

# On the Modernization of ExplorViz towards a Microservice Architecture

**EMLS 2018**

5<sup>th</sup> Collaborative Workshop on Evolution  
and Maintenance of Long-Living Software Systems

March 06, 2018

Christian Zirkelbach, Alexander Krause, and Wilhelm Hasselbring – Kiel University



# ➤ Overview

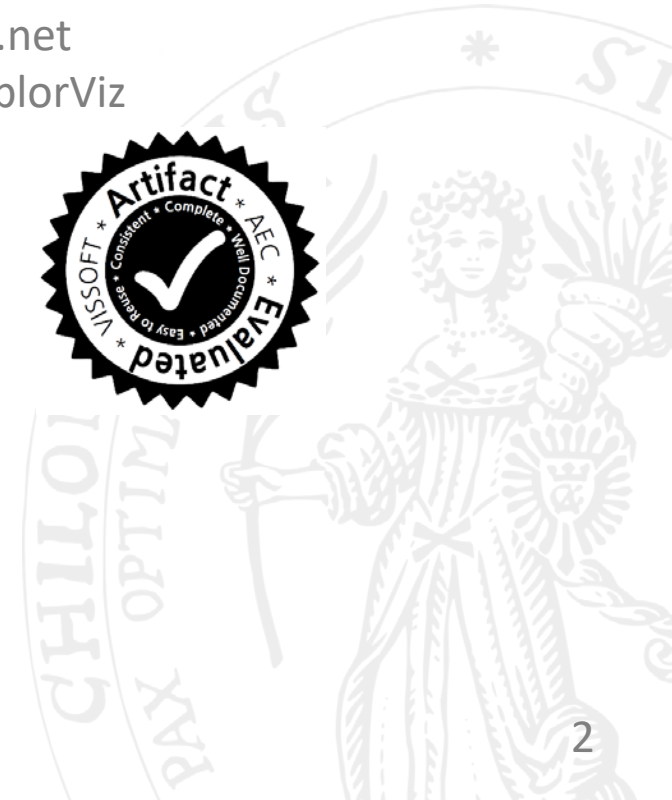
## Live trace visualization for large software landscapes based on monitoring

- 🎯 Main purpose:  
Program- and system comprehension  
for developers
- Started as a Ph.D project in 2012
- Free license (Apache License 2.0)
- Open-source from the beginning
- Repeatedly extended since the beginning
  - more than 25 bachelor's and master's theses
  - multiple student projects

# ExplorViz

<https://www.explorviz.net>

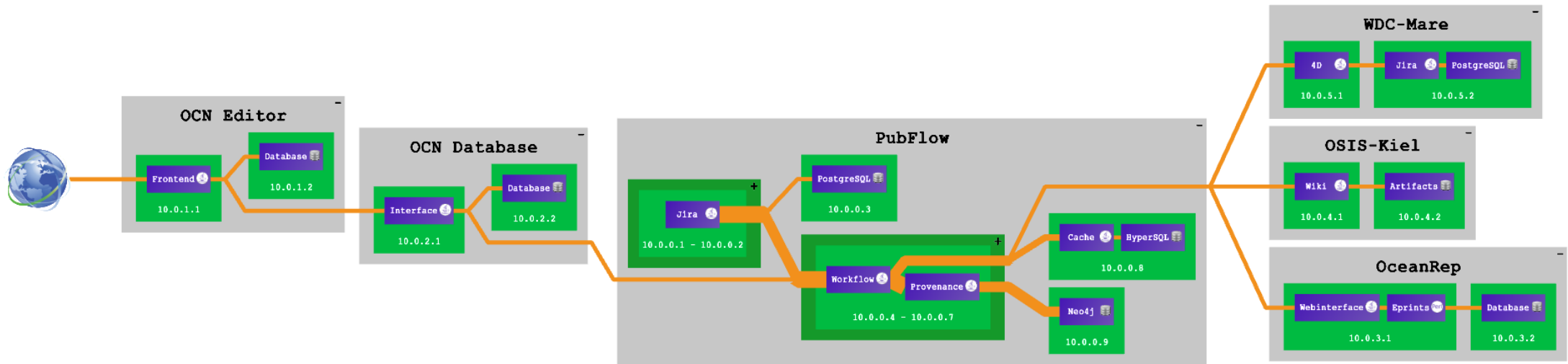
<https://github.com/ExplorViz>



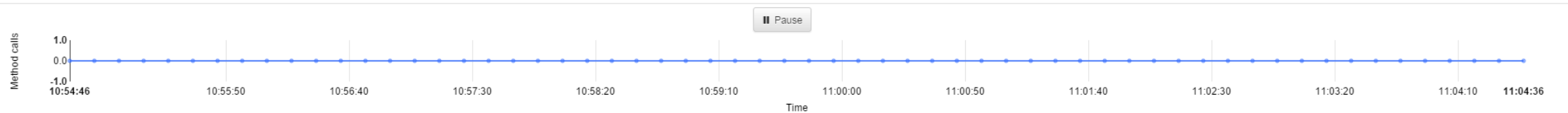
# Visualization (Landscape-Level)

View Events

View Exceptions



Source → Target



# Visualization (Application-Level)

ExplorViz

Visualization

Tutorial

Signed in as DemoUser (logout)

← Back to Landscape

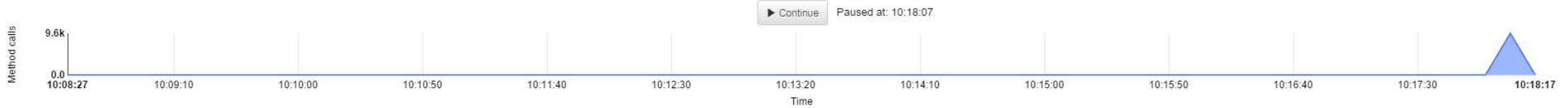
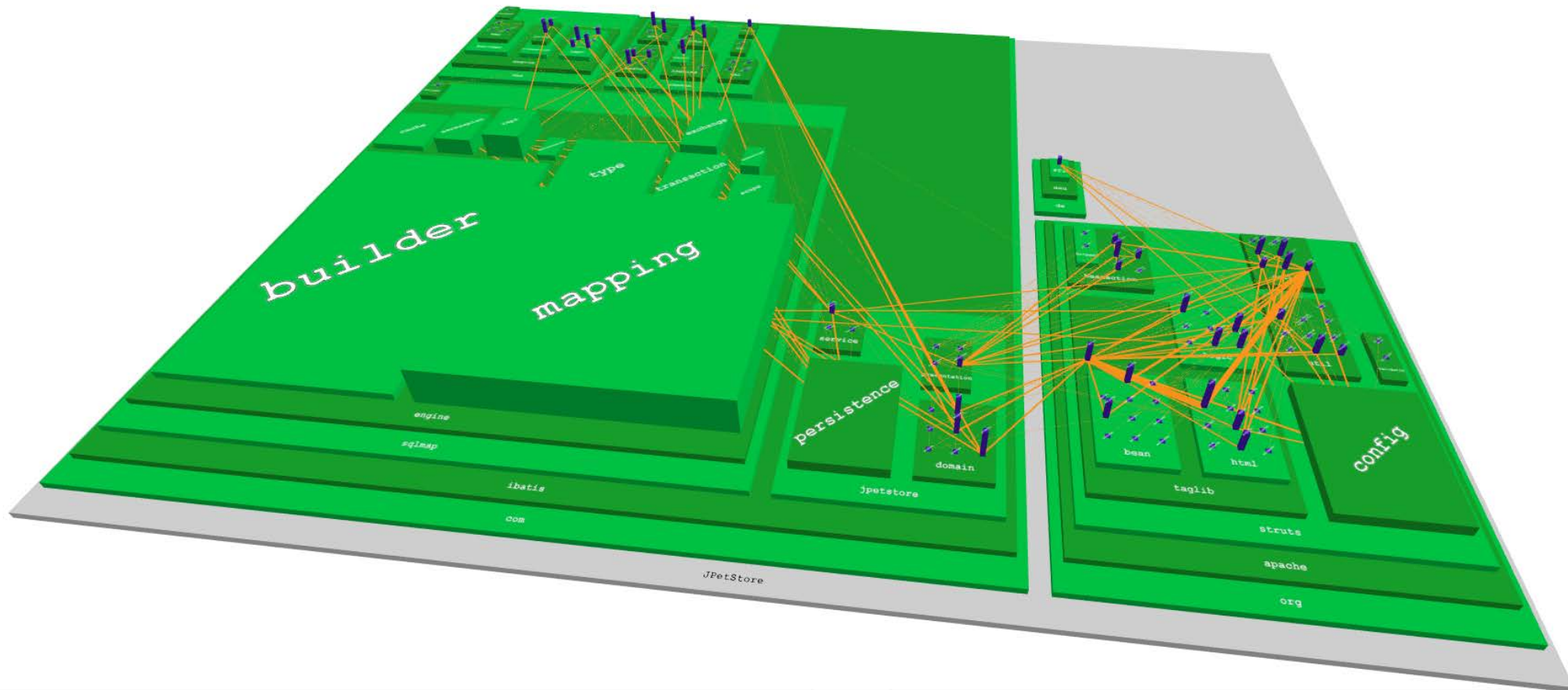
Open All Components

Export 3D Model

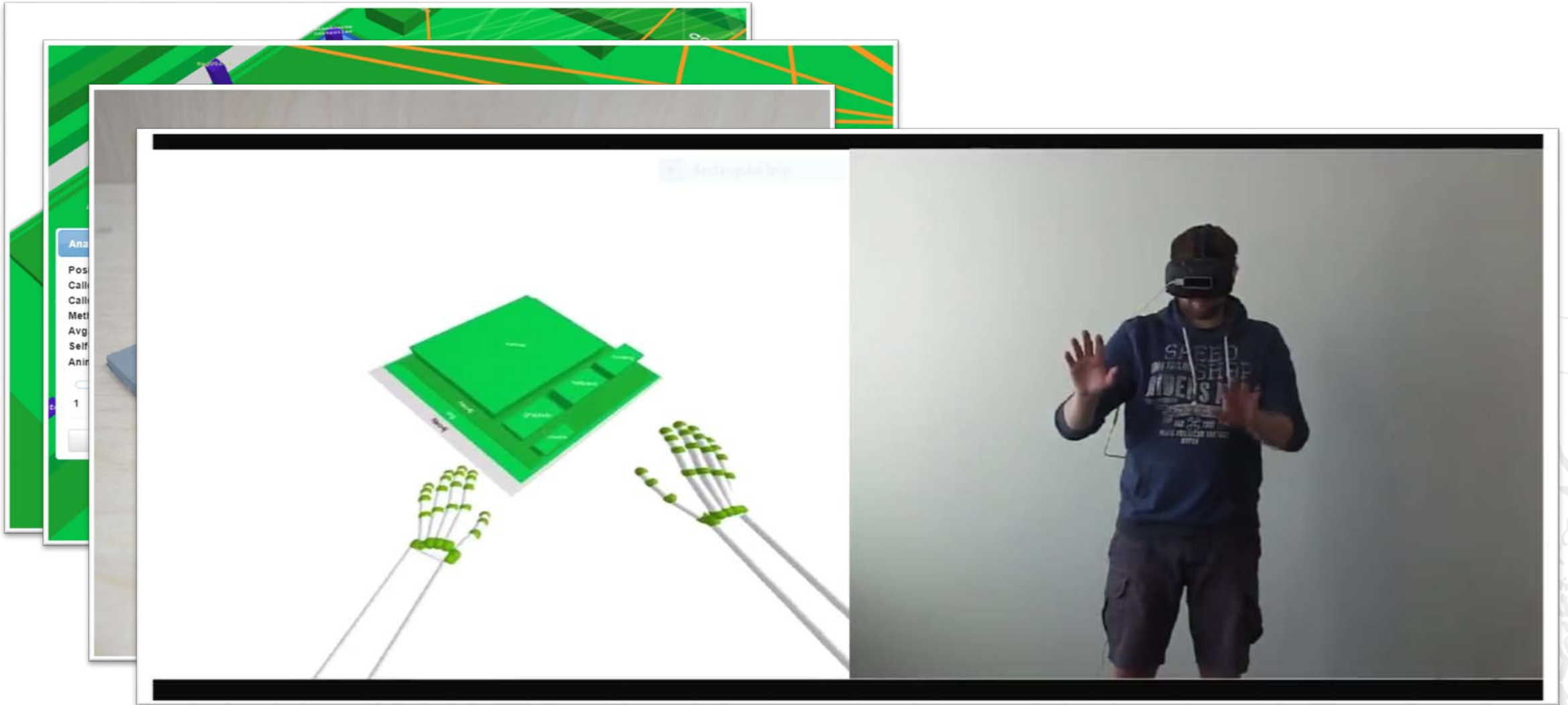
Performance Analysis

Virtual Reality Mode

Database Queries



# ➤ Selected Features



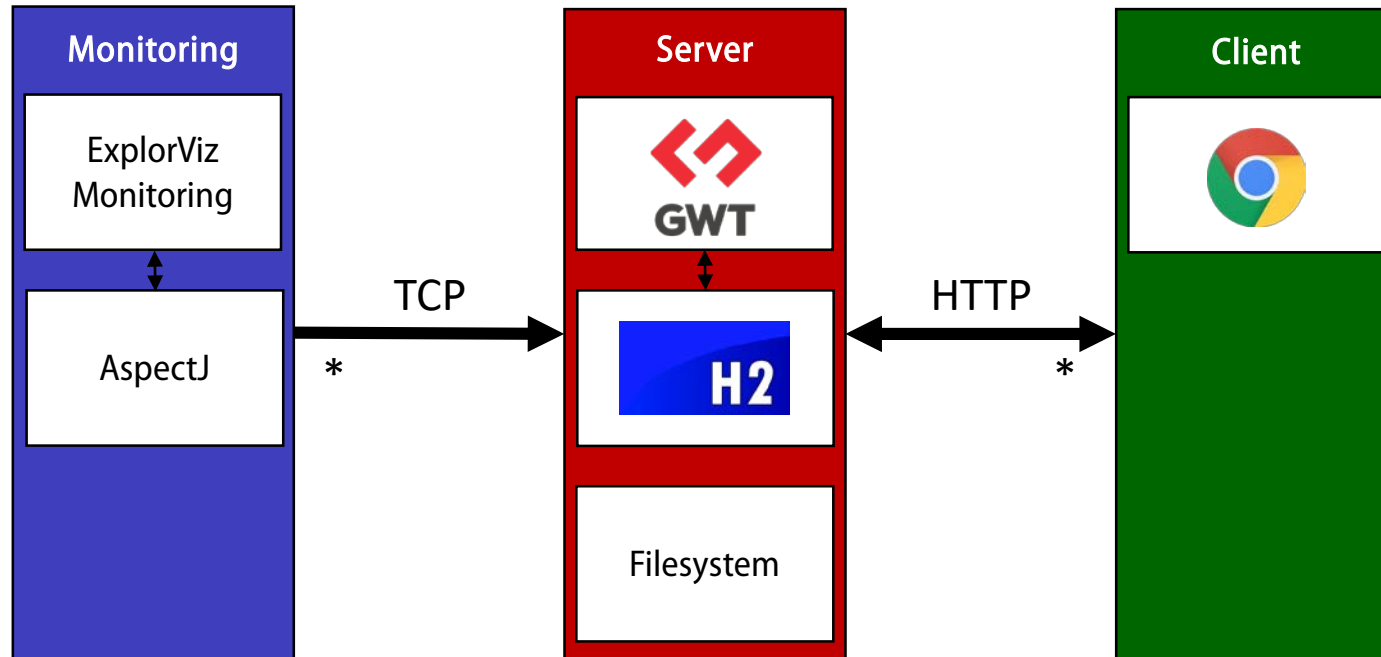
# ➤ Modernization Motivation

- Further development of GWT (Google Web Toolkit<sup>1</sup>) was unreliable
- Maintenance became more and more difficult
- Extensibility regarding new features was inconvenient
  - student theses or projects
  - external contributors

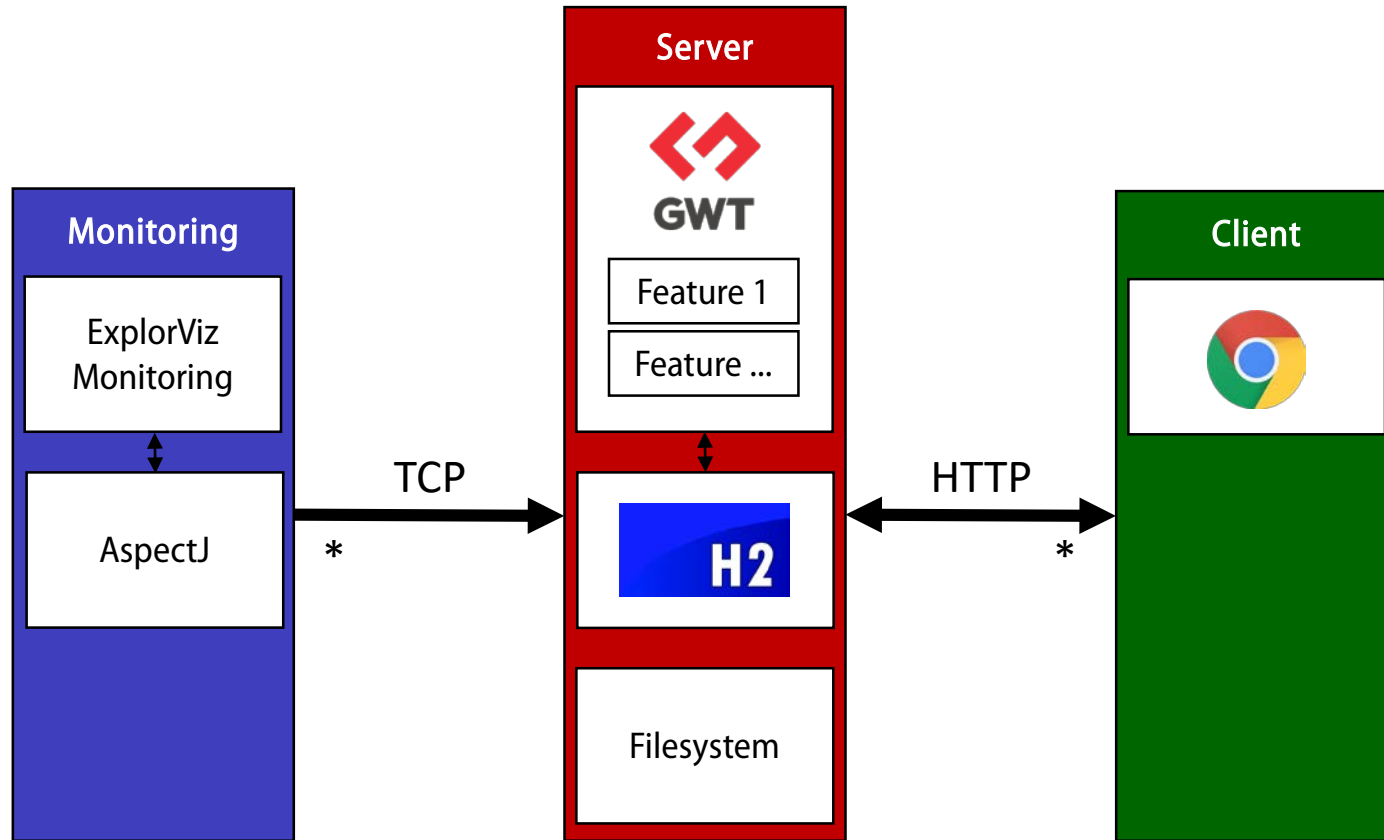
[1] <http://www.gwtproject.org>



# ➤ Architecture **before** modernization



# ➤ Architecture **before** modernization

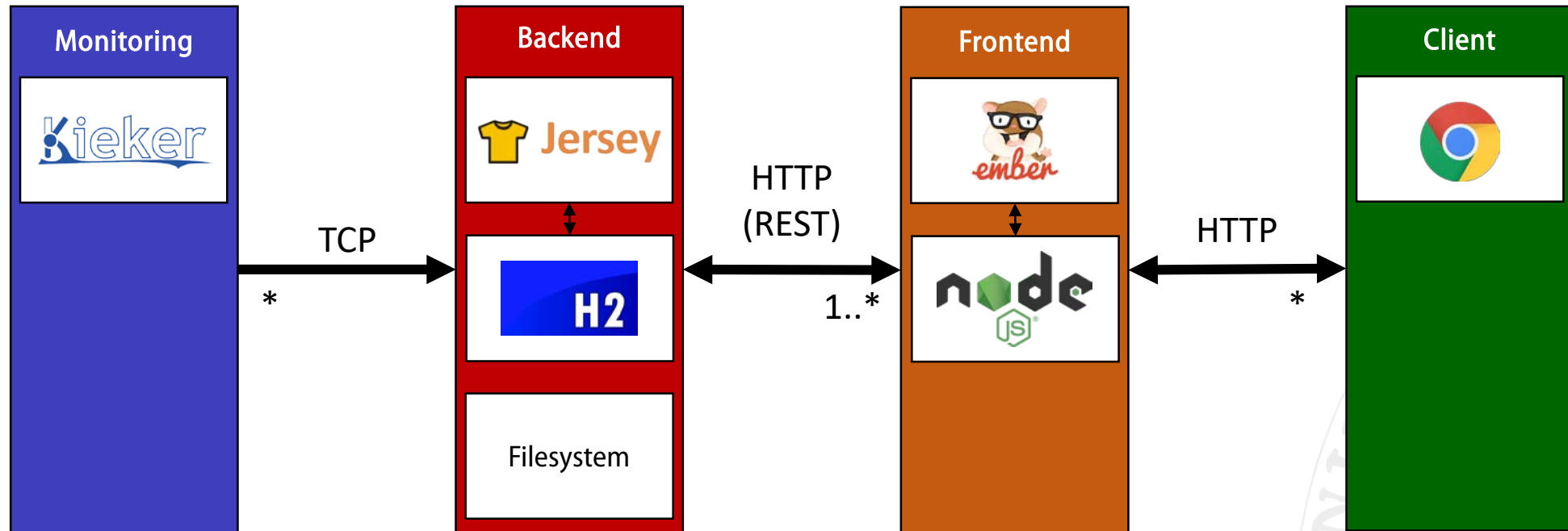


**i** Architecture: Monolith

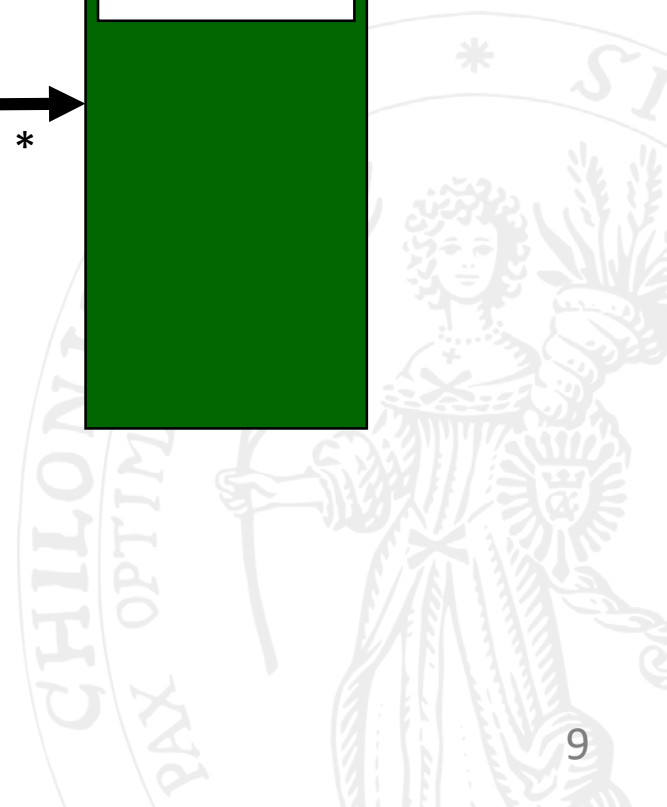




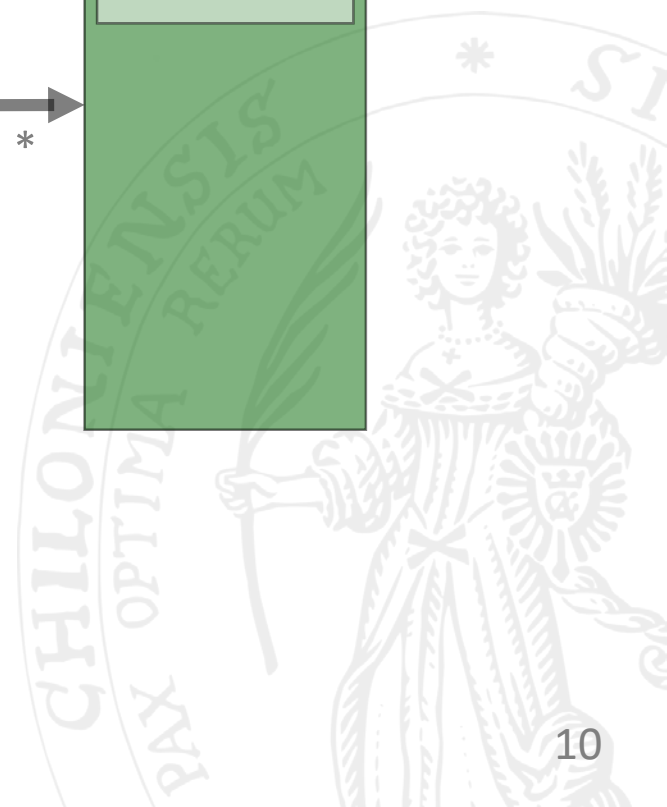
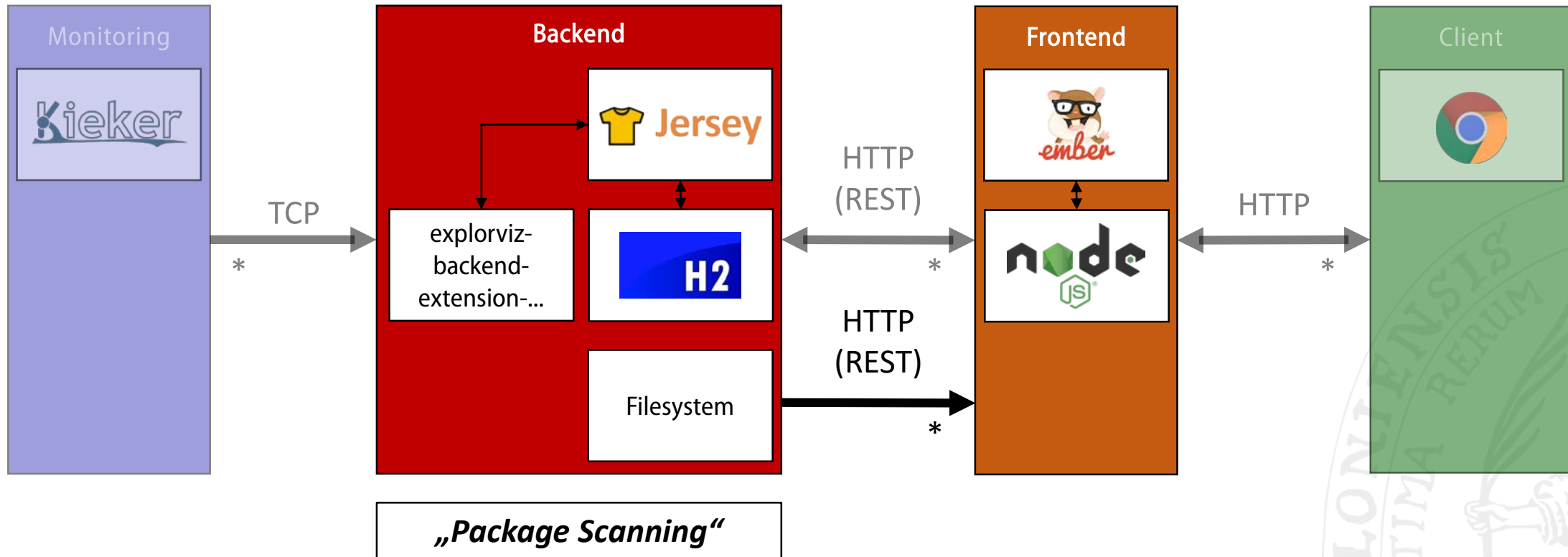
# ➤ Architecture after modernization



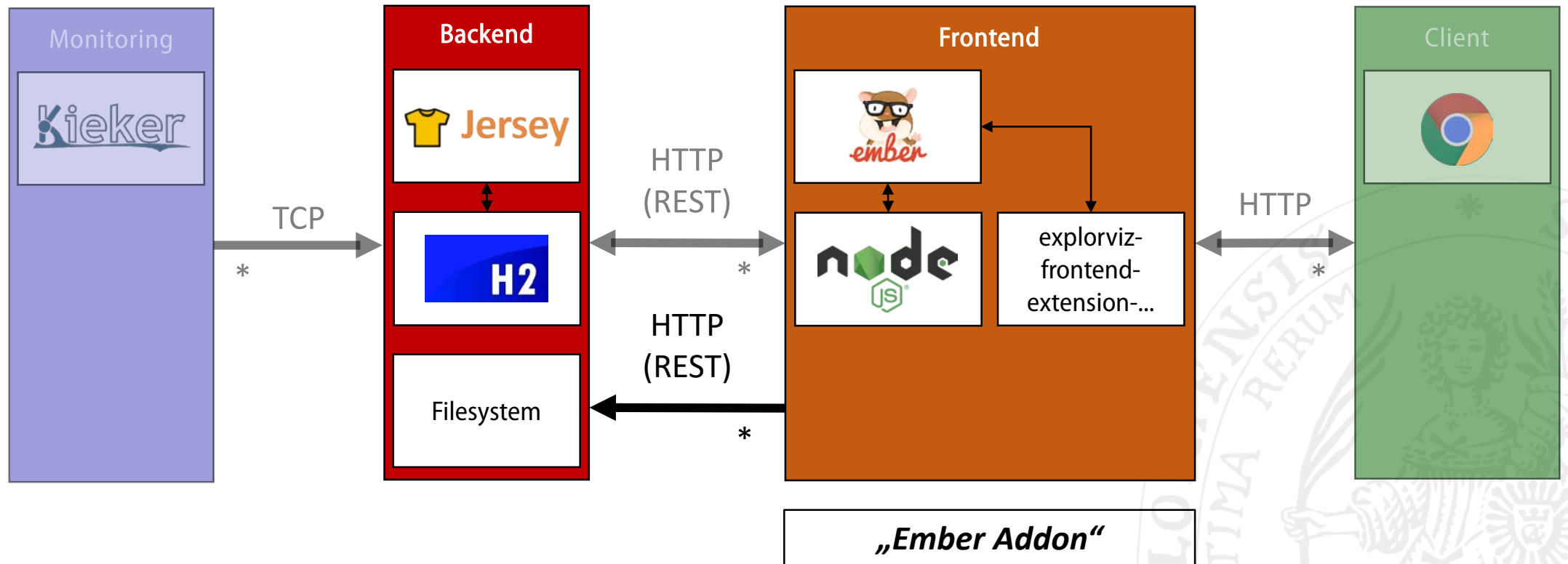
**i** Architecture: Microservice



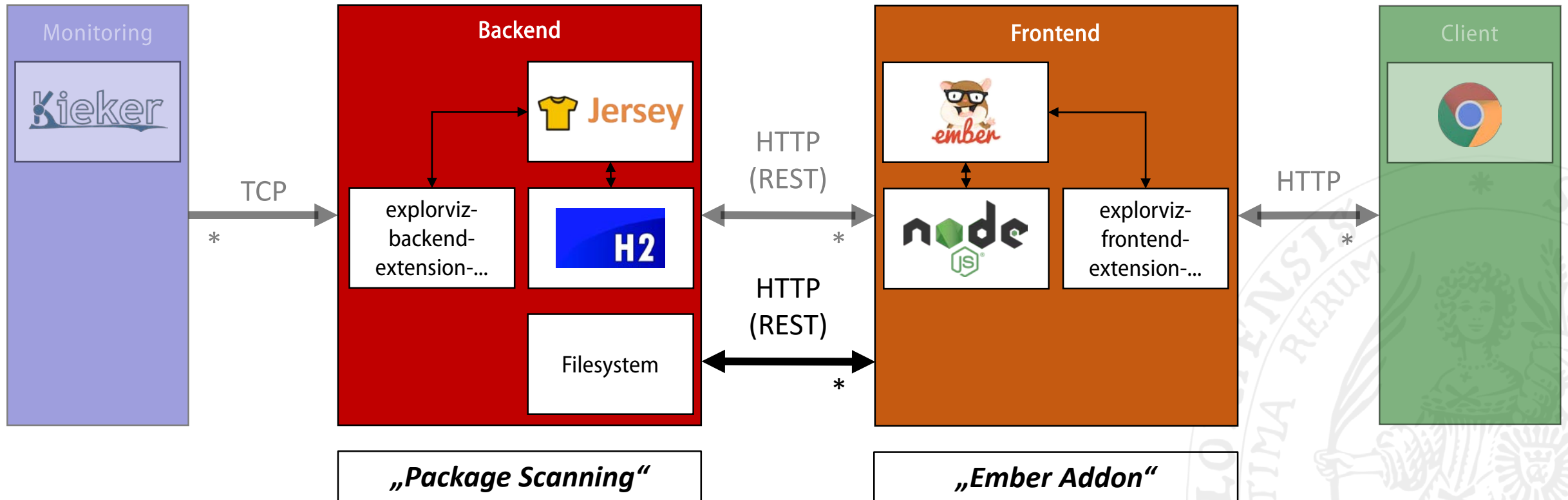
# ➤ Extensibility Concept



# ➤ Extensibility Concept



# ➤ Extensibility Concept



# ➤ Conclusions

- Migrated our architecture from a Monolith to Microservices
- Simplified the development of extensions
- Ongoing development
  - Further refactoring
  - Migrating missing features
  - Adding new features



Reduced code base from >50k to less than 44k LOC  
(without comments and blank)



## ➤ Open Questions



How can we derive best practice guidelines from our migrations for other projects?



Does the rapid evolution of JS frontend frameworks influence the ongoing evolution of ExplorViz?



How can we reposition ExplorViz as an open source visualization framework for diverse data, e.g., database visualizations?