

# Managing your application with Apache Sling Launchpad and OSGi Installer

**Carsten Ziegeler**  
**[chiegeler@apache.org](mailto:chiegeler@apache.org)**

.adaptTo 2011 - Berlin



- Member of the ASF
  - Current PMC Chair of Apache Sling
  - Apache Sling, Felix, Portals, Incubator
- RnD Team at Adobe Research Switzerland
- Article/Book Author, Conference Speaker
- Technical Reviewer
- JSR 286 Spec Group (Portlet API 2.0)



# Agenda

- **1 Motivation**
- **2 Apache Sling Launchpad**
- **3 Apache Sling OSGi Installer**
- **4 Apache Sling Maven Launchpad Plugin**
- **5 Useful Things**
- **6 Summary**

# 1 Motivation

# Motivation I

- Modularity is key
  - Manage growing complexity
  - Support dynamic extensibility
- Embrace change – Embrace OSGi
  - Only a few concepts – easy to get started
  - Minor “overhead”
  - High coherence
- Loose Coupling
  - Modules and Services

## Motivation II

- OSGi is everywhere!
- State of the Art solution for modularity in Java
- Ongoing and evolving development
  - Cloud, Enterprise etc.
- Modules are fine, but what about?
  - Assembling an application
  - Deployment
  - Provisioning
  - Development

- Apache Sling provides ecosystem
- Launchpad
- Launchpad Maven Plugin
- Partial Bundle Lists
- OSGi Installer
- Installer Providers
- Installer Factories
- And more...

# 2 Apache Sling Launchpad



- Bootstrap code
- Launch an OSGi framework
  - Standalone jar with http service
  - As a web application with http bridge
- Includes Apache Felix
  - Handles framework updates
- Deploy packaged bundles
- Configurable
  - `sling.properties`
  - (system properties / servlet parameters)

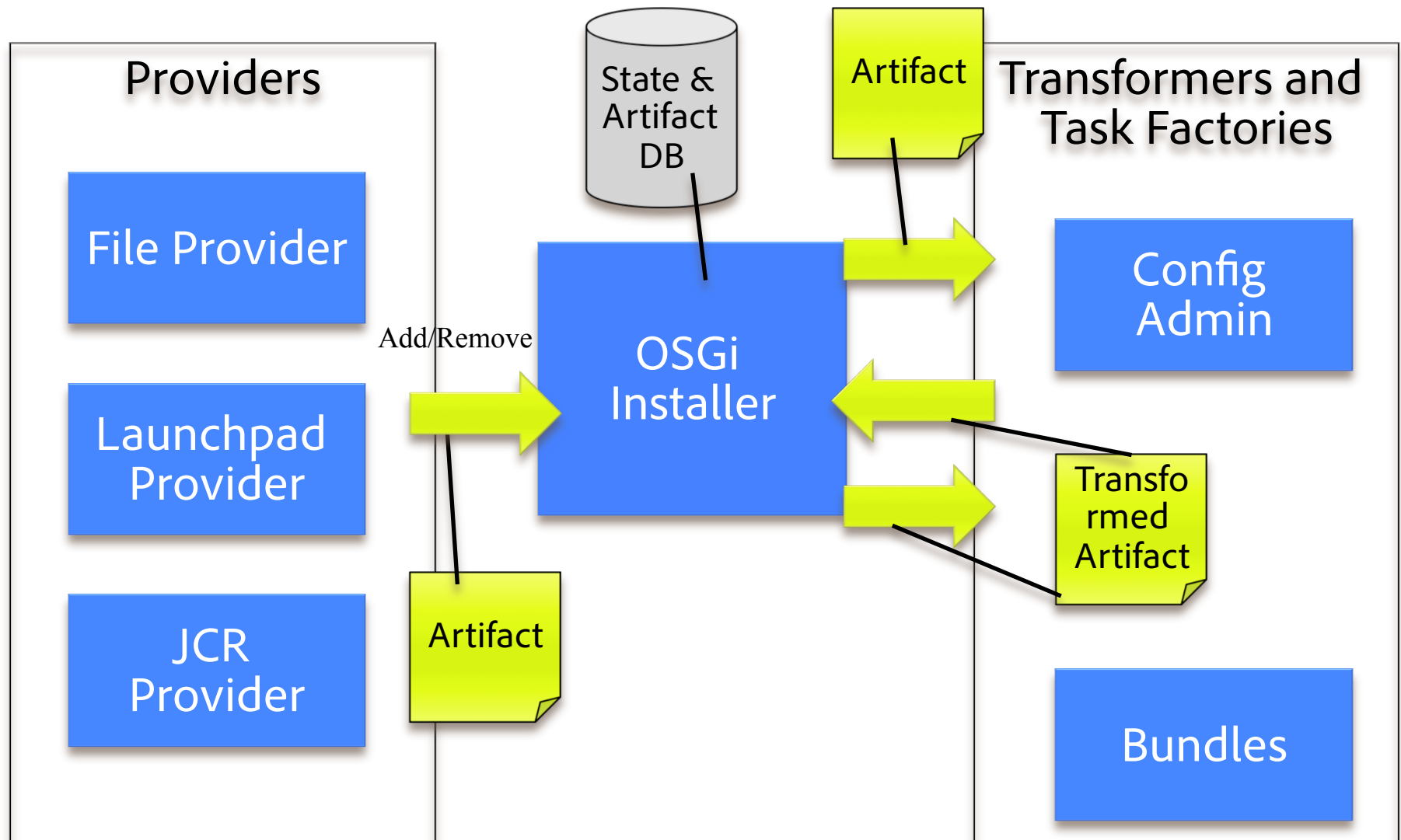
- Central configuration file
- OSGi framework configuration
  - System packages
  - Start level
- Bootstrap logging
- Configurations which should be there before the configuration admin is available

- Bootstrap bundles
  - Directly installed by the Sling Launchpad
  - E.g. logging, Apache Sling OSGi installer etc.
- Start levels can be used
- Advice: only bootstrapping bundles!
- Optional Bootstrap command file
  - Uninstalling bundles for upgrades

# 3 Apache Sling OSGi Installer

- Bundled service for
  - Install
  - Update
  - Uninstall
- Of
  - Bundles
  - Configurations
  - Custom artifacts

# Apache Sling OSGi Installer



# Installer Providers

- Just provide artifacts or inform about removal
- Don't need knowledge about artifacts
- Simple trackers

- Service for monitoring directories in the file system
  - Configurable through sling.properties (and system properties)
- Informs the OSGi Installer about
  - Add
  - Update
  - Delete
- Of artifacts
- Like Apache Felix FileInstall but delegates work!



- Service for monitoring nodes in the JCR
  - Configurable through Config Admin
- Informs the OSGi Installer about
  - Add
  - Update
  - Delete
- Of artifacts

- Many different tools for managing bundles and configurations
  - OSGi Installer, shell, web console etc.
- OSGi Installer checks for changes from other sources
  - Currently configurations only
  - Provider might support write back of changes
    - File and JCR provider do!

- Transformers inspect artifacts
- Detect artifact type
  - Bundle, configuration etc.
- Might transform artifact
- Provide unique id
  - Symbolic name + version
  - Configuration PID

- Register for artifact types
- Get transformed artifacts or removal event
- Perform their task
- OSGi Installer manages retry and failure
- A task might add another task
  - Like bundle refresh after all installs etc.

- Informs the OSGi Installer about
  - Artifacts bundled with the launchpad
- Scans predefined directories inside launchpad
- Supports start level
- Very handy for provisioning in combination with the Maven Launchpad Plugin

- Central mechanism for handling
  - Install
  - Update
  - Delete
- Triggered by various providers
- Highly customizable

# 4 Apache Sling Maven Launchpad Plugin

- A deliverable / distribution consists of
  - Apache Sling Launchpad
  - OSGi framework
  - Bootstrap bundles
  - Infrastructure bundles
  - Application bundles
  - Configurations
  - (Custom artifacts)



- Creates two artifacts for a Maven project
  - Standalone jar
  - Web application
- Use Apache Sling Launchpad Base Module
  - Launchpad
  - OSGi framework
  - Predefined sling.properties

- XML Configuration for the Maven plugin
- Contains list of bundles with their start levels
  - “boot” : bootstrap bundles handled by Launchpad
  - N : Bundle with start level N handled by OSGi installer and Launchpad Provider
  - 0 : Bundle with default framework start level
- Bundle list can contain other artifacts!

# Bundle List Example

```
<?xml version="1.0"?>
<bundles>
  <startLevel level="boot">
    <bundle>
      <groupId>org.apache.sling</groupId>
      <artifactId>org.apache.sling.commons.log</artifactId>
      <version>2.1.2</version>
    </bundle>

    ...

  <startLevel level="9">
    <bundle>
      <groupId>org.apache.felix</groupId>
      <artifactId>org.apache.felix.eventadmin</artifactId>
      <version>1.2.14</version>
    </bundle>

    ...

  <startLevel level="0">
    <bundle>
      <groupId>org.some.company</groupId>
      <artifactId>a.strange.artifact</artifactId>
      <version>1.2.0</version>
      <type>zip</zip>
    </bundle>
```

- Maven artifact for
  - A bundle list
  - Configurations
  - Additional sling.properties
  - Additional bootstrap commands
- Final deliverable can aggregate partial bundle lists

- Configuration through
  - Launchpad base artifact
  - Bundle list (+ partial bundle lists)
  - OSGi configurations
  - Additional sling.properties
    - Common, standalone, webapp
  - Bootstrap command file
    - Common, standalone, webapp
- Creates the deliverable

# 5 Useful Things

- Unique instance id : Sling ID
  - Created on first installation
- Run mode support
  - Can be used for different setups like
    - Prod, dev, test etc.
- Sling home directory
  - If something needs to be stored outside bundle data

- Very useful web based management console
- Supports plugins
- Plugins installed by
  - OSGi Installer
  - Sling settings service



# 6 Summary

- Creating a distribution is easy
  - Standalone jar files
  - Web application
- Bootstrap bundles get packaged into launchpad
  - Including OSGi installer and friends
- Leveraging JCR Installer (or others)
- Maven Launchpad Plugin
  - Define your deliverable as a package list

***The End...***