



Heiko Koziolk, Emanuel Kolb, Jens Doppelhamer • Industrial Software Systems, ABB Corporate Research

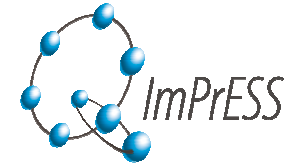
Architecture Model Reconstruction Towards Change Scenario Evaluation

Power and productivity
for a better world™



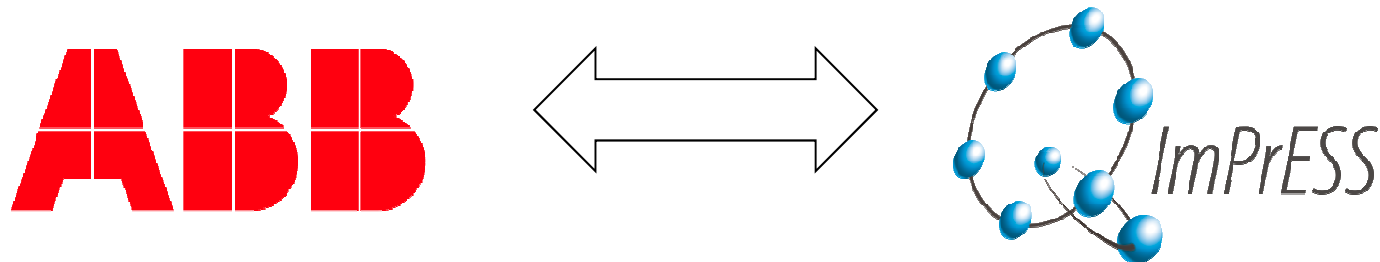
Introduction

Software Quality Prediction



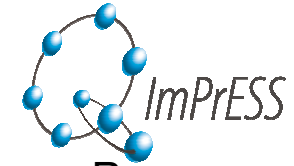
Goals for ABB

- Understand consequences that system changes have on quality attributes
 - Without costly try-and-error
 - Including existing/legacy software
- Understand trade-offs between quality attributes during evolution



Introduction: Q-ImPrESS¹

Software Quality Prediction



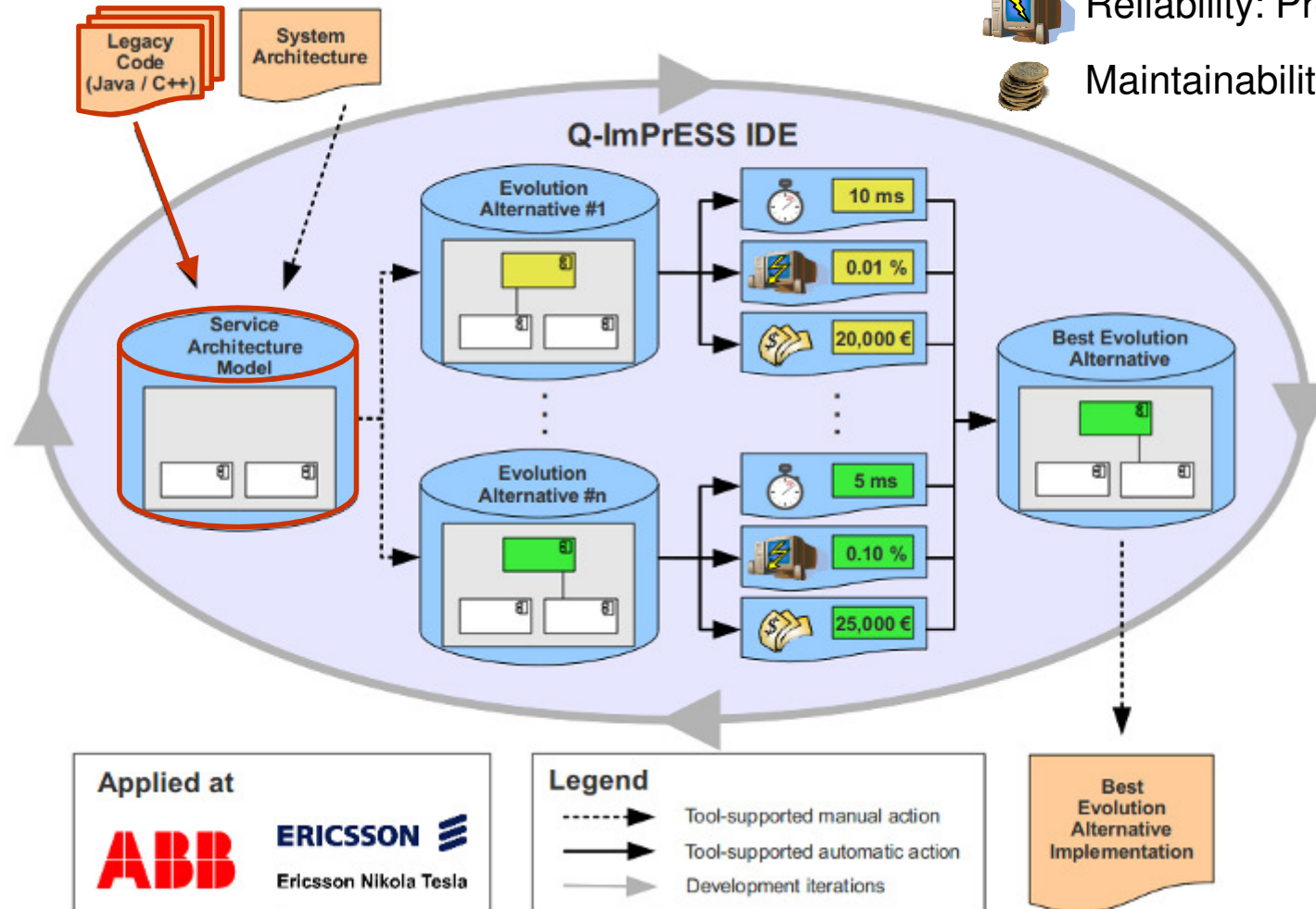
Performance: Response time



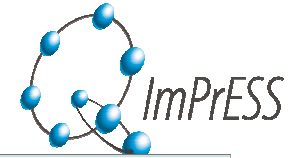
Reliability: Prob. of failure



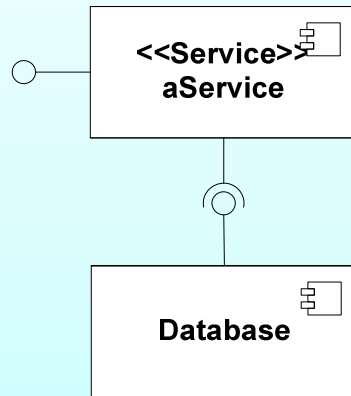
Maintainability: Cost



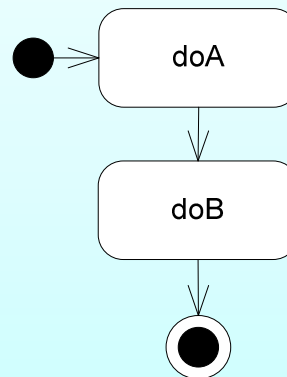
Q-ImPrESS Meta Model – Sample Models



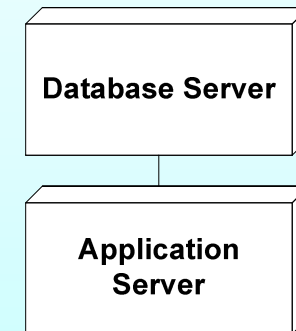
Static



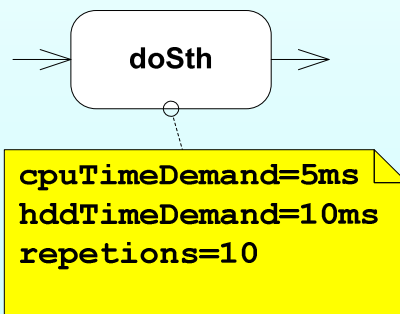
Behaviour



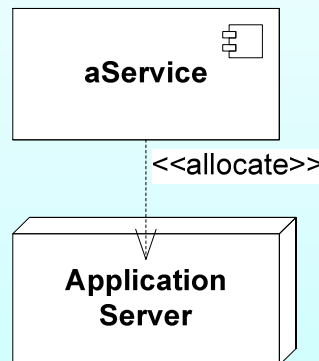
Resources



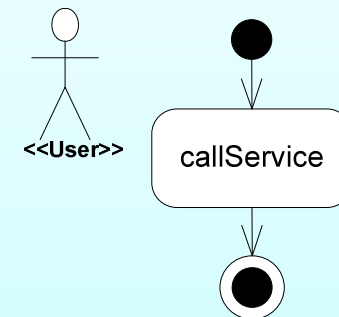
Annotations



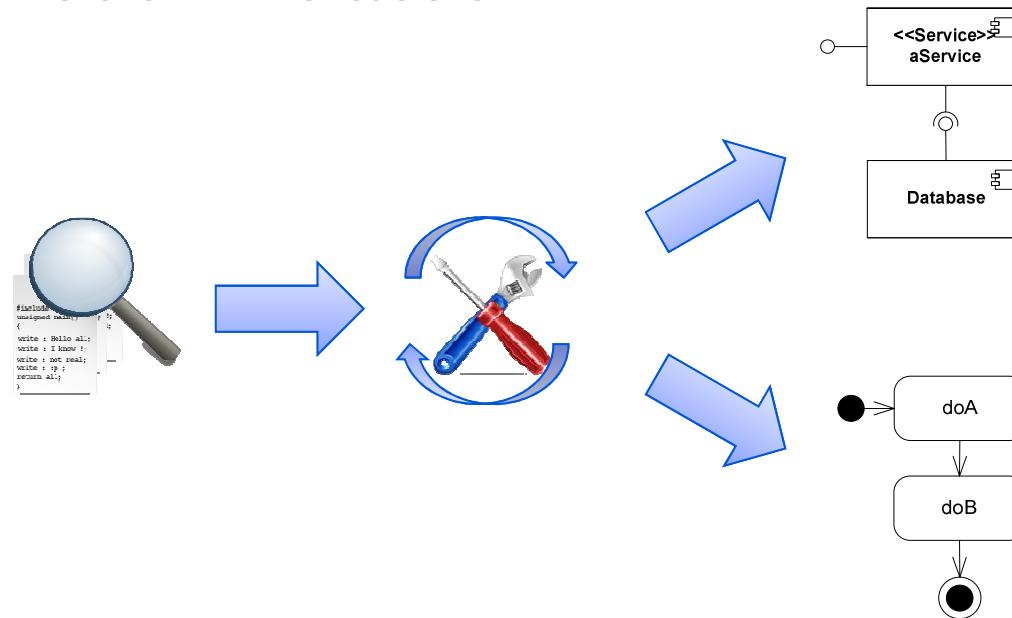
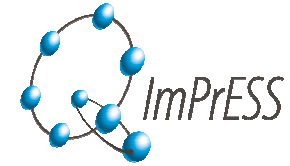
Allocation



Usage



Reverse Engineering Architecture Model Extraction

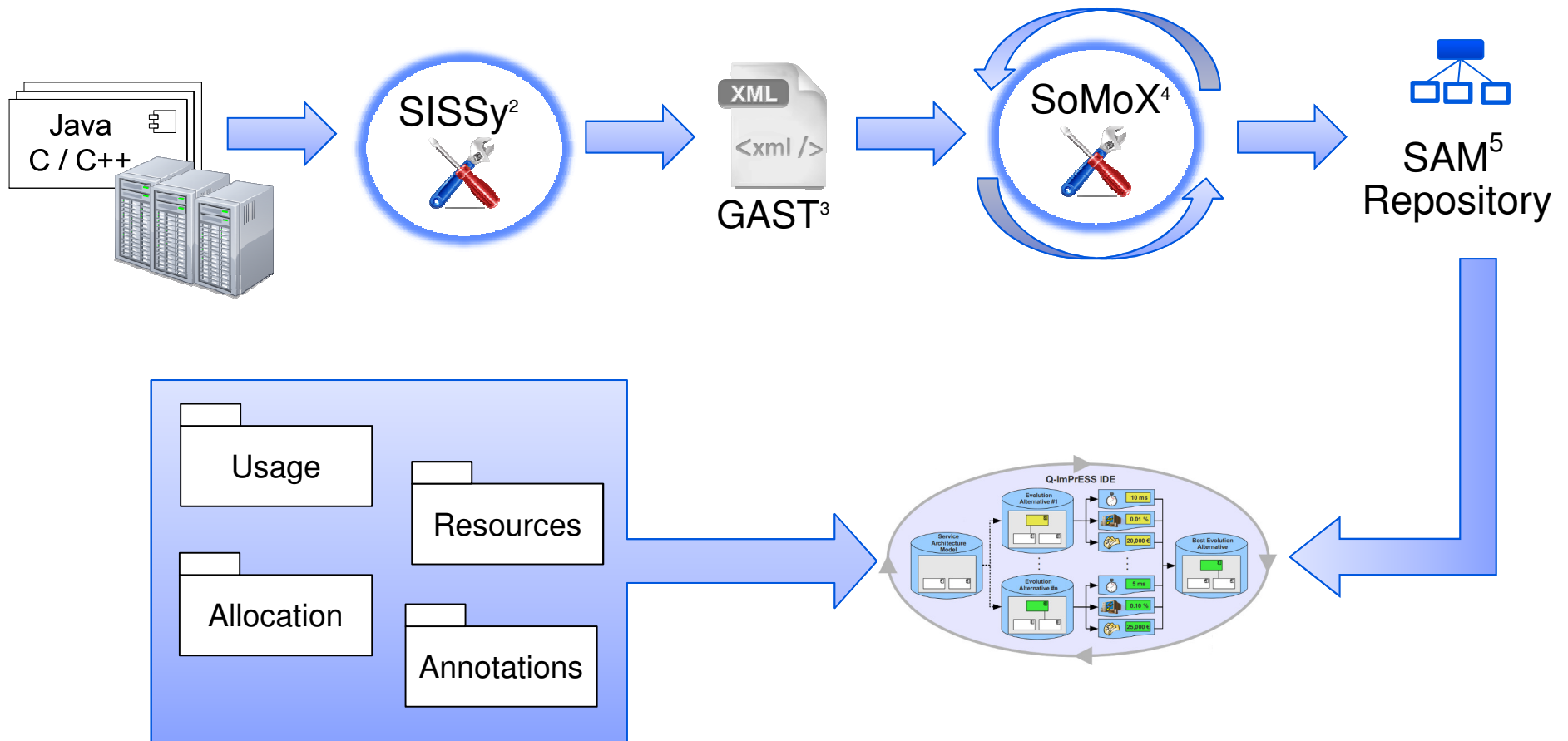
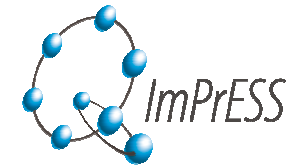


Problem:

- Suitable abstraction of the code base
- Suited for quality predictions
- Higher level components not explicit in the code

Q-ImPrESS

Reverse Engineering – Overall Workflow



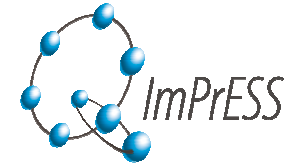
²Structural Investigation of Software Systems

³Generalized Abstract Syntax Tree

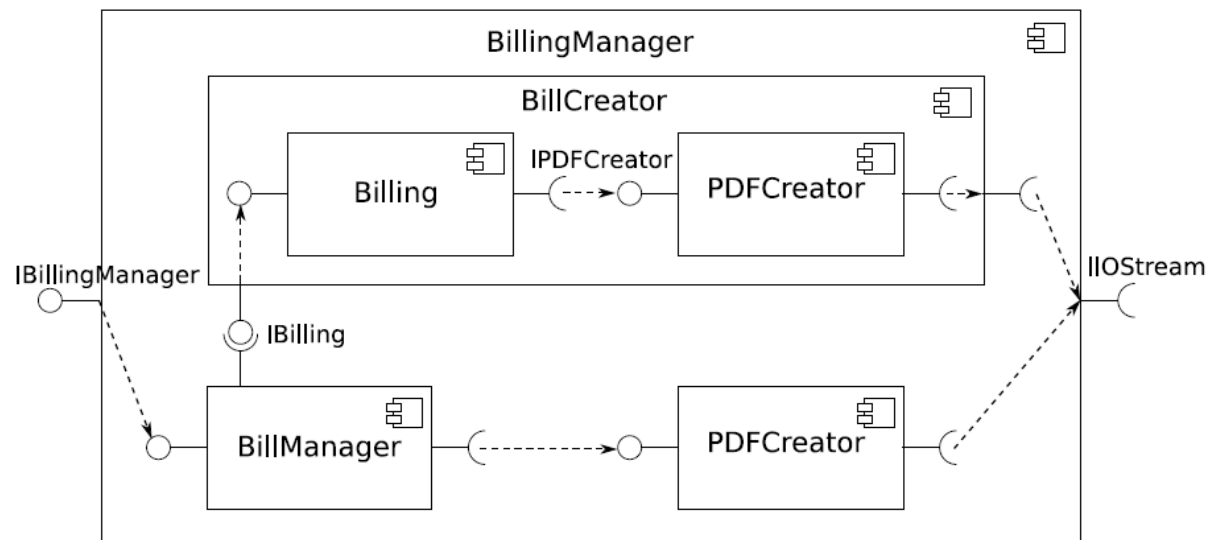
⁴Software Model eXtractor

⁵Service Architecture Model

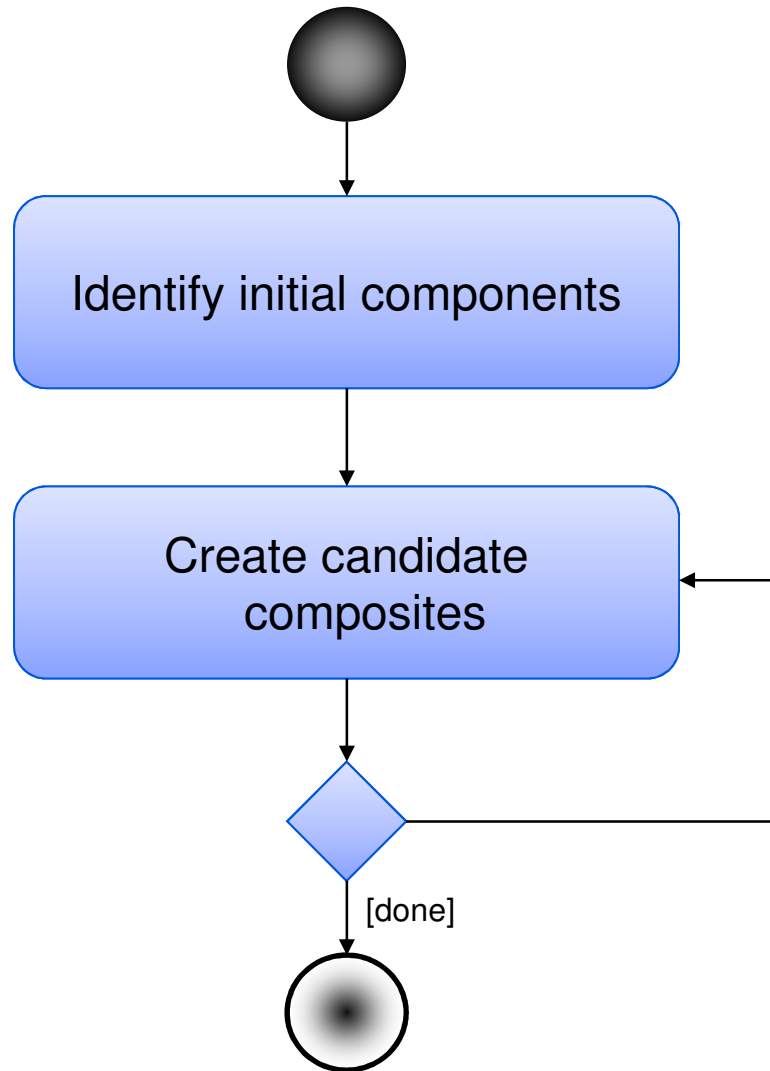
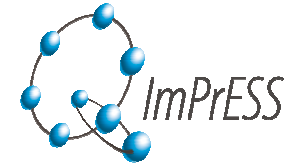
Model Extraction SoMoX Overview



- Underlying component concept:
 - Explicit interfaces
 - Composite components
- Target model defined within the Q-ImPRESS meta-model



SoMoX Component Identification

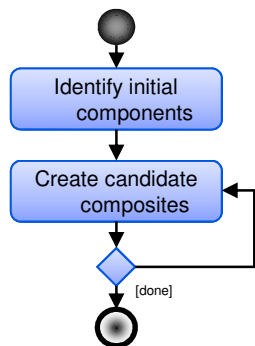
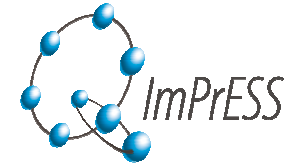


Classes with their required and provided interfaces

Based on OO metrics, compute score for candidates

SoMoX

Candidate Composites



OO Metrics used to evaluate candidates

- Distance from the main sequence
- Coupling and
 - Name resemblance
 - Interface violation
- + X

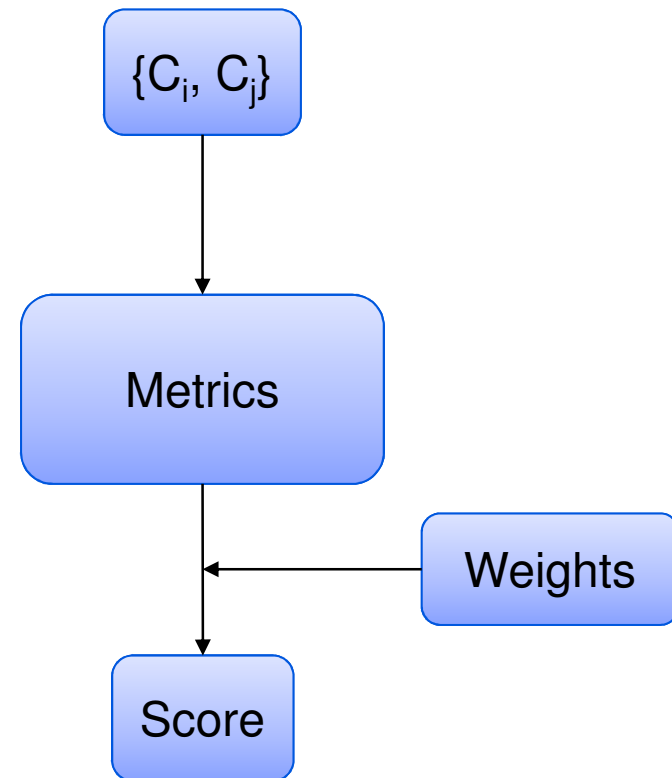
Metrics are combined

Metrics are weighted

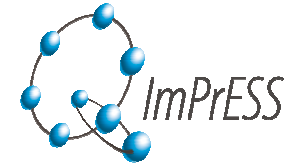
- System/technology specific

Metrics are computed pair wise

- Subsequent clustering



Case studies Overview

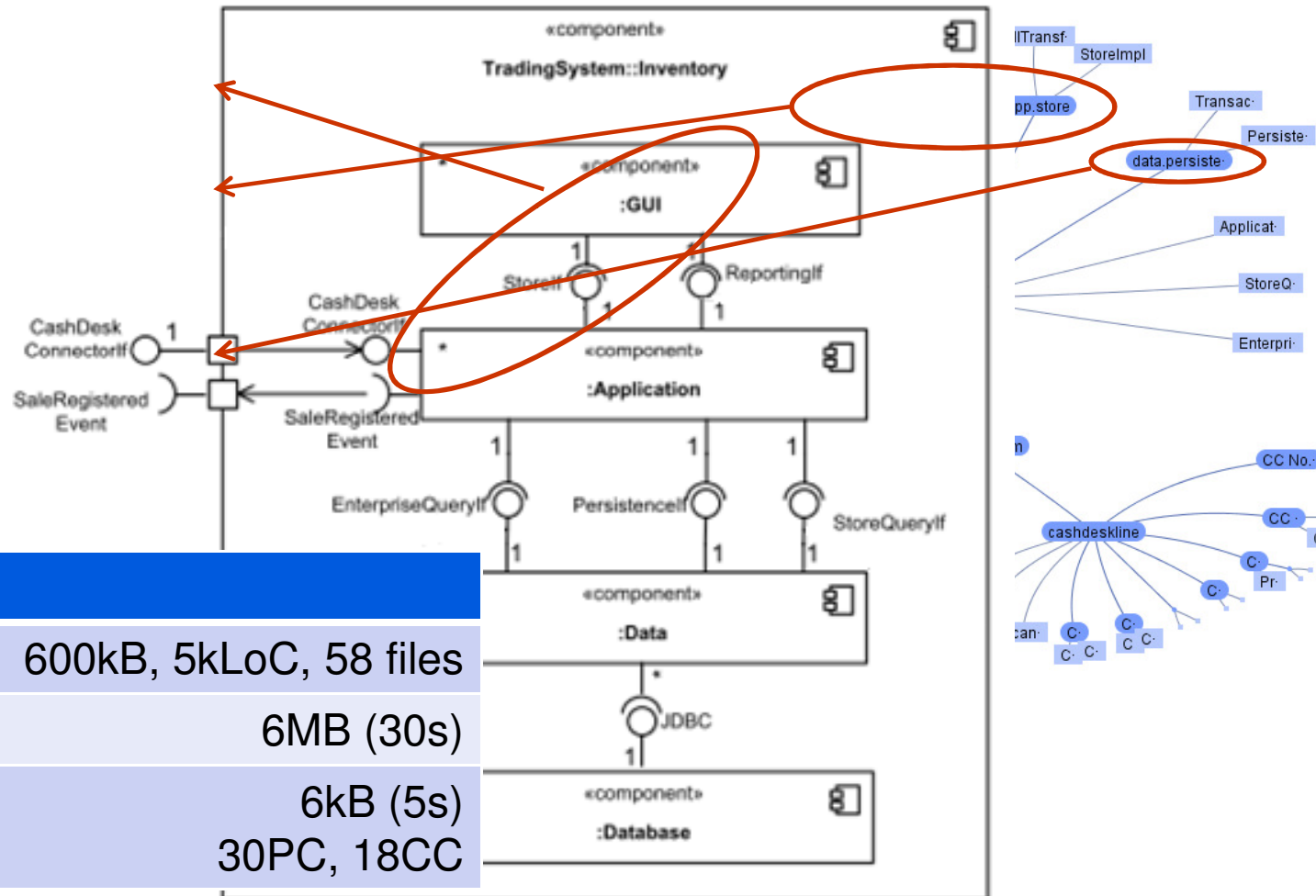
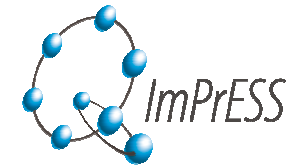


Java	LoC	# Files	Total file size
CoCoME	5k	125	600kB
Ant	200k	1690	10MB

C++	LoC	# Files	Total file size
Industrial I	10k	15	150 KB
Industrial II	50k	58	750 KB
Industrial III	150k	65	2.4 MB

Java: CoCoME⁶

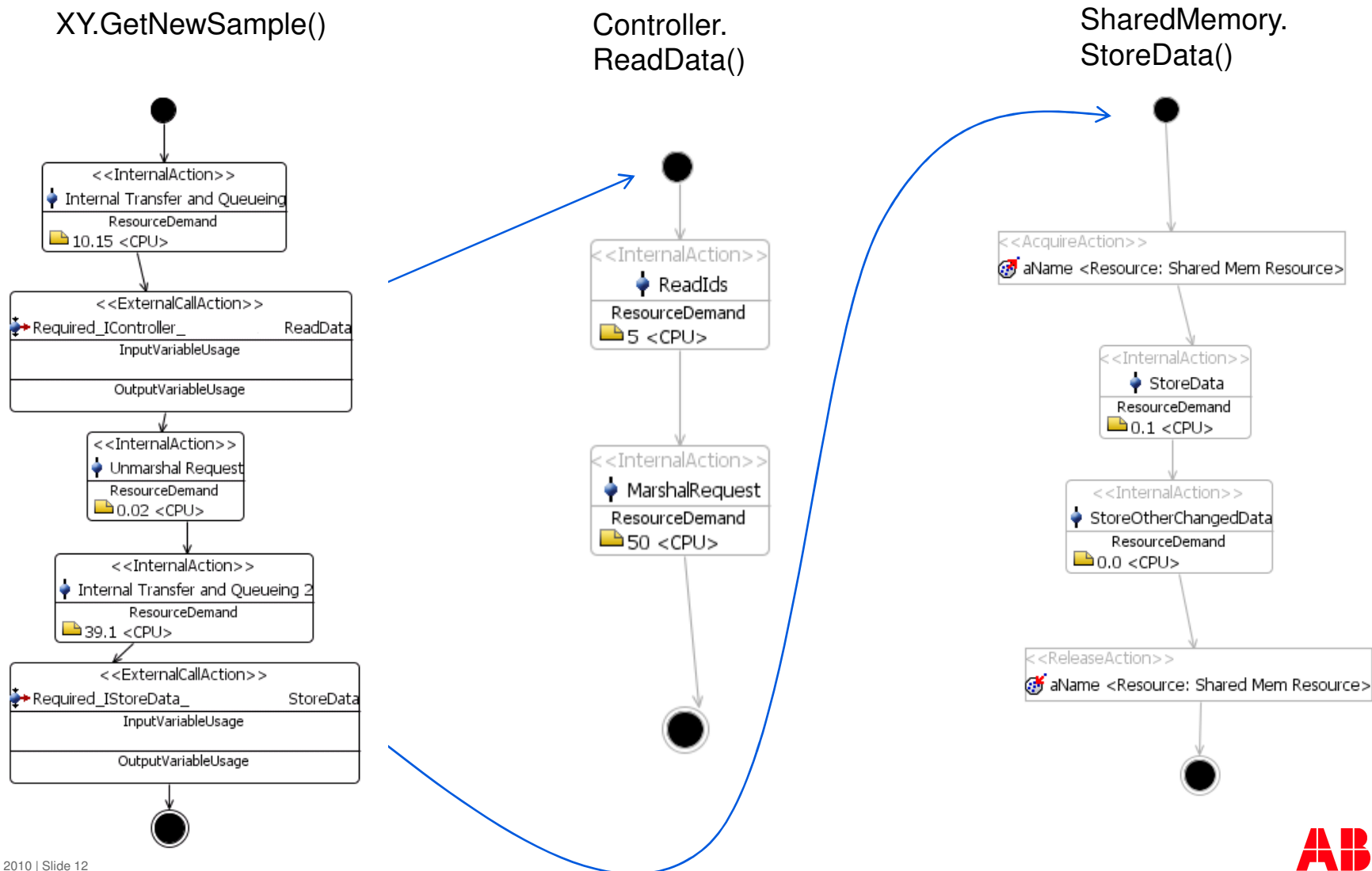
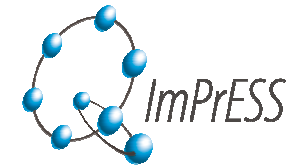
Exemplary implementation of a SOA



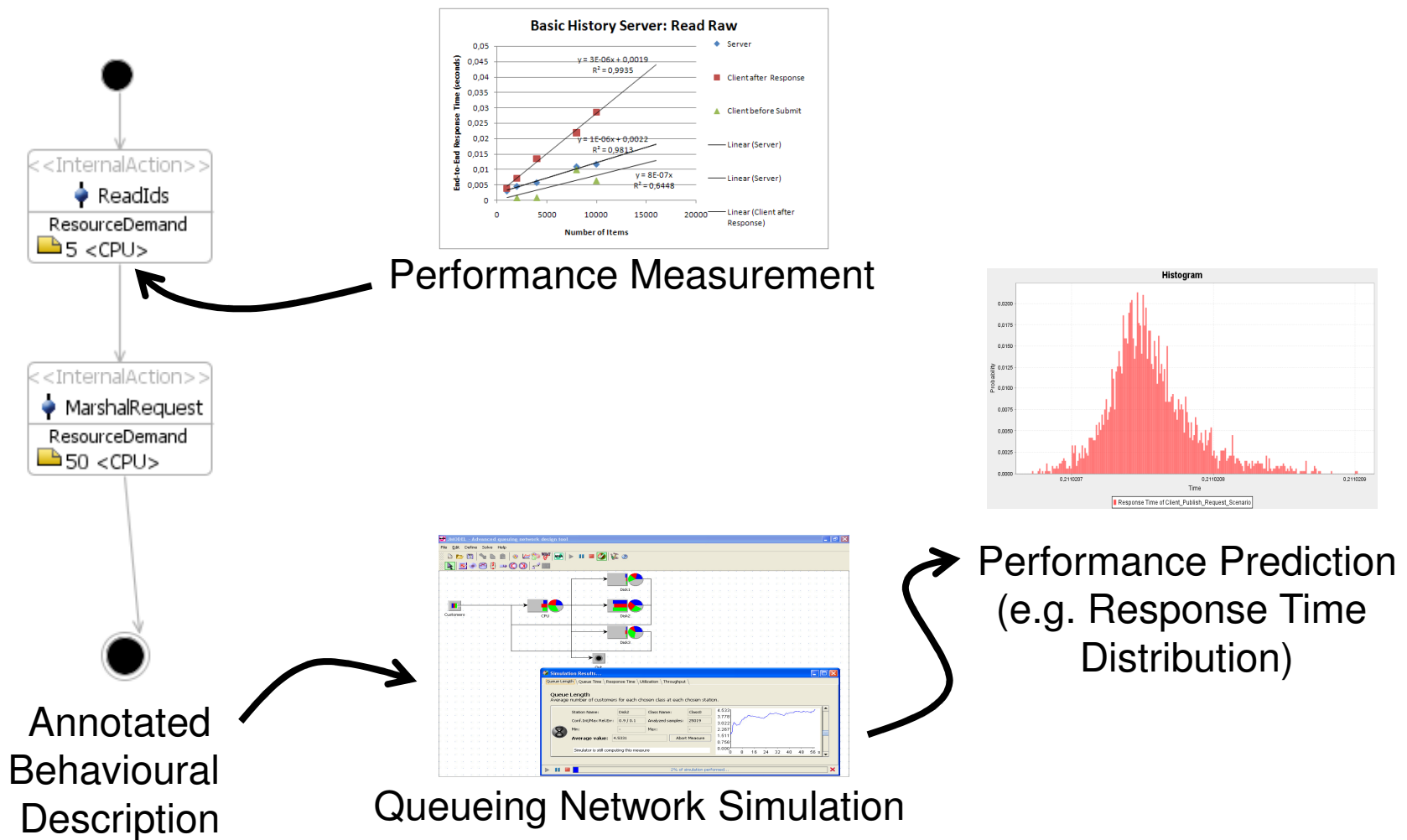
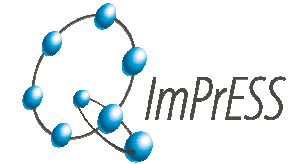
Stats

Sources	600kB, 5kLoC, 58 files
GAST	6MB (30s)
SAM Rep.	6kB (5s) 30PC, 18CC

Service Effect Specifications (SEFFs) Component Behavior

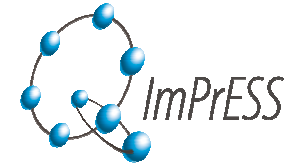


Industrial Case Studies Performance Prediction



Case studies

Summing up the results



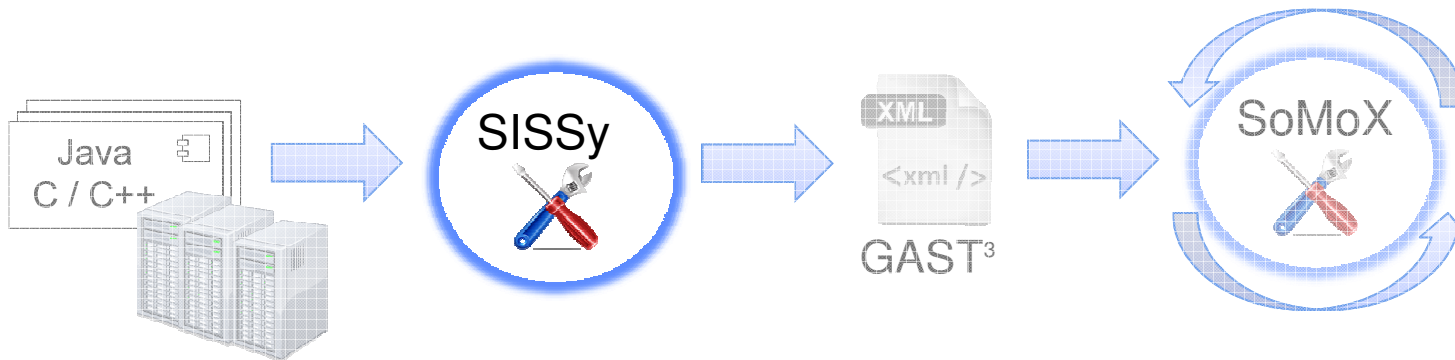
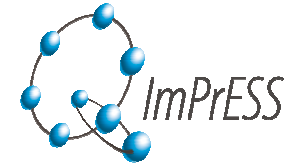
Java	LoC	Size	SISSy	SoMox
CoCoME	5k	600KB	++	++
Ant	200k	10MB	++	(++)

C++	LoC	Size	SISSy	SoMox
Industrial I	10k	150kB	●	●
Industrial II	50k	750kB	+	++
Industrial III	150k	2,4MB	✗	

++: very good; +: good; ●: ok; -: bad; --: very bad; ✗: No results

Q-ImPrESS

Reverse Engineering – Conclusion



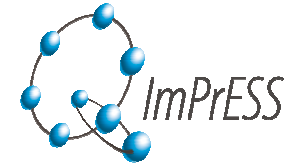
- Promising approach to reverse engineering for model based software quality prediction
- Support for C++ incomplete
 - Real world dialects (MS C++) problematic
 - No precompiled header support
=> Performance issues

Power and productivity
for a better world™



Q-ImPRESS

Further information



- <http://www.q-impress.eu>
- <http://jira.ow2.org/browse/QIMPRESS>
- <http://sdqweb.ipd.kit.edu/wiki/SoMoX>
- <http://sissy.fzi.de/>
- http://sdqweb.ipd.kit.edu/wiki/Palladio_Component_Model