

# Combining Kieker with Gephi for Performance Analysis and Interactive Trace Visualization

Christian Zirkelbach, Wilhelm Hasselbring, and Leslie Carr

2015-11-06



## Introduction

### Performance issues regarding established Perl software

- ▶ Often difficult to identify potential bottlenecks
- ▶ Architectural discovery
- ▶ Program comprehension
- ▶ Performance analysis
- Visualization of results
- Kieker Monitoring Framework



# Approach

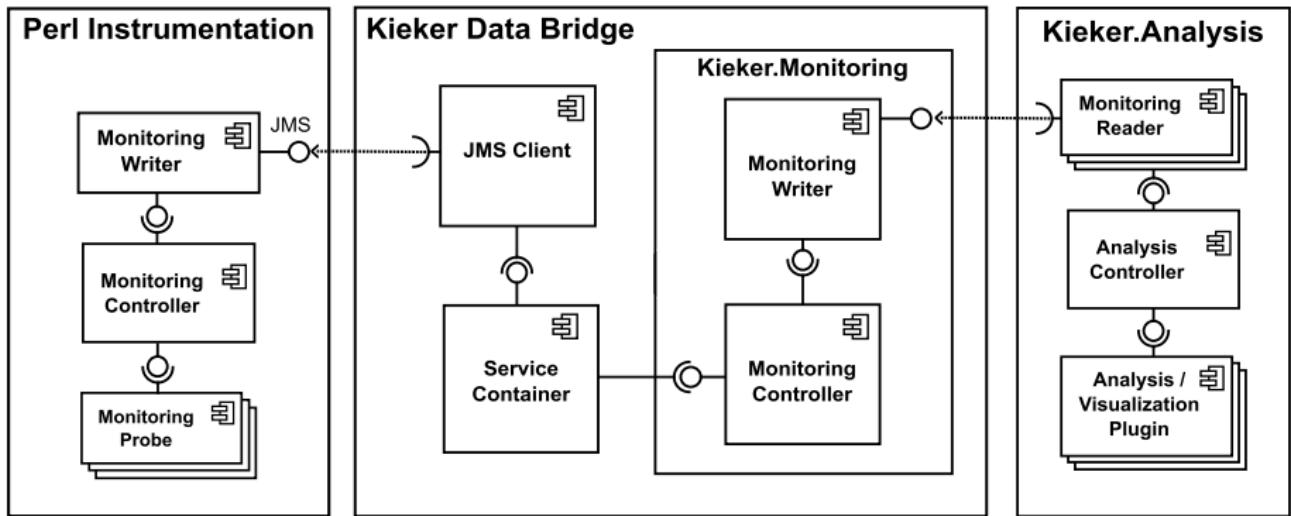
## Approach

```
1  use Sub :: WrapPackages
2      packages = > [qw(EPrints EPrints ::*)] ,
3      pre = > sub {
4          use Kieker ;
5          my $kieker = Kieker -> new () ;
6          my $packageName = $_[0];
7          $packageName =~ s /:::/./ g ;
8          $packageName =~ /^(.*) \..*? $ /;
9          $kieker -> EntryEvent ($packageName,$1);
10     },
11     post = > sub {
12         use Kieker ;
13         my $kieker = Kieker -> new () ;
14         my $packageName = $_[0];
15         $packageName =~ s /:::/./ g ;
16         $packageName =~ /^(.*) \..*? $ /;
17         $kieker -> ExitEvent ($packageName,$1);
18     };

```

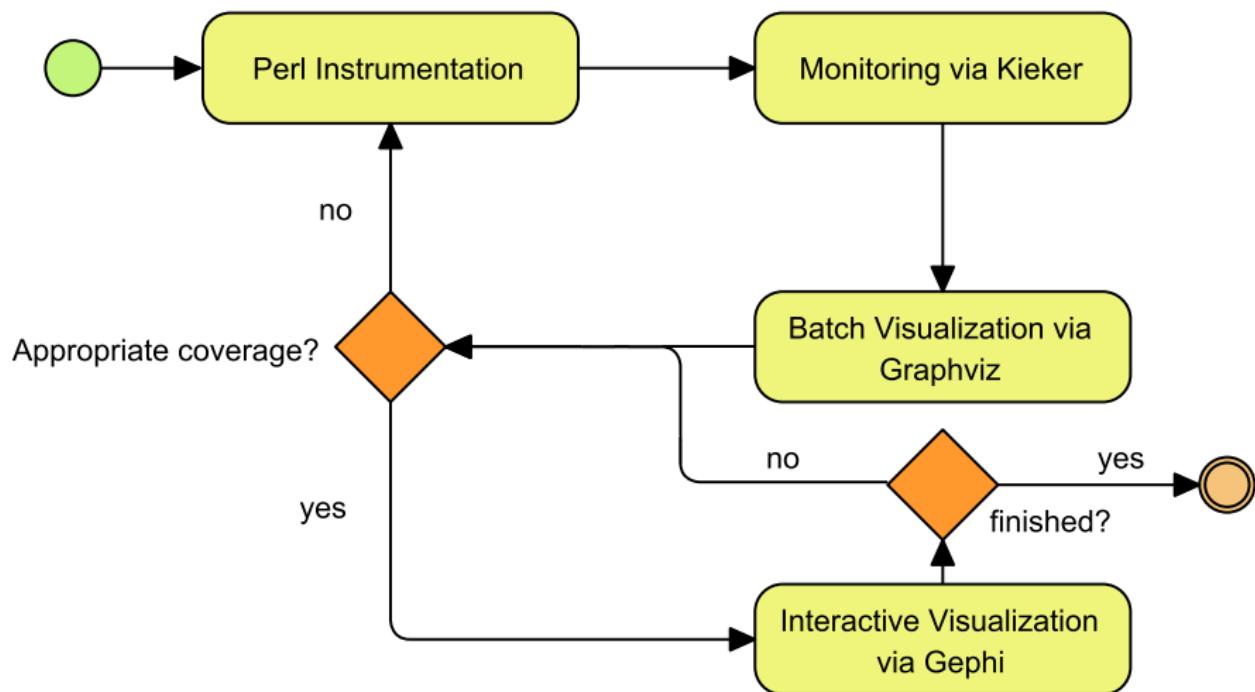
# Instrumentation Components

## Approach



# Performance Analysis Workflow

Approach



## Approach

### 1. Manipulating the generated dependency call graph (Kieker)

- ▶ Architecture Discovery & Program Comprehension
  - ▶ Aggregation based on the package hierarchy
- ▶ Performance Analysis
  - ▶ Number of edges (calls) in combination with response times

### 2. Visualizing the manipulated graph

# Further analysis and visualization via Gephi

## Approach

Data Table

Nodes Edges Configuration Add node Add edge Search/Replace Import Spreadsheets

Nodes	Id
wishes_to_export() min: 0ms, avg: 0.00ms, max: 0ms	depNode_117
who_filter() min: 0ms, avg: 0.00ms, max: 0ms	depNode_128
show_columns() min: 0ms, avg: 0.00ms, max: 0ms	depNode_106
render_title() min: 3ms, avg: 3.00ms, max: 3ms	depNode_169
render_title() min: 1ms, avg: 2.50ms, max: 4ms	depNode_149
render_title() min: 0ms, avg: 0.00ms, max: 0ms	depNode_150
render_tab_title() min: 0ms, avg: 0.00ms, max: 0ms	depNode_36
render_tab_title() min: 0ms, avg: 0.00ms, max: 0ms	depNode_64
render_items() min: 505ms, avg: 505.00ms, max: 505ms	depNode_135
render_import_bar() min: 21ms, avg: 21.00ms, max: 21ms	depNode_132

# Further analysis and visualization via Gephi

## Approach

Data Table

Nodes Edges Configuration Add node Add edge Search/Replace

Source	Target	Type	Id	Label
depNode_1	depNode_11/	Directed	1//5	1
depNode_1	depNode_125	Directed	1789	2
depNode_1	depNode_111	Directed	1792	1
depNode_1	depNode_116	Directed	1804	1
depNode_1	depNode_170	Directed	1807	6
depNode_1	depNode_101	Directed	1818	131
depNode_1	depNode_118	Directed	1826	1
depNode_1	depNode_2	Directed	1838	10
depNode_1	depNode_115	Directed	1886	1
depNode_1	depNode_155	Directed	1870	1
depNode_1	depNode_173	Directed	1797	1

# Case Study

**Let's evaluate our approach!**

## Performance analysis of EPrints 3.3.12 using Kieker

- ▶ Potential bottleneck detection for upcoming release version 4
- ▶ Evaluation in collaboration with the development team
- ▶ Software maintenance & modernization
- ▶ Different instrumentation levels are possible
- ▶ Combined visualization: GraphViz and Gephi



## Case Study

- ▶ Is it possible to identify an architecture? (Architectural Discovery)
- ▶ Can we support other developers by providing information about the software? (Program Comprehension)
- ▶ Can we detect potential bottlenecks? (Performance Analysis)

# Initial Request

## Case Study

Manage deposits - Default

vmdev1.eprints.org/cgi/users/home?screen=items

Apps FS OLAT Debian Sources ... unilink U1 unilink U2 SUSU eprintsVM: Activ... eprintsVM - Web... DevGuide/kdb - ... Stundenplan SS14 Ruby on Rails API

eprints repository software

Home About Browse

Logged in as Unnamed user with email [st2@ecs.soton.ac.uk](mailto:st2@ecs.soton.ac.uk) | Manage deposits | Manage records | Profile | Saved searches | Review | Admin | Edit page phrases | Logout

Search

### Manage deposits

[Help](#)

New Item

Import from:  Import

User Workarea,  Under Review,  Live Archive,  Retired.

Last Modified	Title	Item Type	Item Status	
30 Apr 2014 13:56	Thomas Ker of Redden's trip to the Low Countries, 1620	Book Section	User Workarea	
30 Apr 2014 13:50	Playing with Haddon's string figures	Article	User Workarea	
30 Apr 2014 13:50	Outside in: making sense of the deliberate concealment of garments within buildings	Article	User Workarea	
30 Apr 2014 13:50	[Response to Philip Whalen's article 'Burgundian regionalism and French Republican commercial culture at the 1937 Paris International Exposition']	Article	User Workarea	

Abstract Add Column

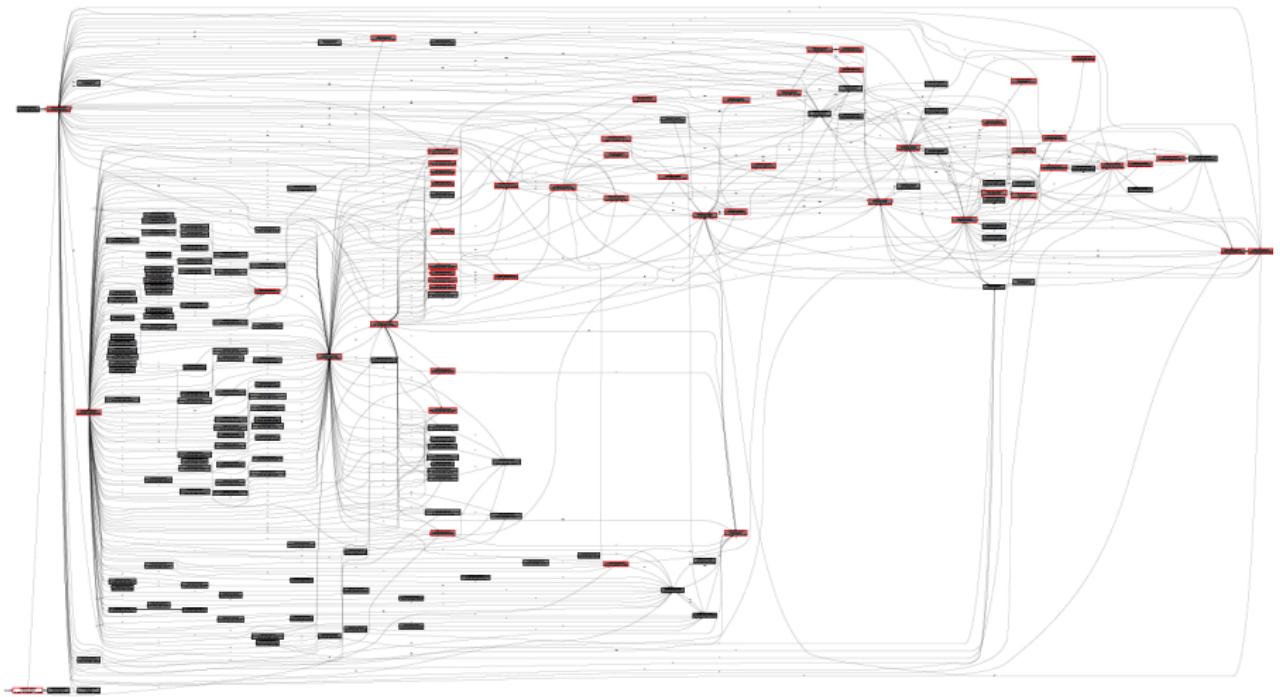
Default VM is powered by [EPrints 3](#) which is developed by the [School of Electronics and Computer Science](#) at the University of Southampton. [More information and software credits](#).

# Case Study

## Architecture Discovery

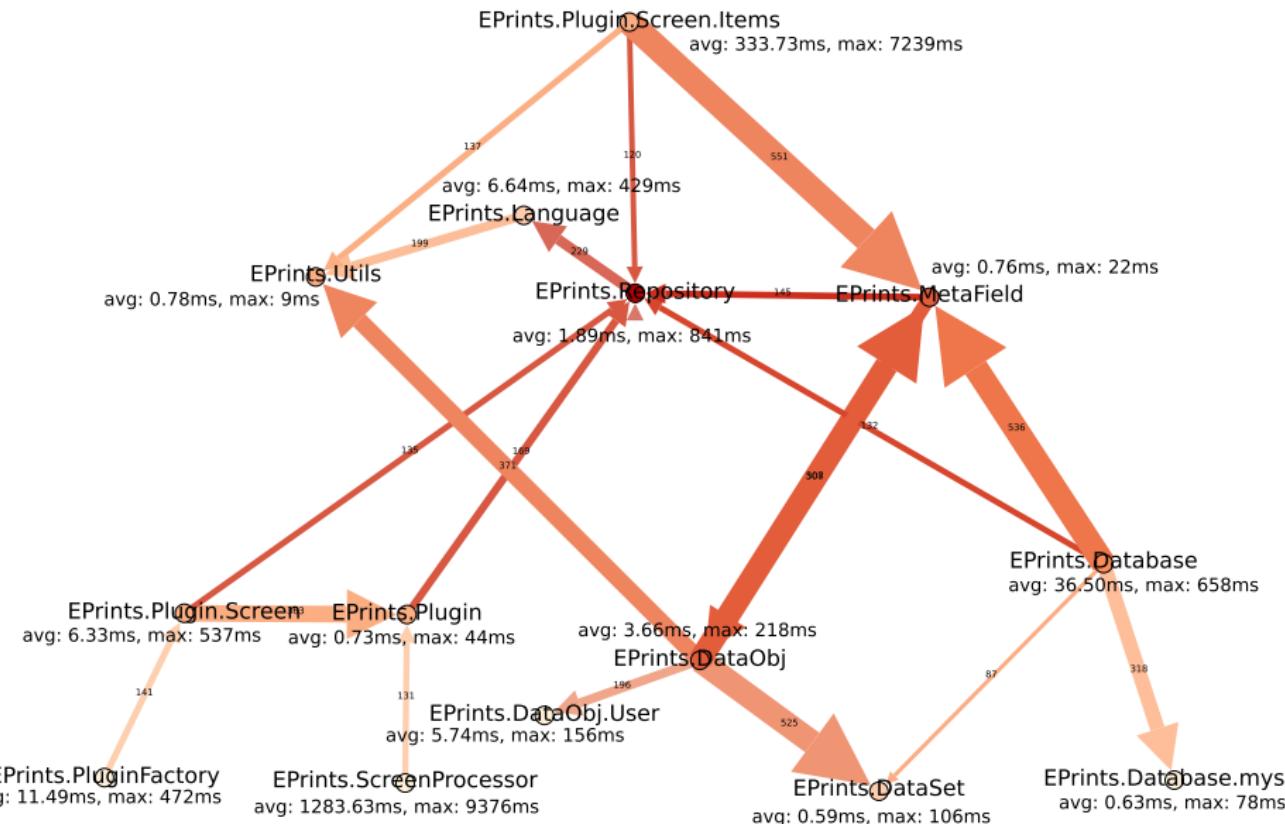
# Batch Visualization via Graphviz

Case Study ▷ Architecture Discovery



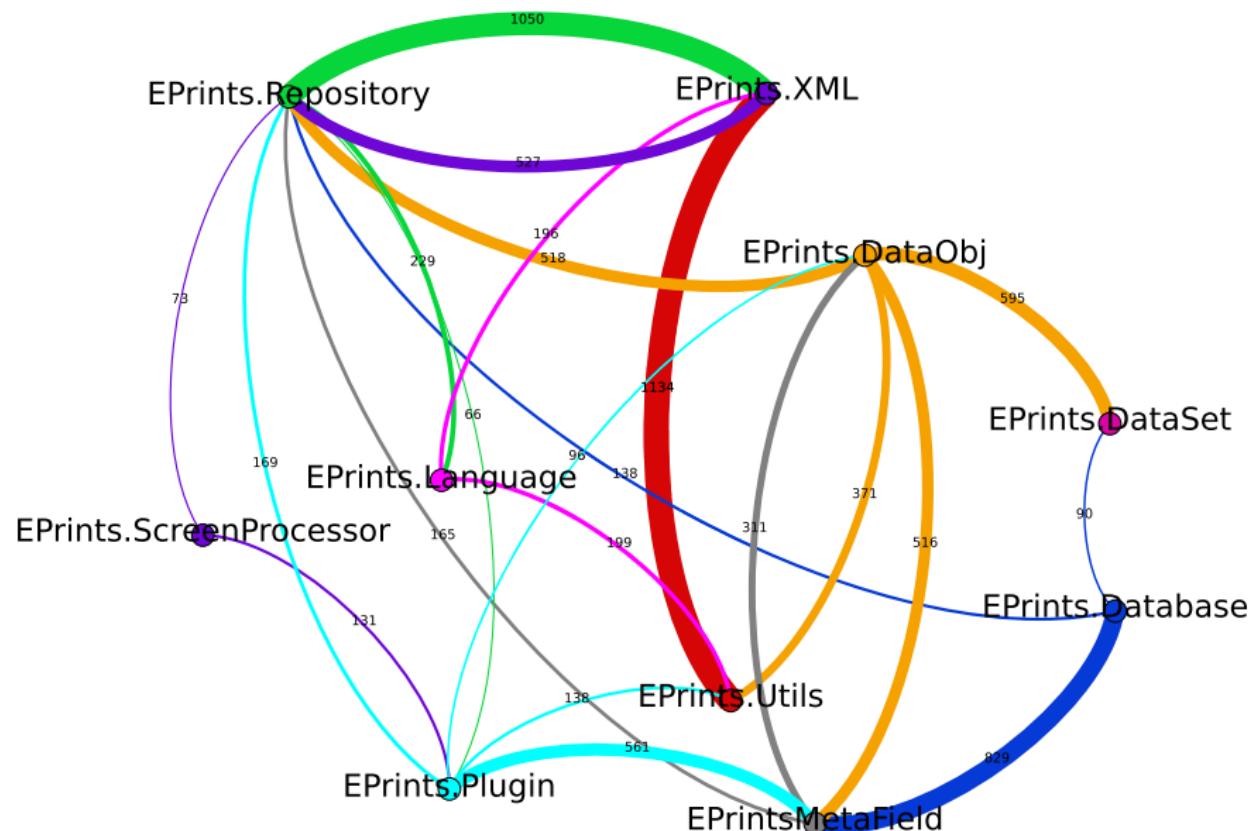
# Gephi: System Architecture Level

## Case Study ▷ Architecture Discovery



# Gephi: System Architecture Level

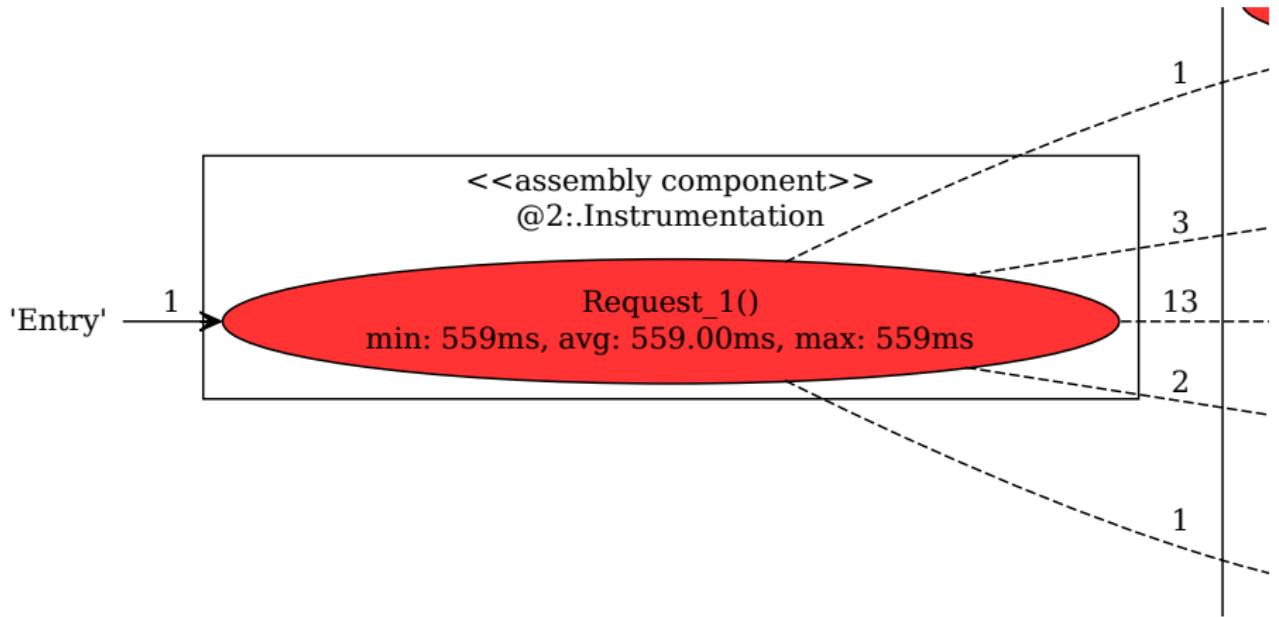
## Case Study ▷ Architecture Discovery



# Case Study

## Performance Bottleneck Identification

## Case Study ▷ Performance Bottleneck Identification



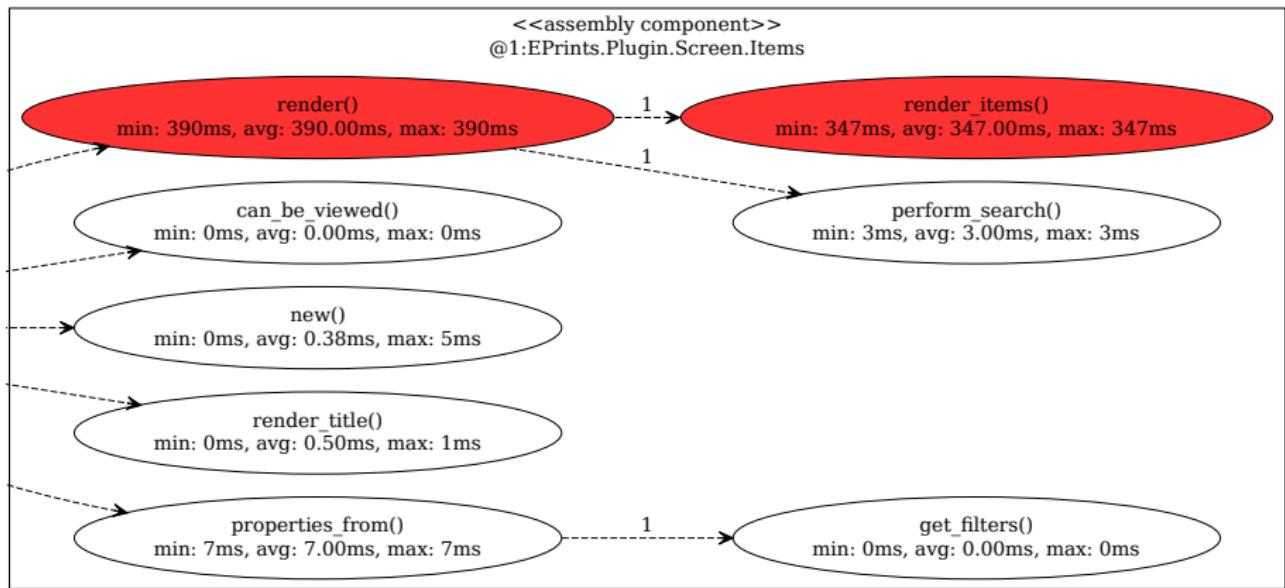
# GraphViz: Operations - EPrints.Screen.Items

C | A | U

Christian-Albrechts-Universität zu Kiel

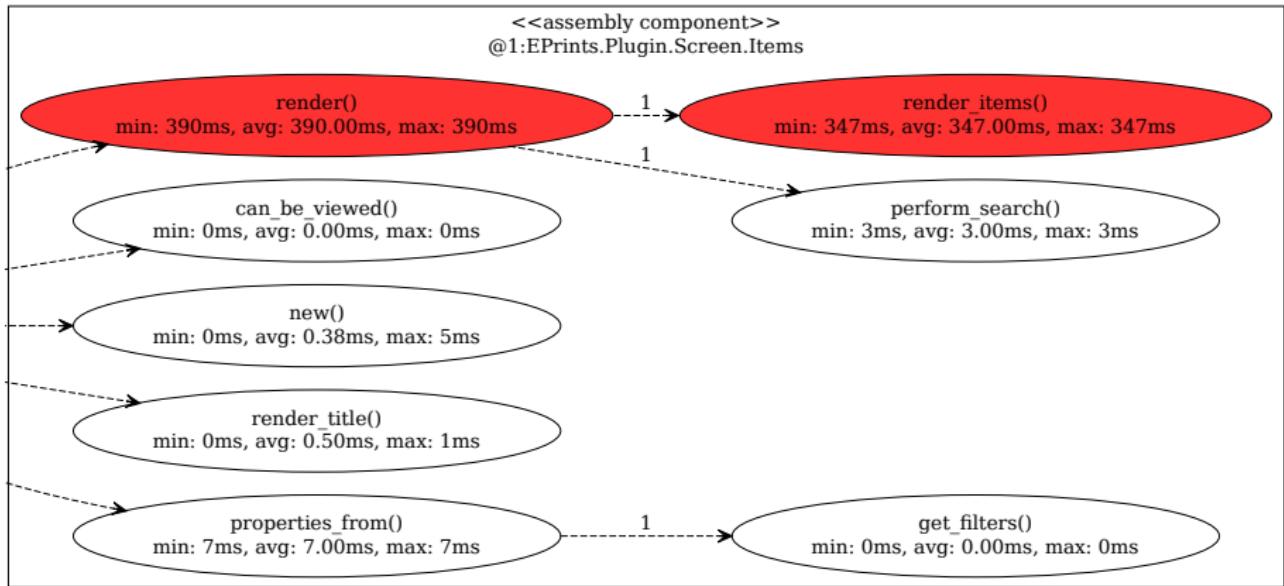
Technische Fakultät

## Case Study ▷ Performance Bottleneck Identification



# GraphViz: Operations - EPrints.Screen.Items

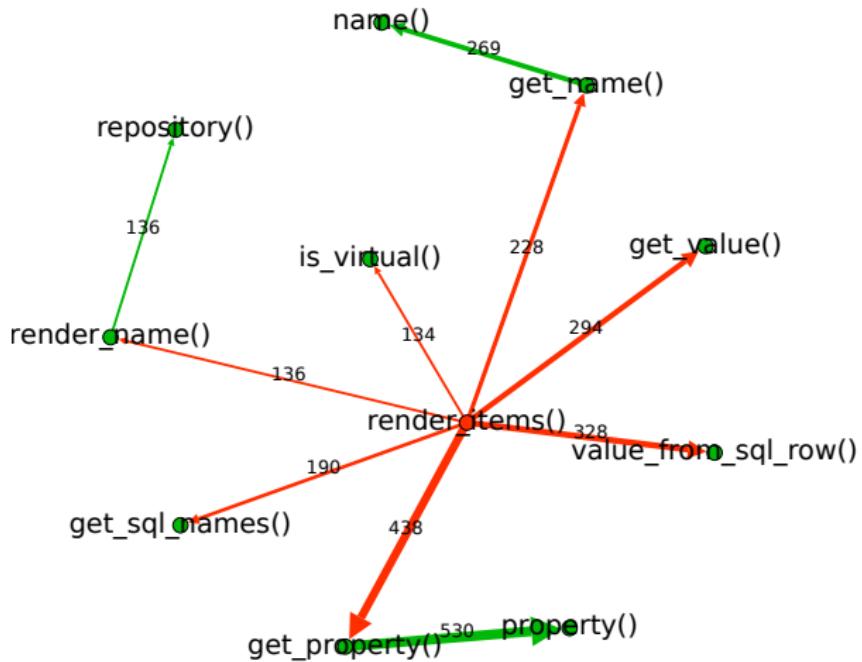
## Case Study ▷ Performance Bottleneck Identification



Potential bottleneck: database-oriented operations

# Gephi: Operations - EPrints.MetaField

Case Study ▷ Performance Bottleneck Identification



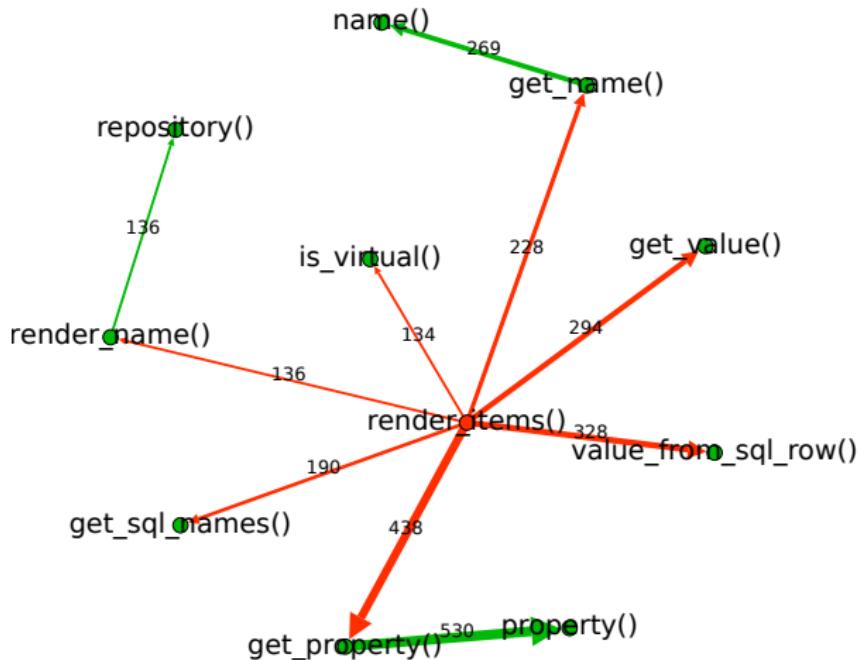
# Gephi: Operations - EPrints.MetaField

C | A | U

Christian-Albrechts-Universität zu Kiel

Technische Fakultät

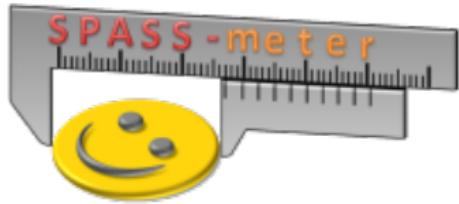
Case Study ▷ Performance Bottleneck Identification



Potential bottleneck: database-oriented operations

# Related Work

## Related Work

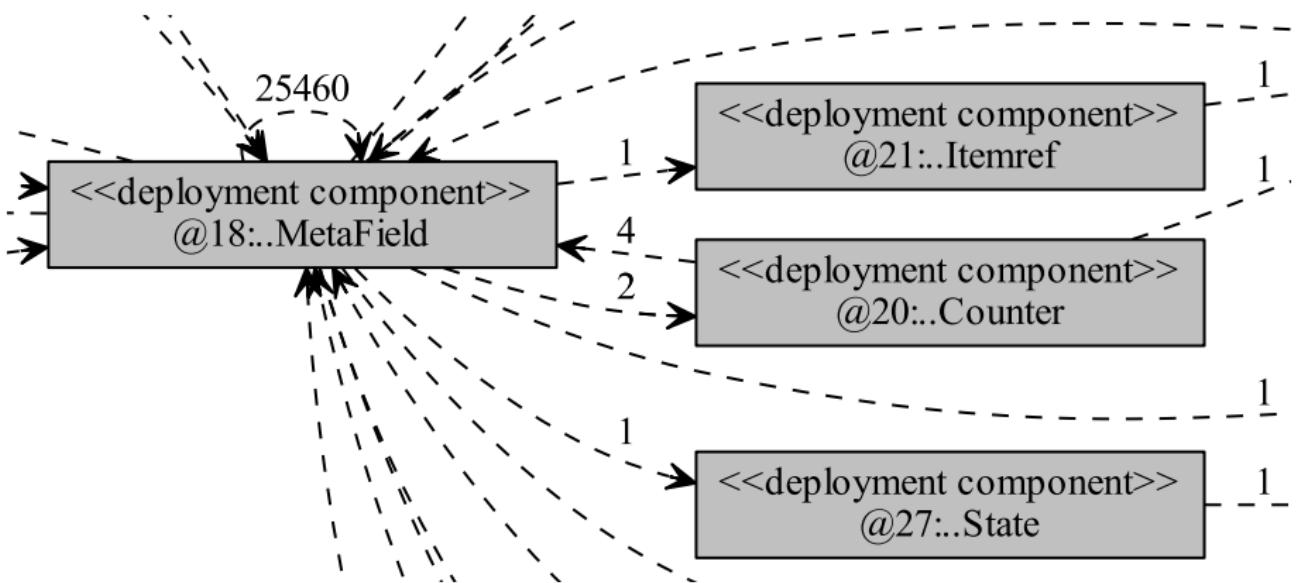


## Summary

- ▶ Well-structured workflow for a performance analysis
- ▶ Verification through a case study of an established application
- ▶ Detection of performance issues → basis for recommendations
  
- Continuous integration

# Infinite Loop Detection in Version 4

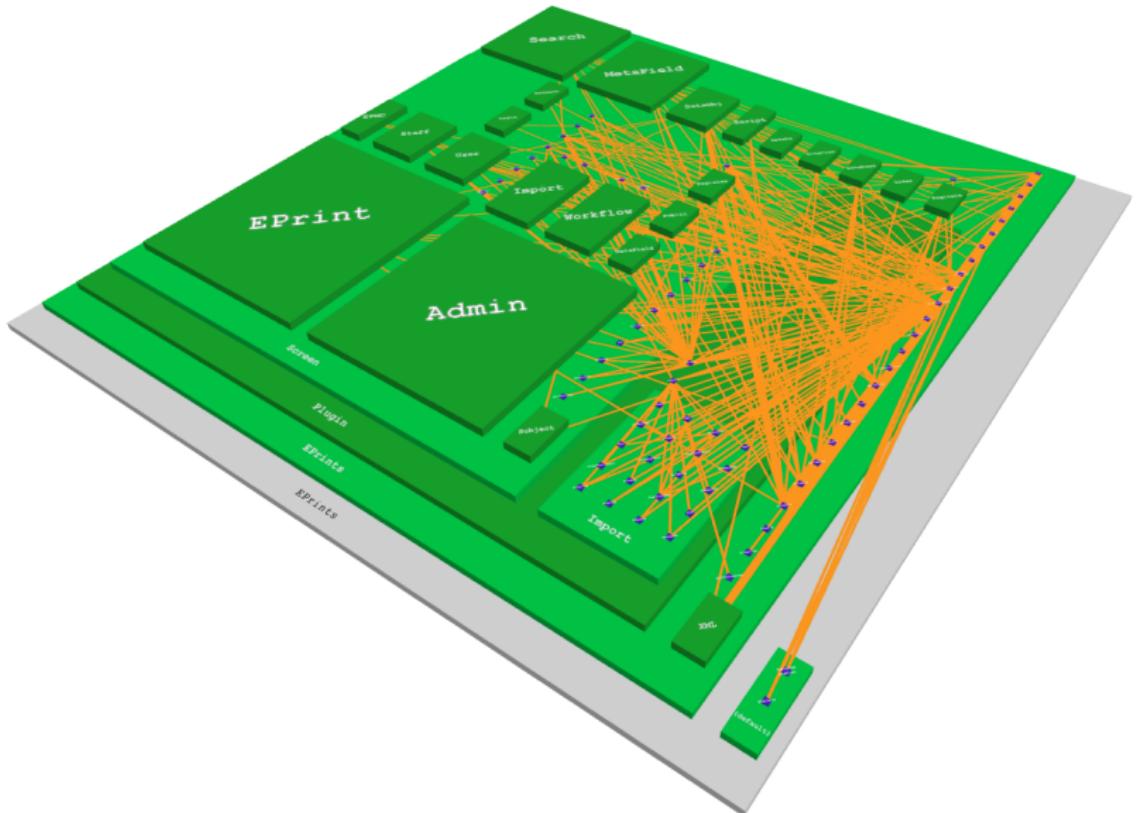
Outlook



# What happened after our Case Study?

# Application-level Perspective in ExplorViz

Outlook



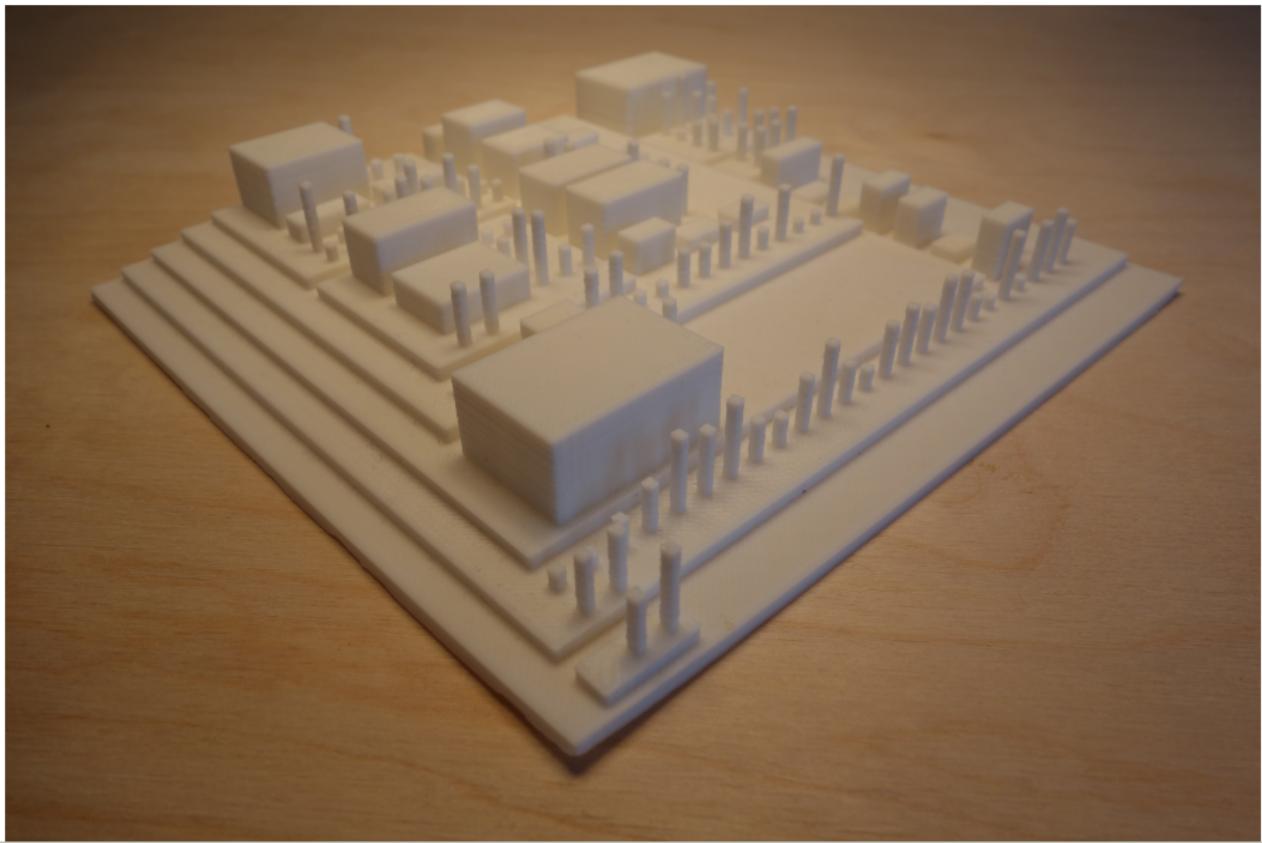
# Physical 3D Model based on ExplorViz

C | A | U

Christian-Albrechts-Universität zu Kiel

Technische Fakultät

Outlook



# Physical 3D Model based on ExplorViz

C | A | U

Christian-Albrechts-Universität zu Kiel

Technische Fakultät

Outlook

