

# Reconfigurable SCA Applications with the FraSCAti Platform

Lionel Seinturier - Philippe Merle - Damien Fournier - Nicolas Dolet (\*)

Valerio Schiavoni - Jean-Bernard Stefani (\*\*)

\* University of Lille & INRIA, France

\*\* INRIA, France

SCC 2009 – Bangalore, India

September 21-25, 2009

# Plan

1. Background & Motivations
2. Extended SCA Component Model
3. Platform Architecture
4. Platform Implementation & Evaluation
5. Conclusion & Future Work

# 1. Background & Motivations

## SCA (Service Component Architecture)

- a component model for SOA
- 11/2005

## Hosted by the Open SOA consortium

- <http://www.osoa.org>

## Standardised by OASIS

- <http://www.oasis-opencsa.org>

## Platform providers

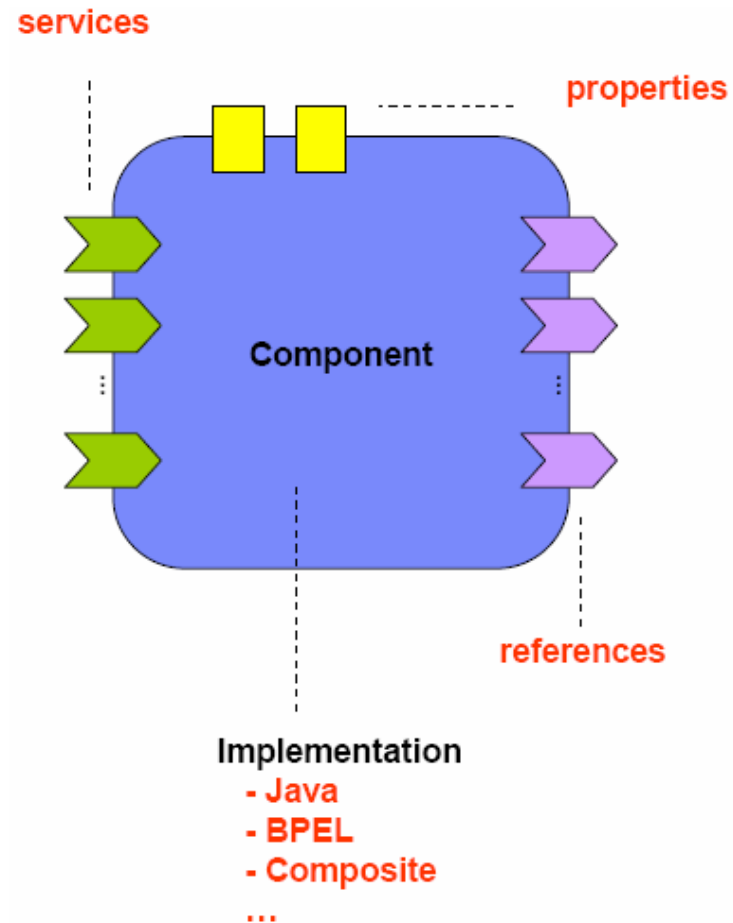
- Open Source: Apache Tuscany, Fabric3, FraSCAti, Mule, Newton
- Vendors: IBM WebSphere FP for SOA, TIBCO ActiveMatrix, Covansys SCA Framework, Paremus, Newton, Rogue Wave HydraSCA, Oracle Fusion Middleware



# 1. Background & Motivations

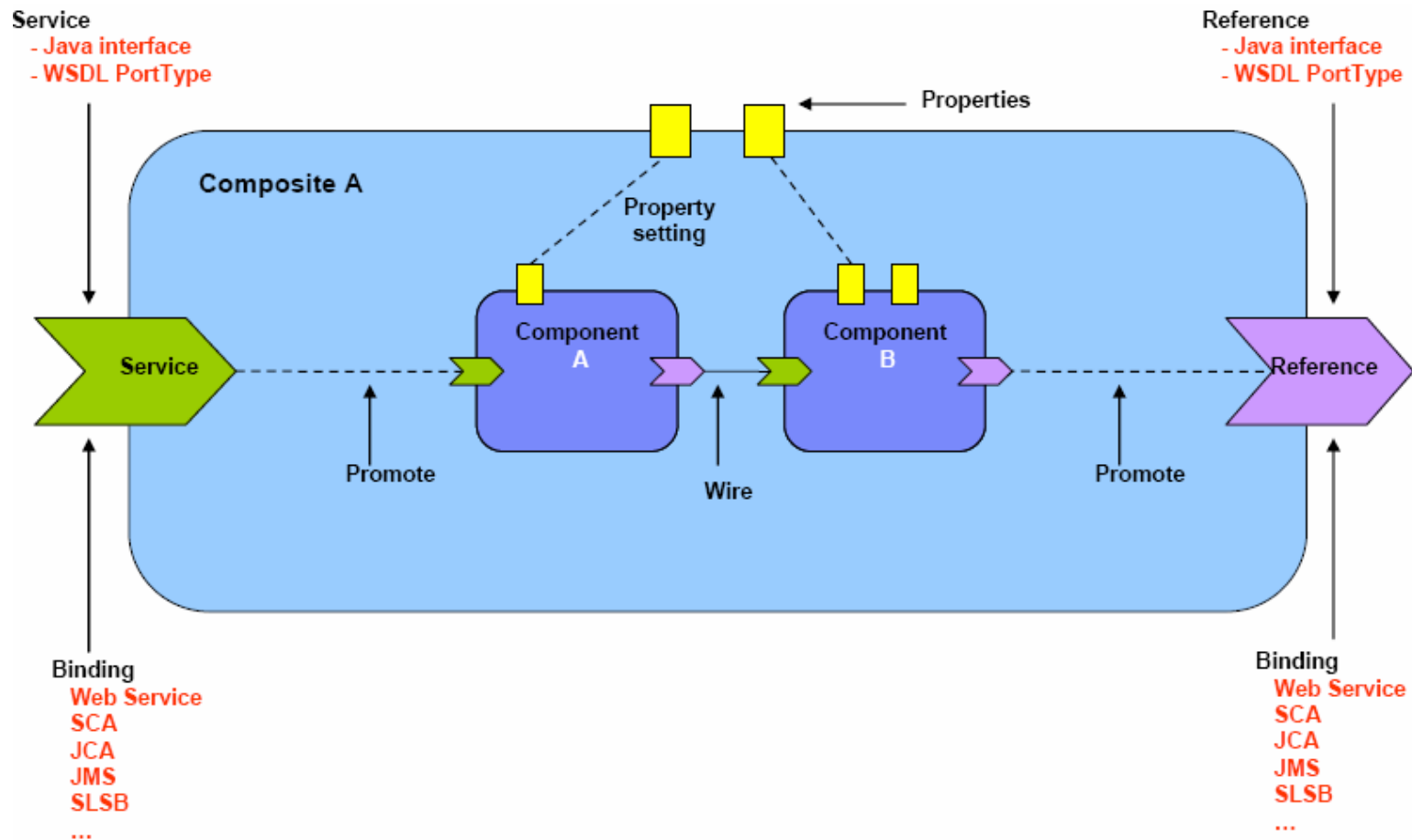
## SCA Component Implementation Specifications

- service/reference
- property
- implementation
  
- non functional property (intent & policy)



# 1. Background & Motivations

## SCA Assembly Model Specification



# 1. Background & Motivations

SCA is independent from

- programming languages
- interface definition languages
- communication protocols
- non-functional properties

SCA platform challenge

- dynamically reconfigurable runtime architectures  
research challenge #1 in Service foundations [Papazoglou 07]

## 2. Extended SCA Component Model

To meet this challenge

- extension of the SCA component model with reflective capabilities
  - for introspection, monitoring, control, dynamic reconfiguration

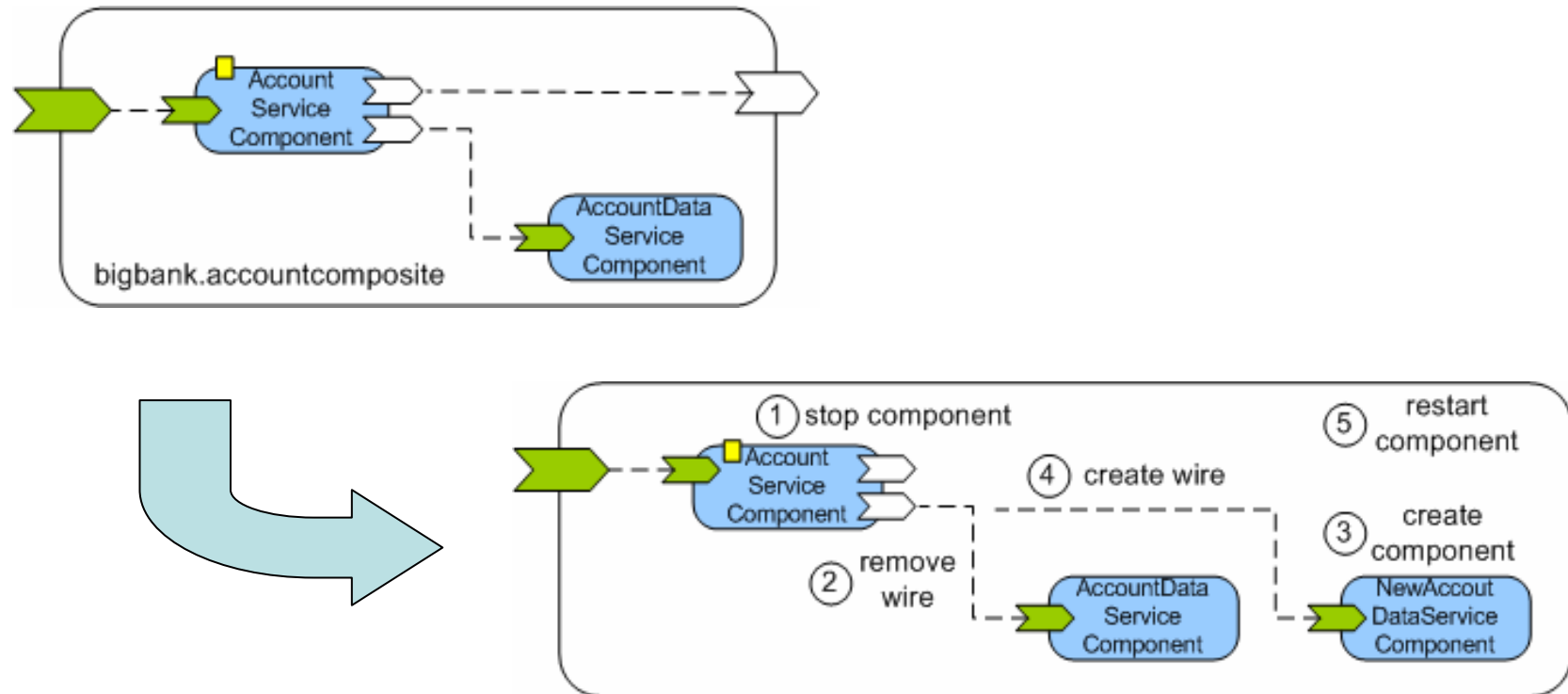
at 2 levels

- component container architecture
- support for non-functional services

## 2. Extended SCA Component Model

Component container architecture

- provide support for component & architecture reconfiguration

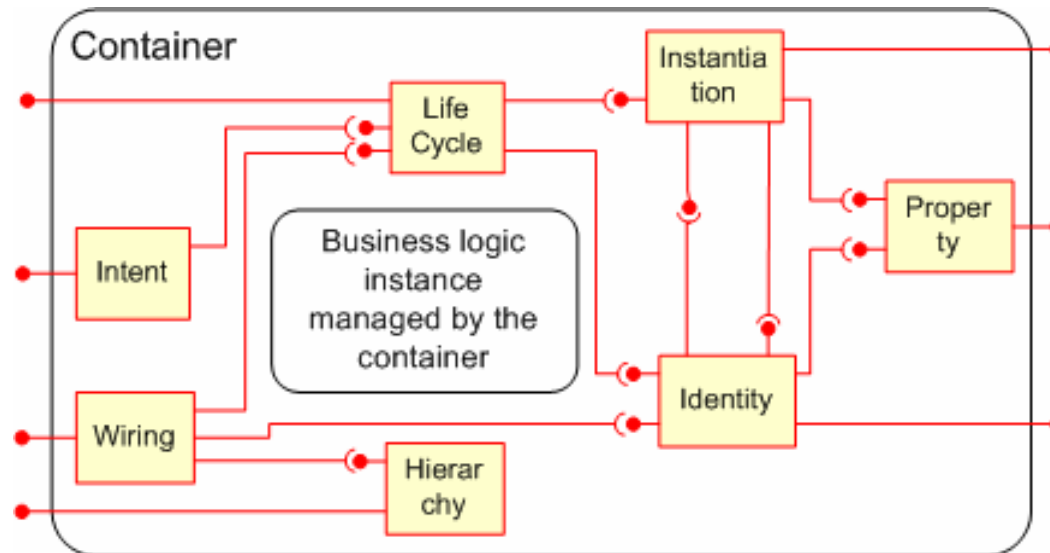




## 2. Extended SCA Component Model

### Component container architecture

- host a business logic instance which provides the component services
- generalize the notion of a meta-level
- decomposed in some fine-grained services

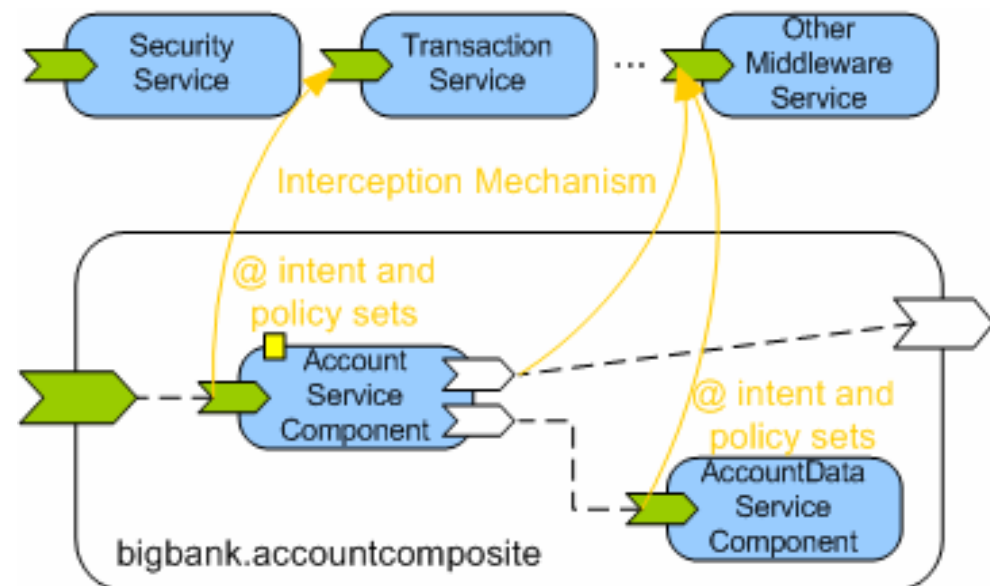


## 2. Extended SCA Component Model

Support for non-functional (NF) services

- SCA Policy Framework provides some metadata
  - @Confidentiality, @Integrity, @Authentication
  - general purpose: @Intent, @Requires

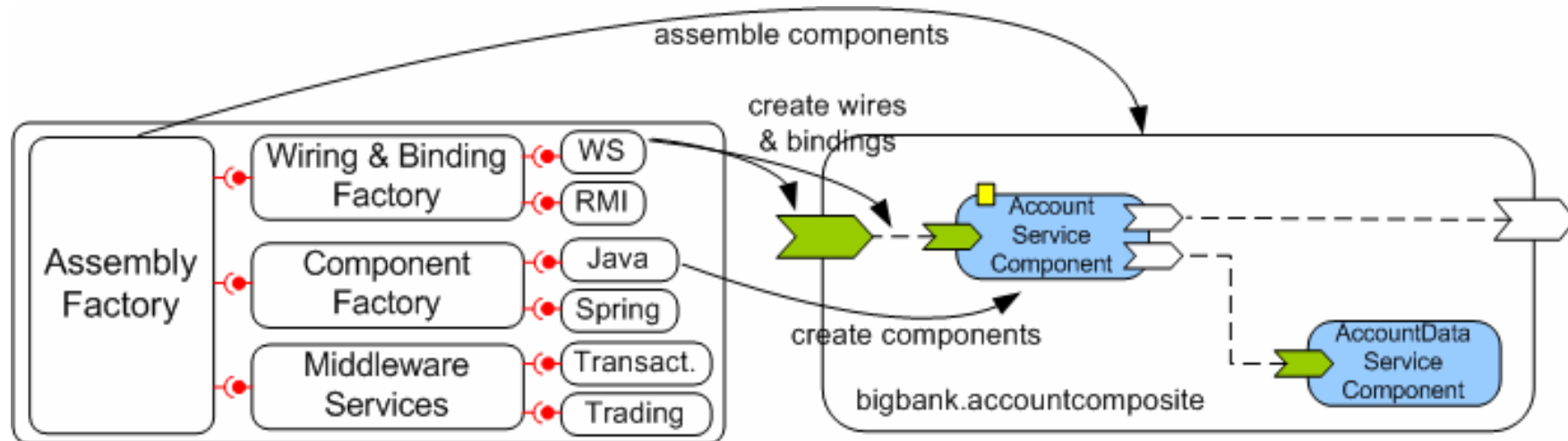
- NF services implemented as SCA components
- NF wiring between business components and NF components
- API for dynamic management



# 3. Platform Architecture

Modular (plugin like) architecture to support variation points

- implementation types
  - Java 5, Java POJO, Spring, OSGi, Java supported scripting languages, Scala
- binding types
  - SOAP, RMI, OSGi, REST



# 4. Platform Implementation & Evaluation

OW2 (consortium for open source middleware) project

- [frascati.ow2.org](http://frascati.ow2.org)

Architecture implemented in Java

with the Fractal OW2 lightweight component framework

- dependency injection
- Java 5 @-based development style
- XML-based component & architecture descriptors
- structuring concepts (component personality, membrane, control interface)

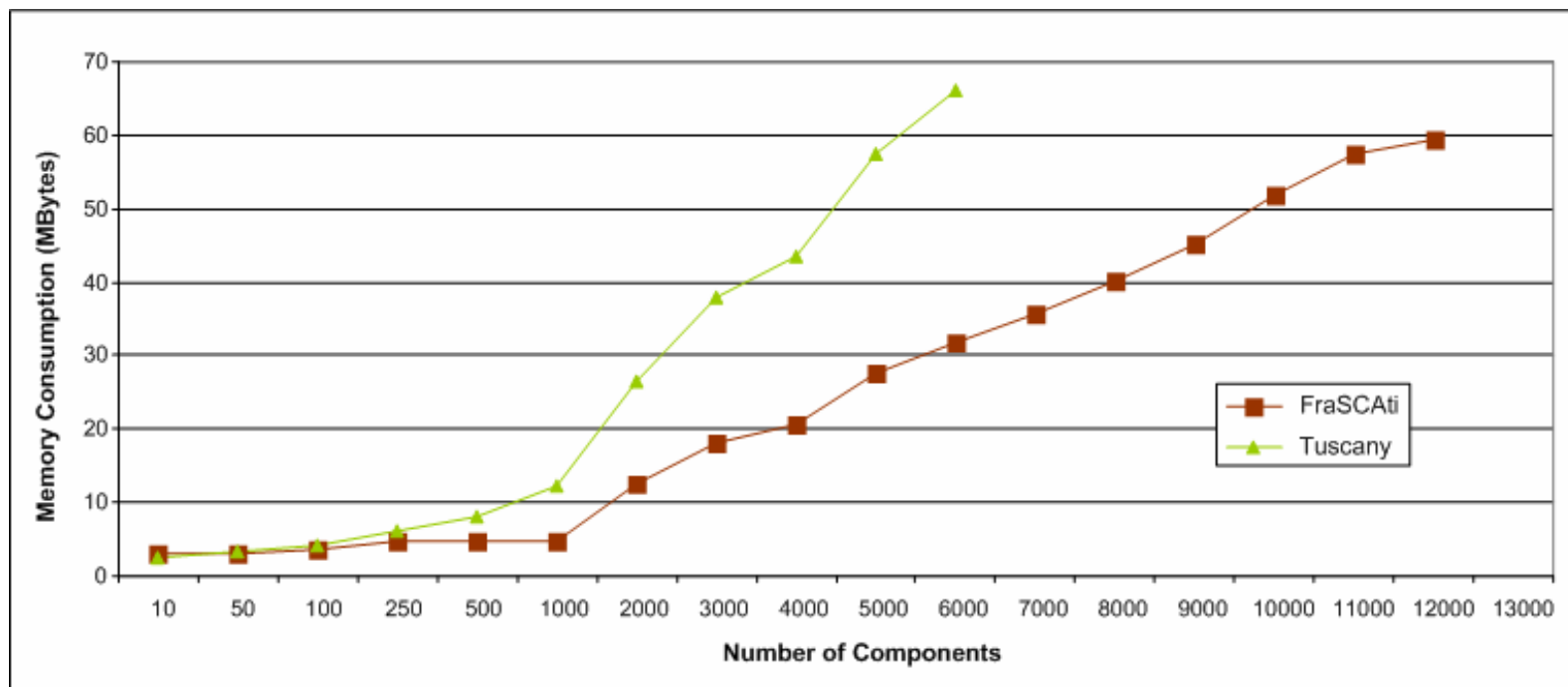
2 modes for the FraSCAti platform

- standalone application server
- integrated in the PEtALS OW2 JBI ESB

# 4. Platform Implementation & Evaluation

## Micro-benchmark

- vs Apache Tuscany

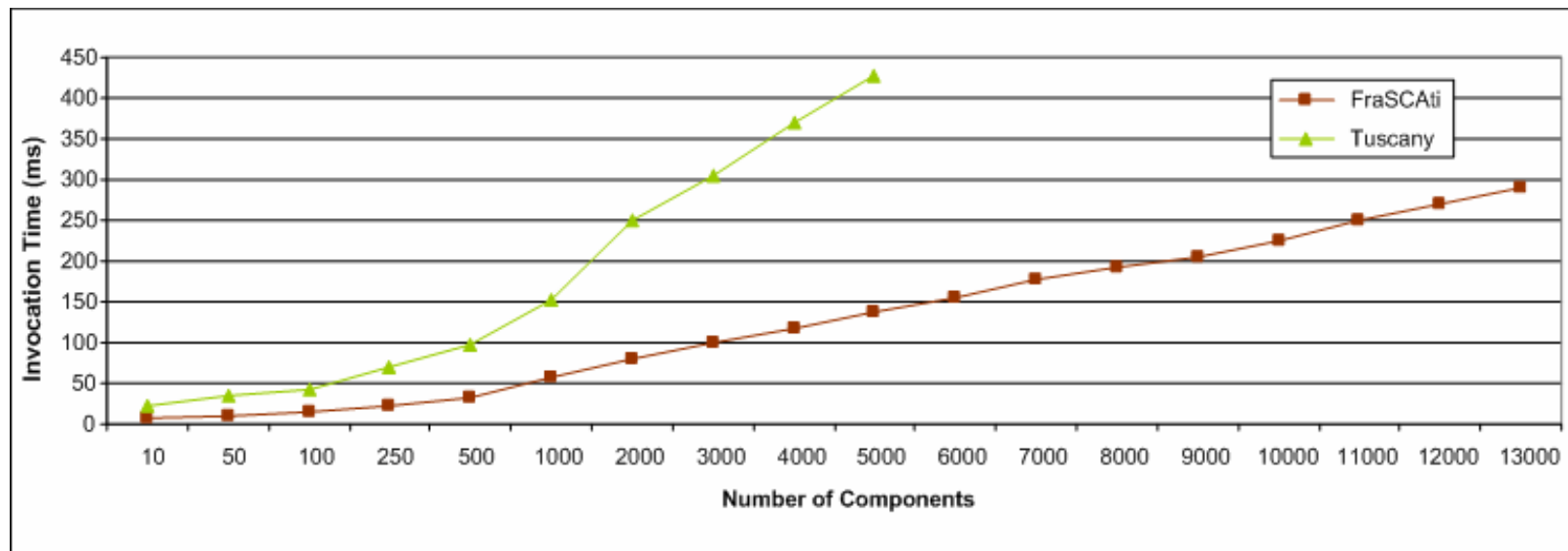


Platform memory consumption (per instantiated components)

# 4. Platform Implementation & Evaluation

Micro-benchmark

- vs Apache Tuscany



Component invocation time

# 5. Conclusion & Future Work

FraSCAti: A flexible & extensible SCA platform with

- runtime adaptation
- manageability properties
- reflective capabilities

3 original characteristics

- component-based structure for the platform itself
- non-functional (NF) properties provided as SCA components
- dynamic wiring/unwiring of NF components on business components

# 5. Conclusion & Future Work

## Future work

- extend the platform with new implementation and binding types
- platform as an assembly of SCA components
- from interception-based to aspect-orientation
  - extension of the Assembly Language grammar with AOP notions (pointcut)
- widen the scope of targeted application domains
  - from SOA for IT to SOA for the Internet of Things