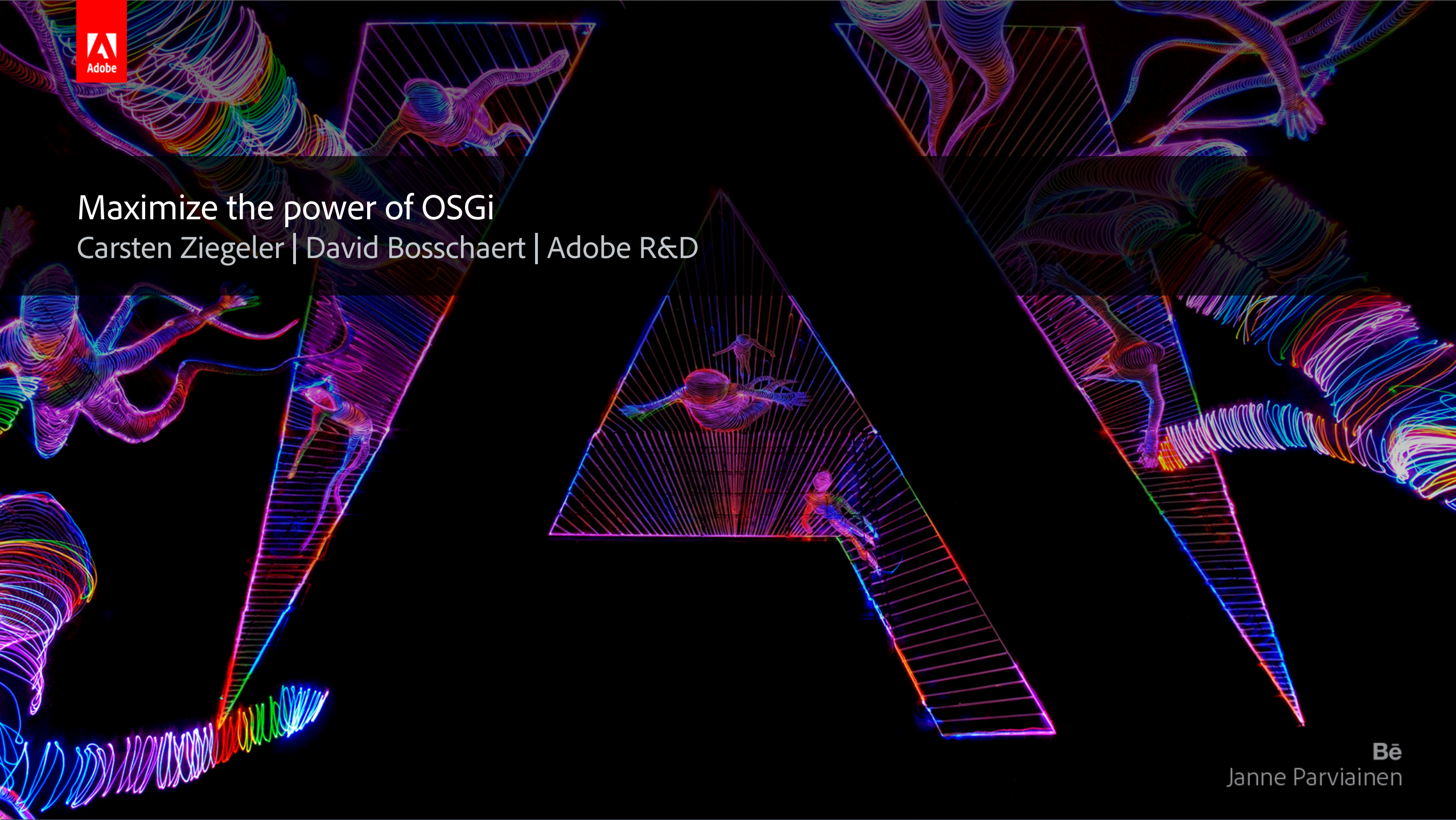

Adobe
Maximize the power of OSGi
Carsten Ziegeler | David Bosschaert | Adobe R&D
Bē
Janne Parviainen

# About

Carsten Ziegeler cziegeler@apache.org @cziegeler

- RnD Adobe Research Switzerland
- Member of the Apache Software Foundation
- VP of Apache Felix and Apache Sling
- OSGi Core Platform, OSGi Enterprise, OSGi IoT Expert Groups
- Member on the board of the OSGi Alliance
- Book / article author, technical reviewer, conference speaker

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# About

David Bosschaert davidb@apache.org @davidbosschaert

- R&D Adobe Ireland
- Co-chair OSGi Enterprise Expert Group
- ISO/IEC JTC1 SC38 Cloud Computing committee member for Ireland
- Apache Felix, Aries PMC member and committer
  - … other opensource projects
- Cloud and embedded computing enthusiast

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# Today's contents

- Asynchronous OSGi Services
- Declarative Services
  - with Configuration Admin
- HTTP Whiteboard
- Subsystems

*Image by Joadl: Lyme_Regis_harbour_02b on wikipedia*

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.



Adobe

# Async programming: OSGi Promises

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.



Adobe

# OSGi Promises

Javascript-style promises, can be used with Java 5 or later.

- Asynchronous chaining
- Very simple programming model

Promises can be used outside of OSGi framework

Recommended implementation:
https://svn.apache.org/repos/asf/aries/trunk/async

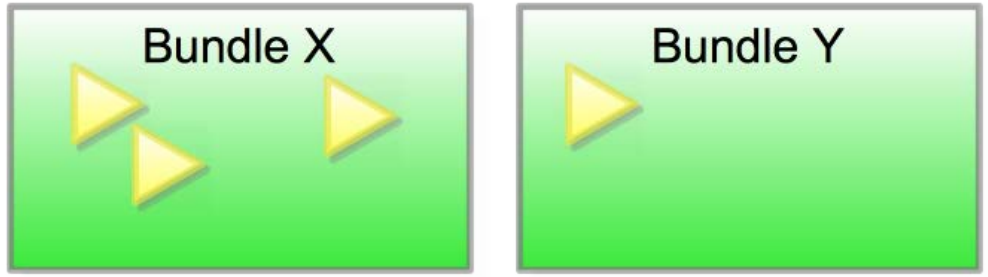
```
public class PromisesTest {

    public static void main(String... args) {
        System.out.println("Starting");
        takesLongToDo(21)
            .then(p -> intermediateResult(p.getValue()))
            .then(p -> finalResult(p.getValue()));
        System.out.println("Async computation kicked off");
    }

    public static Promise<Long> intermediateResult(Long l) {
        System.out.println("Intermediate result: " + l);
        return takesLongToDo(l * 2);
    }

    public static Promise<Void> finalResult(Long l) {
        System.out.println("Computation done. Result: " + l);
        return Promises.resolved(null);
    }

    public static Promise<Long> takesLongToDo(long in) {
```

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe


© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

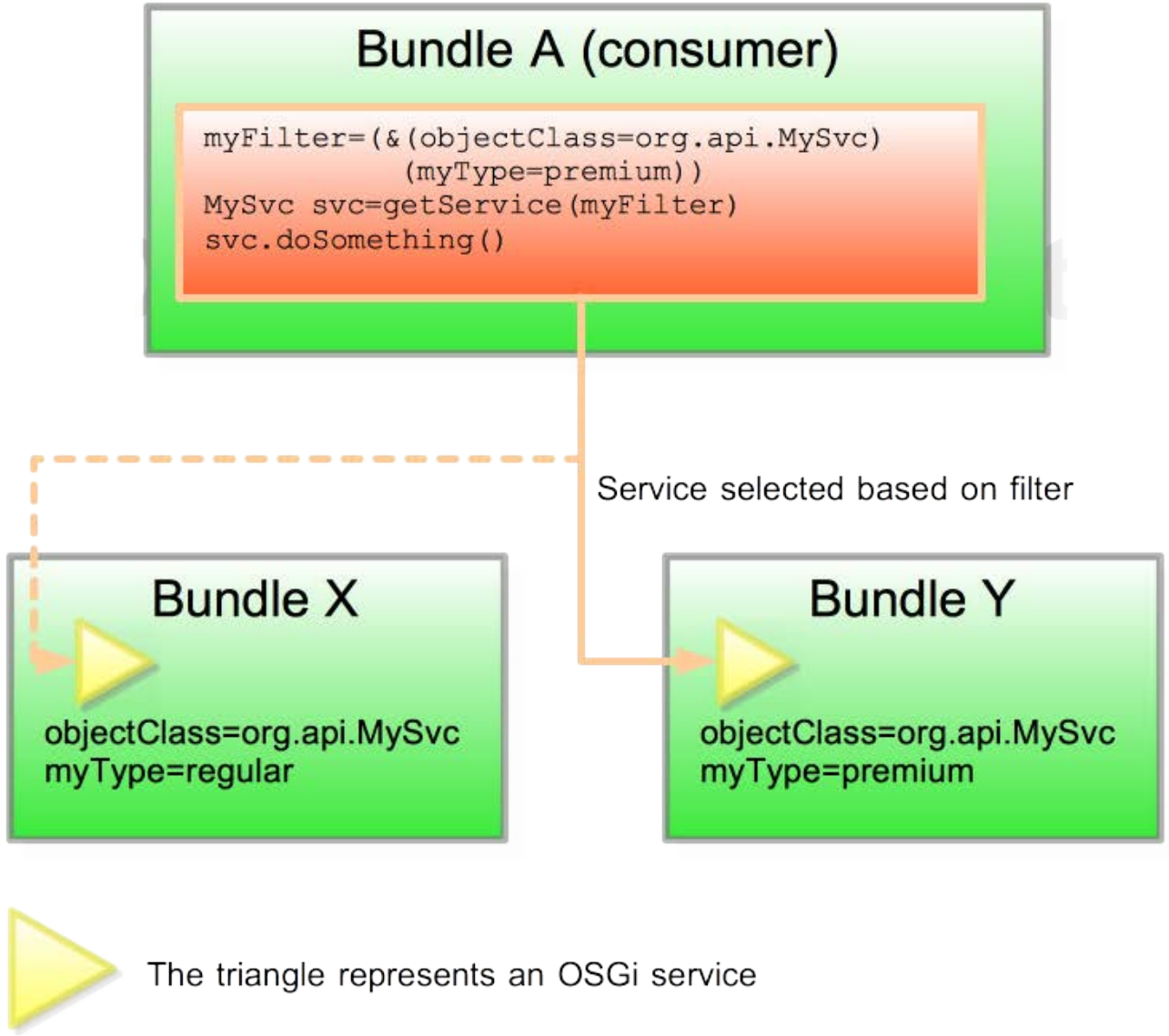
# Brief intro to OSGi Services




- Services are Java Objects (POJOs)
  - registered by Bundles
  - consumed by Bundles
- "SOA inside the JVM"
- Services looked up by type and/or custom filter
  - "I want a service that implements `org.acme.Payment` where `location=US`"
  - One or many
- Dynamic! Services can be updated without taking down the consumers
  - OSGi Service Consumers react to dynamism

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# Dynamic Service Selection




© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# Async Services

- Take an existing OSGi Service …
  - … and make it async!
- Or use an async-optimized one
- Also works great with Remote Services

- Recommended implementation:
  https://svn.apache.org/repos/asf/aries/trunk/async

Normal service use:

```
TimeConsumingService tcs = ... // from Service Registry
System.out.println("Invoking Big Task Synchronously...");
System.out.println("Big Task returned: " + tcs.bigTask(1));
// further code
```

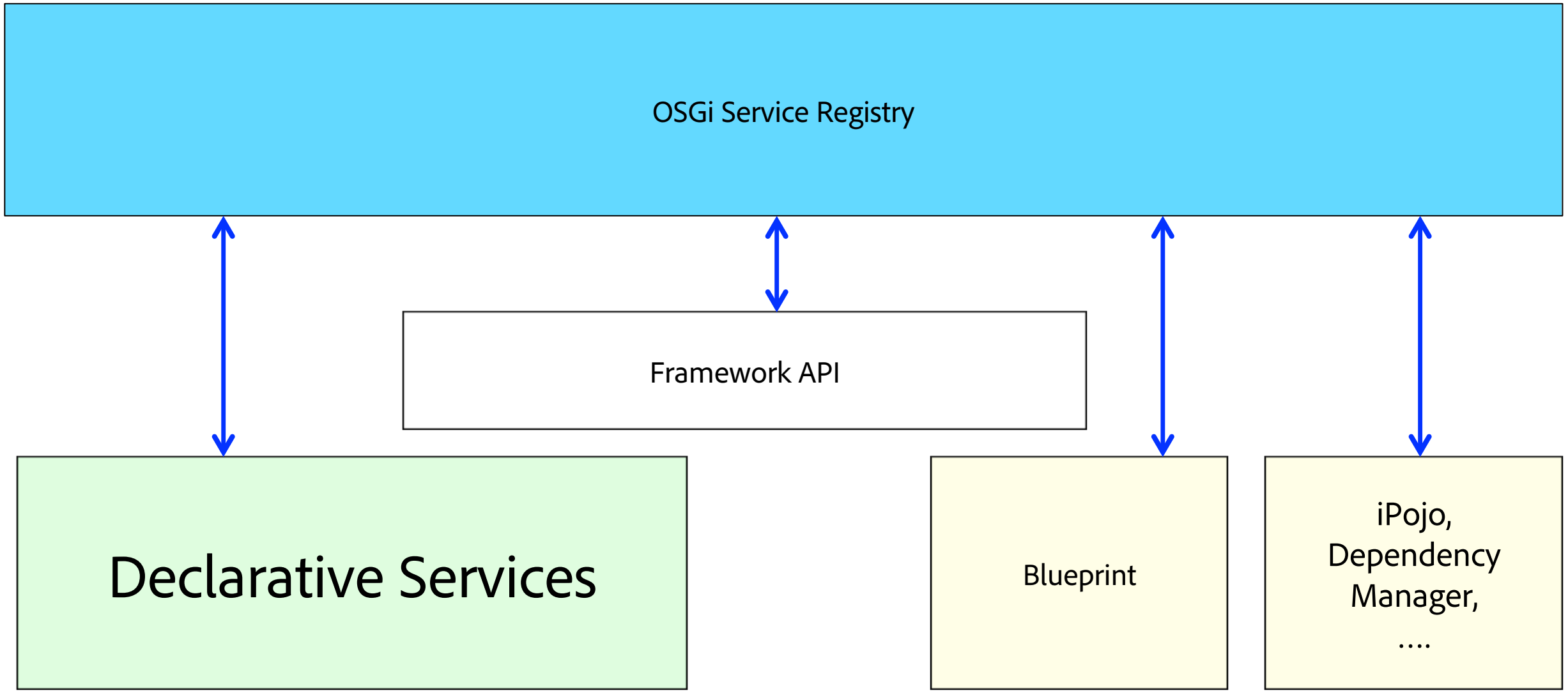
- Async service use
  - via mediator obtained from Async Service

```
TimeConsumingService tcs = ... // from Service Registry
Async async = ... // Async Service from Service Registry
TimeConsumingService mediated = async.mediate(
            tcs, TimeConsumingService.class);

System.out.println("Invoking Big Task Asynchronously...");
async.call(mediated.bigTask(1))
  .then(p -> bigTaskFinished(p.getValue()));
System.out.println("Big Task submitted");
```

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

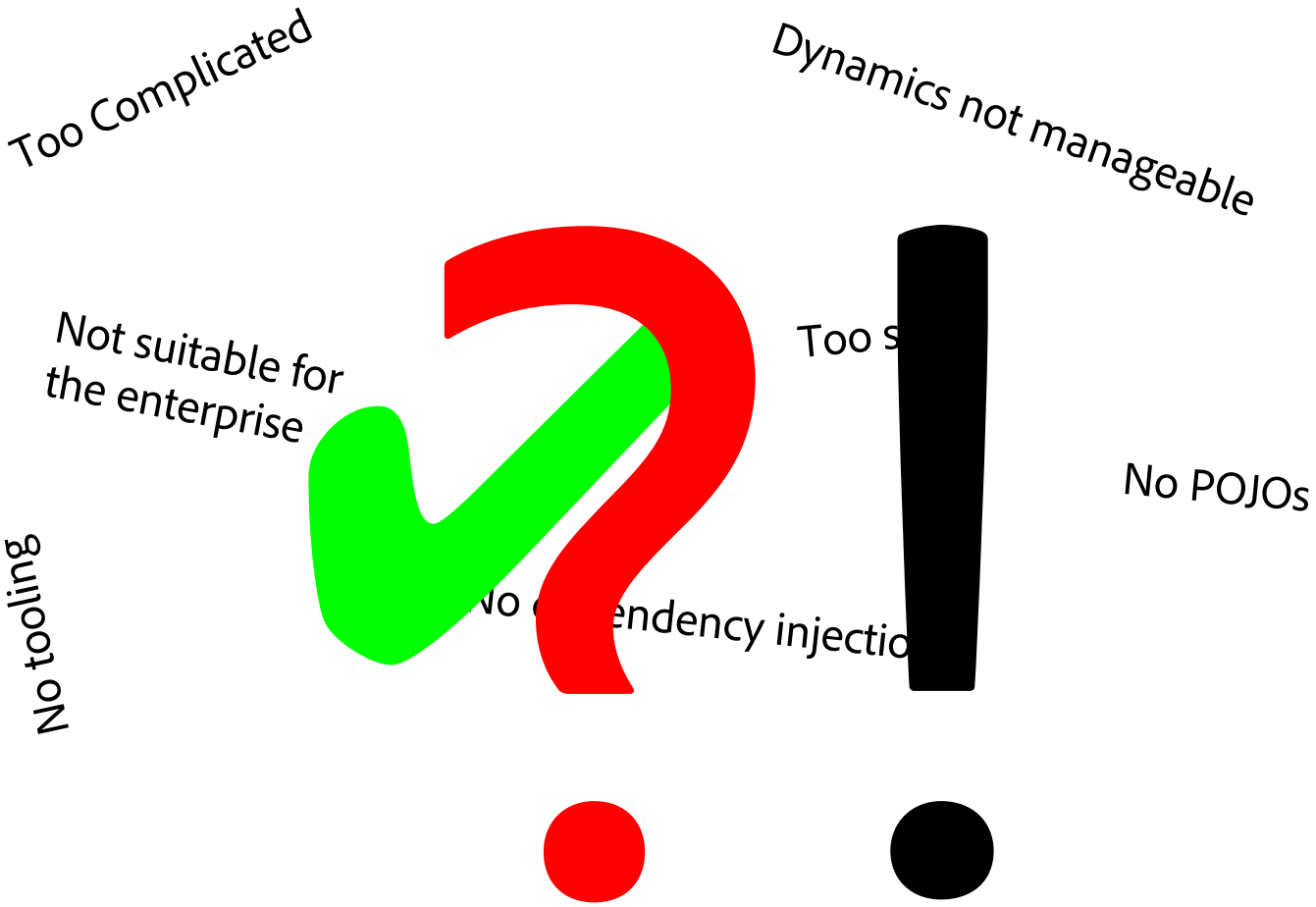
# Component Container Interaction




© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# Declarative Services

# OSGi Preconceptions




© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.


© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

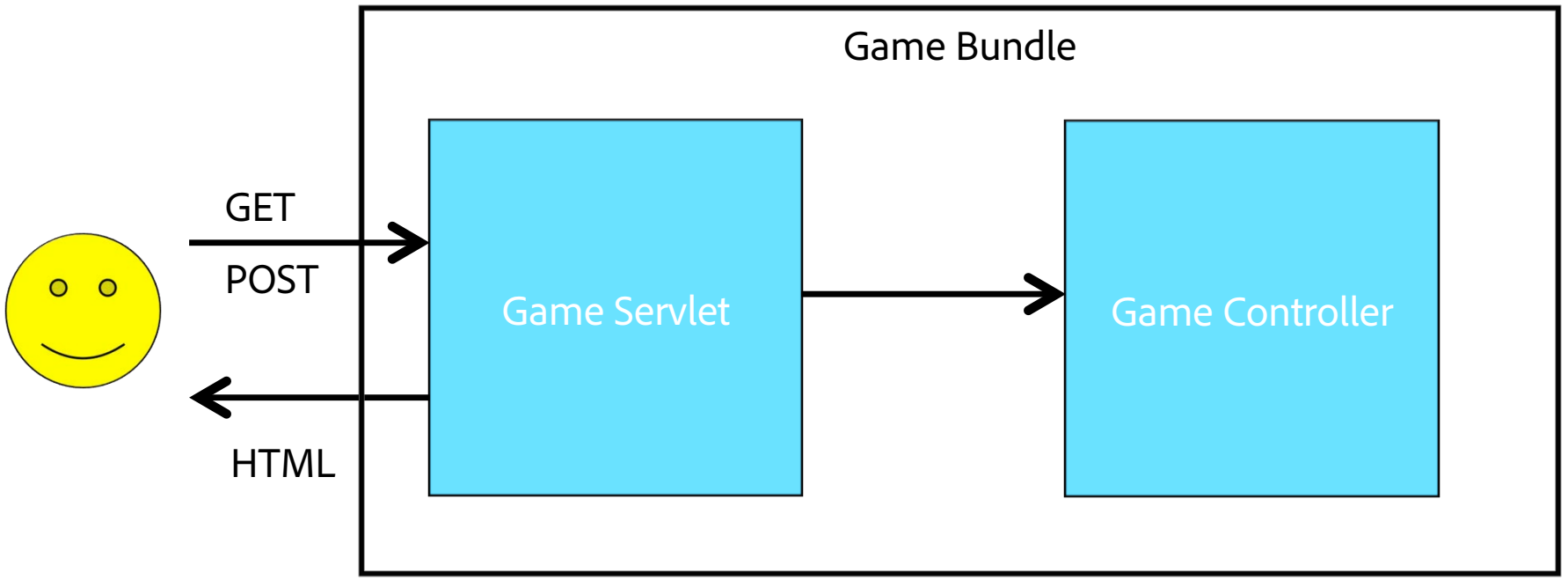
# Building Blocks

- Components
- Services
- Module aka Bundle

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# Game Design




© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# Game Design - API

```
public enum Level {

        EASY,
        MEDIUM,
        HARD
}
```

```
public class Game {

        // game status
}
```

```
public interface GameController {


    Game startGame(final String name,
                   final Level level);


    int nextGuess(final Game status,
                  final int guess);


    int getMax(final Level level);
}
```

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# Implementation

```
import org.osgi.service.component.annotations.Component;



@Component
public class GameControllerImpl implements GameController {

    ...
```

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# Range from 1 to a configurable max value per level

```
public @interface Config {

        int easy_max() default 10;
        int medium_max() default 50;
        int hard_max() default 100;

}
```

| Configuration (Dictionary) |
| --- |
| easy.max = "8"<br>medium.max = 40L |

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

```
private Config configuration;

@Activate
protected void activate(final Config config) {

    this.configuration = config;
}
```

```
public @interface Config {

            int easy_max() default 10;
            int medium_max() default 50;
            int hard_max() default 100;

}
```

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.



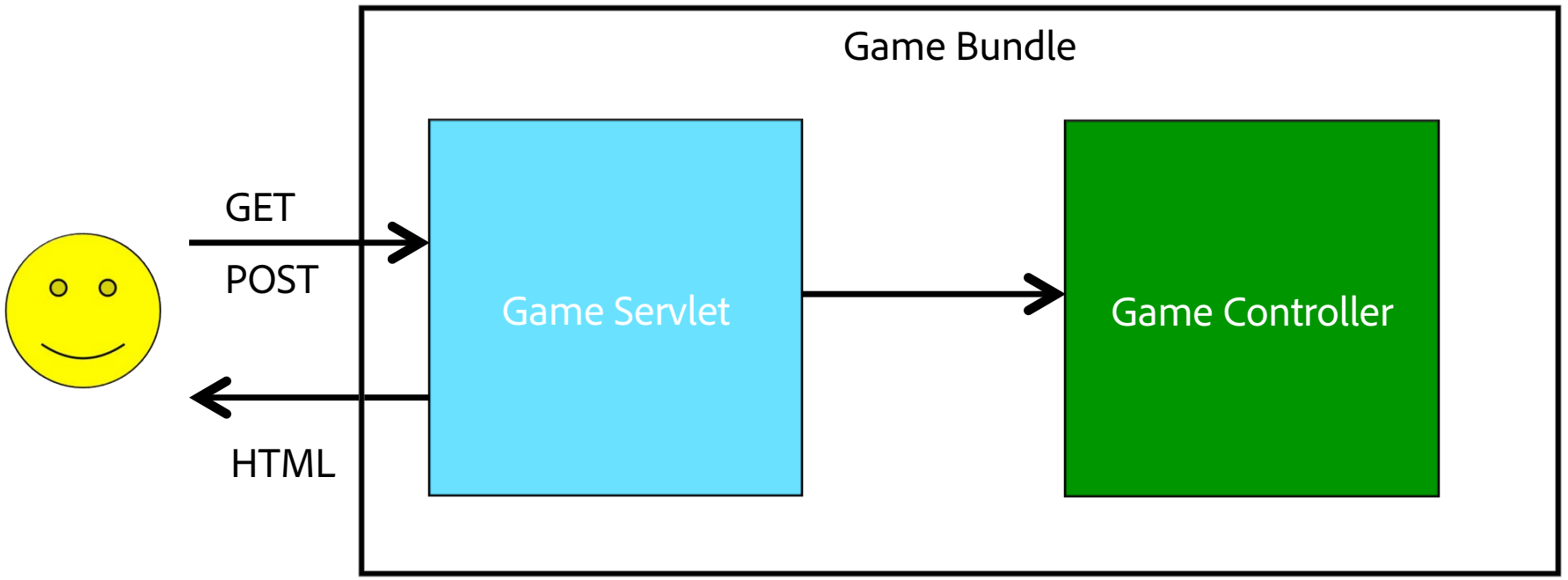
Adobe

```
public int getMax(final Level level) {

    int max = 0;

    switch (level) {
              case EASY : max = configuration.easy_max(); break;
              case MEDIUM : max = configuration.medium_max(); break;
              case HARD : max = configuration.hard_max(); break;
     }
    return max;
}
```

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# Game Design




© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

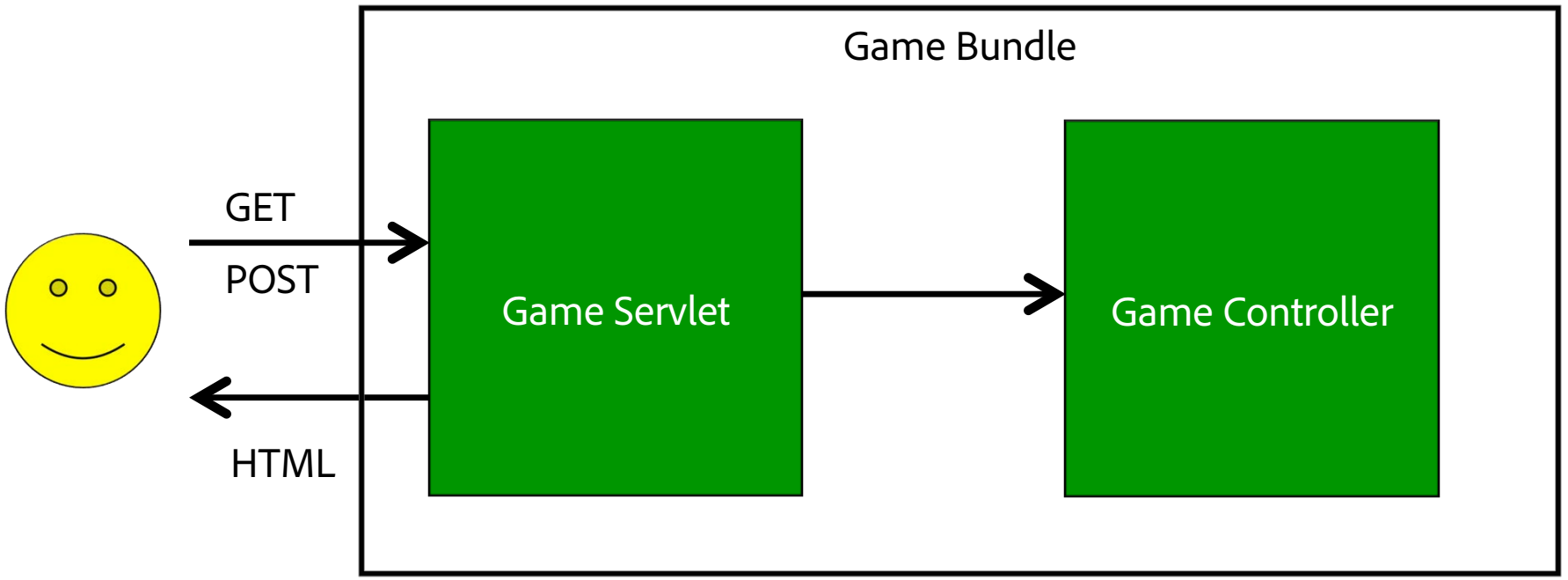
# Web?

```
@Component( service = Servlet.class ,
  property="osgi.http.whiteboard.servlet.pattern=/game")
public class GameServlet extends HttpServlet {
      @Reference
      private GameController controller;
```

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

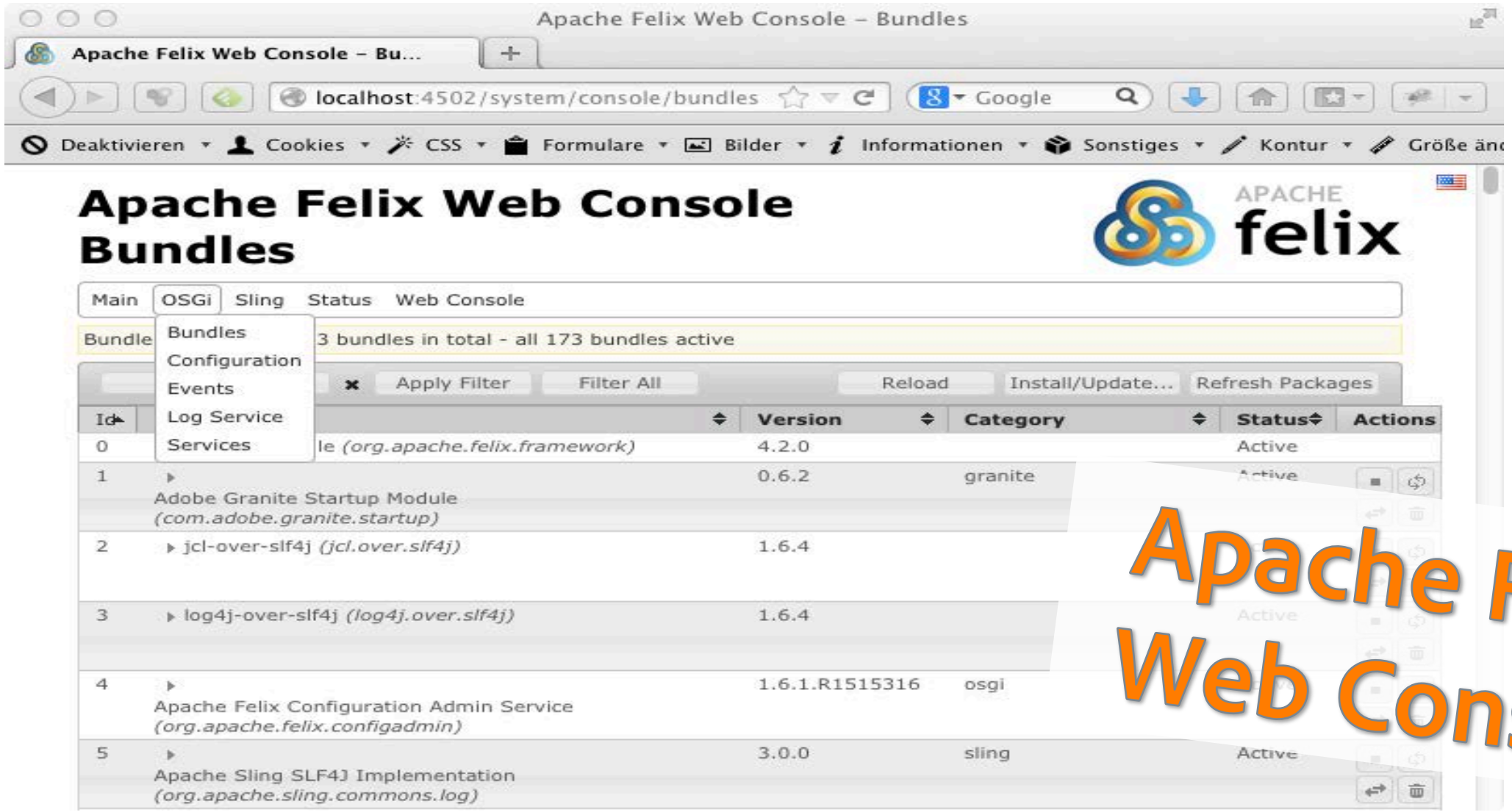
Adobe

# Game Design




© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# Management



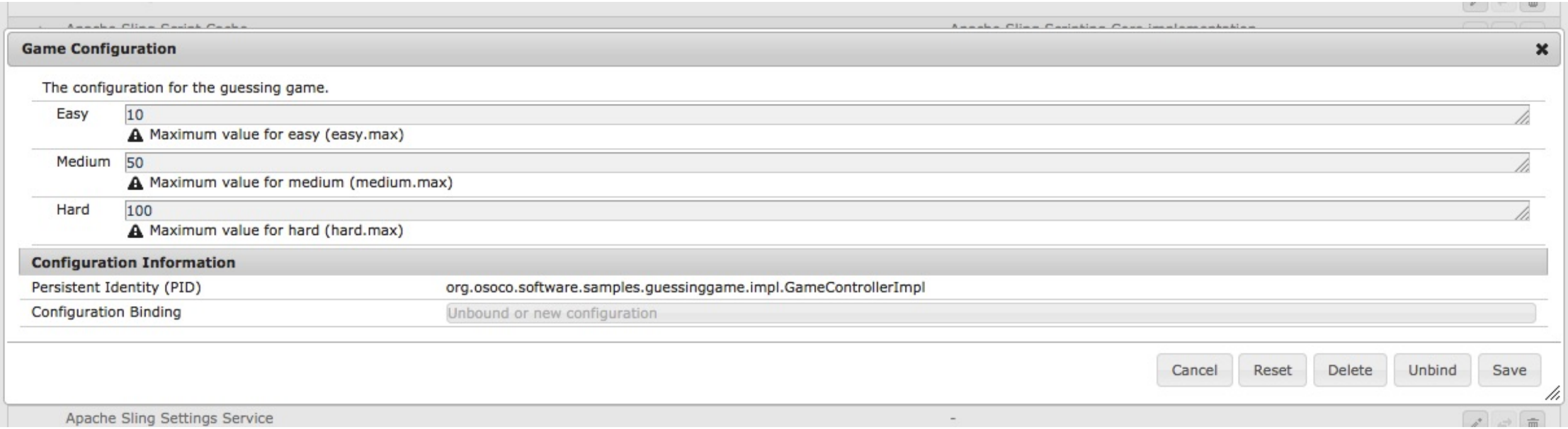

Apache Felix Web Console

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.



Adobe

# Component Management




© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# Metatype I

```
@ObjectClassDefinition(
        name = "Game Configuration",
        description = "The configuration for the guessing game.")


public @interface Config {

        @AttributeDefinition(name="Easy",
                        description="Maximum value for easy")
        int easy_max() default 10;
```

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# Metatype II

```
@Component
@Designate( ocd = Config.class )

public class GameControllerImpl
             implements GameController {
```

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# Dynamics

- Lazy instantiation
- References
- Reconfiguration

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# Unary References

```
@Reference
private GameController controller;
```

```
@Reference(cardinality=ReferenceCardinality.OPTIONAL
           policy=ReferencePolicy.DYNAMIC)
private volatile GameStatistics stats;
```

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# Multiple References

```
@Reference(cardinality=ReferenceCardinality.MULTIPLE)
private volatile List<Highscore> highscores;
```

```
@Reference
private final Set<Highscore> highscores =
        new ConcurrentSkipListSet<Highscore>();
```

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

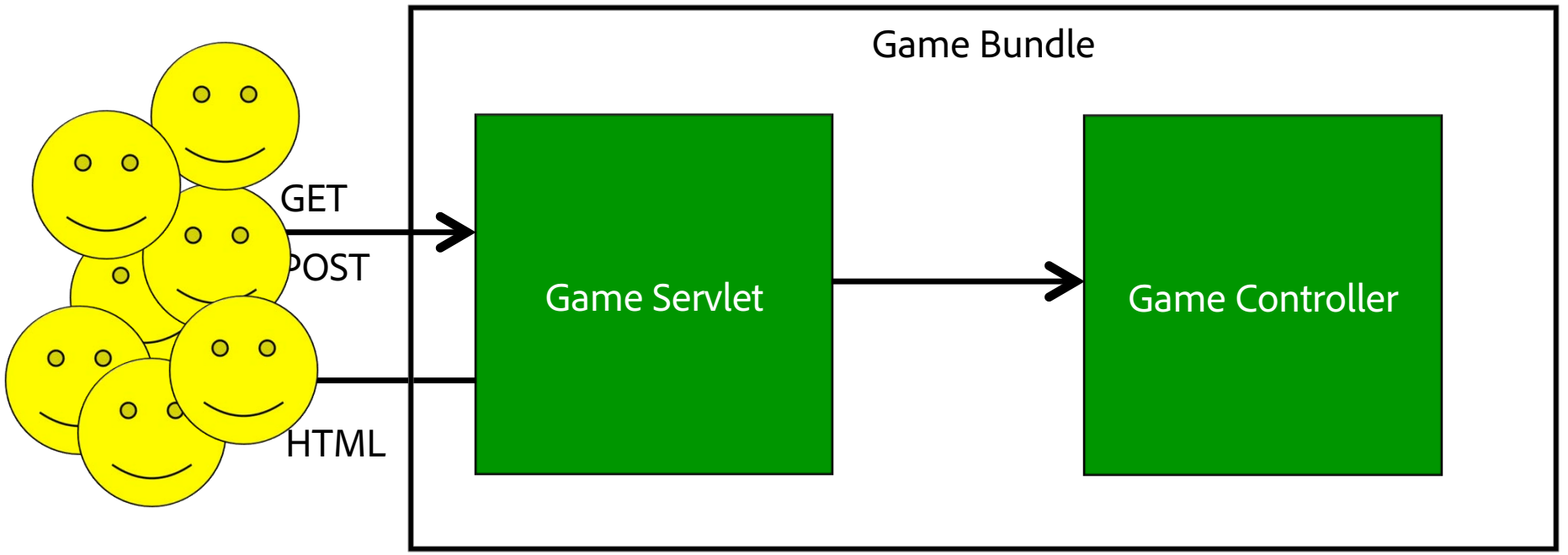
Adobe

# Reconfiguration

```
private volatile Config configuration;

@Activate
@Modified
protected void activate(final Config config) {

    this.configuration = config;
}
```

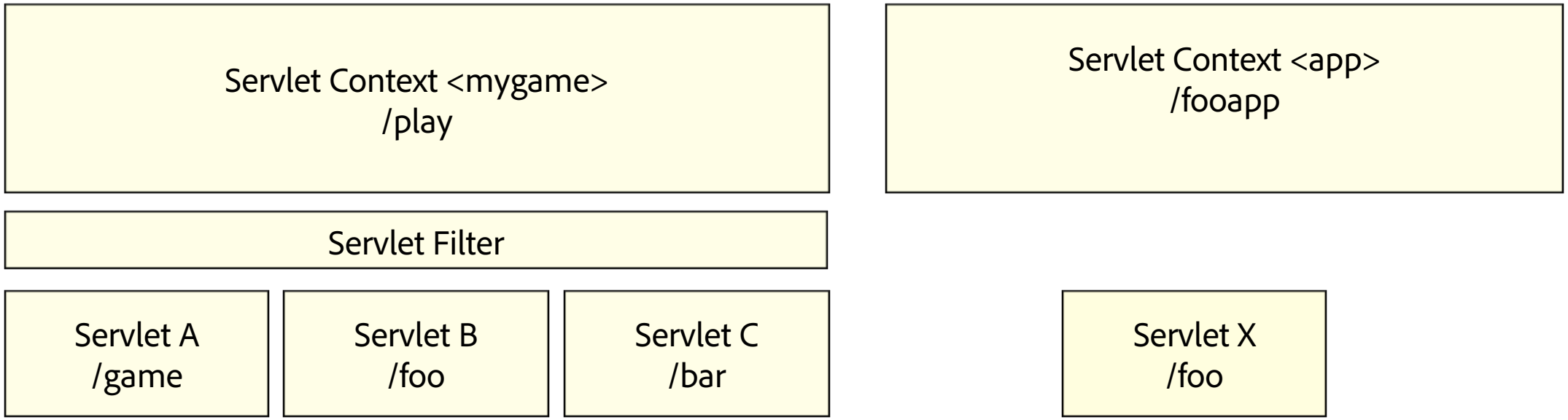
© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe




© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# Servlet Contexts




© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# Web Contexts

```
@Component( service = Servlet.class ,
            scope = ServiceScope.PROTOTYPE,
  property={"osgi.http.whiteboard.servlet.pattern=/foo",
            "osgi.http.whiteboard.context.select=mygame"}

public class ServletB extends HttpServlet {
```

```
 @Component( service = Servlet.class ,
             scope = ServiceScope.PROTOTYPE,
   property={"osgi.http.whiteboard.servlet.pattern=/bar",
             "osgi.http.whiteboard.context.select=mygame"}

 public class ServletC extends HttpServlet {
```

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# HTTP Whiteboard Service

- Servlet contexts (grouping, authentication)
- Servlets
- Resources
- Filters
- Listeners

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# Declarative Services

- Easy too use
- Pojos
- DI with handling dynamics
- Integrates with Configuration Admin and Metatype

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# Try it out soon

- Tooling *is* available
- Official open source implementations available

- But how do I get *this* easily deployed?

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# OSGi Subsystems

Adobe
Bē
The Made Shop

# Package your app for deployment

- Most applications are composed of multiple bundles
- Need a neat deployment package
- Previously: proprietary solutions

*"Airdrop pallets" by Senior Airman Ricky J. Best - defenselink.mil (search for "airdrop"). Licensed under Public Domain via Wikimedia Commons*

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# OSGi Subsystems

- Chapter 134 of Enterprise spec
- Packaging of multi-bundle deployments
  - in .esa file (a zip file)
- Optional isolation models
- Bundles either in:
  - .esa file
  - OSGi Repository

*Jack Kennard TSA 3-1-1 rule*
*https://www.flickr.com/photos/javajoba/4013349543*

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

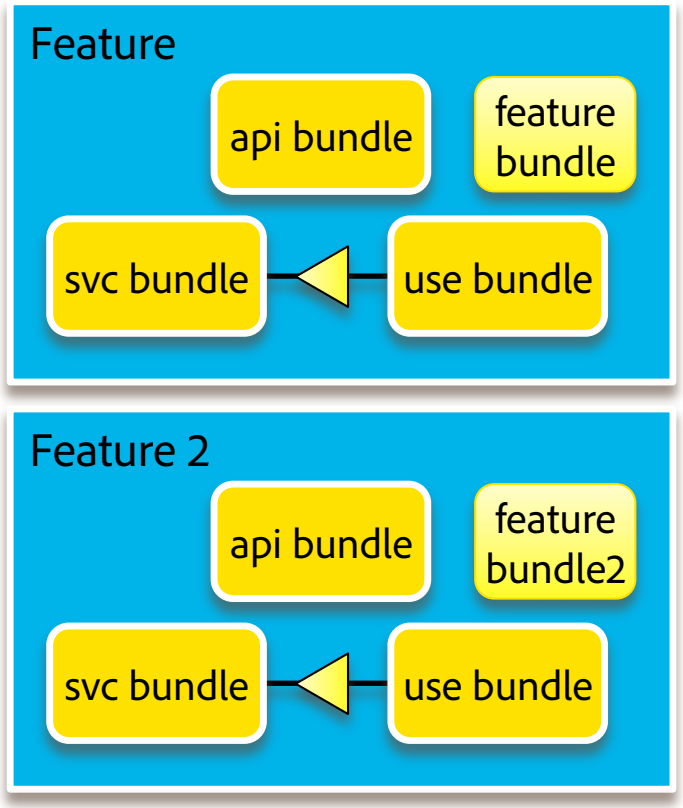
# Subsystem types

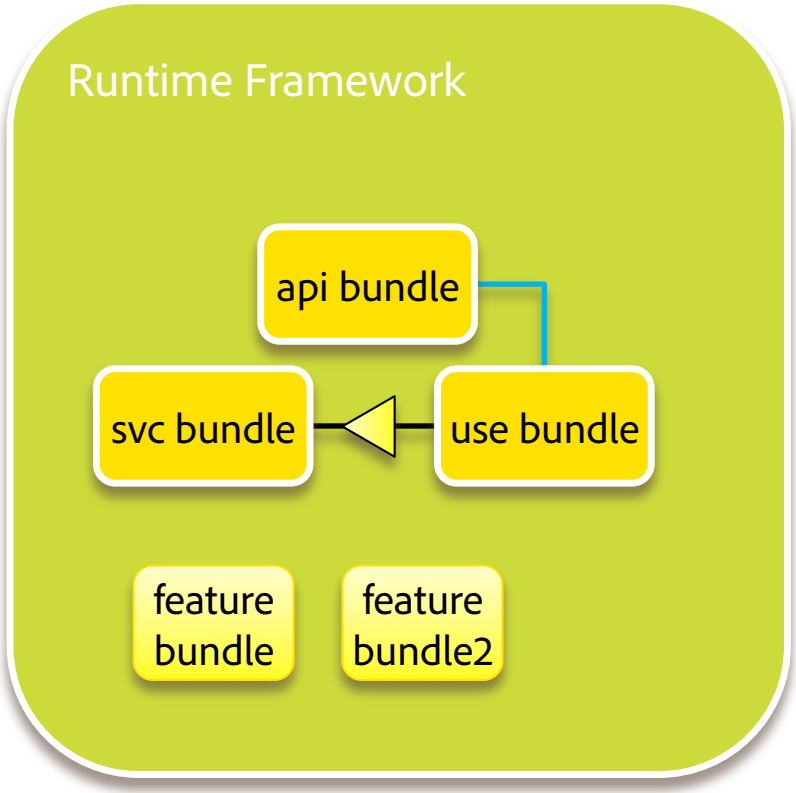
- Features – everything shared
- Applications – isolated
  - for ‘friendly’ multi-tenancy – keeps bundles from interfering with each other
- Composites – configurable isolation
  - generally useful for larger infrastructure components

*Chiot's Run*
*A Few Evenings of Work*
*flickr.com*

# Feature Subsystems

## Deployment artifacts

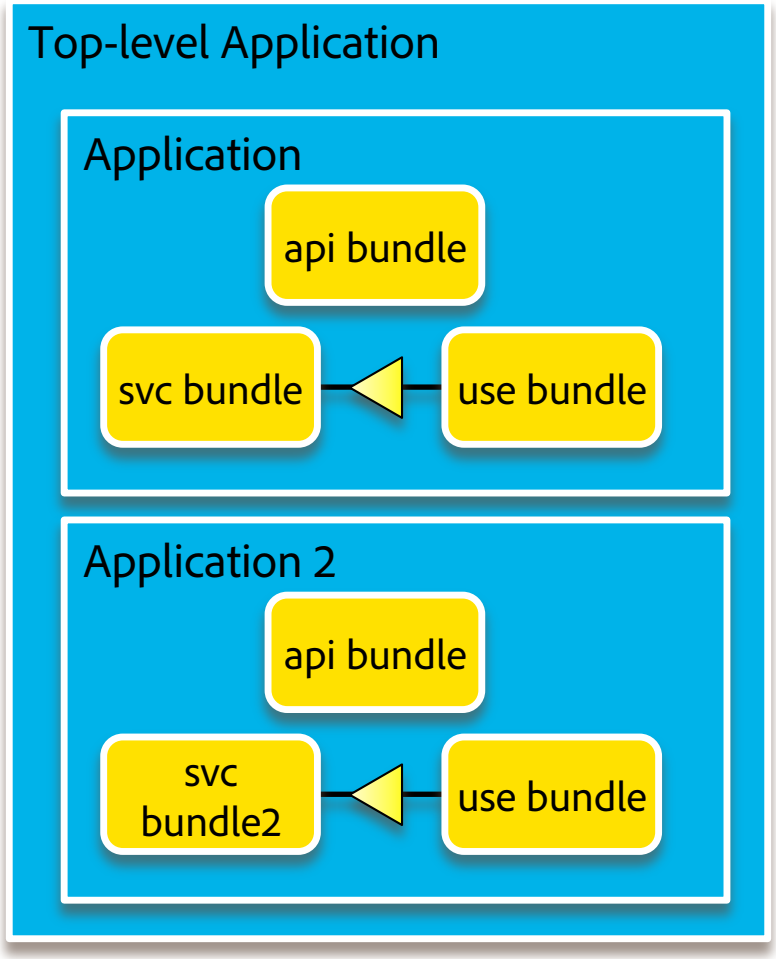


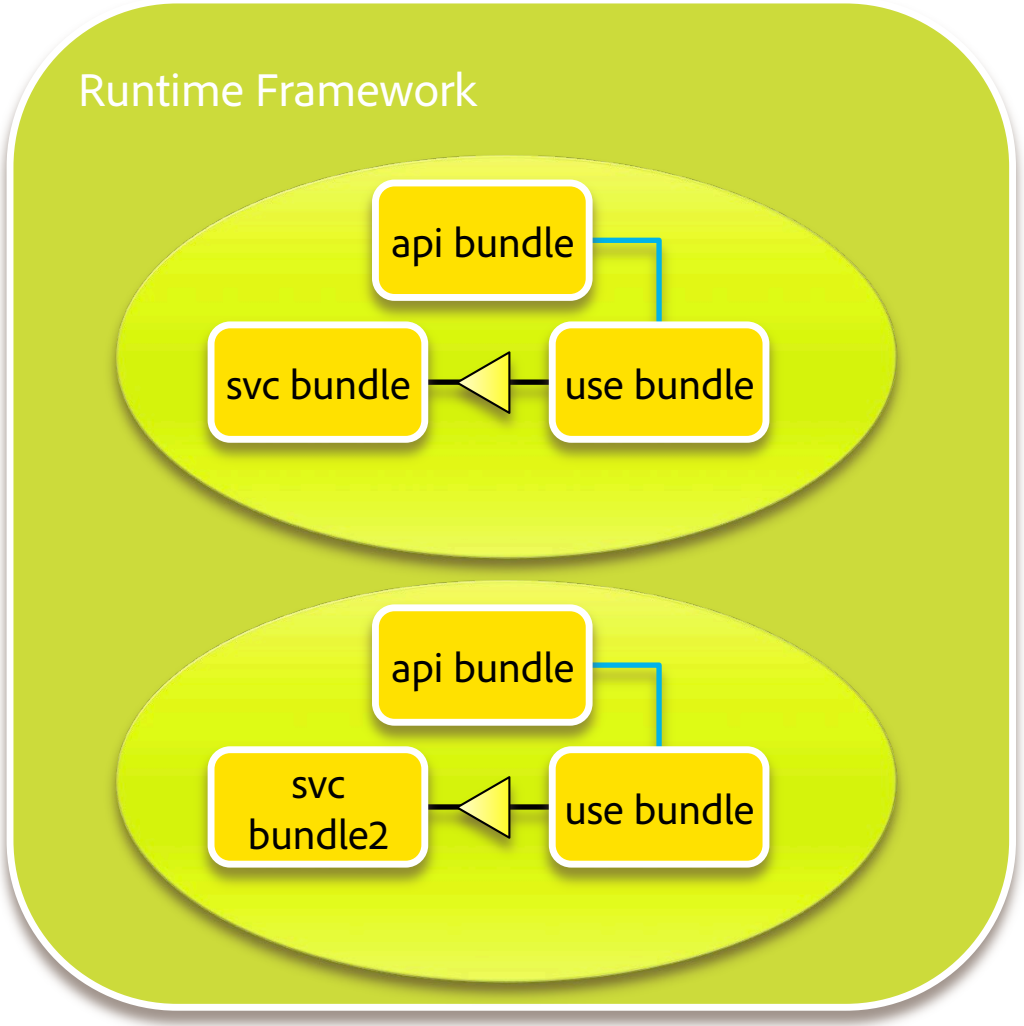




© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# Application Subsystems

## Deployment artifacts




© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# Build a subsystem

- Use maven:

```
<project>
 <artifactId>mysubsystem</artifactId>
 <packaging>esa</packaging>

 <dependencies>
  <!- the bundles you want in your subsystem -->
 </dependencies>

 <build>
  <plugins>
   <plugin>
    <groupId>org.apache.aries</groupId>
     <artifactId>esa-maven-plugin</artifactId>
     <extensions>true</extensions>
     <configuration>
      <generateManifest>true</generateManifest>
      <instructions>
       <Subsystem-Type>osgi.subsystem.feature
                                    </Subsystem-Type>
      </instructions>
     </configuration>
    </plugin>
```

to produce mysubsystem.esa

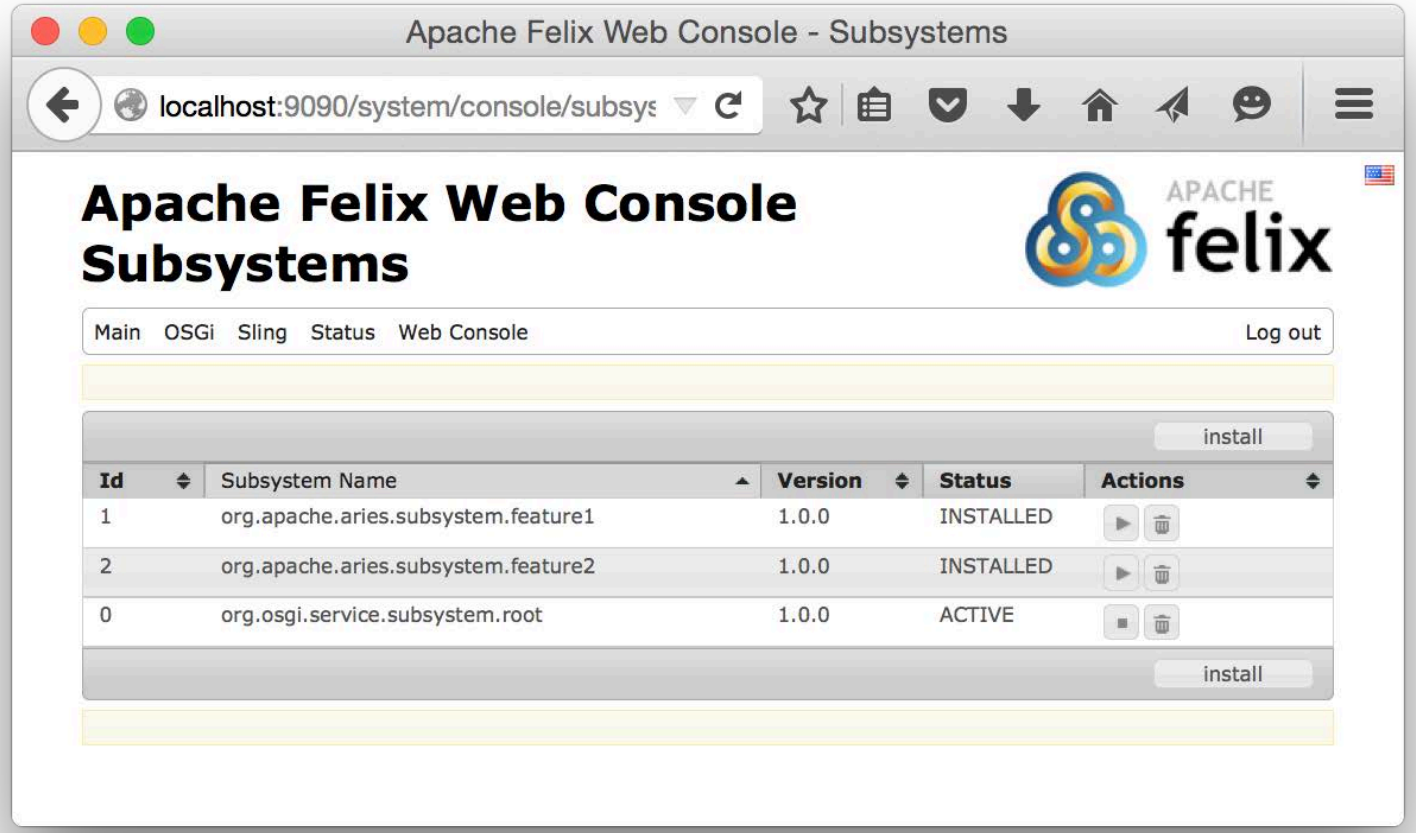
*"Sausage making-H-1" by László Szalai (Beyond silence) - Own work. Licensed under Public Domain via Wikimedia Commons*

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.



Adobe

# Deploy




https://svn.apache.org/repos/asf/felix/trunk/webconsole-plugins/subsystems

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

# More info on creating a subsystem

- OSGi Enterprise R6 spec: http://www.osgi.org/Download/Release6

- Apache Aries website http://aries.apache.org/modules/subsystems.html
- esa-maven-plugin
  http://aries.apache.org/modules/esamavenpluginproject.html

- For examples see:
  https://github.com/coderthoughts/subsystem-examples

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# Recipe – OSGi R6 Compendium

- OSGi Declarative Services (Chapter 112)
- OSGi Http Whiteboard Service (Chapter 140)
- OSGi Configuration Admin (Chapter 104)
- OSGi Metatype Service (Chapter 105)

Implementations from Apache Felix

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

# Q&(A)

© 2015 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.