# Spring and Java EE

## Snowdrop, Spring for JBoss AS

JBoss QE Associate, Red Hat

Karel Piwko

May 4, 2011

# Agenda

1. **What is Snowdrop?**
2. **How to use Snowdrop?**
   - Spring deployer
   - Snowdrop VFS
   - Persistence
   - Messaging
   - JMX
3. **Demo application**

Section 1

# What is Snowdrop?

redhat.

# Spring and Java EE

## Do you need Java EE with Spring?

- Spring is a light-weight container
- Spring allows you to mimic Java EE in a non-compliant container

## What Spring provides for Java EE?

- EJB interceptor for injecting `@Autowired` beans
- JMS producer and consumer templates
- JMX MBean exposing, registration and proxy
- JCA access in a "Spring way"
- `jee` namespace (`jndi-lookup`, `environment`, `local-slsb`, `remote-slsb`)
- JSR-303 (Bean validation), JSR-330 (Dependency injection)
- ...

redhat.

# What is Snowdrop?

**Pick the correct answer**

(a) An extension to Spring Framework classes to make Spring behave correctly with JBoss AS

(b) JBoss AS specific component for deploying and running Spring based applications

(c) Genus of plants flowering usually before vernal equinox

Property of Hermes, http://goldenagegardens.blogspot.com/

redhat.

# What is Snowdrop?

## Pick the correct answer

(a) An extension to Spring Framework classes to make Spring behave correctly with JBoss AS ✓

(b) JBoss AS specific component for deploying and running Spring based applications ✓

(c) Genus of plants flowering usually before vernal equinox ✓

redhat.

# Snowdrop overview

## What Snowdrop provides?

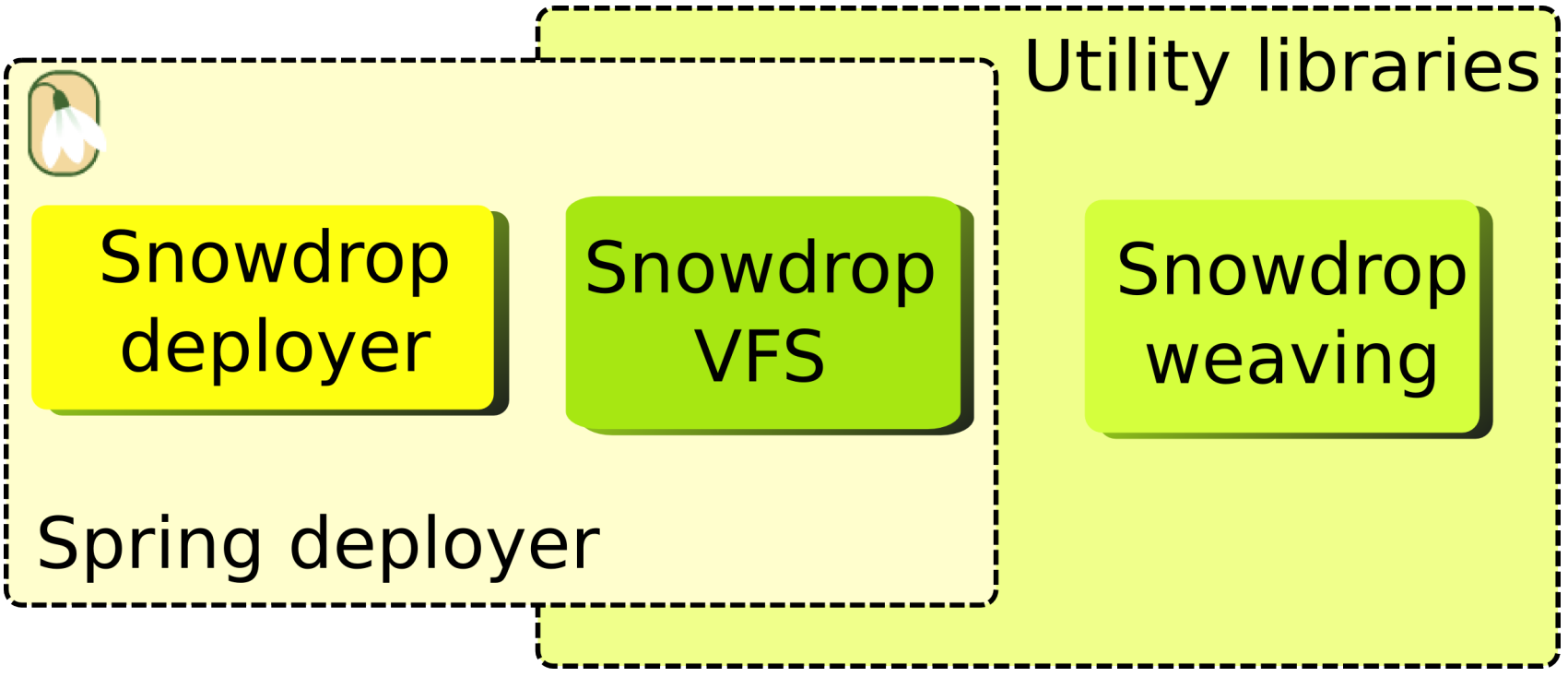
- Bootstrapping and registering application contexts for easier integration with Java EE
- Improved resource scanning for EARs and JBoss VFS
- Simplified configuration for JBoss AS related Spring beans via namespace

## What environment Snowdrop targets?

- JBoss AS 5.x
- Either Spring 2.5.x or 3.0.x

redhat

# Snowdrop architecture



Section 2

# How to use Snowdrop?

redhat.

# Spring deployer

## Overview

- An JBoss deployer to create Spring deployments
- Allows bootstrapping of Spring context and binding it in JNDI
- Integrates EJBs with Spring beans

## Installation

- Simply extract zip into `${jboss.home}/server/${jboss.domain}/deployers`
- Add appropriate Spring jars to the directory (beans, context, core, web)

redhat.

# A Spring deployment

## What it does?

- Registers Spring beans factory to JNDI
- `META-INF/jboss-spring.xml` file
- Supports JARs, WARs, EARs, etc.

```
<beans>
  <description>BeanFactory=(SpringDao)</description>
  ...
  <bean id="flowerDao" class="example.FlowerDaoImpl" />
</beans>
```

```
<description>BeanFactory=(AnotherApp)
    ParentBeanFactory=(MyApp)</description>
```

redhat.

# Injection of Spring beans

## What it does?

- Injects Spring beans from registered Bean Factory

```
@Stateless
@Interceptors(SpringLifecycleInterceptor.class)
public class FlowerServiceImpl implements FlowerService
{
   @Spring(bean = "flowerDao", jndiName = "SpringDao")
   private FlowerDao flowerDao;

   ...
}
```

redhat.

# Injection of EJBs into Spring beans

## Overview

- Supported for `@Stateless` EJBs
- Requires JNDI mapping (can be defined via `jee` namespace)

```
@Stateless
@LocalBinding(jndiBinding = "primrose/FlowerService")
public class FlowerServiceImpl implements FlowerService
{
   ...
}
```

```
@EJB(mappedName = "primrose/FlowerService")
FlowerService flowerService;
```

redhat.

# Snowdrop VFS

## Overview

- Provides resource and class path resolution for JBoss VFS
- Acts as Spring `ApplicationContext` replacement
- Packaged in `spring-vfs.jar`

**org.jboss.spring.vfs.context.VFSClassPathXmlApplicationContext**
org.springframework.context.support.ClassPathXmlApplicationContext replacement.

**org.jboss.spring.vfs.context.VFSXmlWebApplicationContext**
org.springframework.web.context.support.XmlWebApplicationContext replacement.


Property of Tara Dawley, http://taradawley.wordpress.com/

redhat.

# JPA and JTA

## Configuration of persistence unit

- Does not really differ from any Java EE application

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
 version="1.0">
  <persistence-unit name="primrosePU"
     transaction-type="JTA">
    <jta-data-source>java:/PrimroseDS</jta-data-source>
    <properties>
       <property name="jboss.entity.manager.jndi.name"
          value="java:/primrose/em" />
       <property name="hibernate.show_sql" value="true" />
    </properties>
  </persistence-unit>
</persistence>
```

redhat.

# JPA and JTA cont'd

## Creating `EntityManager` bean

- Using default Spring means via `jee` namespace support
- Same approach can be used for `EntityManagerFactory`

```
<jee:jndi-lookup id="entityManager"
   jndi-name="java:/primrose/em" />
```

```
@Autowired
EntityManager em;
```

redhat.

# Enabling JTA transaction demarcation

## Creating `JtaTransactionManager` bean

- Using default Spring means via `tx` namespace support

```
<tx:jta-transaction-manager/>
```

## Transaction demarcation

- Backed by JBoss AS
  - Programmatic
  - Declarative via AOP or `@Transactional`

```
<tx:annotation-driven/>
```

Property of Jamie Williamson, http://www.topnews.in/

redhat.

# JMS POJO with JCA

## Spring bean as JMS POJO

- A POJO is bounded as listener to specific messaging destination
- Processes received message

## Setting JCA listener

```
<jms:jca-listener-container
   resource-adapter="resourceAdapter"
   acknowledge="auto"
   activation-spec-factory="activationSpecFactory">
  <jms:listener  destination="/queue/primrose"
     ref="wateringNotificationProcessor"
     method="processWateringNotification"  />
</jms:jca-listener-container>
```

redhat.

# Messaging

## Spring messaging support

- Can be both based on JMS/**JCA**
- Snowdrop contains a namespace for JCA, so does Spring
- JCA Connection Factory retrieved from JBoss AS

```
<jboss:activation-spec-factory id="activationSpecFactory" />
<jboss:resource-adapter id="resourceAdapter" />
```

redhat.

# Exposing Spring beans to JMX server

## Exposing Spring beans to JMX

- Spring allows you to export beans to JMX server
- Snowdrop allows you to easily bound MBean Server from JBoss AS to Spring

```
<jboss:mbean-server />
```

```
<bean class="org.springframework.jmx.export.MBeanExporter">
    ...
  <property name="server" ref="mbeanServer" />
</bean>
```


Property of Magic Box, http://magicbox.cz/

Section 3

# Demo application

redhat.

# Why to use Spring + Java EE

You have a Spring application and you want to enrich it with some Java EE functionality

or

You have a Java EE application and you want to include Spring functionality

or

You want to deploy your Spring application to Java EE container

redhat.

# Why not to use Spring + Java EE

## Spring

- Fragmentation due to backward compatibility
- Relying on IDE for pre-runtime check
- Need to enable features

## Java EE 6

- Concise and well defined model
- Type safety is a heart of the technology
- Features available out from the box

## Java EE 6

- Standard developed within many parties
- Multiple implementations available
- Extension SPI
- Can be tested in real environment easily

redhat.

# What was covered?

## Covered features

- Spring `jee` namespace
- Spring Deployer
- Snowdrop VFS
- Snowdrop Namespace
- JPA, JTA, JCA, JMS
- JMX

redhat

# Where to continue?

## Questions, feature requests, bug reports

- Report bugs in JIRA https://issues.jboss.org/browse/SNOWDROP
- http://www.jboss.org/snowdrop

## Sample applications

- https://github.com/kpiwko/primrose
- https://github.com/snowdrop/snowdrop-examples

## Development

- https://github.com/snowdrop/snowdrop/tree/1.2

Part of the JBUG series.

See http://community.jboss.org/groups/jbugcz

Follow us on #jbugcz


# The end.

Thanks for listening.