



# Java Portlet (JSR 168)

i benefici di sviluppare portlet standard

Simone Federici

K-Tech

[www.javaportal.it](http://www.javaportal.it)



# Roadmap

1. Cosa sono le portlet
2. Architettura
3. Usare le portlet
4. Approfondimenti
5. Conclusioni



**Il talk si rivolge agli  
sviluppatori e  
analisti J2EE**

**Perché usare le  
Java portlet ?**



# Cosa sono le portlet?

**1**





Java Portlet = [ X ]

1

E' un Java Web Component !

E il suo ciclo di vita è gestito da un portal container.

Ma cosa è un Portal Container?



# Portals JSR Complaint

1

Pluto	Sun Java System
Jetspeed2	Plumtree
LifeRay	BEA
eXo	Oracle 9i
gridsphere	WebSphere
uPortal	MS SharePoint
	Broadvision



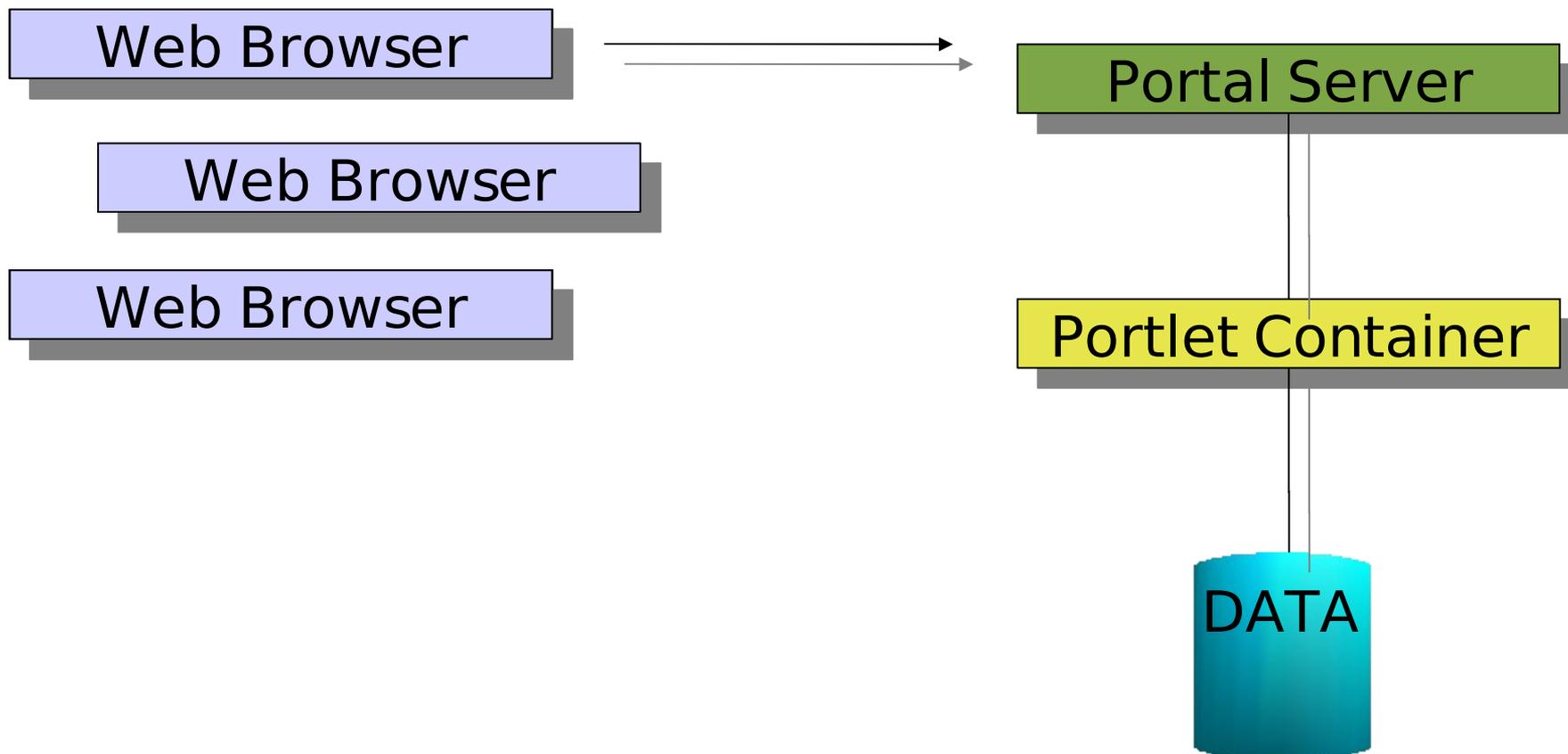
# Standard JSR 168

1

- Portlet API
- Portlet Requests
- Preferences
- User information
- Security
- Deployment packaging
- Portlet Container
- Extension of servlet container
- Contract between component and container

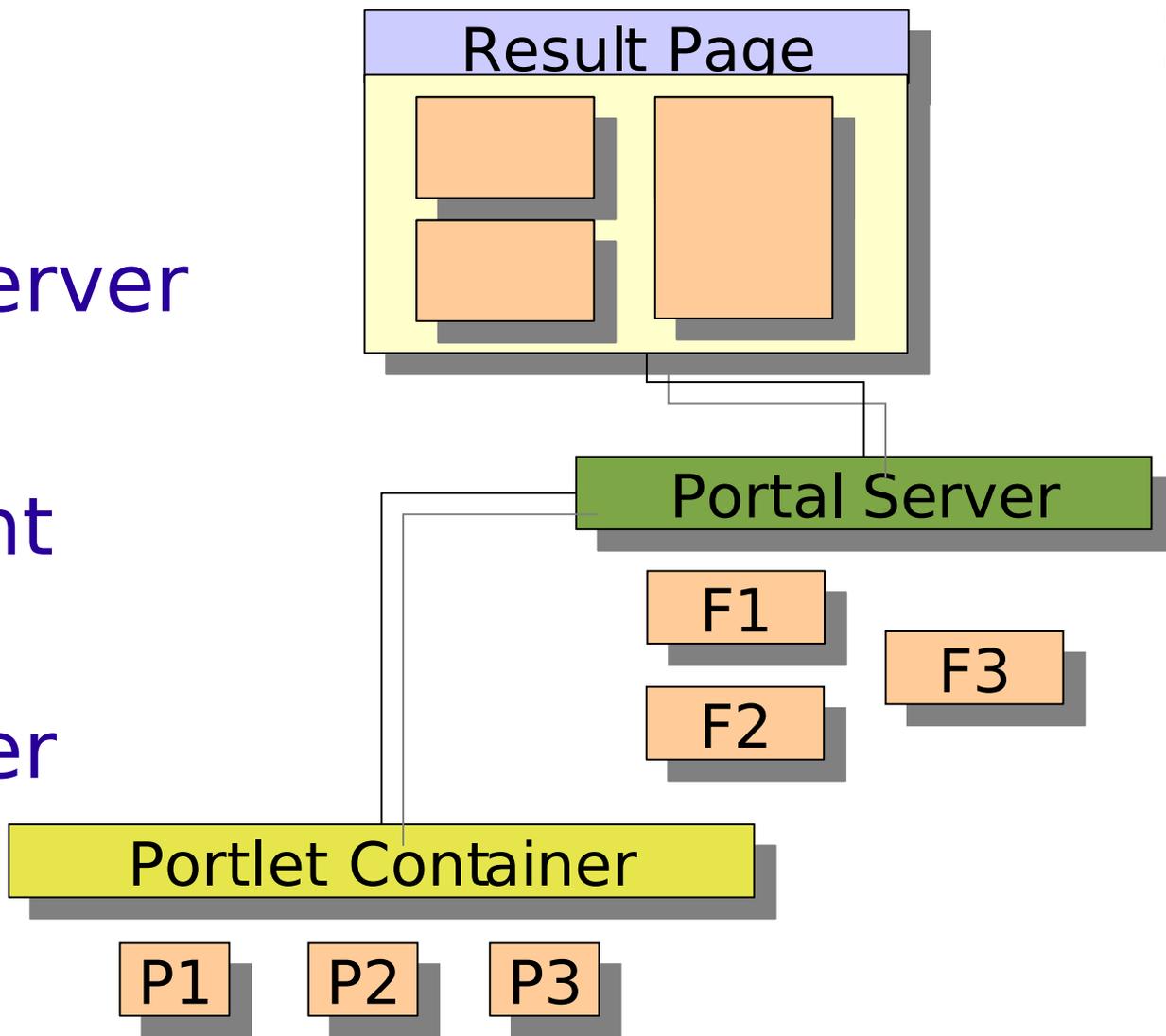


# Architecture





- Portal Server
- Portlet
- Fragment
- Portlet Container





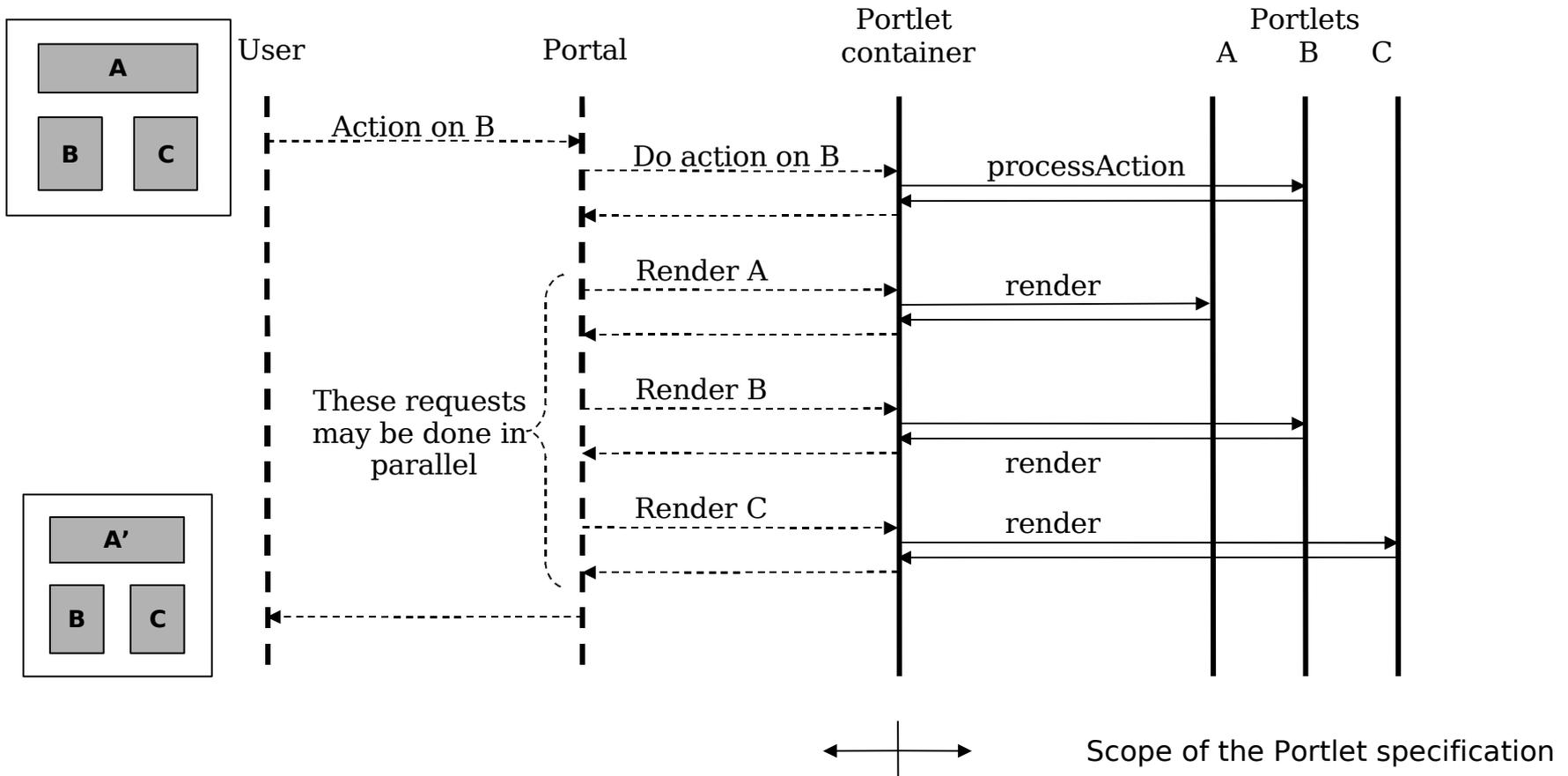
# Compiti del Portal server <sup>2</sup>

- Aggregation
- Layout management
- Page personalization and configuration engines
- Portal administration and configuration



# Portal and Portlet Interaction

2





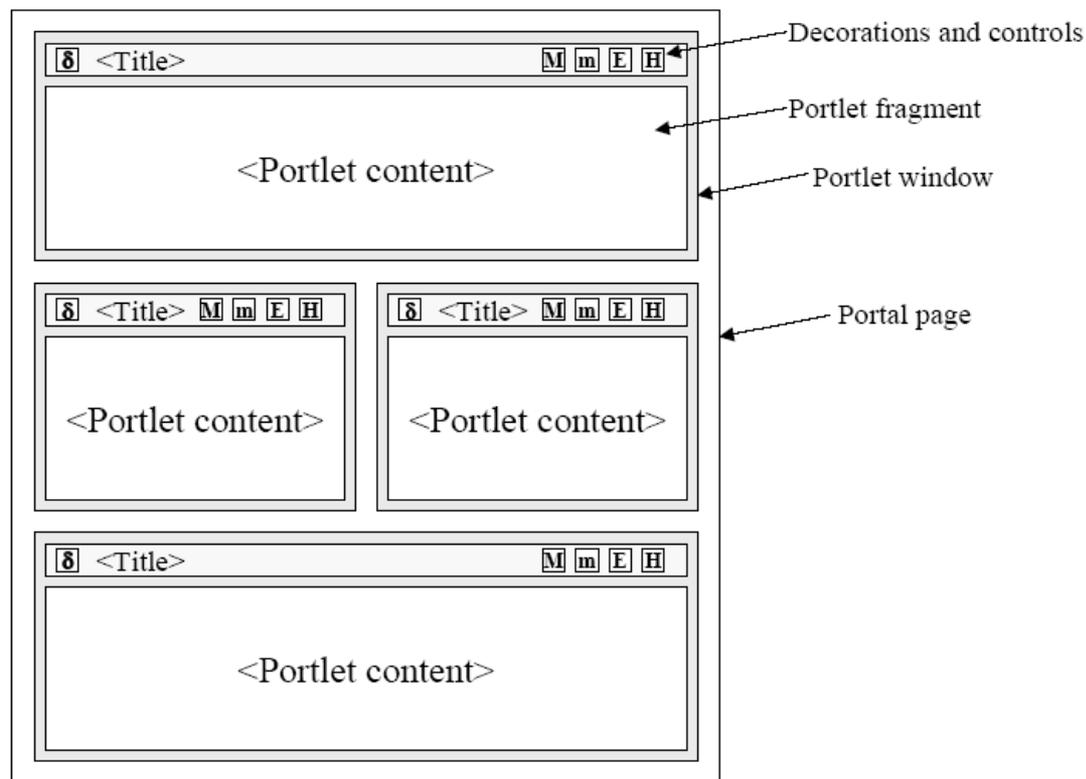
# Java Portlet

## State:

- Normal
- Massimizza
- Minimizza

## Modes:

- VIEW
- EDIT
- HELP





# States

Normal, minimizzata e massimizzata.

Il portal server ha il compito di gestire la **renderizzazione** delle portlet nei deversi stati.

Il compito del container è **gestire il cambiamento** dello stato delle portlet.



# Modes

## VIEW, EDIT e HELP

Il Portal Server puo' aggiungere altri stati. (*deve implementarli*)

E' poi possibile accedere a questi aggiungendo un  
<custom-portlet-mode>  
nel file descriptor.



# Perche le portlet? non posso fare tutto tramite servlet?

in comune:

- J2EE Web Component
- Gestita dal container
- Generate da un paradigma request/response



## Differenze Portlet & Servlet

- Mentre la servlet genera una pagina la portlet genera un fragment
- Le portlet non mappate da una specifica url
- Le P. hanno un sofisticato schema di request action / render
- Le P. hanno stati e “modes” standard che definiscono le regole di azione e visualizzazione
- Le P. usano dei meccanismi di accesso e persistenza alle informazioni di configurazione
- Le P. possono fare portlet rewriting così da essere indipendenti dall'implementazione del portale
- Le P. hanno due differenti session scope: application e private.
- Le P. non possono alterare l'HTTP header o setare la response encoding.



# Portability

2

La “**stessa**” portlet puo' essere usata  
in contesti estremamente diversi

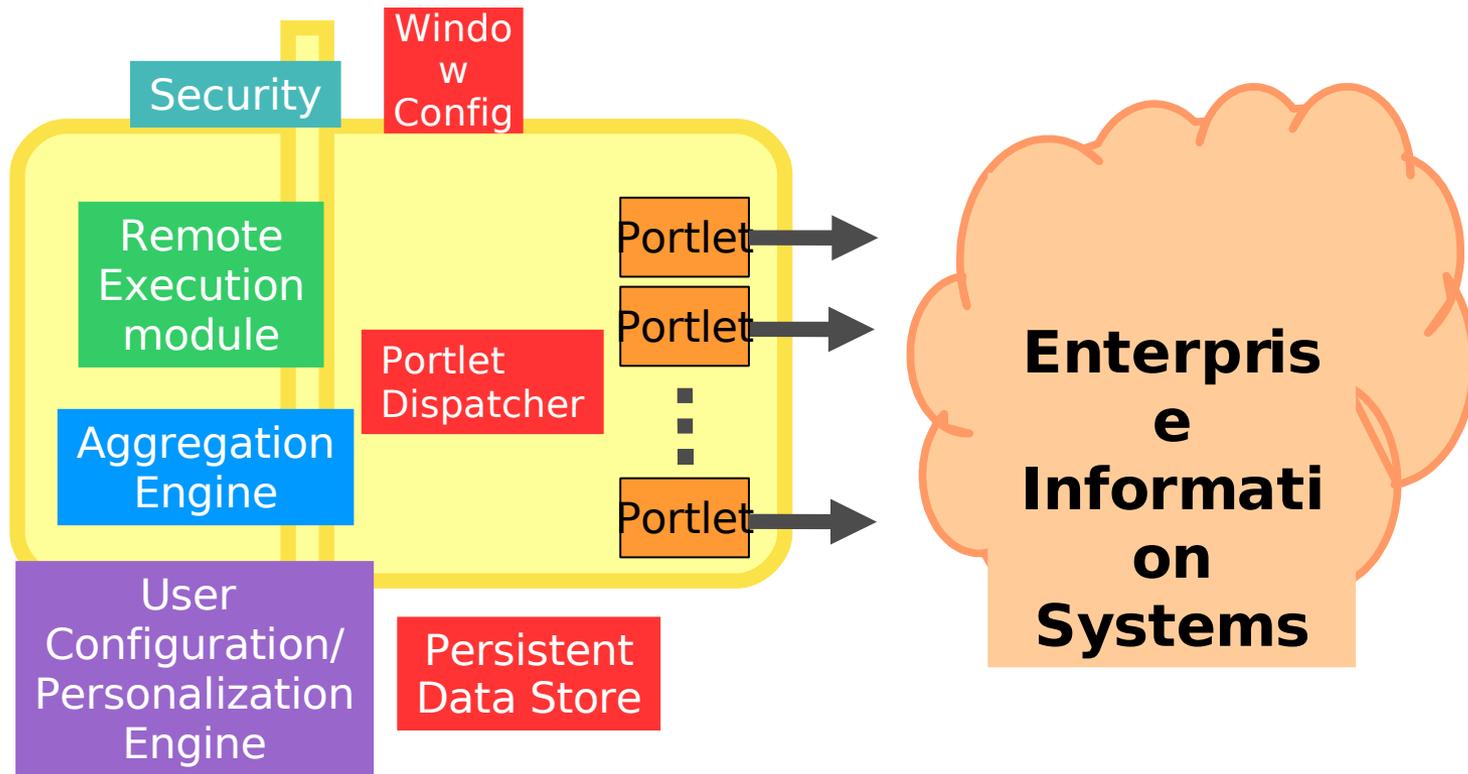
basta attivarla (deploy) in un portal  
Compliant alle specifiche JSR168



# Portal/Portlet Architecture

2

Portal server





## Portlet Life Cycle

3

I metodi chiamati dal container sono:

- **init()**
- **destroy()**
- **processAction()**
- **render()**

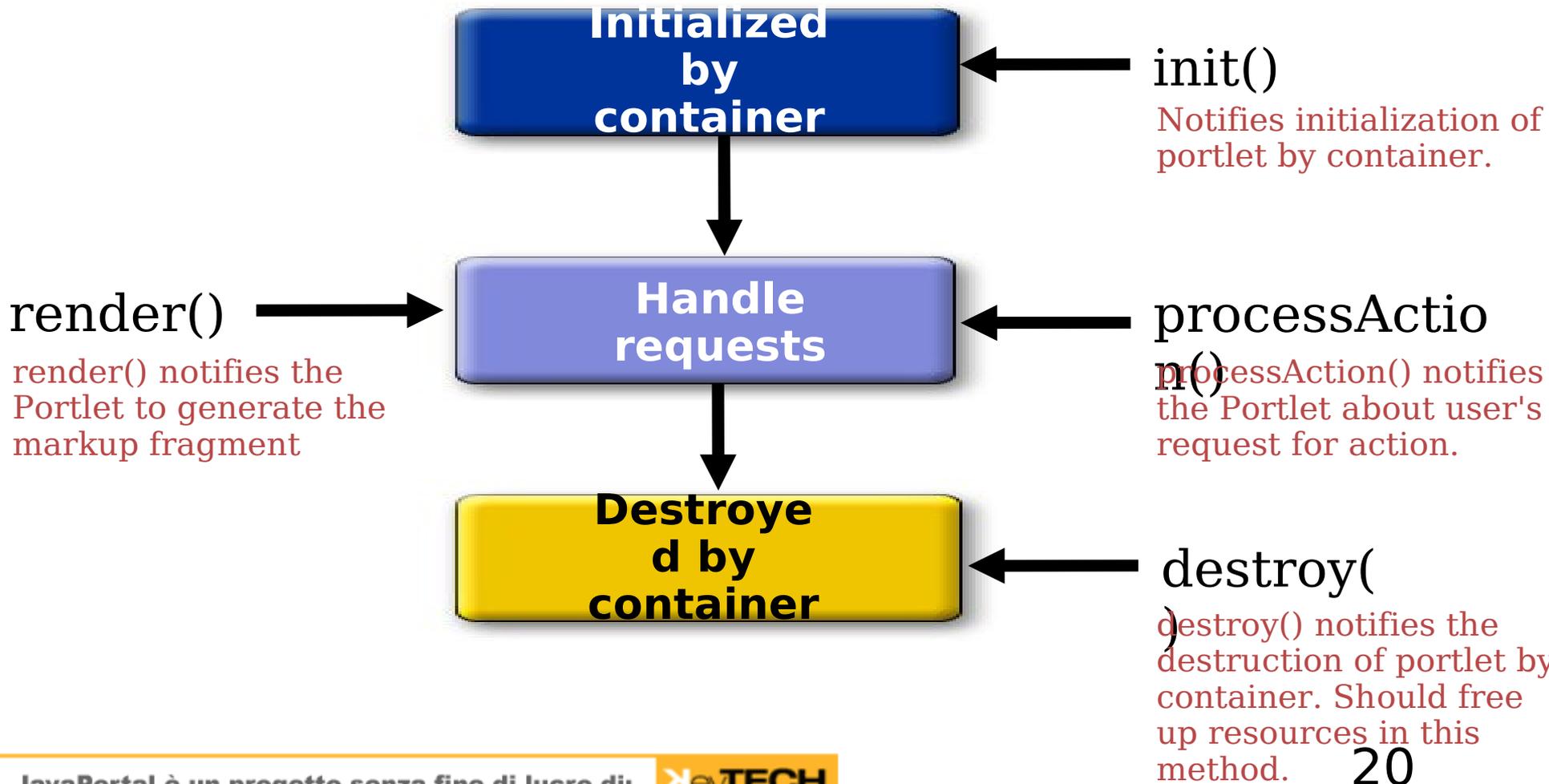
+ chiamati dal render di GenericPortlet:

- **doView()**
- **doEdit()**
- **doHelp()**



# Portlet Life Cycle

3





## Init / destroy

```
void init(PortletConfig config) {  
    config.getInitParameter(java.lang.String name);  
    config.getInitParameterNames();  
    config.getPortletContext();  
    config.getPortletName();  
    config.getResourceBundle(java.util.Locale locale);  
        this.config = config;  
}
```

```
void destroy () { config=null; }
```



# processAction()

3

Viene chiamato dal container quando una “azione” viene effettuata nella portlet.

*PortletURL()*  
*createActionURL()*



# render()

Viene chiamato ogni volta che una portlet deve essere visualizzata su una pagina

*PortletURL()*  
*createActionURL()*



# GenericPortlet

3

E' una implementazione  
`javax.portlet.Portlet`

sostanzialmente aggiunge un `dispatcher()` che chiama i seguenti metodi a secondo dello stato della portlet.

- `doView()`
- `doEdit()`
- `doHelp()`



## Portlet Example

3

```
public class HelloWorldPortlet extends
    GenericPortlet {

    public void doView ( RenderRequest request,
                        RenderResponse response)
                        throws PortletException {

        response.setContentType("text/html");
        PrintWriter writer = response.getWriter();
        Writer.write("Hello World");
    }
}
```



## Render by JSP?

3

```
public class HTMLPortlet extends GenericPortlet
{
    public void doView(RenderRequest request, RenderResponse response) throws ...
    {
        PortletPreferences pref = request.getPreferences();
        String path = pref.getValue("path", "/hello_jip.jsp");

        PortletRequestDispatcher rd = getPortletContext().getRequestDispatcher(path);
        rd.include(request, response);
    }
}
```



# CustomTag

```
<%@taglib uri='http://java.sun.com/portlet'  
          prefix="portlet" %>
```

```
<portlet:defineObjects/>
```

```
<!-- RenderRequest renderRequest -->
```

```
<!-- RenderResponse renderResponse -->
```

```
<% PortletSession renderSession =  
renderRequest.getPortletSession(); %>
```



# Multithread

Il container può chiamare i render delle varie portlet da visualizzare in modo concorrente.

Ricordiamo che le azioni sono gestite nel `processAction()`

Il render va usato solamente per renderizzare



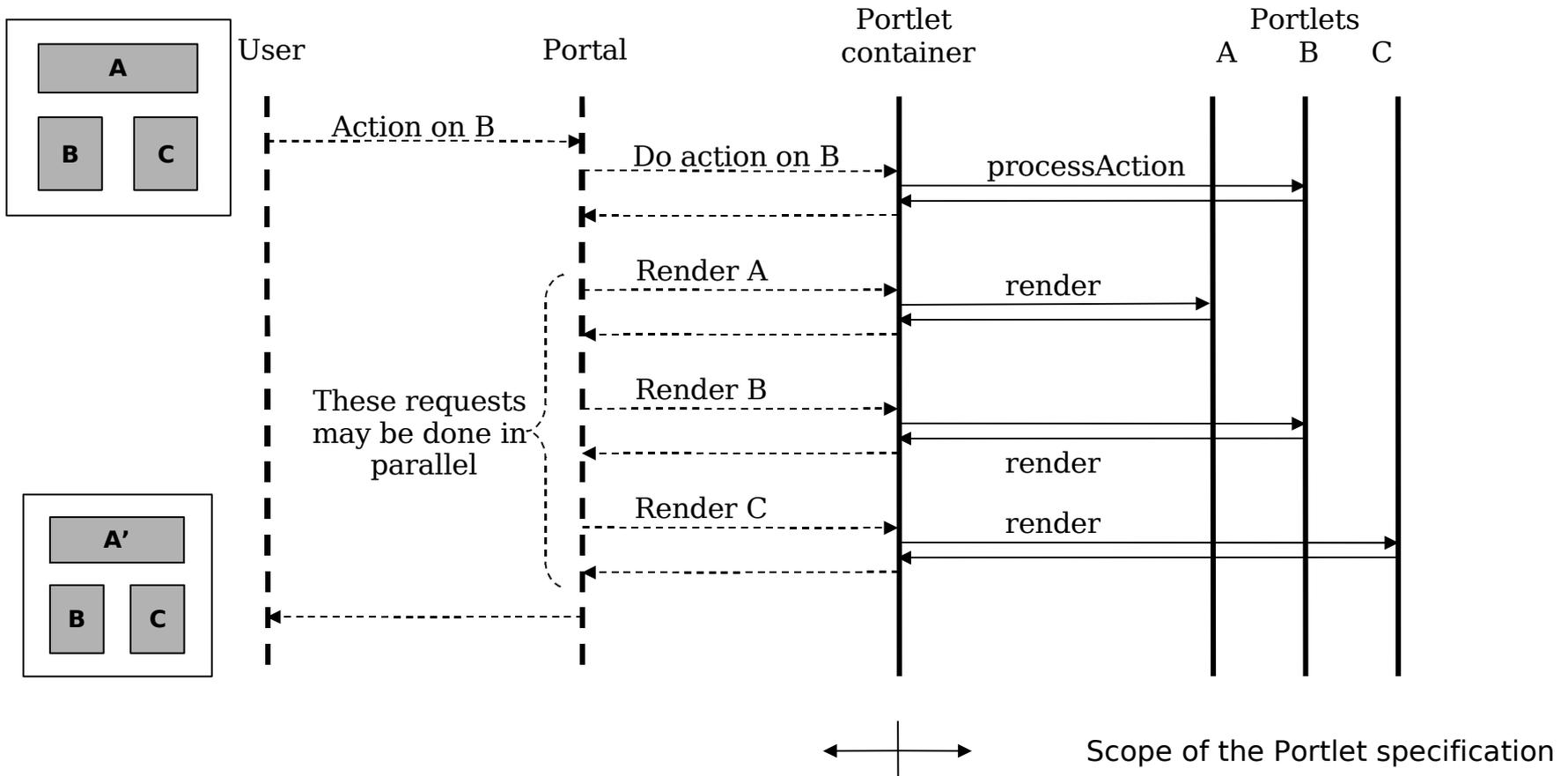
Se il contenuto di una portlet dipende da un'altra?

Ci sono strutture dati condivise dalle portlet / servlet

Non usate mai variabili di classe !!



# Portal and Portlet Interaction





# Comunicazione attraverso <sup>4</sup> PortletSession

**APPLICATION\_SCOPE**  
ossia nell'HttpSession

**PORTLET\_SCOPE**  
*scope privato (\*)*



# Portlet Web Application

4

- All resources, portlets, deployment descriptors are packaged in one web application archive (**WAR** file)
- PortletApp\
  - jsp
  - htmls
  - WEB-INF\
    - web.xml
    - portlet.xml
    - sun-portlet.xml
    - classes\
      - Lib\
        -



# Localization

4

- Portlets can be localized by using resource bundles
- Resource bundles are specified in deployment descriptor
- Portlet can access resource bundle via  
`PortletContext.getResourceBundle()`  
API



## Portlet.xml

```
<portlet-app>
  <portlet>
    <portlet-name>HelloWorldPortlet</portlet-name>
    <portlet-class>
      Custom.HelloWorldPortlet
    </portlet-class>
    <expiration-cache>0</expiration-cache>
    <supports>
      <mime-type>text/html</mime-type>
      <portlet-mode>VIEW</portlet-mode>
    </supports>
    <portlet-preference>
      <preference>
        <name>locale </name>
        <value>USA</value>
      </preference>
    </portlet-preference>
  </portlet>
</portlet-app>
```



# Apache Bridge Struts

4

<http://portals.apache.org/bridges/multiproject/portals-bridges-struts/>

Sviluppate una Applicazione Struts!!

l'applicazione potrà poi essere usata come una webapp standard oppure messa dentro un portal come una portlet.

Richiede che il container implementi una ulteriore interfaccia...



Grazie alla Community di Javaportal  
che sempre cresce sempre più

Grazie a tutti voi di essere qui

Restate con noi.. perchè

Programmiamo da far Paura!



# Framework

Si sposano bene con altri  
framework?



## Resources

- Javapassion  
<http://www.javapassion.com>
- Java Community Process  
<http://www.jcp.org/en/jsr/detail?id=168>