

MINT Style Based Architectural Migration: Method and Case Study

Simon Giesecke, Johannes Bornhold

<simon.giesecke@acm.org>

Software Engineering Group

Carl von Ossietzky University

Oldenburg, Germany

WMR 2006, Bari, Italy

2006-03-24



Outline

- MIDARCH Method & Context
 - Research Project
 - Vision
 - Activities
- Case Study
 - Setting
 - Preliminary Results
- Conclusion



Trustworthy
Software Systems

Certification

Part I

MIDARCH Method & Context

Correctness

Safety

Availability

Reliability

Performance

Security

Quality of Service

Component



Overall Goal

- Improvement of Integration and Migration Processes through Reuse of Design Knowledge
 - **Application Domain:** Distributed and Web-based Business Information Systems
 - **Architectural Level:** Exploration of Candidate Architectures based on Different Middleware Platforms
 - **Means:** Binding Design Knowledge to Architectural Styles

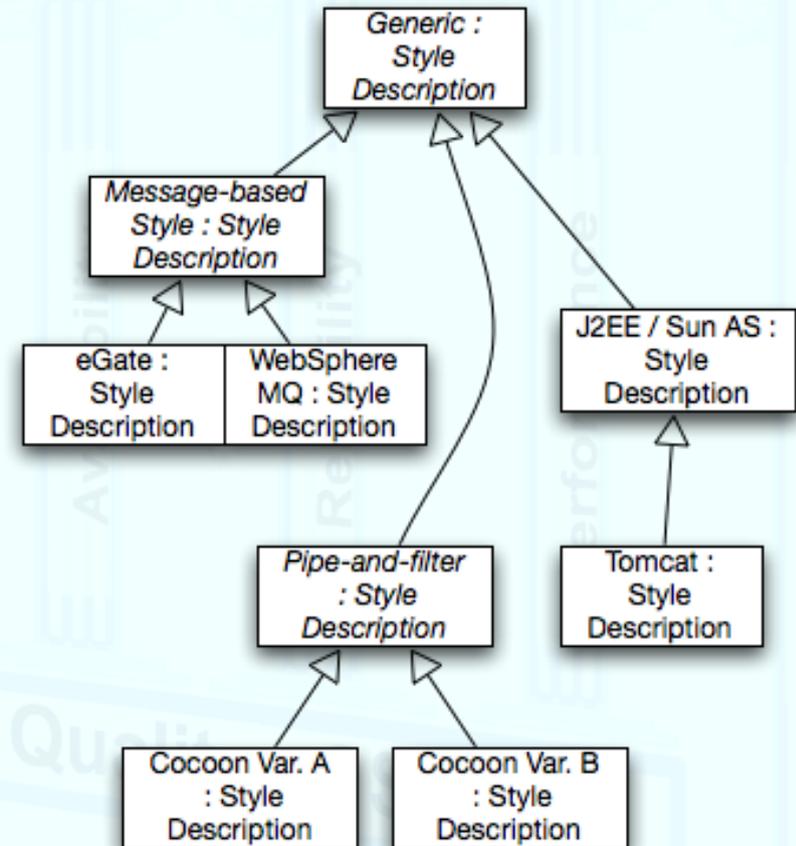
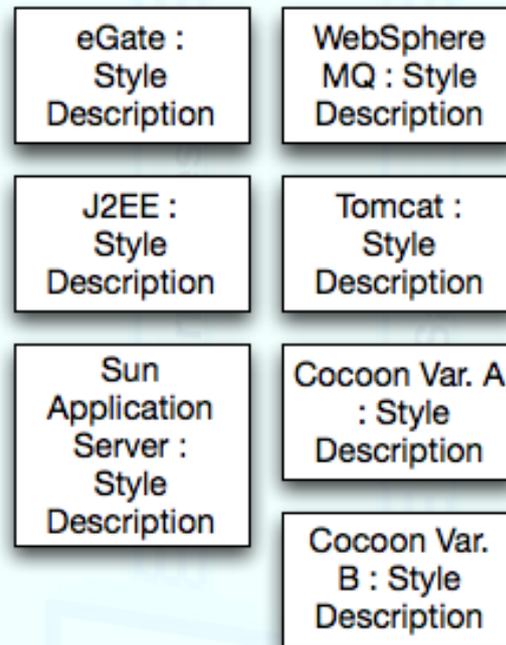
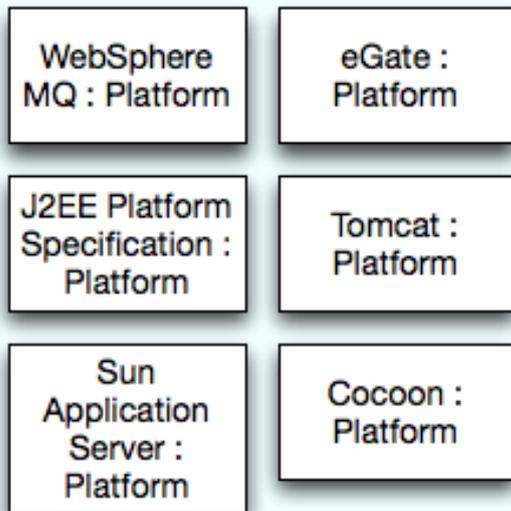


Architectural Styles

- Specify constraints on
 - Component types
 - Connector types
 - Composition rules
- Generic styles
 - pipe-and-filter
 - event-based
- Specialised styles: endorsed by an implementation platform
 - Middleware INTegration Styles (MINT Styles)

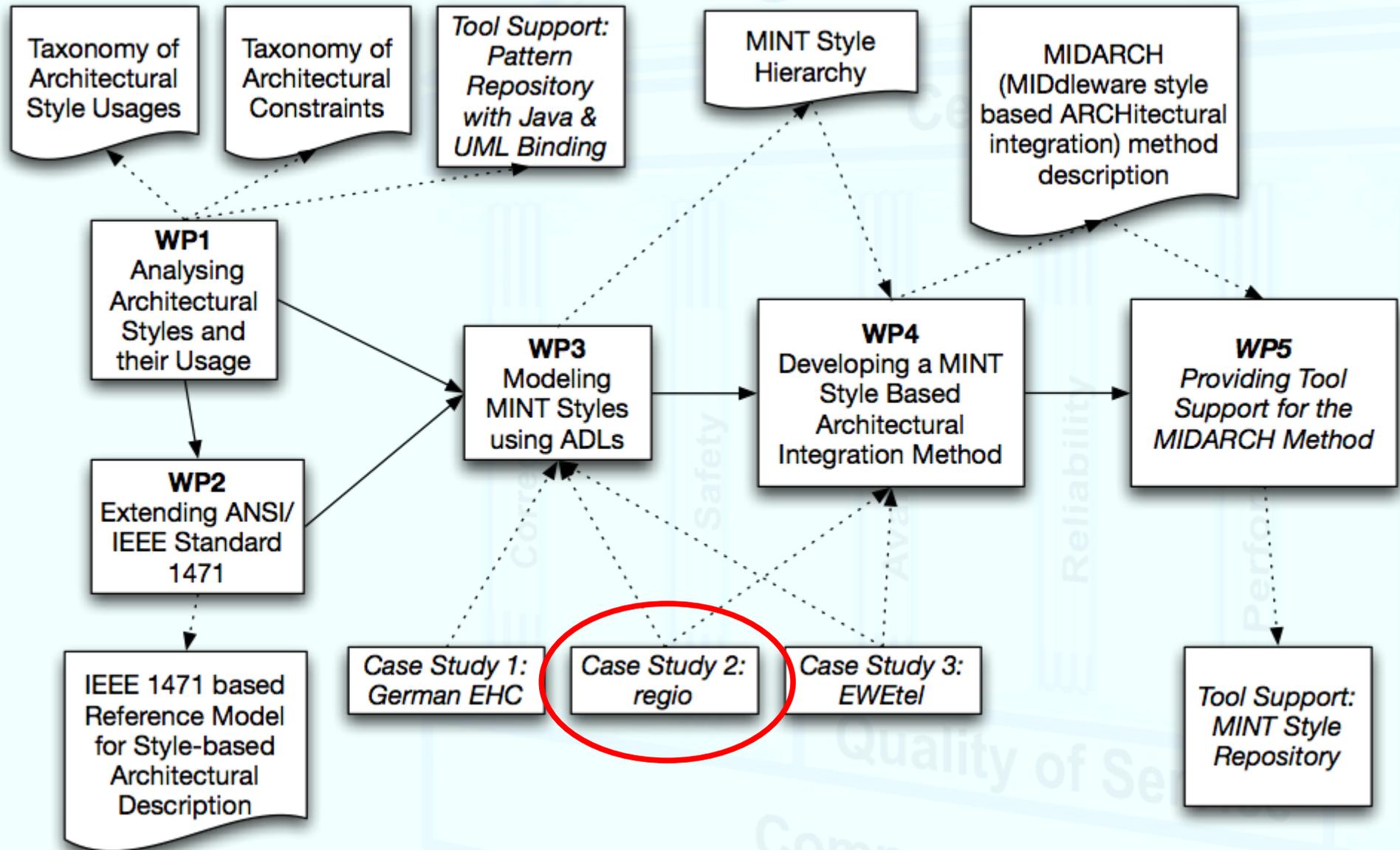


Middleware to Styles



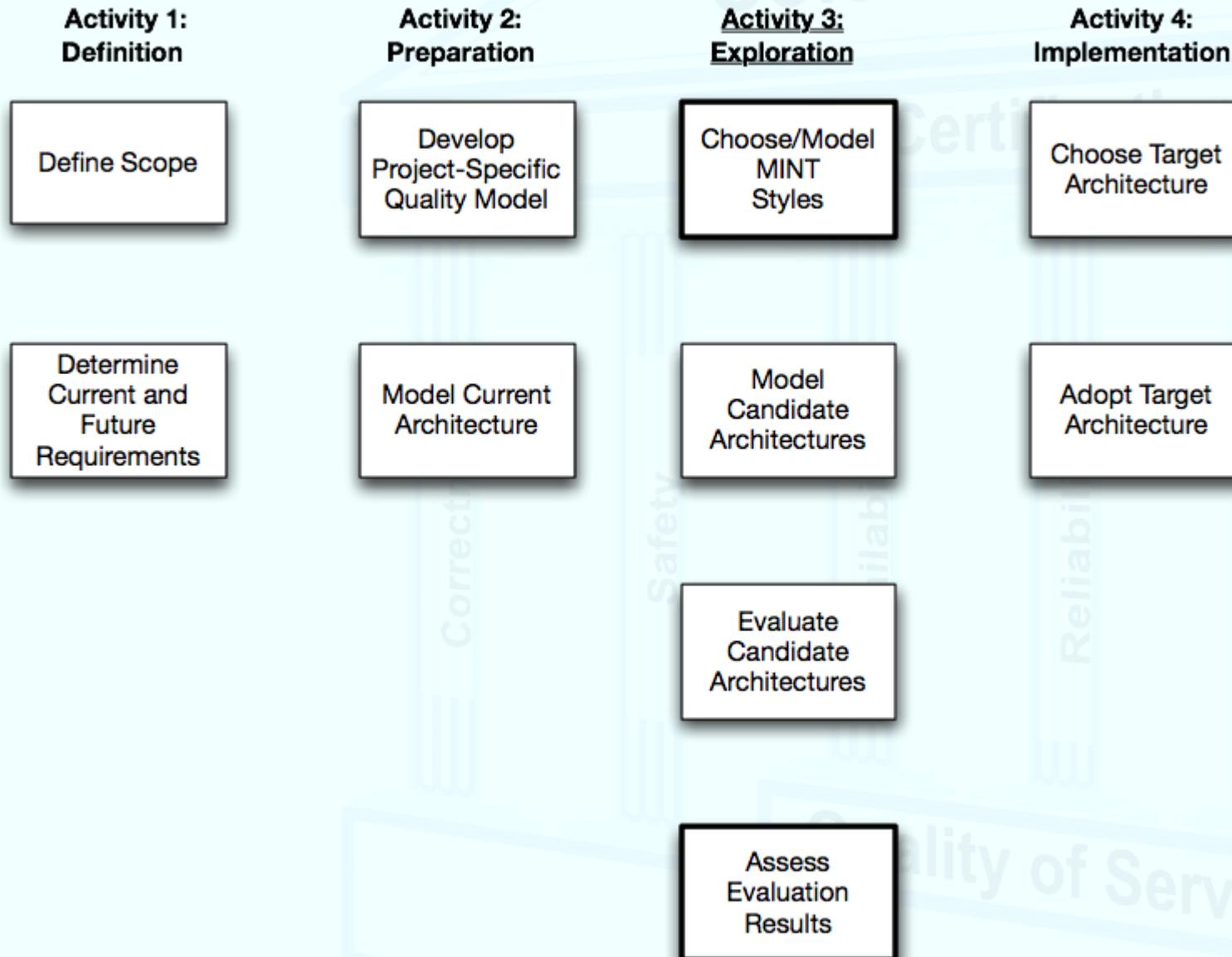


Research Project Overview





Method Overview





Trustworthy Software Systems

Certification

Part II Case Study

Correctness

Safety

Availability

Reliability

Performance

Security

Quality of Service

Component Technology



Case Study 2

- Role in Research Project
 - Apply Preliminary MIDARCH Method
 - Evaluate Feasibility of General Idea
 - Explore Refinements and Issues for MIDARCH Activities
- Current State
 - Ongoing
 - Activities 1 and 2 have been partially completed
- Regional Trade Information System
 - Local authorities <-> Local Companies



Activity 1: Definition

- Scope
 - Currently independent subsystems
 - Query Interface (web-based)
 - Management Interface (web-based)
 - Legacy Management and Analysis Interface (Java Swing)
 - Two separate relational databases with distinct schemas; proprietary file-based database
- Target: Coherent, more integrated system architecture

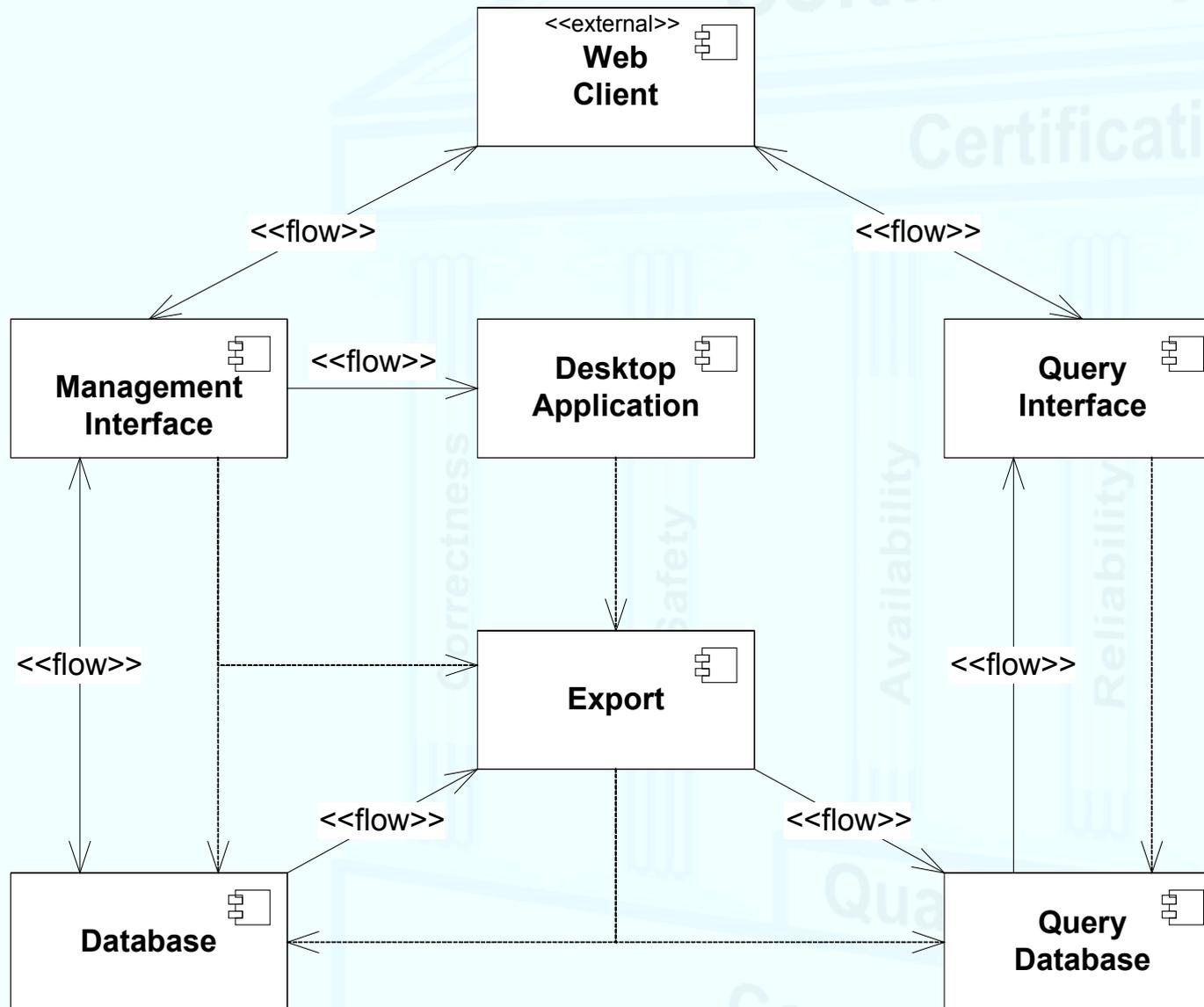


Activity 1: Definition (2)

- **Customer Goals**
 - **Improve Evolvability** (Architectural Quality)
 - Multiple Customer Support
 - Integration with other Information Systems
 - **Improve Availability** (System Quality)
 - **Improve Maintainability** (System Quality)
 - Enactment of Architectural Changes in the System Implementation

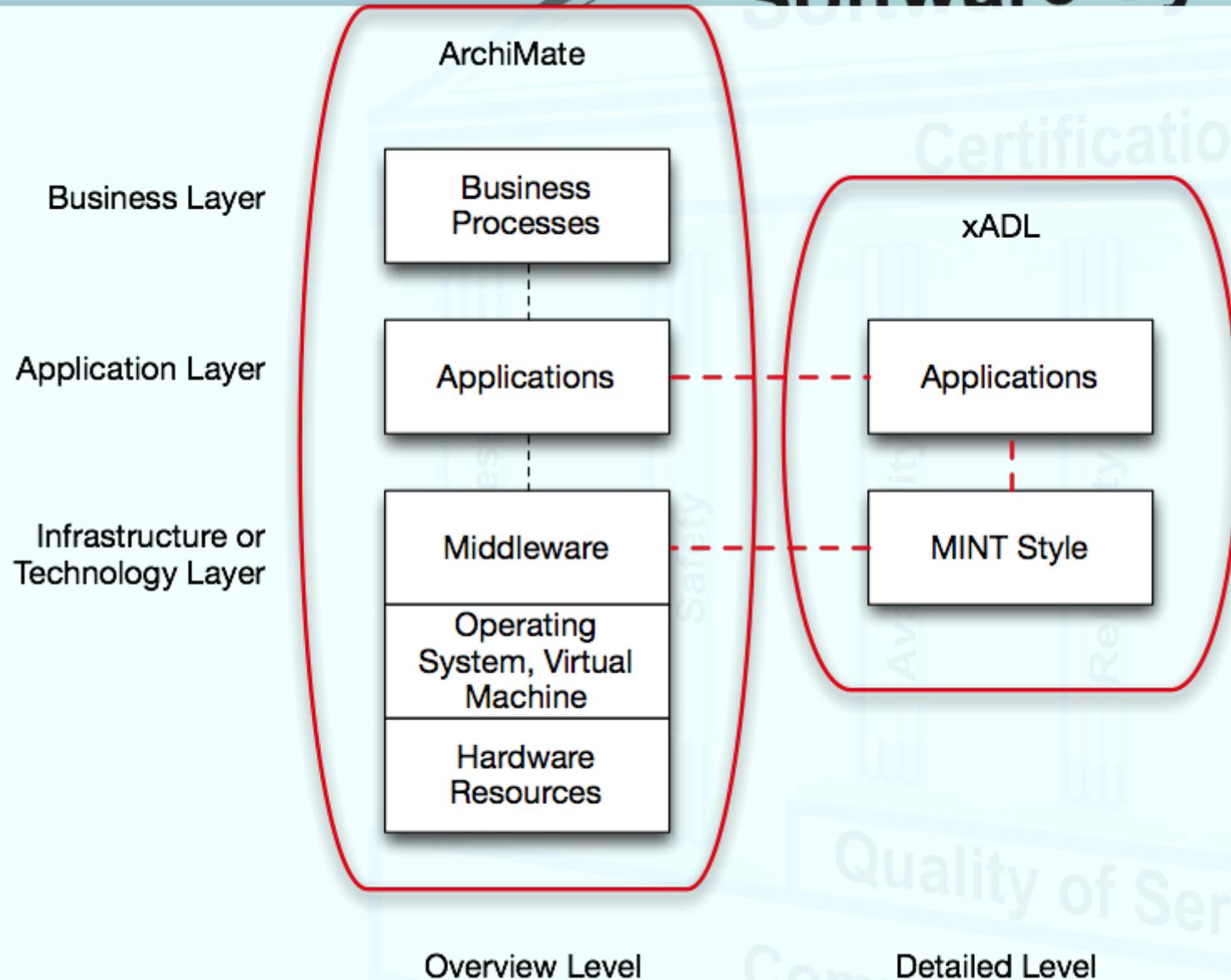


Activity 2: Preparation





Activity 3: Exploration





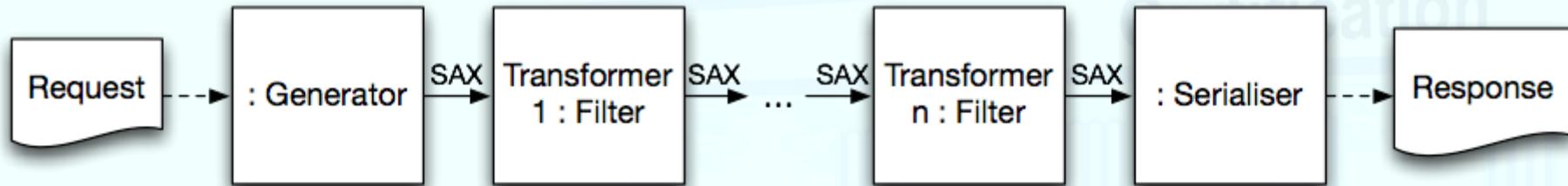
Problem Areas

- Lack of Coherence
 - Violations of Cocoon Style
 - Variations between Query and Management Intf.
- Proprietary Page Description Language
- Tight Coupling
 - Access to underlying database hard-coded
 - Within Data Tier of both Subsystems
- Code Clones
 - Parts of Query Interface copied to Management Interface
 - Independent evolution, manual synchronisation

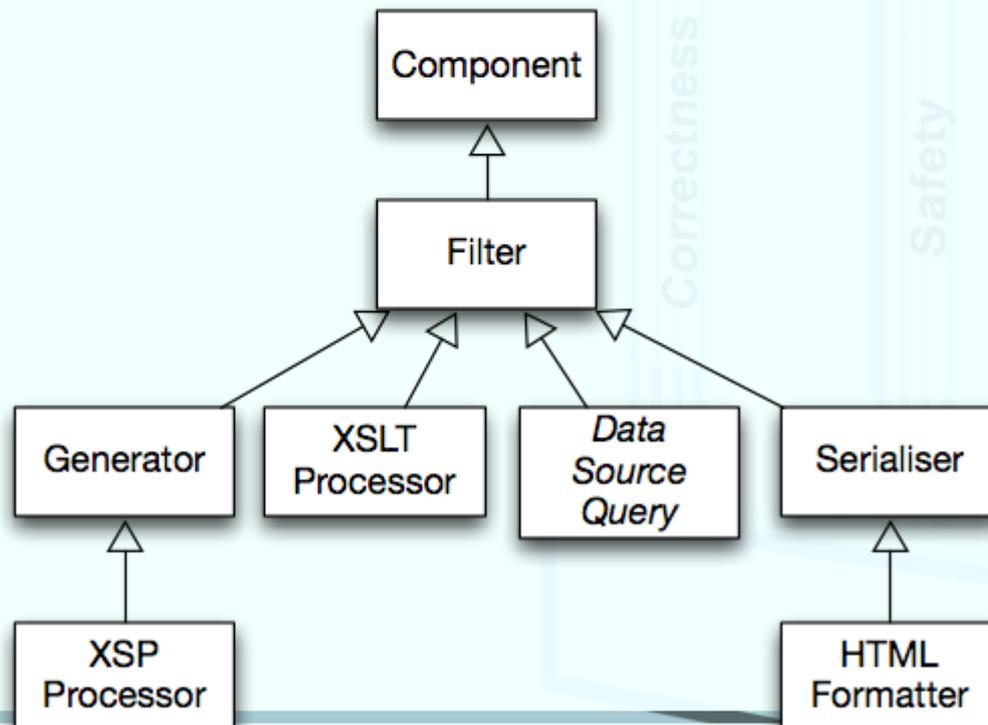


Cocoon Style

Generic Cocoon Style



Component Type Hierarchy



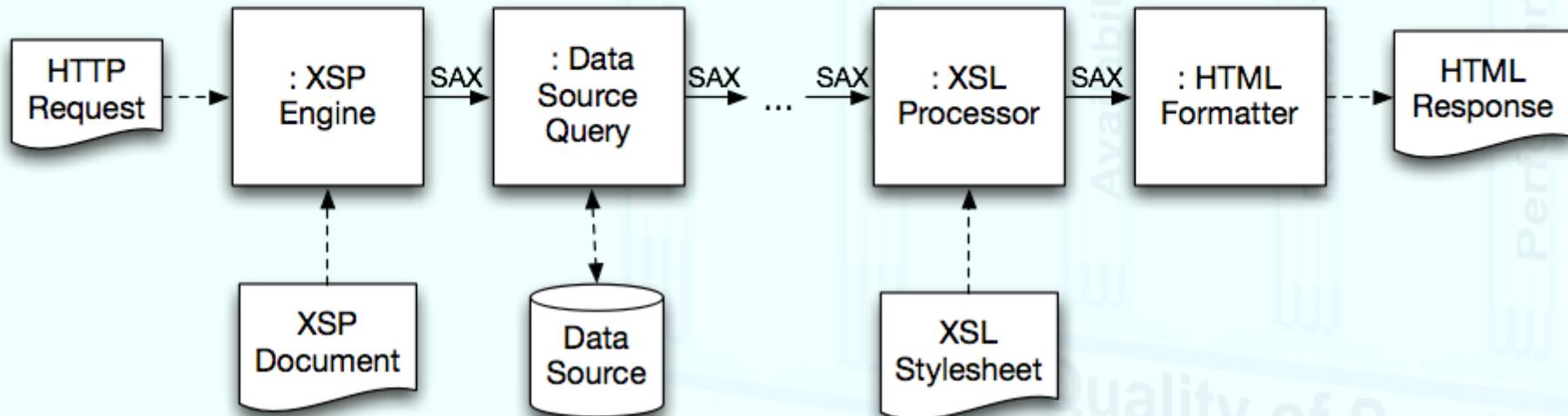


Cocoon Style (2)

Generic Cocoon Style



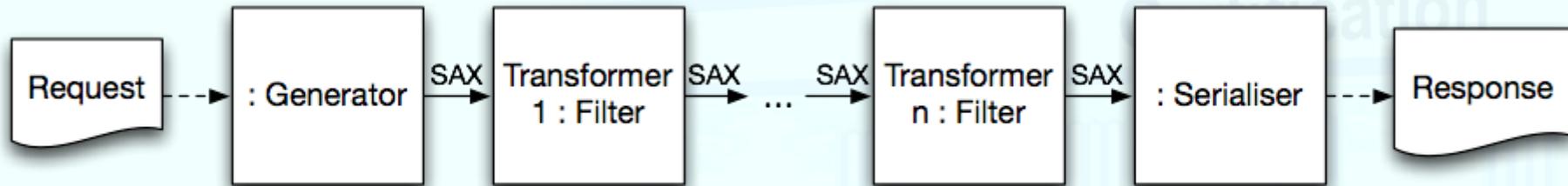
Var. A Cocoon Style



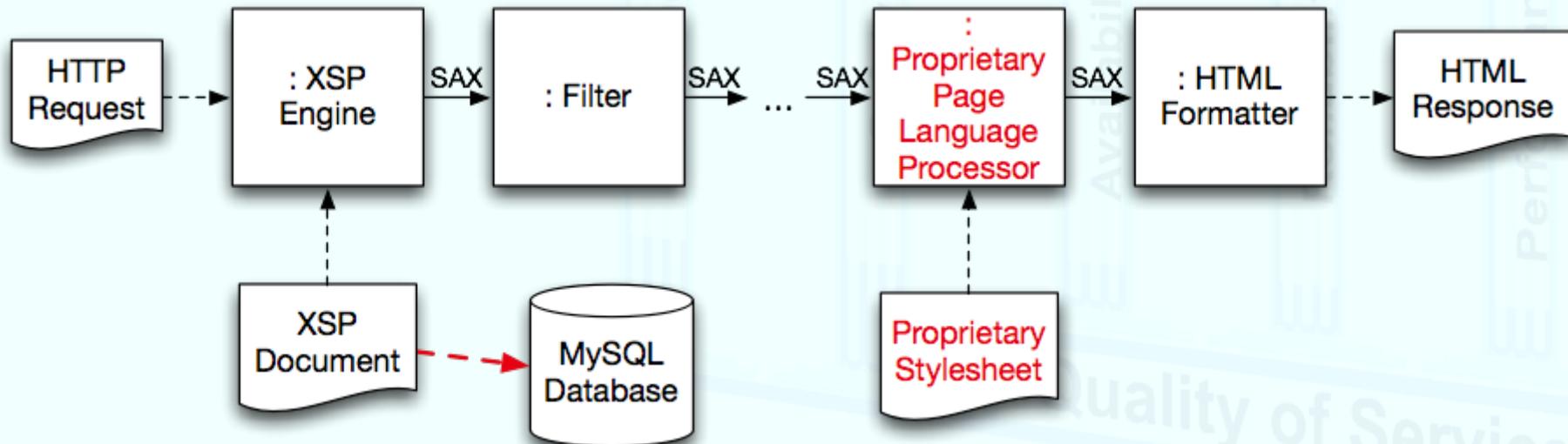


Cocoon Style (3)

Generic Cocoon Style



Actual REGIS Query Interface Pipeline





Conclusion

- MIDARCH Method for Supporting Integration/Migration Projects
- Case Study: Web-based Regional Trade Information System
- Next Steps
 - Define Cocoon Style Descriptions in xADL
 - Define Target Architectures Based on these Style Description
- Questions?