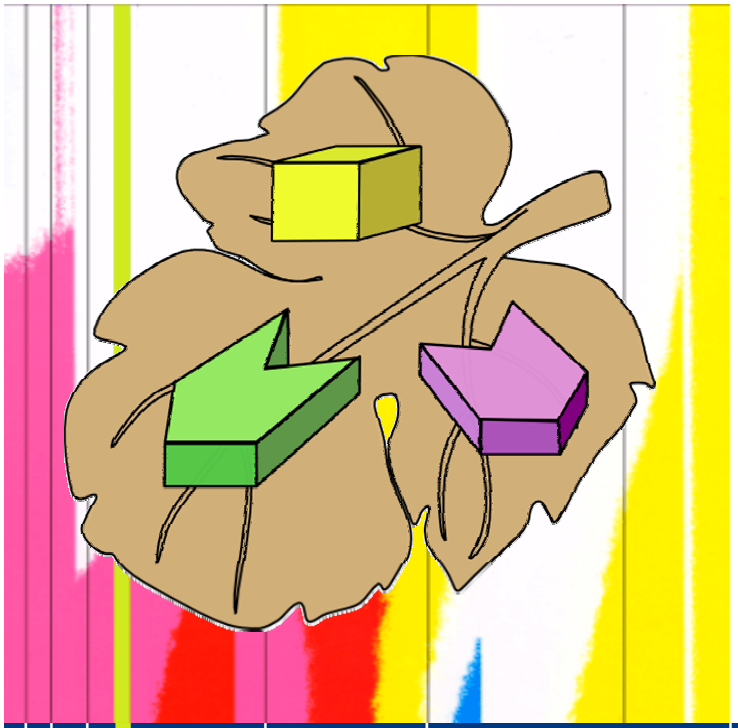


# FraSCAti

*«Open SCA Platform»*


The FraSCAti team
Presenter: Lionel Seinturier


INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche LILLE - NORD EUROPE

Université Lille1 Sciences et Technologies

lifL

cnrs

INRIA Team ADAM

OW2 Consortium

# Outline

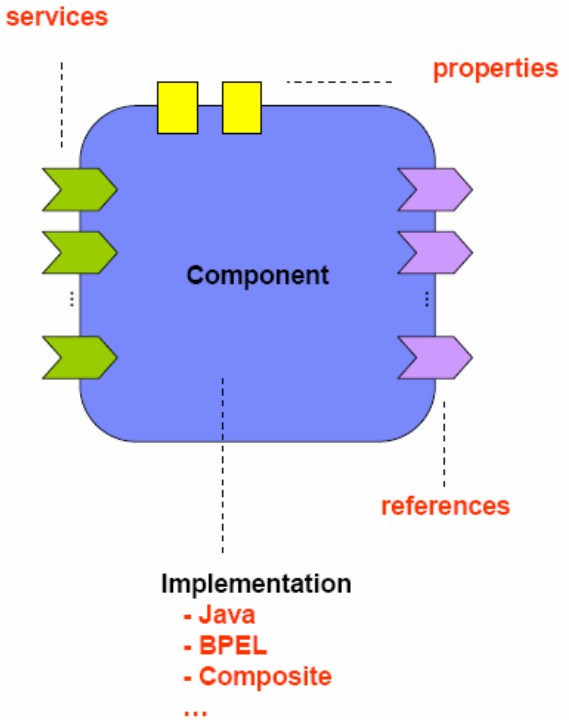
OW2 Consortium
FraSCAti

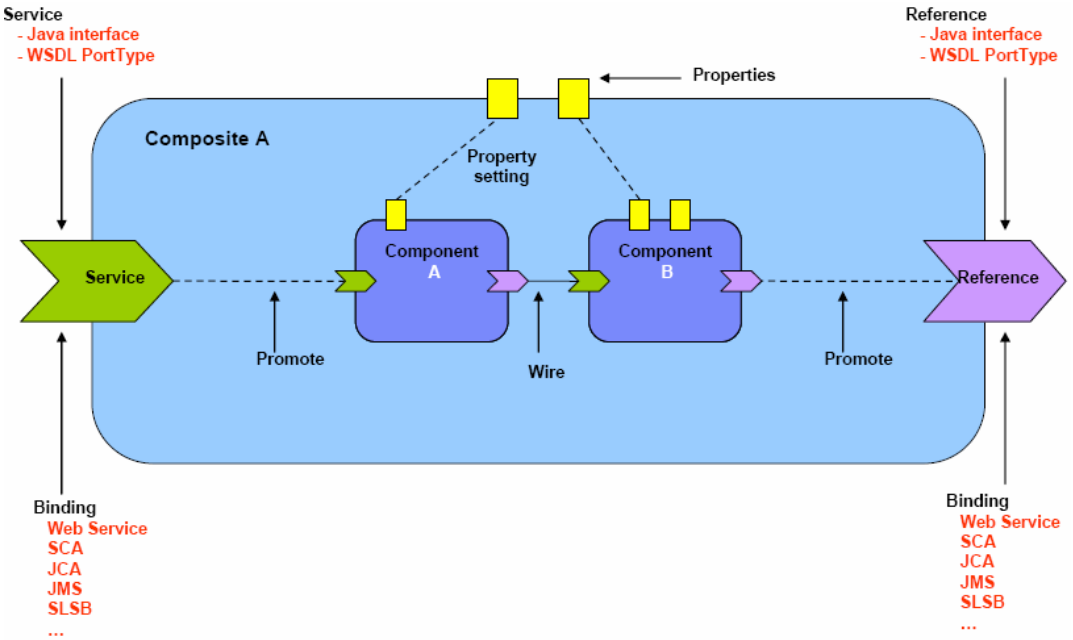
- SCA in a nutshell
- Motivation & principles
- Architecture
- Positioning wrt other platforms
- Conclusion

# SCA in a Nutshell

OW2 Consortium
FraSCAti

## Business Components & Assemblies



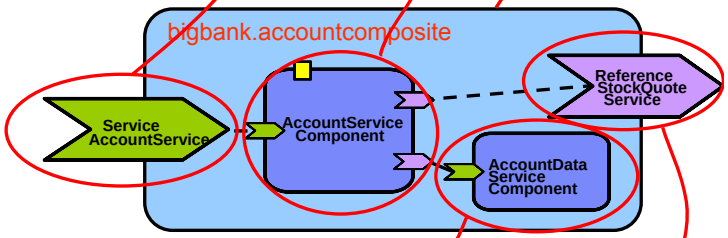
```
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0"
           name="bigbank.accountcomposite" >

    <service name="AccountService" promote="AccountServiceComponent">
        <interface.java interface="services.account.AccountService"/>
        <binding.ws port="http://www.example.org/AccountService#
          wsdl.endpoint(AccountService/AccountServiceSOAP)"/>
    </service>

    <component name="AccountServiceComponent">
        <implementation.java class="services.account.AccountServiceImpl"/>
        <reference name="StockQuoteService"/>
        <reference name="AccountDataService"
                   target="AccountDataServiceComponent/AccountDataService"/>
        <property name="currency">EURO</property>
    </component>

    <component name="AccountDataServiceComponent">
        <implementation.bpel process="“QName"/>
        <service name="AccountDataService">
            <interface.java interface="services.accountdata.AccountDataService"/>
        </service>
    </component>

    <reference name="“StockQuoteService" promote="AccountServiceComponent/StockQuoteService">
        <interface.java interface="services.stockquote.StockQuoteService"/>
        <binding.ws port="http://example.org/StockQuoteService#
          wsdl.endpoint(StockQuoteService/StockQuoteServiceSOAP)"/>
    </reference>
</composite>
```



# Java Implementation Example

```
package org.example.services.account;

import org.osoa.sca.annotations.*;

@Service(interfaces = AccountService.class)
public class AccountServiceImpl implements AccountService {

    private String currency = "USD";
    private AccountDataService accountDataService;
    private StockQuoteService stockQuoteService;

    public AccountServiceImpl(
            @Property("currency") String currency,
            @Reference("accountDataService") AccountDataService dataService
            @Reference("stockQuoteService") StockQuoteService stockService) {
       this.currency           = currency;
       this.accountDataService = dataService;
       this.stockQuoteService  = stockService;
    }
```

Annotation for the service offered by this class

Constructor with annotations for injected property and references

# Outline



- SCA in a nutshell
- Motivation & principles for FraSCAti
- Architecture
- Positioning wrt other platforms
- Conclusion

# FraSCAti = SCA++

OW2 Consortium
FraSCAti

## Dynamic deployment & configuration

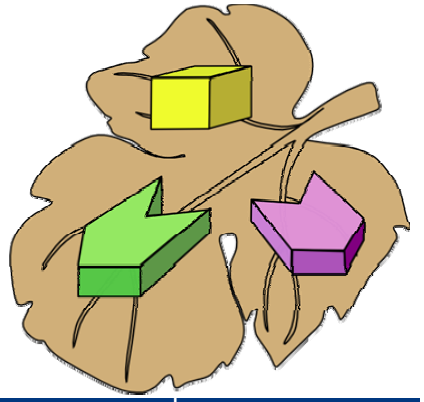
- Distributed deployment with FDF/Deployware

## Runtime adaptation & reconfiguration

- Introspection & reconfiguration support via Fractal
- Reconfiguration of SCA components & FraSCAti itself

## Reflective SCA platform

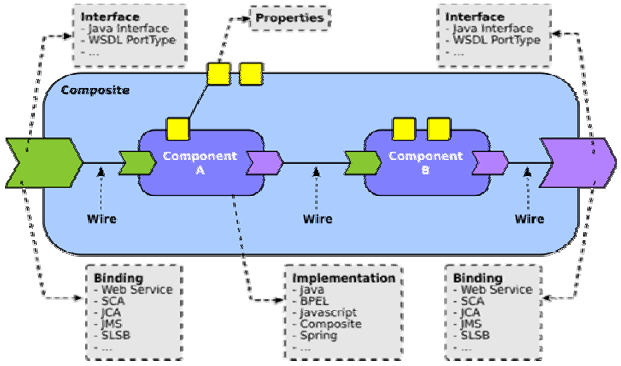
- Lightweight, efficient, predictable, scalable



# FraSCAti = SCA++

OW2 Consortium

FraSCAti



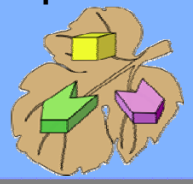

**SCA**

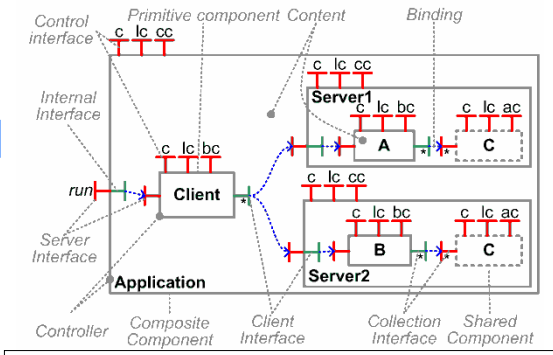
***The standard component model for SOA***

OW2 Consortium

# FraSCAti

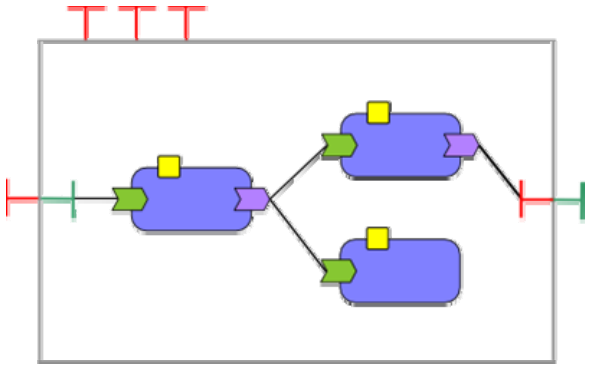
**An open SCA runtime platform built on top of OW2 Fractal**



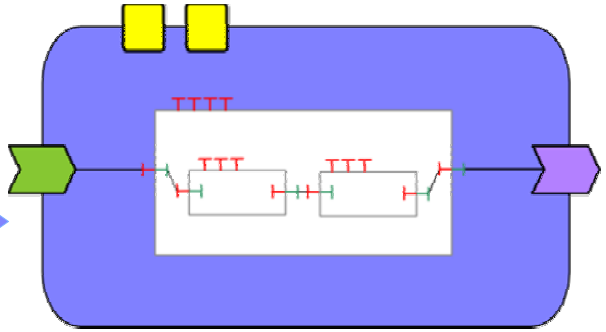



**Fractal**

***A modular and reflective component model***



**Reconfigurable SCA Applications**



**SOA for Fractal**

# FraSCAti Principles

OW2 Consortium
FraSCAti

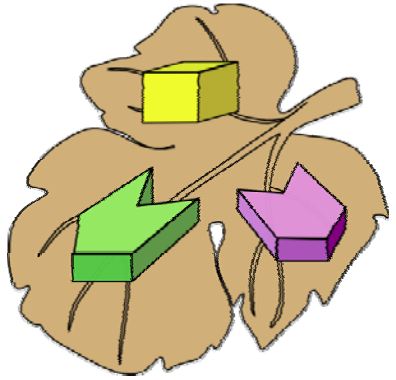
Designed with **adaptability/extensibility/flexibility** in mind

**Component**-based architecture to support *protocols* and *implementations*

- Communication protocols plugged within a binding factory
- Component implementation languages encapsulated as platform components

**AOP**-based mechanism to integrate *intents* and *policies*

- Non-functional services developed as regular SCA components
- Non-functional policies dynamically woven into the base architecture



**Fractal**-based runtime substrate (cf. http://fractal.ow2.org)

- Dynamic reconfiguration capabilities
- Java 5 @-based development style (dependency injection)
- XML-based architecture descriptors
- Structuring concepts (component personality, membrane, control interface, etc.)

2 execution modes for the FraSCAti platform

- Standalone application server (support for 2 backends)
- Integrated in the PEtALS JBI ESB (cf. http://petals.ow2.org)

# FraSCAti Features



## SCA component implementation

- Java POJO and SCA annotations
- Spring
- Fractal

## SCA binding

- Web Services via Apache CXF
- Java RMI

## Under development

- OSGi implementation and binding
- JMS, JSONRPC

# FraSCAti and SCA Spec.

OW2 Consortium
FraSCAti

| SCA Specification | FraSCAti | |
|---|---|---|
| | State | Component |
| SCA Assembly Model (v1.0) | ☺ | Assembly Factory |
| SCA Policy Framework (v1.0) | ☺ / ☹ | Assembly Factory |
| SCA Transaction Policy (v1.0) | ☺ | Transaction Service |
| SCA Java Common Annotations & APIs (v1.0) | ☺ | Tinfi |
| SCA Java Component Implementation (v1.0) | ☺ | Tinfi |
| SCA Web Services Binding (v1.0) | ☺ | Binding Factory |

☺ = supported

☺ / ☹ = under development

# FraSCAti and SCA Spec.

OW2 Consortium
FraSCAti

| SCA Specification | FraSCAti | |
|---|---|---|
| | State | Components |
| SCA Spring Component Implementation (v1.0) | ☺ / ☹ | Plug-in Assembly Factory |
| SCA BPEL Client & Implementation (v1.0) | ☹ ☹ | Plug-in Assembly Factory |
| SCA C++ Client & Implementation (v1.0) | ☹ ☹ ☹ | |
| SCA C Client & Implementation (v1.0) | ☹ ☹ ☹ | |
| SCA COBOL Client & Implementation (v1.0) | ☹ ☹ ☹ | |
| SCA JMS Binding (v1.0) | ☹ ☹ | Plug-in Binding Factory |
| SCA EJB Session Bean Binding (v1.0) | ☹ ☹ | Plug-in Binding Factory |
| SCA JCA Binding (v1.0) | ☹ ☹ ☹ | Plug-in Binding Factory |
| SCA Java EE Integration (v0.9) | ☹ ☹ ☹ | |

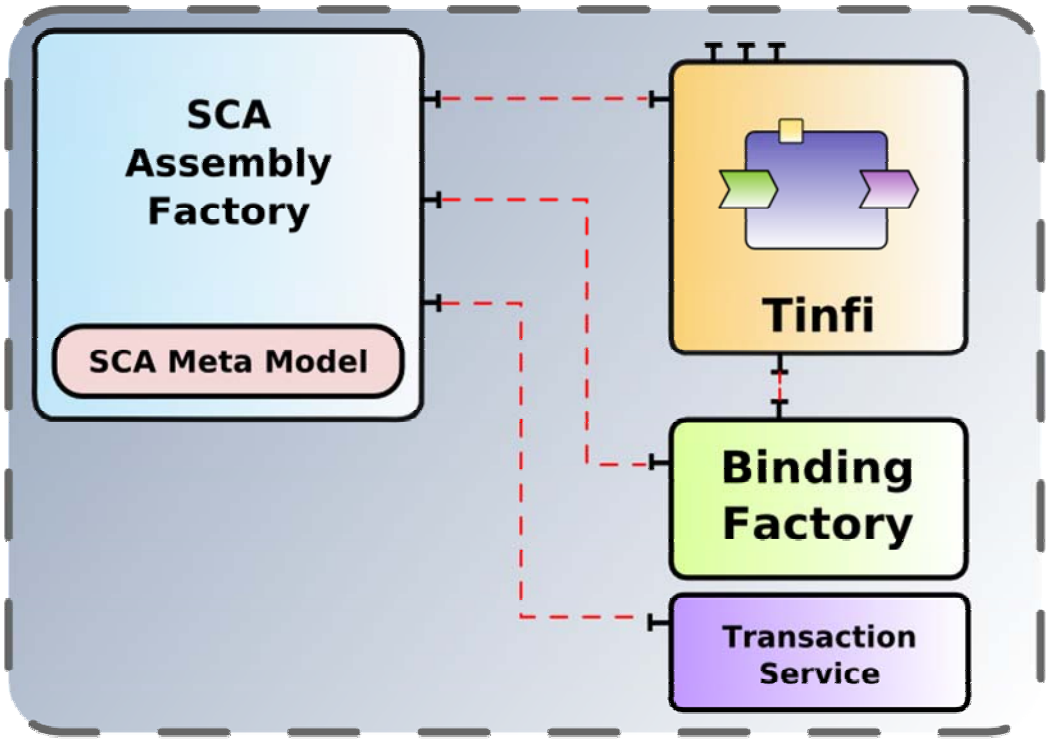
# Outline

OW2 Consortium
FraSCAti

- SCA in a nutshell
- Motivation & principles for FraSCAti
- Architecture
- Positioning wrt other platforms
- Conclusion

# FraSCAti Architecture

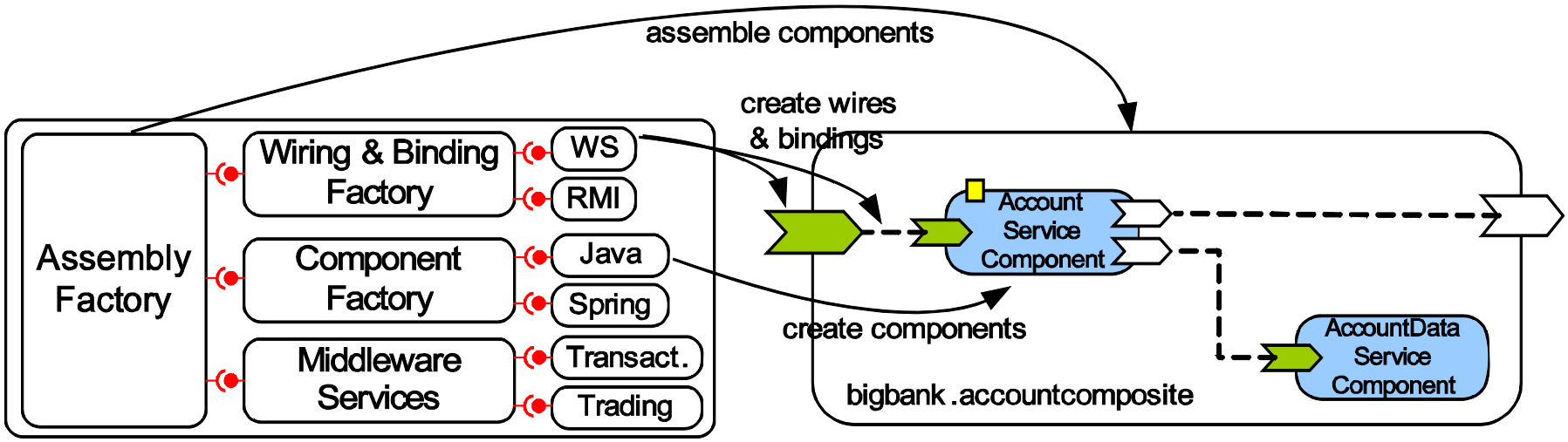





- **Tinfi** generates the SCA components glue code and create component instances
- **Binding Factory** imports & exports the SCA components via specific communication protocols
- **Transaction** controls local & distributed transactions between the SCA components
- **Assembly Factory** processes and deploys SCA assembly models

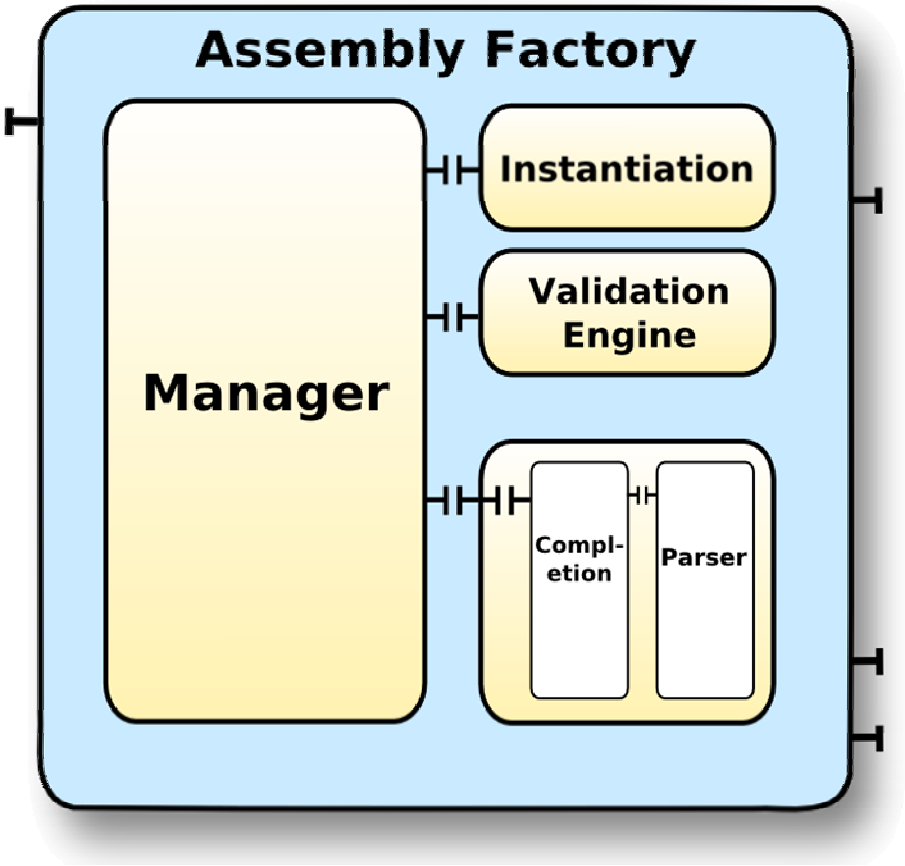
# FraSCAti Architecture





# FraSCAti Assembly Factory



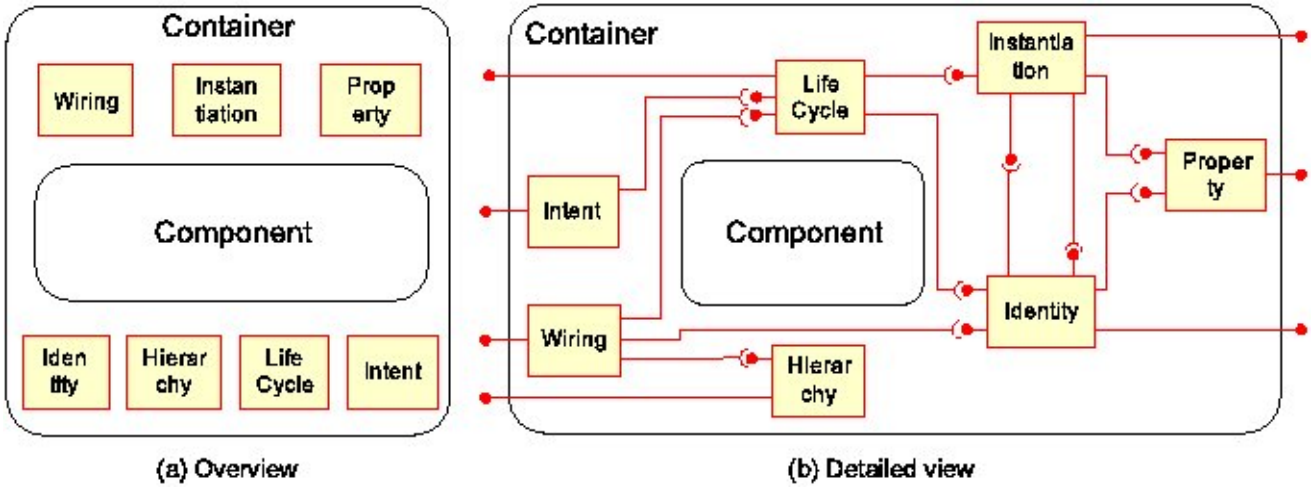



- **Manager** loads resources and invokes sub components
- **Parser** creates a model instance from composite definition and implementation.
  *Use Eclipse STP SCA model*
- **Validation Engine** validates additional model constraints.
  *Implementation in progress*
- **Instantiation** creates new component instances.
  *Use Tinfi & Binding Factory*

# FraSCAti Tinfi

OW2 Consortium
FraSCAti

- Runtime kernel for SCA component
  - Implements the SCA component runtime semantics
  - Implements the SCA Java Component API
- Provide a container for hosting/managing SCA components



# FraSCAti Tinfi

OW2 Consortium
FraSCAti

Fractal architecturing principles for the implementation of the container (control membranes)

6 controllers

- **SCAComponent** : component identity dedicated interface (ComponentContext) and implementation
- **SCAContentController** : component instantiation policy
  dedicated implementation, private interface (no need to export it)
- **SCAIntentController** : intent handlers management
- **SCAPropertyController** : component properties management
- **SCALifeCycleController** : component initialization (@EagerInit)
  same interface as Fractal LC, dedicated implementation
- **SCABindingController** : component bindings
  same interface as Fractal BC, dedicated implementation

Interceptors

- lifecycle management
- component instantiation policy
- intent dispatch

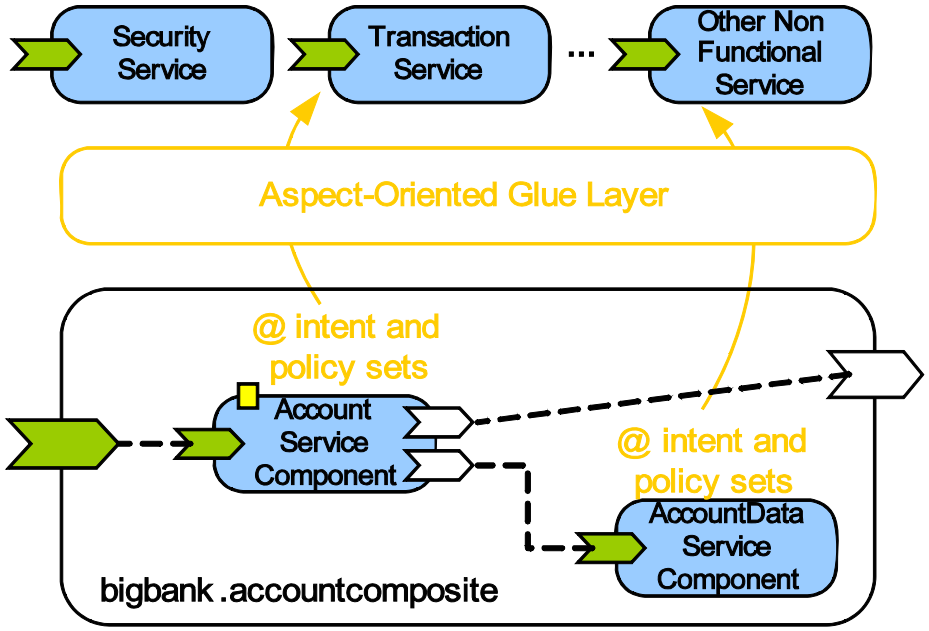
# FraSCAti Tinfi

OW2 Consortium
FraSCAti

## Aspect-Oriented Glue Layer

- AOP [Kiczales97]
- AspectJ, JBoss AOP, …

- non functional services implemented as SCA components
- glued to the application based on the intent and policy sets annotations

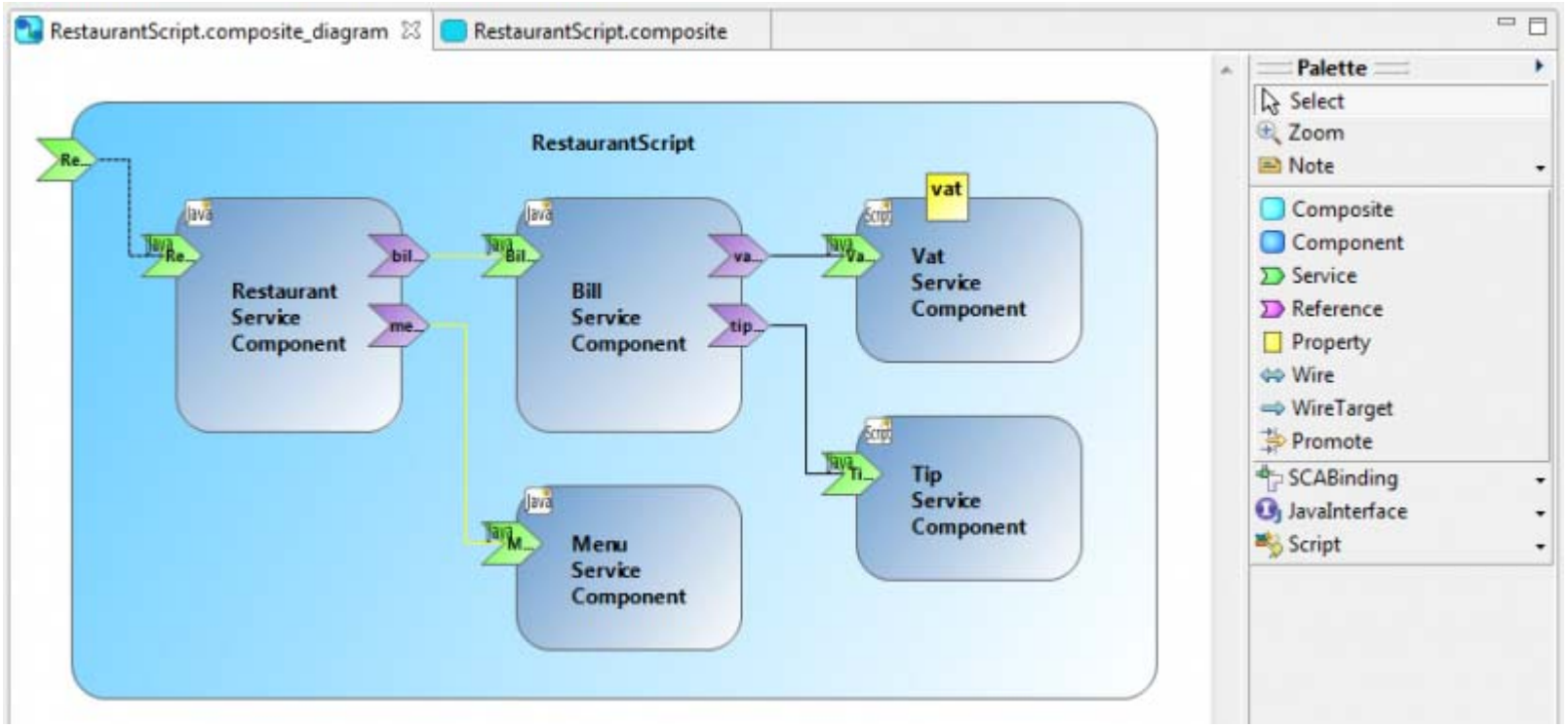


# FraSCAti Tooling

OW2 Consortium
FraSCAti

## Eclipse STP/SCA modeler

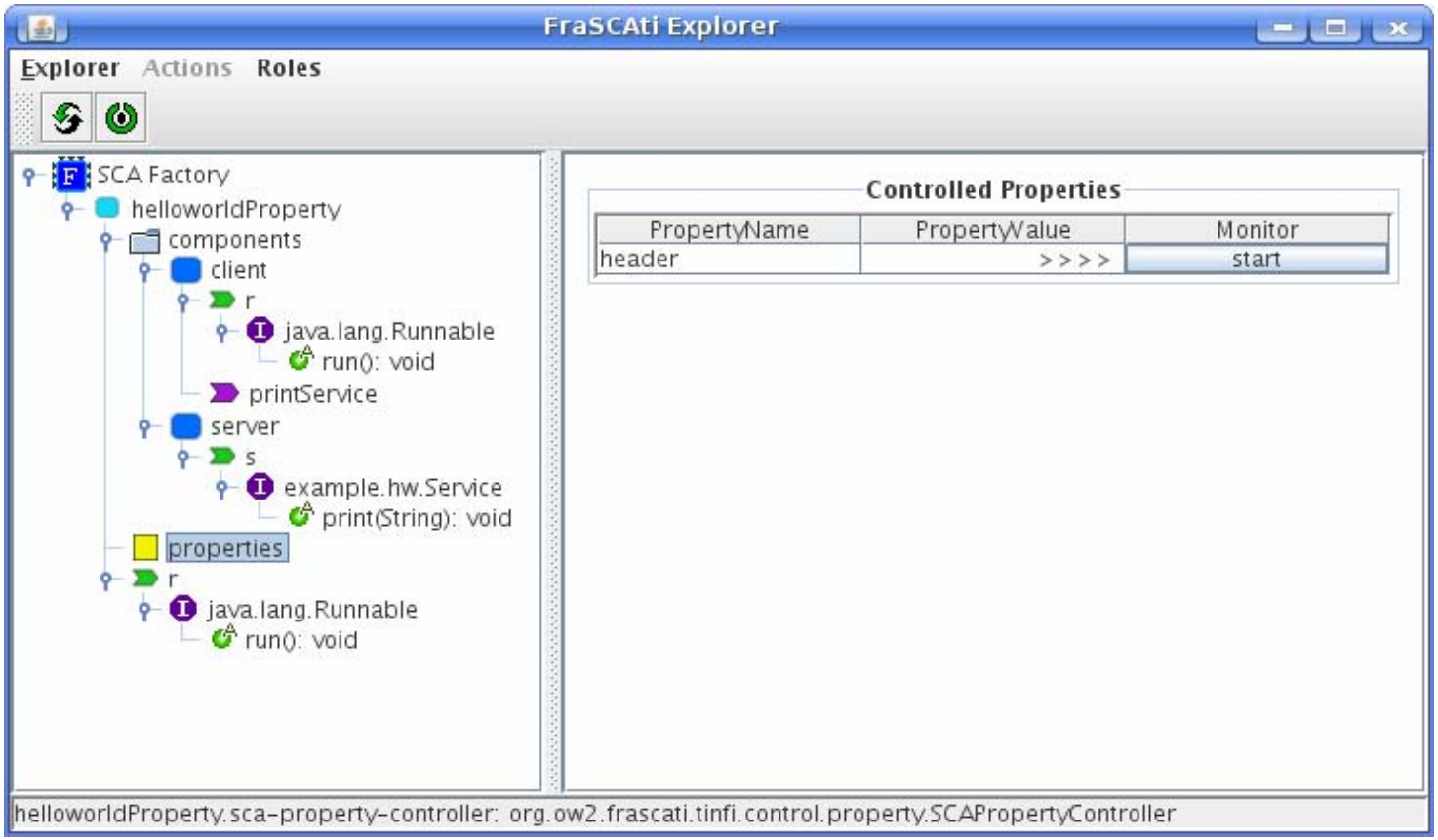
- http://www.eclipse.org/stp/sca/



# FraSCAti Tooling



## FraSCAti Explorer



# Outline



- SCA in a nutshell
- Motivation & principles for FraSCAti
- Architecture
- Positioning wrt other platforms
- Conclusion

# FraSCAti Vs. Apache Tuscany

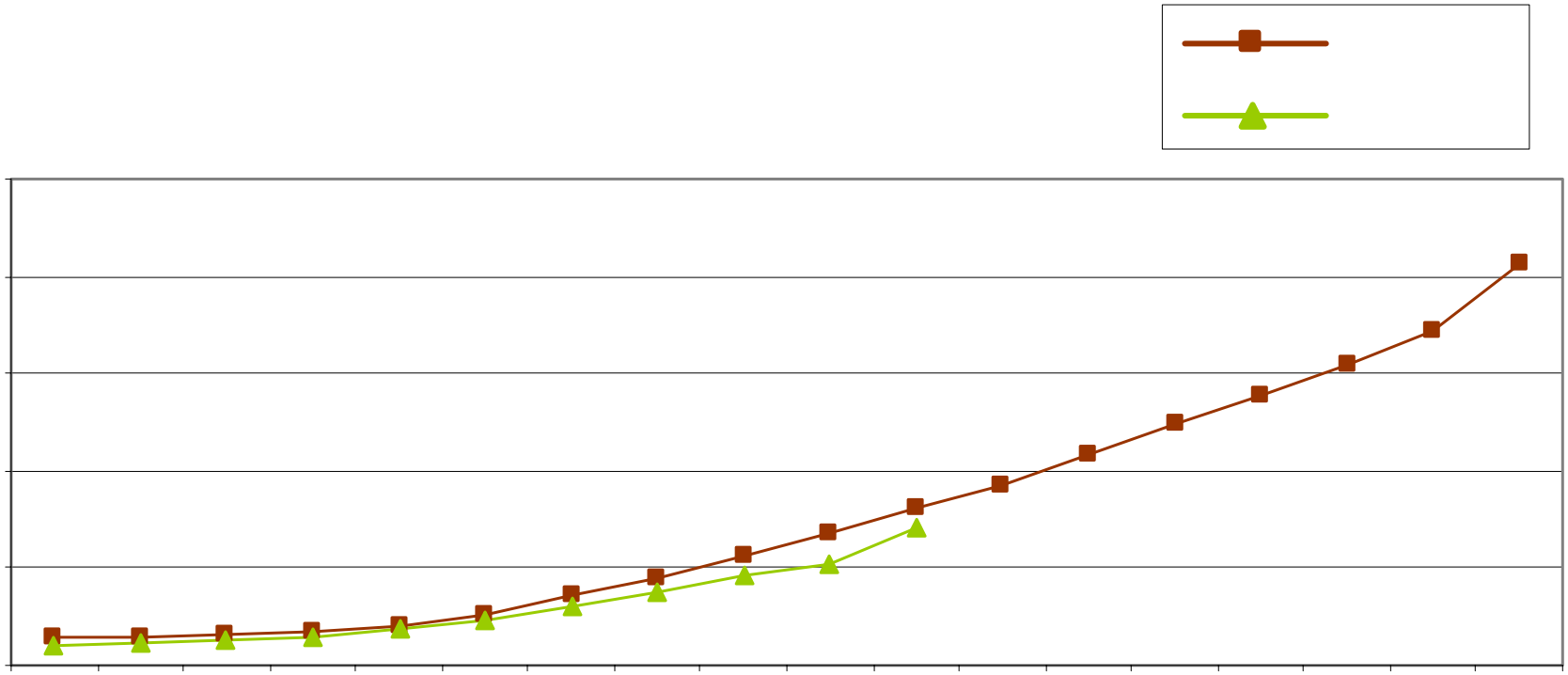
OW2 Consortium
FraSCAti

- ☹ *Less SCA features* supported
  - Less implementation languages and binding protocols
- ☹ Smaller ecosystem
  - Less sponsoring companies, developers, and users
- ☺ **Better continuum** from SCA tooling to runtime platform
  - Share the same SCA metamodel with Eclipse STP SCA project
- ☺ **Better footprint** to target embedded systems
  - Smaller disk and memory footprints
- ☺ Ready for **dynamic runtime reconfiguration**
  - Based on OW2 Fractal component model and associated tools

# FraSCAti Vs. Apache Tuscany

OW2 Consortium
FraSCAti

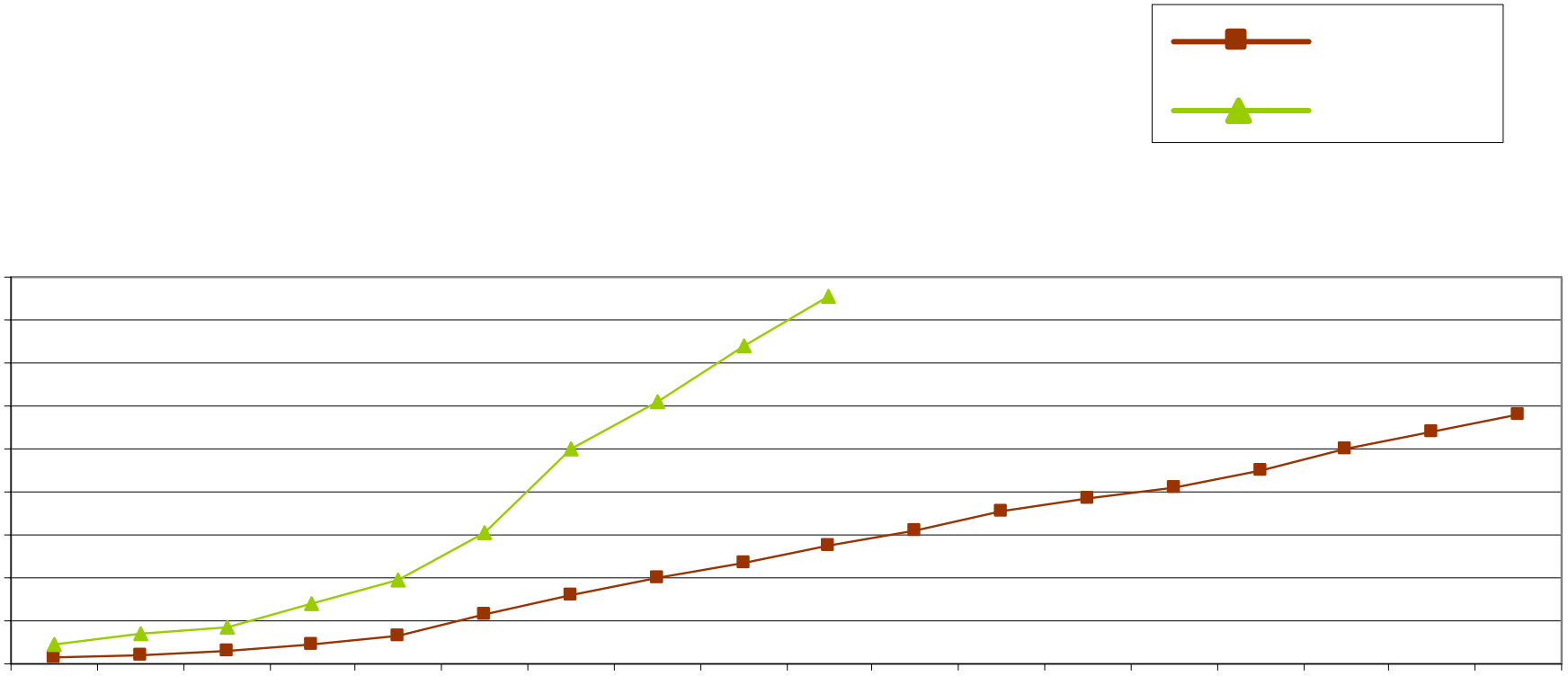
## Performance Evaluation: Instantiation Time



# FraSCAti Vs. Apache Tuscany

OW2 Consortium
FraSCAti

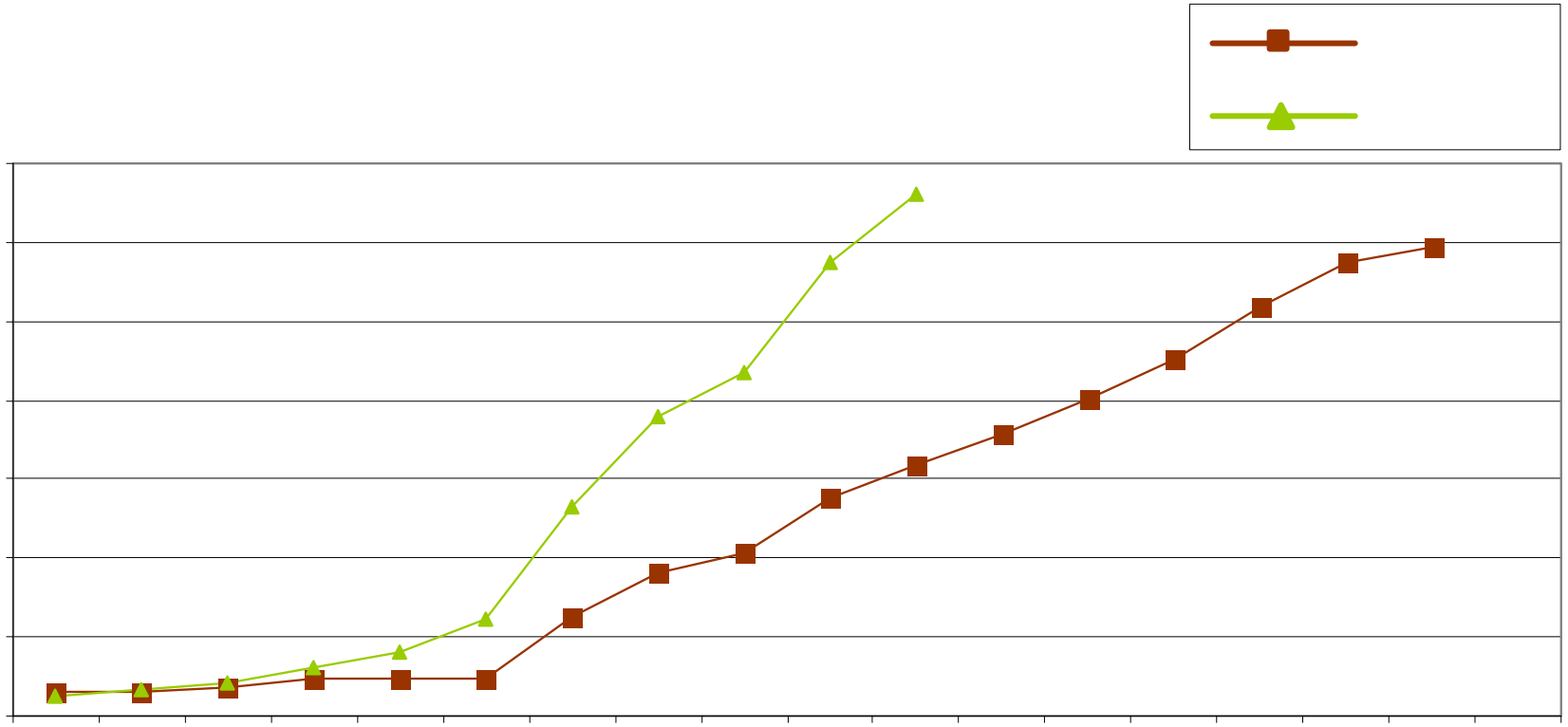
## Performance Evaluation: Invocation Time



# FraSCAti Vs. Apache Tuscany

OW2 Consortium
FraSCAti

## Performance Evaluation: Memory Consumption



# Other OSS Competitors

OW2 Consortium
FraSCAti

- Fabric3 
  - ☹/☺ Fork from the Apache Tuscany project
  - ☹ Developed by fewer contributors
- The Newton Project 
  - ☺ Distributed runtime framework based on OSGi, Jini, and SCA
  - ☺ SCA bindings for OSGi and Jini
  - ☹ Does not target a fully-compliant SCA framework
    - ☹ No support for SCA Java annotations
    - ☹ No Web Service binding
- The Mule Project - MuleSCA activity 
  - ☺ Some Web pages
  - ☹ No open source code currently available

# Outline

OW2 Consortium
FraSCAti

- SCA in a nutshell
- Motivation & principles for FraSCAti
- Architecture
- Positioning wrt other platforms
- Conclusion

# Conclusion



## FraSCAti

- an open and extensible implementation of the SCA specifications
  - continuum from tooling to runtime (common SCA metamodel shared with STP)
  - reconfigurable SCA applications
  - lightweight version for embedded devices currently being developed
- based on OW2 code blocks
- developed by the ANR SCOrWare project (ending 04/2009)

# Perspectives

OW2 Consortium

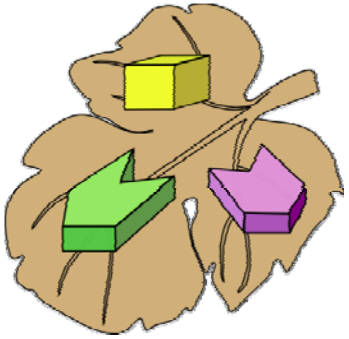
FraSCAti

- INRIA ADT galaxy – Agile SOA Platform
  - SCA management at runtime
    - FraSCAti Explorer
    - Eclipse STP/SCA Composite Designer
  - SCA scripting with FScript
  - SCA monitoring with WildCat
  - SCA BPEL Client and Implementation (v1.0)
- CAPPUCINO – eCommerce
  - FraSCAti in mobile devices (PDA & smart phones)
- ANR ITEmIS – Marriage of IT and embedded systems
  - PEtALS/FraSCAti & OSGi
  - FraSCAti & JMS/JORAM
  - Formal specification of SCA
- IST SOA4All – A Web of billions of services
  - Large-scale PEtALS/FraSCAti deployment
  - SCA binding for SOA4All Semantic Spaces



# Contact

FraSCAti



## Web site

- http://frascati.ow2.org

## Project heads


- Philippe Merle: Philippe.Merle@inria.fr
- Lionel Seinturier: Lionel.Seinturier@univ-lille1.fr


## Development team

- INRIA ADAM & SARDES
  - Pierre Carton, Christophe Demarey, Nicolas Dolet, Damien Fournier, Philippe Merle, Nicolas Pessemier, Valerio Schiavoni, Lionel Seinturier

## Acknowledgements


- Vivien Quéma, Jean-Bernard Stefani, Alain Boulze, Adrian Mos, Adrien Louis, Stéphane Bagnier, Daniel Hagimont, Etienne Juliot, Gaël Blondelle, Jean-Pierre Lorre, Marc Dutoo, Marc Pantel, Mickael Istria, Mohammed Eljai, Nicolas Salatge, Samir Tata, Roland Naudin, Samuel Quaireau, Stéphane Drapeau, Thomas Darbois
- and all ANR SCOrWare project members (past and present) that I may have forgotten…

# SCA References

OW2 Consortium

FraSCAti

## SCA Specifications

- OpenSOA http://www.osoa.org
- OASIS OpenSCA http://www.oasis-opencsa.org

## OSS Implementations

- Tuscany http://tuscany.apache.org
- Newton http://newton.codecauldron.org/site/index.html
- Fabric3 http://xircles.codehaus.org/projects/fabric3
- FraSCAti http://www.scorware.org

## SCA Resources

- http://www.osoa.org/display/Main/SCA+Resources
- http://www-128.ibm.com/developerworks/library/specification/ws-sca
- http://www.davidchappell.com/articles/Introducing_SCA.pdf
- http://www-128.ibm.com/developerworks/websphere/techjournal/0510_brent/0509_brent.html
- http://events.oasis-open.org/home/sites/events.oasis-open.org.home/files/Flexible_Agile_Composition_01.ppt [Mike Edwards]
- http://www.osoa.org/download/attachments/250/Power_Combination_SCA_Spring_OSGi.pdf?version=3