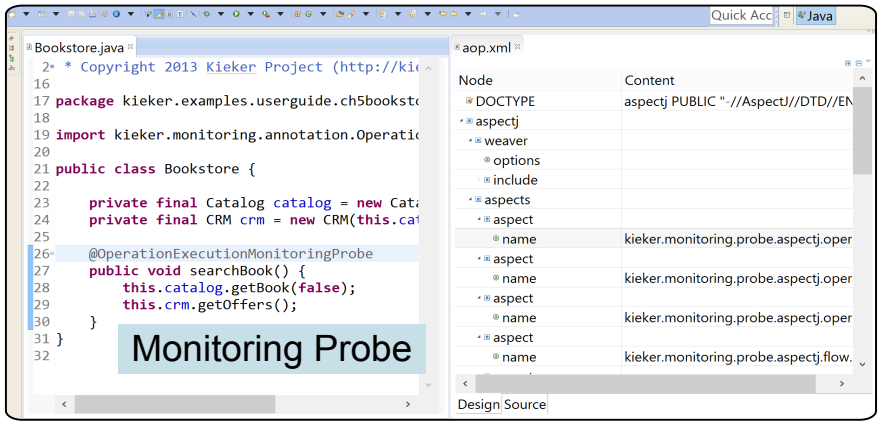
The Kieker Analysis Framework & Kieker's WebGUI

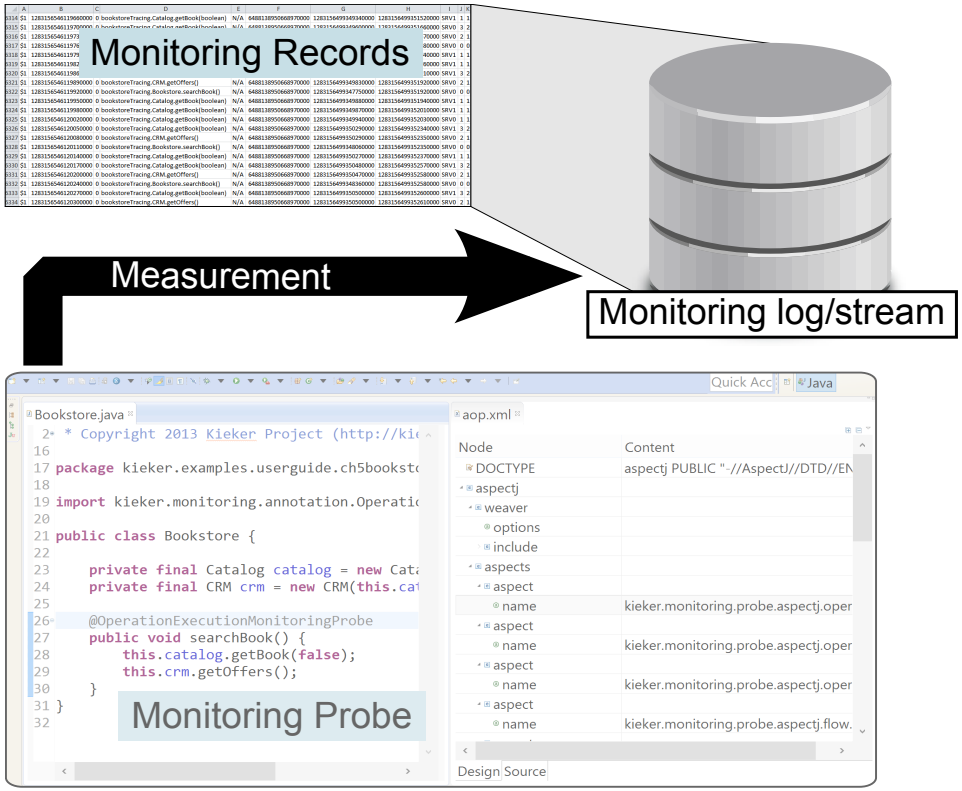
Nils Christian Ehmke, Christian Wulf, and Wilhelm Hasselbring

Software Engineering Group
Kiel University, Germany

November 07, 2014 @ b+m, Melsdorf



Software System with Monitoring Instrumentation

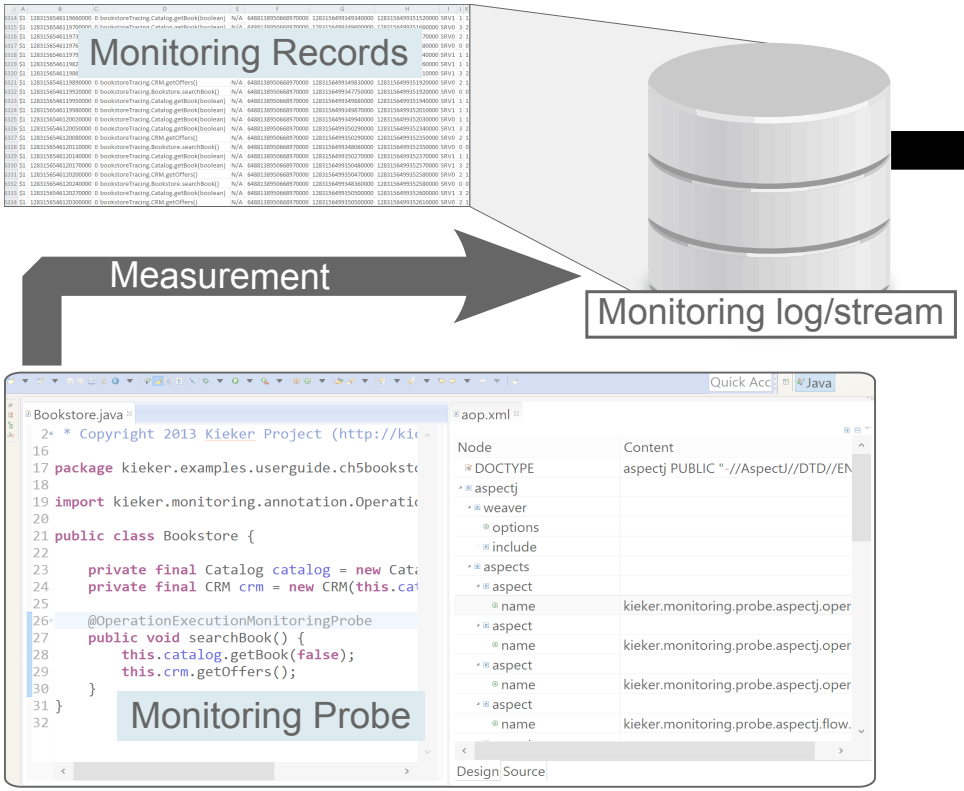


Software System with Monitoring Instrumentation

Dynamic Analysis with Kieker

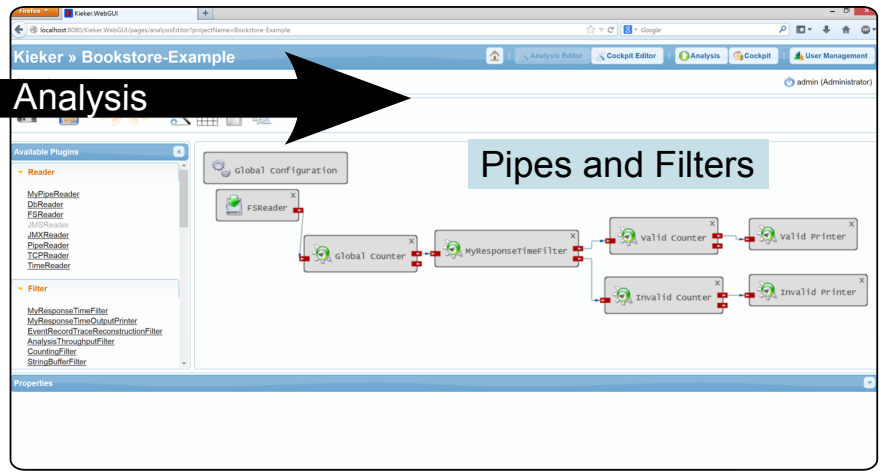
Overview

Christian-Albrechts-Universität zu Kiel



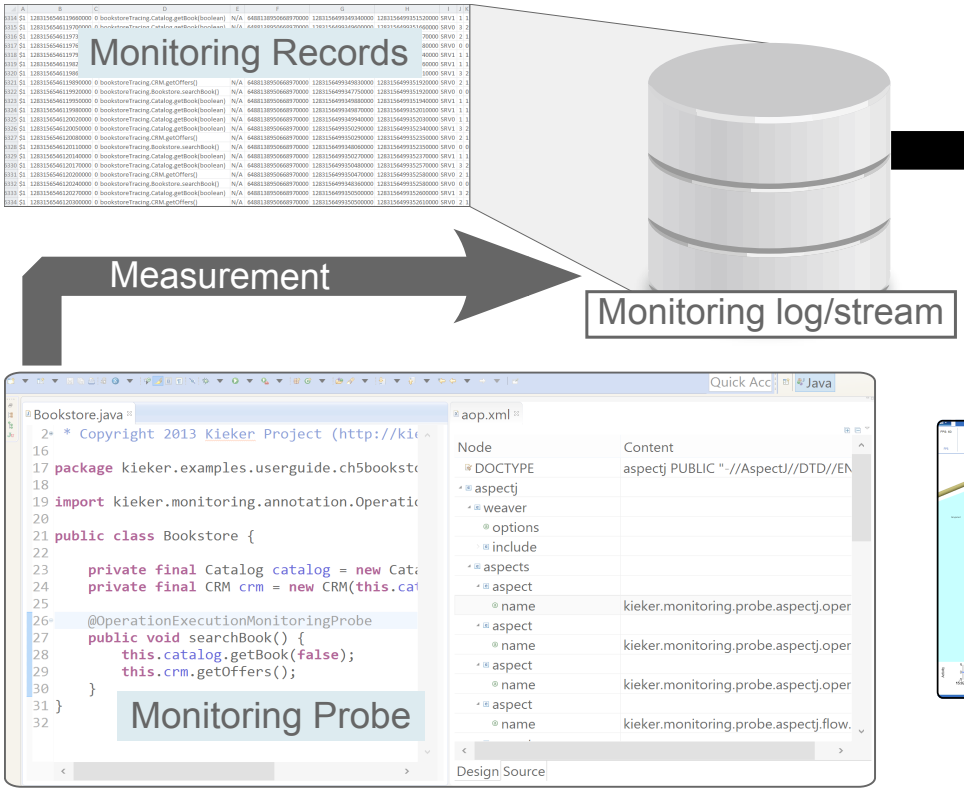
Software System with Monitoring Instrumentation

Analysis Configuration (via API and WebGUI)



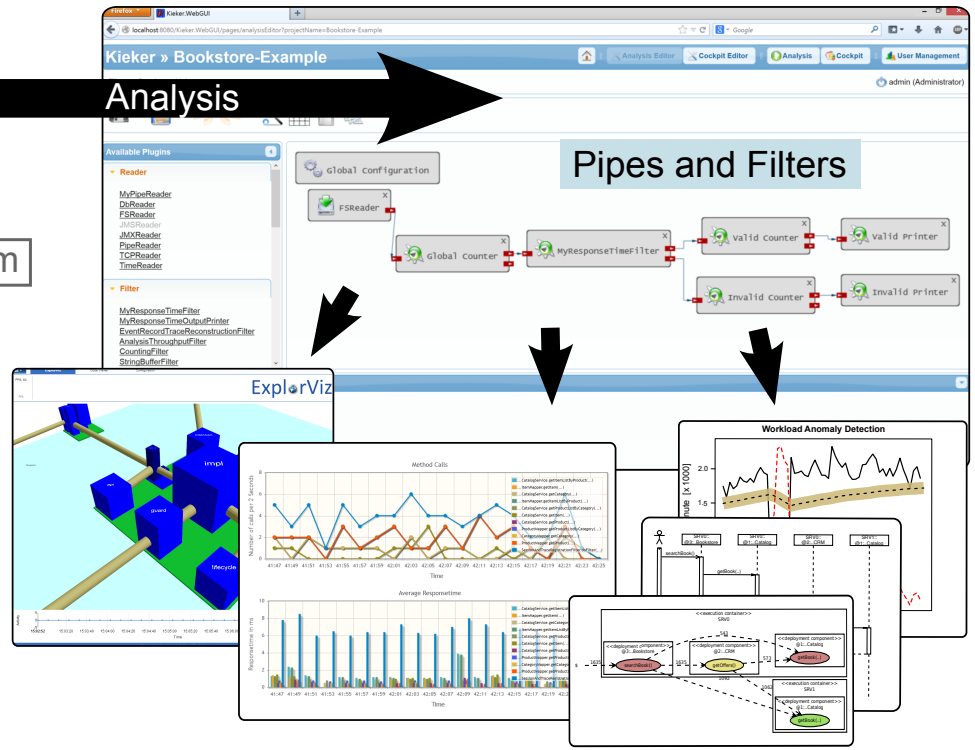
Dynamic Analysis with Kieker

Overview



Software System with Monitoring Instrumentation

Analysis Configuration (via API and WebGUI)

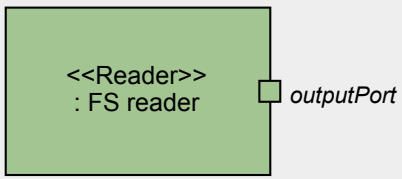


Online and Offline Visualization

Agenda

Example Pipe-and-Filter Configuration

Christian-Albrechts-Universität zu Kiel

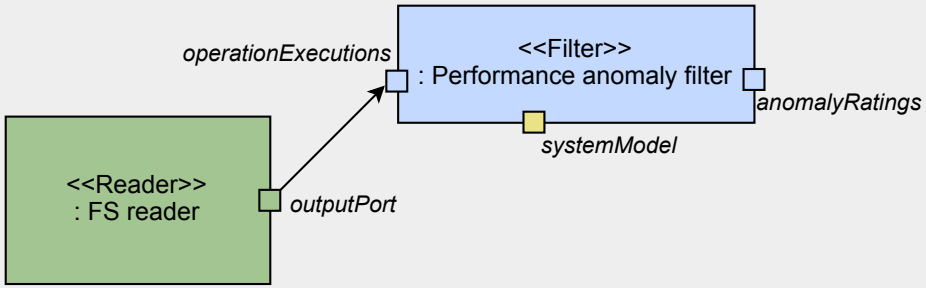


Kieker.Analysis example pipes-and-filters configuration

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

Example Pipe-and-Filter Configuration

Christian-Albrechts-Universität zu Kiel

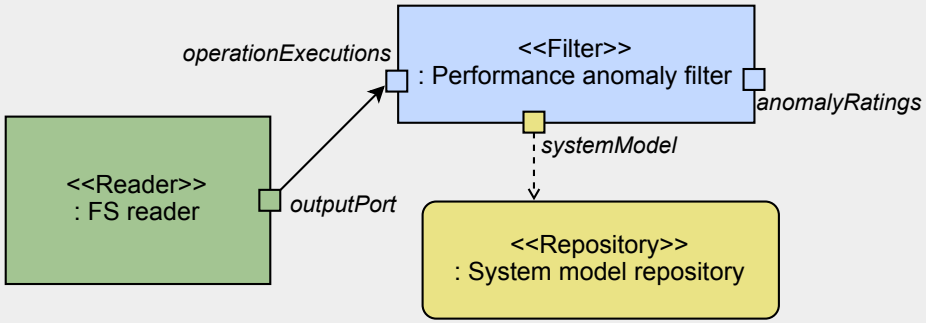


Kieker.Analysis example pipes-and-filters configuration

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

Example Pipe-and-Filter Configuration

Christian-Albrechts-Universität zu Kiel

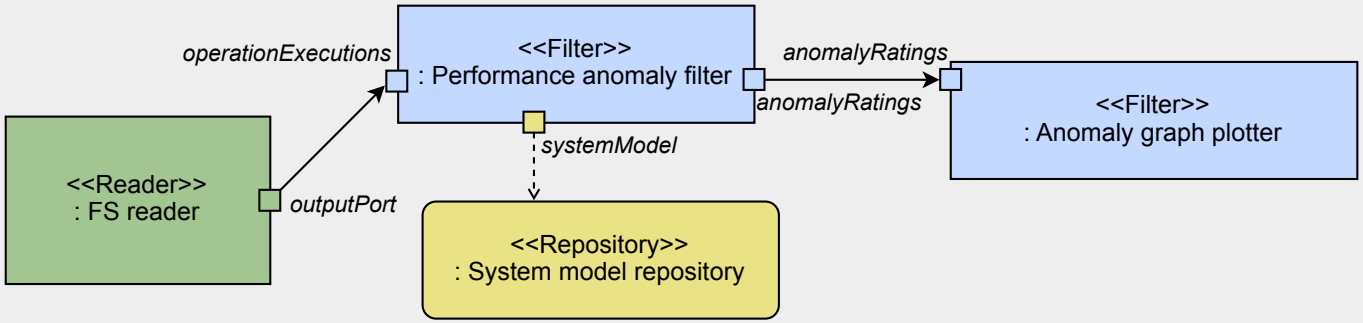


Kieker.Analysis example pipes-and-filters configuration

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

Example Pipe-and-Filter Configuration

Christian-Albrechts-Universität zu Kiel

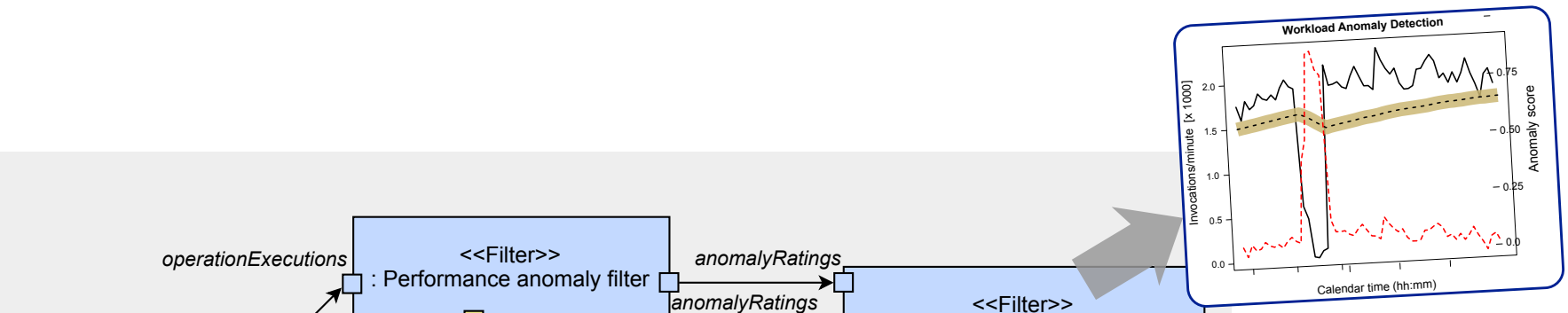


Kieker.Analysis example pipes-and-filters configuration

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

Example Pipe-and-Filter Configuration

Christian-Albrechts-Universität zu Kiel

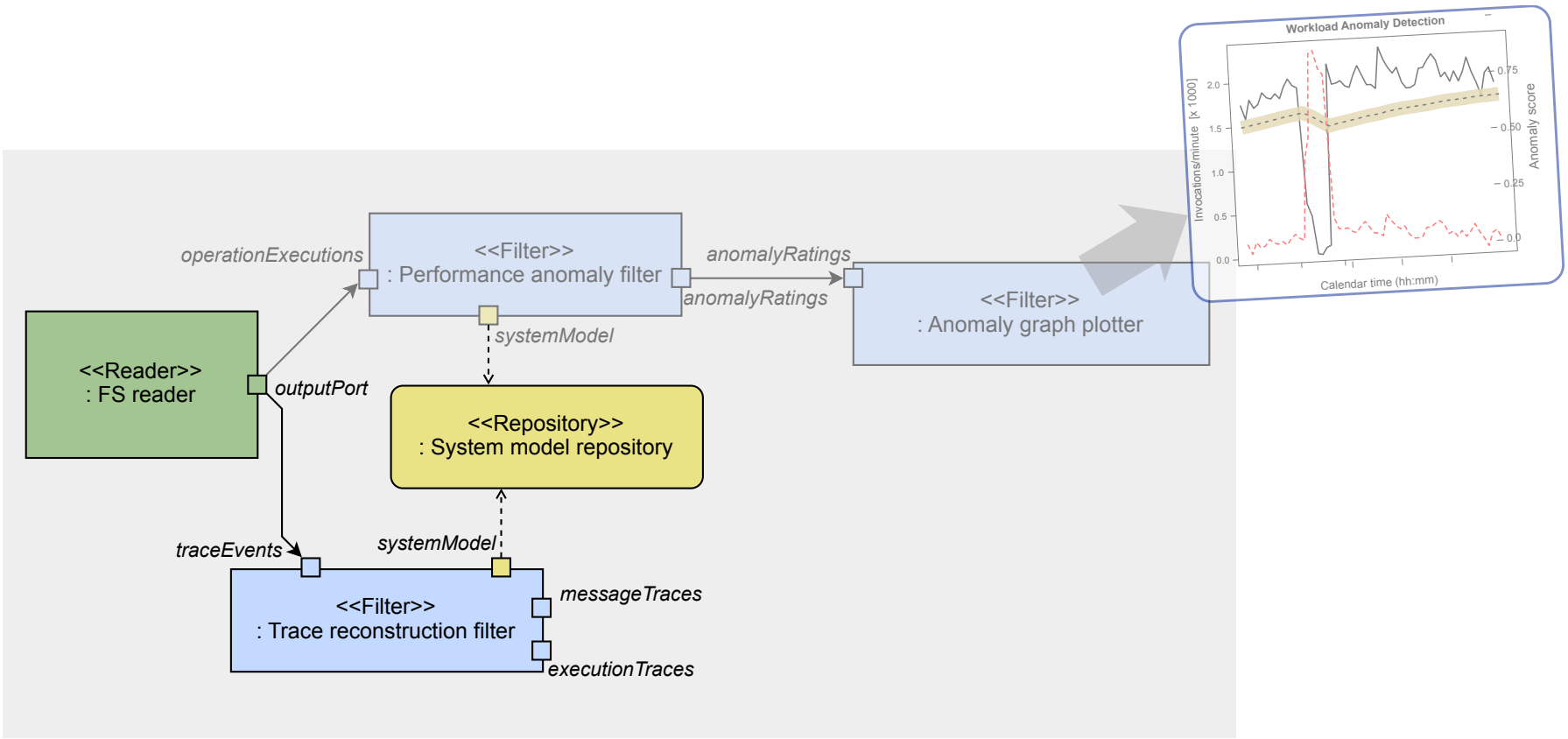


Kieker.Analysis example pipes-and-filters configuration

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

Example Pipe-and-Filter Configuration

Christian-Albrechts-Universität zu Kiel

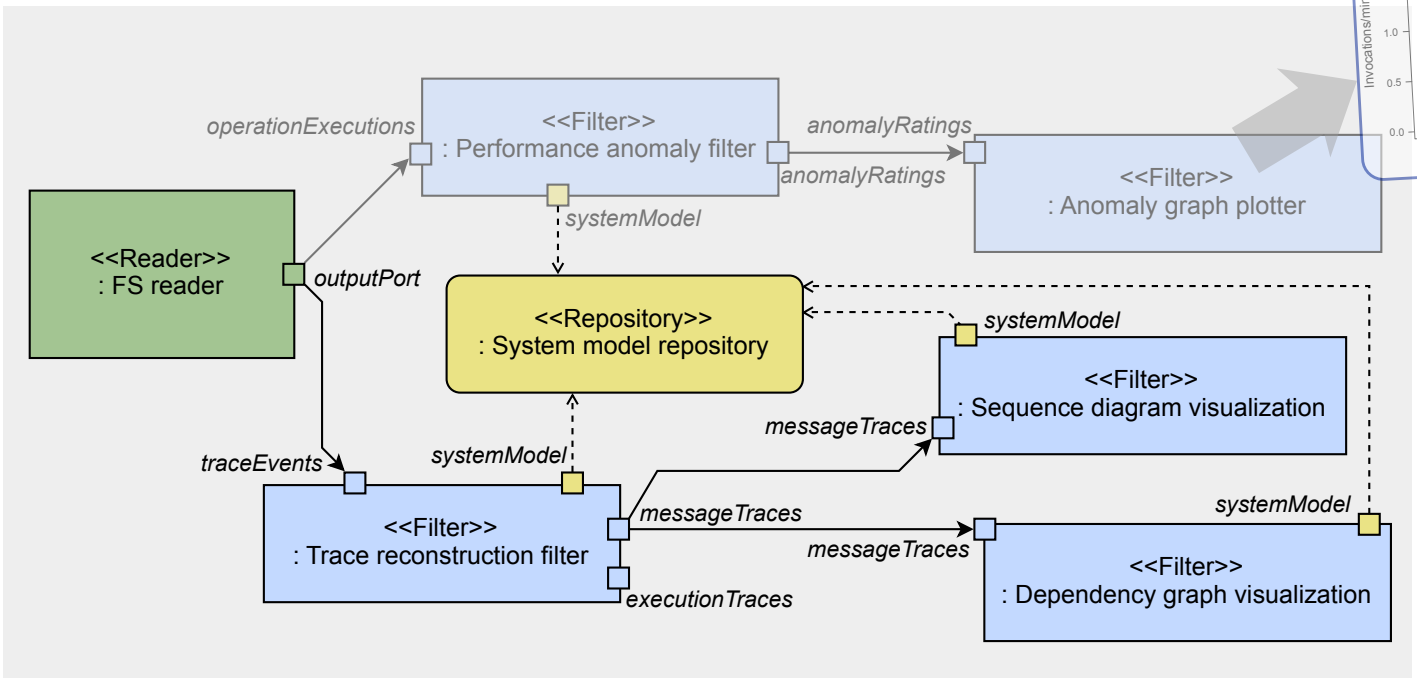


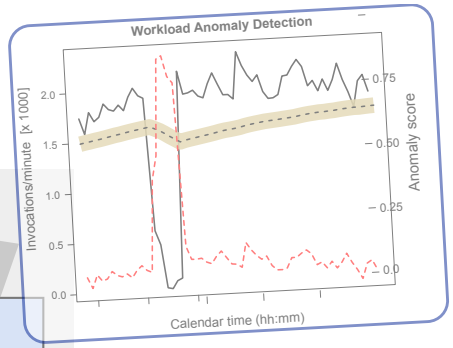
Kieker.Analysis example pipes-and-filters configuration

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

Example Pipe-and-Filter Configuration

Christian-Albrechts-Universität zu Kiel



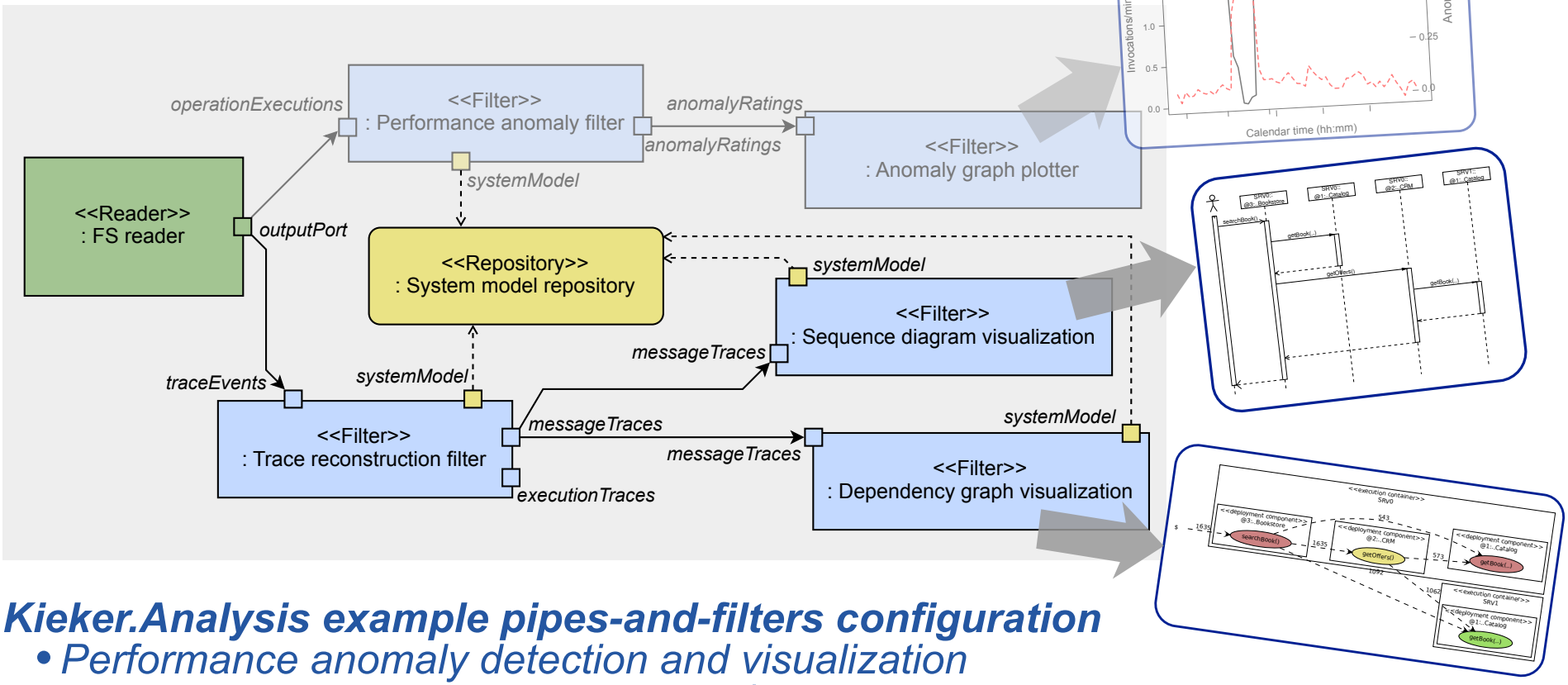


Kieker.Analysis example pipes-and-filters configuration

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

Example Pipe-and-Filter Configuration

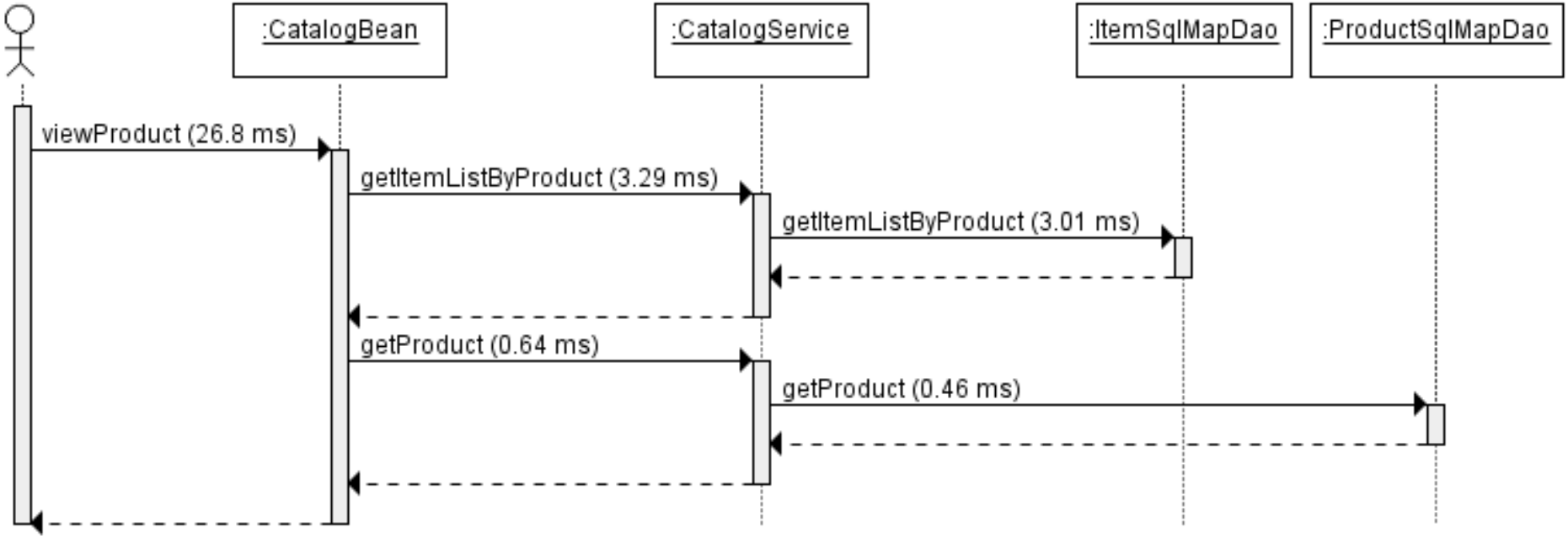
Christian-Albrechts-Universität zu Kiel



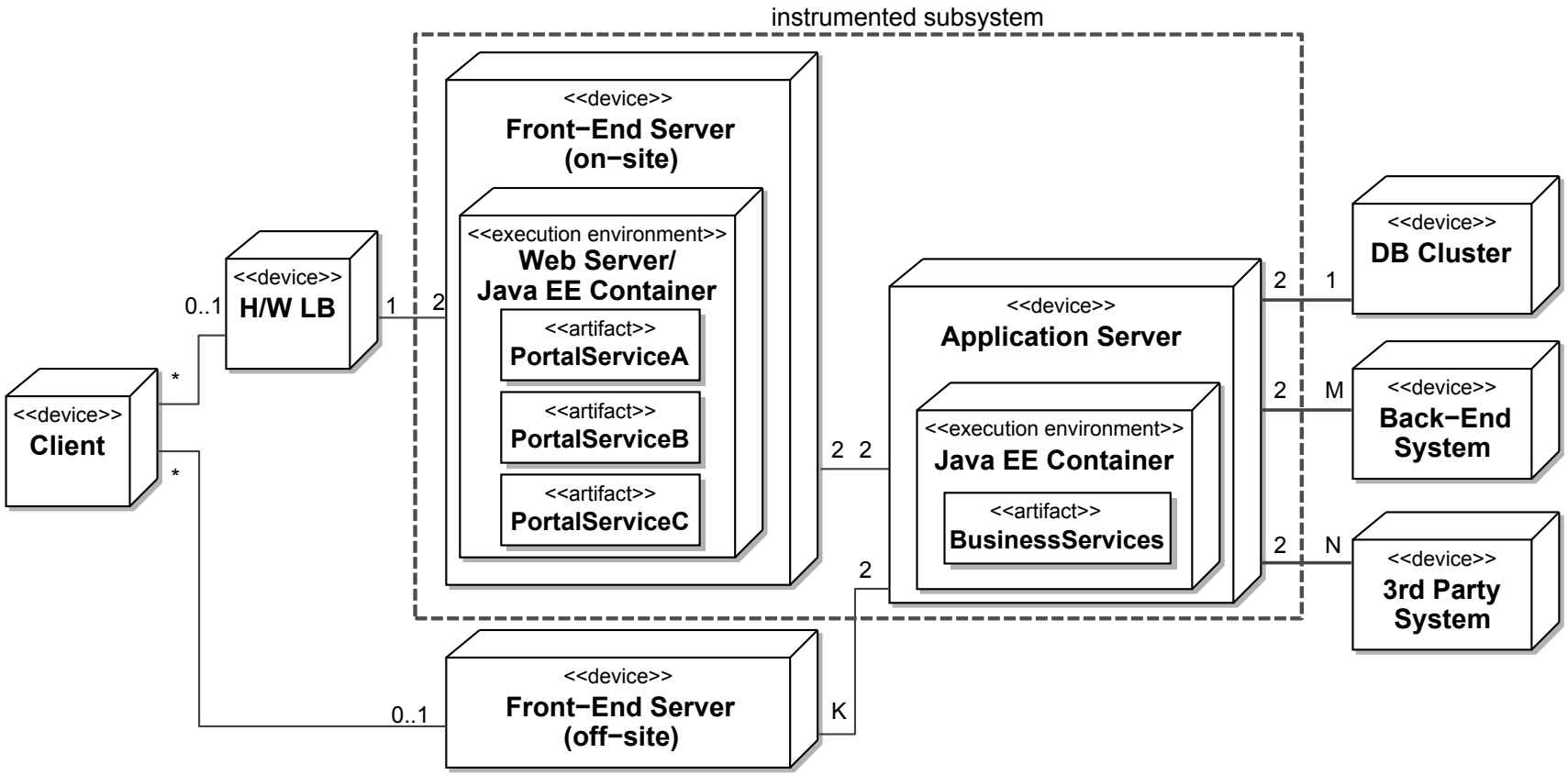
Kieker.Analysis example pipes-and-filters configuration

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

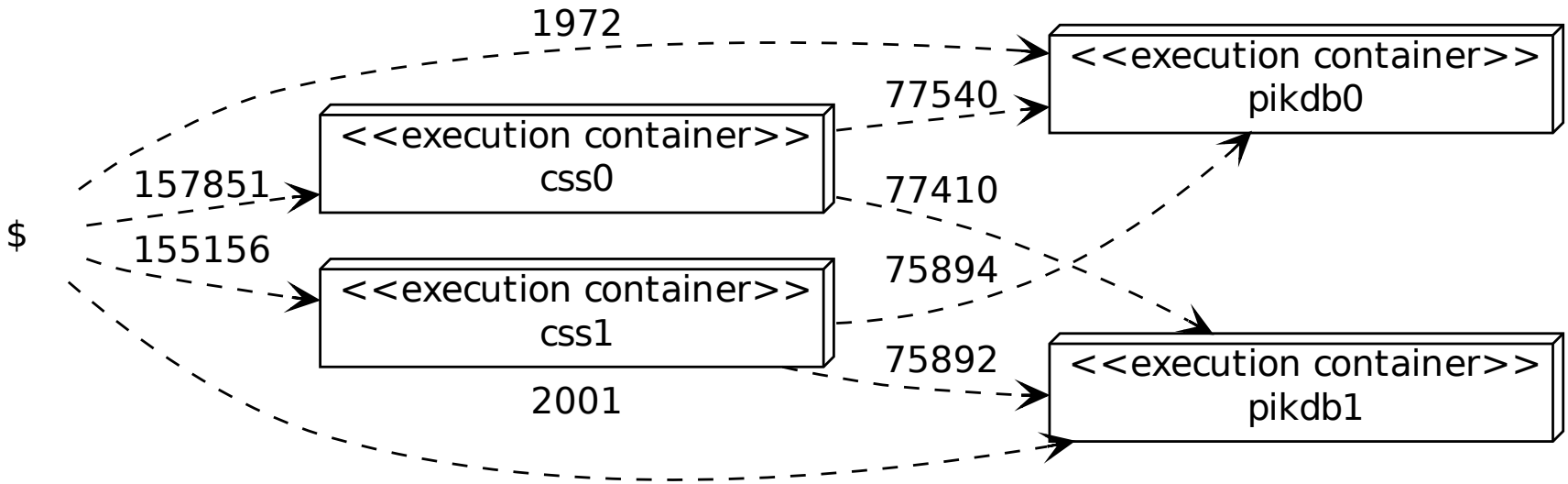
Generated Sequence Diagram

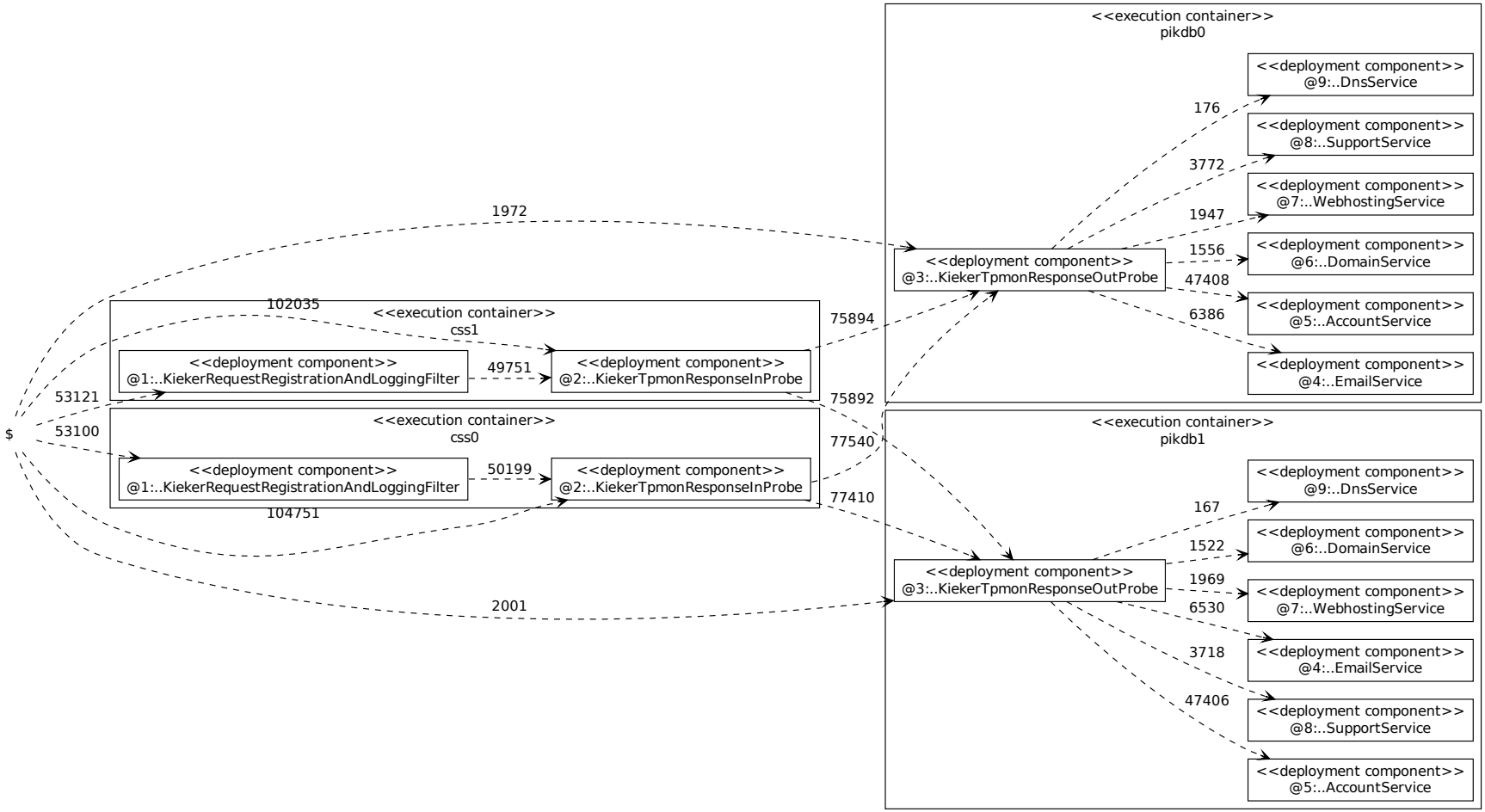


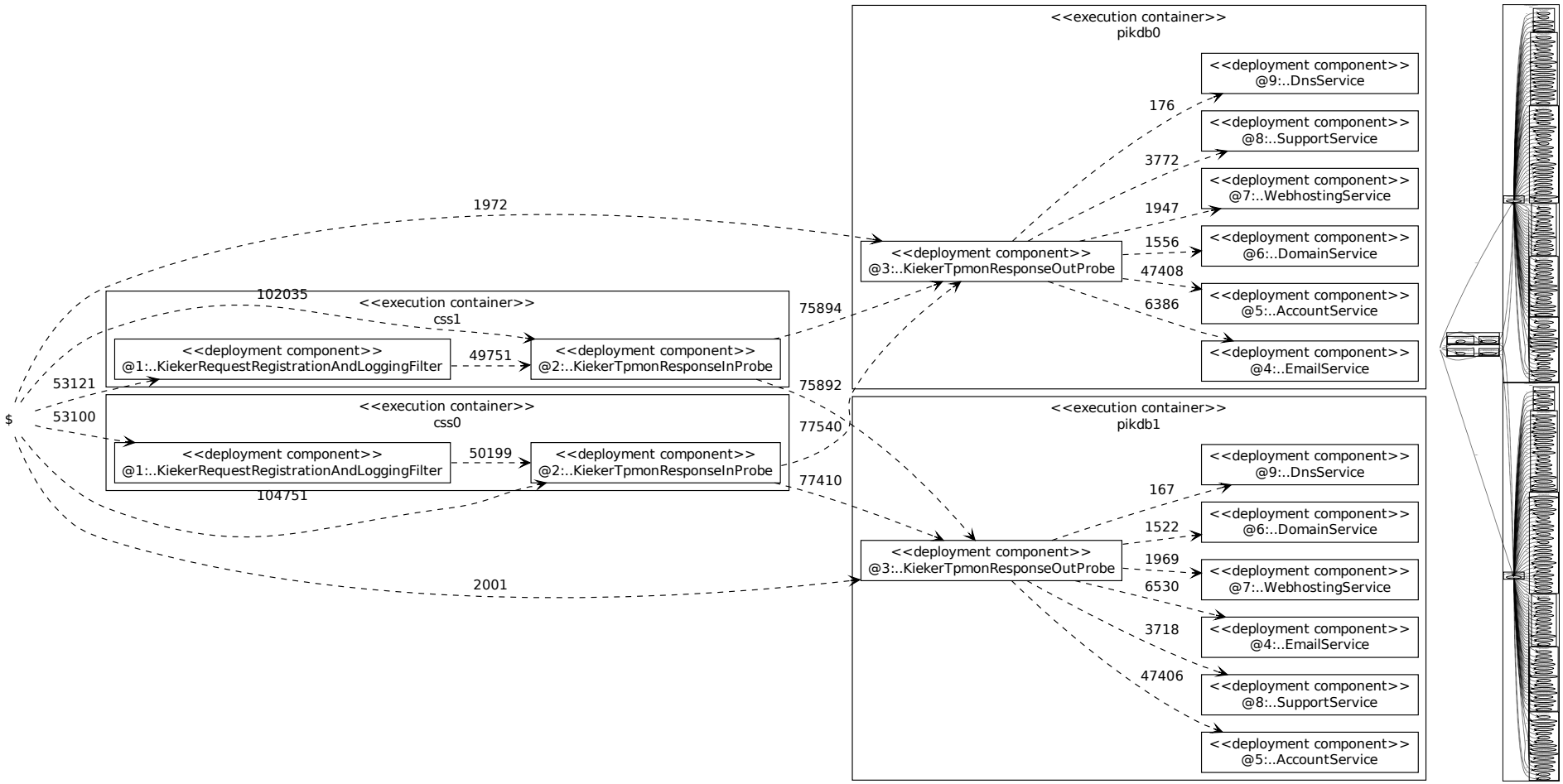
Distributed Enterprise Java System



Servlet, Spring and CXF/SOAP probes


1972
77540
157851
77410
$
155156
75894
75892
2001
<<execution container>>
pikdb0
<<execution container>>
css0
<<execution container>>
css1
<<execution container>>
pikdb1


$
53121
53100
1972
102035
<<execution container>>
css1
<<deployment component>>
@1:..KiekerRequestRegistrationAndLoggingFilter
49751
<<deployment component>>
@2:..KiekerTpmonResponseInProbe
<<execution container>>
css0
<<deployment component>>
@1:..KiekerRequestRegistrationAndLoggingFilter
50199
<<deployment component>>
@2:..KiekerTpmonResponseInProbe
104751
2001
75894
75892
77540
77410
<<execution container>>
pikdb0
<<deployment component>>
@3:..KiekerTpmonResponseOutProbe
176
3772
1947
1556
47408
6386
<<deployment component>>
@9:..DnsService
<<deployment component>>
@8:..SupportService
<<deployment component>>
@7:..WebhostingService
<<deployment component>>
@6:..DomainService
<<deployment component>>
@5:..AccountService
<<deployment component>>
@4:..EmailService
<<execution container>>
pikdb1
<<deployment component>>
@3:..KiekerTpmonResponseOutProbe
167
1522
1969
6530
3718
47406
<<deployment component>>
@9:..DnsService
<<deployment component>>
@6:..DomainService
<<deployment component>>
@7:..WebhostingService
<<deployment component>>
@4:..EmailService
<<deployment component>>
@8:..SupportService
<<deployment component>>
@5:..AccountService

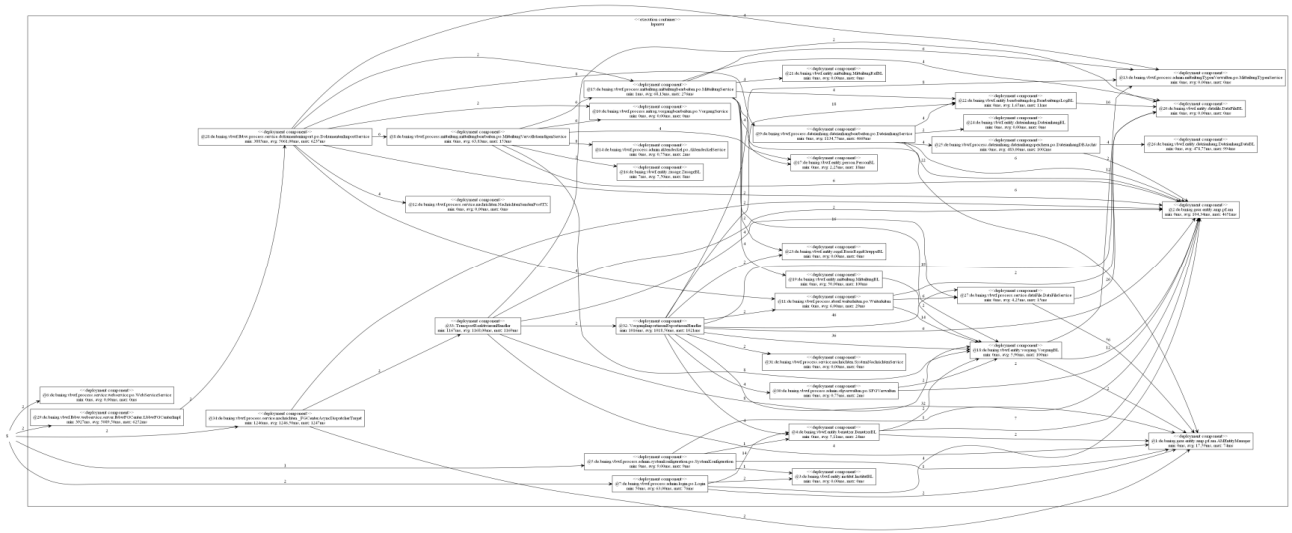

$
53121
53100
1972
102035
<<execution container>>
css1
<<deployment component>>
@1:..KiekerRequestRegistrationAndLoggingFilter
49751
<<deployment component>>
@2:..KiekerTpmonResponseInProbe
<<execution container>>
css0
<<deployment component>>
@1:..KiekerRequestRegistrationAndLoggingFilter
50199
<<deployment component>>
@2:..KiekerTpmonResponseInProbe
104751
2001
75894
75892
77540
77410
<<execution container>>
pikdb0
<<deployment component>>
@3:..KiekerTpmonResponseOutProbe
176
3772
1947
1556
47408
6386
<<deployment component>>
@9:..DnsService
<<deployment component>>
@8:..SupportService
<<deployment component>>
@7:..WebhostingService
<<deployment component>>
@6:..DomainService
<<deployment component>>
@5:..AccountService
<<deployment component>>
@4:..EmailService
<<execution container>>
pikdb1
<<deployment component>>
@3:..KiekerTpmonResponseOutProbe
167
1522
1969
6530
3718
47406
<<deployment component>>
@9:..DnsService
<<deployment component>>
@6:..DomainService
<<deployment component>>
@7:..WebhostingService
<<deployment component>>
@4:..EmailService
<<deployment component>>
@8:..SupportService
<<deployment component>>
@5:..AccountService

Beispiele aus Entwicklung und Produktion

b+m FGCenter

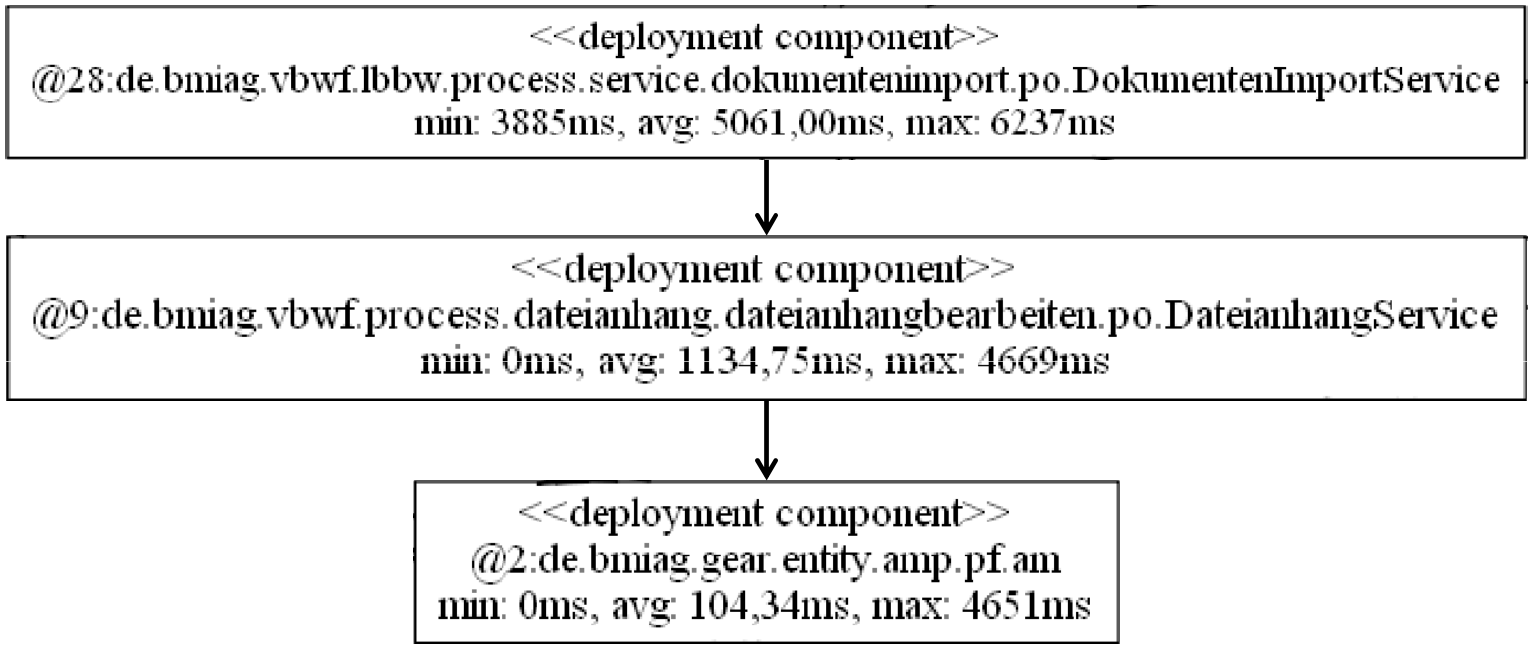
Marktführende Lösung für das Förderkreditgeschäft mit über 40.000 Anwendern und einem Marktanteil von insgesamt 30%. Bildet den gesamten Workflow der Beratung, Bearbeitung und Auszahlung von Krediten ab.

- Performanceprobleme beim Importieren eines Dokumentes per Webservice
- Aktivierung der Messpunkte im relevanten fachlichen Bereich in Produktion
- Erstellung eines Abhängigkeitsgraphen

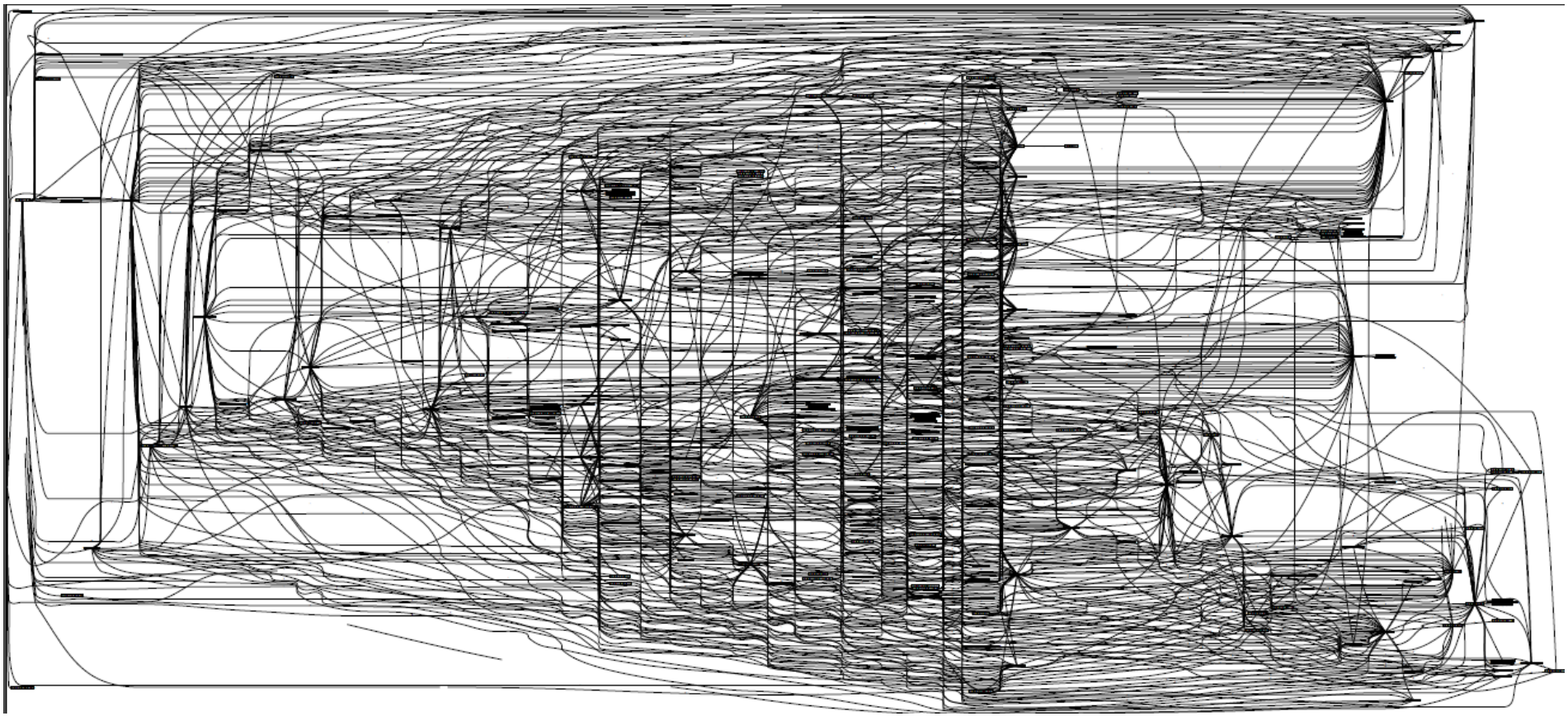


Beispiele aus Entwicklung und Produktion

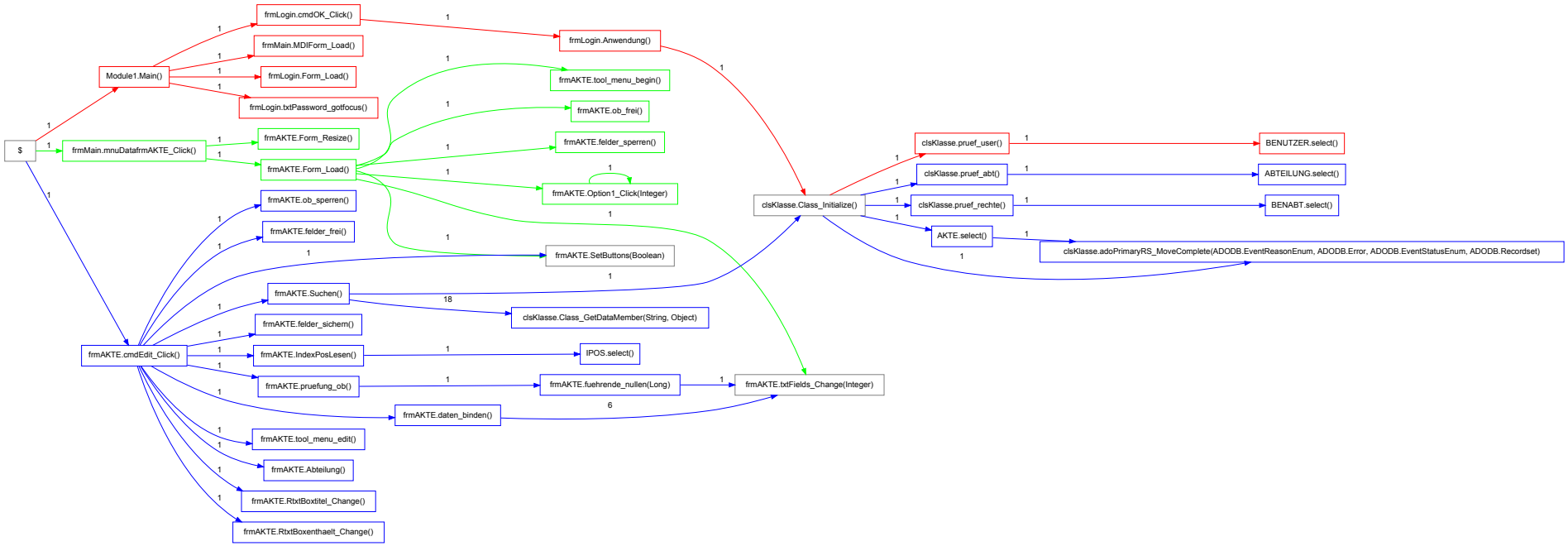
- Reproduzierbare Ausreißer im Bereich Entity beim Import



- Identifizierung eines fehlenden Indices als Ursache
- Nach Korrektur Verbesserung der Performance um Faktor 9 -10

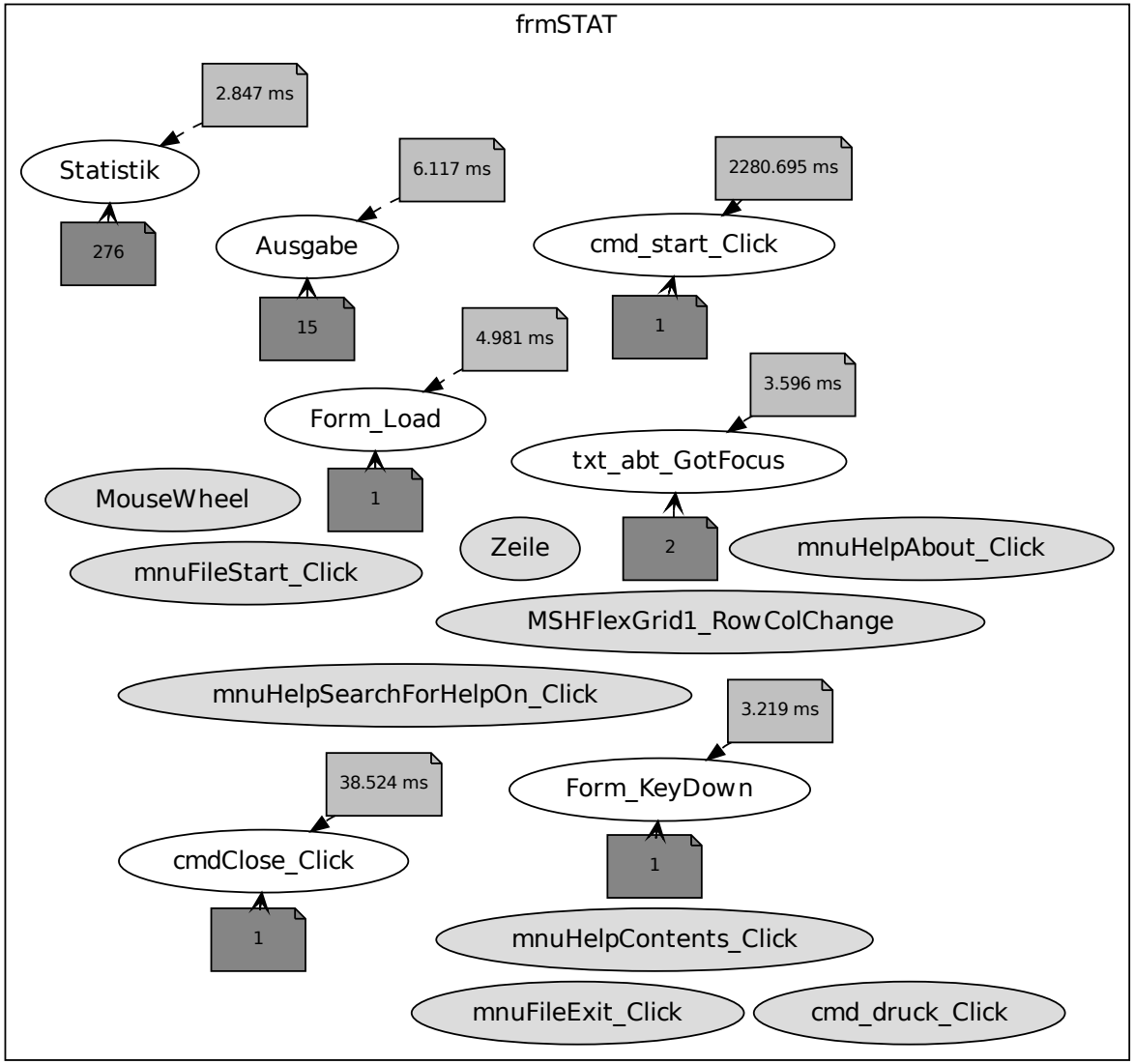


DynaMod Case study at HSH Nordbank AG.



dataport

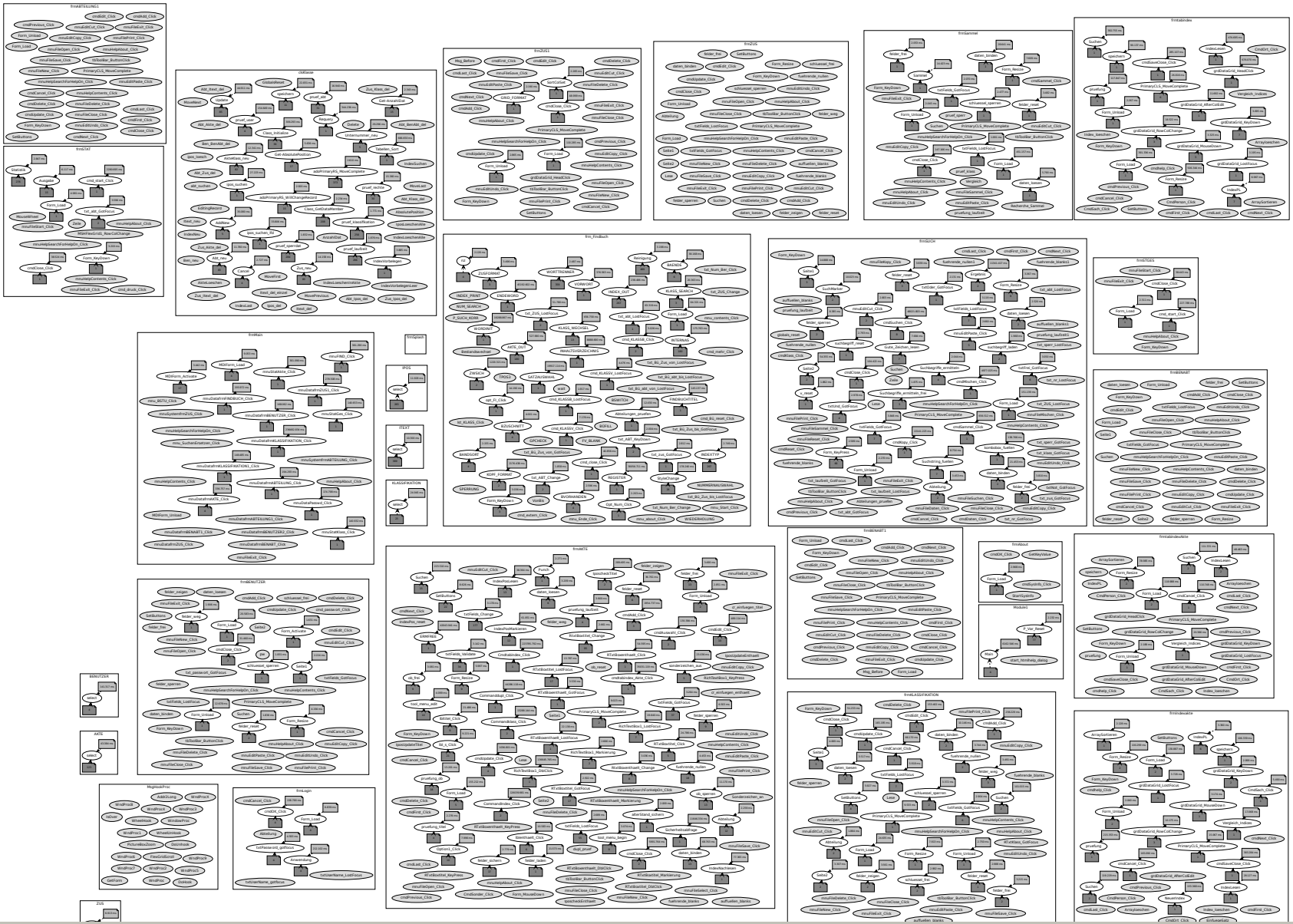
DynaMod Case study at Dataport
(Kieker-Erweiterungen durch Holger Knoche, b+m Informatik AG).


frmSTAT
2.847 ms
Statistik
276
6.117 ms
Ausgabe
15
2280.695 ms
cmd_start_Click
1
4.981 ms
Form_Load
1
3.596 ms
txt_abt_GotFocus
2
MouseWheel
mnuFileStart_Click
Zeile
mnuHelpAbout_Click
MSHFlexGrid1_RowColChange
mnuHelpSearchForHelpOn_Click
3.219 ms
Form_KeyDown
1
38.524 ms
cmdClose_Click
1
mnuHelpContents_Click
mnuFileExit_Click
cmd_druck_Click

Identification of unused Functions

Reverse Engineering of Visual Basic 6

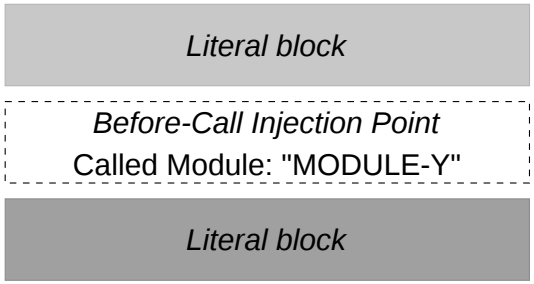
Christian-Albrechts-Universität zu Kiel



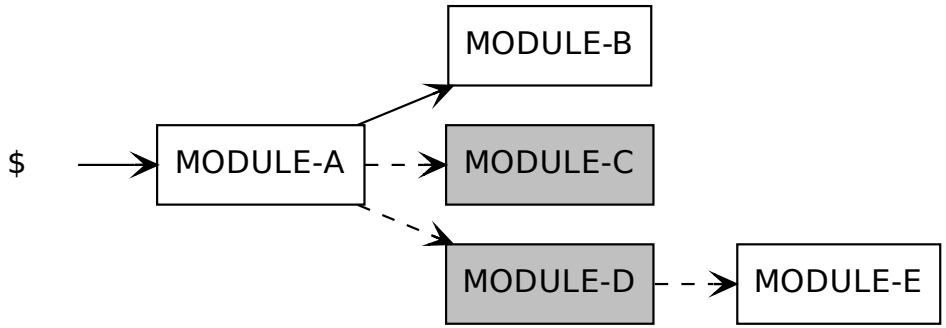

```
IDENTIFICATION DIVISION.
PROGRAM-ID. MODULE-X.

PROCEDURE DIVISION.

    CALL "MODULE-Y".
    GOBACK.
```



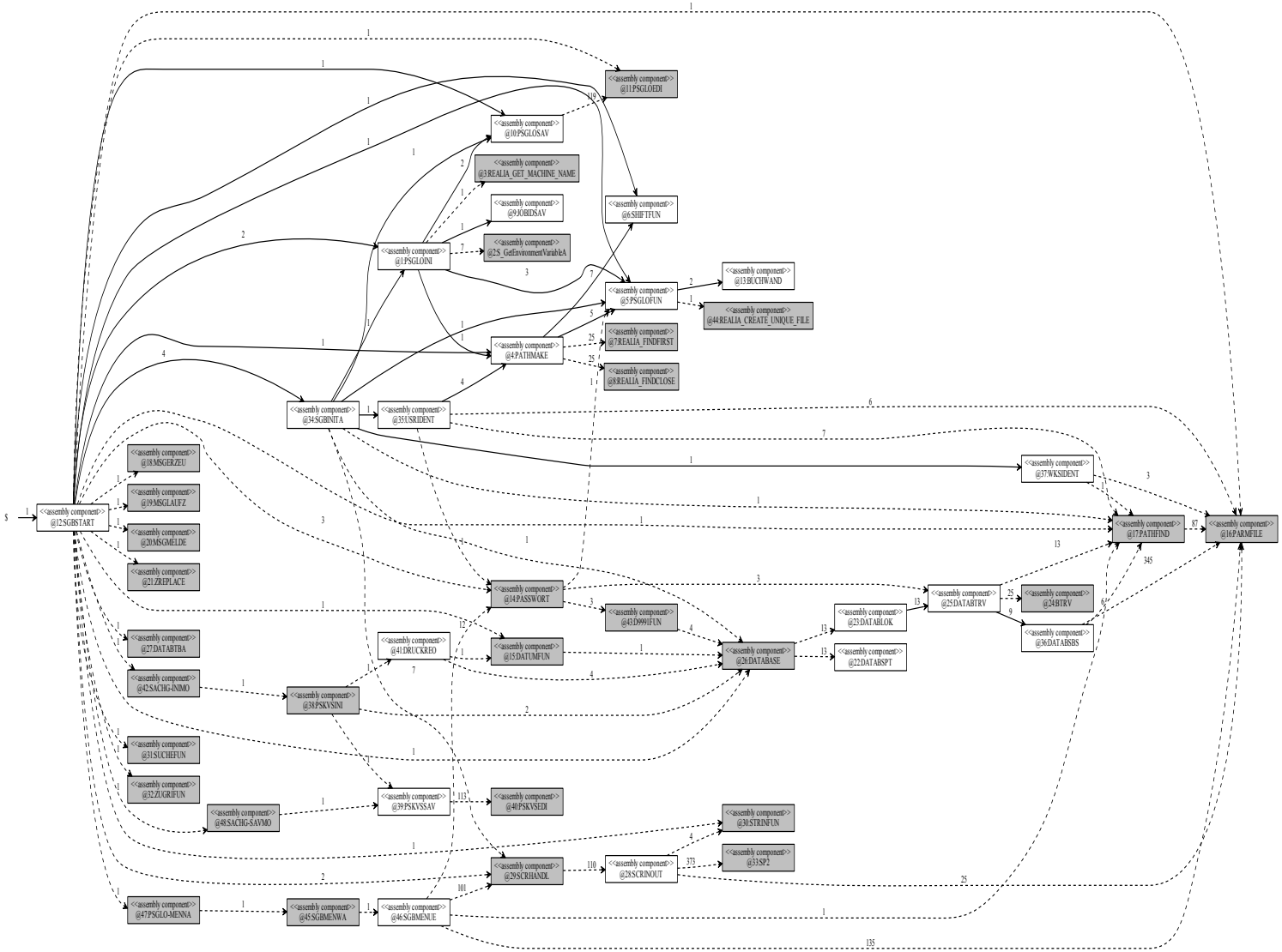
AOP-based COBOL instrumentation



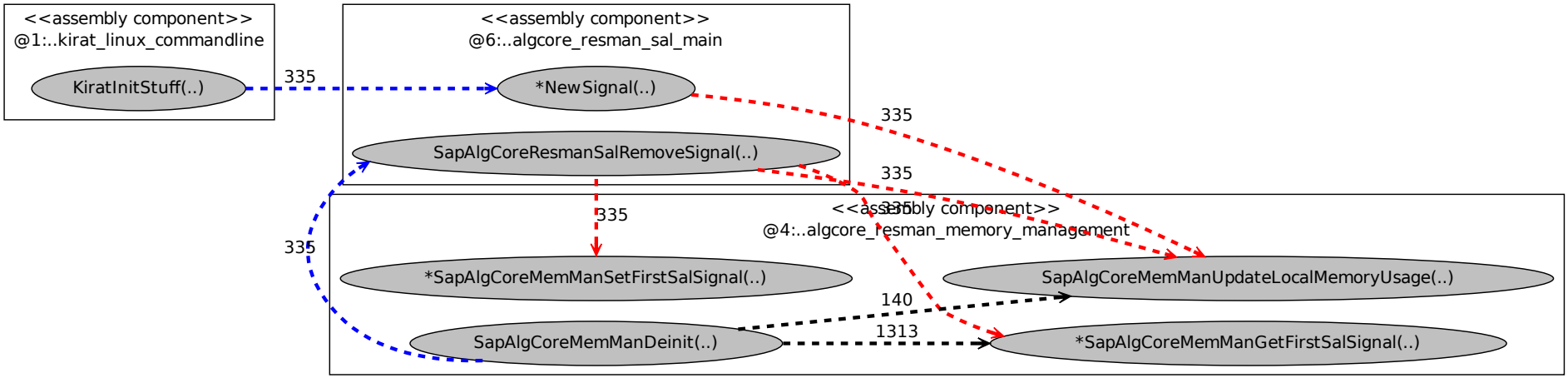
Module-level call dependency graph with assumed dependencies

Fallstudie der b+m Informatik AG (Kieker-Erweiterungen durch Holger Knoche):

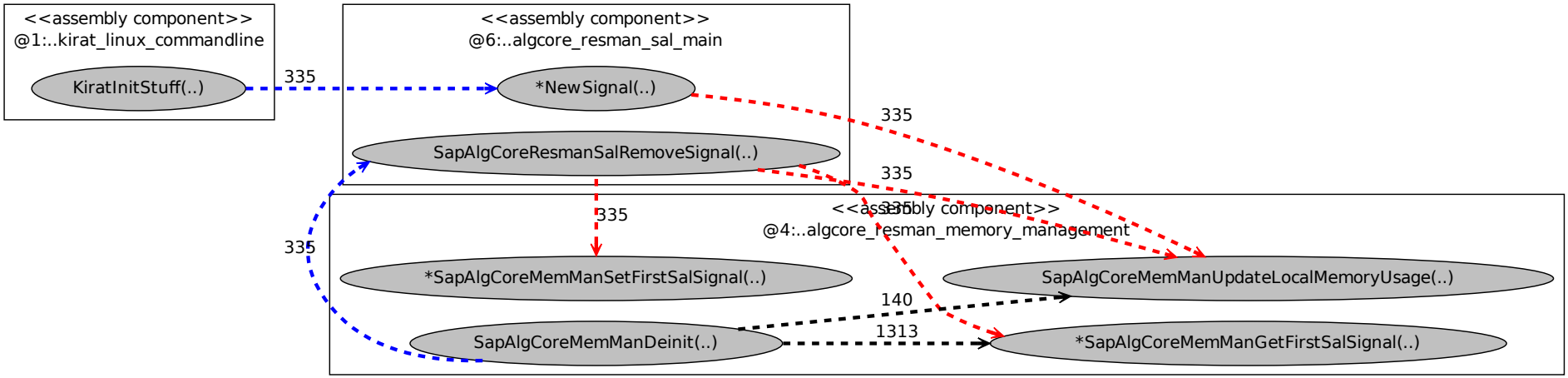
1. Industrieller Kontext mit mehr als 1.000 COBOL Modulen.
2. 140.351 Messpunkte mit 14 verschiedenen Arten von Messsonden.


<<assembly component>> @12:SGBSTART
<<assembly component>> @11:PSGLOEDI
<<assembly component>> @10:PSGLOSAV
<<assembly component>> @3:REALIA_GET_MACHINE_NAME
<<assembly component>> @9:JOBIDSAV
<<assembly component>> @6:SHIFTFUN
<<assembly component>> @2:S_GetEnvironmentVariableA
<<assembly component>> @1:PSGLOINI
<<assembly component>> @13:BUCHWAND
<<assembly component>> @5:PSGLOFUN
<<assembly component>> @44:REALIA_CREATE_UNIQUE_FILE
<<assembly component>> @7:REALIA_FINDFIRST
<<assembly component>> @8:REALIA_FINDCLOSE
<<assembly component>> @4:PATHMAKE
<<assembly component>> @34:SGBINITA
<<assembly component>> @35:USRIDENT
<<assembly component>> @18:MSGERZEU
<<assembly component>> @19:MSGLAUFZ
<<assembly component>> @20:MSGMELDE
<<assembly component>> @21:ZREPLACE
<<assembly component>> @37:WKSIDENT
<<assembly component>> @17:PATHFIND
<<assembly component>> @16:PARMFILE
<<assembly component>> @14:PASSWORT
<<assembly component>> @43:D9991FUN
<<assembly component>> @41:DRUCKREQ
<<assembly component>> @15:DATUMFUN
<<assembly component>> @26:DATABASE
<<assembly component>> @23:DATABLOK
<<assembly component>> @22:DATABSPT
<<assembly component>> @25:DATABTRV
<<assembly component>> @24:BTRV
<<assembly component>> @36:DATABSBS
<<assembly component>> @27:DATABTBA
<<assembly component>> @42:SACHG-INMO
<<assembly component>> @38:PSKVSINI
<<assembly component>> @31:SUCHEFUN
<<assembly component>> @32:ZUGRIFUN
<<assembly component>> @48:SACHG-SAVMO
<<assembly component>> @39:PSKVSSAV
<<assembly component>> @40:PSKVSEDI
<<assembly component>> @30:STRINFUN
<<assembly component>> @33:SP2
<<assembly component>> @29:SCRHANDL
<<assembly component>> @28:SCRINOUT
<<assembly component>> @47:PSGLO-MENNA
<<assembly component>> @45:SGBMENWA
<<assembly component>> @46:SGBMENUE

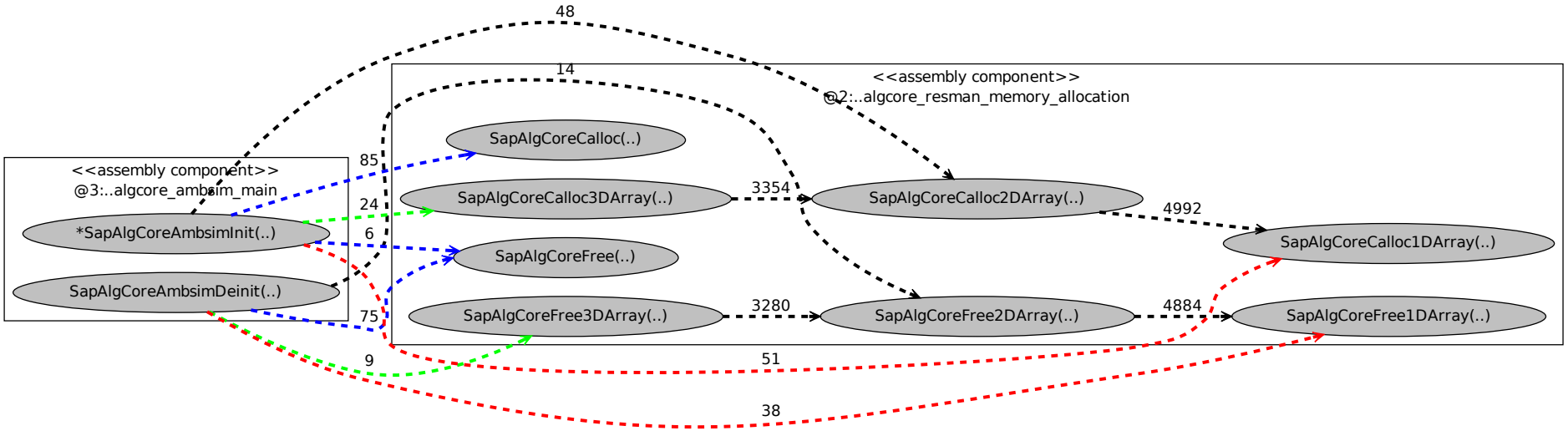
Source: `http://www.dss.tf.uni-kiel.de/en/research/kirat`



Identification of cyclic dependencies.

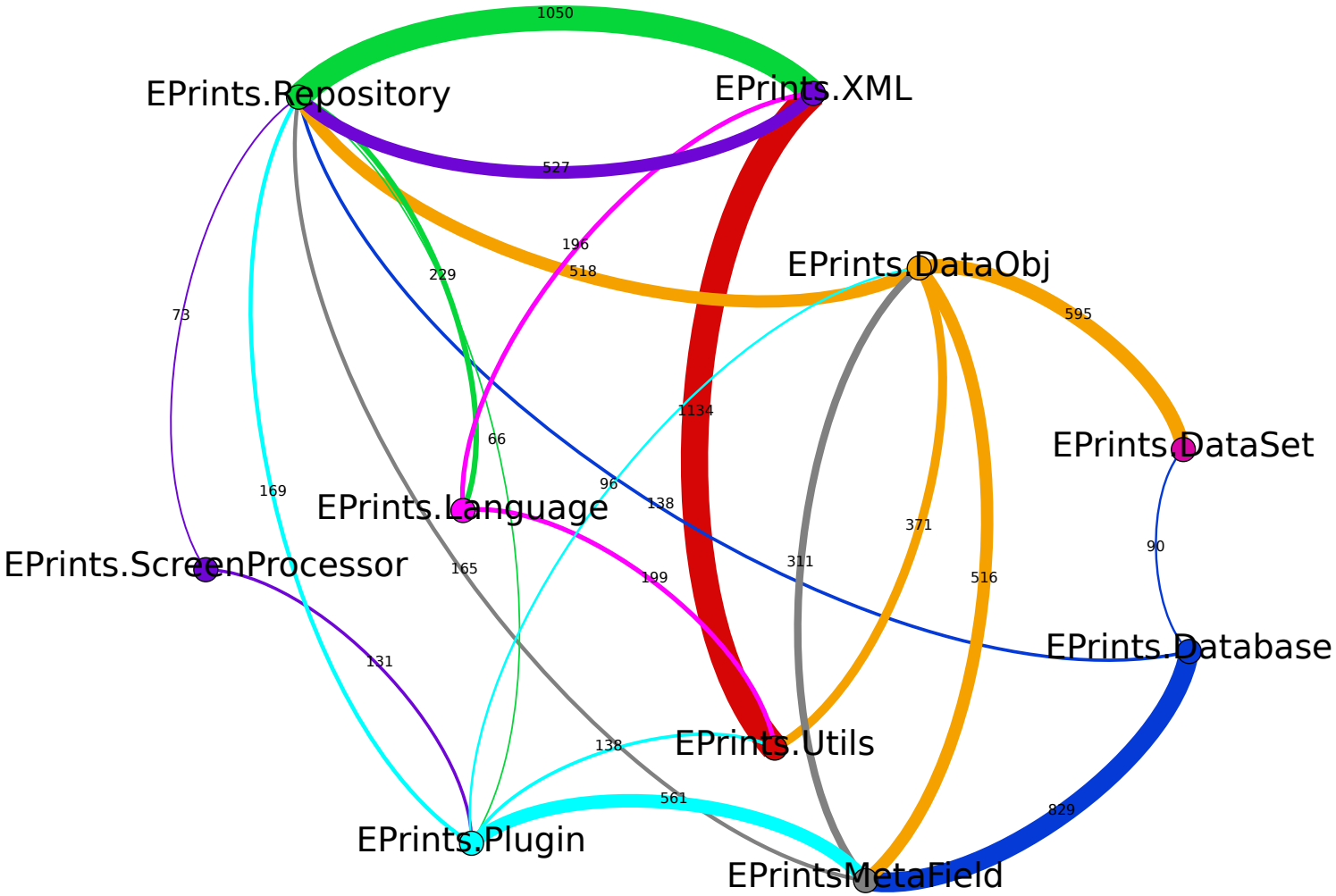


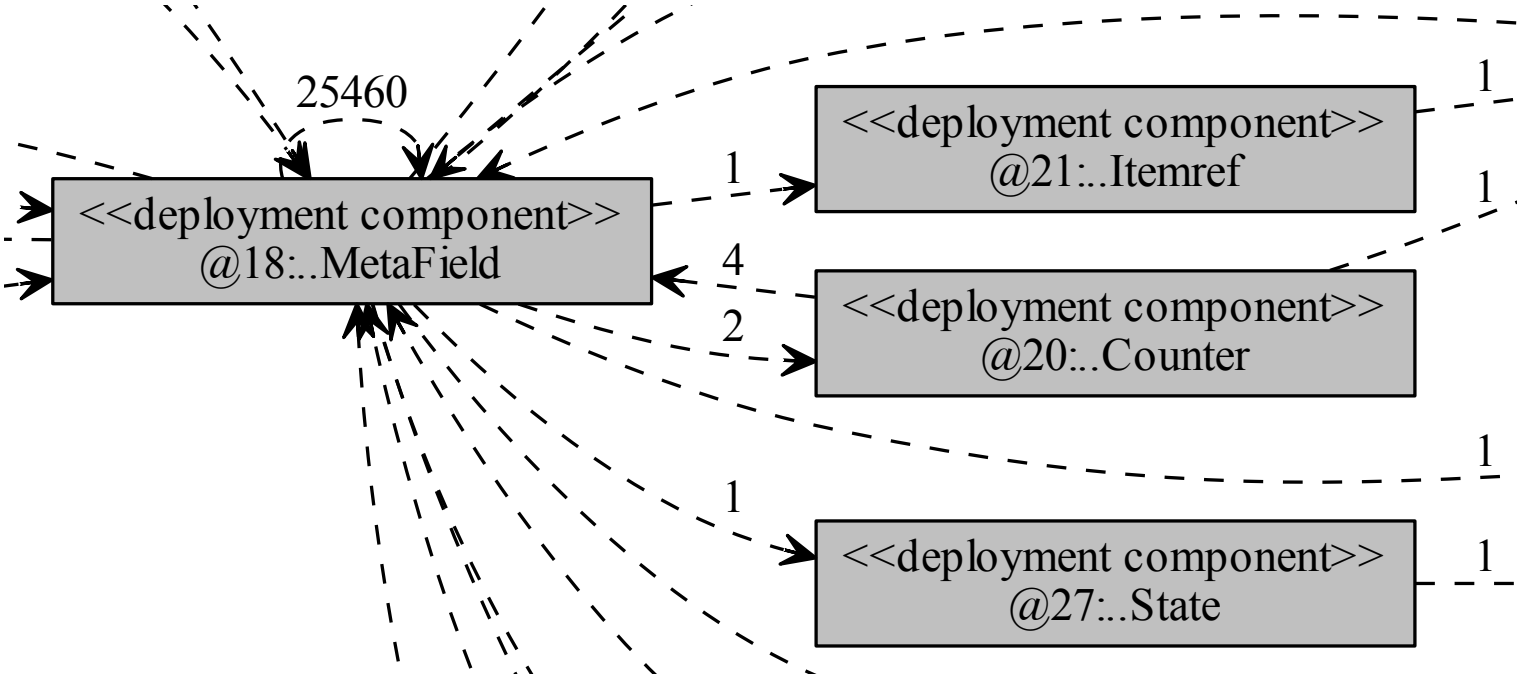
Identification of cyclic dependencies.



Identification of memory leaks.

First shot with full instrumentation:

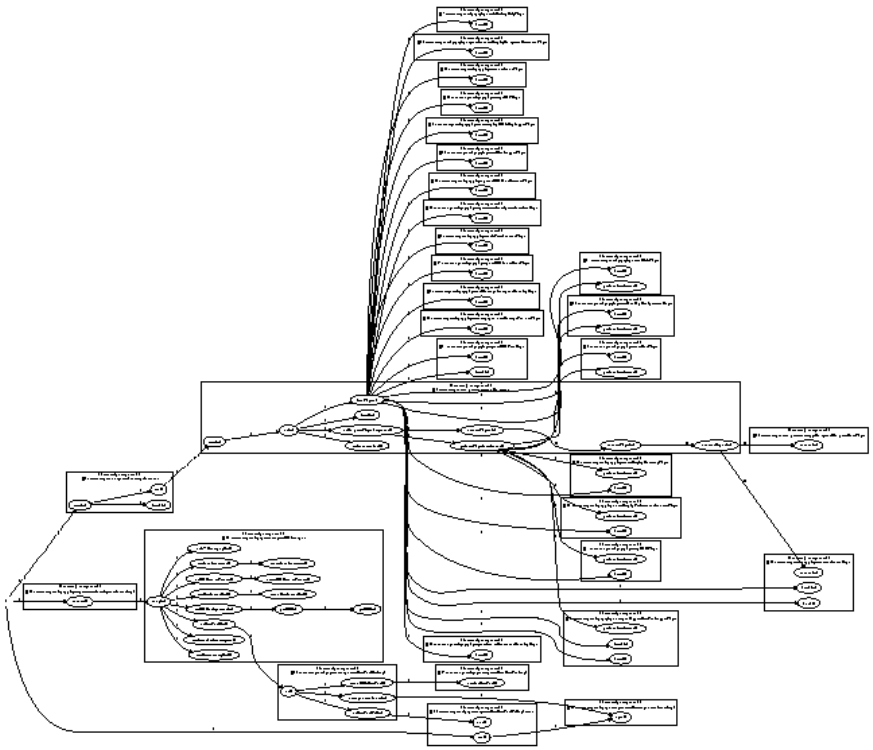

EPrints.Repository
EPrints.XML
EPrints.DataObj
EPrints.DataSet
EPrints.Language
EPrints.ScreenProcessor
EPrints.Database
EPrints.Utils
EPrints.Plugin
EPrintsMetaField
1050
527
196
518
229
73
595
1134
66
96
138
169
371
90
311
165
199
516
131
138
561
829


25460
<<deployment component>>
@18:..MetaField
<<deployment component>>
@21:..Itemref
<<deployment component>>
@20:..Counter
<<deployment component>>
@27:..State
1
1
1
4
2
1
1
1

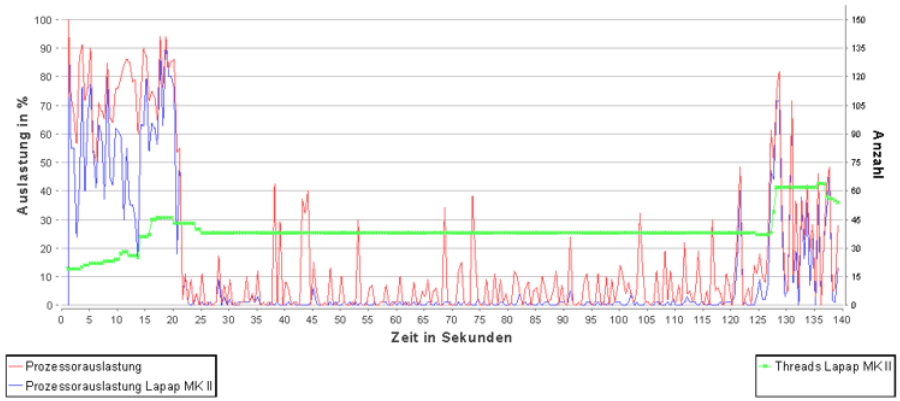
Kieker in Space

C A U
Christian-Albrechts-Universität zu Kiel

S122E008911



AIRBUS DEFENCE & SPACE



Agenda

Motivation

- An API can be used to create, configure, and execute analyses.
- But what about...
  - ...multiple projects and users?
  - ...larger analysis networks?
  - ...interactive visualizations?

Features

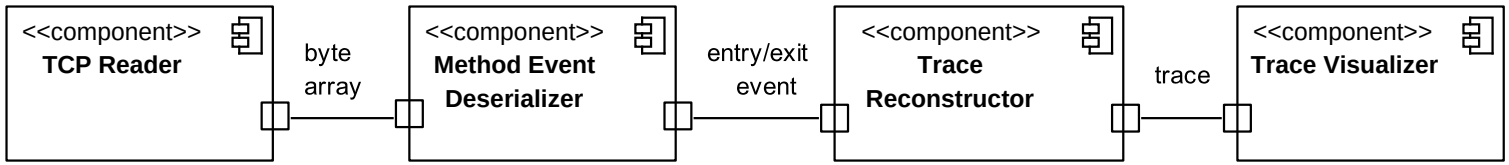
- A multi-user web application
- Kieker analyses can be configured and controlled in a browser
- Cockpits visualize live results from running analyses

- Developed since 2011
- Still beta-stated, but an experimental approach for analyses cockpits

- Included in the Kieker releases
- Open-source (Apache License, V. 2.0)

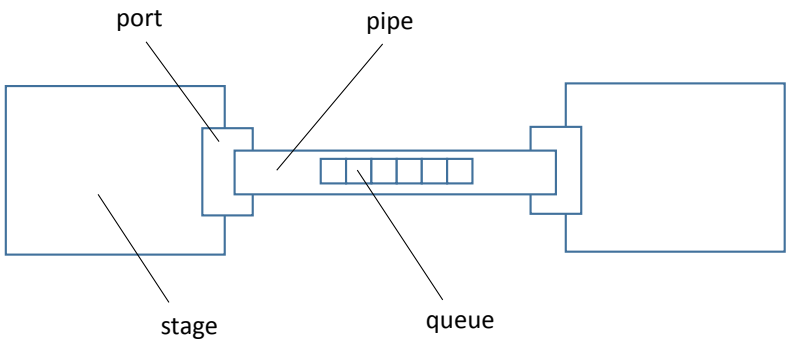
Let's take a look at Kieker's WebGUI!

Agenda



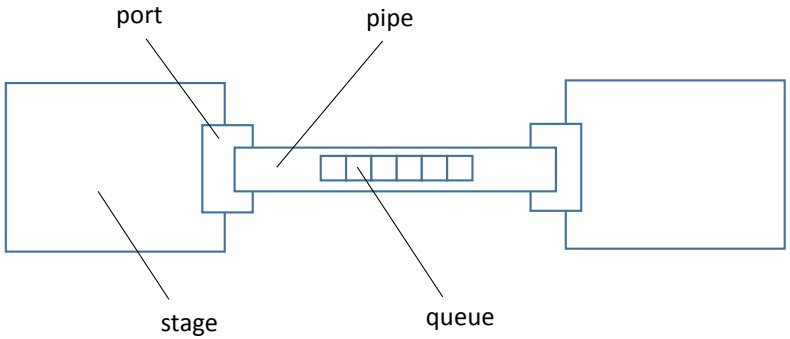
- Use of Java's Reflection API to invoke next stage
- Check for type-safetiness on each `deliver`
- Only readers are executed in a separate thread
- Limited support for stage composition

- Ports are realized by annotations
- Unnecessary synchronization overhead



The pipe abstraction encapsulates

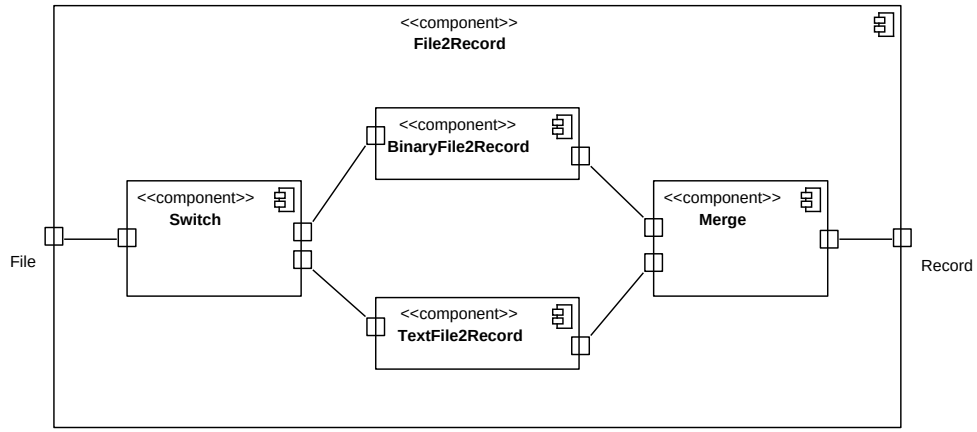
- stage execution, and
- synchronization between stages

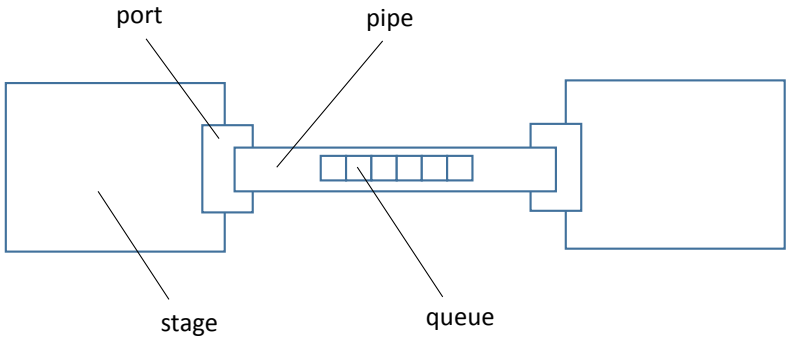


The pipe abstraction encapsulates

- stage execution, and
- synchronization between stages

A stage composed of multiple other stages

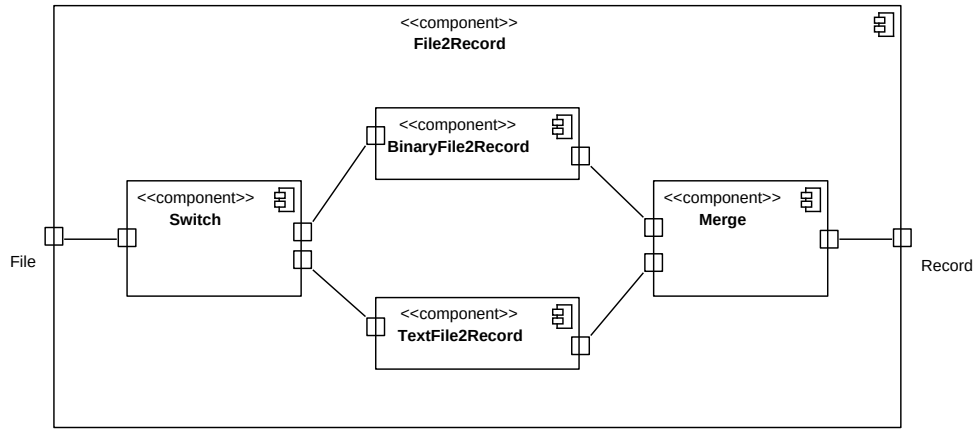


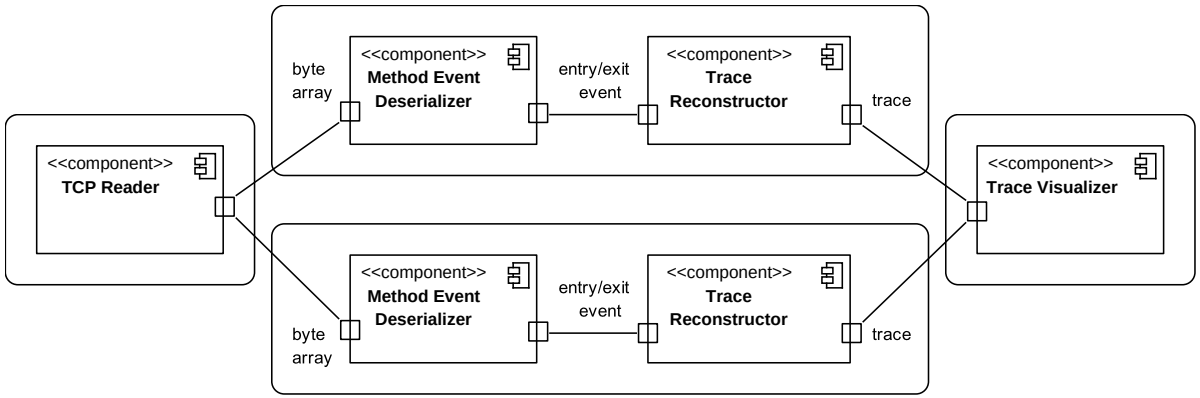


The pipe abstraction encapsulates

- stage execution, and
- synchronization between stages

A stage composed of multiple other stages



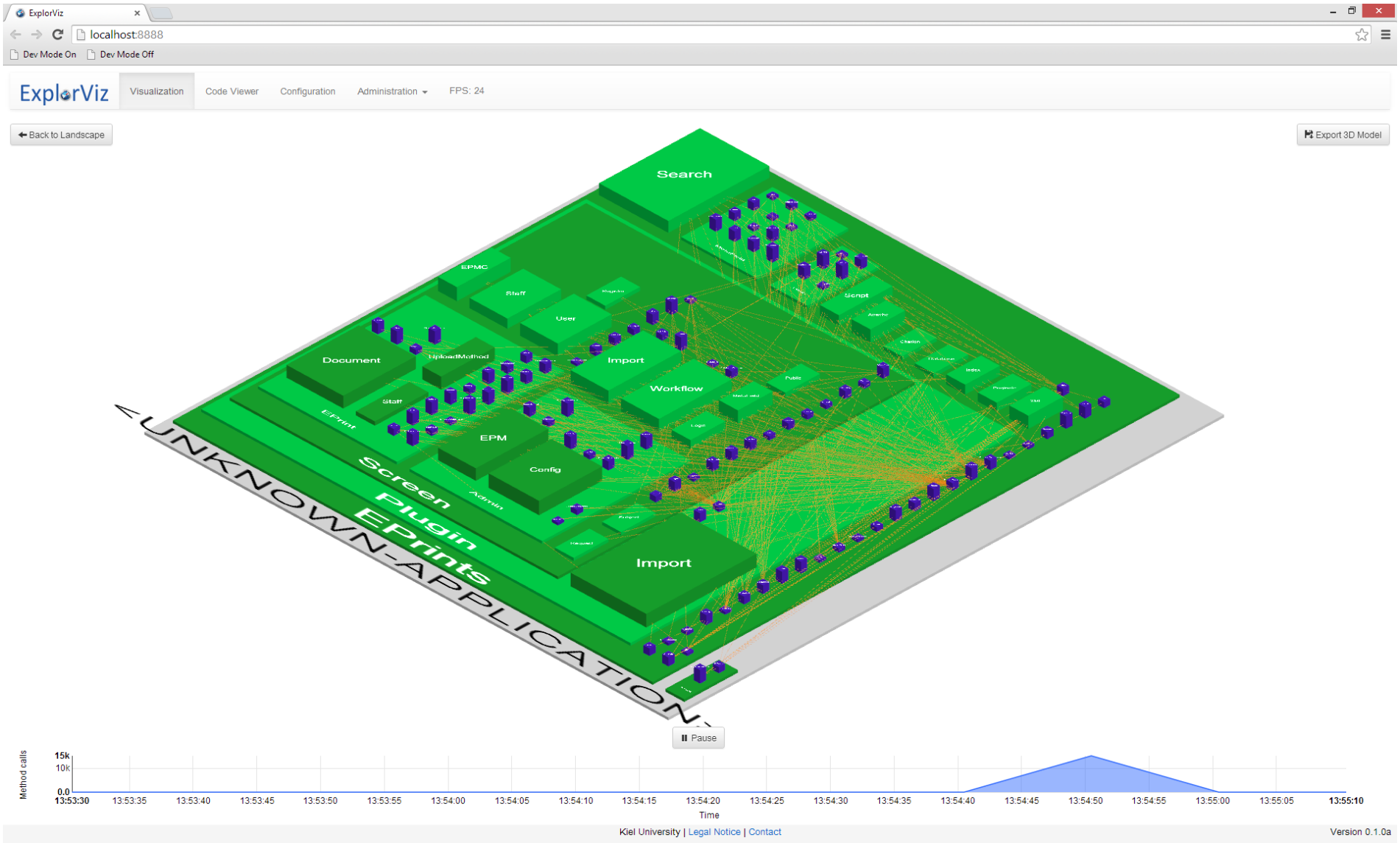


Sample thread assignment

3D Visualization of the EPrints Landscape

ExplorViz Fittkau et al. [2013], http://www.explorviz.net/

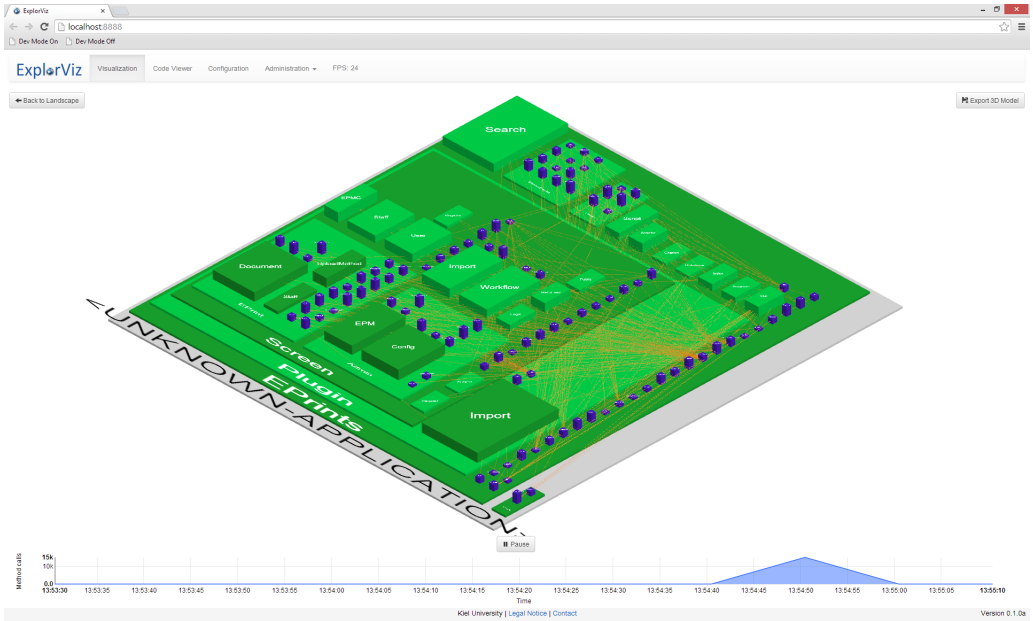
Christian-Albrechts-Universität zu Kiel

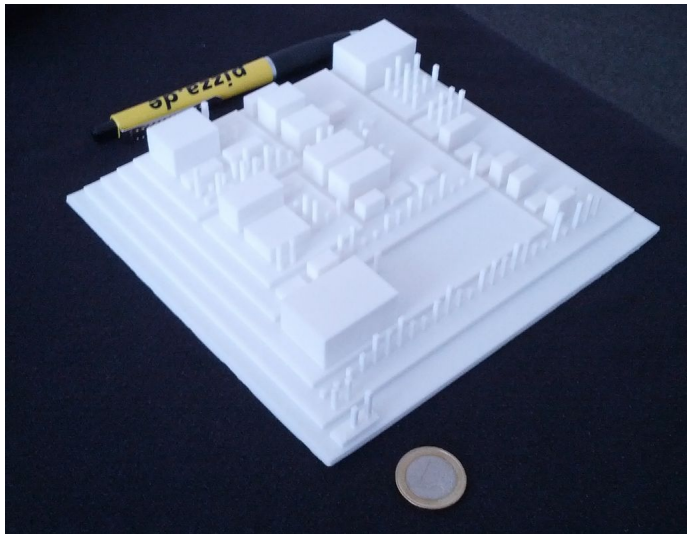


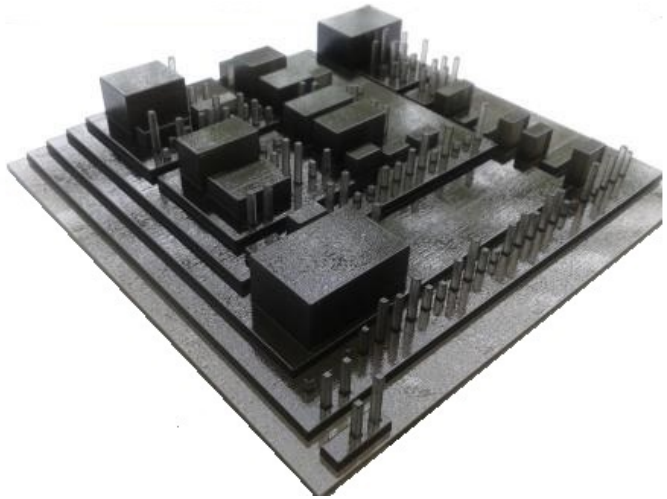
3D Visualization of the EPrints Landscape

ExplorViz Fittkau et al. [2013], `http://www.explorviz.net/`

Christian-Albrechts-Universität zu Kiel







F. Fittkau, J. Waller, C. Wulf, and W. Hasselbring. Live trace visualization for comprehending large software landscapes: The ExplorViz approach. In *1st IEEE International Working Conference on Software Visualization (VISSOFT 2013)*, Sept. 2013.

B. Harms. Reverse-Engineering und Analyse einer Plug-in-basierten Java-Anwendung. Master's thesis, Kiel University, Nov. 2013. URL `http://eprints.uni-kiel.de/23733/`.

H. Knoche, A. van Hoorn, W. Goerigk, and W. Hasselbring. Automated source-level instrumentation for dynamic dependency analysis of COBOL systems. In *Proceedings of the 14. Workshop Software-Reengineering (WSR '12)*, pages 33–34, May 2012. URL `http://eprints.uni-kiel.de/14417/`.

F. Magendanz. Dynamic analysis of .NET applications for architecture-based model extraction and test generation, Oct. 2011. URL `http://eprints.uni-kiel.de/15489/`.

S. Mahmens. Architektur-Rekonstrukion mit Kieker durch AOP-basierte Instrumentierung einer C++-Anwendung. Master's thesis, Institut für Informatik, Apr. 2014. URL `http://eprints.uni-kiel.de/24349/`.

B. Richter. Dynamische Analyse von COBOL-Systemarchitekturen zum modellbasierten Testen. Diploma thesis, University of Kiel, Germany, Aug. 2012. URL `http://eprints.uni-kiel.de/15489/`.

N. B. Wechselberg. Monitoring von Perl-basierten Webanwendungen mittels Kieker. Bachelorarbeit, Kiel University, April 2013. URL `http://eprints.uni-kiel.de/21141/`.