Deep Dive into OSGi

Tyler Maynard
AEM DEVELOPER

@TylersDesk www.tylermaynard.com

Overview

Overview of OSGi

Dependency management resolution

Components and annotations

Create an OSGi service

Code OSGi configurations

Handling OSGi events

OSGi Overview

Summarizing an OSGi Application

Independently developed and deployed

Appear or disappear at any time

Advantages of OSGi

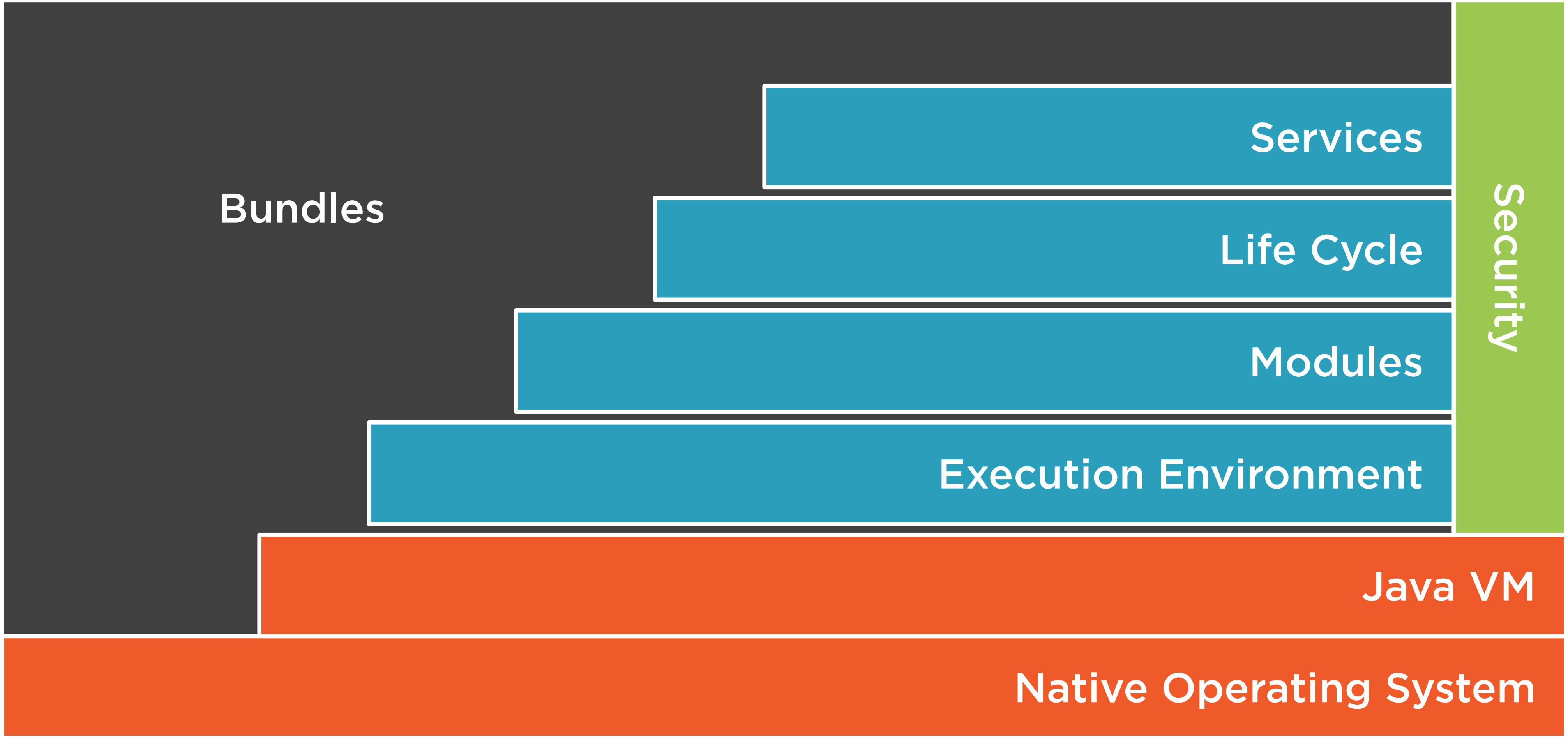
Manage from anywhere

Simplified development

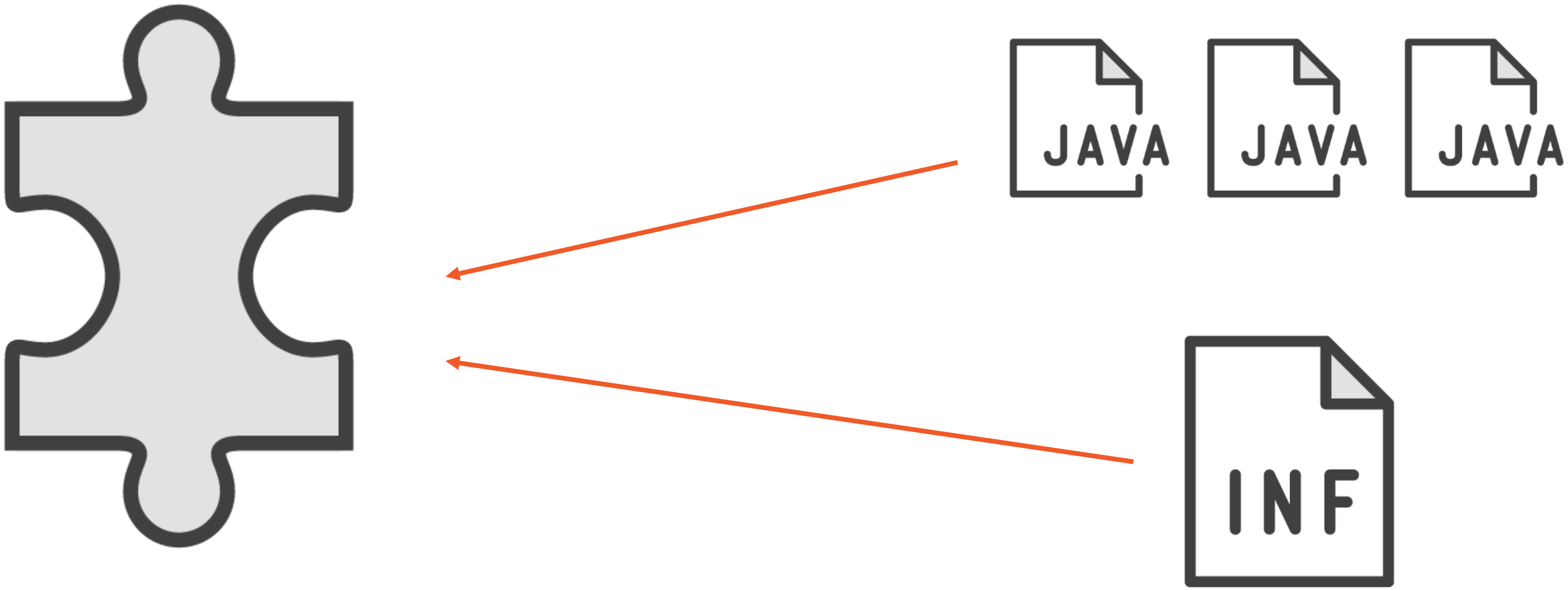
Efficiency and security

Other advantages

OSGi Architecture



Bundles



Jar file

Bundle Meta Information

Names

Version

Services imported and exported

Optional meta

Loosely Coupled

Imports and exports per version

Independent dependencies

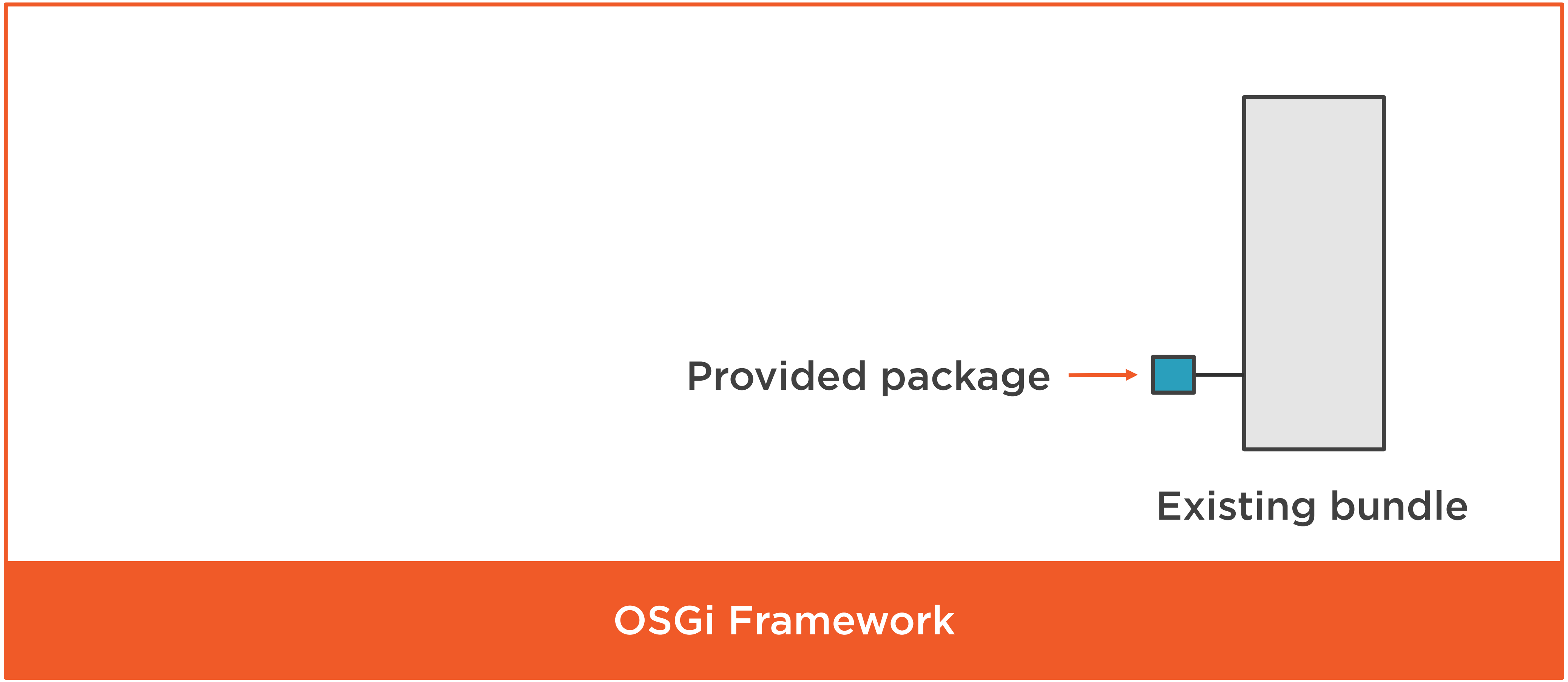
Error management

Modular thinking

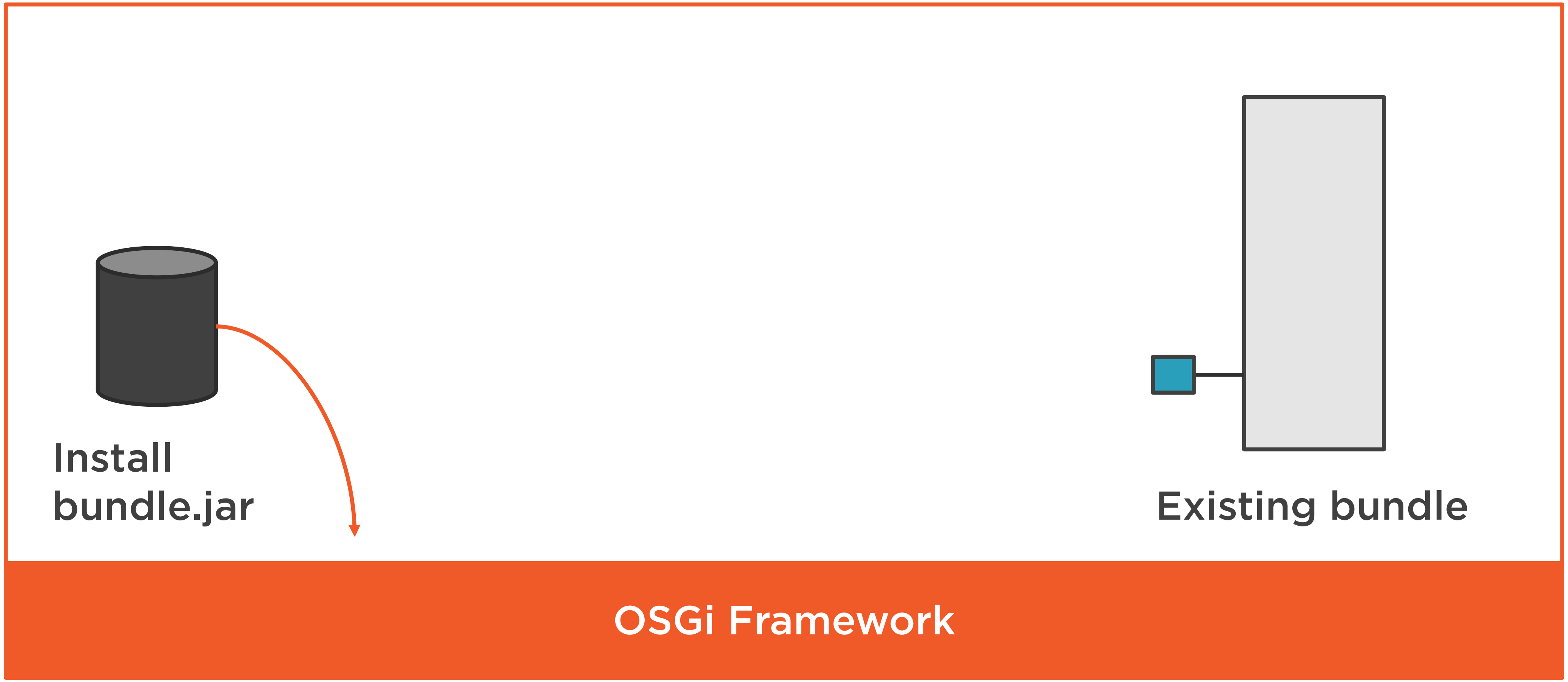
Proper metadata and version management

OSGi Dependency Management Resolution

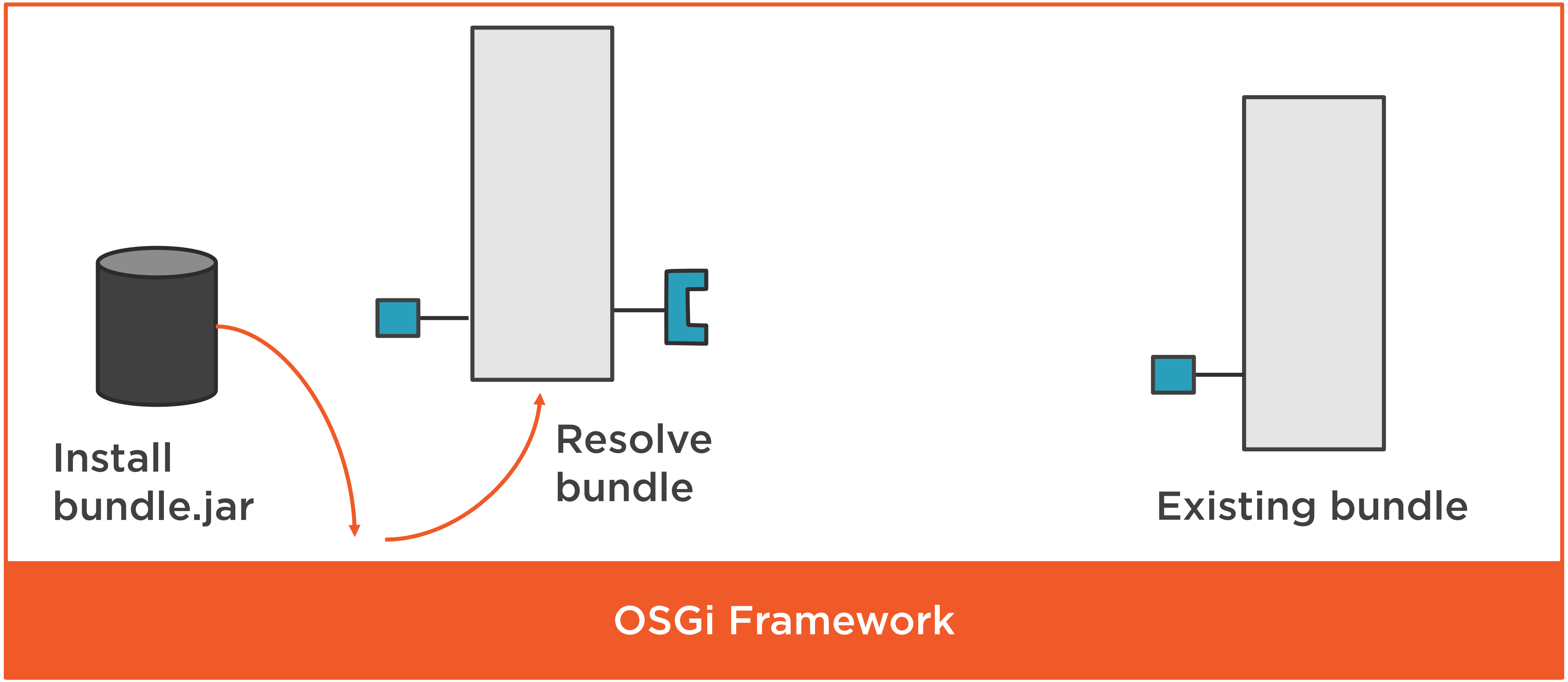
Bundle in Container



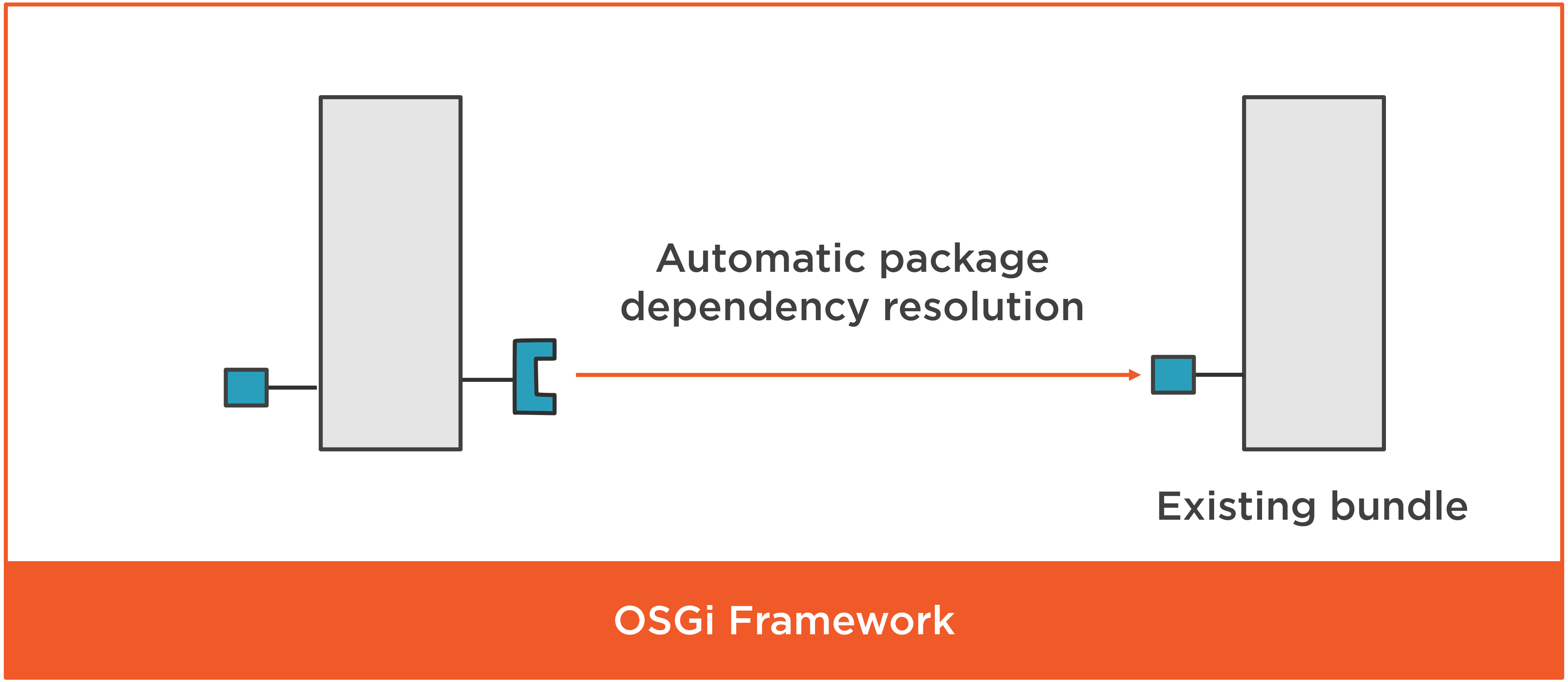
New Bundle in Container



Container Resolves Bundle



Automatic Dependency Resolution



Automatic package dependency resolution

Existing bundle

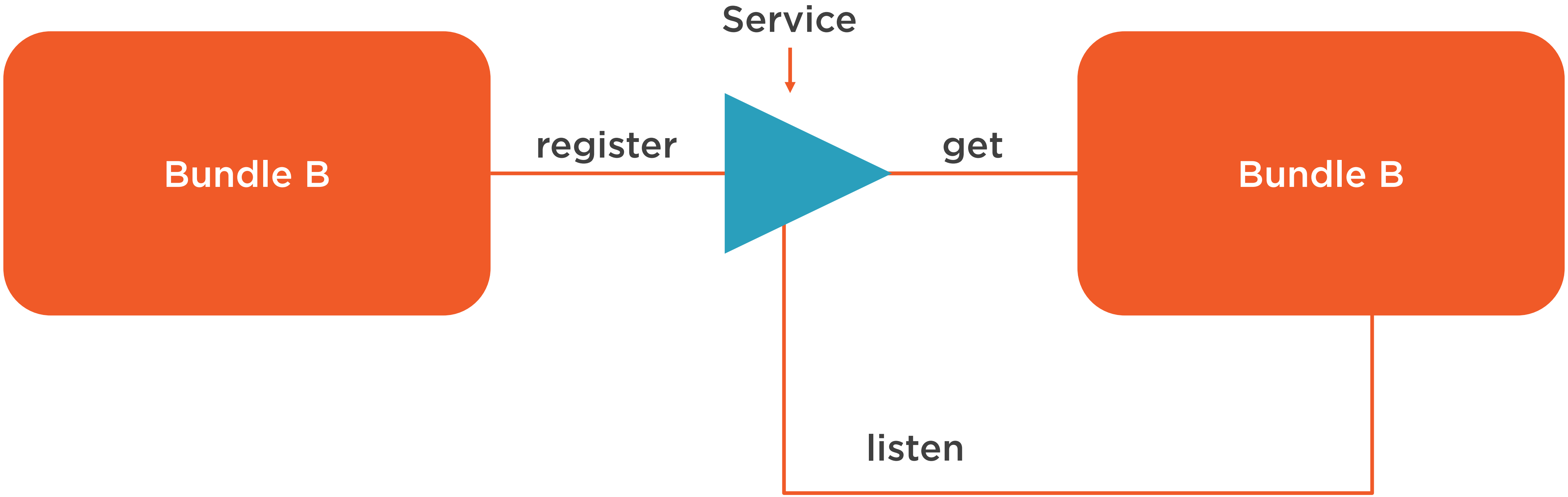
OSGi Framework

Modules

Hidden JAR content by default

Explicitly import

Services


Service

Bundle B

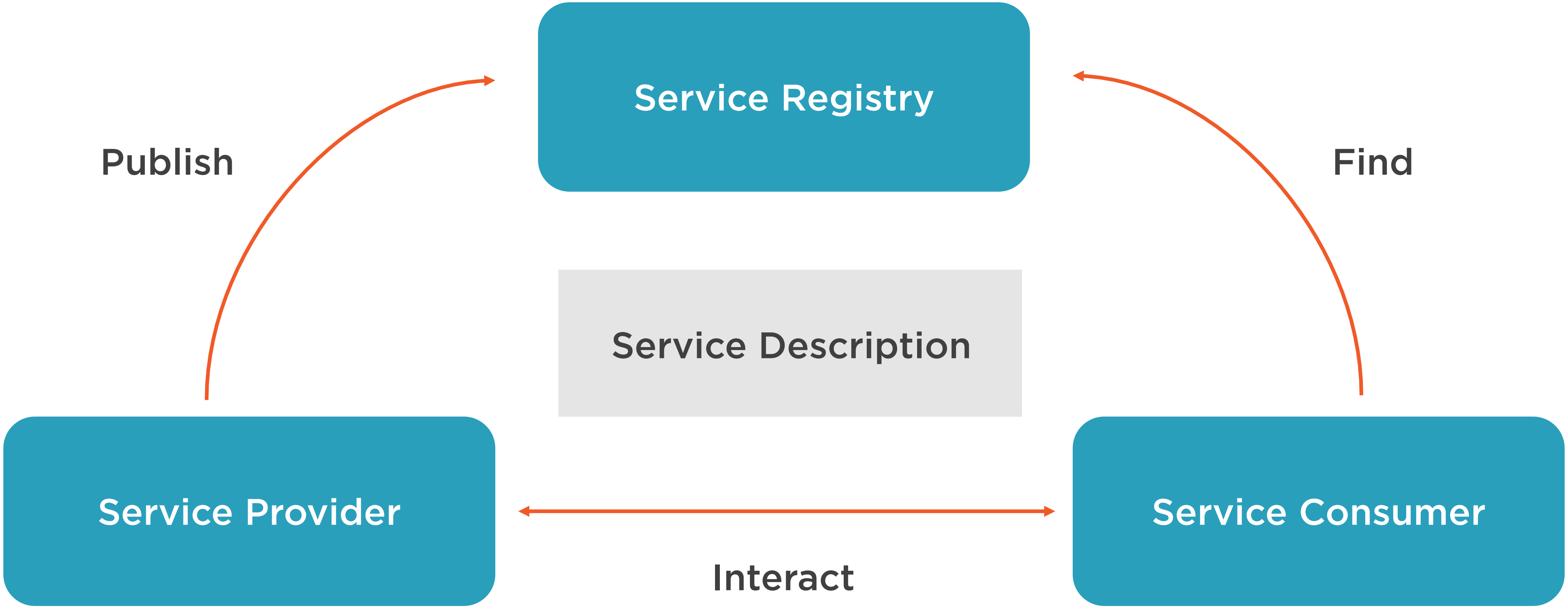
register

get

Bundle B

listen

Service Registry Model



Service Registry

Publish

Find

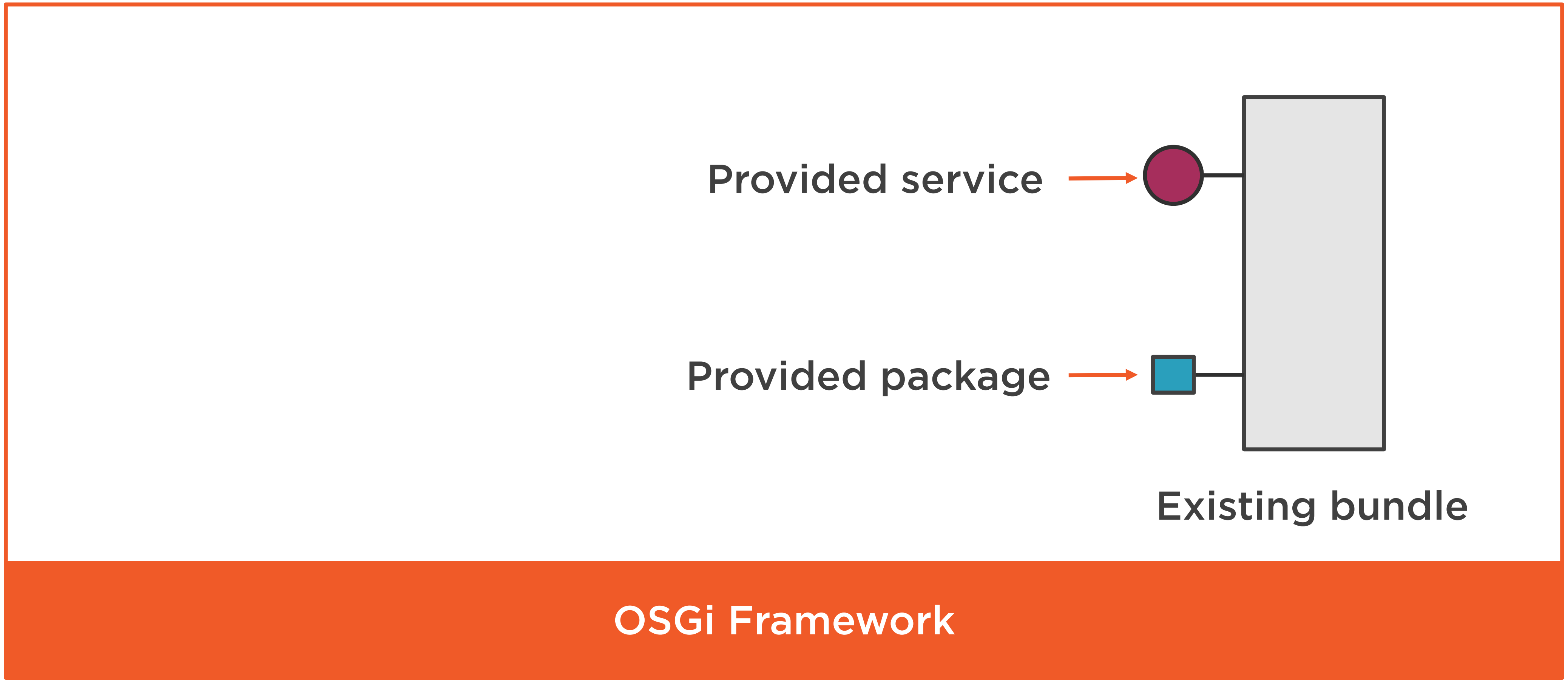
Service Description

Service Provider

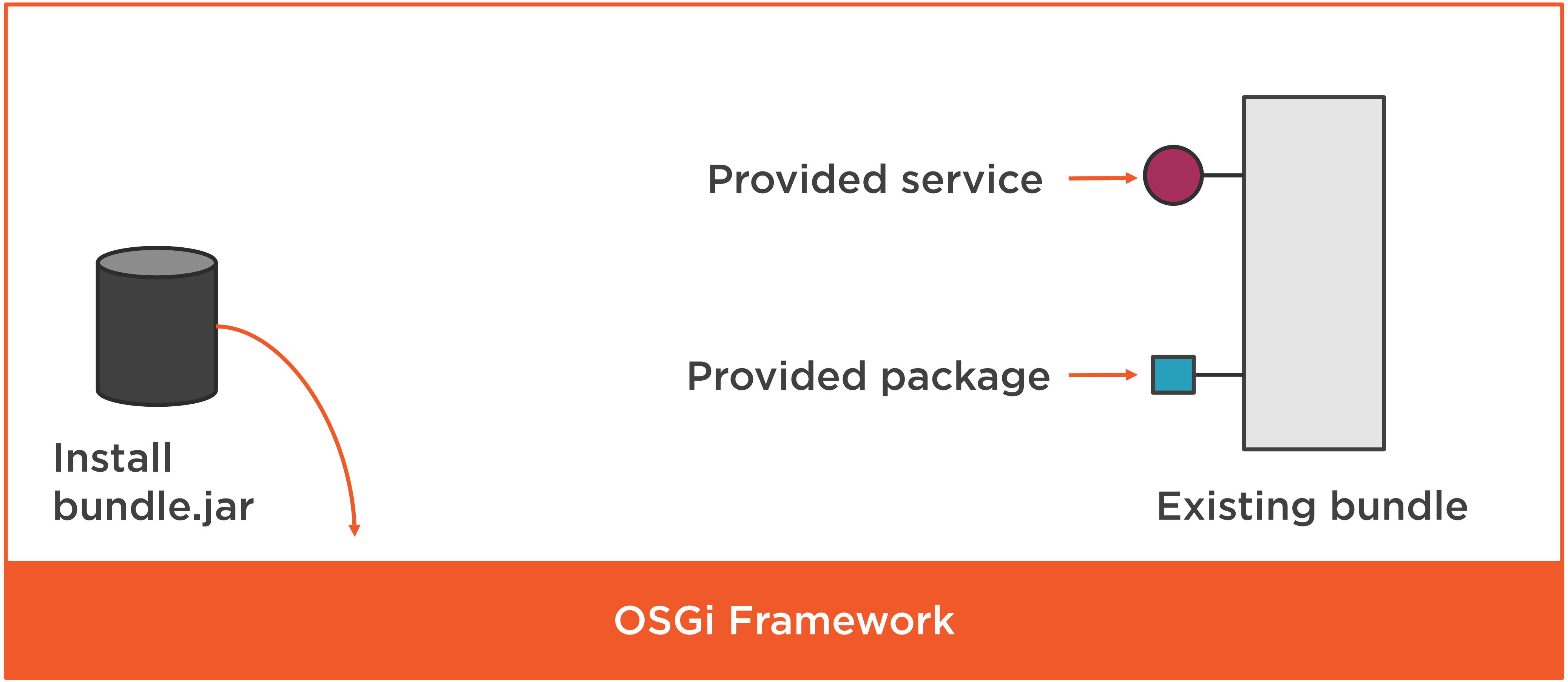
Service Consumer

Interact

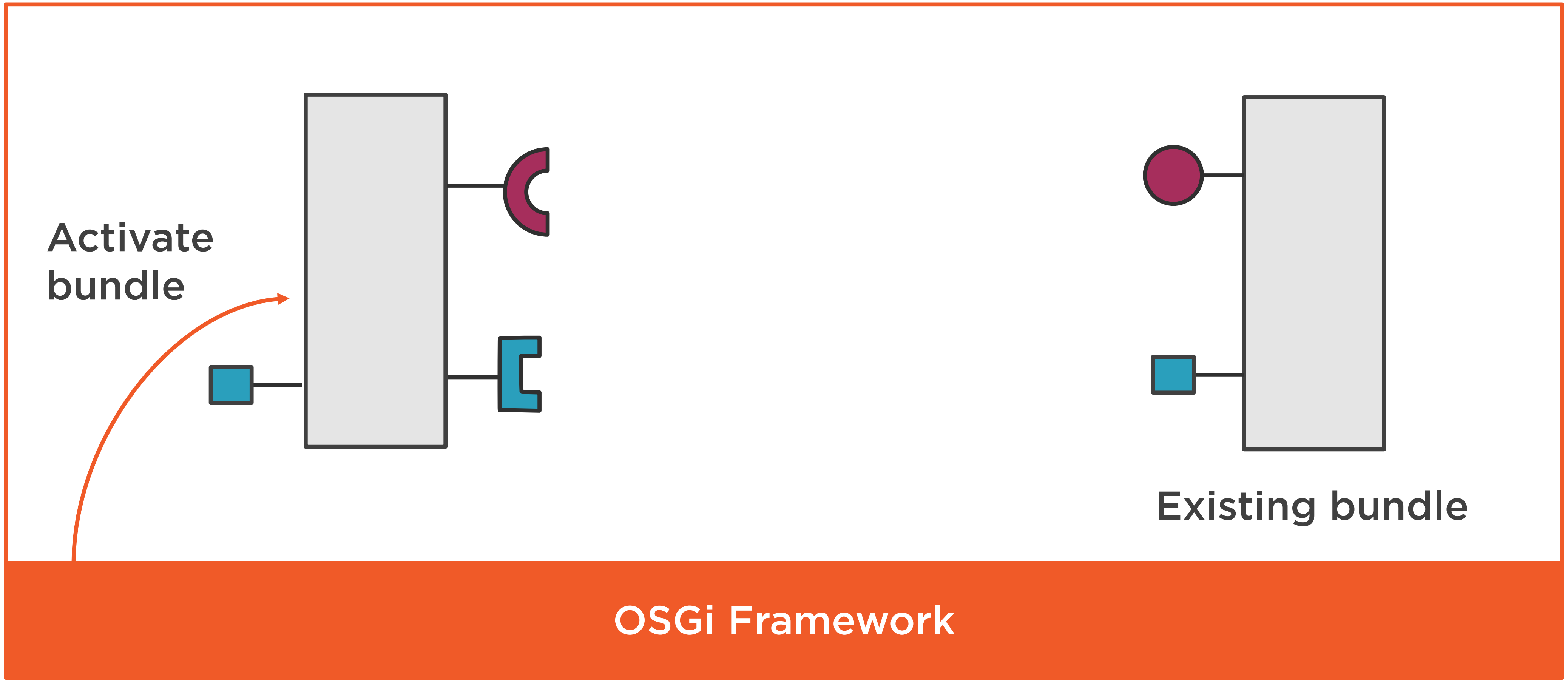
Dynamic Service Lookup



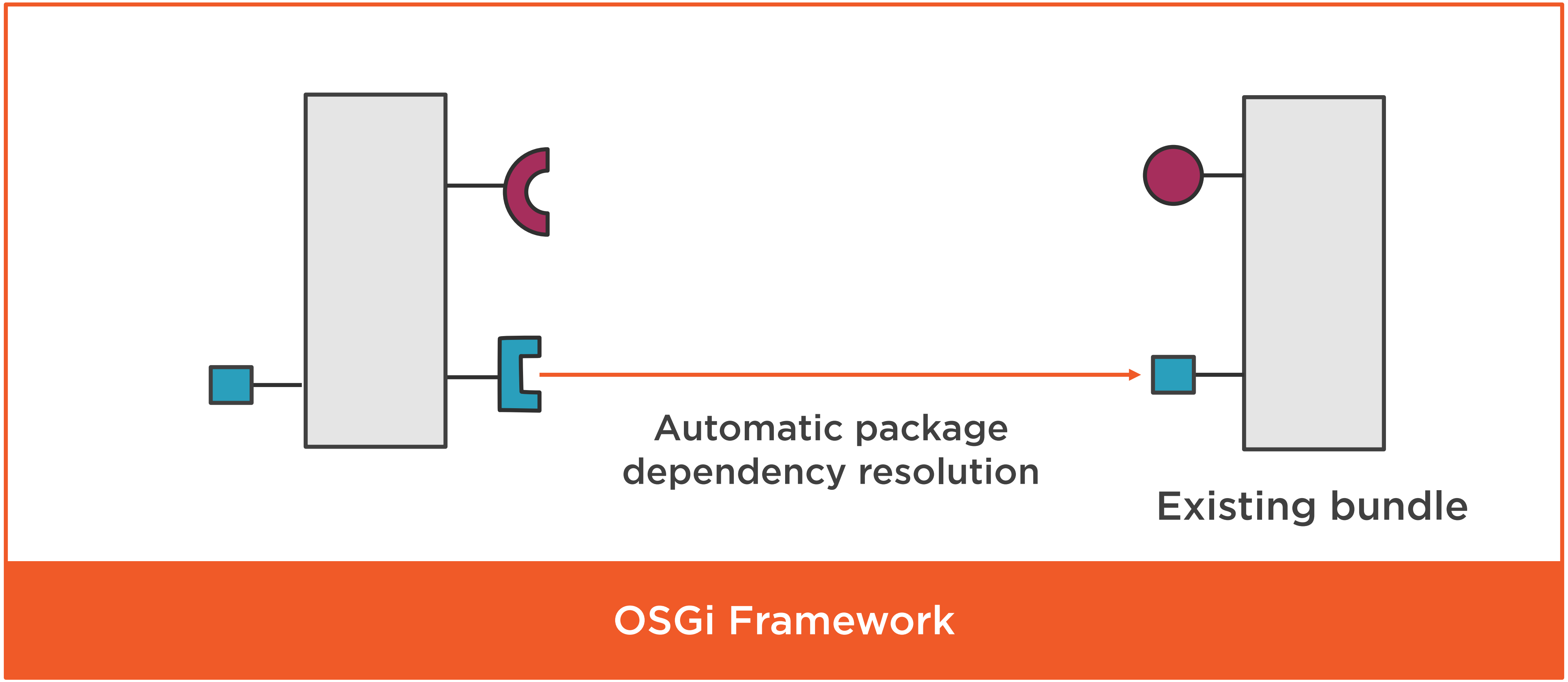
Install New Bundle



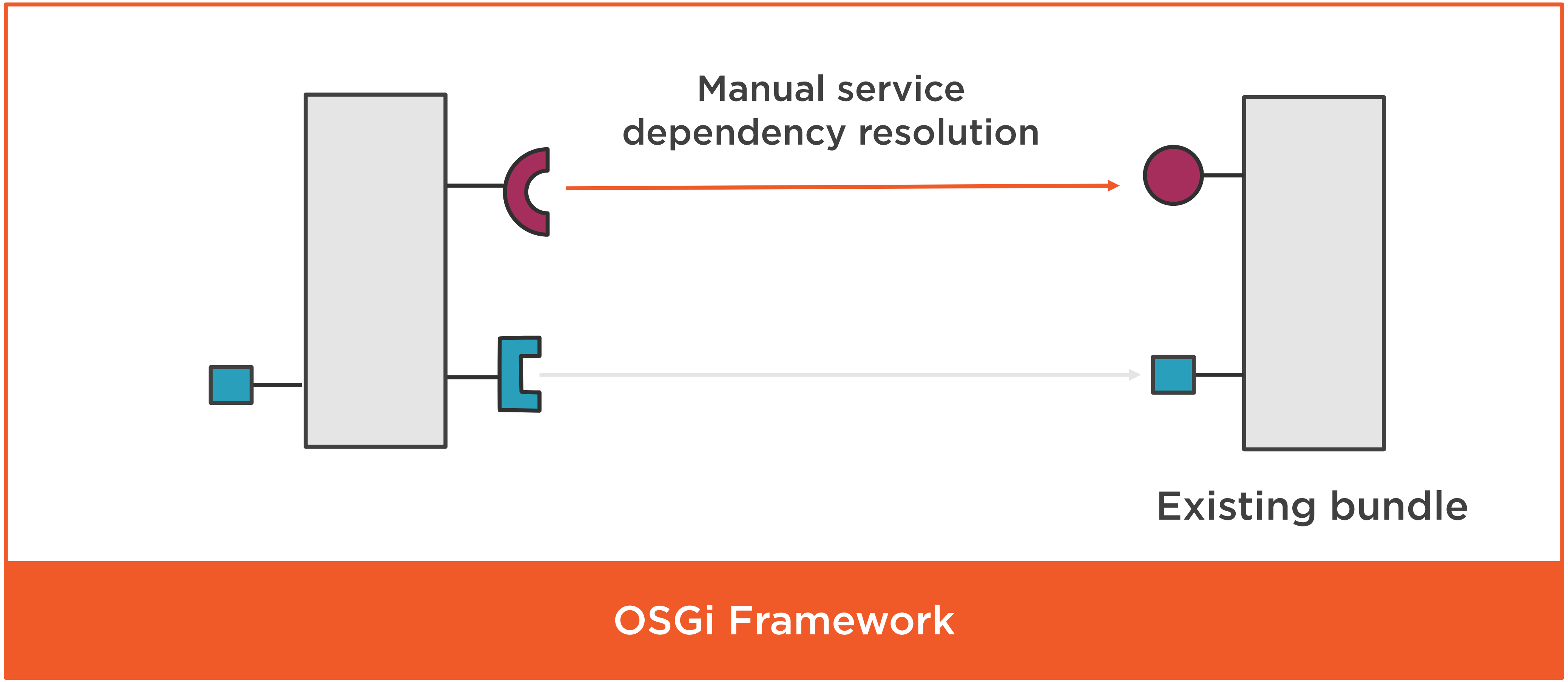
Activate New Bundle



Package Dependency Resolution



Service Dependency Resolution



OSGi Service Advantages

Lightweight

Lookup based on the interface name

Direct method invocation

Good design practice

Separates the interface from the implementation

Enables reuse, substitutability, loose coupling, and late binding

Declarative Services

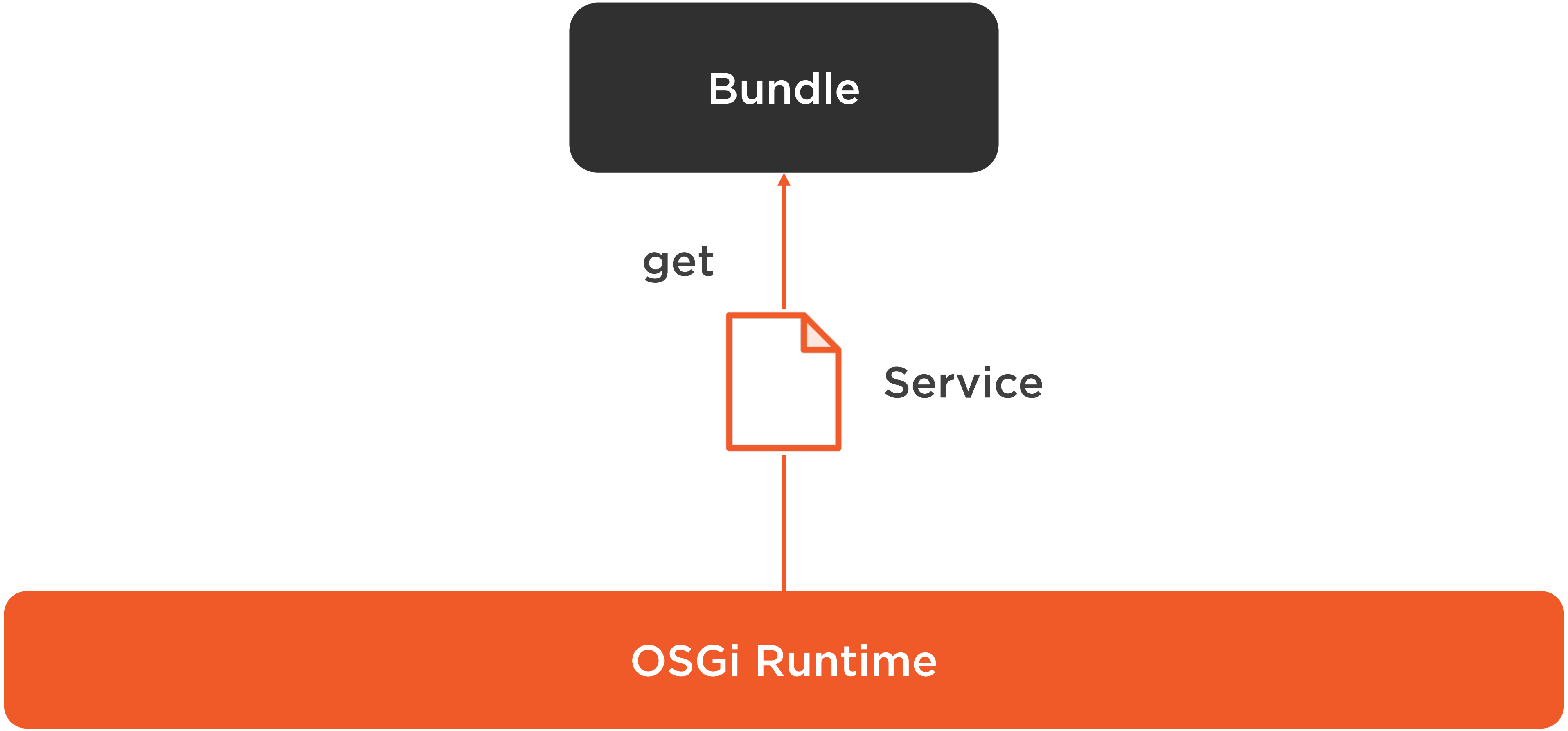
No explicit code

Instantiated on request

Own lifecycle

Configured from Configuration Admin

Benefits of Declarative Services



Benefits of OSGi

Reduced complexity

Reuse

Easy deployment

Dynamic updates

Simple

Implement a Bundle Activator

Demo

Create Activator java class

Add the bundle to the POM file

Deploy project to AEM

Check that the bundle has started

Understanding Sling Events

OSGi API for Events

Sending an event

Receiving an event

Types of Events

Predefined events

Service events

Configuration events

JSR observation events

Listening to OSGi Events

Implement the eventHandler interface

Subscribe by service registration

Select the event with service properties

Listen to OSGi Events in Sling

```
@scr.property name="event.topics"

valueRef="ReplicationAction.EVENT _ TOPIC“
```

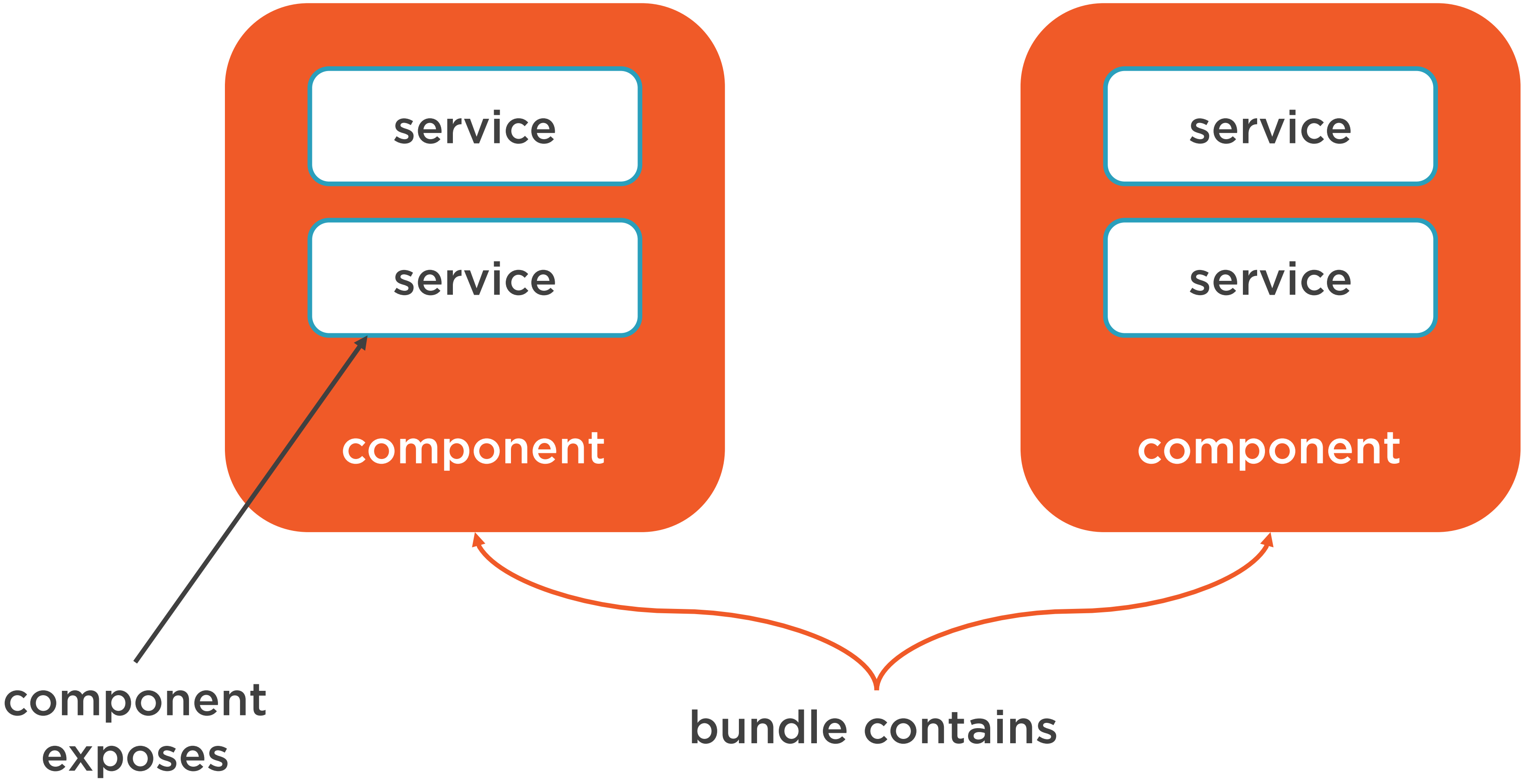
```
@scr.property name="event.topics"

valueRef="org.apache.sling.api.SlingConstants.TOPIC _ RESOURCE _ ADDED"
```

Indicating an Event Handler Service

```
@Component

@Property(name = "event.topics", value =

ReplicationAction.EVENT_TOPIC)

@Service(value = EventHandler.class)

public class MyEventListener implements JobConsumer,EventHandler {

public void handleEvent(Event event) {

if (EventUtil.isLocal(event)) {

JobUtil.processJob(event, this);

}

}
```

Components and Annotations in OSGi

OSGi Components



OSGi Components Continued

XML file

Manifest file header

Activate and Deactivate methods

Service Component Runtime (SCR)

Installing Bundles

Upload via Web Console

Folder named ‘Install’ in the JCR

Annotations

Component

Activate

Deactivate

Modified

Service

Reference

Property

Java Compiler Annotations

@Override

@SuppressWarnings

Configurable Services

Persistent Identifiers (PID)

Component name

Create and Use Custom Service

Demo

Create Interface and Service

Deploy project to AEM

Test java logic

Code OSGi Configurations

Demo

Add configuration properties

Check OSGi configuration in console

Check page to verify OSGi configurations

Handle OSGi Events

Demo

Create ReplicationListener

Deploy the project into AEM and test

Verify

Summary

Overview of OSGi

Dependency management resolution

Components and annotations

Create an OSGi service

Code OSGi configurations

Handling OSGi events