



..|| QuASoQ 2014

Towards an Architecture Quality Index for the Behavior of Software Systems

Ana Dragomir, Horst Lichter
RWTH Aachen University
Research Group Software Construction
Aachen, Germany

Presenter: Morakot Choetkiertikul
1 December 2014

Agenda

- Introduction
- Architecture quality index
- Evaluation
- Conclusion
- Discussion

Introduction

- Software architecture is very important
 - affects non-functional characteristics e.g. maintainability, extensibility.
- This paper presents an **architecture quality index** based on information extracted during run-time e.g. number of calls inside the system.
- This approach is implemented based on the Architecture Analysis and Monitoring Infrastructure (ARAMIS)

Architecture Quality Index

- Architecture quality index consists two main components:
 - Bidirectional quality model
Mapping between the measurable attributes and the subjective quality characteristics.
 - Quality Benchmark levels
Assigning a quality level, the higher the assigned level, the better the architecture is.

Bidirectional quality model

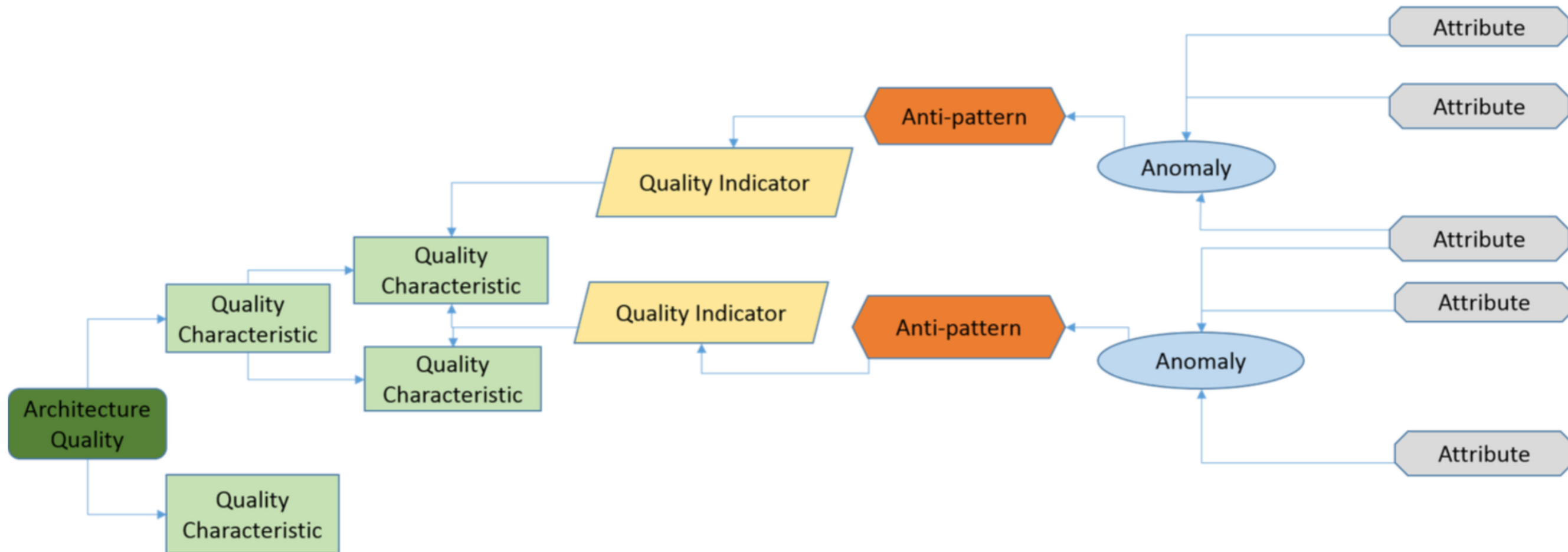


Fig. 1: Bidirectional Quality Model [1]

Bidirectional quality model

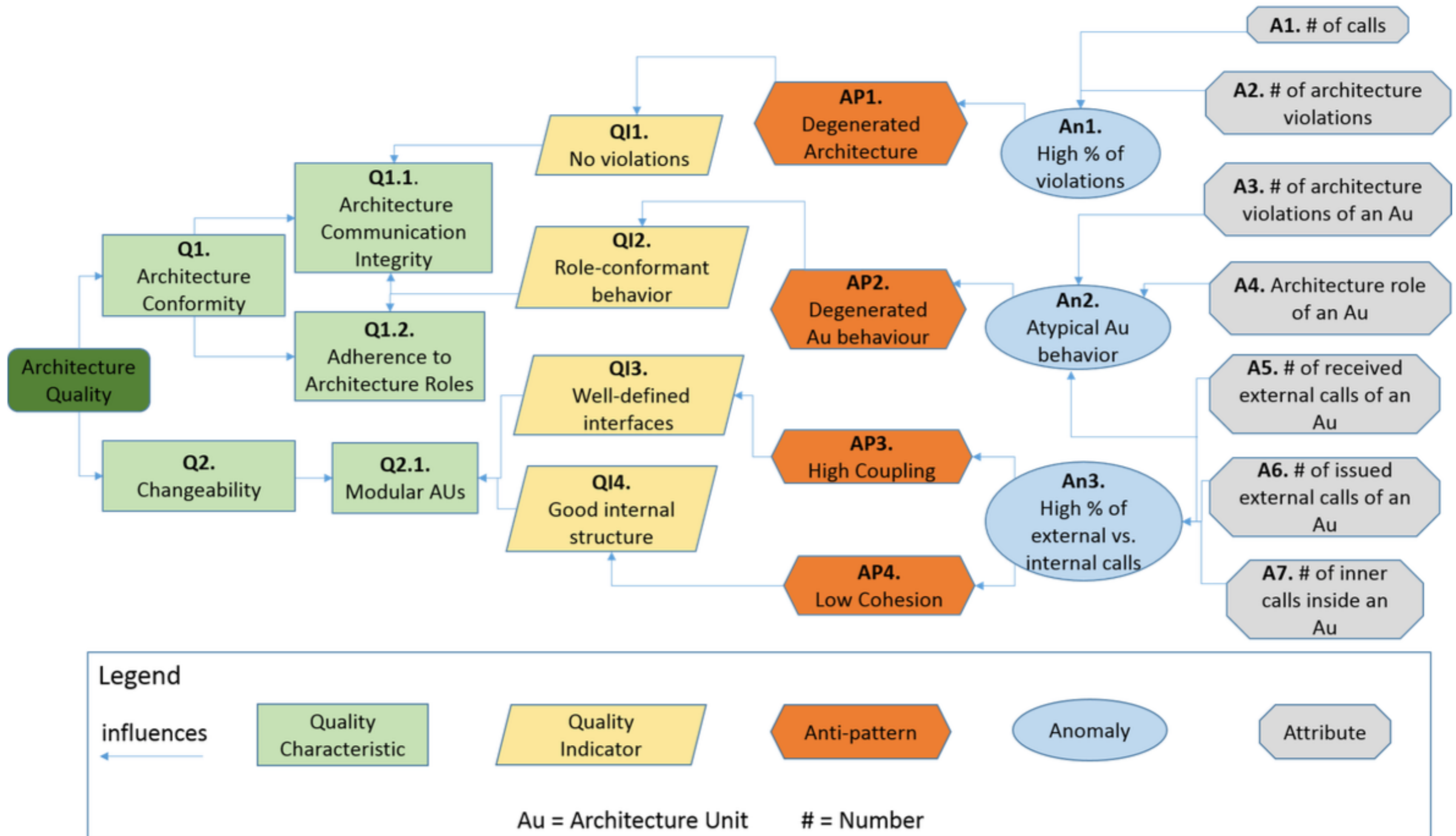


Fig. 6: Bidirectional Quality Model for ARAMIS

Quality Benchmark Levels

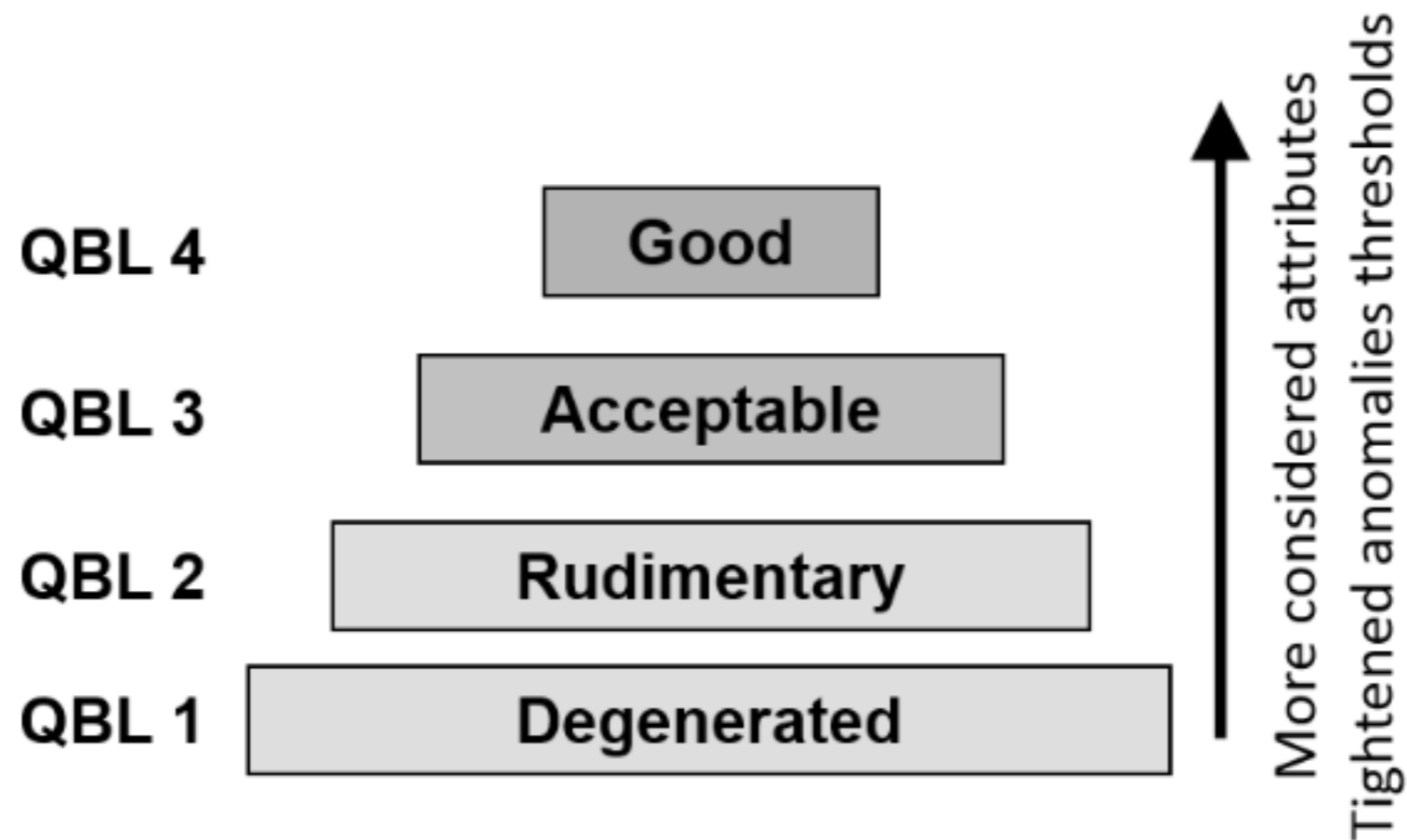
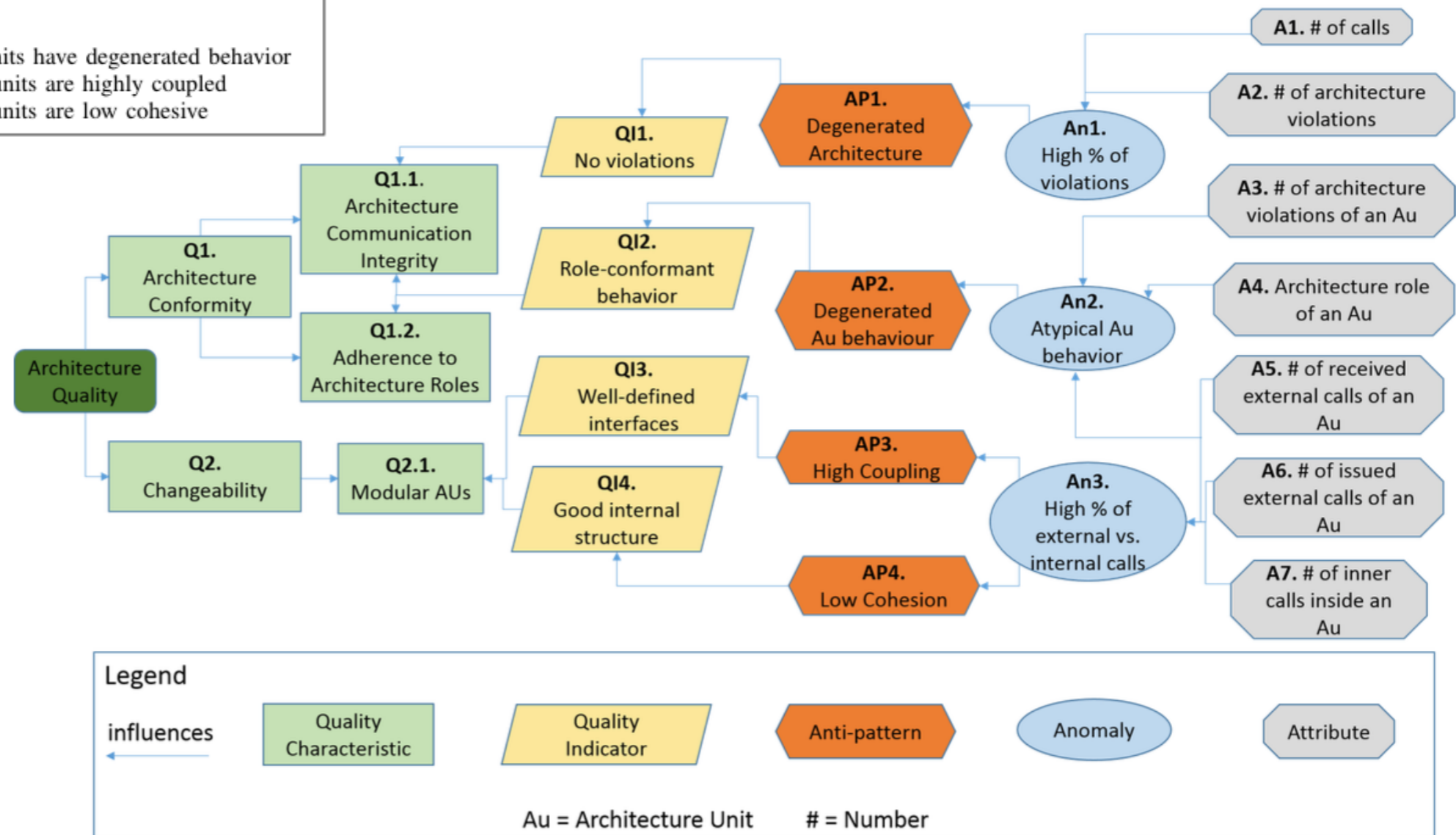


Fig. 2: Quality Benchmark Levels

Quality Benchmark Levels

TABLE I: Quality benchmark for ARAMIS

Level Name	Conditions
Degenerated	-
Rudimentary	< 50% violations < 50% of the architecture units have degenerated behavior
Acceptable	< 25% violations < 25% of the architecture units have degenerated behavior
Good	< 5% violations < 5% of the architecture units have degenerated behavior < 40% of the architecture units are highly coupled < 40% of the architecture units are low cohesive



ARAMIS-CICE

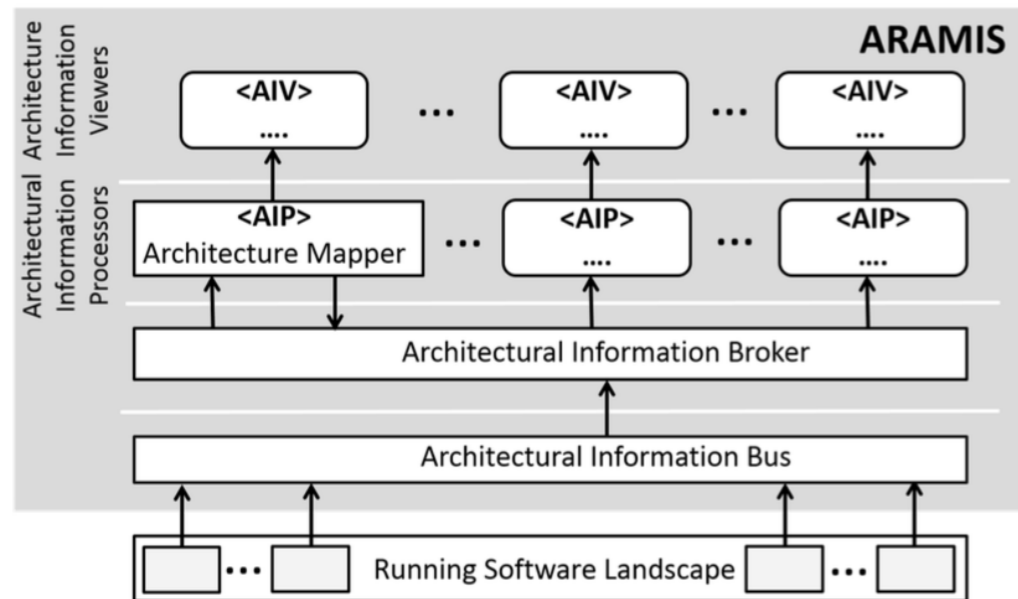


Fig. 3: ARAMIS - General Architecture [2]

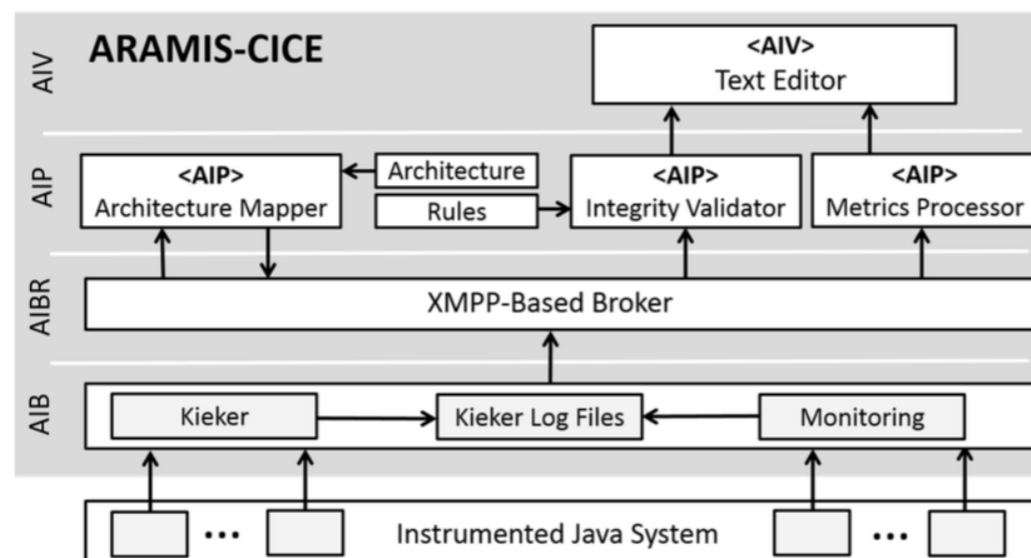


Fig. 5: ARAMIS-CICE Architecture

- ARAMIS is a tool-based concept to monitor software on different levels of abstract.
- ARAMIS-CICE is an instance of ARAMIS to collect the run-time traces and monitoring a communication between various architecture units.

Evaluation

- This approach is evaluated using the open-source framework “JHot-Draw” (26068 LOC, 529 classes and 38 packages.)
- 7 architecture units involved in the scenario
- 0% violations (using ARAMIS-CICE)
- jhotdraw.geom - called by other 71%, call inside the unit 29%
- “Acceptable” (There are two units have not met the changeability quality)

Conclusion

- To maintain the quality of software architecture,
 - Measurement!!!!
 - Benchmarking!!!!
- This paper presents a quantitative approach that provides the answer to measure and benchmark the quality of software architecture.

Discussion...

- How to validate the architecture quality model and the thresholds of the quality benchmark.