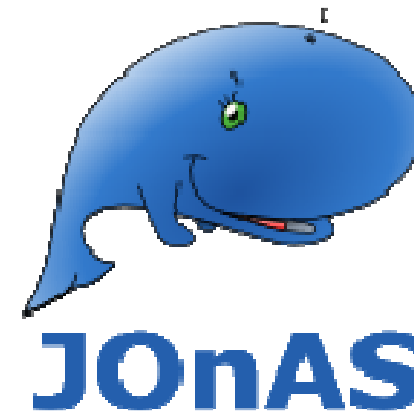




OW2
Consortium

JOnAS 5



Francois.Exertier@bull.net

JOnAS: Enterprise Class Java EE Application Server



- Java EE Compliant (Sun Certification) *[J2EE 1.4 for now]*
- Scalability and Availability
 - Clustering (HTTP, RMI, DB), Failover (HTTP/EJB Sessions replication)
 - Optimization mechanisms (pooling, caching, ...)
- Enterprise Integration
 - Multi-tier Infrastructure
 - Apache, LDAP, DBMSs, J2EE CA connectors to ERPs, mainframes
 - Web Services
- Integrated Development Environments
 - Eclipse plug-ins
- Administration
 - Web Console, script commands, API (JMX, JSR77), JASMINe
- Professional Support Offer (Bull)
- Operational, critical deployment
 - Banking, telecom, administration...



JOnAS Development Community



■ Bull

- Leader
- Support and services based on JOnAS



■ France Telecom

- Intensive operational use of JOnAS
- Clustering (validation, migration)



■ Peking University, CVICSE

- Services architecture, Clustering, Web Services, Management



■ INRIA

- Self Management



■ UPM

- Clustering



■ LIFL

- Administration/monitoring, deployment



■ UNIFOR

- Clustering



■ UJF

- OSGi



JOnAS 5



- New architecture, services oriented

- OSGi
- Code refactoring

- Java EE 5

- EJB3 (EasyBeans OW2 Project)
- Tomcat 6
- JSF, JSTL...

- Advanced Administration

- auto-adaptability, autonomous behaviour
- JASMINe OW2 Project

- Clustering Enhancements

- CMI V2 (EJB clustering)

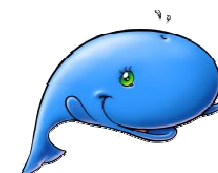
- Misc

- Deployment
- Maven2

This presentation
JOnAS 5.0
15.00 – 15.30

JASMINe 1.0
12.00 – 12.30

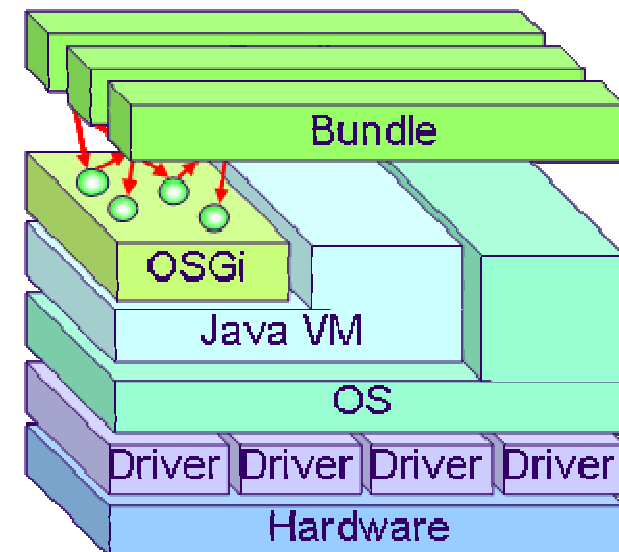
EJB3 Clustering...
16.00 – 16.30



What Is OSGi



- OSGi Alliance Consortium founded in March 1999
- Objective
 - Create open specifications for delivering administrated services through a network
- Define
 - A common platform (framework)
 - Services deployment
 - Services execution and administration
 - A set of based services:
 - Log Service, Http Service...
 - A device access specification
 - A deployment unit, a bundle
- First Applications
 - Automotive, home...
- Recent trends
 - Server side, middleware



OSGi Main Concepts



■ Framework:

- Bundles execution environment
 - Felix (Apache), SMF, Knopperfish, Equinox
- Event notification



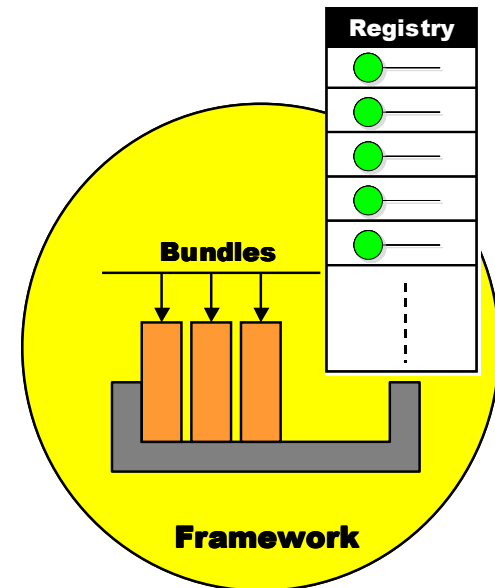
■ Bundles:

- Services diffusion and deployment unit



■ Services:

- Java Object implementing a well define contract



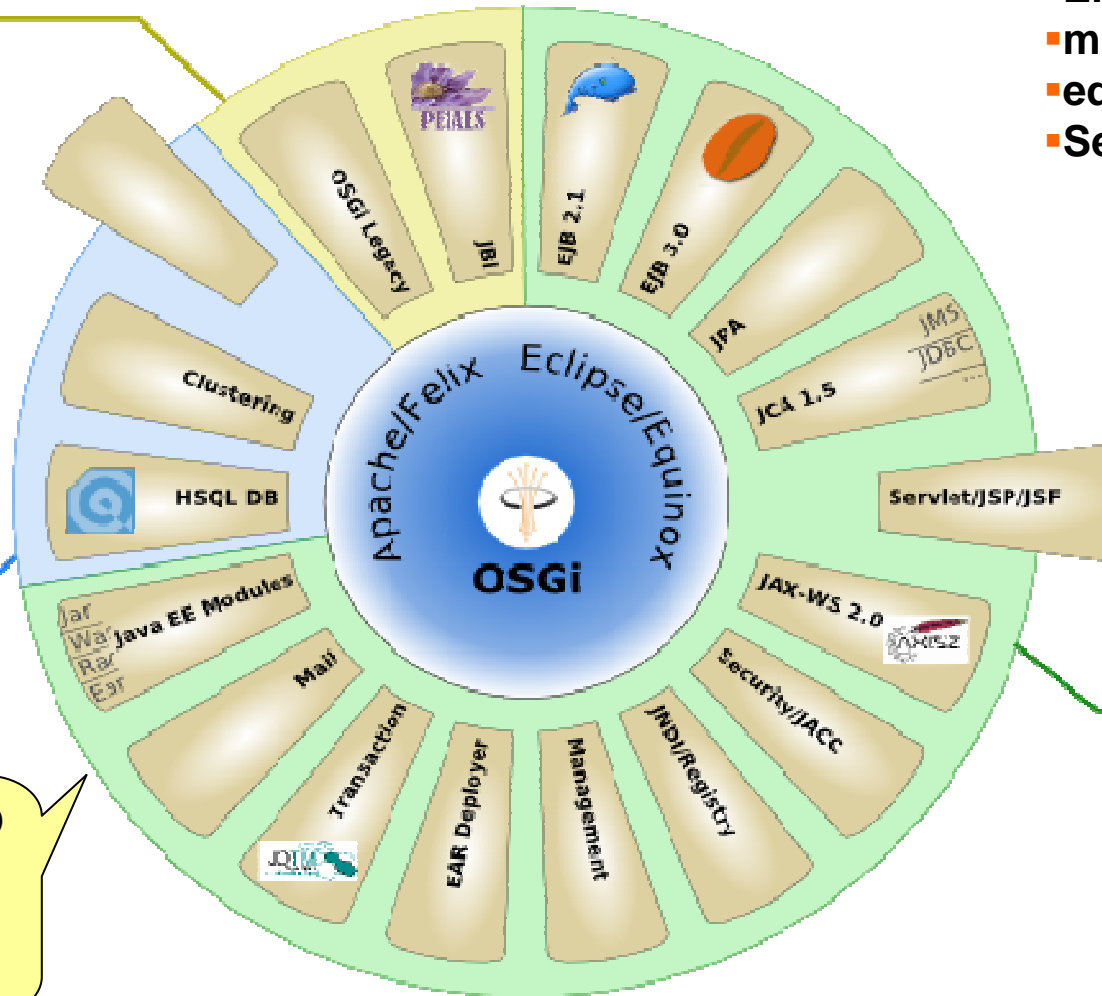
JOnAS 5 / OSGi

- Target :
 - Embedded,
 - multi-paradigm,
 - edge...
 - Self Mgt

Extensions

Add-ons

Use of IPOJO
OSGI
Component
Model



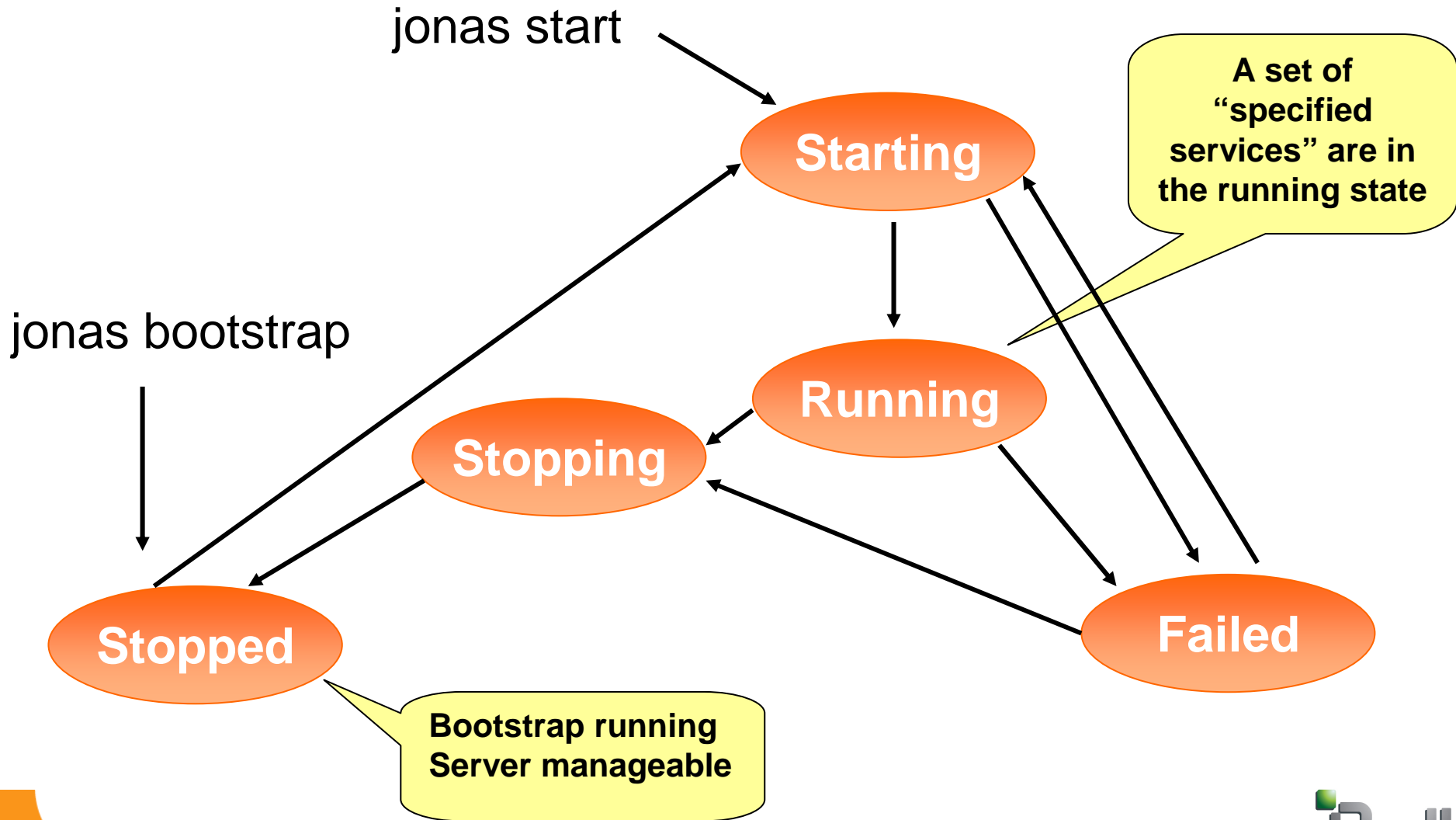
Java EE

- Dynamic Configuration and re-configuration of servers
 - Start/stop services at runtime
 - On demand incremental services delivery
- Modularity
 - Services delivered in « bundles »
 - Code readability
 - Reduced memory footprint
 - Explicit Dependencies
- OSGi other add ons
 - Services lifecycle management (incl. versions)
 - Services dependencies management (OBR)
 - Dynamic class loading (flat class loaders)
 - OSGi world accessibility (RFID, probes ...)



- Enhance Server State Control
 - control JOnAS services lifecycle: be able to start/stop/restart a service, be aware of its state
 - base server state management on JSR77 State Management specification
 - define and implement correlation between services states and server state
- Provide more flexibility and dynamicity in services control
 - Control services individually, at runtime (without stopping the server, as in JOnAS 4)

Server Lifecycle [1/2]



Server Lifecycle [2/2]



- Server Bootstrap
 - Felix
 - jmx-remote connector
 - some MBeans like the J2EEServer, J2EEDomain, JavaVM
 - Bundles
 - CAROL registry service (connector registration)
 - JMX service

- Running state / critical services (future)
 - Some services could be specified as non critical
 - Failure of such services will not prevent the server to reach the running state
 - So running state could be annotated with some info related to a degraded behaviour
 - a *non critical* service could not be started
 - an applicative module of the deploy directory could not be loaded



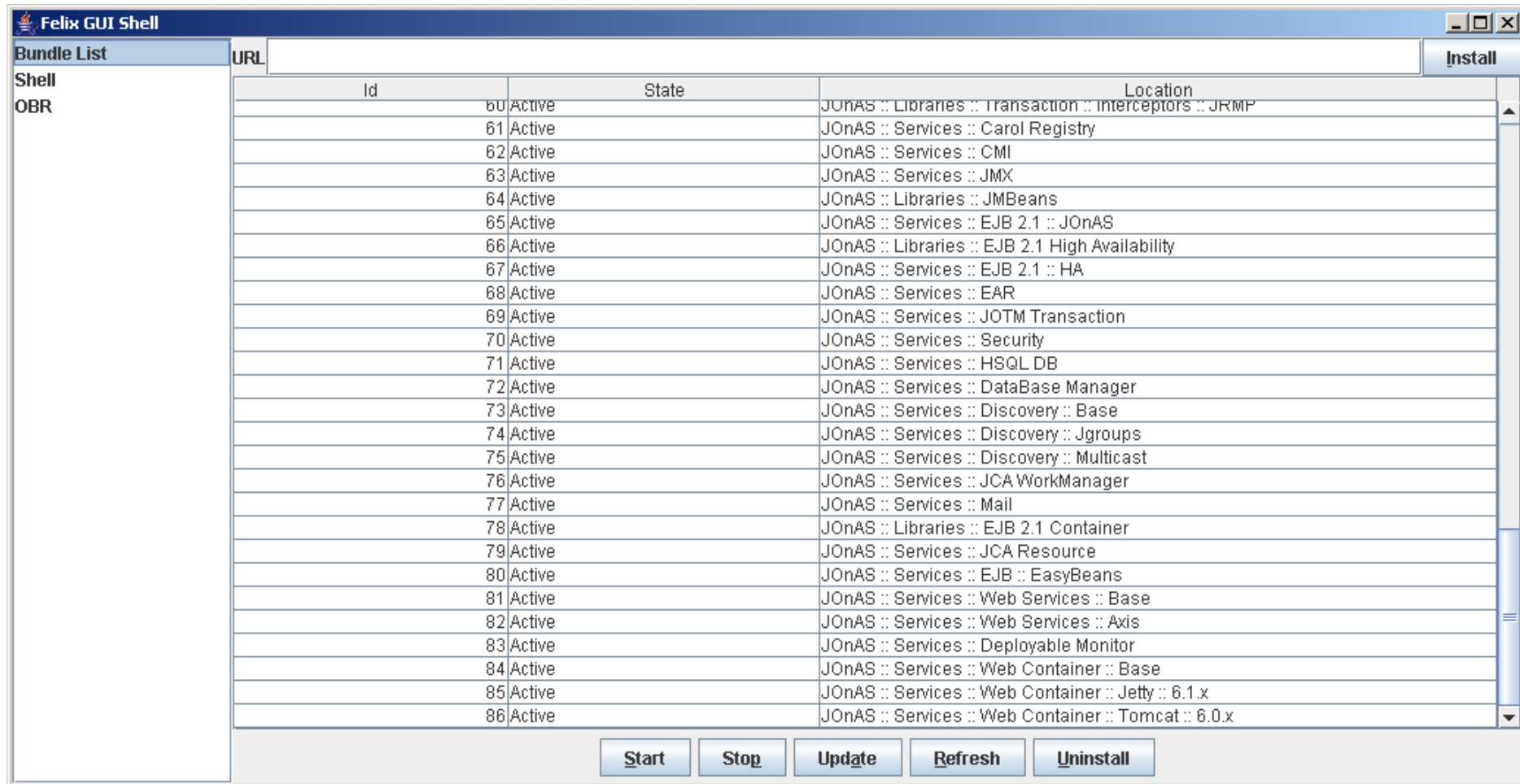
JOnAS 5 commands



- **jonas start**
 - services defined in `jonas.properties` configuration file are launched (if not already running, JVM and bootstrap are launched)
- **jonas bootstrap**
 - only bootstrap services ('registry' and 'jmx') are launched
- **jonas stop**
 - All services, except the bootstrap ones, are stopped
- **jonas halt**
 - Stop all services and exit JVM
- **jonas admin**
 - Use admin command based on JMX remote to administer the server



View of JOnAS bundles



Id	State	Location
60	Active	JOnAS :: Libraries :: Transaction :: interceptors :: JRMF
61	Active	JOnAS :: Services :: Carol Registry
62	Active	JOnAS :: Services :: CMI
63	Active	JOnAS :: Services :: JMX
64	Active	JOnAS :: Libraries :: JMBEANS
65	Active	JOnAS :: Services :: EJB 2.1 :: JOnAS
66	Active	JOnAS :: Libraries :: EJB 2.1 High Availability
67	Active	JOnAS :: Services :: EJB 2.1 :: HA
68	Active	JOnAS :: Services :: EAR
69	Active	JOnAS :: Services :: JOTM Transaction
70	Active	JOnAS :: Services :: Security
71	Active	JOnAS :: Services :: HSQL DB
72	Active	JOnAS :: Services :: DataBase Manager
73	Active	JOnAS :: Services :: Discovery :: Base
74	Active	JOnAS :: Services :: Discovery :: Jgroups
75	Active	JOnAS :: Services :: Discovery :: Multicast
76	Active	JOnAS :: Services :: JCA WorkManager
77	Active	JOnAS :: Services :: Mail
78	Active	JOnAS :: Libraries :: EJB 2.1 Container
79	Active	JOnAS :: Services :: JCA Resource
80	Active	JOnAS :: Services :: EJB :: EasyBeans
81	Active	JOnAS :: Services :: Web Services :: Base
82	Active	JOnAS :: Services :: Web Services :: Axis
83	Active	JOnAS :: Services :: Deployable Monitor
84	Active	JOnAS :: Services :: Web Container :: Base
85	Active	JOnAS :: Services :: Web Container :: Jetty :: 6.1.x
86	Active	JOnAS :: Services :: Web Container :: Tomcat :: 6.0.x

Services Lifecycle [future]

- Ready to be started.
- Manageable (MBean registered in the JMX service)
- OSGi bundle(s) stopped
- Bundles installed and resolved

**Installed/
Stopped**

**OSGi bundle(s)
started, in the
ACTIVE state,
and IPOJO
VALID**

Failed

Running

**Some OSGi bundles
not able to start, or
have stopped,
INVALID IPOJO State**

Services related commands

- Future potential commands
 - Installation (= bundle installation). Service is manageable
 - **jonas install -service sv1**
 - Un-installation (= bundle uninstall). Service is no more manageable.
 - **jonas uninstall - service sv1**
 - Services start
 - **jonas start -service sv1[,sv2]**
 - Services stop
 - **jonas stop -service sv1[,sv2]**
 - Service update
 - **jonas update -service sv1**
 - Service status
 - **jonas status -service sv1**

JOnAS 5: On Demand Services

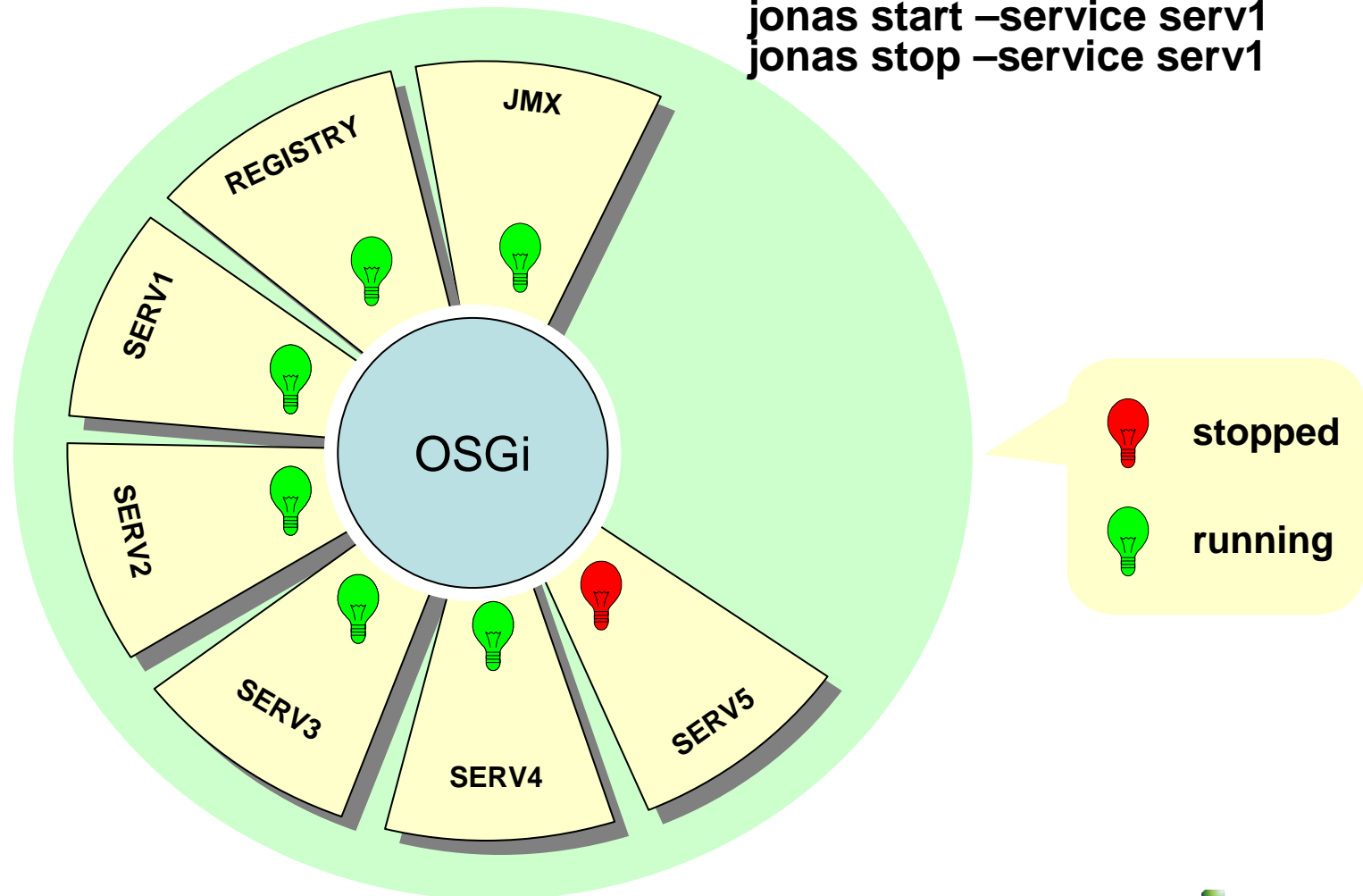


- Service installation
 - Install all dependent bundles
- Service start
 - Start all dependent services
- Service stop
 - Stop all dependent services (if not used by third party)



On Demand Services

```
jonas install - service serv1  
jonas start -service serv1  
jonas stop -service serv1
```



serv1 depends on serv2, serv3, serv4

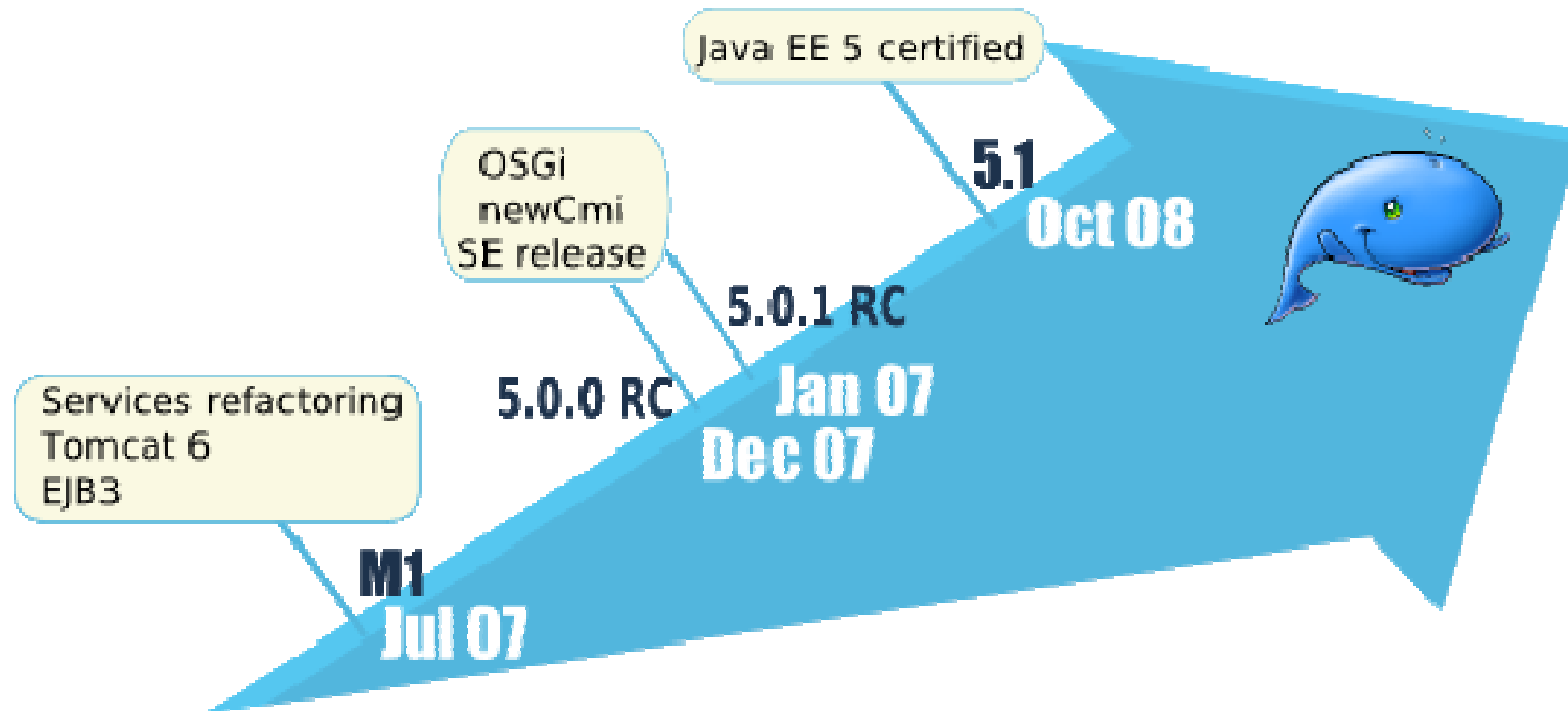
Other JOnAS 5 new features



- Java EE 5 features
 - EJB3
 - JSF, JSTL
- Clustering
 - CMI V2
- Deployment: a new service 'depmonitor'
 - deployment of all module types and J2EE applications present in JONAS_BASE/deploy directory at server start time
 - may be configured in 'development' mode to allow
 - Dynamic undeploy/re-deploy of modified files
 - Dynamic deploy of files dropped into the directory
 - Dynamic undeploy of files removed from the directory
 - the old layout based on ejbjars, webapps, rars and apps directories, is still supported
- jonasAdmin console enhancements
 - Cluster/CMI management



JOnAS 5 ROADMAP



EasyBeans: EJB3 Container (core)



- OW2 Project: EasyBeans, **1.0 RC2**
 - Lead Bull
 - Lightweight and modular container
 - www.easybeans.org / www.easybeans.net
- Recent Features
 - Integrated in JOnAS 5
 - EJB clustering (Failover and Load Balancing for SLSB)
 - Access to OSGi world
- Features
 - Efficiency : New container architecture (code injection/ASM, proxy...)
 - Embeddable in JOnAS 4 (as RAR), Jetty & Tomcat
 - Standalone mode
 - Available as OSGi bundles
 - Ease of use: exploded archive, class reloading, quick deployment, easier client launching, etc.
 - Work with Hibernate EJB3, Apache OpenJPA, TopLink Essential...



EasyBeans: Ease of development



■ EJB3 specification

- Home interface no more required
- Annotations (DD optionnels)
- Interceptors
- dependencies Injection
- JPA persistence
 - POJO
 - inheritance
 - Best of CMP2/JDO/Hibernate models
 - Integration of EJB3 persistence layers (Hibernate EJB3, TopLink Essential, Xplus...)



■ EasyBeans

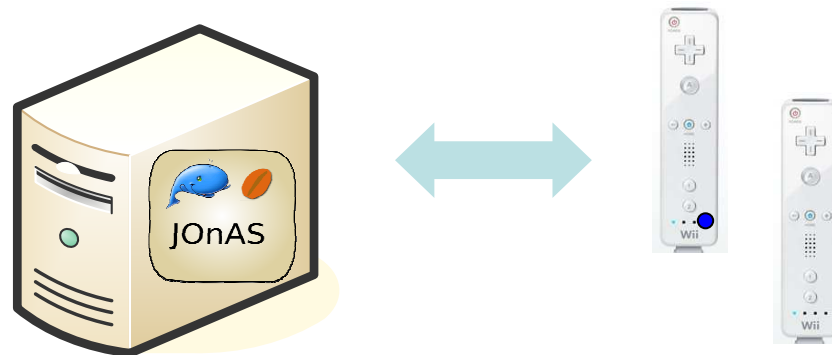
- No client side deployment (smartfactory)
- Automatic hot deployment,
- exploded archives
- Simple log API



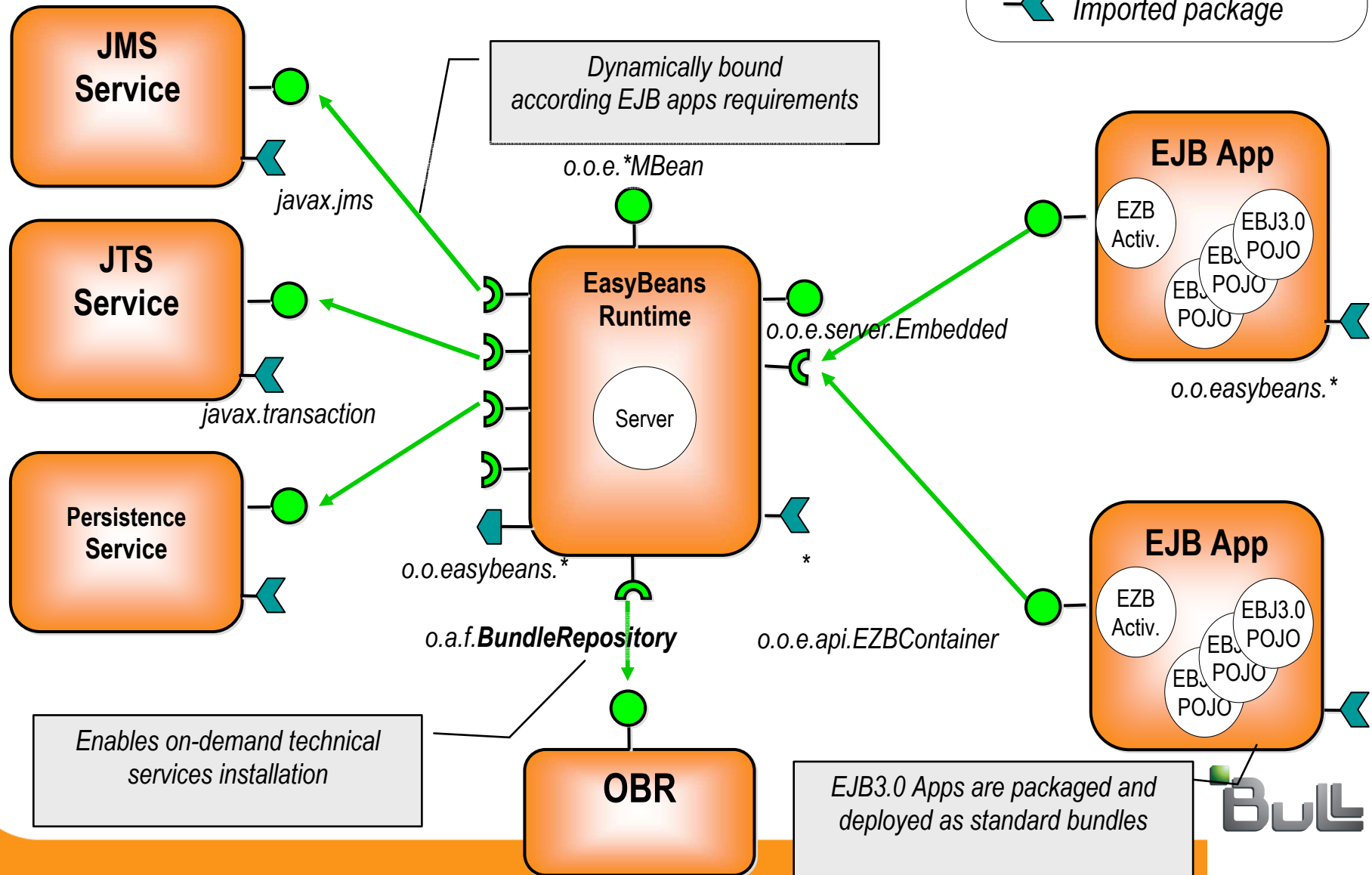
Access to OSGi World

- BundleContext injection:
 - From an EJB, it is then possible to access OSGi services (get, register...)

```
/**  
 * Bundle context that will be injected by the container  
 */  
@OSGiResource  
private BundleContext bundleContextField = null;
```



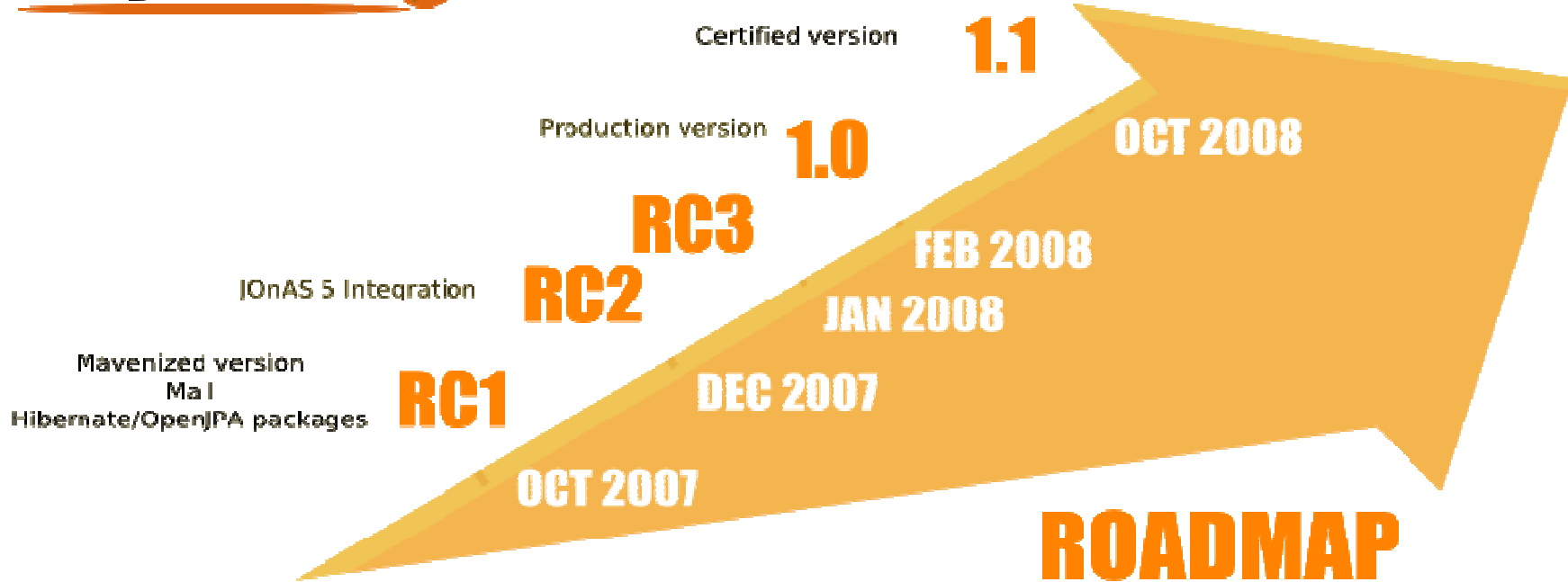
EasyBeans: OSGi bundles



EasyBeans Roadmap



EasyBeans



www.easybeans.net



The screenshot shows the EasyBeans website with the following content:

EasyBeans

Open Source & Lightweight EJB3 Container

Project Demos Documentation Downloads Support News Developers Wiki

:: What is EasyBeans ?

EasyBeans is an open-source [Enterprise Java Beans \(EJB\)](#) container hosted by the [OW2 consortium](#). The License used by EasyBeans is the [LGPL](#).

EasyBeans main goal is to ease the development of Enterprise Java Beans. It uses some new architecture design like the bytecode injection (with [ASM ObjectWeb tool](#)), [IoC](#), [POJO](#) and can be embedded in [OSGi](#) bundles or other frameworks ([Spring](#), [Eclipse](#) plugins, etc.).

It aims to provide an EJB3 container as specified in the [Java Platform Enterprise Edition \(Java EE\)](#) in its fifth version. It means that Session beans (Stateless or Stateful), Message Driven Beans (MDB) are available on EasyBeans.

The new persistence layer used by EJB 3.0 is now called Java Persistence API (or JPA). It replaces the CMP (Container Managed Persistence) model used by EJB 2.x. The default persistence provider used in EasyBeans is Hibernate Entity Manager or Apache OpenJPA but other JPA providers have been tested like for example Oracle TopLink Essentials.

Recent News

- 12.14.2007 :: 1.0 RC2 available
- 10.17.2007 :: 1.0 RC 1 Released
- 08.29.2007 :: EasyBeans On OSGi

[Read More >>](#)





```
graph TD; Applet[Applet] --> WebContainer[Web Container]; Client[Client] --> EasyBeans[EasyBeans]; WebContainer --> DB[(DB)]; EasyBeans --> DB;
```

:: Why EasyBeans ?

For a long time, EJB (Enterprise Java Beans) has been considered as a complex technology. Many developers were not using EJBs when developing Java EE applications and then use partials J2EE servers like [Apache Tomcat](#). With the new Java EE 5 version, and EJB 3.0 which is part of this specification, one of the focus was on EoD (Ease of Development) So, Ease of Development was also the main item used to design the EJB3 container of [JOnAS application server](#). All is done in the JOnAS EJB3 container to develop EJBs in an easy way.

Some people said that the EJB container in JOnAS was too tied to the application server. An answer to these people was done by extracting the EJB container of the JOnAS core and making it available as a lightweight container. Also, this allows to use this JOnAS EJB3 container in other application servers, in standalone mode, bundle it for OSGi platform, in [JUnit/TestNG](#), etc. This is why JOnAS EJB3 container was named EasyBeans and has its own web site.

:: Link with JOnAS J2EE application server.

- Open Source
 - Worldwide development and users community
- Enterprise Class Java EE Application Server
 - Java EE Certified, Clustering, High Level Administration, Enterprise Integration, IDE
- EasyBeans  EJB3 lightweight container
- OSGi based Architecture
 - Dynamic [Re-]Configuration
 - On demand services
 - Modularity
 - -> *Embedded, multi-paradigm, edge, self-managed application servers*
- JASMINE 
 - Intelligent SOA platform Management (JOnAS clusters design, configuration, deployment, monitoring)

