Model-based Software Architecture Evolution and Evaluation

- Who We Are
- Architecture Evolution and Evaluation
- Problem Statement
- Challenges
- State of the Art
- Approach
- Conclusion

H. Lichter

A. Dragomir

Research Group
Software Construction

RWTH Aachen University

horst.lichter@swc.rwth-aachen.de

adragomir@swc.rwth-aachen.de

www.swc.rwth-aachen.de

RWTH AACHEN UNIVERSITY

SOFTWARE CONSTRUCTION SWC

Who we are

- RWTH Aachen University
  - http://www.rwth-aachen.de/
- Chair for Software Construction
  - Head of the Group: Prof. Dr. Rer. Nat. Horst Licter
  - Research focus:
    - Requirements Engineering
    - Metrics and Processes
    - Model Management
    - *Architecture Evolution and Evaluation*
  - https://www2.swc.rwth-aachen.de/

Architecture Evolution & Evaluation

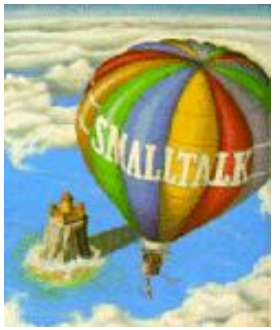
- Cooperation project with Generali Deutschland Informatik Services
  - More than 500 Systems
  - More than 4000 information flows

- Challenge
  - Large IT Landscapes
  - Complex, heterogeneous systems

COBOL

Java

EJB



Problem Statement

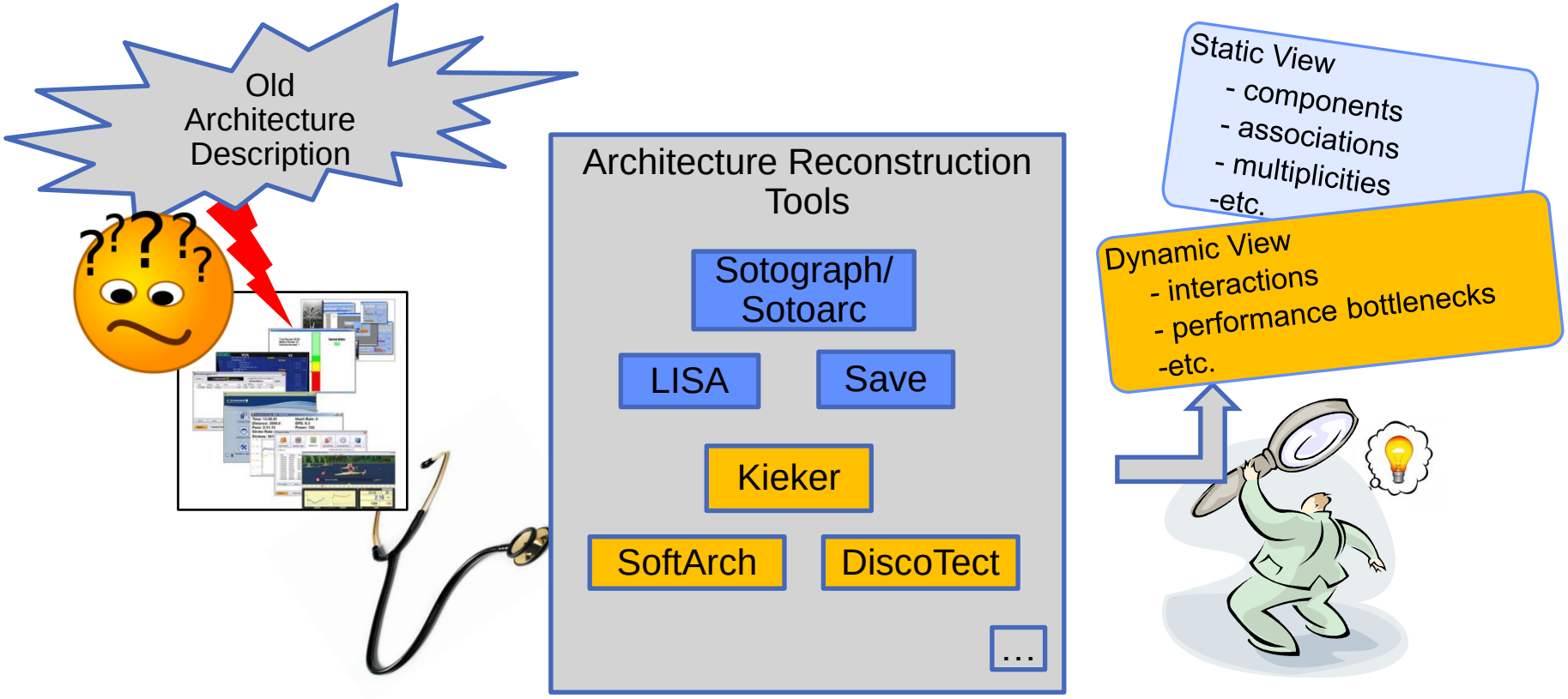
- The importance of software architecture has been widely acknowledged
- planningIT
  - A central repository for enterprise architecture management
    - IT Architecture
    - Business Architecture
    - Service Architecture
    - Project Management
  - **The information is introduced manually**
- Architecture Erosion

Challenges

- No general solution, to recover multiple architecture views
- The recovered behavior is usually not mapped on architecture elements
  - How are the layers collaborating to achieve a certain behavior?
  - How are the components collaborating within a layer?
  - How is a component achieving its task?
- Hard to understand where architecture rules are violated
- Architecture evaluation is rarely automated
- Architecture variants are compared manually

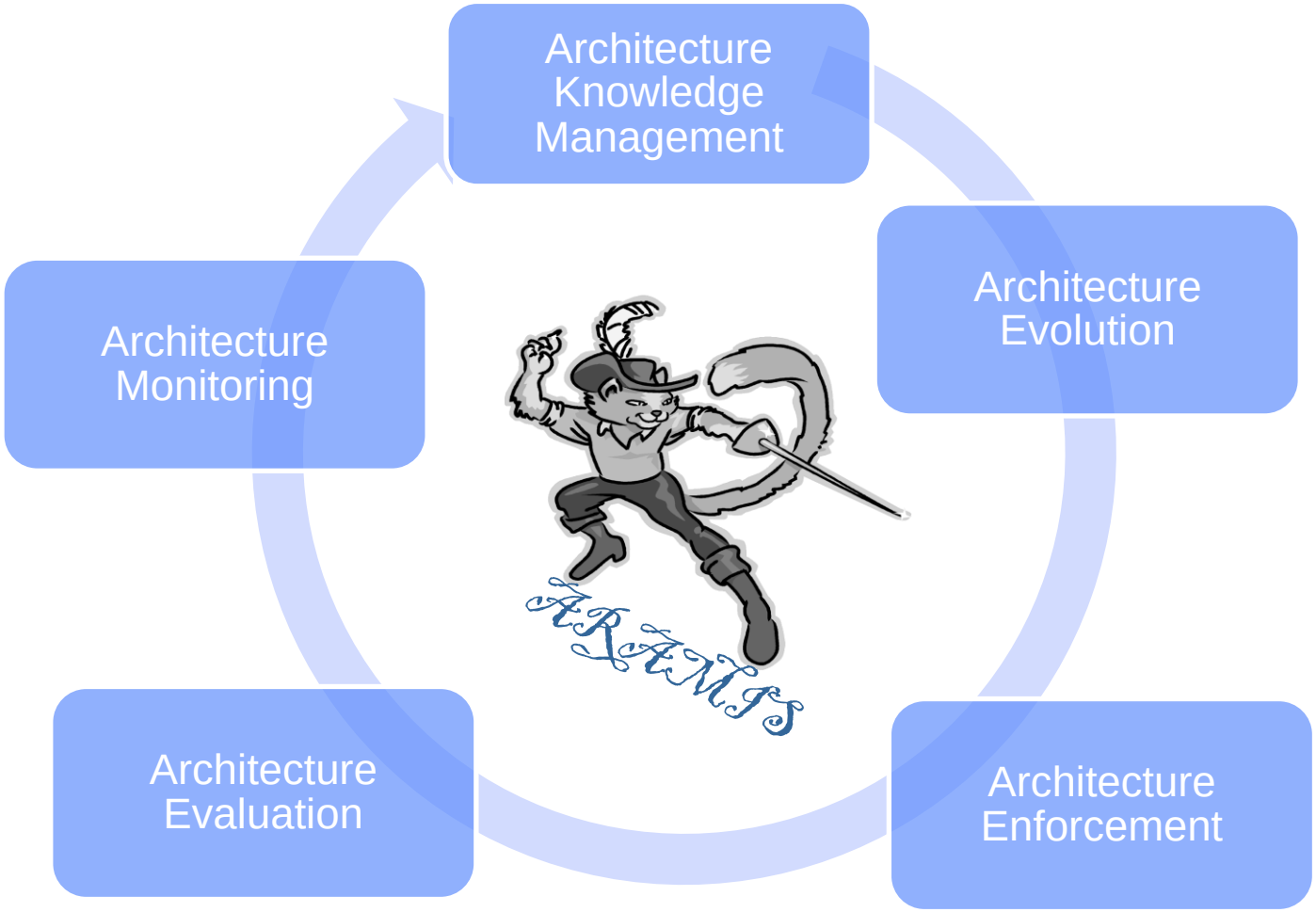
State of the Art

- Multiple solutions exist to recover & visualize the current status of the software architecture
- Software evaluation techniques have been widely proposed

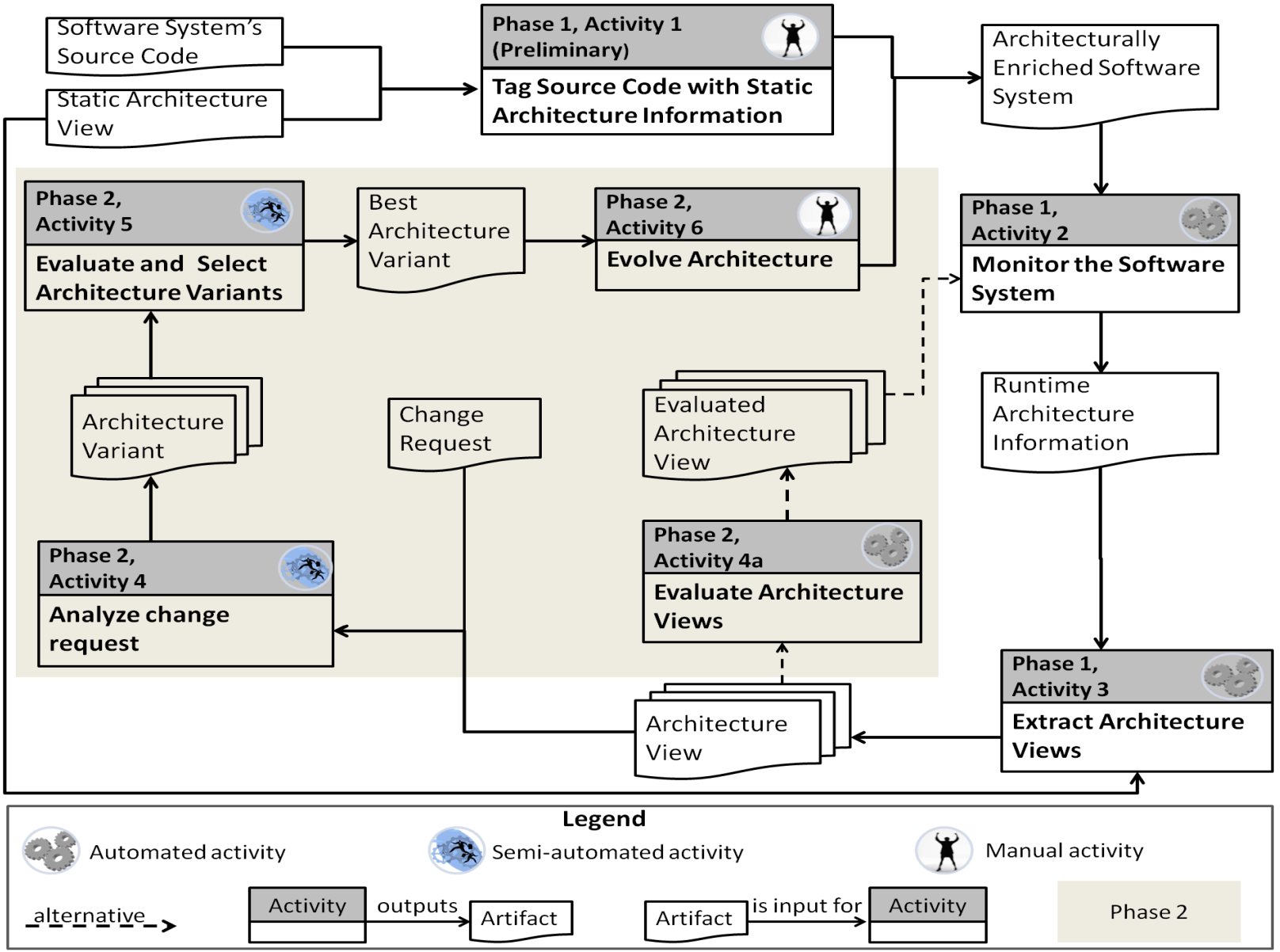


Approach

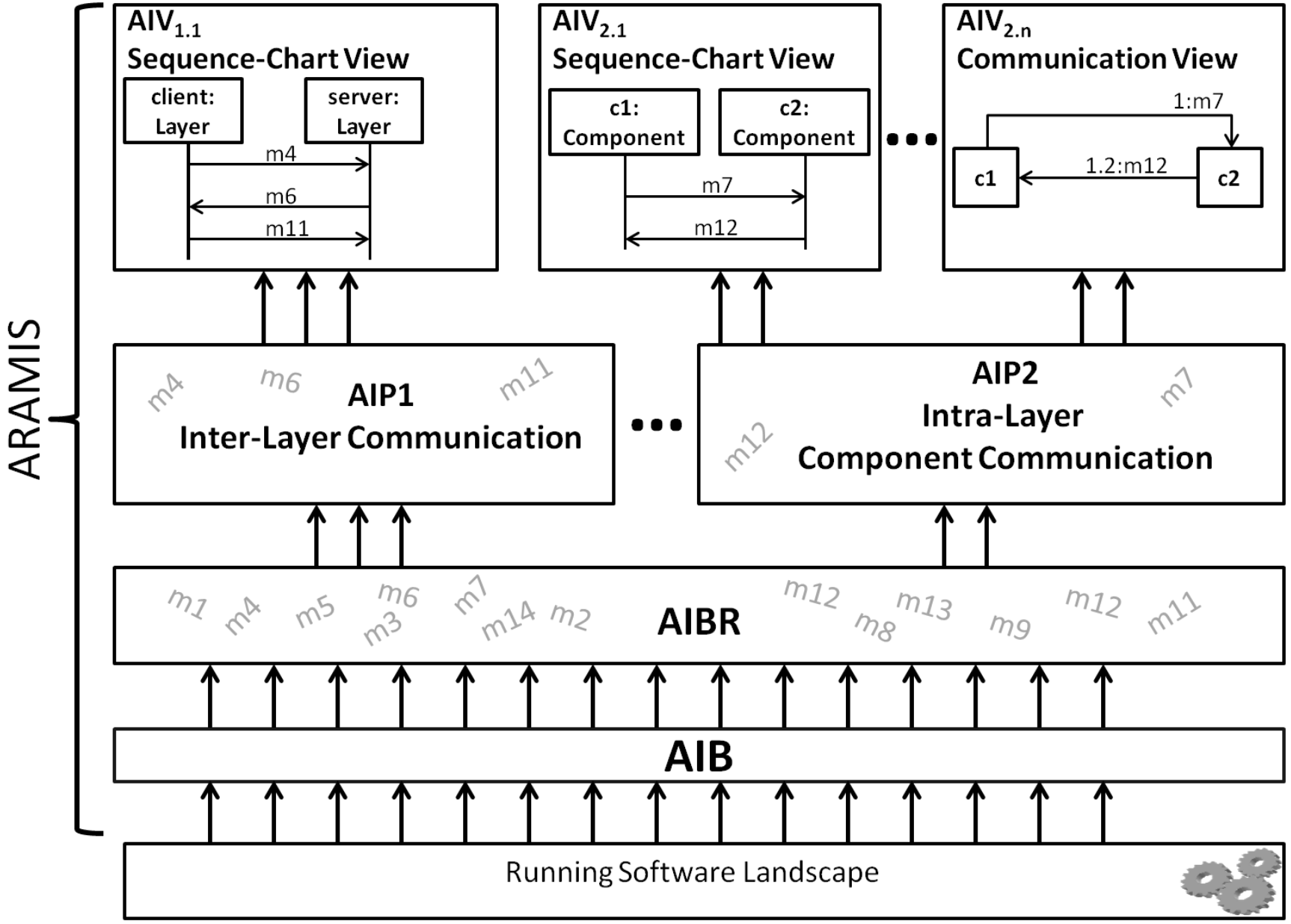
The ARichtecture Analysis and Monitoring InfraStructure



General Workflow



Architecture Monitoring



Architecture Monitoring (2)

- What architecture meta-model to use?
  - What is a component?
- To which upper-level architecture element should a component's behavior be attributed?
- How to technically implement the monitoring?
- What visualization types are useful?

Architecture Evaluation

- Allow the specification of rules, to describe:
  - Allowed/Prohibited interactions
  - Performance restrictions

  } at different abstraction levels
- Visualize violations of rules
- Define metrics, to measure conformance with rules
- Re-use/define metrics, to measure overall architecture quality
- Re-use/define metrics, to compare architecture variants
- Automate the computation of metrics' value
  - Visualize the evolution of software architecture's quality

Conclusion

- ARAMIS aims to offer a holistic approach for architecture evolution and evaluation

- The first steps to achieve this:
  - Develop a model-based software architecture monitoring approach
  - Offer means to define software-architecture related rules
  - Re-use/define metrics to (semi-) automatically evaluate the software architecture on a regular basis