LSR-Adèle
Université Grenoble I, France

A Framework for Constructing Adaptive Component-Based Applications: Concepts and Experiences

Humberto Cervantes
Richard S. Hall

CBSE ,May 24th, 2004

The Gravity Framework

What is Gravity?

- A framework to construct non-distributed component-based applications that can adapt to dynamic availability

Dynamic availability

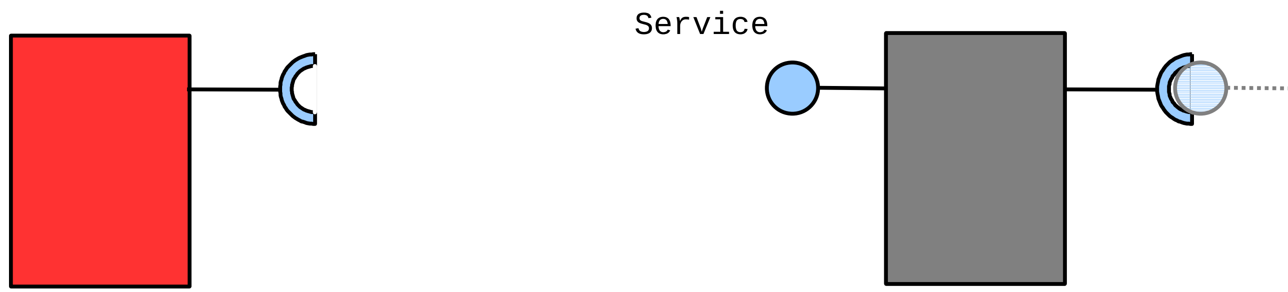
- The situation where functionality that can be used or being used by an application becomes available or unavailable during execution, potentially outside of its control
  - Not supported explicitly in component models

Pillars

- Service-oriented component model
  - Introduces concepts from service orientation into a component model
- Execution environment that provides adaptation logic

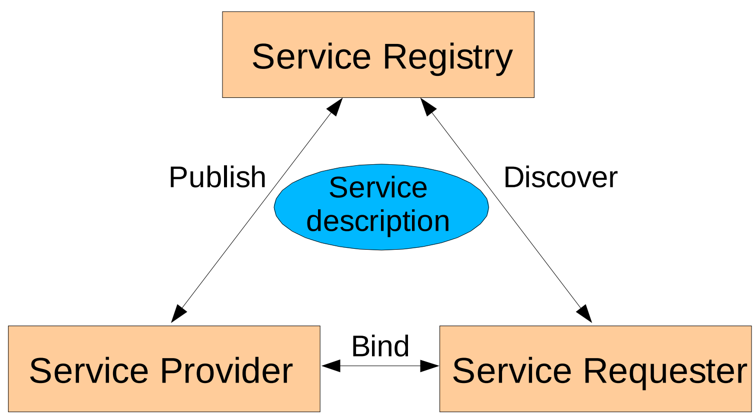
Service-Oriented Component Model

- Component instances provide and require services

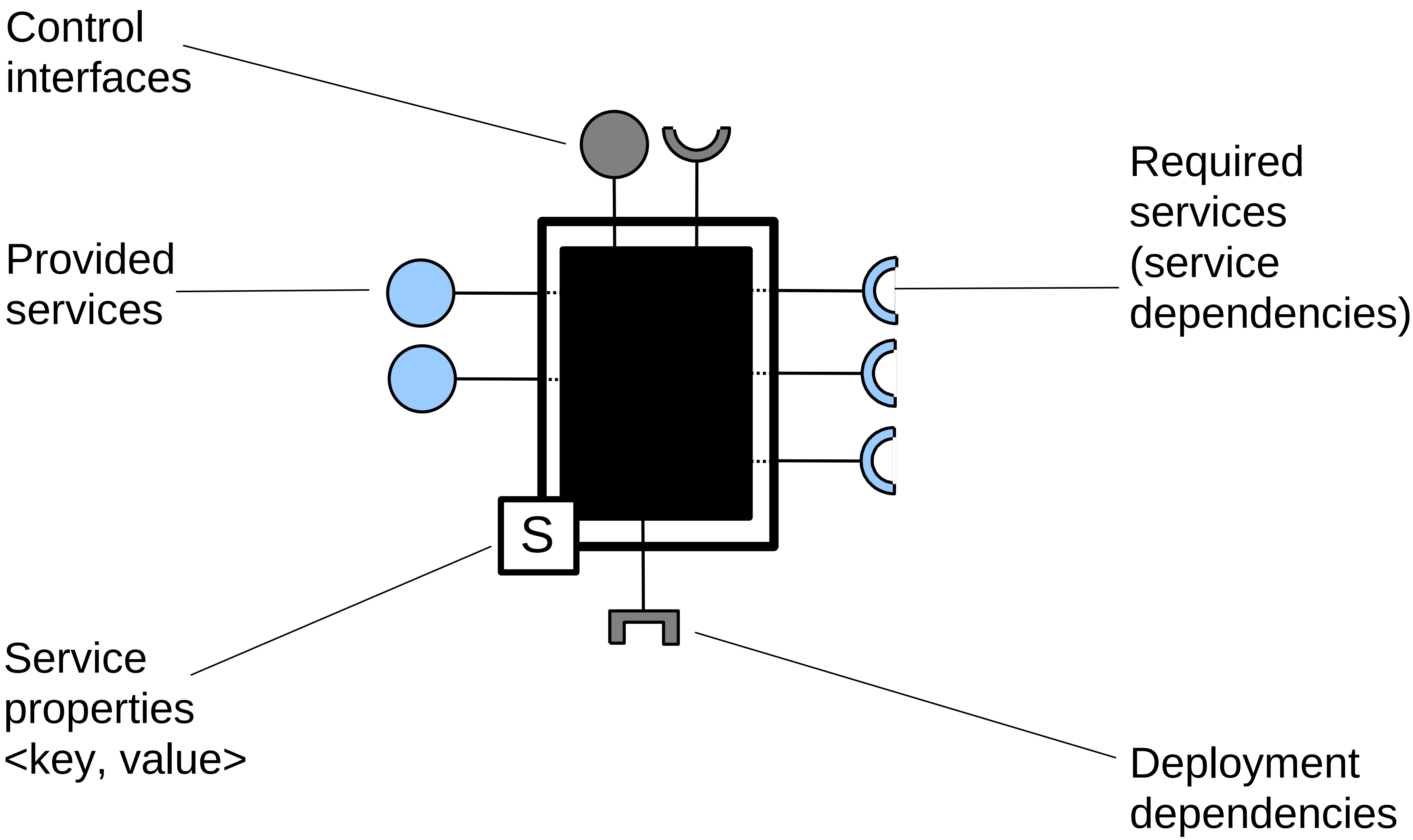


invalid **instance** ***valid*** **instance**

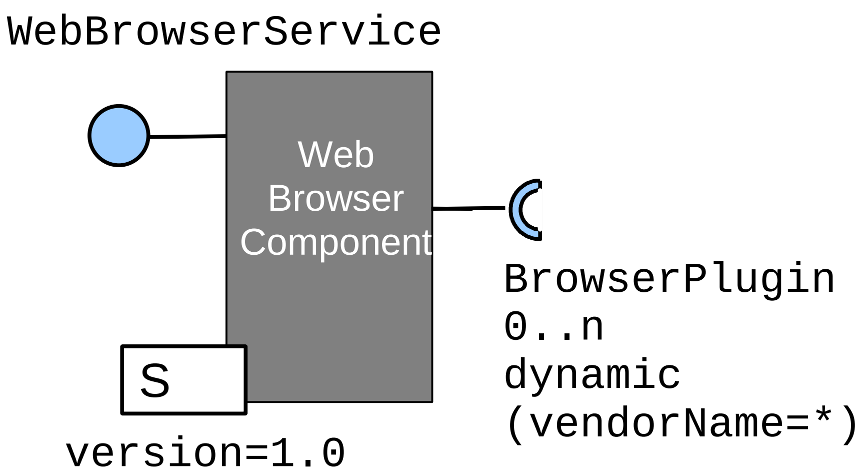
- Instances binding achieved through the *service oriented interaction pattern* (trading)



Service Component



Service Dependency Properties



Name

- Java interface

Cardinality

- Optional, mandatory
  - `0..*`, `1..*`
- Singular, aggregate
  - `*..1`, `*..n`

Policy

- Static, dynamic

Filter

- Boolean expression consisting of service property values
  - Limit ambiguity

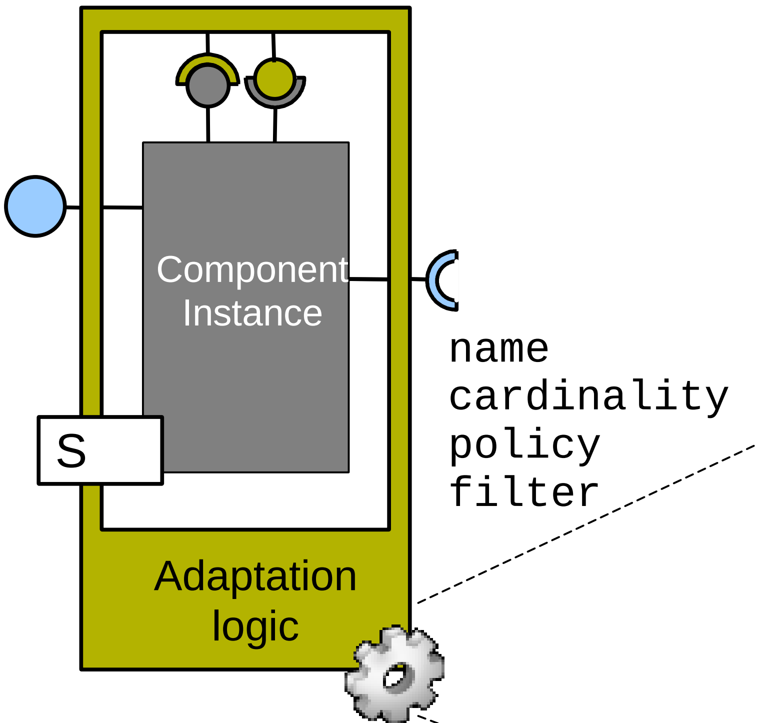
Component Descriptor

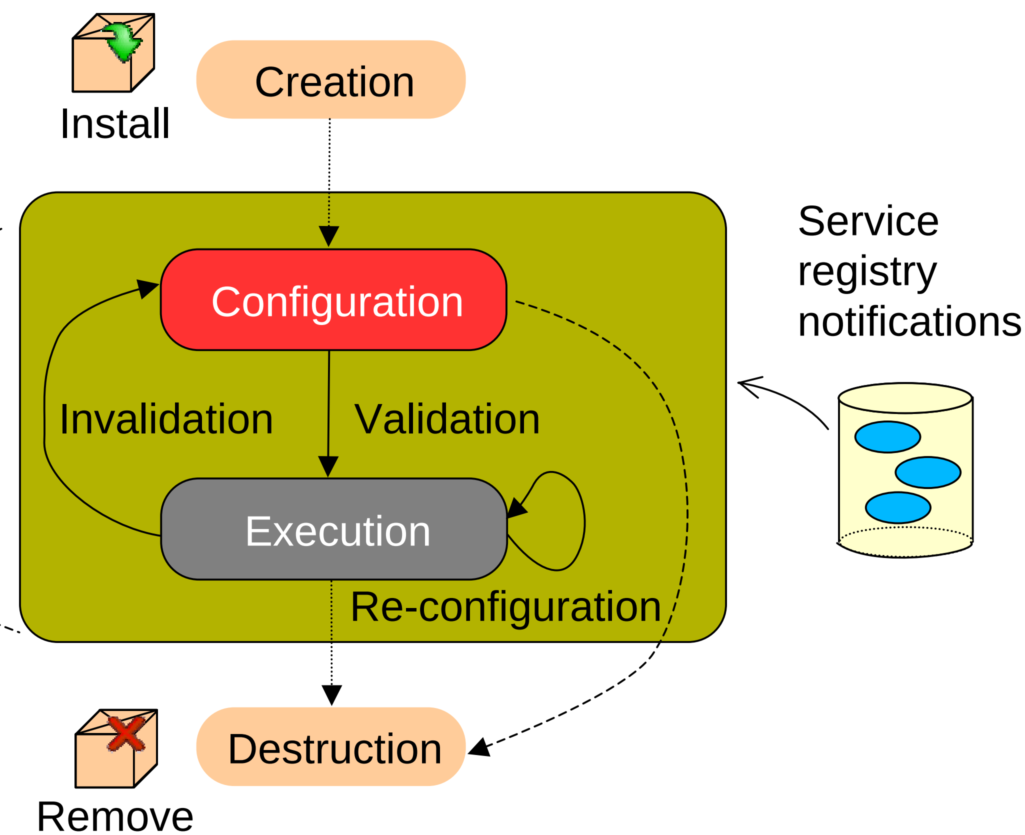
```
<component class="org.gravity.webbrowser.BrowserImpl">

   <provides service="org.gravity.services.WebBrowser"/>

   <property name="version" value="1.0" type="string"/>

   <requires
       service="org.gravity.services.BrowserPlugin"
       cardinality="0..n"
       policy="dynamic"
       filter="(vendorName=*)"
       bind-method="addPlugin"
       unbind-method="removePlugin"
     />

</component>
```

Instance Manager

Responsibilities

- Dependency management
- Service publication
- Maintain Validity





Applications

Built as

- A set of connected component instances at run-time

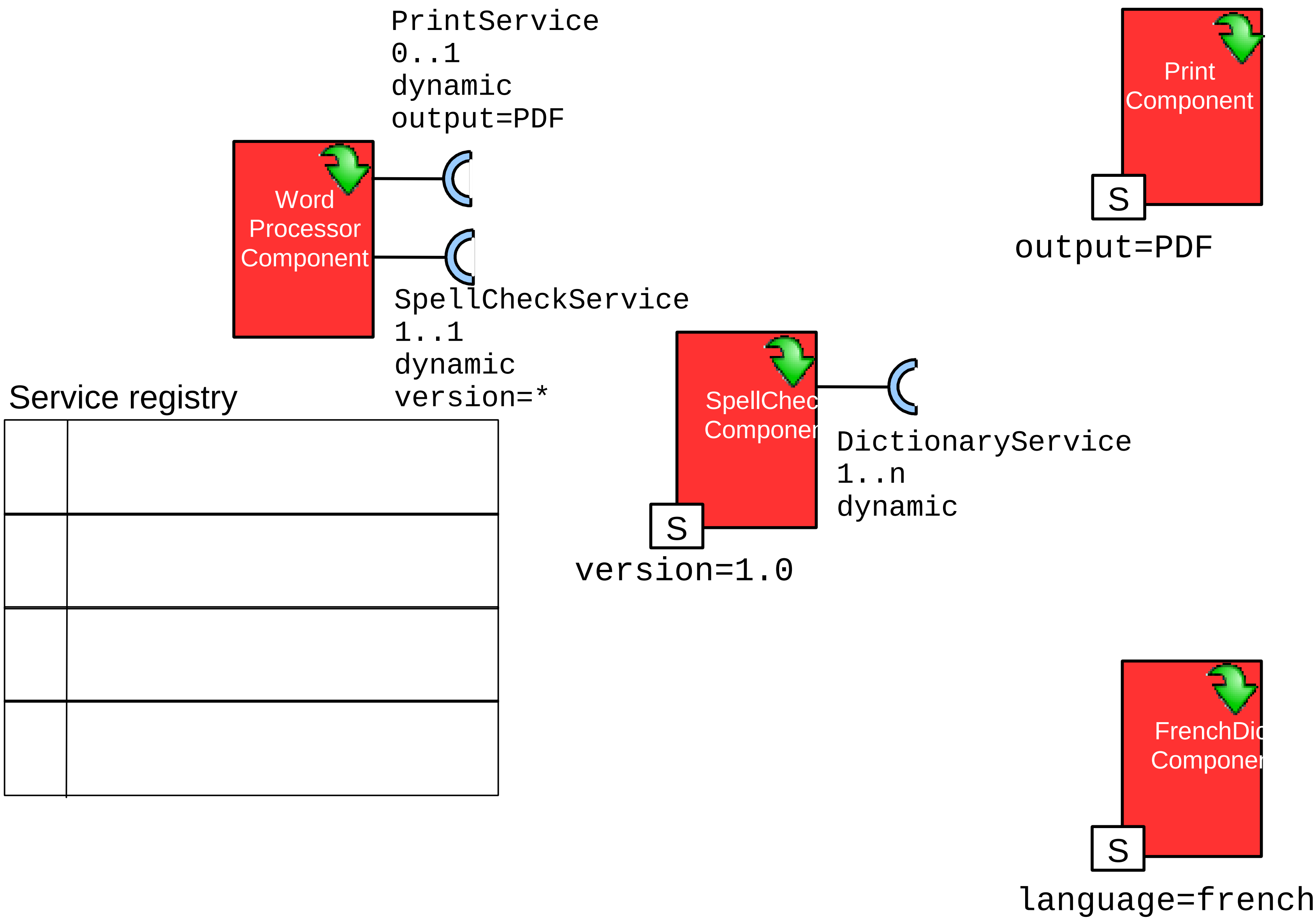
Core instance

- Guides application execution
- Extensible core
- Service provider core

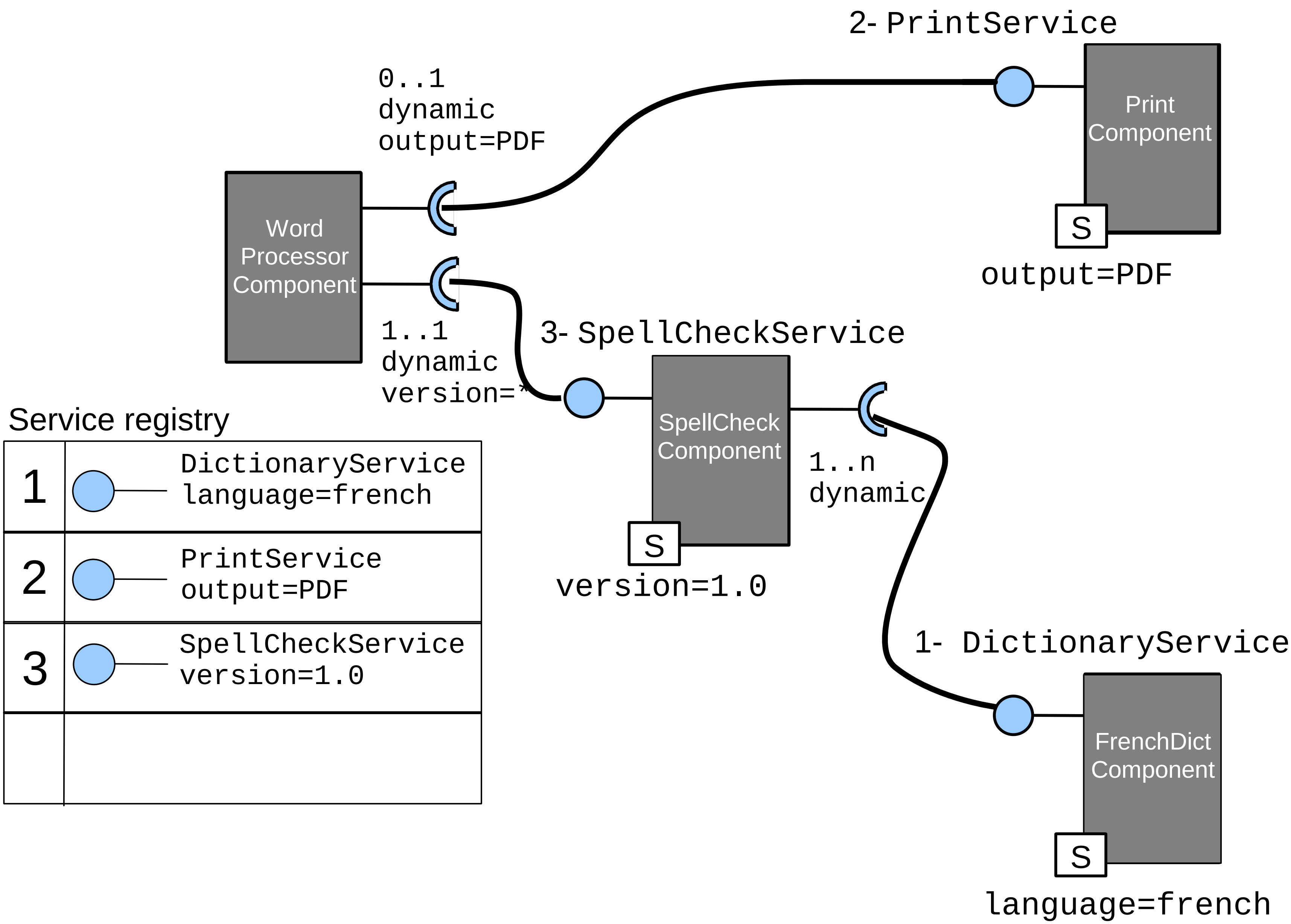
Individual instance management results in applications capable of automatic

- Self-assembly
- Functionality integration
- Functionality removal
- Functionality substitution

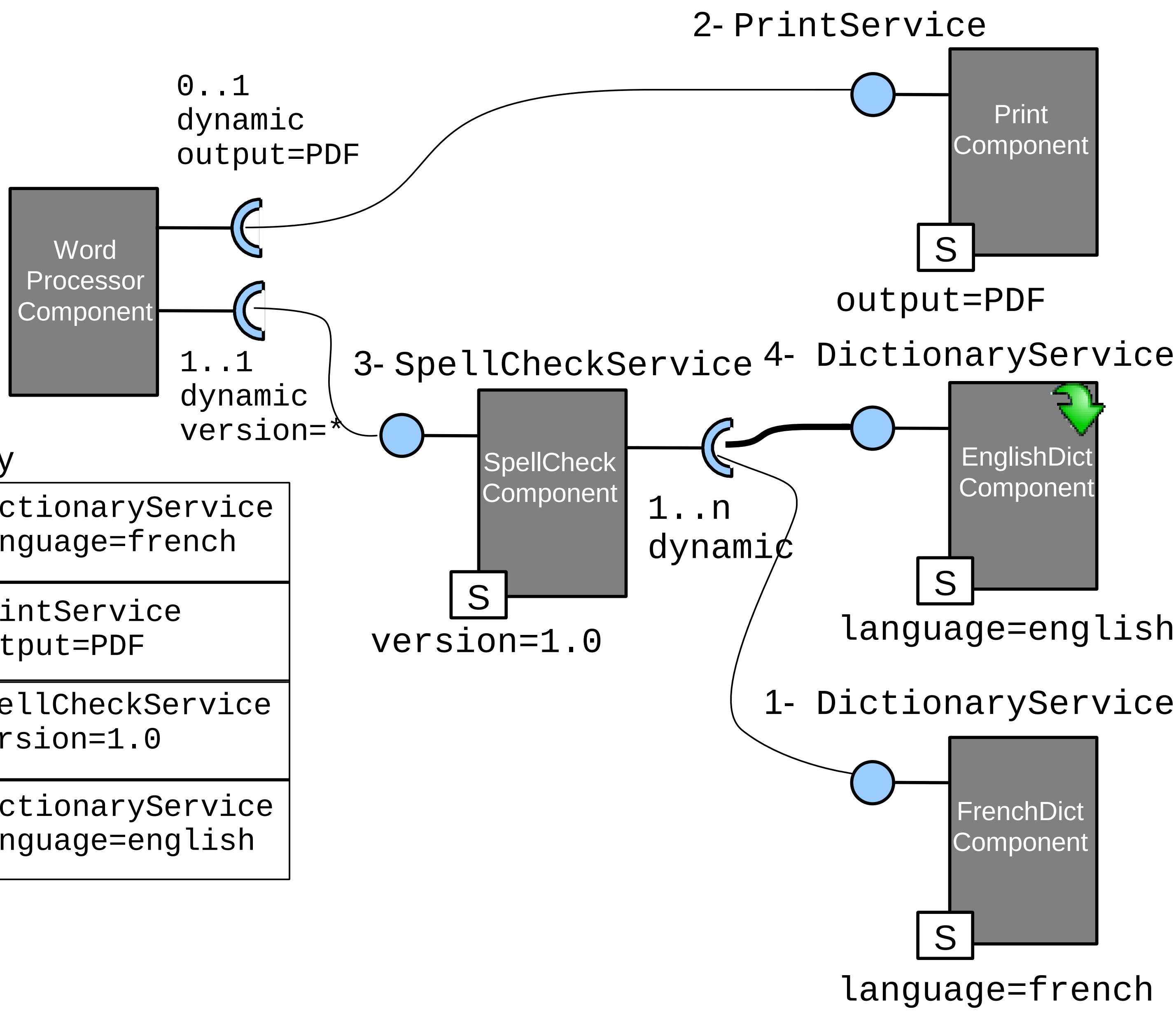
Self-Assembly



Self-Assembly



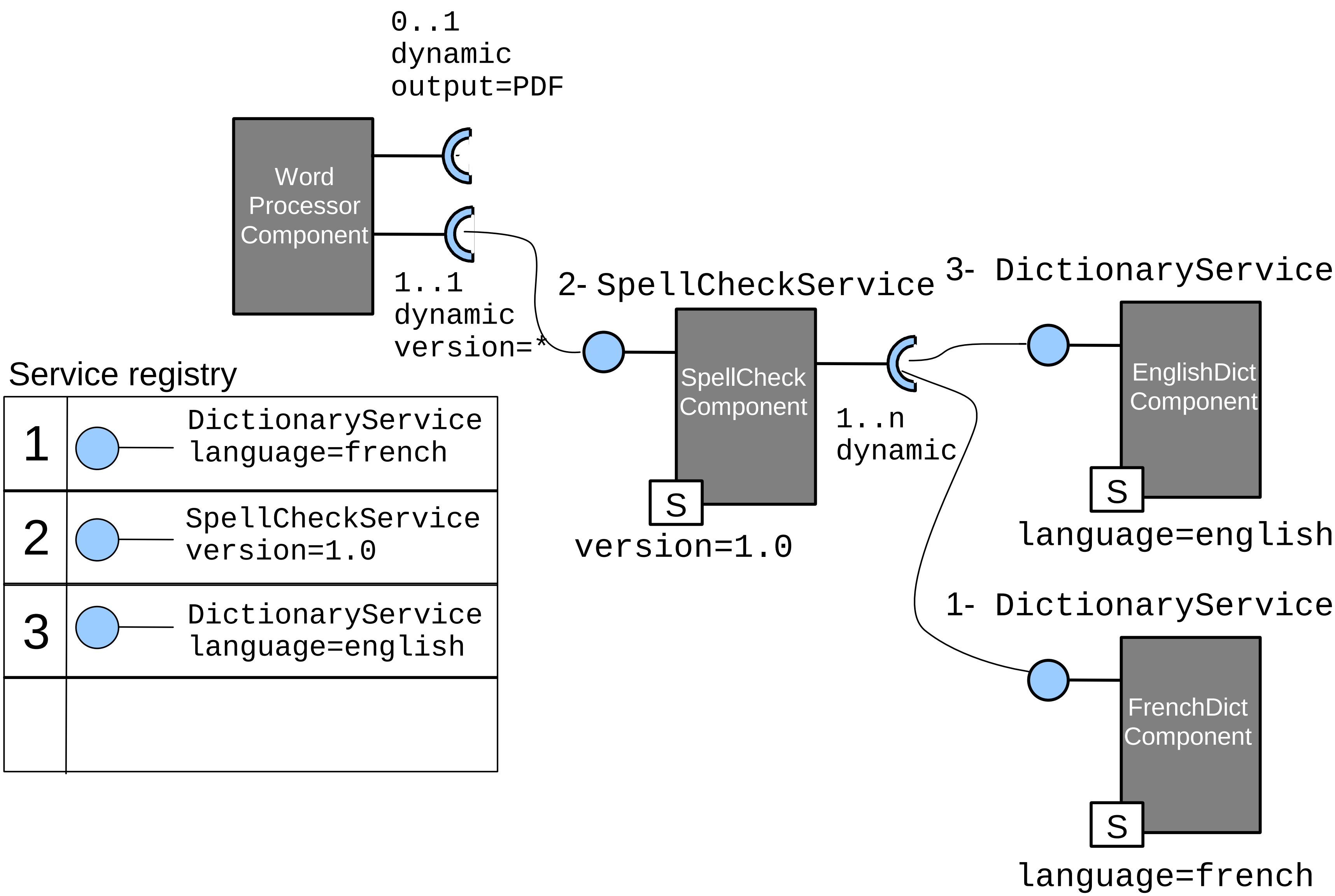
Functionality Integration



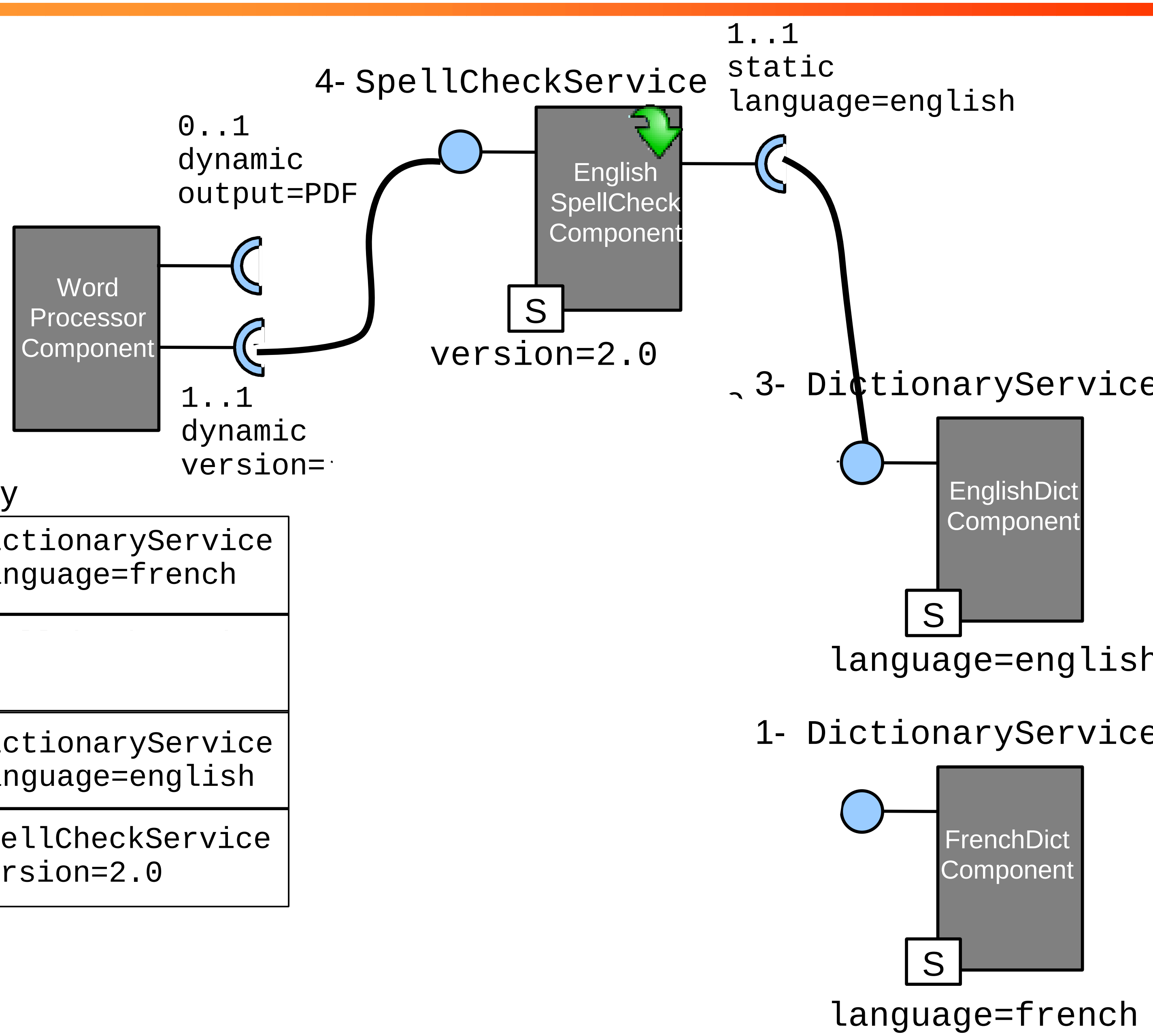
Service registry

1	DictionaryService language=french
2	PrintService output=PDF
3	SpellCheckService version=1.0
4	DictionaryService language=english

Functionality Removal



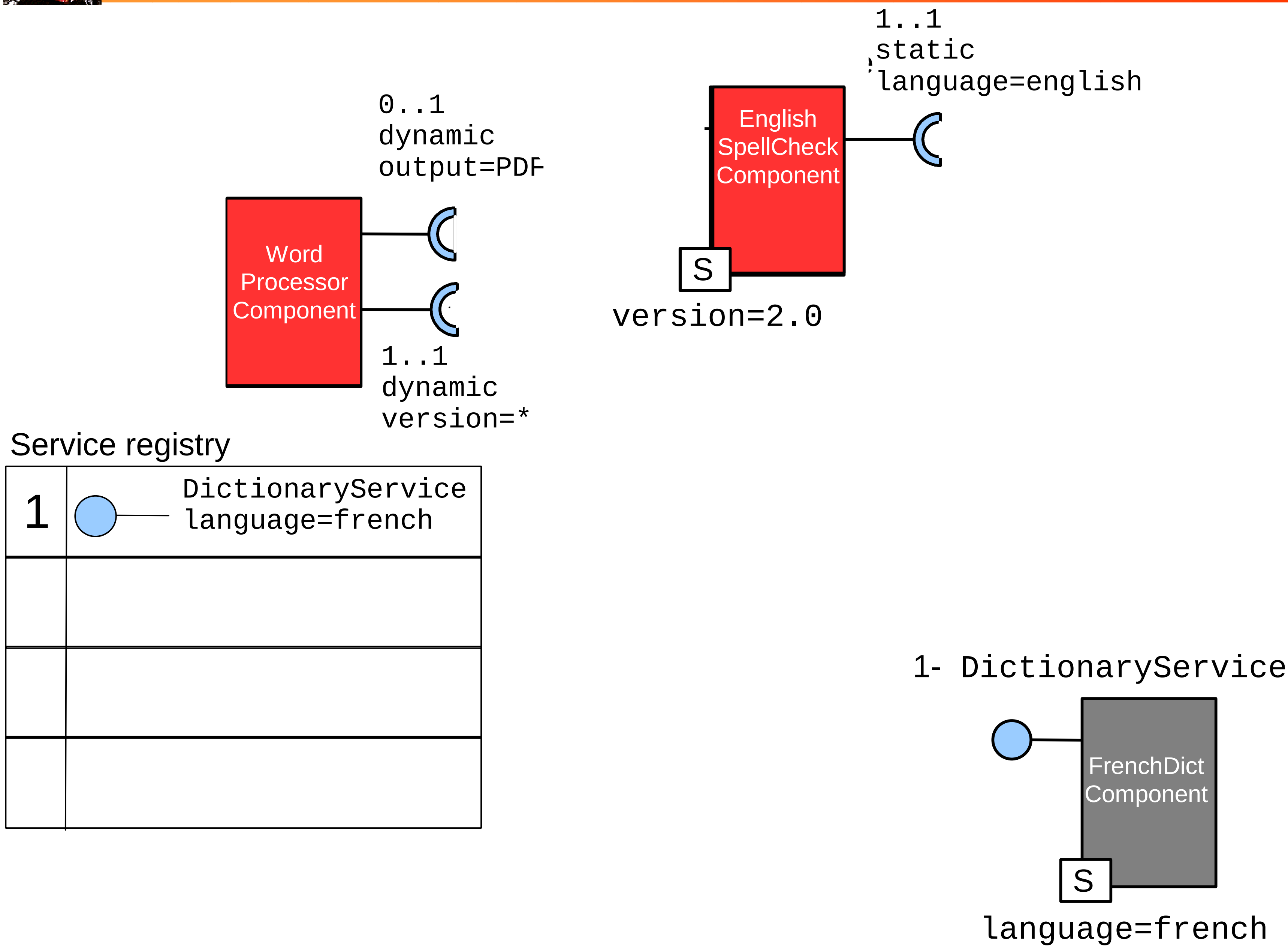
Functionality Substitution



Service registry

1	DictionaryService language=french
3	DictionaryService language=english
4	SpellCheckService version=2.0

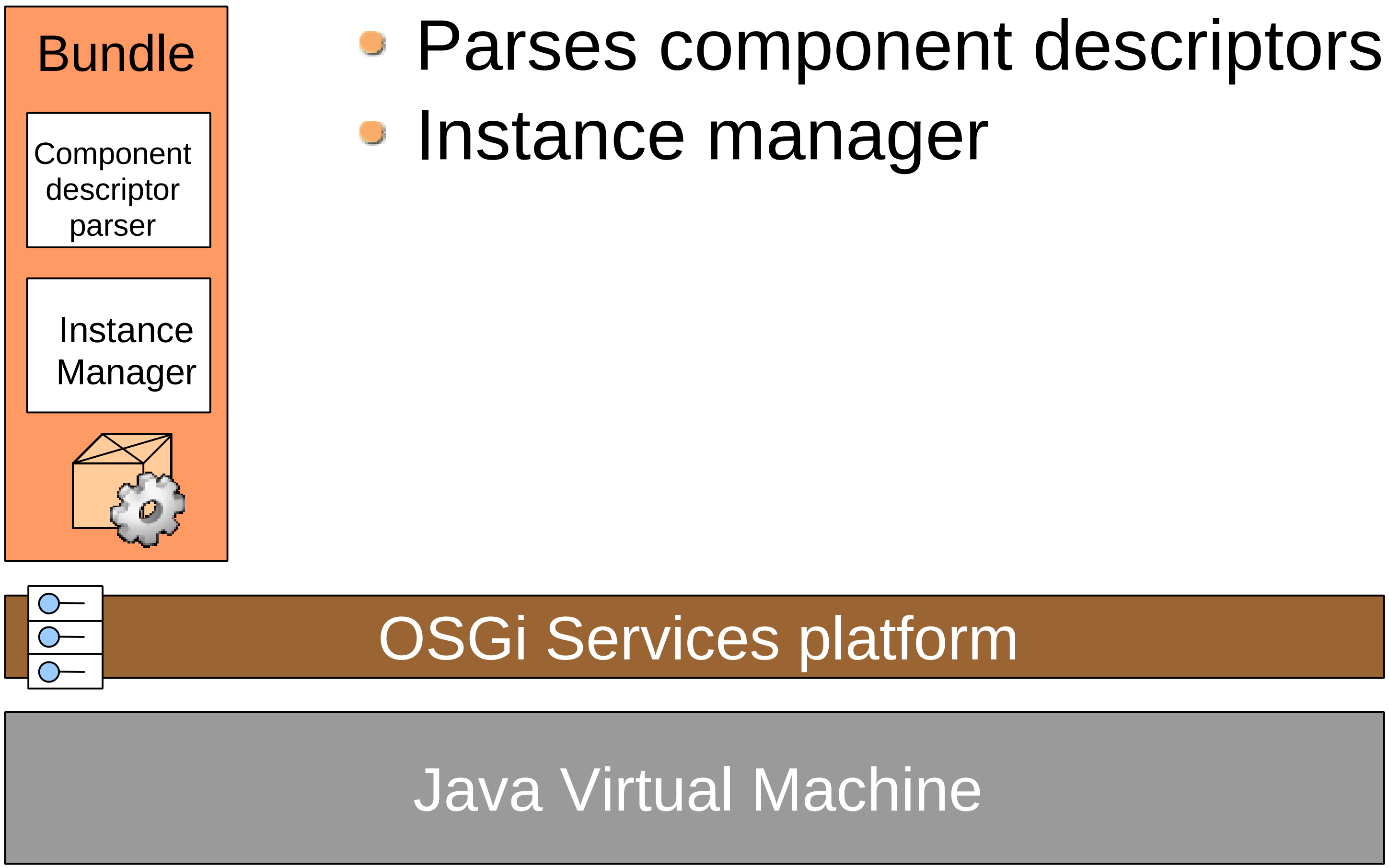
Chain Invalidation



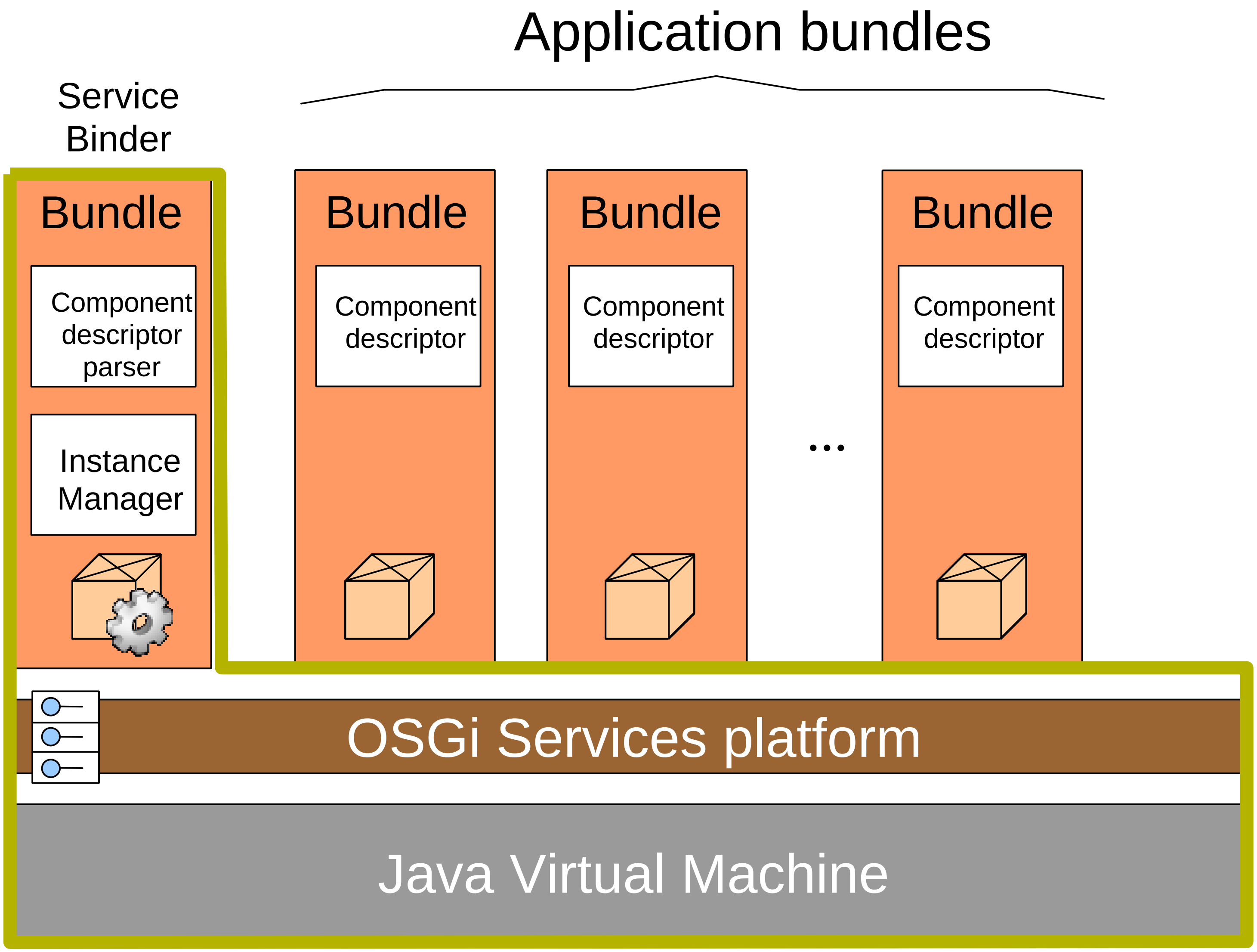
Service Binder

Service Binder

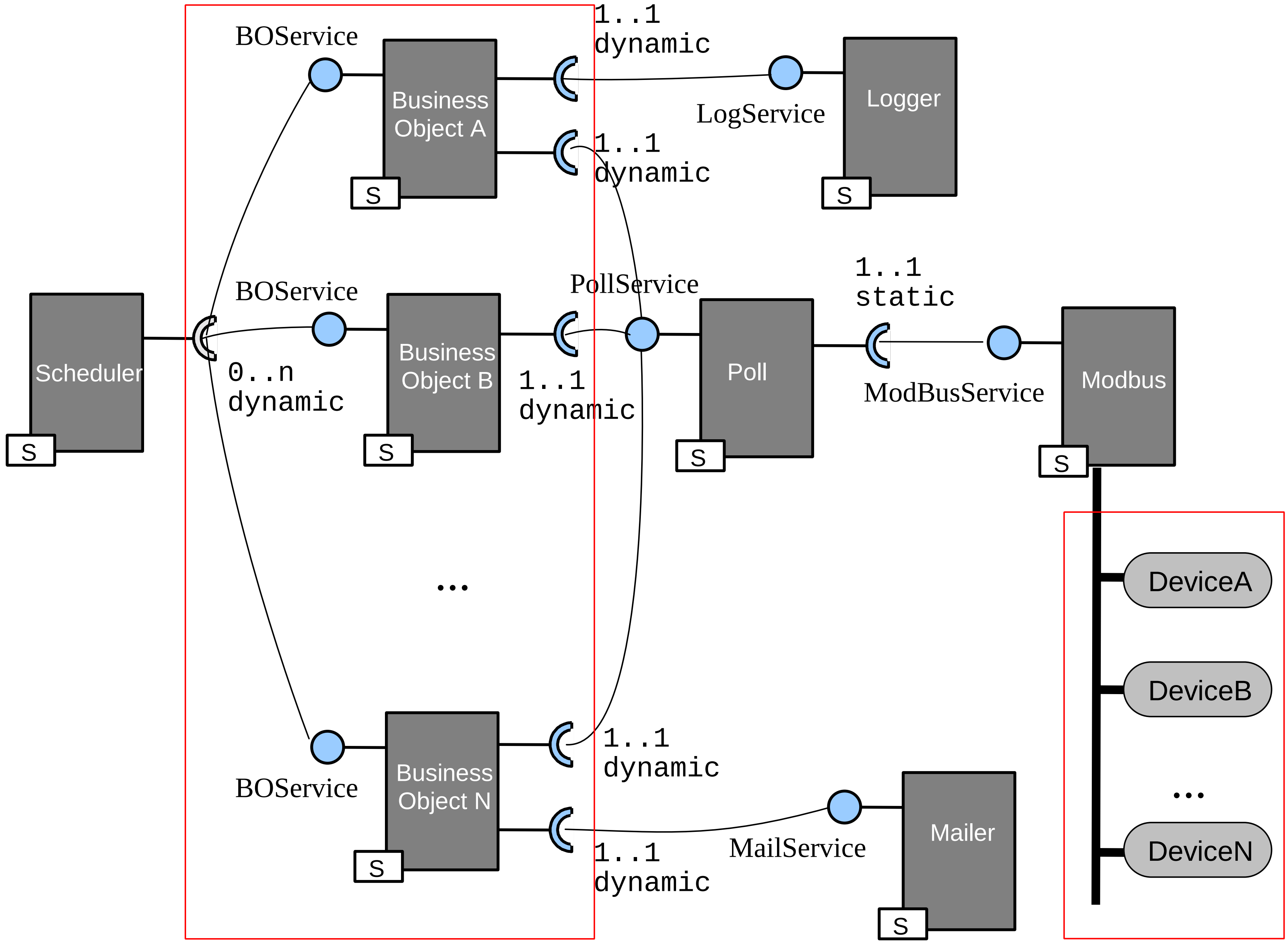
- Parses component descriptors
- Instance manager



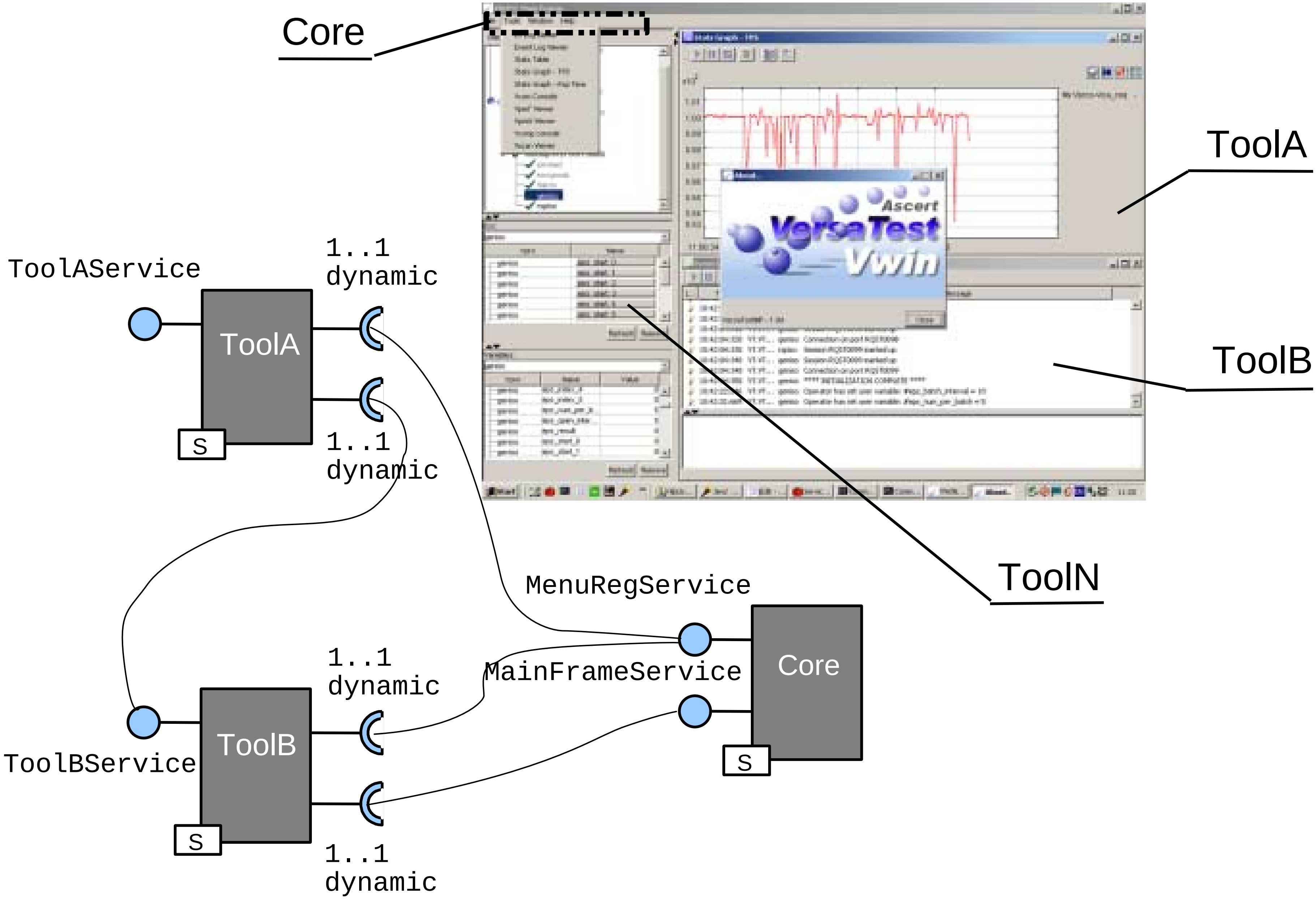
Service Binder



Schneider Electric



VersaTest Client



Conclusions

Service oriented component model + adaptation logic handled by execution environment

- Introduce and support dynamic availability
- Adaptation logic extracted from components
  - Separation of concerns

Simplifies development of OSGi applications

- Used in several R&D projects and a commercial application
- OSGi Alliance interested by these ideas
- http://gravity.sf.net/servicebinder

A basis for our work

- Applications built out of component instance compositions
- Multiple composition instances : ambiguity increases