

# Do Services require Online Testing? A Case Study

Michaela Greiler

Hans-Gerhard Gross

Arie van Deursen

Delft University of Technology  
Software Engineering Research Group

# Overview

- Runtime Evolution Challenges of SOA
- Online Testing Framework
- A Case Study
- Lessons Learned

# Service-Oriented Architectures

A software architecture that aims at reusability and interoperability by exposing functionality as loosely coupled services, that can be composed to form business processes.

# SOA Runtime Evolution: Challenge I

## Stakeholder Separation



- Communication
- Ownership
- Responsibilities
- Influence

# SOA Runtime Evolution: Challenge II

## Service Integration



- Black-boxes
- Test environment
- (Runtime)-Testability

# SOA Runtime Evolution: Challenge III

## Service Versioning and Migration



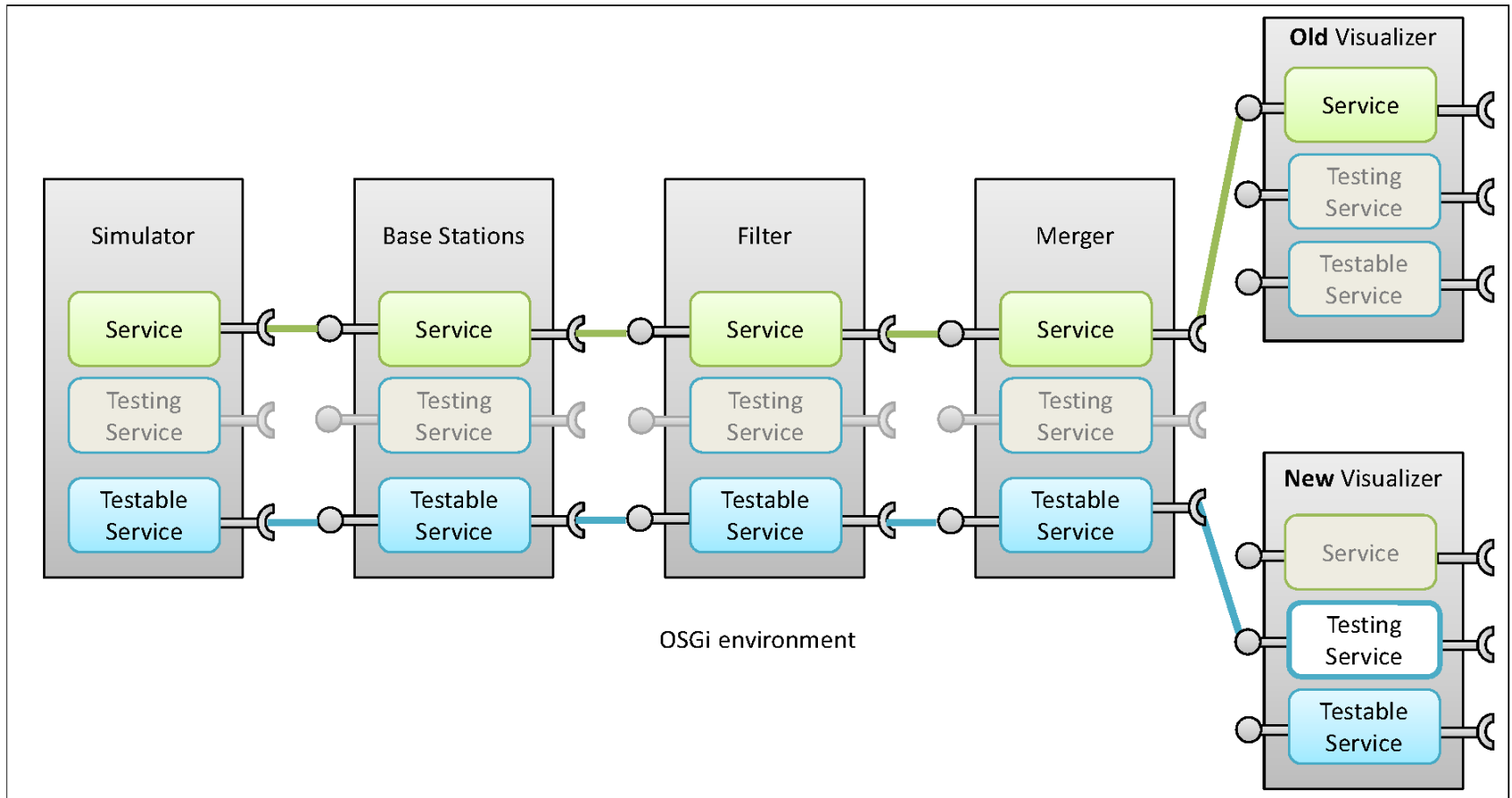
- Notification
- Update Strategy
- Regression Testing

# SOA Runtime Evolution: Challenge IV

## Service Binding and Reconfiguration



# Example System: Sensor network

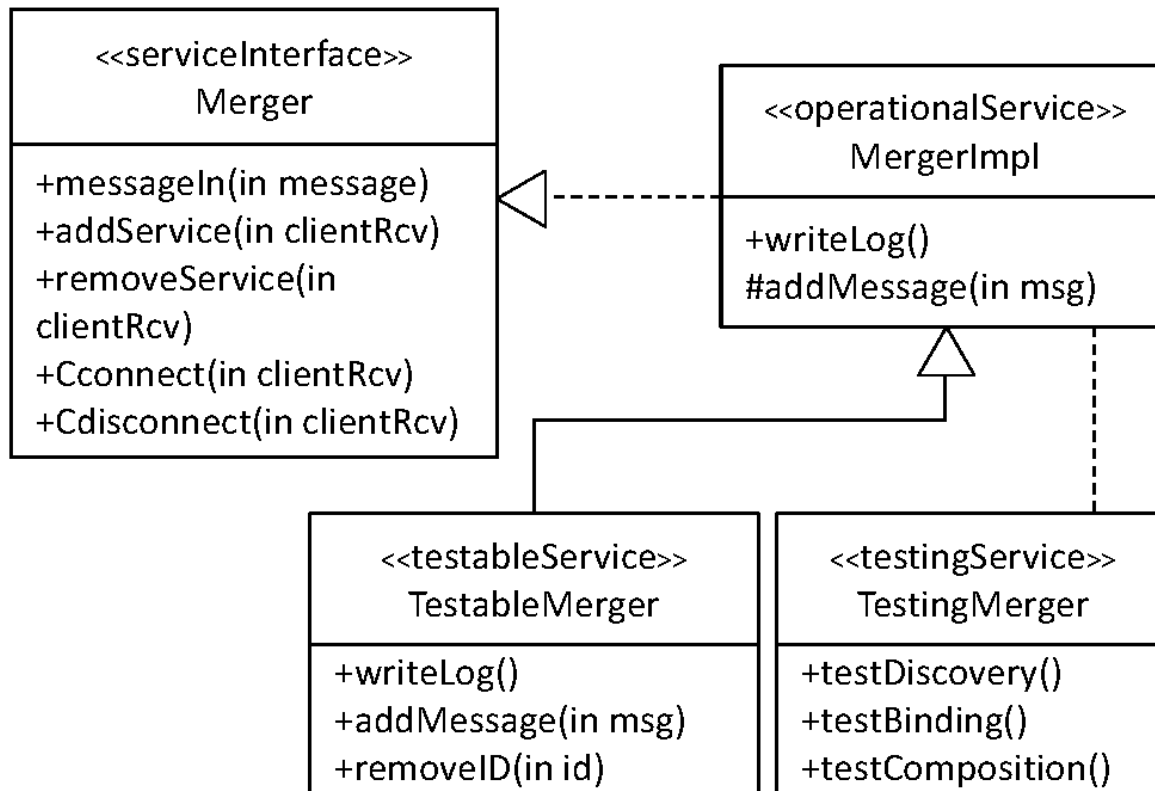




# General Built-in Testing

- Associate the tests to the component
- Isolate effects of testing operations
- Benefits
  - Simplify the management of the tests
  - Can add tests at runtime
  - Components' functionality is separated from the tests
  - At anytime of the components can be re-validated, whenever needed (eg. after a modification of the system)
- Adoption effort
  - Explicit definition of the tests
  - Testing framework has to be linked with the system platform

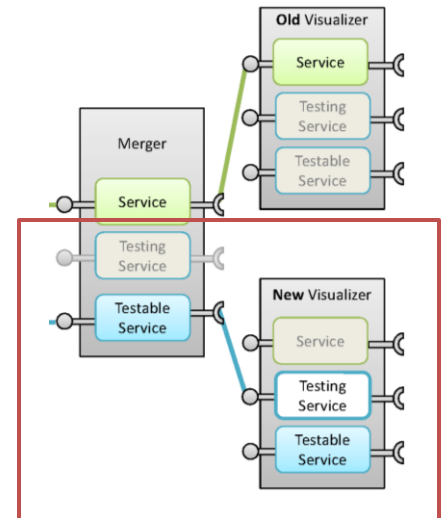
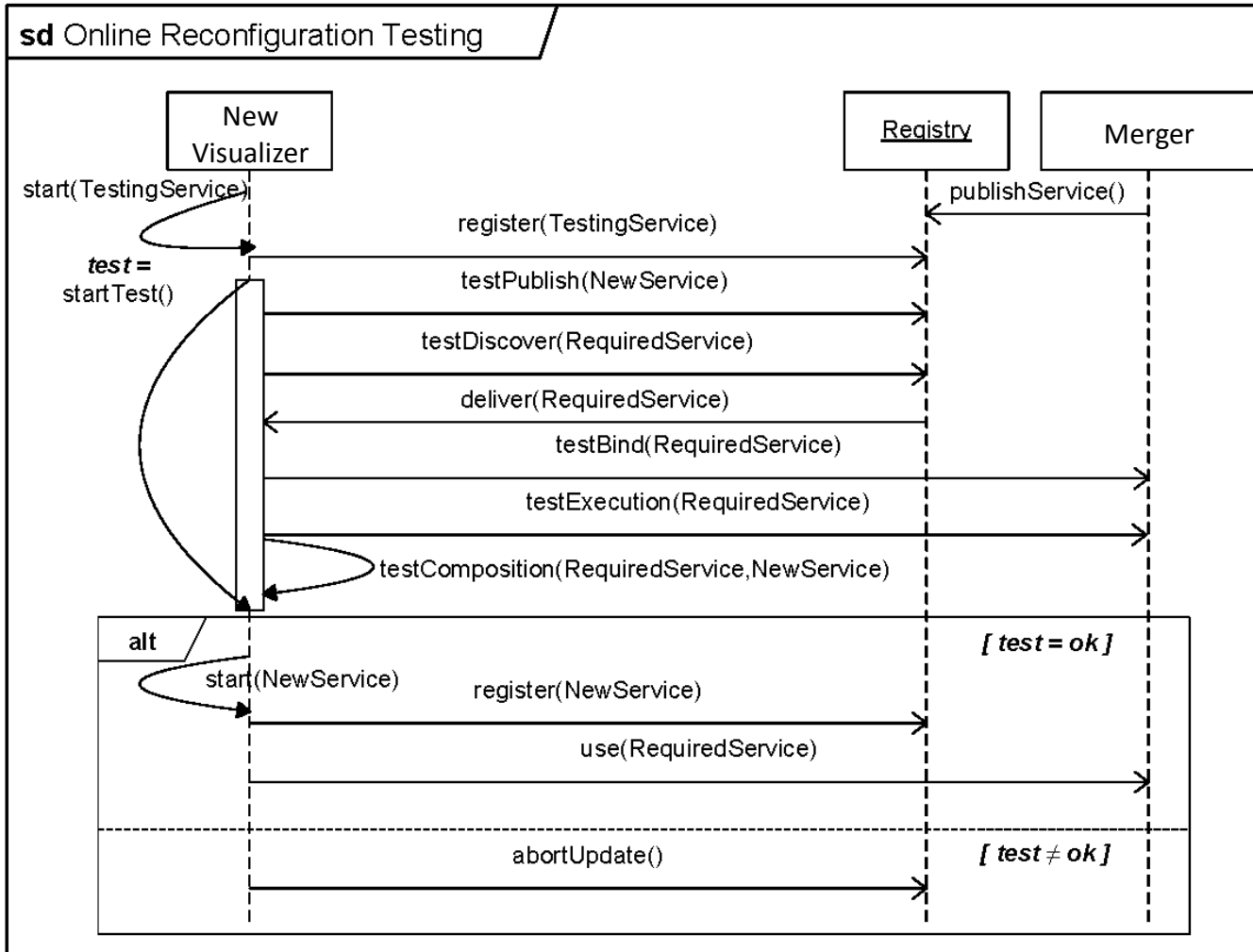
# Built-in Online Testing



# Online Tests

- **Discovery Tests:**
  - Can I find all required services?
  - Are those services the services expected?
- **Binding Tests:**
  - Can I bind required services?
  - Do they conform syntactically (service descriptions) to my expectation?
- **Execution Tests:**
  - Test invocations looking for:
    - expectations about sequential constraints, i.e., the protocol,
    - simple input-output relations
    - non-functional requirements (e.g., timing)

# Test Process



# Case Study Design

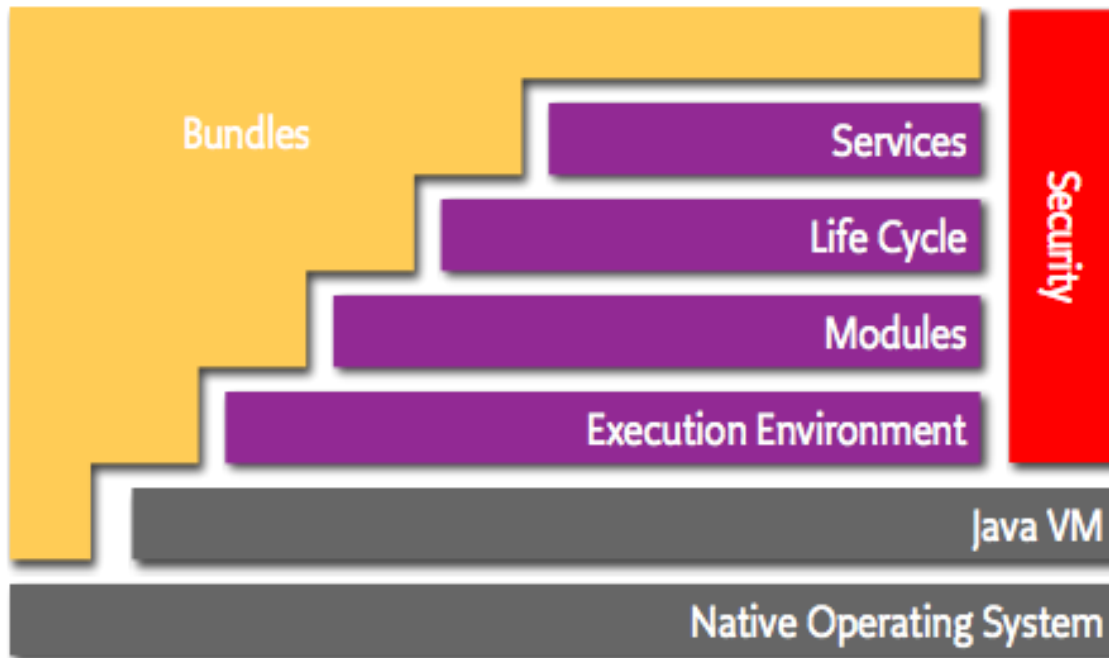
- **Goal:** assess the capabilities of our online testing framework to find typical reconfiguration faults
- **Approach:**
  1. extend original system with online testing infrastructure
  2. reconfiguration scenario, where a number of faults have been seeded
- **Key Metrics and Execution Environment:**
  - **Apache Felix 1.8**
  - OVTS system: 6 key bundles, 1 utility bundle, 2176 LOCs, 41 classes. 349 LOCs dedicated to testing; 39 test cases consisting of 290 LOCs.
- **Units of Analysis:**
  - level of provided test isolation
  - detection rate for the seeded faults
  - how have they been found, or why couldn't they be detected?
  - evaluation of the support of the execution environment to prevent inconsistencies in the overall system

# Research Questions

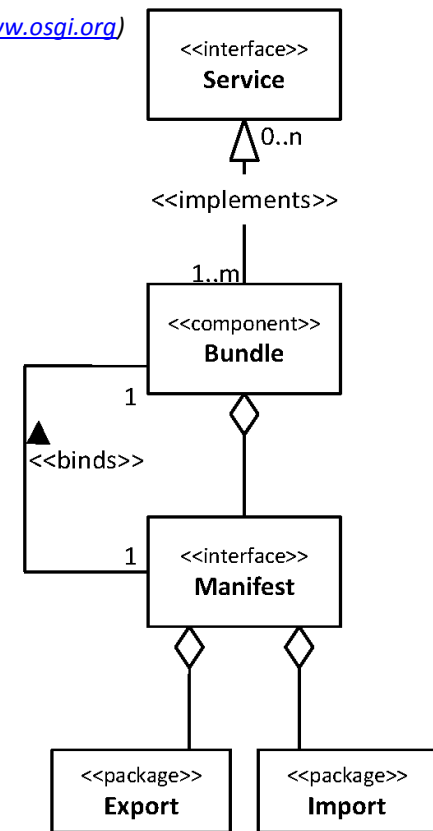
- RQ 1:** “What level of test isolation can we achieve with the proposed infrastructure?”
- RQ 2:** “Which typical reconfiguration faults can be detected with this online testing framework?”
- RQ 3:** “Which faults cannot be detected by the online testing framework?”
- RQ 4:** “To what extent are these faults attributed to the current state and configuration of the production system?”

# Open Service Gateway Initiative (OSGi)

“OSGi technology is **Universal Middleware**. OSGi technology provides a service-oriented, component-based environment for developers and offers standardized ways to manage the software lifecycle.” ([www.osgi.org](http://www.osgi.org))

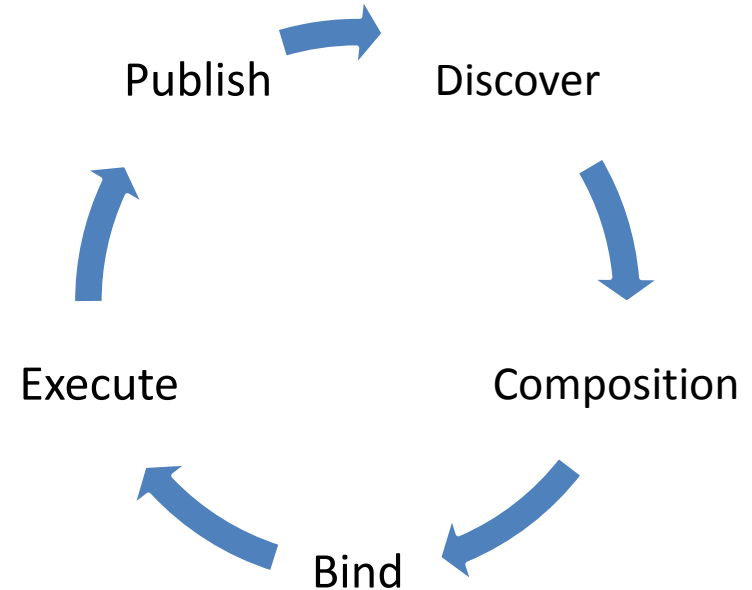


Source: [www.osgi.org](http://www.osgi.org)



# SOA Faults

Fault Classes	Type/ Form
Publishing	Service Description Service Deployment
Discovery	No Service Found Wrong Service Found Timed Out
Composition	No Valid Composition Composition Faulty Timed Out
Binding	Binding Denied Bound to Wrong Service Timed Out
Execution	Service Crashed Incorrect Results Timed Out



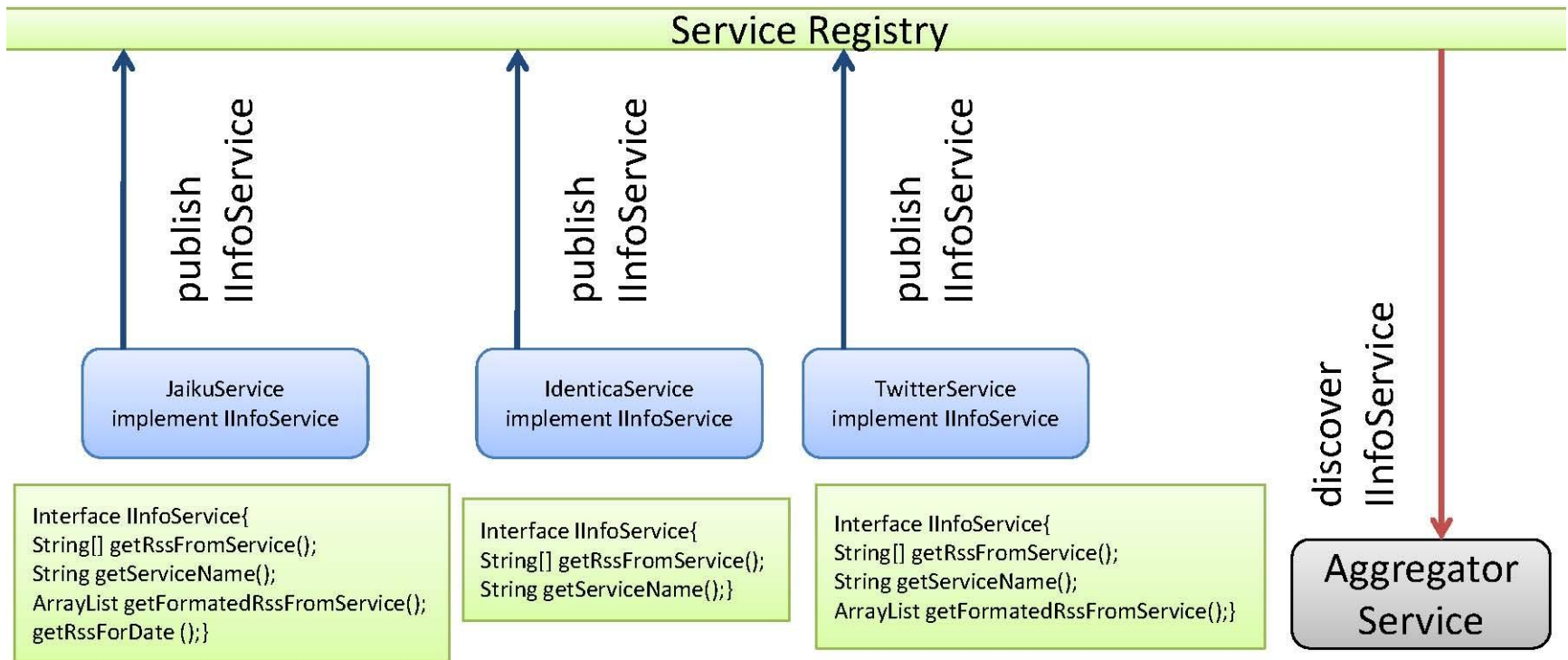
“A Fault Taxonomy for Service-Oriented Architecture” by Bruning et al. 2007



# Faults, Forms and Tests

Fault Classes	Type/ Form	Found by
Publishing	Service Description Service Deployment	Binding Test Binding Test
Discovery	No Service Found Wrong Service Found Timed Out	Discovery Test Discovery Test Time Test
Composition	No Valid Composition Composition Faulty Timed Out	Composition Test Composition Test Time Test
Binding	Binding Denied Bound to Wrong Service Timed Out	Composition Test X Time Test
Execution	Service Crashed Incorrect Results Timed Out	Discovery Test X (unit tests) Time Test

# Example: Faulty Reconfiguration



# Lessons Learned

- Online versus Offline Testing
  - Adequate test environment
  - Current state of the production system
- Shortcomings Fault Taxonomy
  - Faults versus failures
  - Removal and update
- Shortcomings OSGi Framework
  - Support for overall consistency management
  - Middleware versus application

# Summary

- Challenges in Service-Oriented Architectures
- Online built-in testing framework that provides test isolation
- A case study demonstrating the fault finding capabilities
- Lessons Learned

# Thank you!

Michaela Greiler

[m.s.greiler@tudelft.nl](mailto:m.s.greiler@tudelft.nl)

Technical University Delft

Software Engineering Research Group