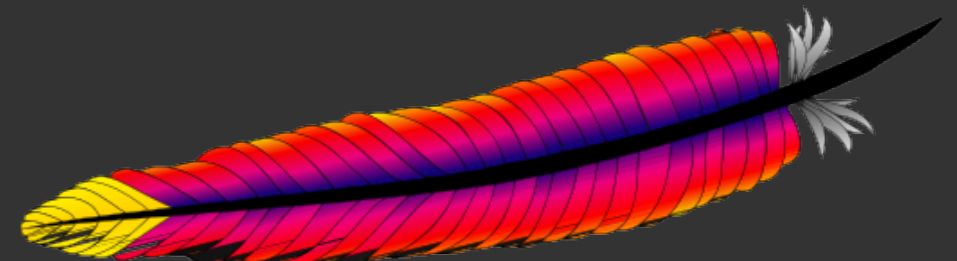


Apache Celix

Universal OSGi?

Alexander Broekhuis
alexander.broekhuis@luminis.eu



Introduction

- **Alexander Broekhuis**
 - Started as Java Engineer
 - Programming C since 2010
- **At Luminis since 2008**
 - Small Software House
 - Research and innovation oriented
 - Involved in Open Source (Apache)
- **Apache committer since 2010**

Agenda

- **Introduction**
- **History**
- **OSGi in C**
- **Remote Services**
- **Development**
- **Universal OSGi**
- **Status**



Introduction

- **OSGi specification implemented in C**
 - Adapted where needed
- **Focussed on Embedded Dynamic Systems**
- **Based on Apache Felix**
- **Remote Services**
 - Distributed Systems
 - Interoperability with Java

History

- **Started in 2010**
- **Middleware**
 - **Mixed Java/C environment**
 - **Based on (de-facto) standards**
- **Open Source**
 - **Generic Middleware -> No core business**
 - **Benefits others, but also benefits FROM others**

OSGi is:

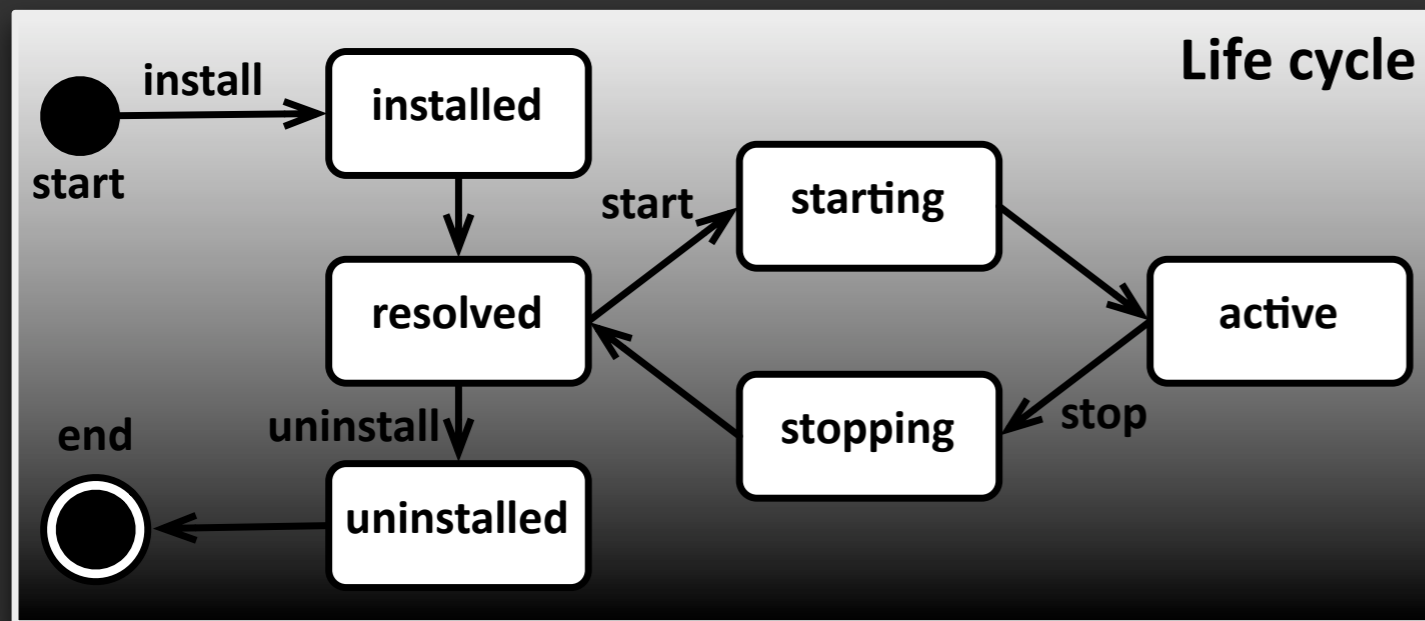
- **component based framework**
- **allows dynamic assembly of components**
- **Java, so independent of operating system**

OSGi technology is the dynamic module system for Java™

OSGi technology is Universal Middleware.

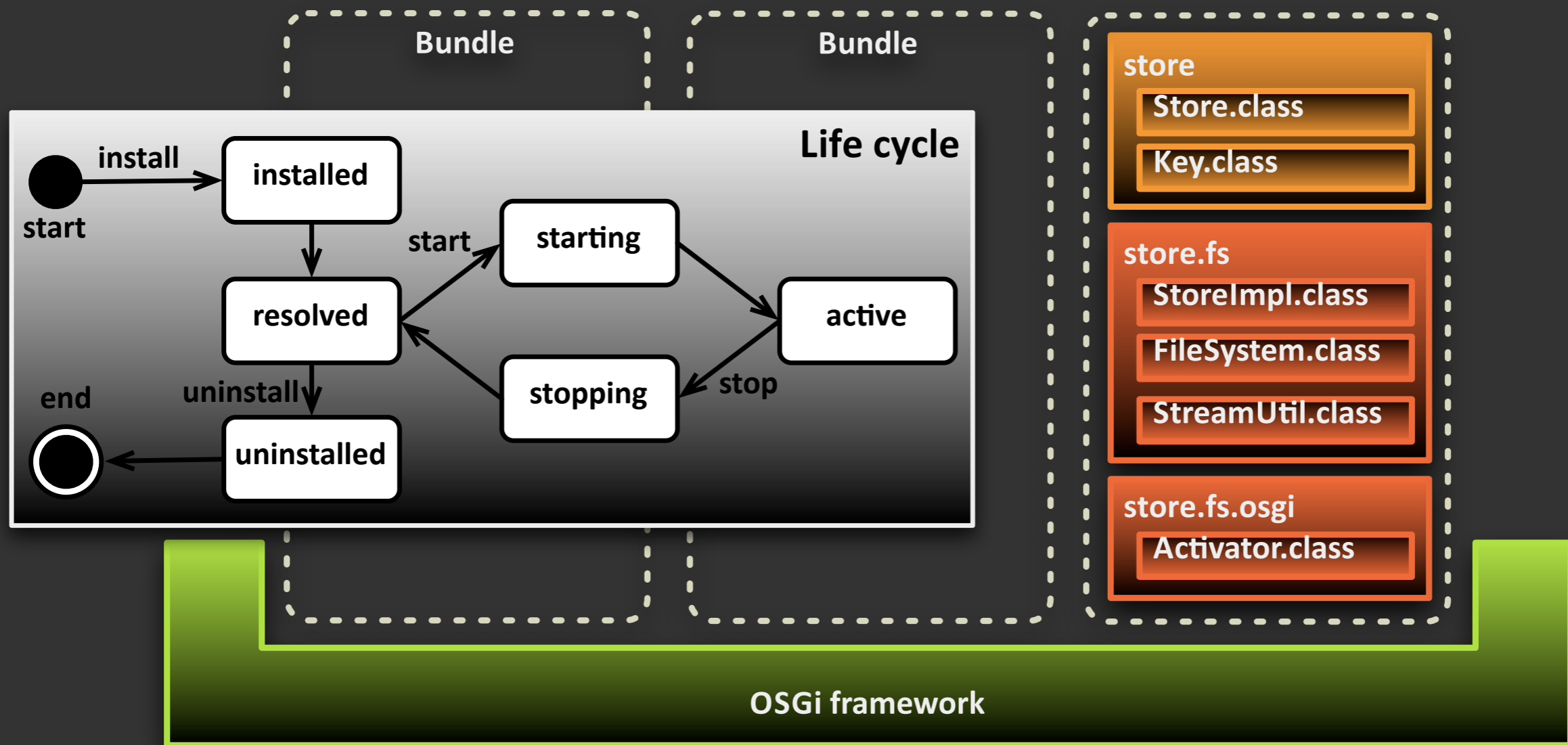
OSGi technology provides a service-oriented, component-based environment for developers and offers standardized ways to manage the software lifecycle. These capabilities greatly increase the value of a wide range of computers and devices that use the Java™ platform.

OSGi

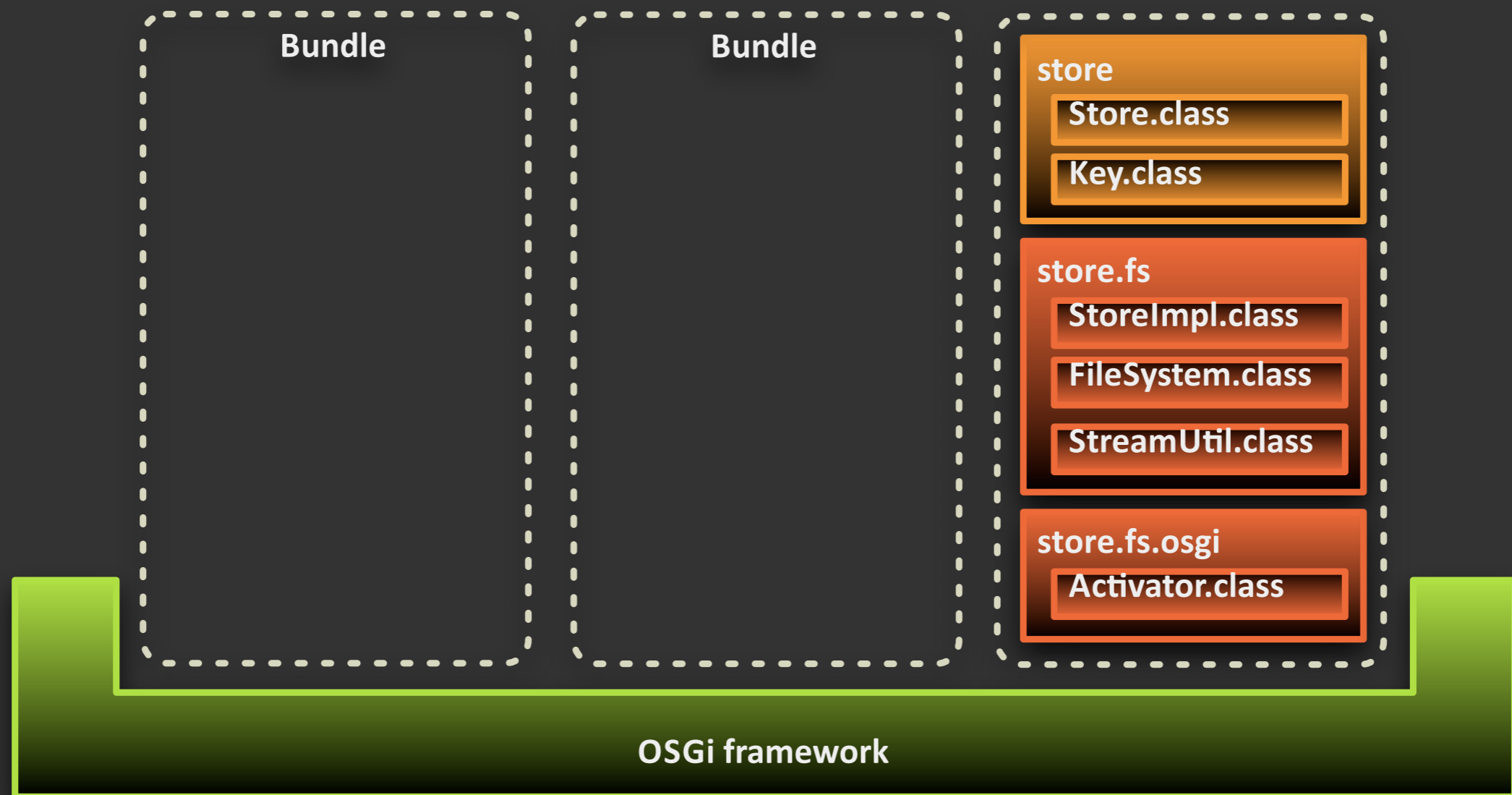


OSGi framework

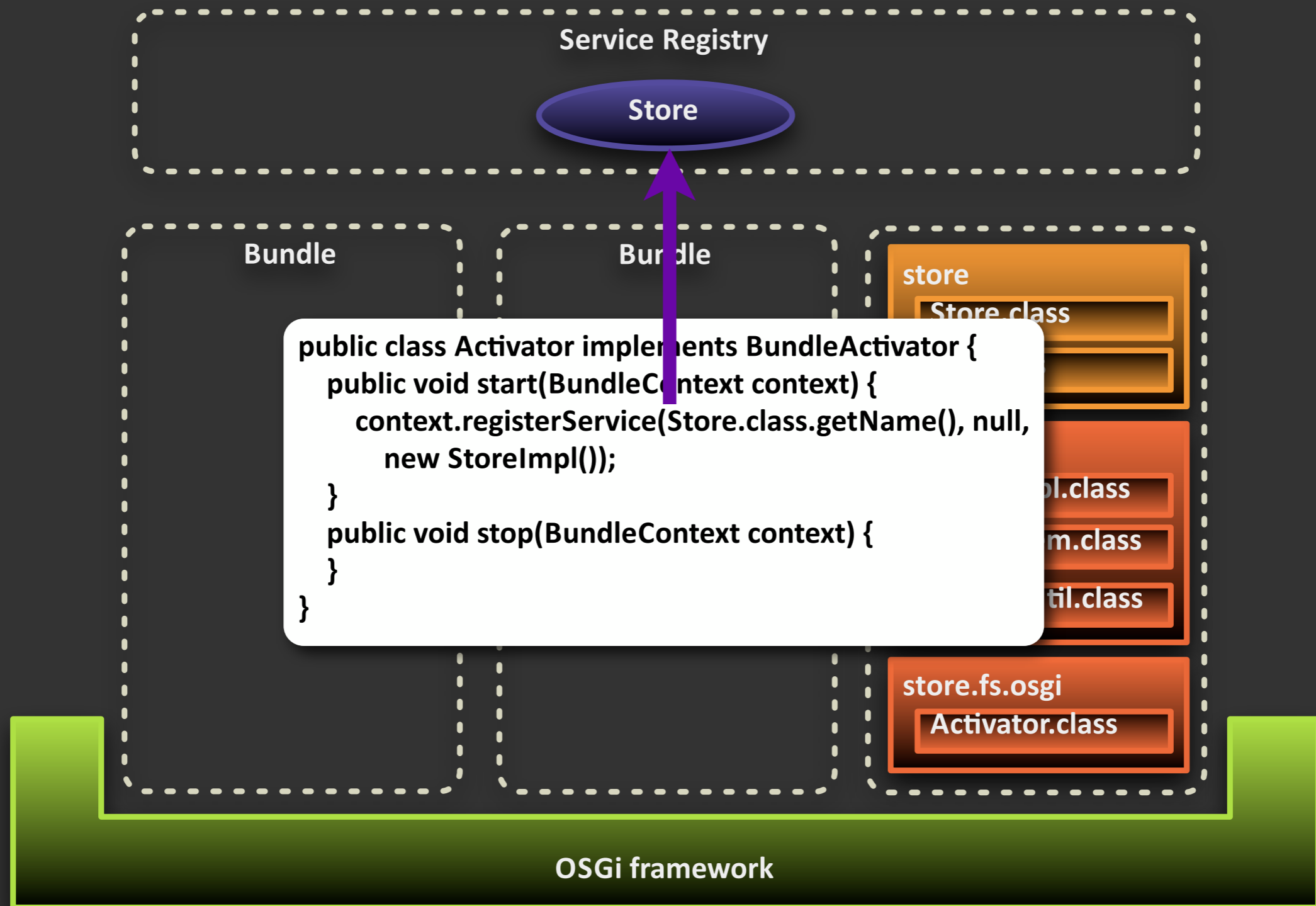
OSGi



OSGi



OSGi



Benefits

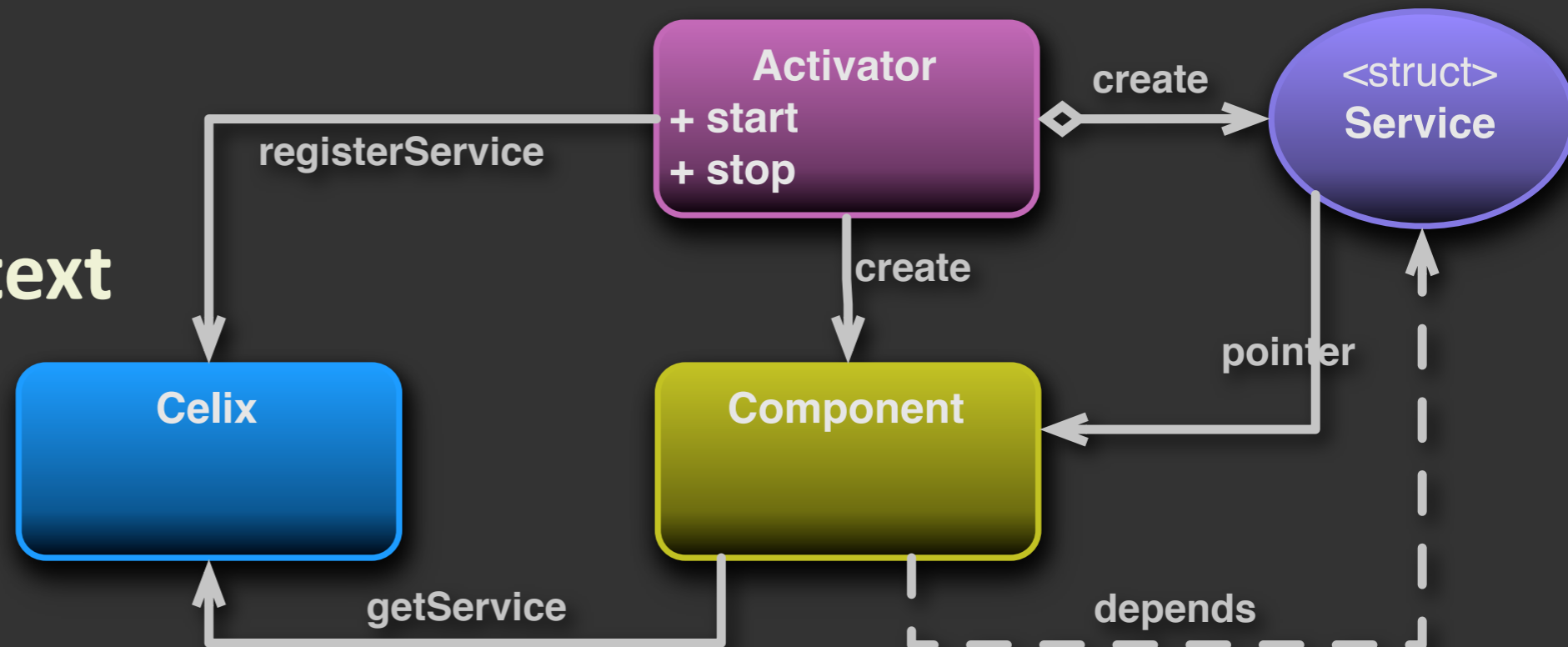
- **Making products with many variations**
- **Improving quality through re-use**
- **Speed: time to market**

OSGi in C

- **Follows the OSGi Specification**
 - **A Service Registry for tracking services**
 - **Bundle Activators for registering services**
 - **Bundle Lifecycle for tracking Bundle state**
- **Structs instead of Interfaces**
 - **Using function pointers to forward calls**

Services in C

- **Service**
 - Struct with function pointers
- **Activator**
 - start / stop
- **Celix**
 - bundleContext
 - registry
 - etc..



Services in C - BundleActivator

Java - BundleActivator

```
@Override  
public void start(BundleContext context) throws Exception {  
    System.out.println("Hello " + m_word);  
}
```

C - BundleActivator

```
celix_status_t bundleActivator_start(void * userData, BUNDLE_CONTEXT context) {  
    struct userData * data = (struct userData *) userData;  
    printf("Hello %s\n", data->word);  
    return CELIX_SUCCESS;  
}
```

Services in C - Service registration

Java - Register service

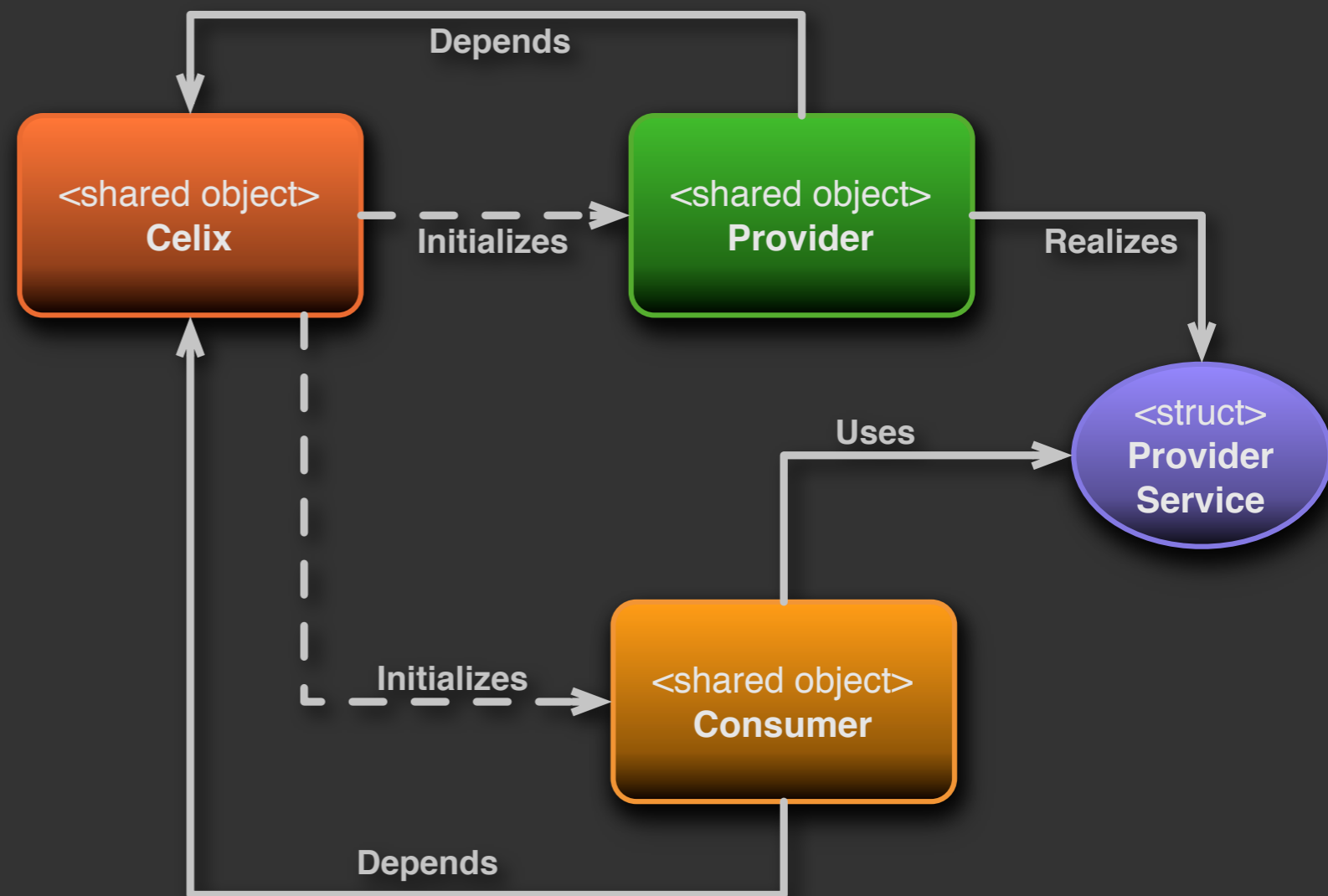
```
context.registerService(  
    ShellService.class.getName(), new ShellServiceImpl(), null);
```

C - Register service

```
SHELL_SERVICE shellService = (SHELL_SERVICE) apr_palloc(pool, (sizeof  
(*shellService));  
shellService->shell = shell;  
shellService->getCommands = shell_getCommands;  
shellService->getCommandDescription = shell_getCommandDescription;  
shellService->getCommandUsage = shell_getCommandUsage;  
shellService->getCommandReference = shell_getCommandReference;  
shellService->executeCommand = shell_executeCommand;  
  
SERVICE_REGISTRATION registration = NULL;  
status = bundleContext_registerService(  
    context, SHELL_SERVICE_NAME, shellService, NULL, &registration);
```

Dynamic Assembly

- Dynamic Loading
 - Library
 - Using dlfcn.h
 - dlopen / dlclose
 - dlsym

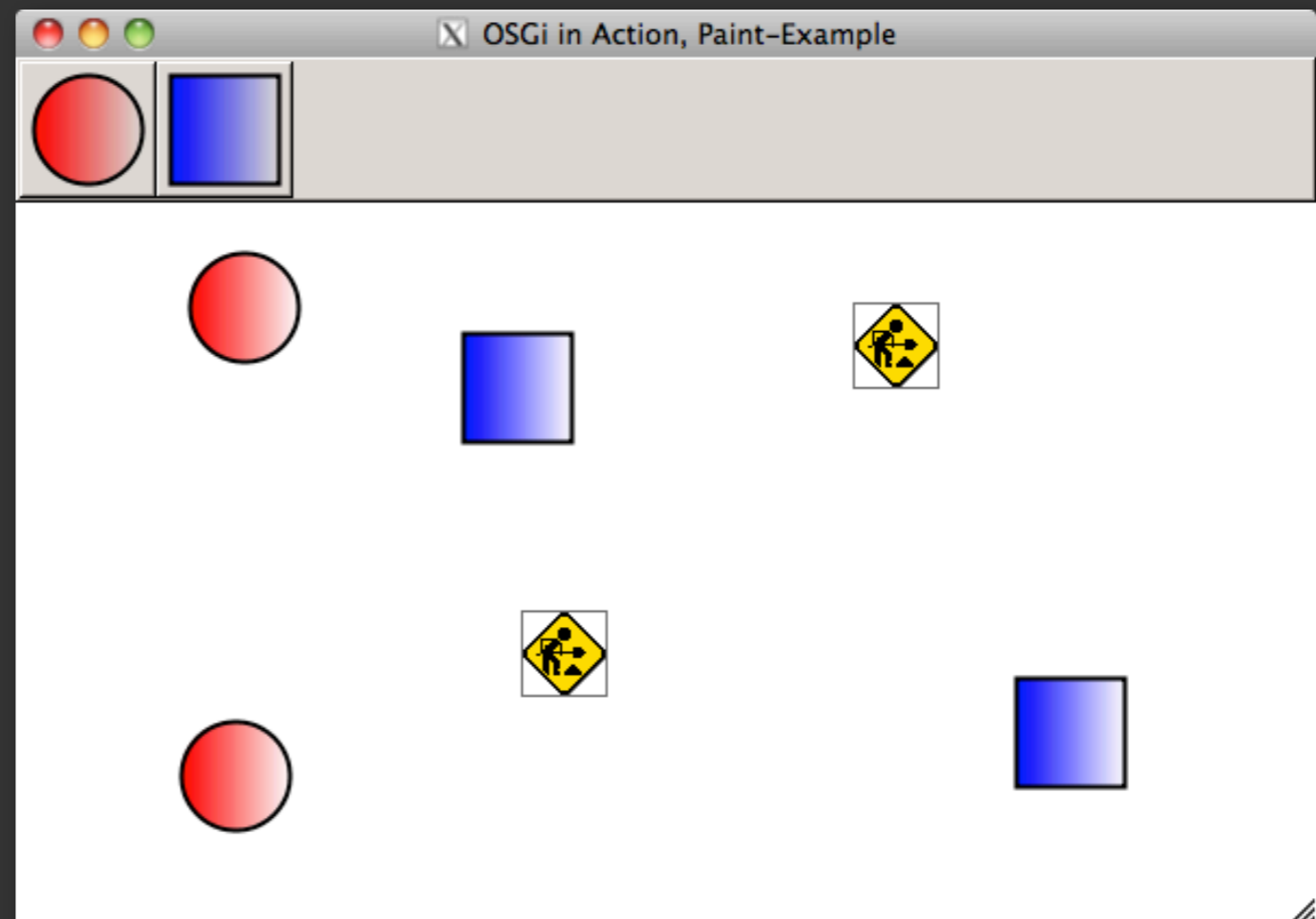


Deployment

- **Bundling**
 - **Zip file with Shared Object**
 - (.so, .dll, .dylib, ...)
 - **Manifest**
 - **Import / Export**
- **Runtime**
 - **Extracted to cache**

Demo

- **Paint Application**
 - From OSGi in Action
- **Install / Start / Stop bundles**

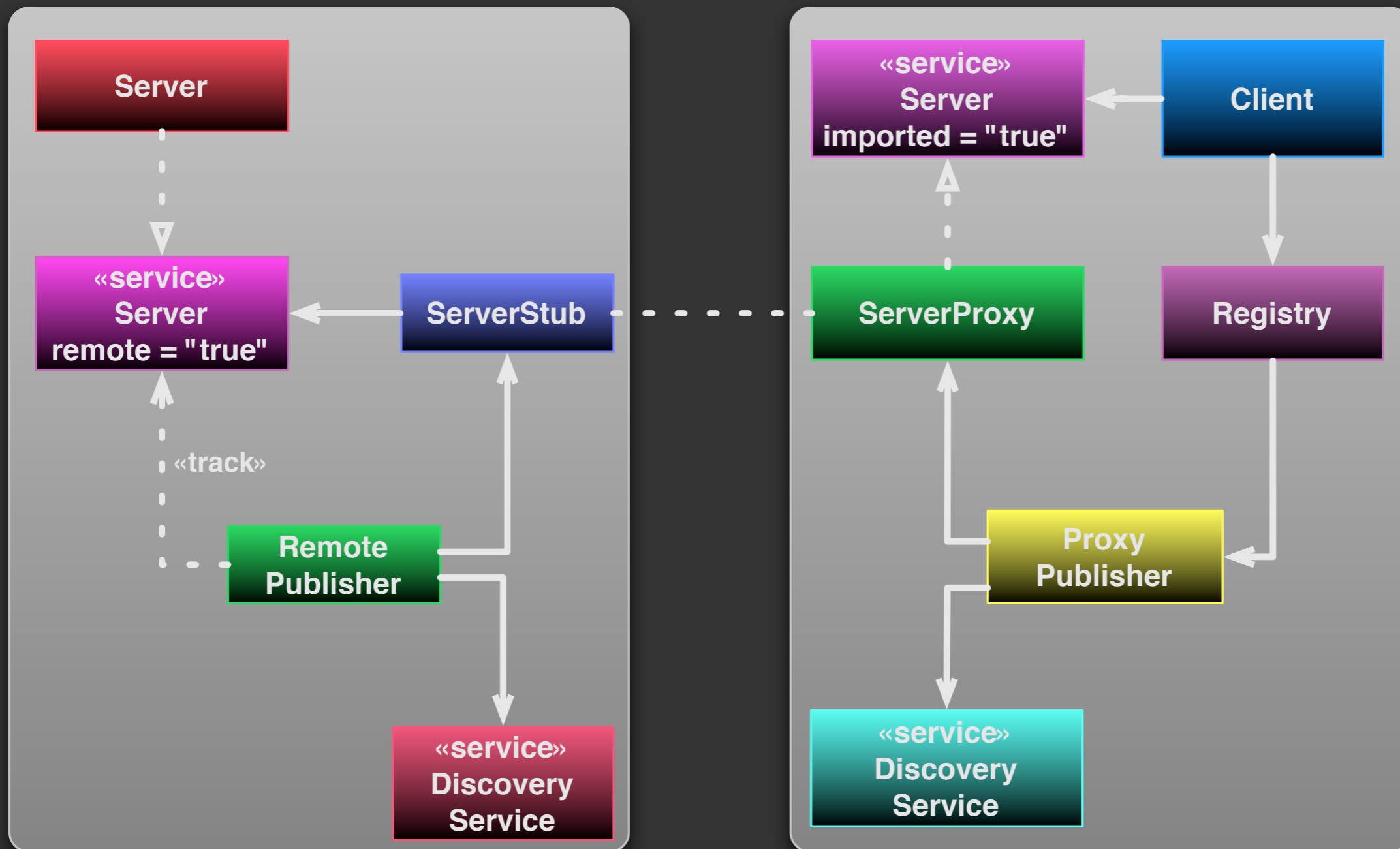


Remote Services

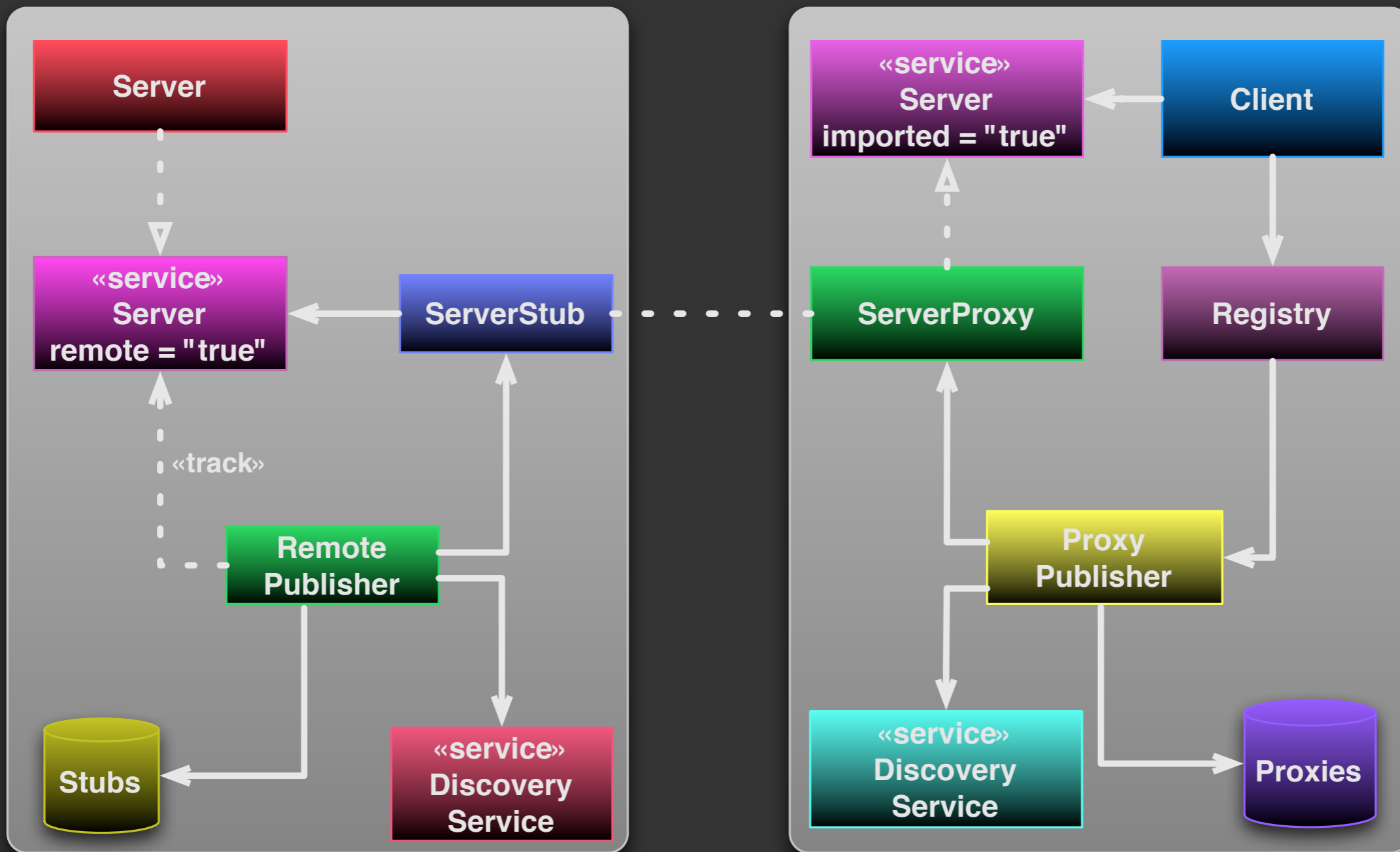
- **Remote Service Admin**
 - Enterprise Specification
- **Interoperability**
- **Protocol**
 - SOAP?
 - REST/JSON
 - Hessian



Remote Services



Remote Services



Remote Services in Celix

- **REST/JSON**
 - Basic implementation working
- **Hessian**
 - Basic (de)serialization implementation made
- **Todo**
 - Generate code for endpoints (stubs/proxies)
 - Test/Implement interoperability with Java

Demo - Remote Services

- **Calculator Example Service**

```
struct example_service {  
    example_t example;  
    celix_status_t (*add)(example_t example,  
                          double a, double b, double *result);  
    celix_status_t (*sub)(example_t example,  
                          double a, double b, double *result);  
    celix_status_t (*sqrt)(example_t example,  
                           double a, double *result);  
};
```

- **Calculator Shell Commands**

- add / sub / sqrt

- **Endpoints using REST / JSON**

- Embedded Mongoose webserver

Demo - Remote Services

- **Calculator Example Service**

```
struct example_service {  
    example_t example;  
    celix_status_t (*add)(example_t example,  
                          double a, double b, double *result);  
    celix_status_t (*sub)(example_t example,  
                          double a, double b, double *result);  
    celix_status_t (*sqrt)(example_t example,  
                           double a, double *result);  
};
```

- **Calculator Shell Commands**

- add / sub / sqrt

- **Endpoints using REST / JSON**

- Embedded Mongoose webserver

Platform Support

- **Apache Portable Runtime**
 - **Linux, Unix, Windows, etc**
 - **Memory pool**
 - **File handling**
 - **Threads**
- **No RealTime/Embedded**
 - **Reuse APR API**

Building



- **CMake**
 - **Cross platform / Multi IDE**
- **Framework**
 - **Standard library**
- **Bundle macros**
 - **Create bundle zip**
 - **Add library and resources**

Universal OSGi

- **RFP-89**
 - Proposed to the mailing list in 2007
 - Since then remained silent
 - Ongoing effort to pick up again
- **Supported Languages**
 - C++ / .NET language (C#) / ECMAScript
 - What about C?
- **Interest from Community?**

Universal OSGi - Celix

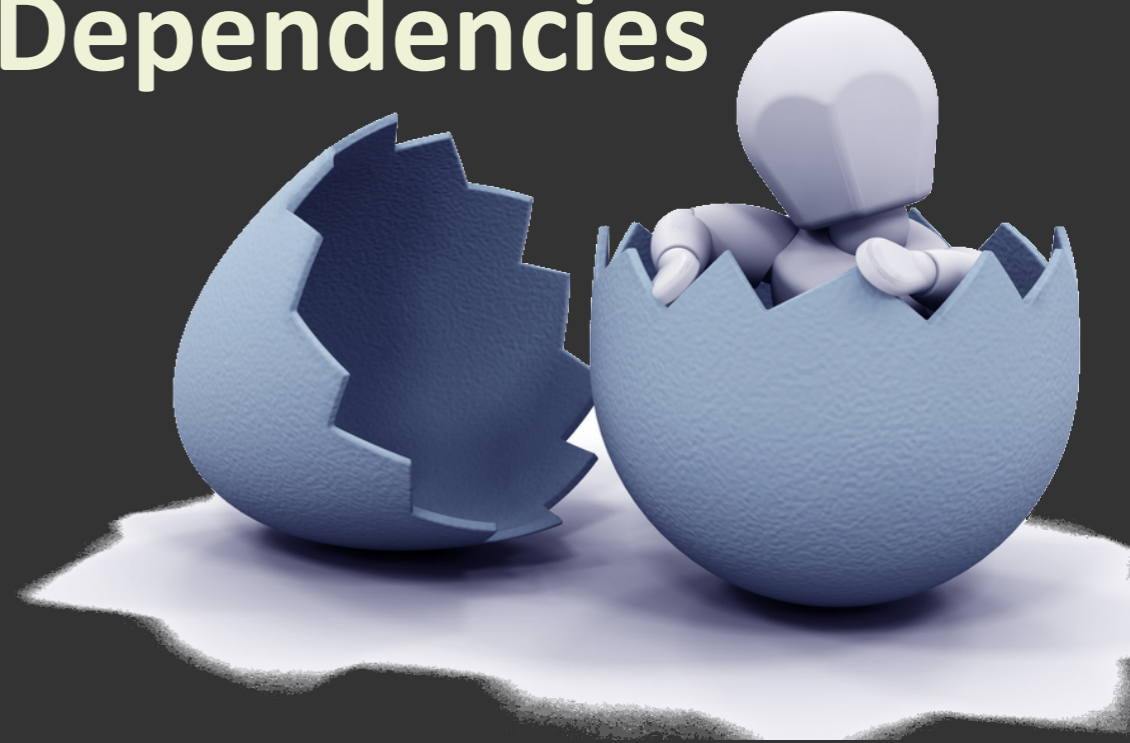
- **Native port in C**
- **Status**
 - **Based on the R4 Specification**
 - **Deployable bundles**
 - **Bundle lifecycle**
 - **Service registry**
 - **Framework API**
 - **Standards for IPC and communications**
 - **Remote Services**

Status - Framework

- **Dynamic Loading of Libraries**
- **Service Registry for Querying and Registering**
- **Bundles with Manifest for Deployment**
- **Service Tracker for tracking Services**
- **Bundle State and Life Cycle**
- **Bundle Cache for Storing State**

Status - Extras

- **Shell Implementation for Inspecting Bundles**
 - `ps/start/stop/install/update/uninstall` commands
- **Dependency Manager for Dependencies**
 - Simple Implementation
 - Static Linked
- **Open Source at Apache**
 - Currently in the Incubator



Apache Celix

- **Roadmap**
 - ...
 - **Deployment Admin**
 - **Remote Services**
 - **Configuration Admin**
 - **Event Admin**
- **Looking for**
 - **Users / Testers / Committers**

Links

- **Original Proposal:**
 - <http://wiki.apache.org/incubator/CelixProposal>
- **Incubator Mailing List Archive:**
 - <http://incubator.markmail.org/search/?q=Celix>
- **Incubation Status:**
 - <http://incubator.apache.org/projects/celix.html>
- **Project Page:**
 - <http://incubator.apache.org/celix/>