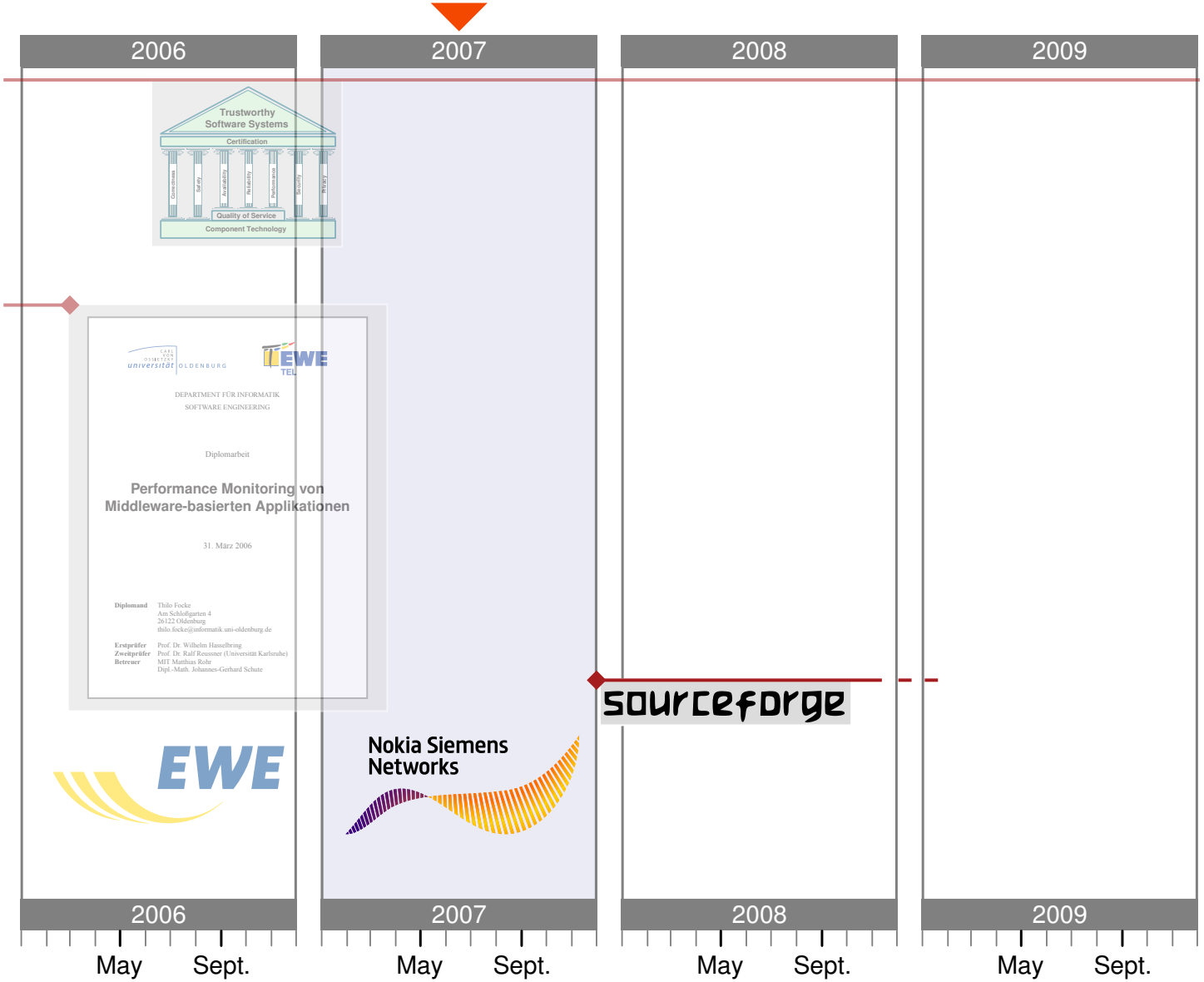
Christian-Albrechts-Universität zu Kiel

Looking back . . . 2006–2009

Review

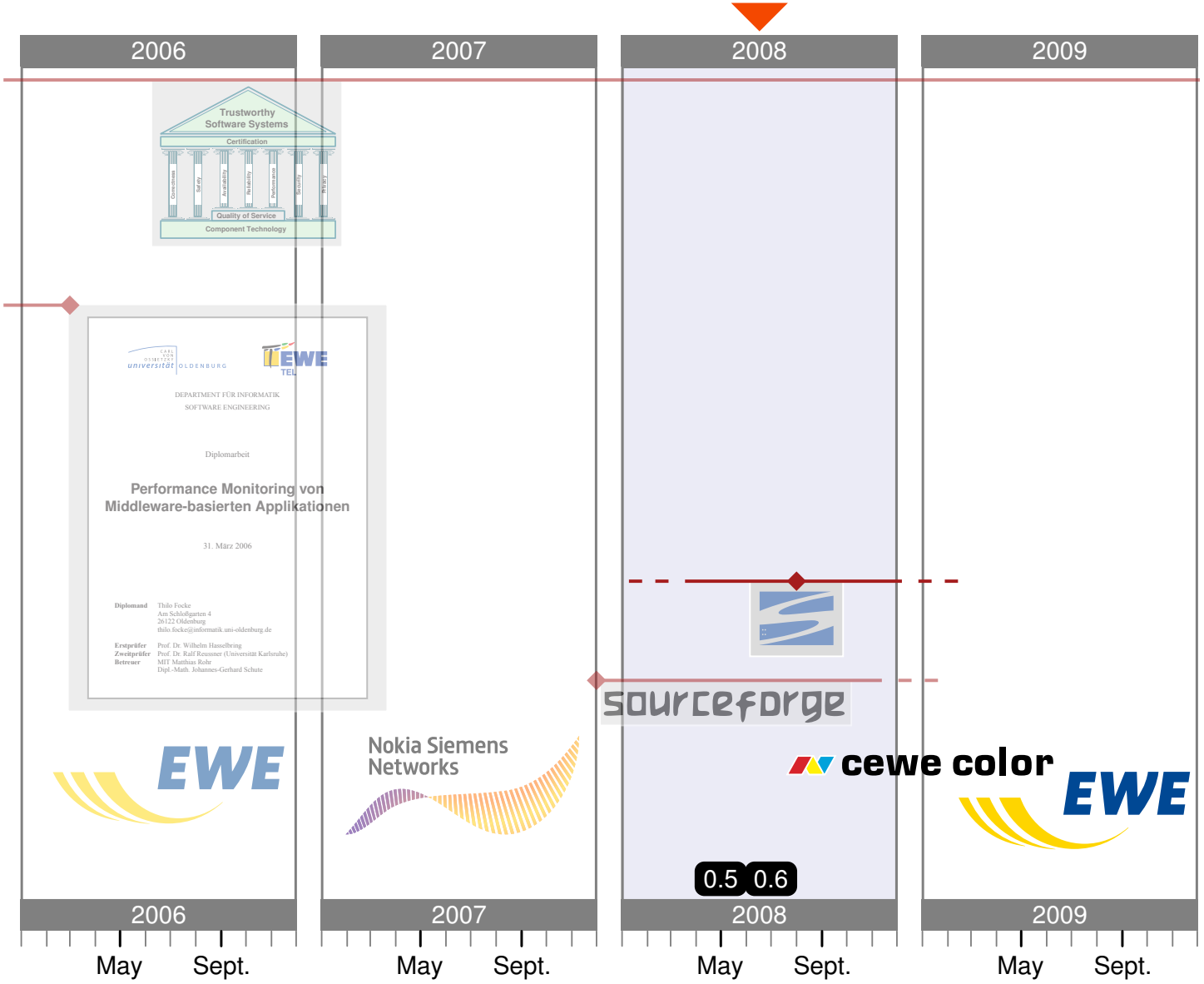
Looking back . . . 2006–2009

Review



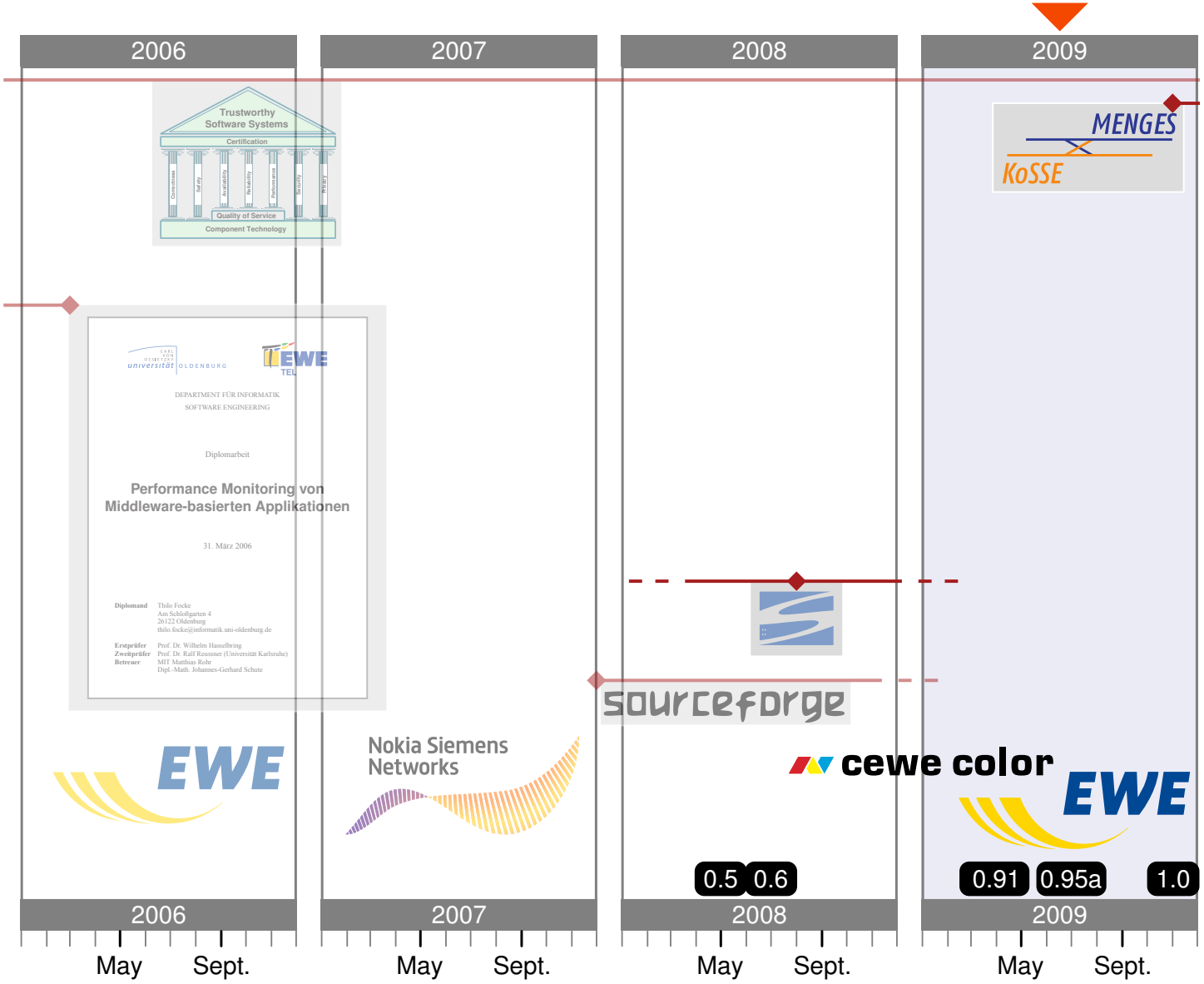
Looking back . . . 2006–2009

Review



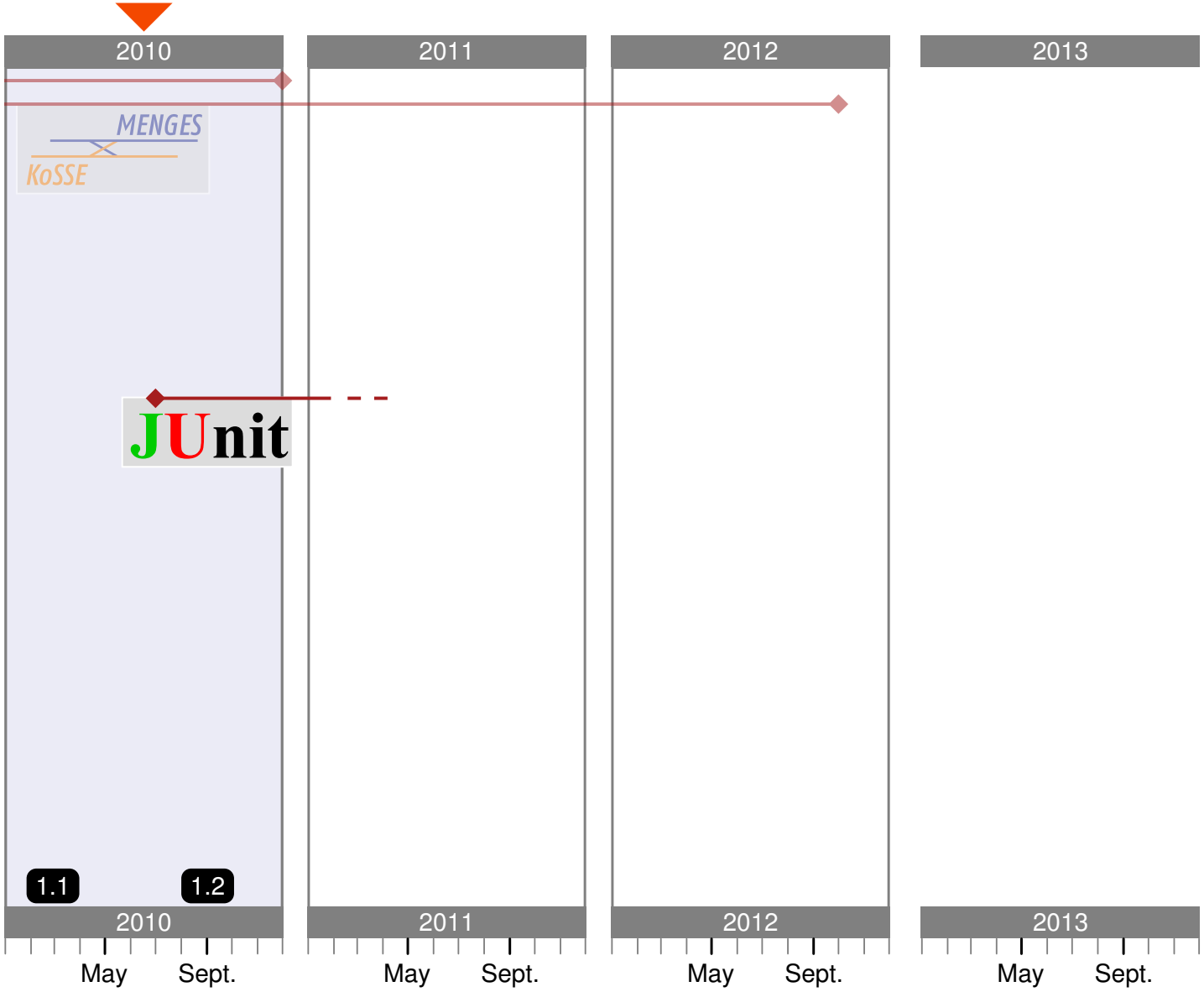
Looking back . . . 2010–2013

Review



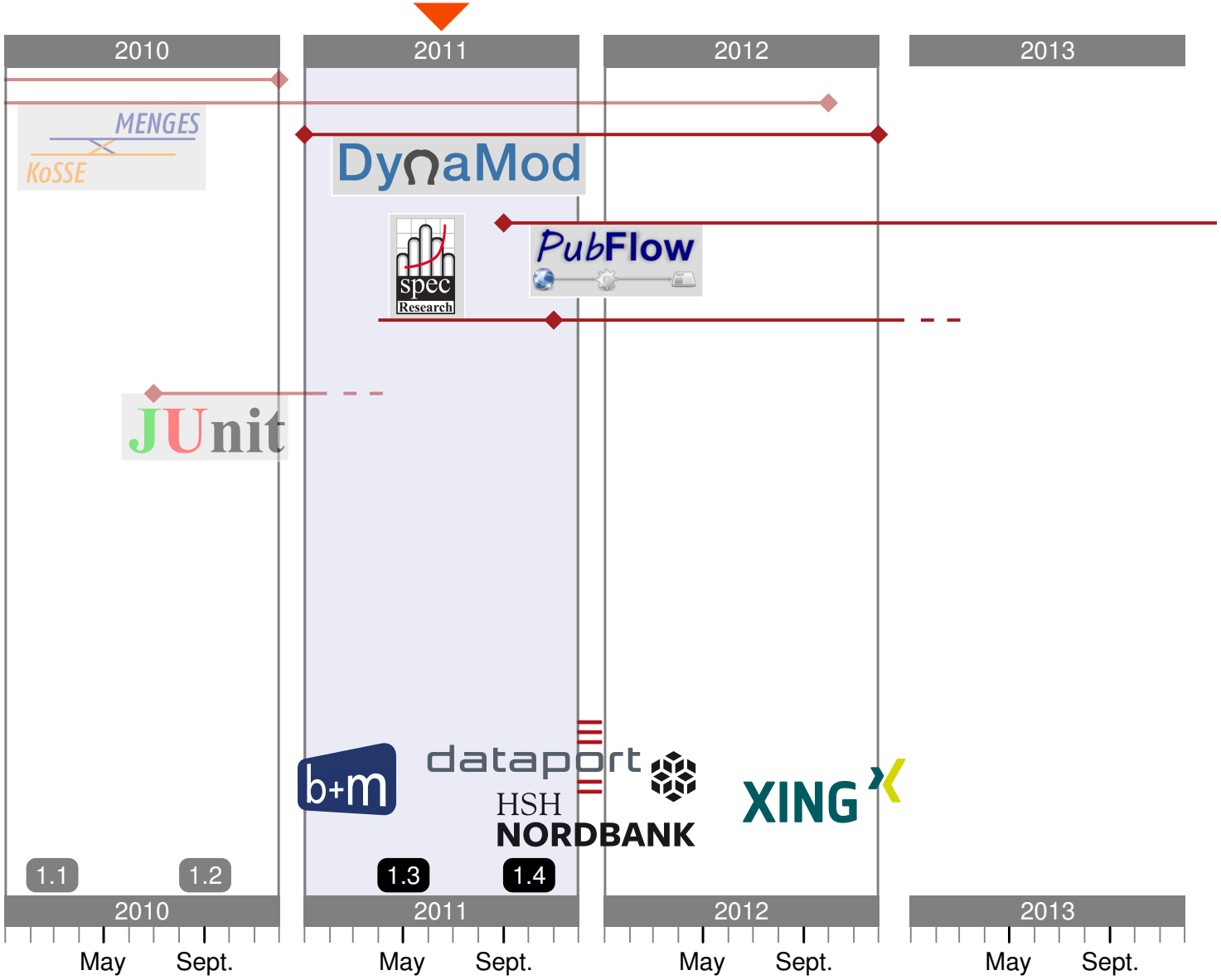
Looking back . . . 2010–2013

Review



Looking back . . . 2010–2013

Review

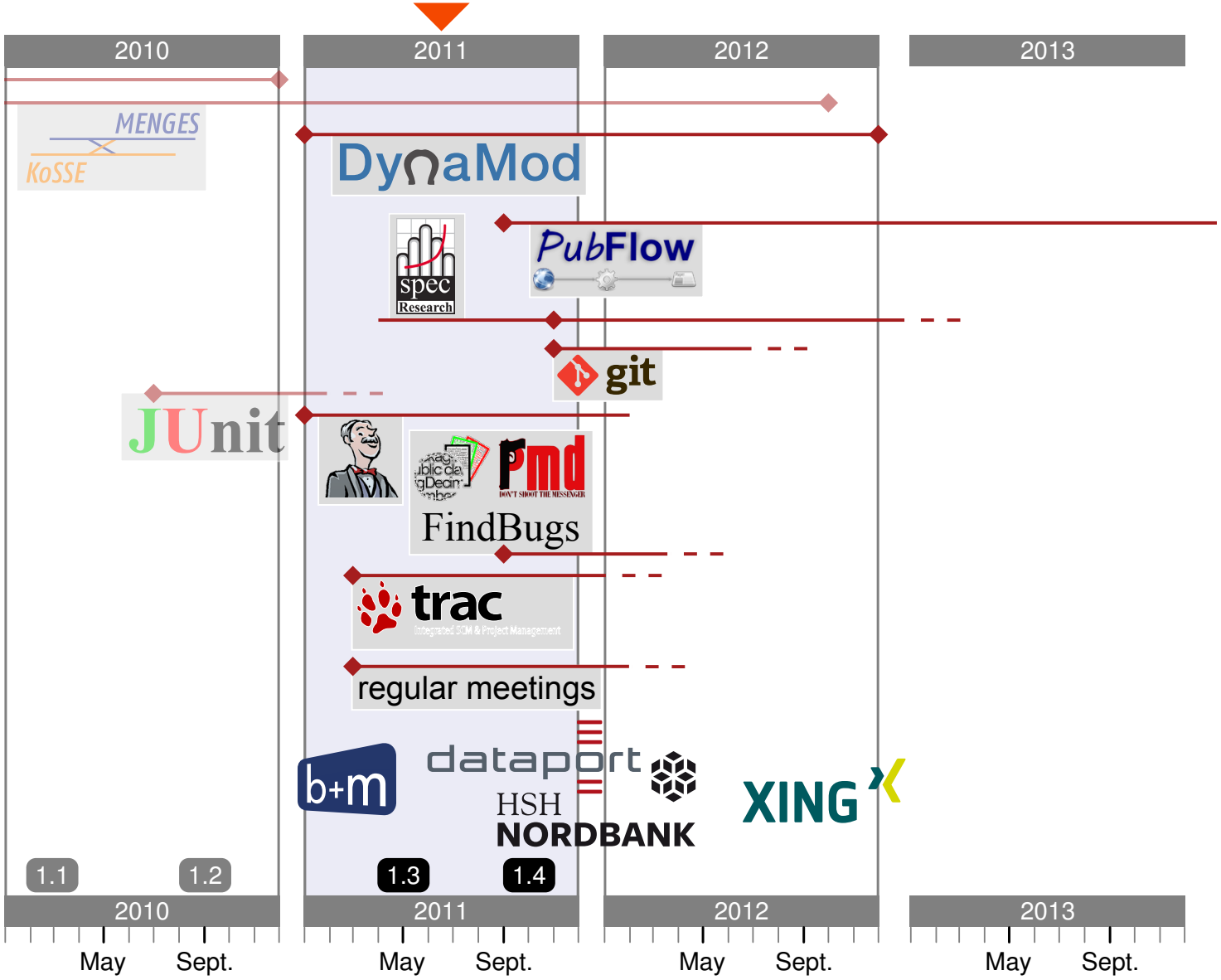


Looking back . . . 2010–2013

Review

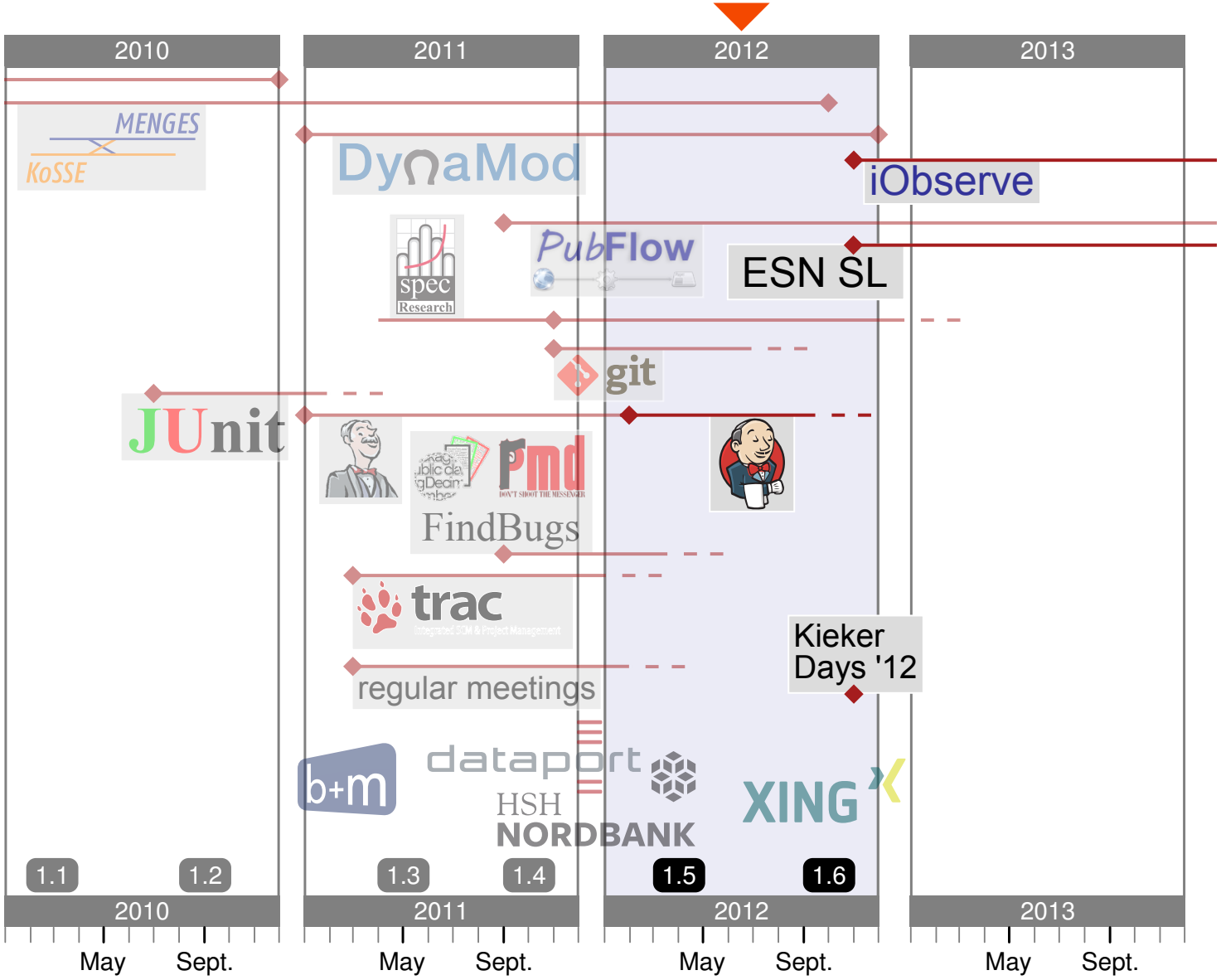
Looking back . . . 2010–2013

Review



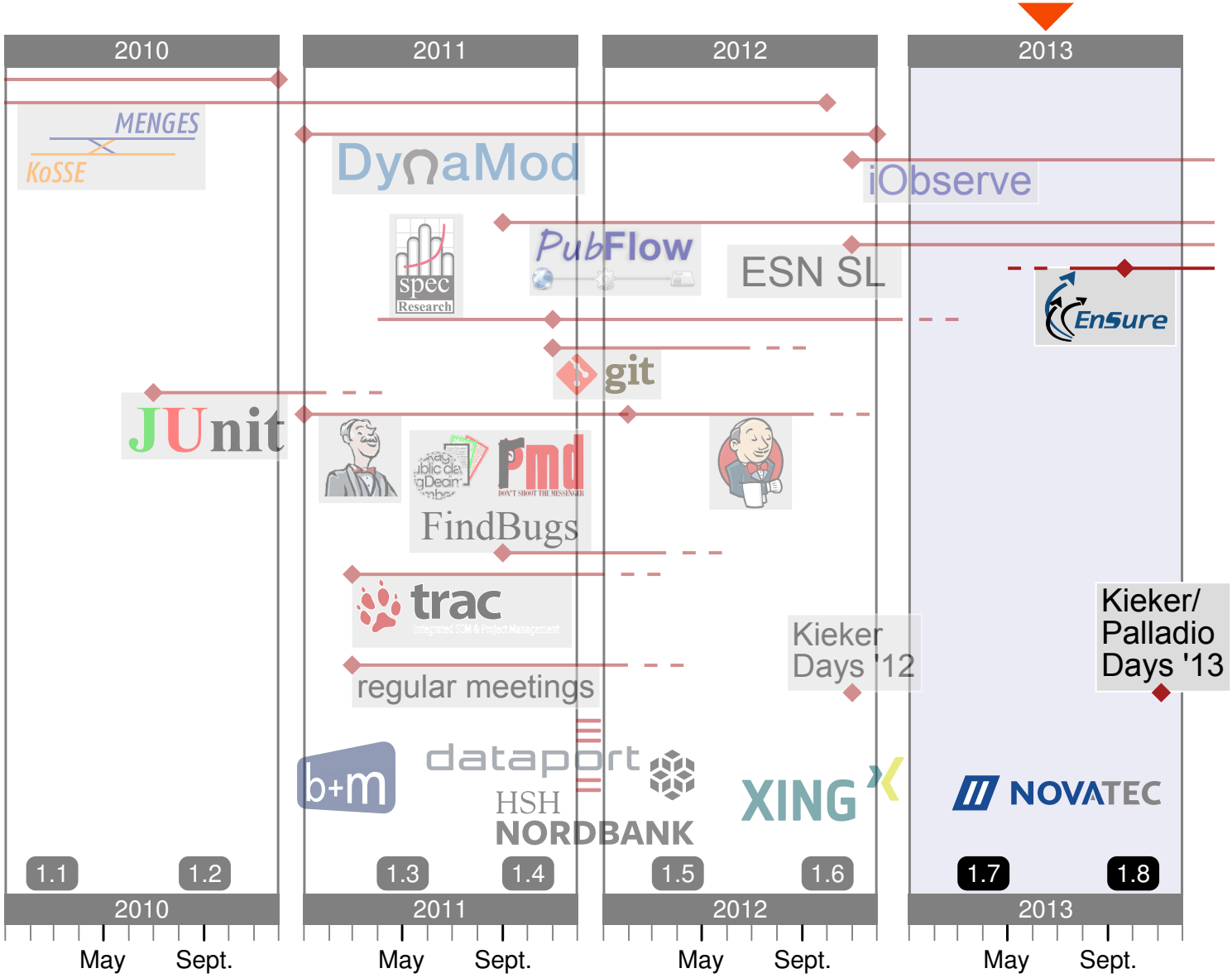
Looking back . . . 2010–2013

Review

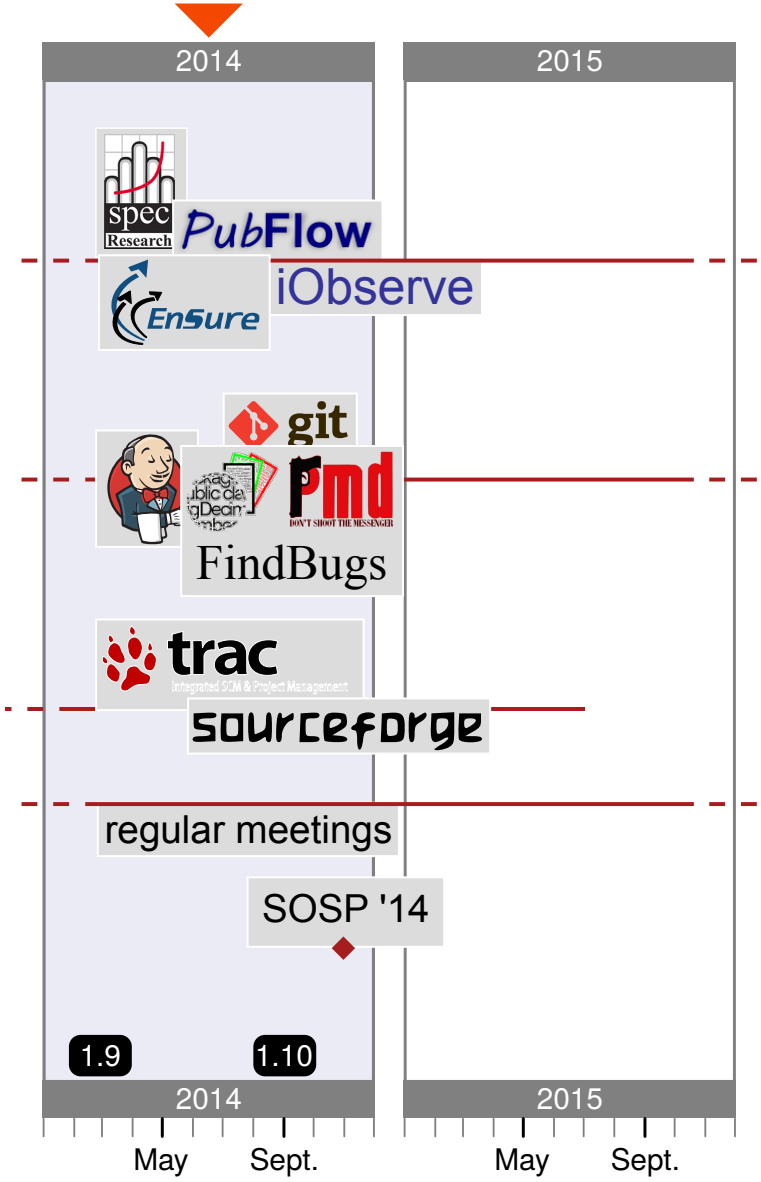


Looking back . . . 2010–2013

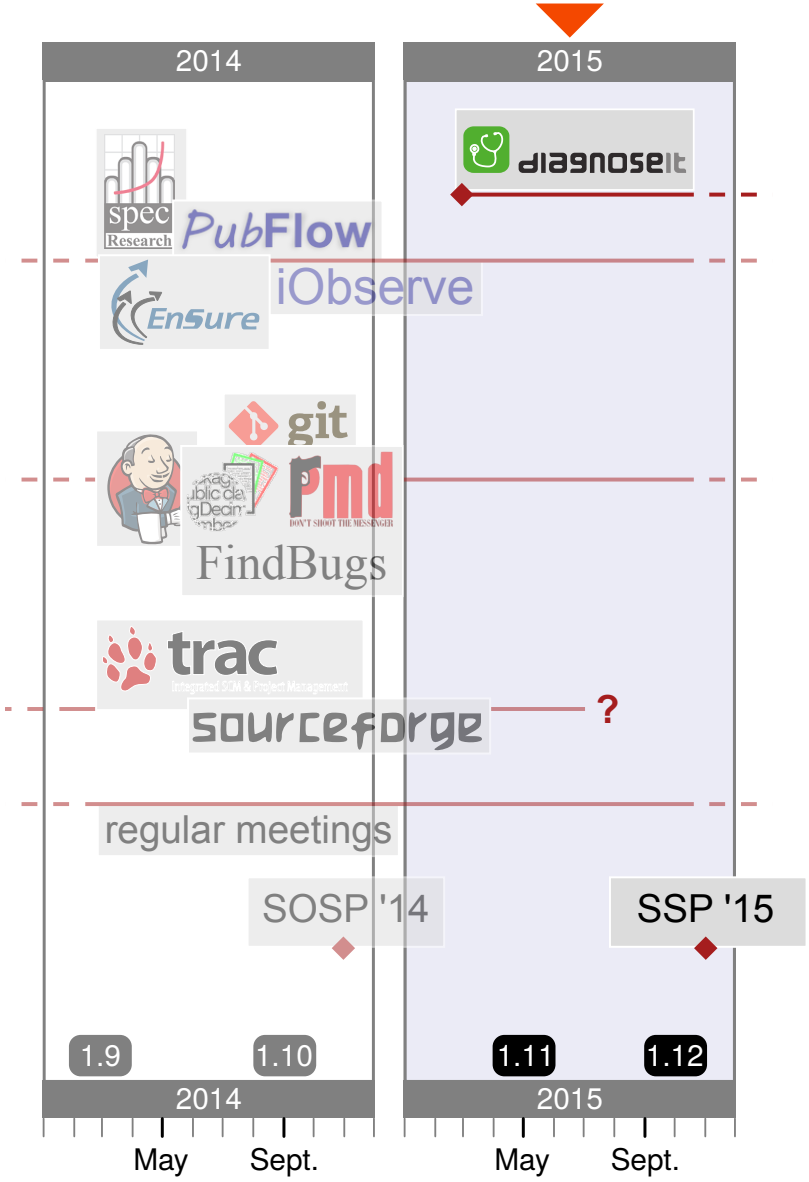
Review



Looking back . . . **2014–2015**



Looking back . . . 2014–2015



Open-Source Software as Catalyzer for Technology Transfer: Kieker's Development and Lessons Learned

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

[1] Kiel University, Department of Computer Science, 24118 Kiel, Germany
[2] University of Stuttgart, Institute of Software Technology, 70569 Stuttgart, Germany

Abstract: The monitoring framework Kieker commenced as a joint diploma thesis of the University of Oldenburg and a telecommunication provider in 2006, and grew toward a high-quality open-source project during the last years. Meanwhile, Kieker has been and is employed in various projects. Several research groups constitute the open-source community to advance the Kieker framework. In this paper, we review Kieker's history, development, and impact as catalyzer for technology transfer.

1 Introduction

The development of tools is common practice for researchers in order to demonstrate the practicality of developed research approaches and to qualitatively and quantitatively evaluate their research results. During the last years, there is an increasing trend that researchers make their tools publicly available under an open-source license, e.g., allowing a more thorough evaluation of work presented in research papers, as well as easing reproducibility of results and building on the work of others. The state of these tools ranges from proof-of-concept implementations to full-blown products. Popular examples of wide-spread and mature open-source tools originally developed and maintained by researchers include the probabilistic model checker PRISM [KNP11] and the R language and environment for statistical computing [R D08].

Since 2006, we have been developing the Kieker framework for dynamic analysis of software systems.[1] In this paper, we review Kieker's history, development, and impact as catalyzer for technology transfer. Parts of this paper have been published in a PhD dissertation [vH14, Chapter 15], which also includes a more detailed description of the framework (in addition to [RvHM+08, vHRH+09, vHWH12]) as well as its development process and infrastructure.

2 Kieker's Development and Impact

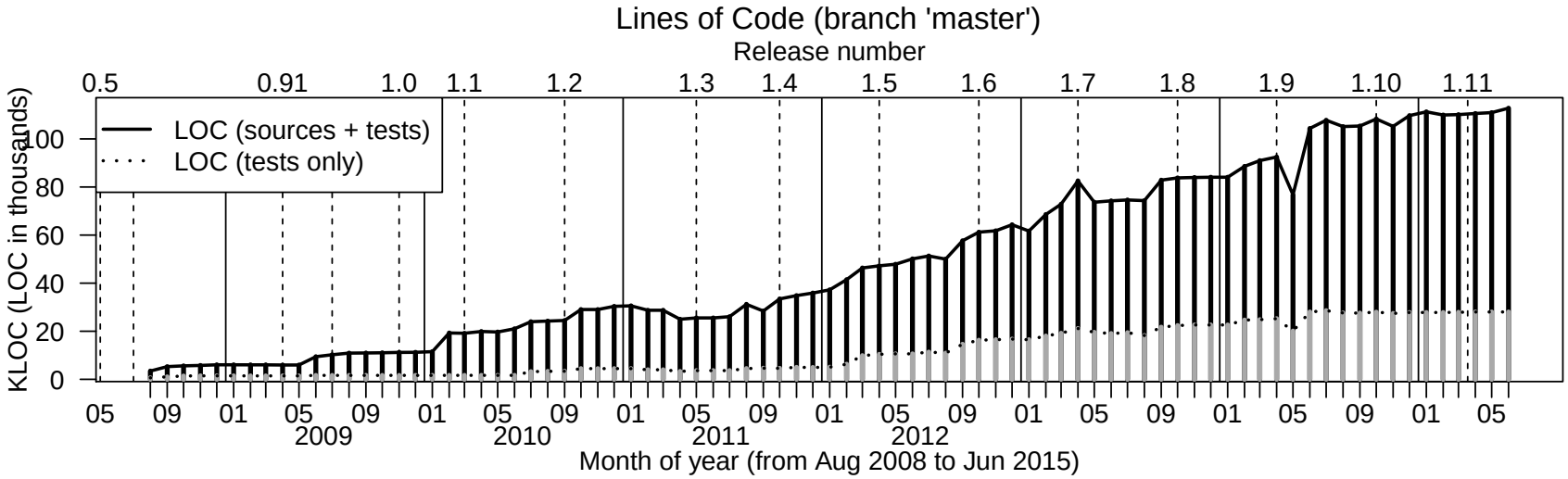
This section reviews the past years of Kieker development and gives some indication of the impact in terms of where and by whom Kieker has been developed and used.

[1] The Kieker framework's web site—including downloads, documentation, publications, and references—is available at http://kieker-monitoring.net

Evolution of Code Size and Downloads

LOC obtained via `wc -l <file>.java`

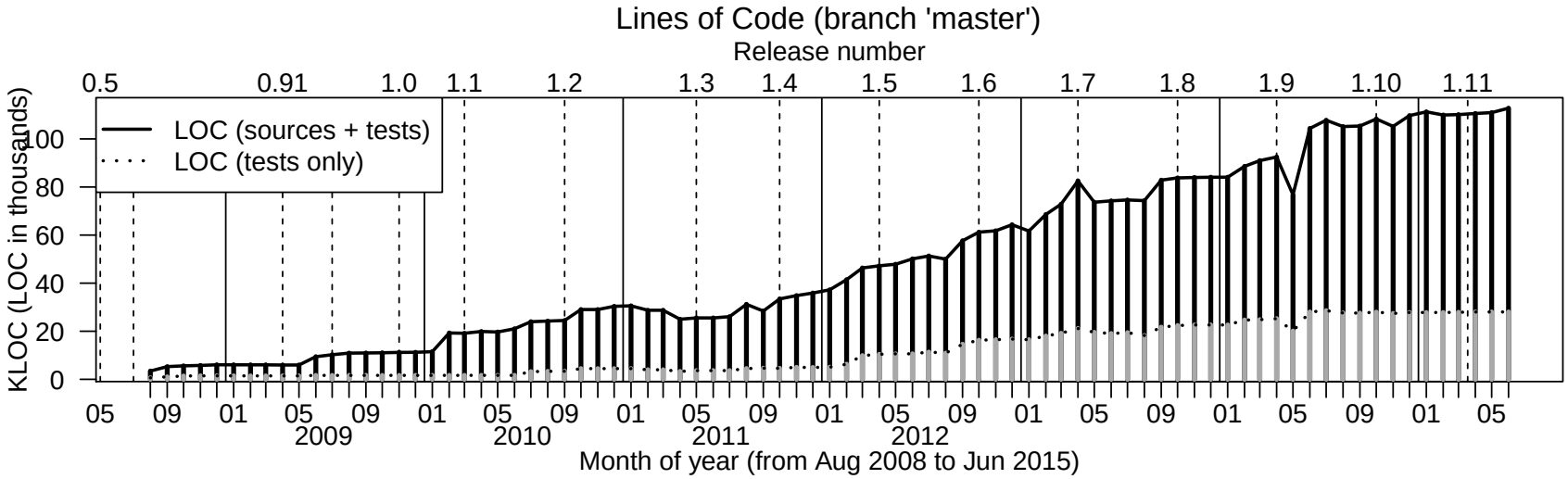
Review

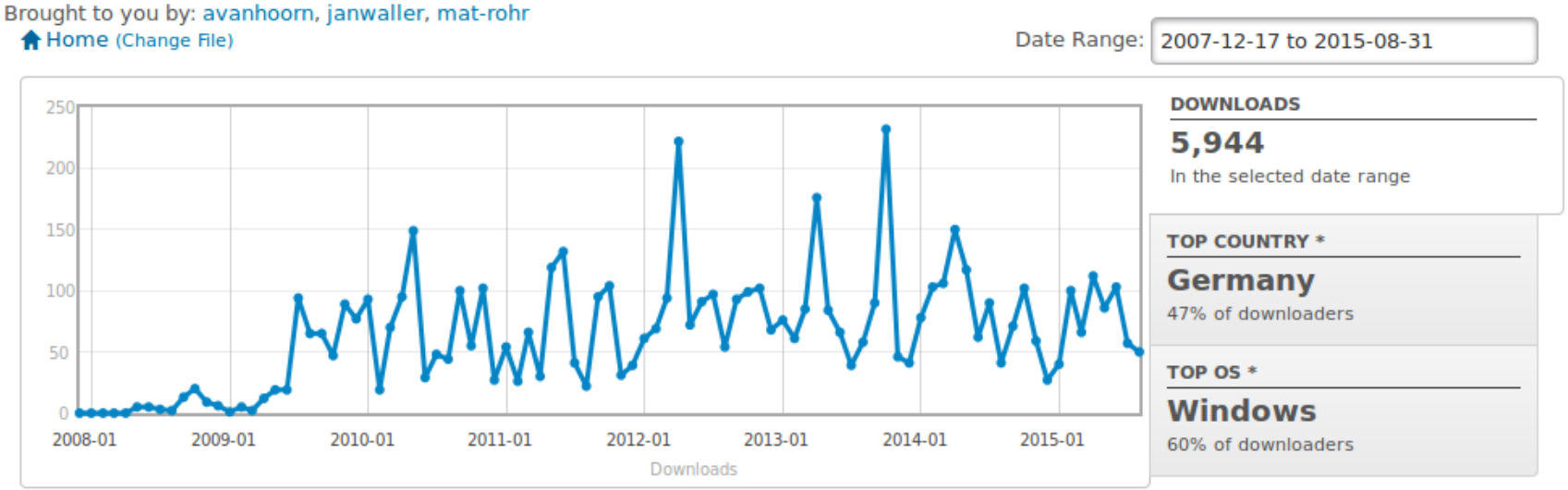


Evolution of Code Size and Downloads

LOC obtained via `wc -l <file>.java`

Review





Agenda

Current Activities (Selection) toward Kieker 2.0

- New high-throughput infrastructure for Kieker Analysis
  - Based on TeeTime [Wulf et al. 2014], `http://teetime.sf.net`
- Kieker Trace Diagnosis
  - Trace diagnosis tool to identify typical performance problems
- Eclipse plugin for profiling
- Docker-based Kieker example (NetflixOSS RSS reader application)
- Interoperability with other monitoring tools via a common trace API
- Analysis of Kieker development process and infrastructure

Ticket System: Current/Upcoming Issues

`http://trac.kieker-monitoring.net`

Literature

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

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

M. Rohr, A. van Hoorn, J. Matevska, N. Sommer, L. Stoever, S. Giesecke, and W. Hasselbring. Kieker: Continuous monitoring and on demand visualization of Java software behavior. In C. Pahl, editor, *Proceedings of the IASTED International Conference on Software Engineering 2008 (SE'08)*, pages 80–85, Feb. 2008.

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.

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.

C. Wulf, N. C. Ehmke, and W. Hasselbring. Toward a generic and concurrency-aware pipes & filters framework. In *Symposium on Software Performance 2014: Joint Descartes/Kieker/Palladio Days*, pages 70–82, Nov. 2014. URL `http://eprints.uni-kiel.de/27382/`.