

# Towards a Dependability Control Center for Large Software Landscapes

Florian Fittkau, André van Hoorn, and Wilhelm Hasselbring

EDCC 2014 @ Newcastle upon Tyne, UK

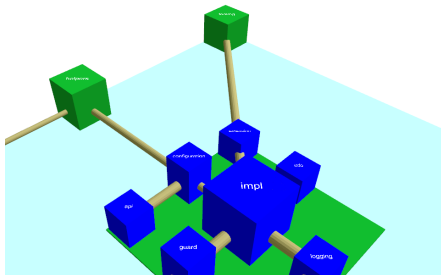
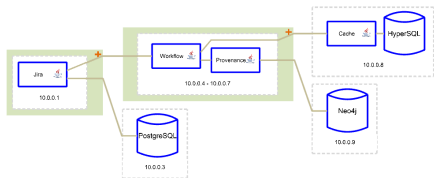
2014-05-14

ExplorViz

- ▶ Dependability: manual management tedious in large systems (e.g., clouds)
- ▶ Automatic techniques proposed
- ▶ Operators often mistrust fully-automatic systems

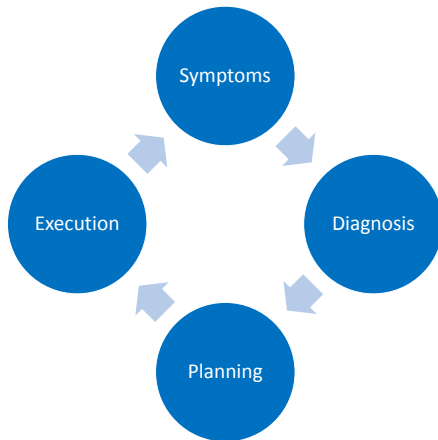
- ▶ Dependability: manual management tedious in large systems (e.g., clouds)
- ▶ Automatic techniques proposed
- ▶ Operators often mistrust fully-automatic systems
- ▶ **Vision: semi-automatic control center**

# ExplorViz



- ▶ Interactive approach for the live, explorable visualization of software landscapes [FWWH13]

► Four perspectives:



# 1. Phase: Symptoms

## Envisioned Control Center

← → ↻ 127.0.0.1:8888

Dev Mode On Dev Mode Off

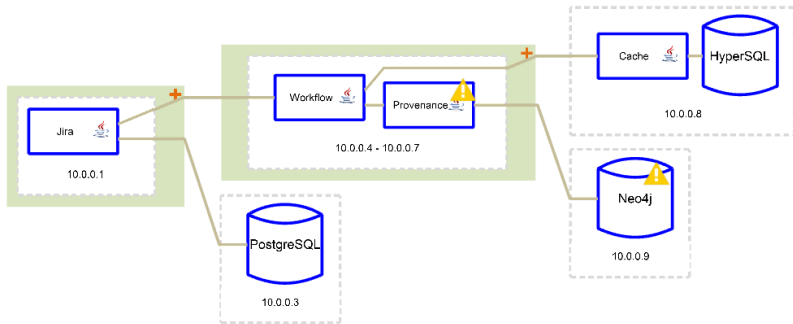
Explorer Code Viewer Configuration

FPS: 60

FPS

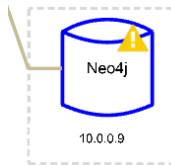
# Symptoms

# ExplorViz



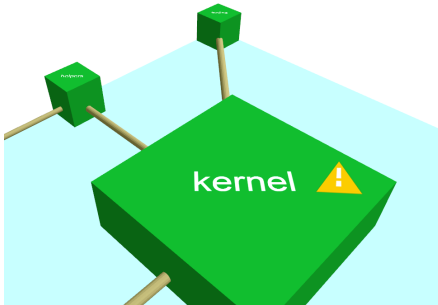
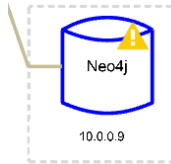
# 2. Phase: Diagnosis

Envisioned Control Center



# 2. Phase: Diagnosis

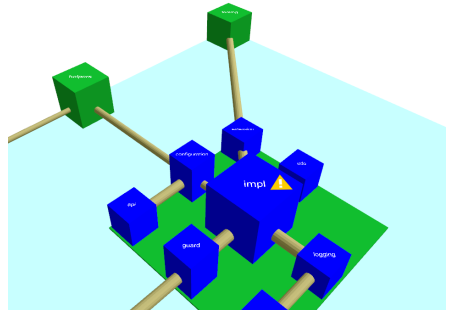
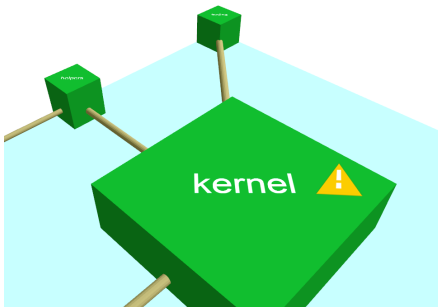
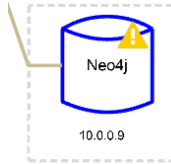
Envisioned Control Center





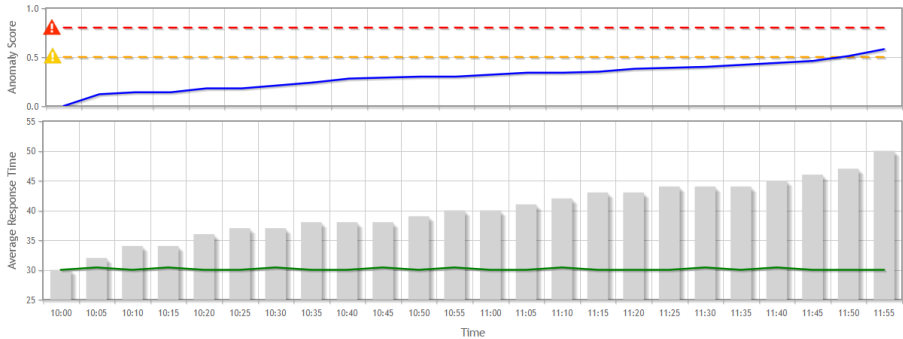
# 2. Phase: Diagnosis

Envisioned Control Center



## 2. Phase: Diagnosis (cont'd)

### Envisioned Control Center



**Warning** ✕

The software landscape violates its requirements for response times.

**Countermeasure**

It is suggested to start a new node of type 'm1.small' with the application 'Neo4J' on it.

**Consequence**

After the change, the response time is improved and the operating costs increase by 5 Euro per hour.

Start the instance?

Automatic change dialog

# 3. Phase: Planning (cont'd)

## Envisioned Control Center

← → ↻ 127.0.0.1:8888

Dev Mode On Dev Mode Off

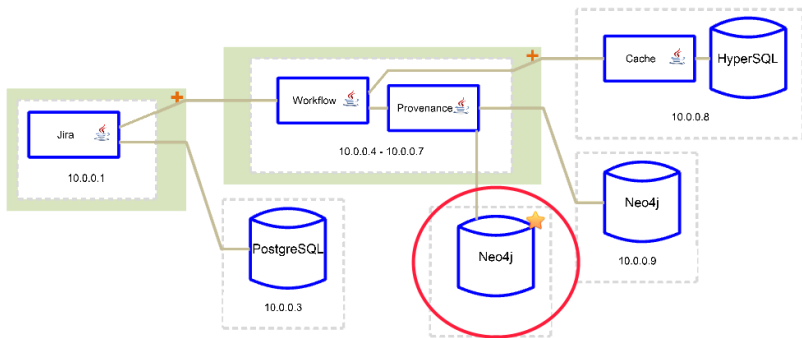
Explorer

Code Viewer Configuration

FPS: 60

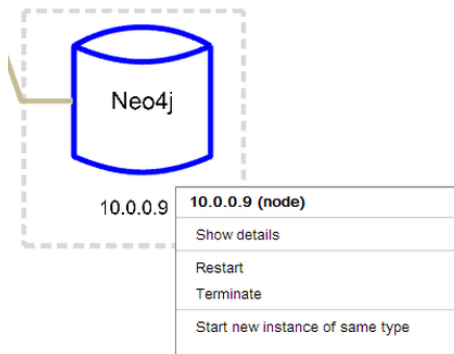
FPS

# Planning ExplorViz



### 3. Phase: Planning (cont'd)

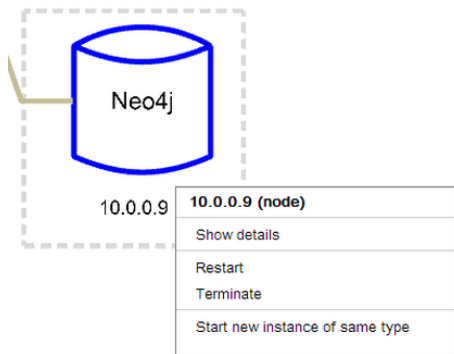
Envisioned Control Center



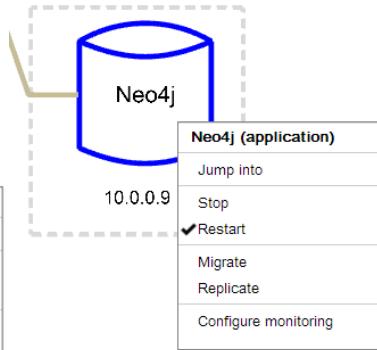
Node context menu

### 3. Phase: Planning (cont'd)

Envisioned Control Center



Node context menu



Application context menu

- ▶ Pushing execute button in the planning perspective
- ▶ Execution perspective is opened
- ▶ Shows **what is planned** and **what has already been conducted**

- ▶ Monitoring
  - ▶ Kieker [vHWH12]
- ▶ Online performance anomaly detection
  - ▶  $\Theta$ PAD [Bie12, Fro13]
- ▶ Root cause localization
  - ▶ RanCorr [MRvHH09]
- ▶ Online capacity management
  - ▶ SLAStic [vMvHH11]



- ▶ Cloud management
  - ▶ E.g., Amazon CloudWatch, Microsoft Azure auto-scaling
  - ▶ No manual refinement of the reconfiguration plan

- ▶ Cloud management
  - ▶ E.g., Amazon CloudWatch, Microsoft Azure auto-scaling
  - ▶ No manual refinement of the reconfiguration plan
- ▶ Application performance monitoring (APM)
  - ▶ E.g., AppDynamics, ExtraHop, or SPASS-meter [ES12]
  - ▶ Most tools only provide monitoring and reactive analysis

- ▶ Cloud management
  - ▶ E.g., Amazon CloudWatch, Microsoft Azure auto-scaling
  - ▶ No manual refinement of the reconfiguration plan
- ▶ Application performance monitoring (APM)
  - ▶ E.g., AppDynamics, ExtraHop, or SPASS-meter [ES12]
  - ▶ Most tools only provide monitoring and reactive analysis
- ▶ MAPE-K control loop tools
  - ▶ E.g., Rainbow [GCH<sup>+</sup>04] or TRAP
  - ▶ Large part focuses on automatic adaptation

## Summary

- ▶ Envisioned semi-automatic control center for cloud dependability basing on ExplorViz<sup>1</sup>
- ▶ Open source tool ExplorViz available at <http://explorviz.net>
- ▶ Visual plug-in-based integration platform for dependability management approaches

ExplorViz



---

<sup>1</sup>[florian.fittkau@email.uni-kiel.de](mailto:florian.fittkau@email.uni-kiel.de)

## Summary

- ▶ Envisioned semi-automatic control center for cloud dependability basing on ExplorViz<sup>1</sup>
- ▶ Open source tool ExplorViz available at <http://explorviz.net>
- ▶ Visual plug-in-based integration platform for dependability management approaches

ExplorViz



## Future Work

- ▶ Provide an implementation for our control center concept
- ▶ Develop plug-ins to integrate a number of reasonable dependability management approaches

---

<sup>1</sup>[florian.fittkau@email.uni-kiel.de](mailto:florian.fittkau@email.uni-kiel.de)



**Tillmann Carlos Bielefeld.**

Online performance anomaly detection for large-scale software systems.  
 Diploma thesis, Kiel University, Germany, 2012.



**Holger Eichelberger and Klaus Schmid.**

Erhebung von Produkt-Laufzeit-Metriken: Ein Vergleich mit dem SPASS-Meter-Werkzeug.  
 In Proceedings of the DASMA Metrik Kongress (MetriKon '12), pages 171–180. Shaker Verlag, November 2012.



**Tom Frotscher.**

Architecture-based multivariate anomaly detection for software systems.  
 Masterarbeit, Kiel University, Oktober 2013.



**Florian Fittkau, Jan Waller, Christian Wulf, and Wilhelm Hasselbring.**

Live trace visualization for comprehending large software landscapes: The ExplorViz approach.  
 In Proc. VISSOFT 2013, 2013.



**D. Garlan, Shang-Wen Cheng, An-Cheng Huang, B. Schmerl, and P. Steenkiste.**

Rainbow: architecture-based self-adaptation with reusable infrastructure.  
Computer, 37(10):46–54, Oct 2004.



**Nina S. Marwede, Matthias Rohr, André van Hoorn, and Wilhelm Hasselbring.**

Automatic failure diagnosis in distributed large-scale software systems based on timing behavior anomaly correlation.

In [Proc. CSMR 2009](#), pages 47–57. IEEE, 2009.



André van Hoorn, Jan Waller, and Wilhelm Hasselbring.

Kieker: A framework for application performance monitoring and dynamic software analysis.

In [Proc. ICPE 2012](#), pages 247–248. ACM, 2012.



Robert von Massow, André van Hoorn, and Wilhelm Hasselbring.

Performance simulation of runtime reconfigurable component-based software architectures.

In [Proc. ECSA 2011](#), pages 43–58. Springer, 2011.