

# Live Visualization of Large Software Landscapes for Ensuring Architecture Conformance

Florian Fittkau, Phil Stelzer, and Wilhelm Hasselbring

SESoS 2014 @ Vienna, Austria

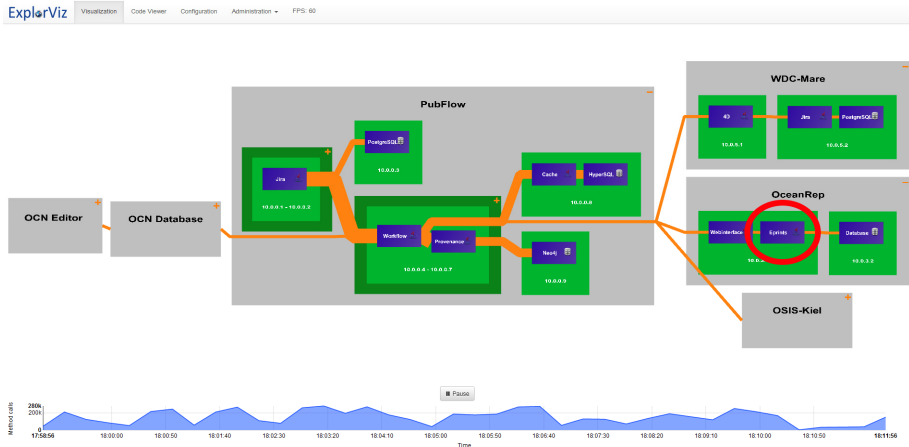
2014-08-26

ExplorViz

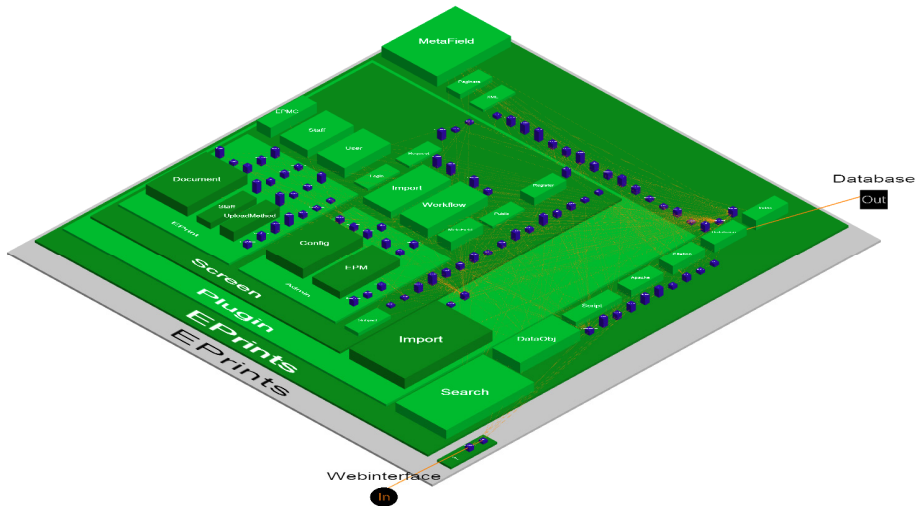
- ▶ Large software landscapes are complex Systems-of-Systems
- ▶ Long-lived systems often tend to **architectural erosion**
- ▶ Continuous monitoring - but huge generated data

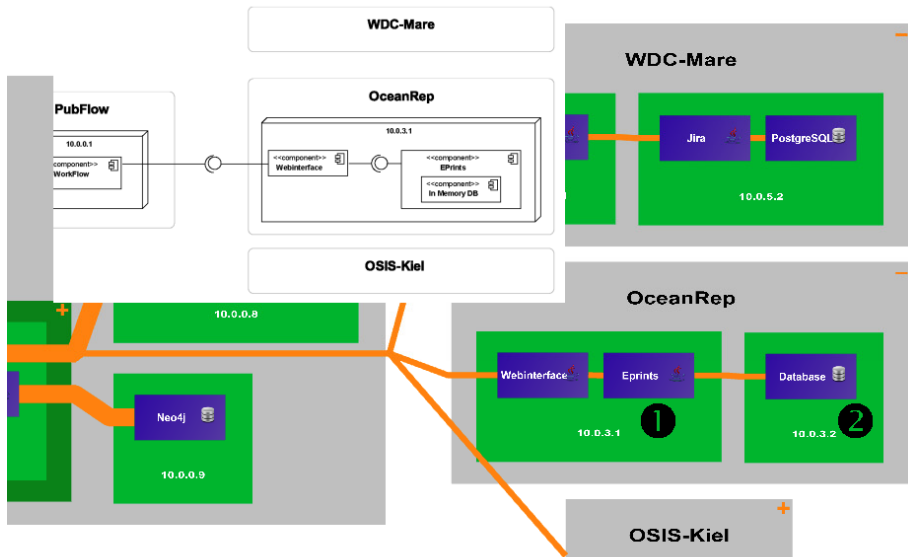
- ▶ Large software landscapes are complex Systems-of-Systems
  - ▶ Long-lived systems often tend to **architectural erosion**
  - ▶ Continuous monitoring - but huge generated data
- ⇒ Scalable, live visualization for checking landscape architecture conformance

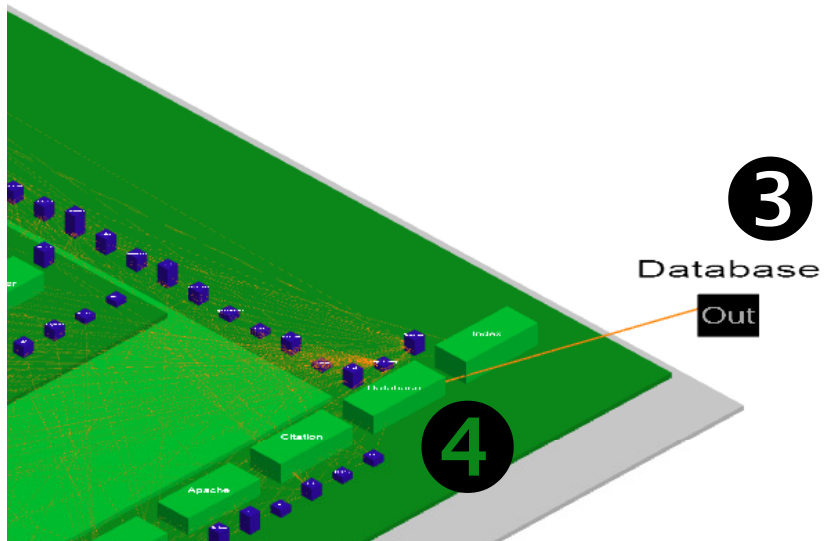
ExplorViz



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







- ▶ **Low overhead and scalability to large systems** for our monitoring solution are **preconditions** of our architecture conformance checking approach
- ▶ We use Cloud infrastructures (elastic, external analysis of monitoring data)
- ▶ Here: **Analysis of our monitoring solution** (overhead and scalability) using Cloud infrastructures
- ▶ Future work: Analysis of the scalability of our visualization

- ▶ *SNoMonitoring*: Generation of workload with JMeter on JPetStore instances **without monitoring**
- ▶ *SWithMonitoring*: Same as *SNoMonitoring* but **with monitoring enabled**

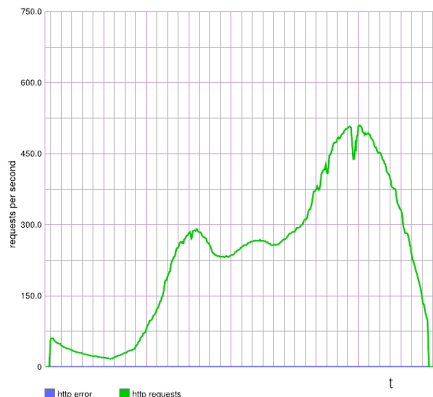


Figure 1 : Workload curve

### ► Private cloud (OpenStack)<sup>1</sup>

<sup>1</sup>8 servers [2x Intel Xeon (2.8GHz, 8 cores), 128 GB RAM]

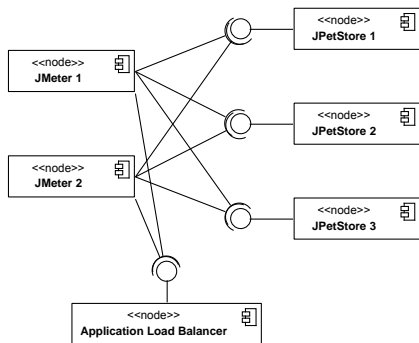


Figure 2 : Deployment sketch in *SNoMonitoring*

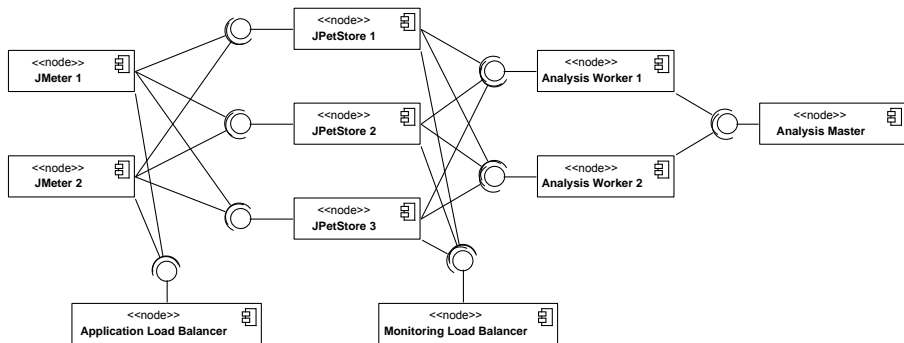


Figure 3 : Deployment sketch in *SWithMonitoring*

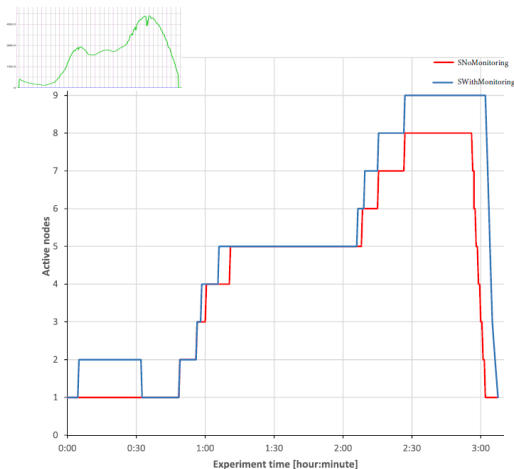


Figure 4 : Node allocation of JPetStore instances in *SNoMonitoring* and *SWithMonitoring*

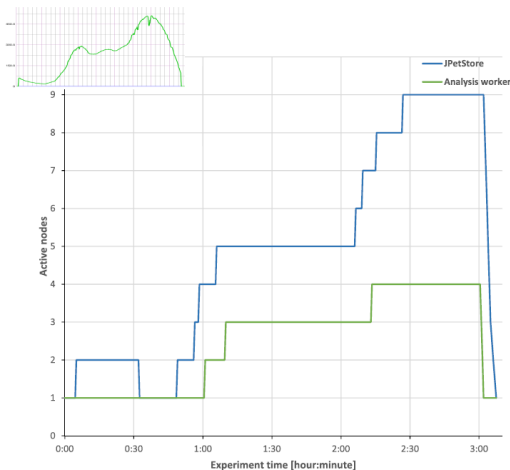


Figure 5 : Node allocation of JPetStore instances and analysis worker instances in *SWithMonitoring*

- ▶ Only on our private cloud
- ⇒ For external validity: **other environments and applications**
- ▶ Duration of our scenarios may be insufficient
- ⇒ **Longer studies** have to be conducted

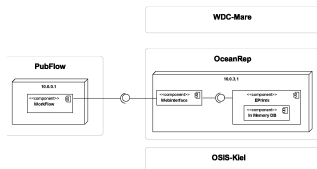
- ▶ Architecture Conformance Checking
  - ▶ Dependency-structure matrices, source code query languages, and reflexion models [PTV<sup>+</sup>10]
  - ⇒ Often **no dynamic analysis** and often only target **one system**

- ▶ Architecture Conformance Checking
  - ▶ Dependency-structure matrices, source code query languages, and reflexion models [PTV<sup>+</sup>10]
  - ⇒ Often **no dynamic analysis** and often only target **one system**
- ▶ Software Visualization
  - ▶ Application performance monitoring (APM) tools like AppDynamics
  - ⇒ Most tools are **limited to visualizing nodes and applications**
  - ▶ Single application: comparison in [FWWH13]

- ▶ Architecture Conformance Checking
  - ▶ Dependency-structure matrices, source code query languages, and reflexion models [PTV<sup>+</sup>10]
  - ⇒ Often **no dynamic analysis** and often only target **one system**
- ▶ Software Visualization
  - ▶ Application performance monitoring (APM) tools like AppDynamics
  - ⇒ Most tools are **limited to visualizing nodes and applications**
  - ▶ Single application: comparison in [FWWH13]
- ▶ Monitoring Solutions
  - ▶ Monitoring-as-a-Service solution [MLS10] for whole cloud topology
  - ▶ Parallel analysis infrastructure [BN03] using shared storage

## Summary

- ▶ ExplorViz provides **abstractions** intended for **visual scalability** to check landscape architecture conformance checking
- ▶ If you are interested: Open source<sup>2</sup> and contact<sup>3</sup>



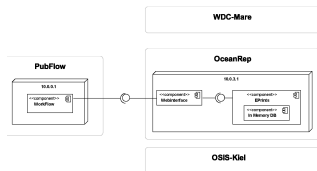
ExplorViz

<sup>2</sup><http://explorviz.net>

<sup>3</sup>[florian.fittkau@email.uni-kiel.de](mailto:florian.fittkau@email.uni-kiel.de)

## Summary

- ▶ ExplorViz provides **abstractions** intended for **visual scalability** to check landscape architecture conformance checking
- ▶ If you are interested: Open source<sup>2</sup> and contact<sup>3</sup>



ExplorViz

## Future Work

- ▶ **Controlled experiment** for scalability evaluation of visualization
- ▶ Provide modeling capabilities of conceptual architecture

<sup>2</sup><http://explorviz.net>

<sup>3</sup>[florian.fittkau@email.uni-kiel.de](mailto:florian.fittkau@email.uni-kiel.de)



Holger Brunst and Wolfgang E. Nagel.

Scalable performance analysis of parallel systems: Concepts and experiences.

In Proc. of the 10th Conf. on Parallel Computing: Software Technology, Algorithms, Architectures, and Applications. Elsevier, 2003.



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

Live trace visualization for comprehending large software landscapes: The ExplorViz approach.

In Proc. of the 1st Int. Working Conf. on Software Visualization (VISOFT 2013), September 2013.



Shicong Meng, Ling Liu, and Vijayaraghavan Soundararajan.

Tide: Achieving self-scaling in virtualized datacenter management middleware.

In Proc. of the 11th Int. Middleware Conf. (Middleware 2010). ACM, 2010.



L. Passos, R. Terra, M.T. Valente, R. Diniz, and N. Mendonca.

Static architecture-conformance checking: An illustrative overview.

IEEE Software, 27, September 2010.