



Leading Open Source SOA

**ebnm**

WEBSOURCING

## ***SCA support in PEtALS with Tinfu / Frascati***

***by Vincent ZURCZAK & Mohammed EL JAI – May 15<sup>th</sup> 2008***



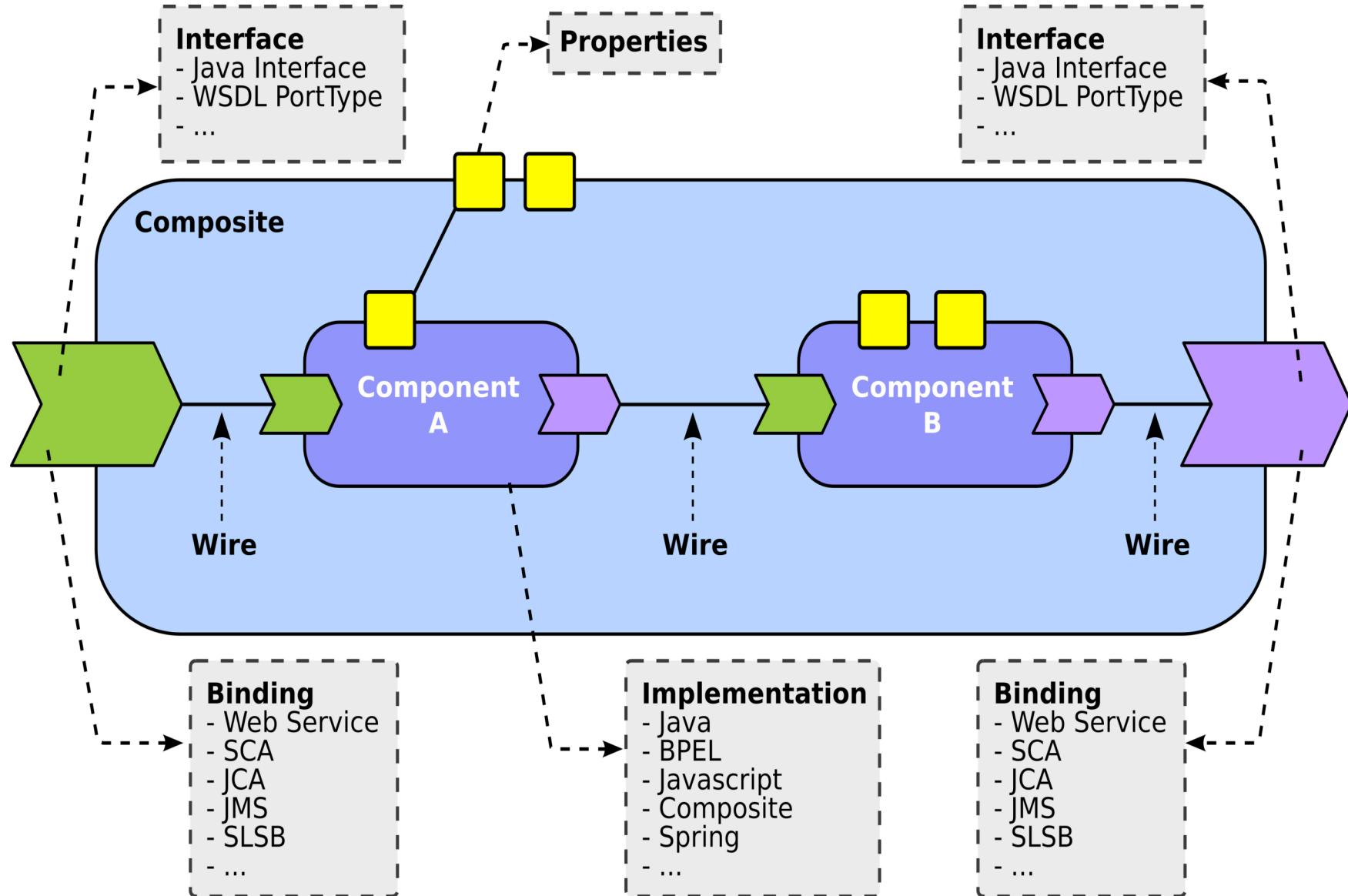
- Reminder about SCA
- SCA vs. JBI ?
- Overview of PEtALS and Tinfì / Frascati
- Architecture
- Benefits of using PEtALS for SCA
- PEtALS – Frascati within SCA World



- Reminder about SCA
- SCA vs. JBI ?
- Overview of PEtALS and Tinfì / Fracati
- Architecture
- Benefits of using PEtALS for SCA
- PEtALS – Fracati within SCA World



- SCA = Service Component Architecture
- Mix between Service and Component paradigms
- Main advantages
  - Several implementation languages (Java, C++, Cobol, PHP...)
  - Several communication protocols (WS, RMI, JMS...)
  - Seen like a powerful way to turn existing code into services
- Specification defined by OSOA Consortium
  - Supported by IBM, BEA, IONA, Oracle, SAP...
- Standardization by OASIS in progress
  - First version expected in December 2008





- Reminder about SCA
- [SCA vs. JBI ?](#)
- Overview of PEtALS and Tinfì / Frascati
- Architecture
- Benefits of using PEtALS for SCA
- PEtALS – Frascati within SCA World



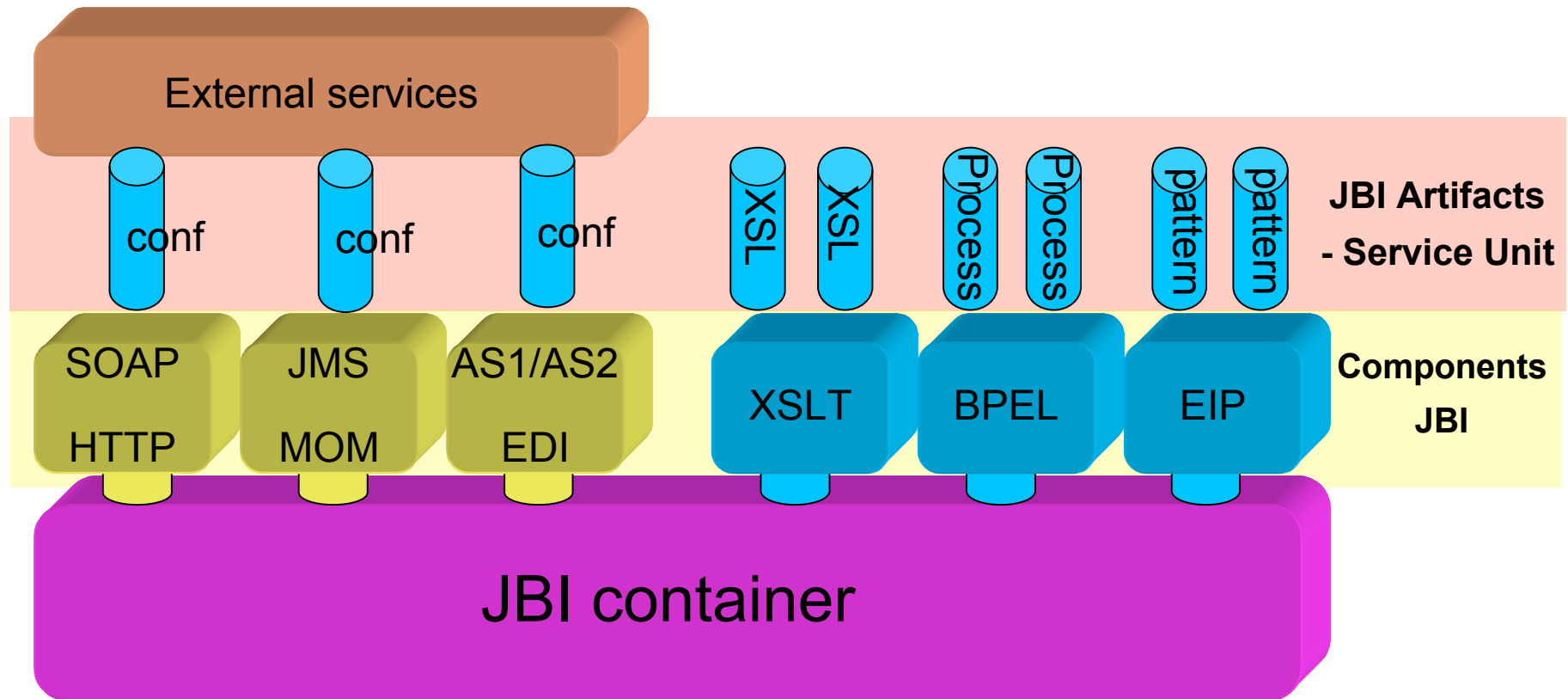
- Many questions about their relation
- Both are focused on Service Oriented Architecture
- JBI is used to implement integration platforms
  - The only standard to focus on loosely coupling
  - Binding components to communicate with external world
  - Service engines to propose technical features
    - Orchestration, transformation...
  - Used to integrate applications in an SOA
- SCA is used to build composite applications
  - Can use JBI bindings
  - Can compose services from JBI



- Reminder about SCA
- SCA vs. JBI ?
- [Overview of PEtALS and Tinfì / Frascati](#)
- Architecture
- Benefits of using PEtALS for SCA
- PEtALS – Frascati within SCA World



- JBI architecture overview



 **Service Engines:** provide transformation, orchestration and business services

 **Binding Components:** connection to external resources





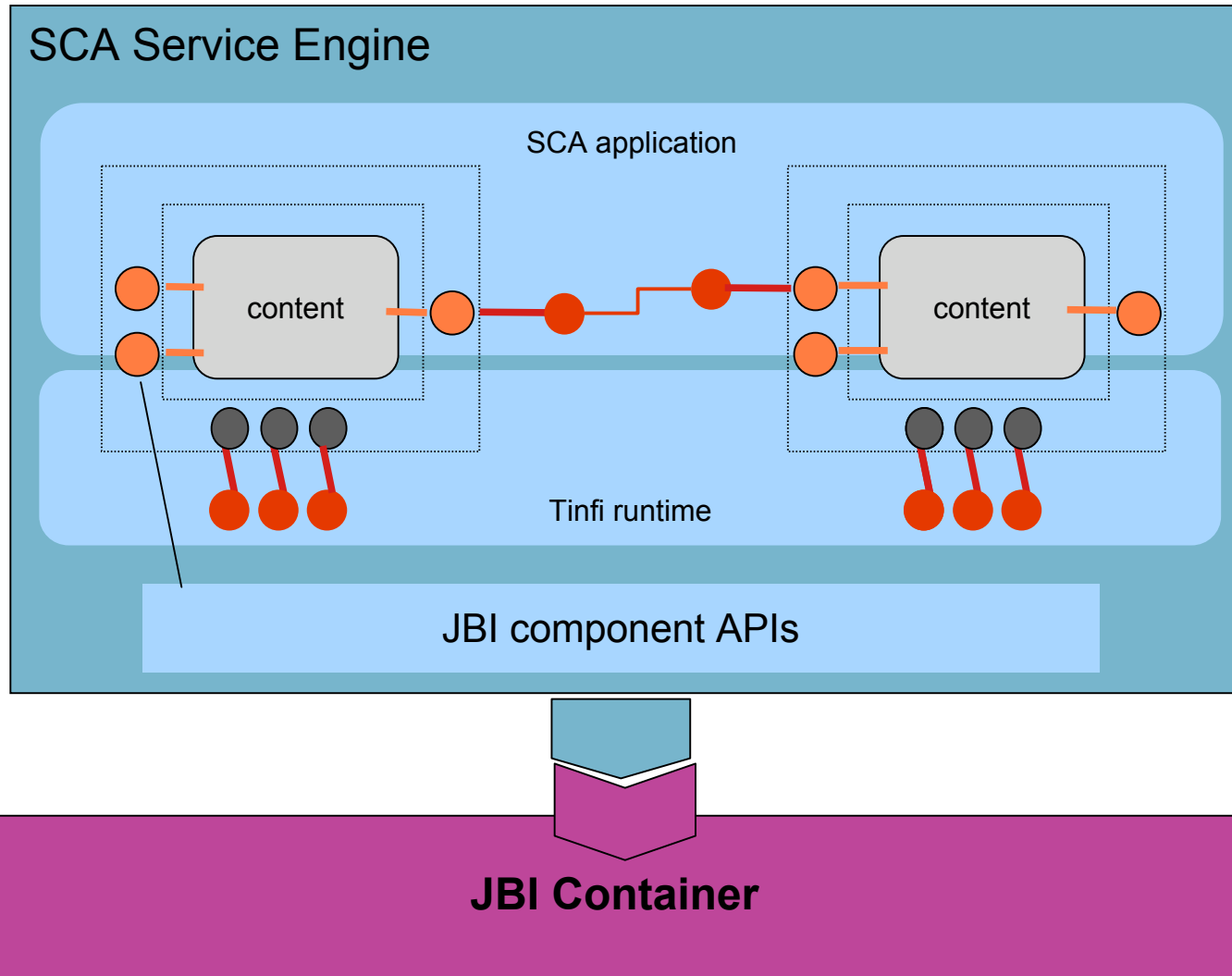
- Tinfi is a Runtime Kernel for SCA Components
- Developed within ANR project SCOrWare
  - Leded by L. Seinturier (INRIA / ADAM team)
- Built above Fractal
  - Fractal Component Model extension
  - Reuses the standard Fractal controllers and Juliac
  - Brings new features for SCA components
    - Dynamic reconfiguration and introspection
  - A Tinfi component is a fractal component AND a SCA component
- Frascati
  - SCA runtime
  - Embeds Tinfi and other elements (assembly and bindings factories)...



- Reminder about SCA
- SCA vs. JBI ?
- Overview of PEtALS and Tinfu
- Architecture
- Benefits of using PEtALS for SCA
- PEtALS – Frascati within SCA World

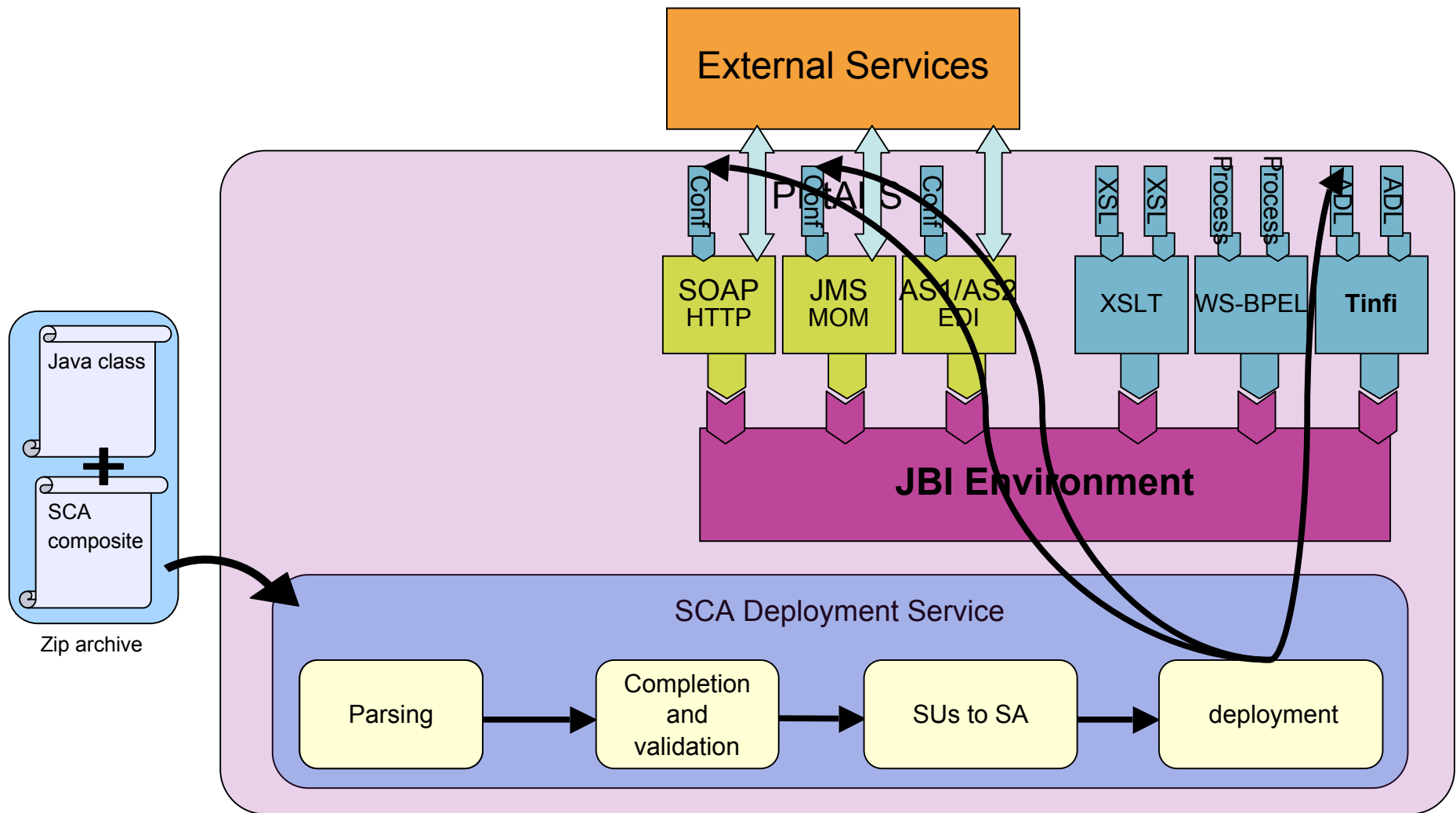


- SCA support in PEtALS: integrate Tinfi as a service engine
- Input
  - SCA description file ( \*.composite )
  - Sources or binaries of the composite implementations
  - JBI description file ( jbi.xml )
- Deployment Time
  - Generation of a Tinfi composite using Juliac
  - Exposition of the composite services as PEtALS services
- Invocation Time
  - Transform JBI messages into Java method calls and vice-versa
  - "Marshalling" and "Unmarshalling" of JBI messages using JAXB





- SCA composite parsing
- Complete the parsed SCA model with implementations data
  - Supported implementations are Java ones
  - The data are in fact Java annotations present into the code
- SCA model validation ( EMF meta-model... )
- Map the SCA model to a Tinfi model
- Generate artifacts...
  - ... to export services using PEtALS BC
  - ... to import references and expose them as PEtALS services





- Reminder about SCA
- SCA vs. JBI ?
- Overview of PEtALS and Tinfu
- Architecture
- [Benefits of using PEtALS for SCA](#)
- PEtALS – Frascati within SCA World



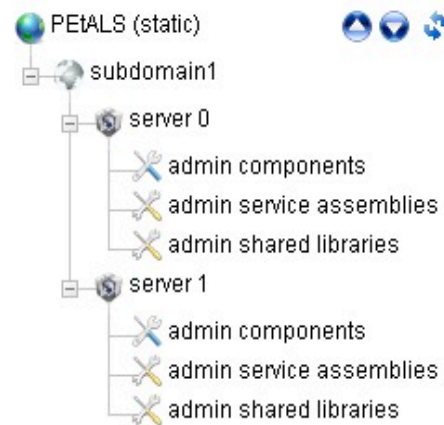
- Existing bindings in PEtALS
  - Currently, composites deployed in the SCA service engine use PEtALS' BCs
  - Consequence:
    - More BCs in PEtALS than bindings in Frascati
    - Dynamic reconfiguration of composite services (no need to update bindings)
    - Add a new binding without modifying the composite
    - Just update the required SUs
  
- PEtALS tools can be used with SCA composites
  - JMX console
  - Administration Console



The open source ESB solution for Services Oriented Architectures



[PETALS Admin](#) | [jUDDI Admin](#) | [Monitoring](#) logged in as **petals** (admin) | [log out](#)



## Component Administration For Server 0

Upload & install a new component

Server has no deployed component yet



The open source ESB solution for Services Oriented Architectures

OW2 Consortium

PETALS Admin | JUDDI Admin | **Monitoring** logged in as **petals** (admin) | [log out](#)

PETALS (static)

- subdomain1
  - server 0
    - binding components
    - service engines
      - petals-sample-client
    - system Info
  - server 1
    - binding components
    - service engines
      - petals-se-random
        - MCEndpoint
        - randomEndpoint
        - testEndpoint
      - petals-sample-clock
        - ClockEndpoint
    - system Info
  - message Exchanges
  - endpoints

## Subdomain1 30 s

Subdomain's statistics

### Messages Per Second

### Messages Repartition

Server	Count	Percentage
Server 1	83	50%
Server 0	83	50%



- « requires » attribute determines PEtALS behaviour at runtime
- Example with already implemented policies in PEtALS

```
<service name="TransformHugeFile"
  requires="reliability.transport">

  <interface.java interface="..."/>
  <operation name="getASummary"
    requires="performance.compression" />
  <binding.ws/>
</service>
```



- Reminder about SCA
- SCA vs. JBI ?
- Overview of PEtALS and Tinfu
- Architecture
- Benefits of using PEtALS for SCA
- [PEtALS – Frascati within SCA World](#)



- EBM WebSourcing is involved in the SCOrWare project
  - Provides an open-source implantation of the SCA specifications
  - Defines a run-time, Frascati, which embeds Tinfi
  
  - PEtALS integrates Frascati components
    - Tinfi and some elements from the assembly-factory
  - SCA tooling contribution
    - Eclipse STP and Eclipse JWT projects (mainly)
  - Relations with industrial companies as SCA users
  
- Reference Open-Source Implementation of SCA = Apache Tuscany
- Links with this project are made through the tooling
- Provide common tooling allows to have a better communication about SCA
- PEtALS will be seen and supported as a possible runtime in the SCA tooling



- OSOA: <http://www.osoa.org/display/Main/Home>
- PEtALS website: <http://petals.objectweb.org/>
- SCOrWare project: <http://www.scorware.org/>
- Apache Tuscany project: <http://incubator.apache.org/tuscany/>
- SCA component of Eclipse STP: <http://www.eclipse.org/stp/sca/index.php>
- Eclipse STP wiki about SCA: [http://wiki.eclipse.org/STP/SCA\\_Component](http://wiki.eclipse.org/STP/SCA_Component)



# Questions

## Contacts:

[mohammed.eljai@ebmwebsourcing.com](mailto:mohammed.eljai@ebmwebsourcing.com)

[vincent.zurczak@ebmwebsourcing.com](mailto:vincent.zurczak@ebmwebsourcing.com)