OSGi in Liferay

OSGi in Liferay 6.2 e 7.0

Mauro Mariuzzo, Software Architect – SMC Treviso

Chi ha tempo non aspetti tempo

Chi ha tempo non aspetti tempo

Tanto materiale su come usare OSGi in Liferay 6.2

- https://www.liferay.com/it/web/eduardo.garcia/blog/-/blogs/leveraging-osgi-to-create-extensible-applications-for-liferay-6-2
- https://dev.liferay.com/develop/tutorials/-/knowledge_base/6-2/developing-osgi-plugins-for-liferay
- https://www.liferay.com/it/web/raymond.auge/blog/-/blogs/places-to-hook-into-liferay-from-osgi
- https://www.youtube.com/watch?v=KPjmB5yj8Og
- https://www.youtube.com/watch?v=k2qxh7FzfH8
- http://www.slideshare.net/MilenDyankov1/extensible-plugins

Google ti è amico

La pazienza è la virtù dei forti 

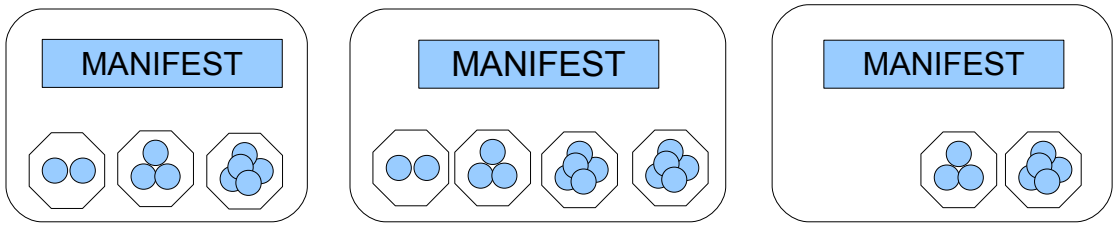
Concetti base di OSGi

3 cose da sapere prima di continuare

#1 OSGi Bundles

Il bundle è il tuo plugin

```
Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: Liferay Application List API
Bundle-SymbolicName: com.liferay.application.list.api
Bundle-Vendor: Liferay, Inc.
Bundle-Version: 1.0.0
Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.7))"
....
```



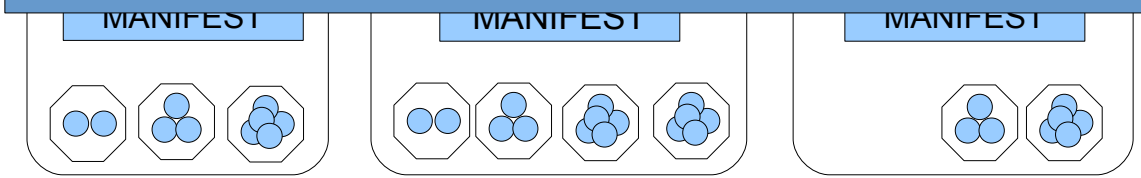
#1 OSGi Bundles

Il bundle è il tuo plugin

Manifest-Version: 1.0
Bundle-N
Bundle-N
Bundle-S
Bundle-V
Bundle-V
Require-
....

Diversi strumenti di build (Ant, Maven, Gradle)
sono in grado di produrre bundle OSGi

Quindi anche Liferay SDK!

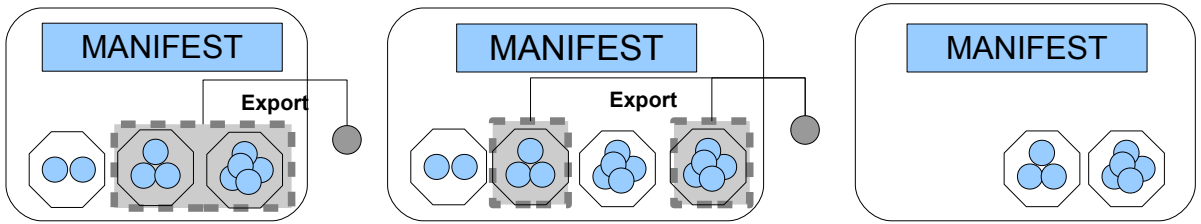


#2 Package dependencies

Descrivere cosa forniamo e cosa ci attendiamo

Export-Package:
com.liferay.application.list;version="1.0.0",
com.liferay.application.list.adapter;version="1.0.0",
.....

Import-Package:
com.liferay.osgi.service.tracker.map,
com.liferay.portal.kernel.deploy.hot,
com.liferay.portal.kernel.exception, com.liferay.portal.kernel.language,
com.liferay.portal.kernel.log, com.liferay.portal.kernel.util,
....

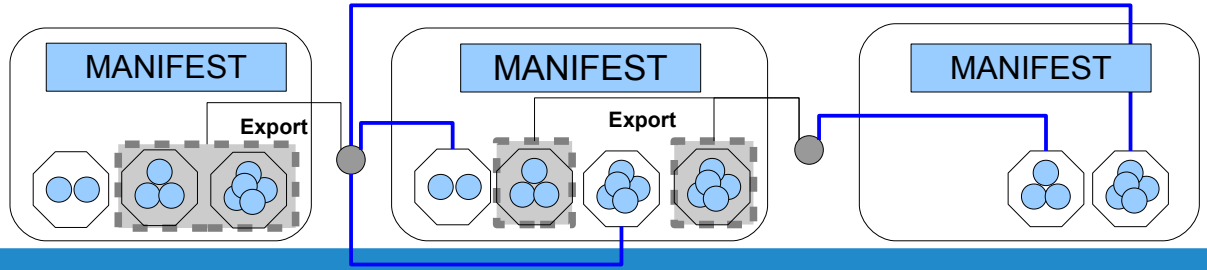


#2 Package dependencies

Descrivere cosa forniamo e cosa ci attendiamo

Export-Package:
com.liferay.application.list;version="1.0.0",
com.liferay.application.list.adapter;version="1.0.0",
.....

Import-Package:
com.liferay.osgi.service.tracker.map,
com.liferay.portal.kernel.deploy.hot,
com.liferay.portal.kernel.exception, com.liferay.portal.kernel.language,
com.liferay.portal.kernel.log, com.liferay.portal.kernel.util,
....



#2 Package dependencies

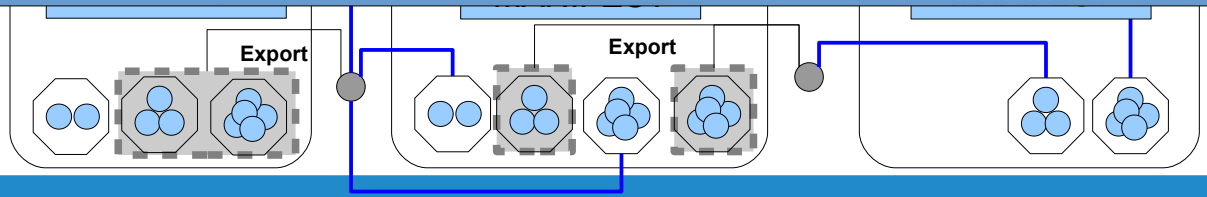
Descrivere cosa forniamo e cosa ci attendiamo

Export-Pac
com.liferay
com.liferay
.....

Il tool BDN (bnd.bndtool.com) è in grado di scoprire le dipendenze e generare automaticamente le voci Import/Export nel MANIFEST.

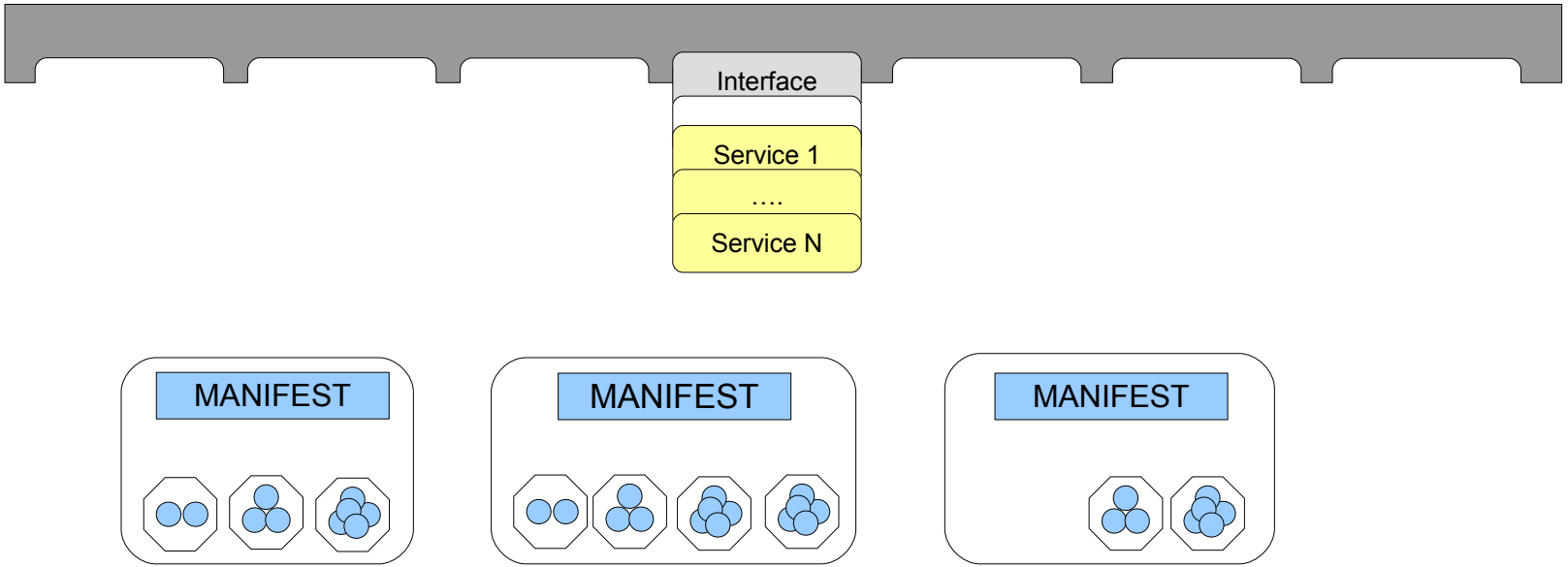
Liferay SDK consente di usare BND

.language,



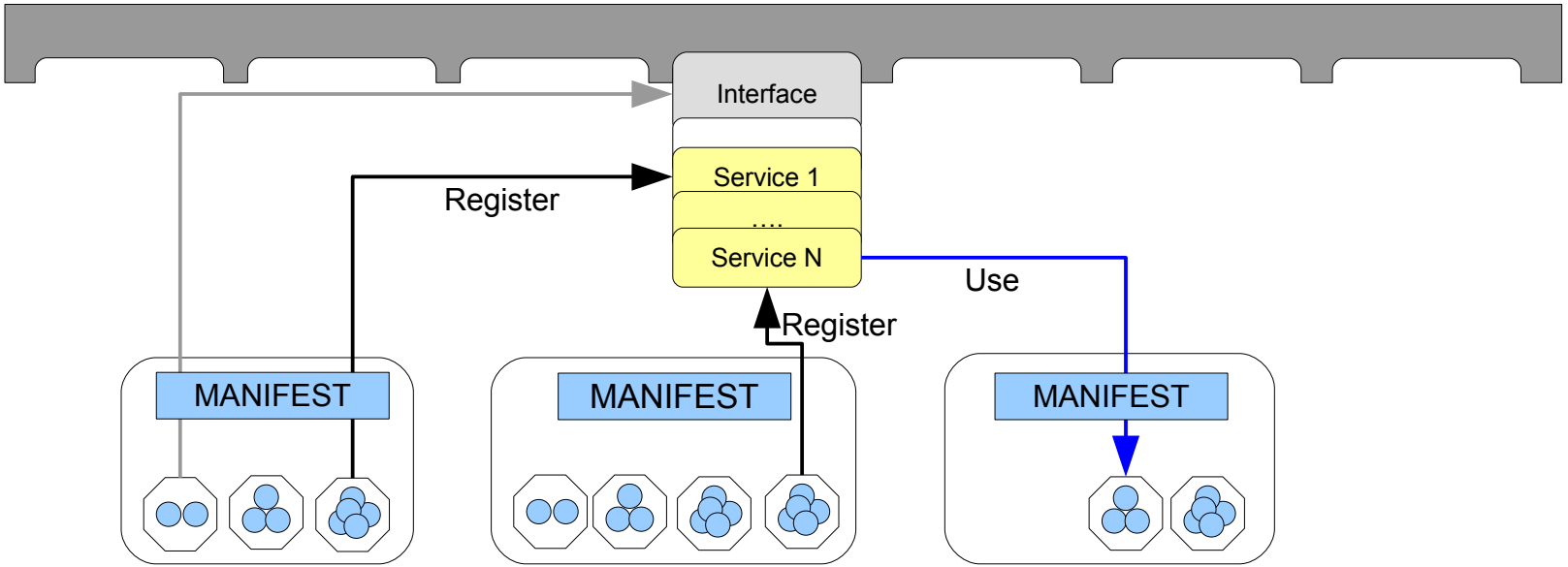
#3 OSGi services

Come forniamo e consumiano funzionalità



#3 OSGi services

Come forniamo e consumiano funzionalità



#3 OSGi services

Come forniamo e consumiano funzionalità

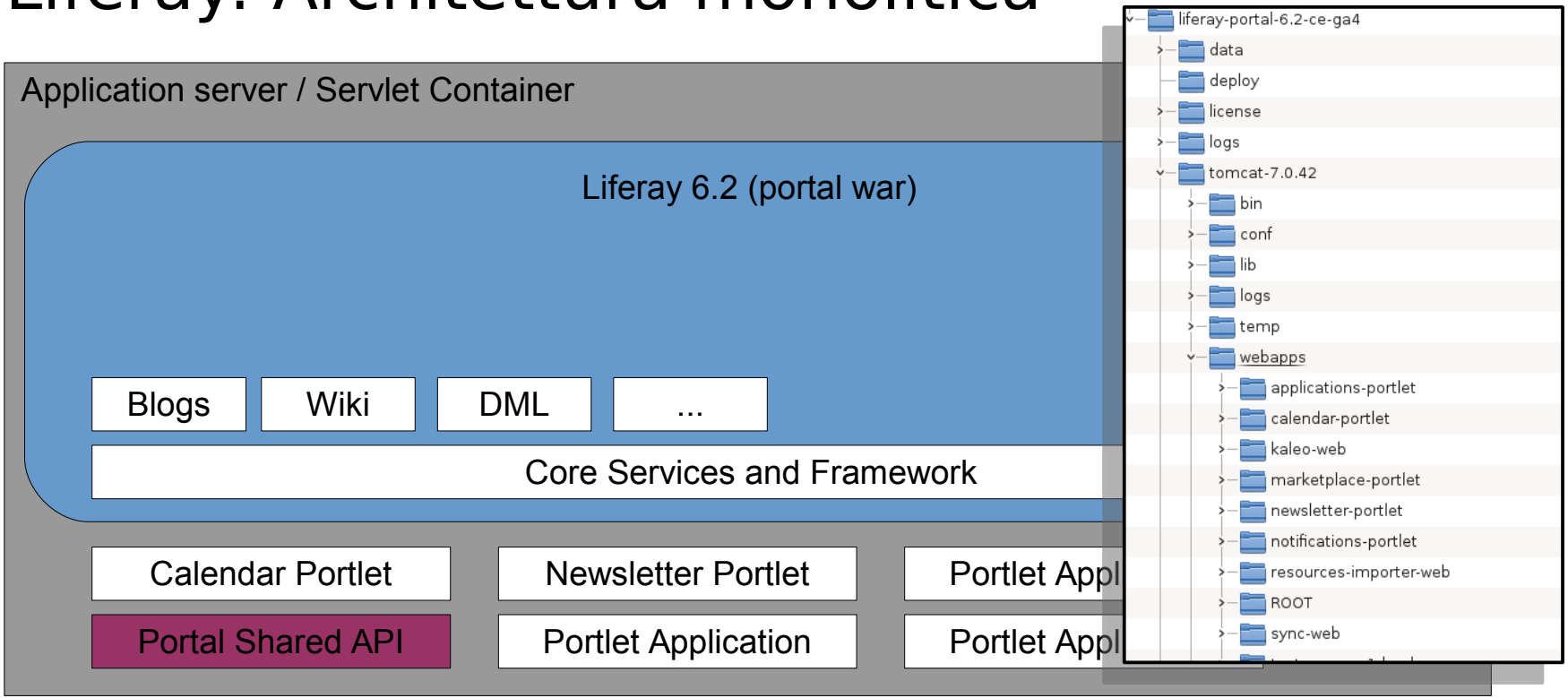
Liferay...

Chi non conosce Liferay?

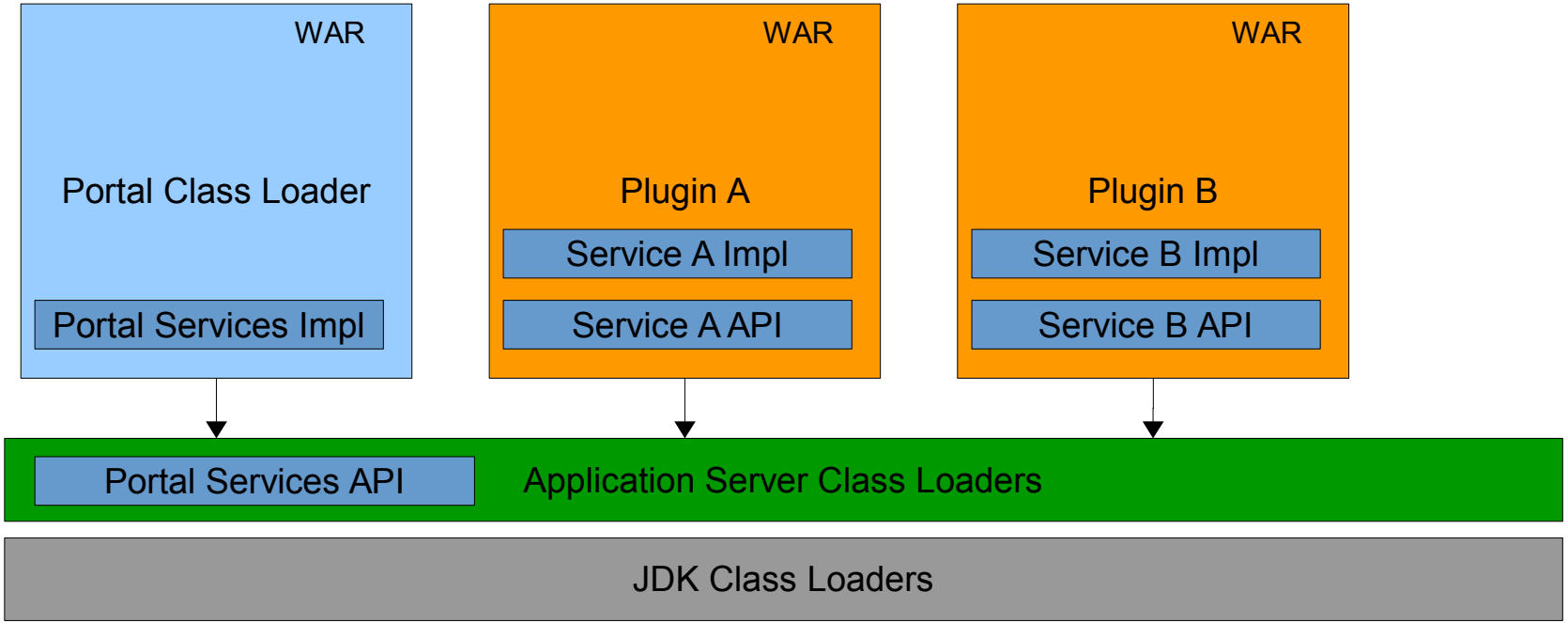
Liferay...

- Soluzione Portal leader di mercato che implementa le specifiche Portlet 1.0 e 2.0
- 5 milioni di linee di codice
- Circa 70 portlet disponibili Out of the Box
- Diverse funzionalità: Web Content Management, Document Management, Workflow, Search, Enterprise Collaboration, Social Networking, ...
- Marketplace con più di 400 applicazioni

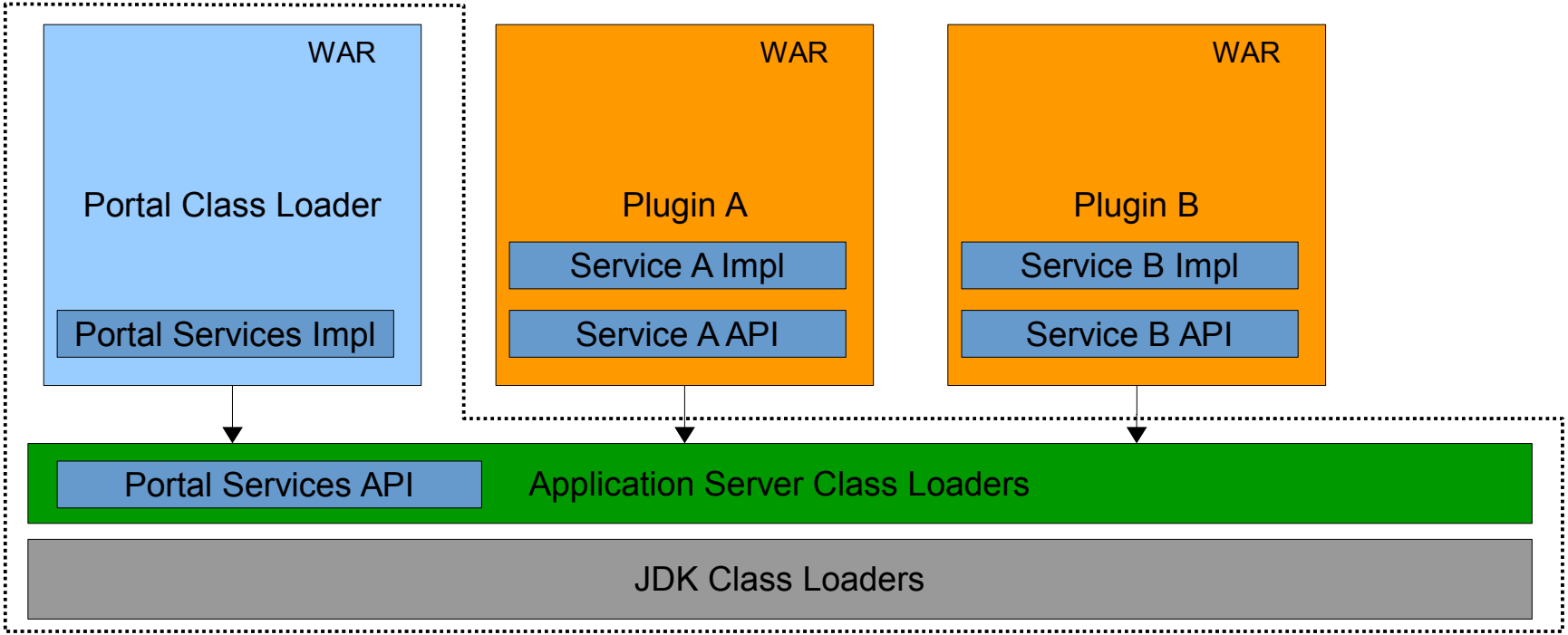
Liferay: Architettura monolitica



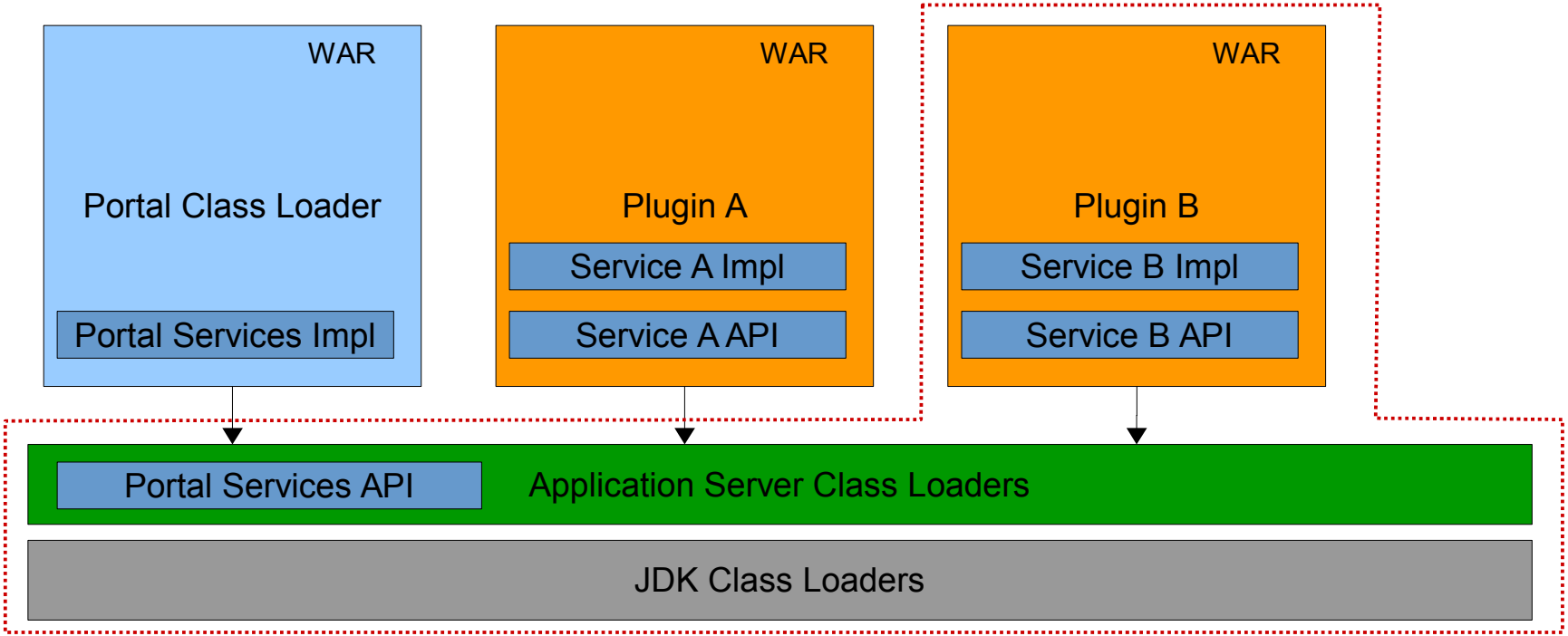
Liferay: class loading hierarchy



Liferay: class loading hierarchy



Liferay: class loading hierarchy



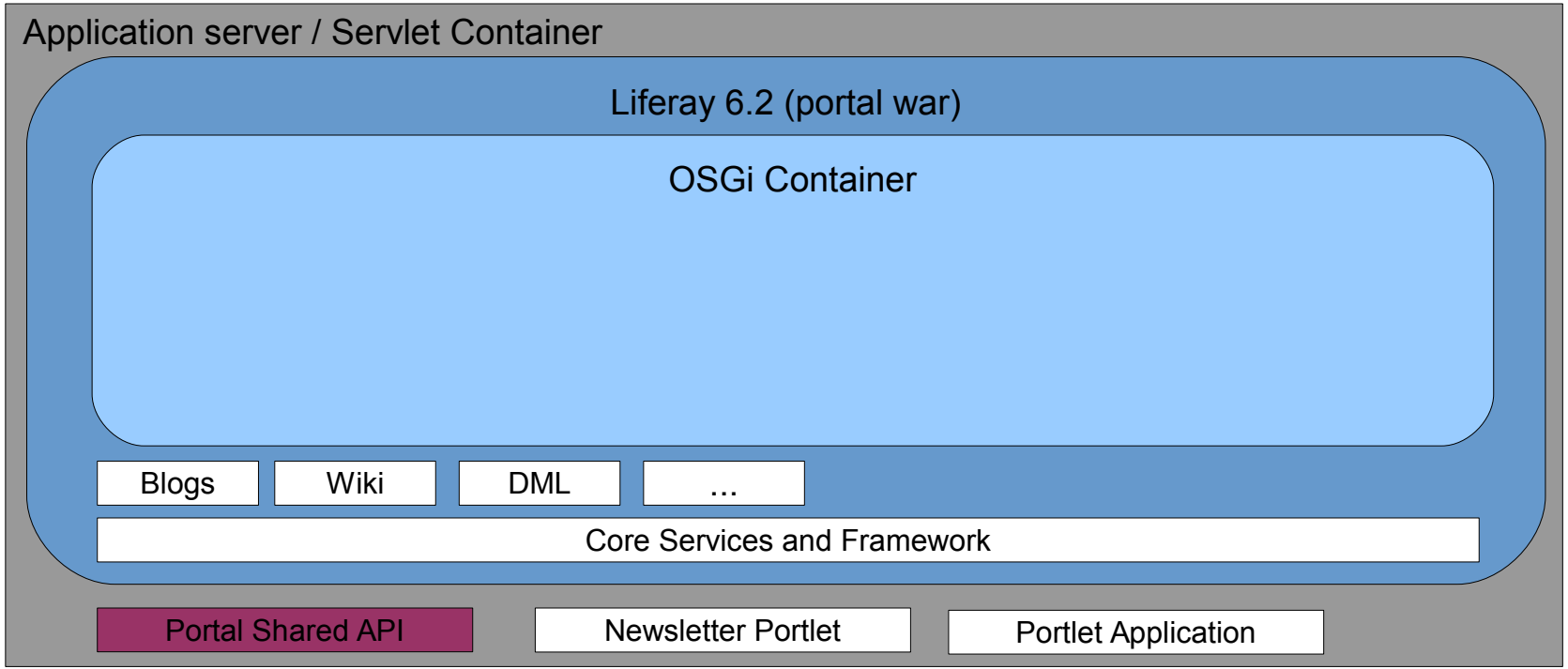
Limite dell'architettura monolitica di Liferay

- Invocazione dei servizi tra plugins / portlet
  - Nessuna soluzione standard
  - Difficoltà nella gestione del workaround
- Un singolo artefatto
  - Siamo a circa 230MB
  - Non è possibile installare solo quello davvero necessario
  - Non è possibile isolare le funzionalità di portale
- Il deploy dipende dall'Application Server
- La scalabilità è su singola dimensione
- Marketplace: una JSP può essere “hookata” da un solo plugin

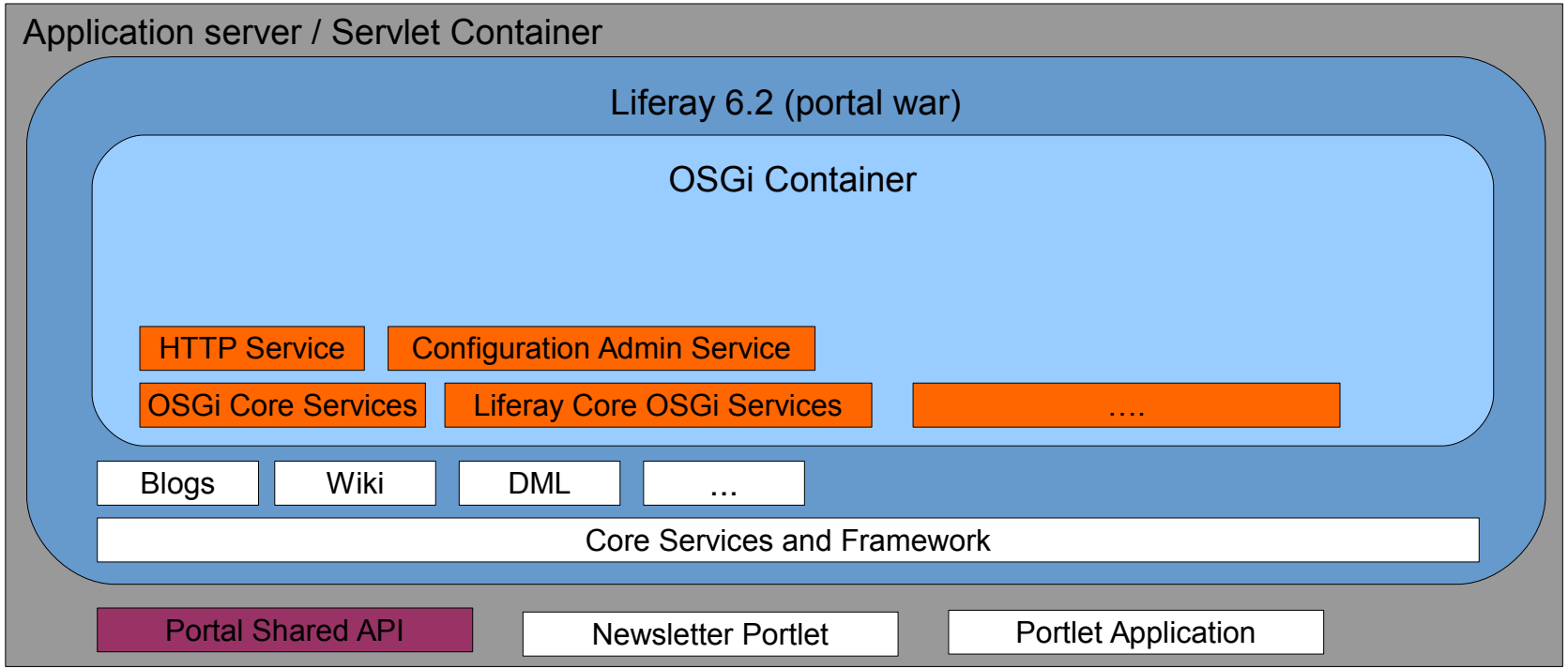
Modularità: tante aspettative

- Versionamento dei plugin indipendente dalla versione di Liferay Portal
  - OSGi semantic versioning
- Business Agility
  - Rilasci più frequenti di nuove funzionalità o migliorie
  - Processo di sviluppo più semplice e disaccoppiato
- Relazioni basate su contratti
  - Loose coupling
- Meccanismo di estensione dinamica

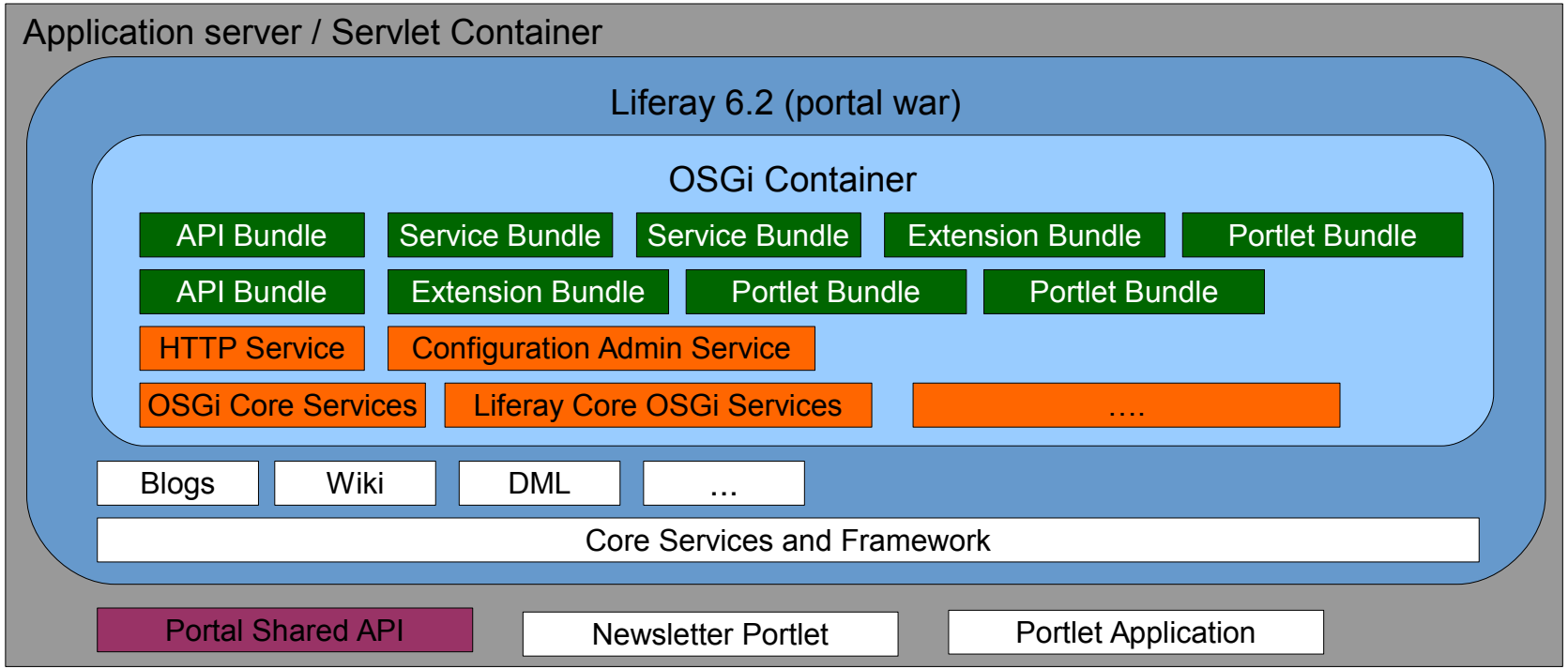
Liferay: da monolitico a modulare



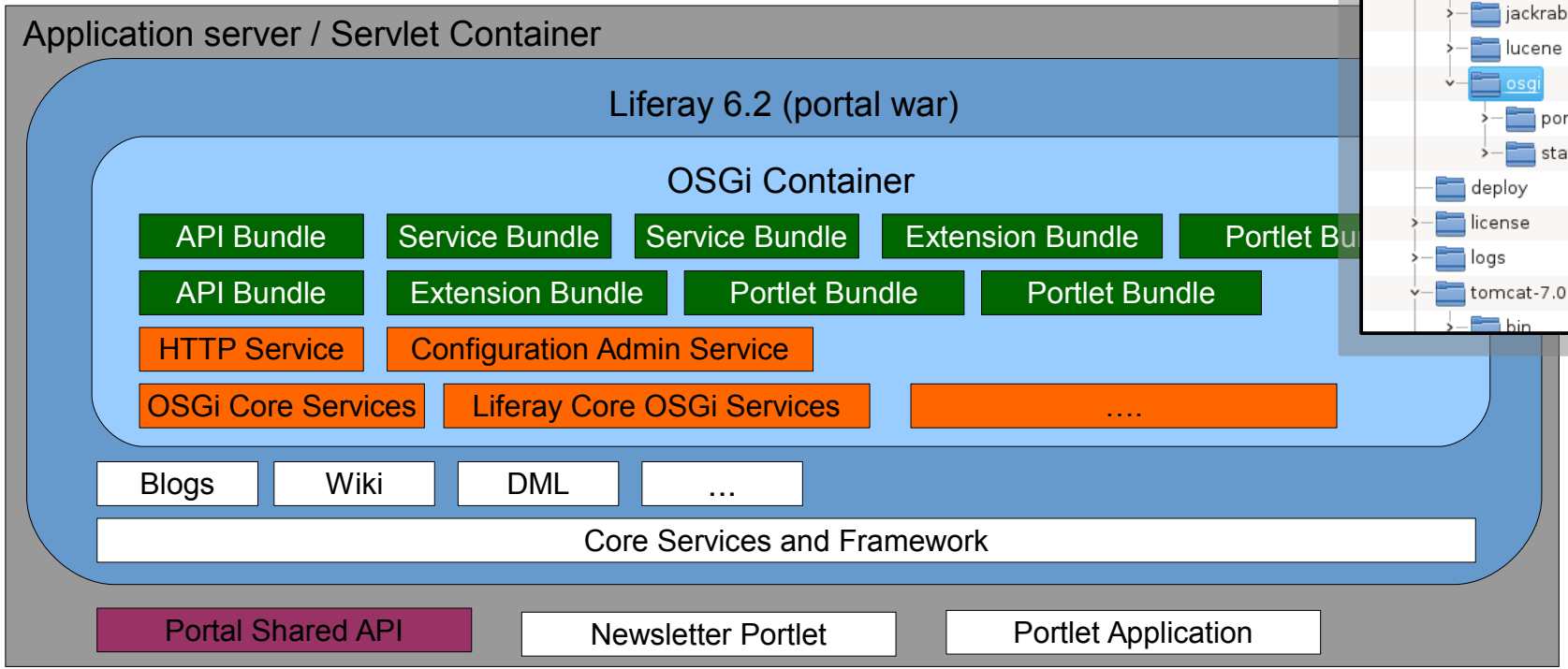
Liferay: da monolitico a modulare

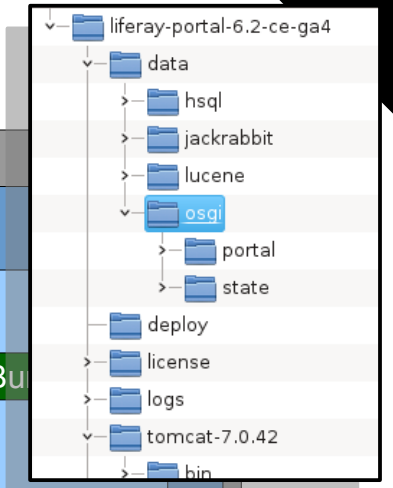


Liferay: da monolitico a modulare

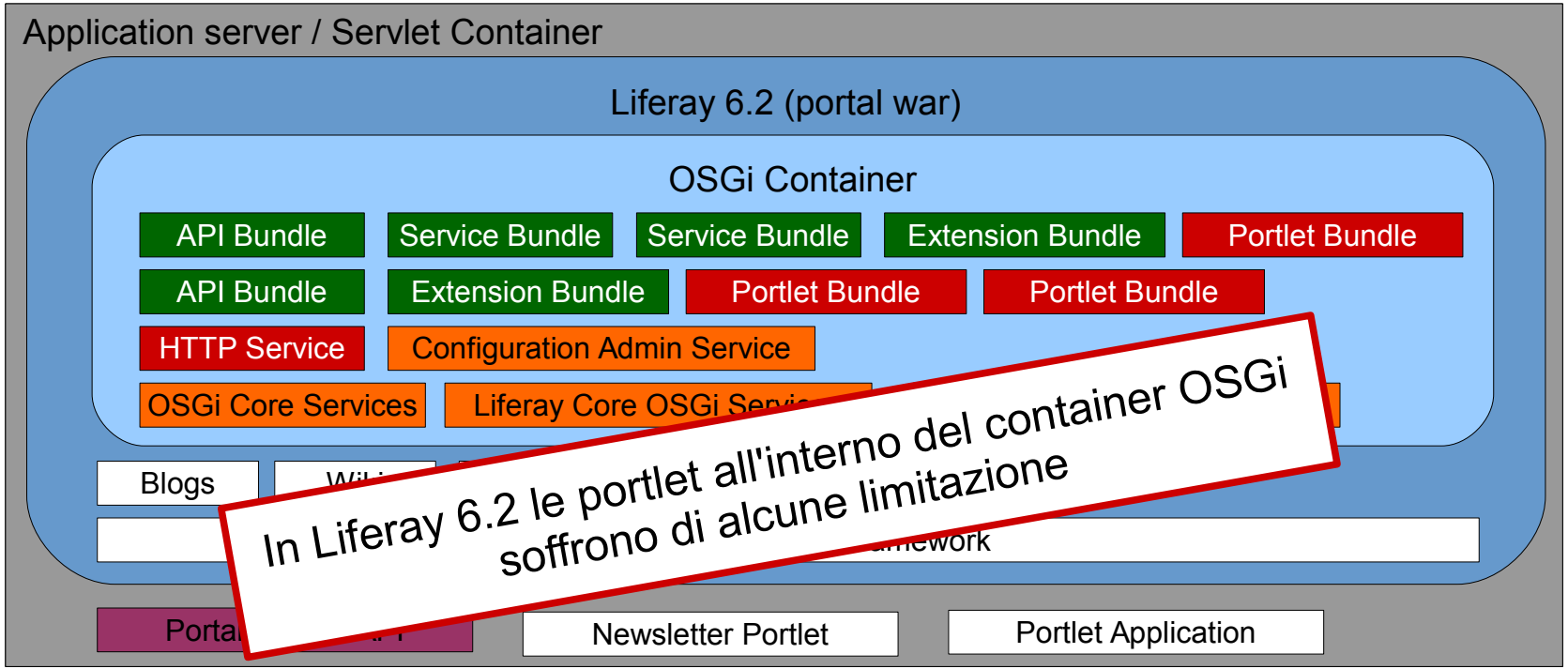


Liferay: da monolitico a modulare

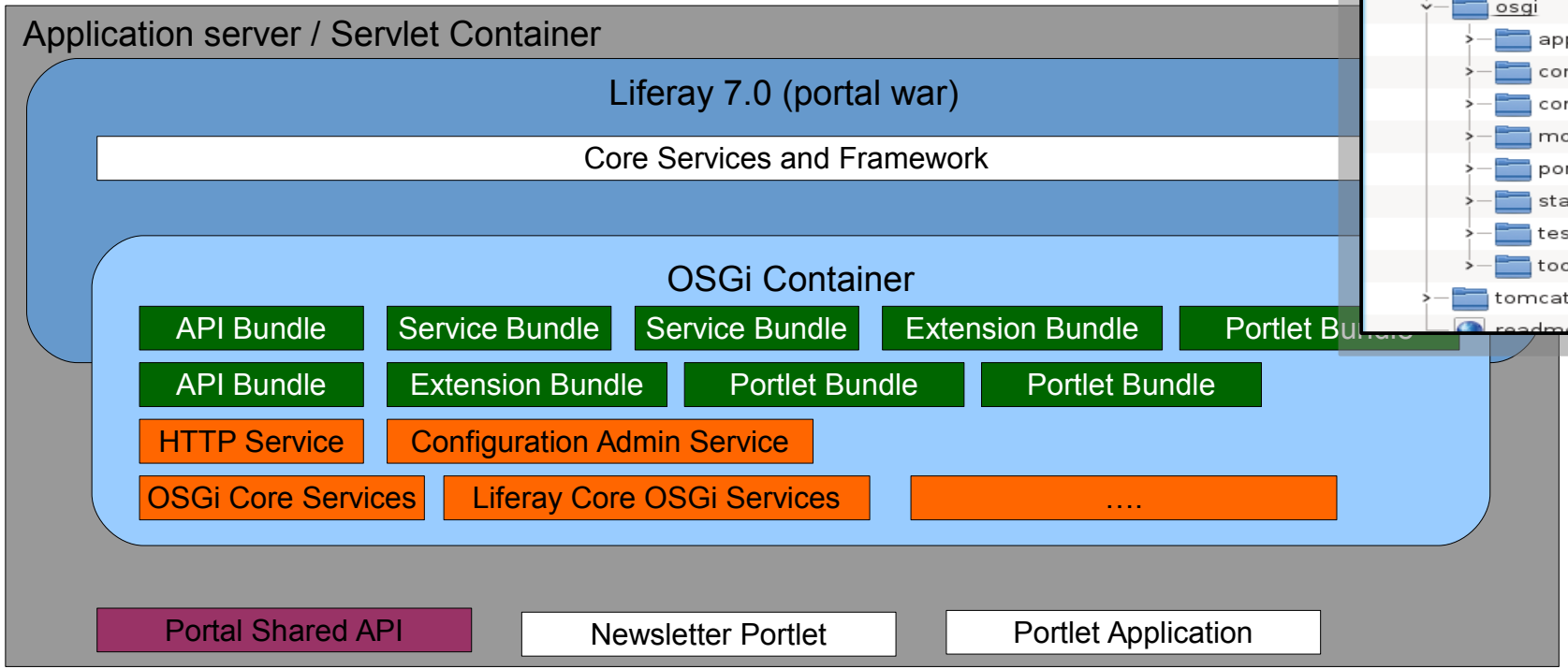


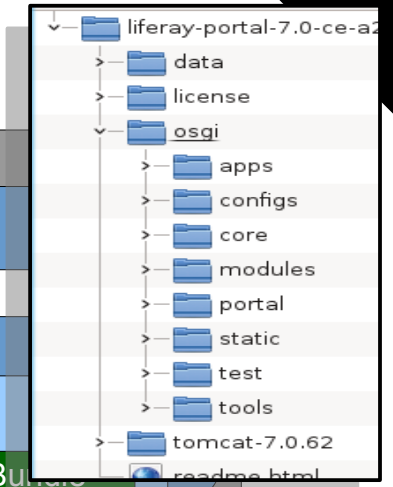


Liferay: da monolitico a modulare

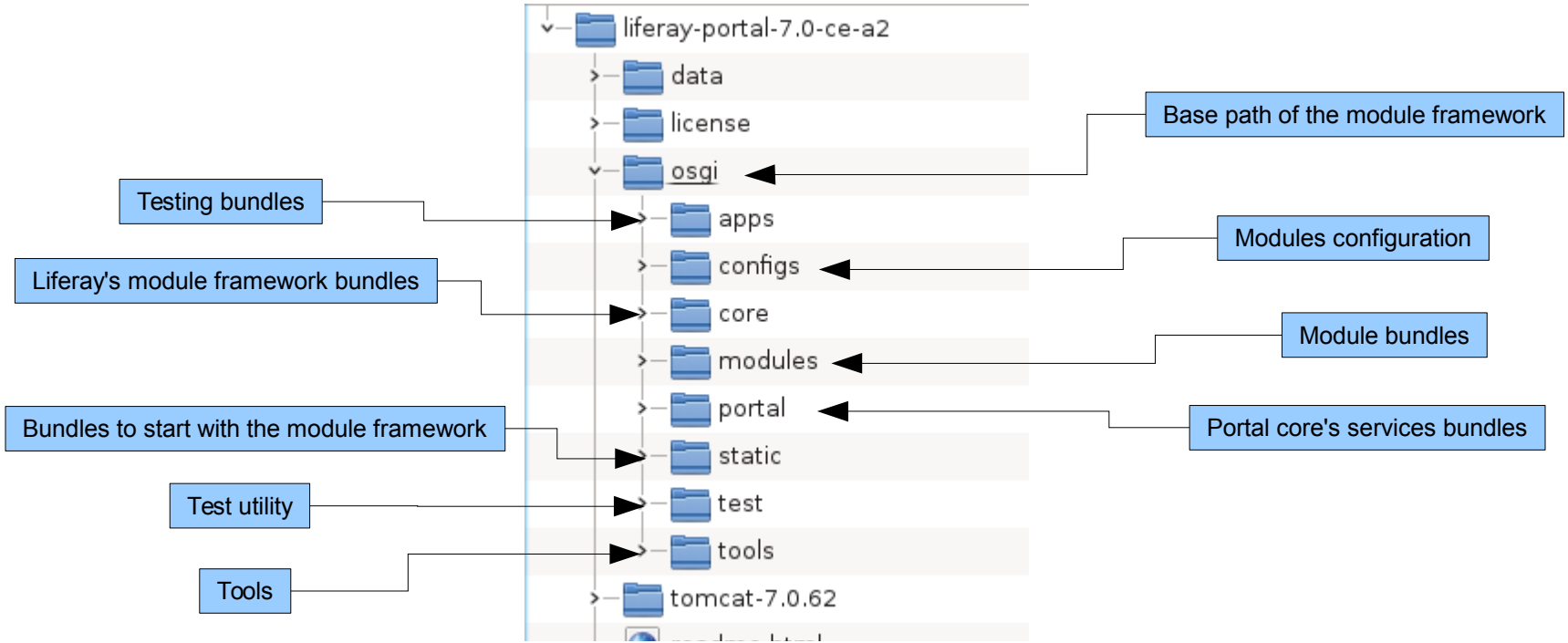


Liferay: da monolitico a modulare





Liferay module framework



Liferay module framework

- Liferay 7 gestisce totalmente la fase di deploy. L'application Server non è coinvolto
- Gestione dinamica del ciclo di vita del modulo
- I moduli Liferay sono versionati e dichiarano le dipendenze in modo esplicito

Building modules with OSGi

- Liferay supporta diversi framework
  - OSGi API
  - Blueprint
  - iPOJO
  - OSGi Declarative Services
- La tecnologia raccomandata è **Declarative Services**

Portlets con i Declarative Services

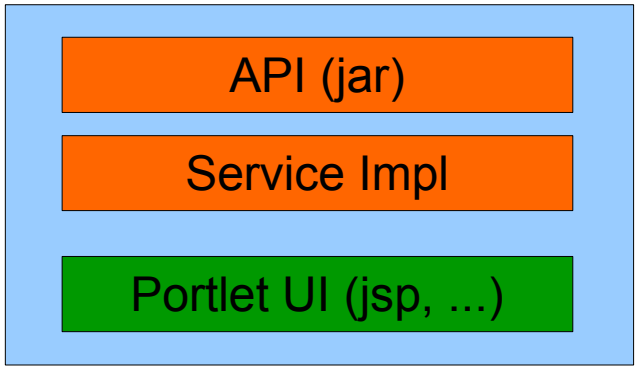
- Liferay 6.2 e precedenti
  - portlet.xml
  - liferay-portlet.xml
  - liferay-display.xml
- Liferay 7.0: annotazioni

```
@Component(
    immediate = true, // starts immediately when import package are resolved
    property = {
        "com.liferay.portlet.display-category=category.sample",
        "com.liferay.portlet.instanceable=true",
        "javax.portlet.display-name=Sample OSGi DS Portlet",
        "javax.portlet.securoty-role-ref=power-user,user"
    },
    service = Portlet.class  // Expose the API, register as Portlet
)
public class SamplePortlet extends GenericPortlet {
```

Portlets Service Builder

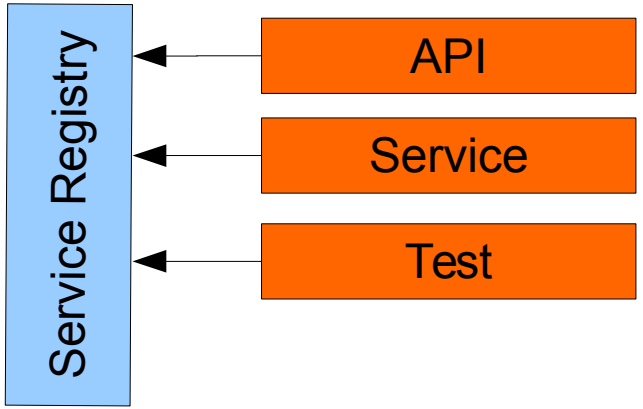
Liferay 6.2 e precedenti

Portlet Application (war)



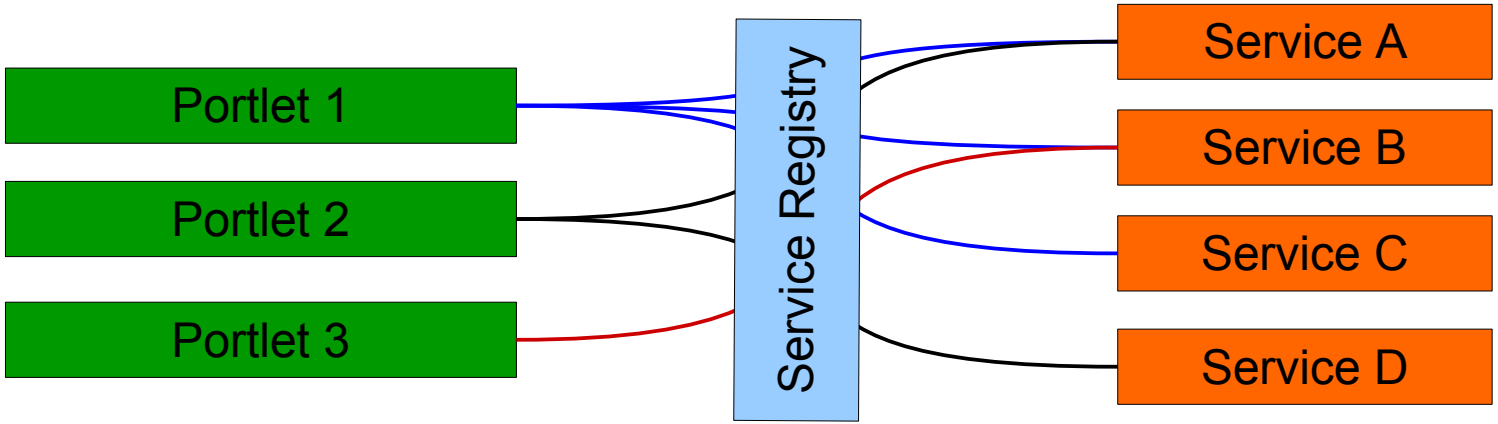
Liferay 7.0

Bundles (jar)



Condividere servizi tra plugins

Gratis!



Estendere Servizi Liferay

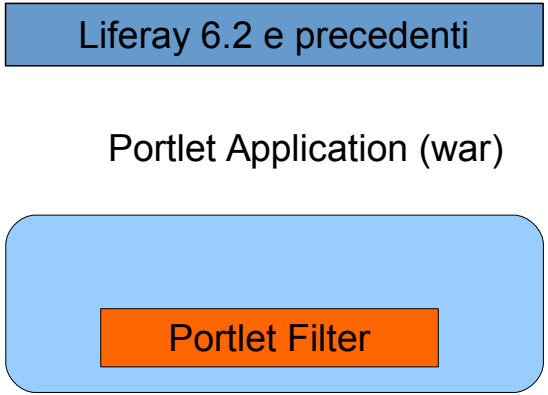
```
@Component(
    immediate = true,
    property = {},
    service = ServiceWrapper.class  // Expose the API, register the hok as ServiceWrapper
)
public class UserLoginTrackerServiceHook extends ServiceWrapper {


    @Override
    public User updateLastLogin(long userId, String loginIP) throws PortalException {

        _log.info("UserId '" + userId + "'");

        return super.updateLastLogin(userId, loginIP);
    }
```

Estendere Servizi core

Si sfrutta il ranking delle implementazioni. La nostra deve avere un punteggio maggiore rispetto al servizio che si vuole estendere

```
property={“service.ranking:Integer=100”}
```

Portlet Filter



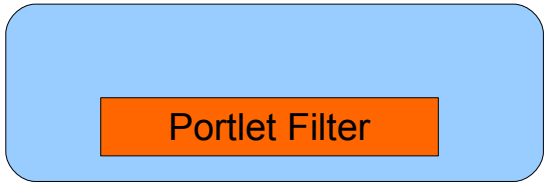

```
<filter>
    <filter-name>LoginPortletFilter</filter-name>
    <filter-class>it.smc.test.portlet.filter.LoginPortletFilter</filter-class>
    <lifecycle>RENDER_PHASE</lifecycle>
</filter>
```

- Molto complesso da estendere

Portlet Filter

Liferay 6.2 e precedenti

Portlet Application (war)




```
<filter>
    <filter-name>LoginPortletFilter</filter-name>
    <filter-class>it.smc.test.portlet.filter.LoginPortletFilter</filter-class>
    <lifecycle>RENDER_PHASE</lifecycle>
</filter>
```

Liferay 7.0

Bundles (jar)

```
@Component(
    immediate = true,
    property = {
        "javax.portlet.name=it_smc_test_web_portlet_LoginPortlet"
    },
    service = PortletFilter.class
)
public class LoginPortletFilter implements RenderFilter {

    @Override
    public void doFilter(
            RenderRequest request, RenderResponse response, FilterChain chain)
        throws IOException, PortalException {

        // todo
    }
```

UI Customization & Extensions

Nuova taglib per includere dinamicamente funzionalità

```
<liferay-util:dynamic-include key=”com.liferay.frontend.editors.web” />
```

Specializzare la configurazione dei componenti UI

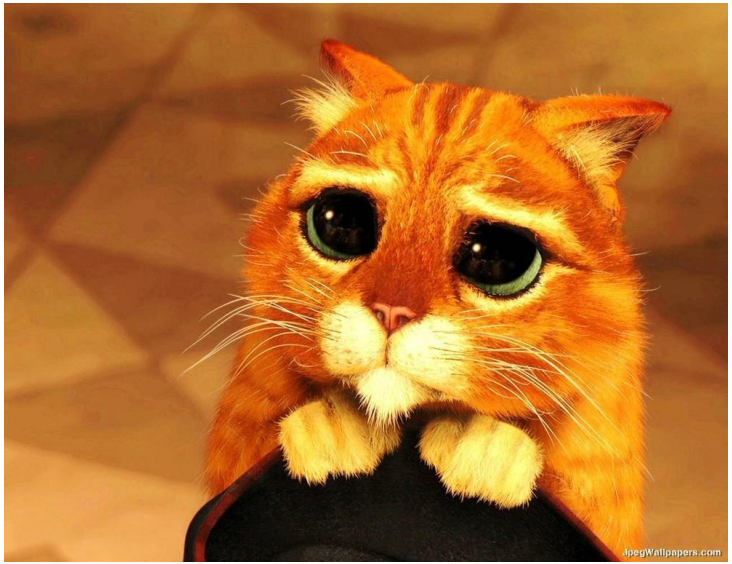
```
@Component(
    property = {
        "editor.config.key=contentEditor",
        "editor.name=alloyeditor",
        "editor.name=ckeditor",
        "javax.portlet.name=33", "javax.portlet.name=my-custom-portlet-id",
        "service.ranking:Integer=100"
    },
    service = EditorConfigContributor.class
)
public class MyEditorAddon extends BaseEditorConfigContributor {
```

Conclusione

- Usa un approccio graduale: una montagna si scala un passo alla volta!
  - Prima porti i tuoi plugin sulla 7.0 as-is
  - Poi li trasformi completamente in bundle
- Ogni @Component può essere rimpiazzato da una tua versione
- Approccio a componenti riutilizzabili (Taglib, Item Selector, Portlet decorator, ...)

- Pensa modulare, ma senza esagerare!

Come? Niente codice?



OSGi in Liferay

OSGi in Liferay 6.2 e 7.0

Mauro Mariuzzo
Software Architect – SMC Treviso

mauro.mariuzzo@smc.it
@MariuzzoMauro

Credits

Liferay Portal modern architecturing and development – Rafik HARABI

Leveraging OSGi to create extensible plugins for Liferay 6.2 – Julio Camarero & Milen Dyankov

Immagini da:

www.cupofbrain.it