# OSGi - the Dynamic Module System for Java

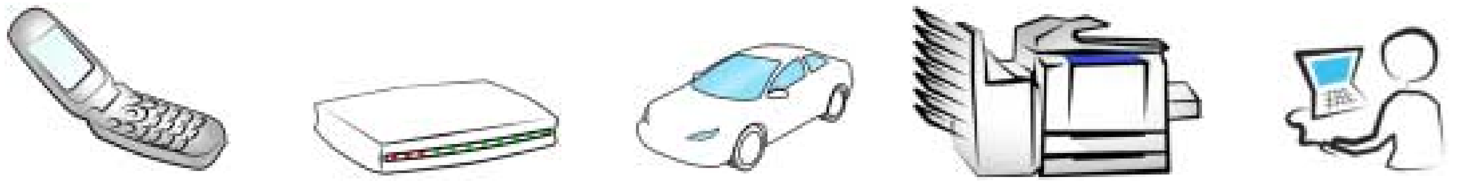
Christer Larsson
VP EMEA OSGi Alliance
CEO Makewave

makewave

## Innehåll

# Makewave (formerly Gatespace Telematics)

- Taking middleware forward



| Nomadic Device | Residential | Telematics | Industrial | Software Developers |
|---|---|---|---|---|
| SmartPhone<br>PDA | DSL<br>Set-top<br>IPTV<br>HomeHub | TCU<br>Navigation system<br>Diagnostics | M2M<br>Equipment monitoring | Software protection<br>&<br>License Management |

**Universal Middleware**

OSGi Platform – (Knopflerfish, Java, Ubiserv, Ubicore), License Management – (Lime)

**Global Services**

Design, Implementation & Integration,Workshops, Seminars and Training
Expertise in Java, Embedded Systems, Compilers, .Net, License Mgmt)

## Makewave Short facts

- Company Facts
  - OSGi member since 1999
  - Located in Gothenburg, Sweden
  - Highly educated personnel, all with PhD or Master degree in engineering and computer science
  - Privately owned
- Management
  - Christer Larsson, CEO, co-founder Makewave, board member of Telematics Valley, VP EMEA of OSGi Alliance.
  - Staffan Truvé, Chairman, founder of Gatespace, CEO of SICS and Interactive Institute
  - Per Gustafson, CTO, co-founder Makewave, OSGi expert group repr.
- Subsidiaries
  - Secureon AB, fully owned, License Management System

## Knopflerfish Project

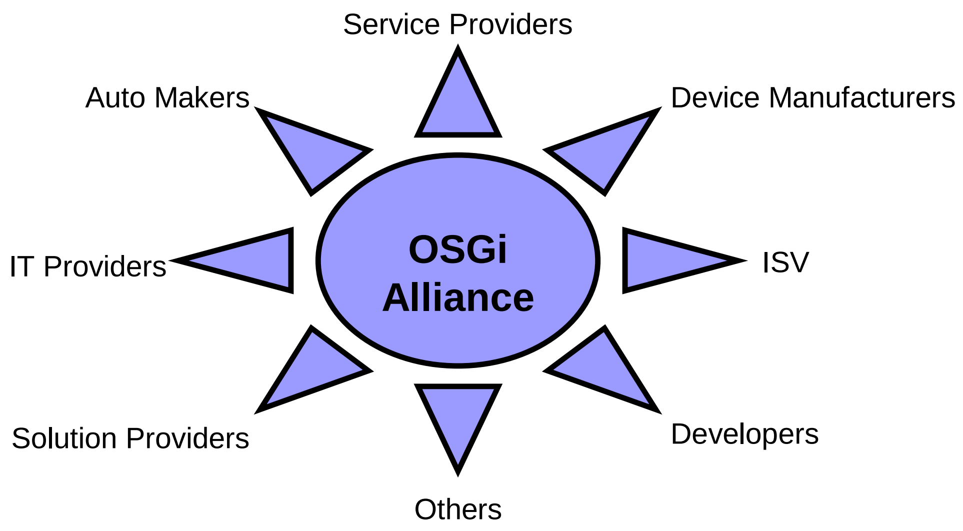
- The Knopflerfish project is a open source implementation of the OSGi Specifications
  - Maintained by Makewave
  - www.knopflerfish.org
  - BSD style licence, free for all use, no run-time licences
  - Implements OSGi R4
- Knopflerfish Pro is the fully certified and supported version for commercial use
  - OSGi Certified
  - Comes with Support Agreements (SLA)
  - Expertise, training and consulting services

## Introduktion till OSGi

- Några ord om Makewave
- Introduktion till OSGi
- Lite om historien bakom OSGi
- Genomgång av OSGi teknologin
- Exempel på hur OSGi används
- Ytterligare fördjupning

## The OSGi Alliance

- The OSGi Alliance is an independent non-profit corporation comprised of technology innovators and developers and focused on the interoperability of applications and services based on its component integration platform.
  - Established in 1999, members worldwide
  - Membership spans many industries
    - Residential, Automotive, Industrial, Mobile, Enterprise
  - Membership information at www.osgi.org

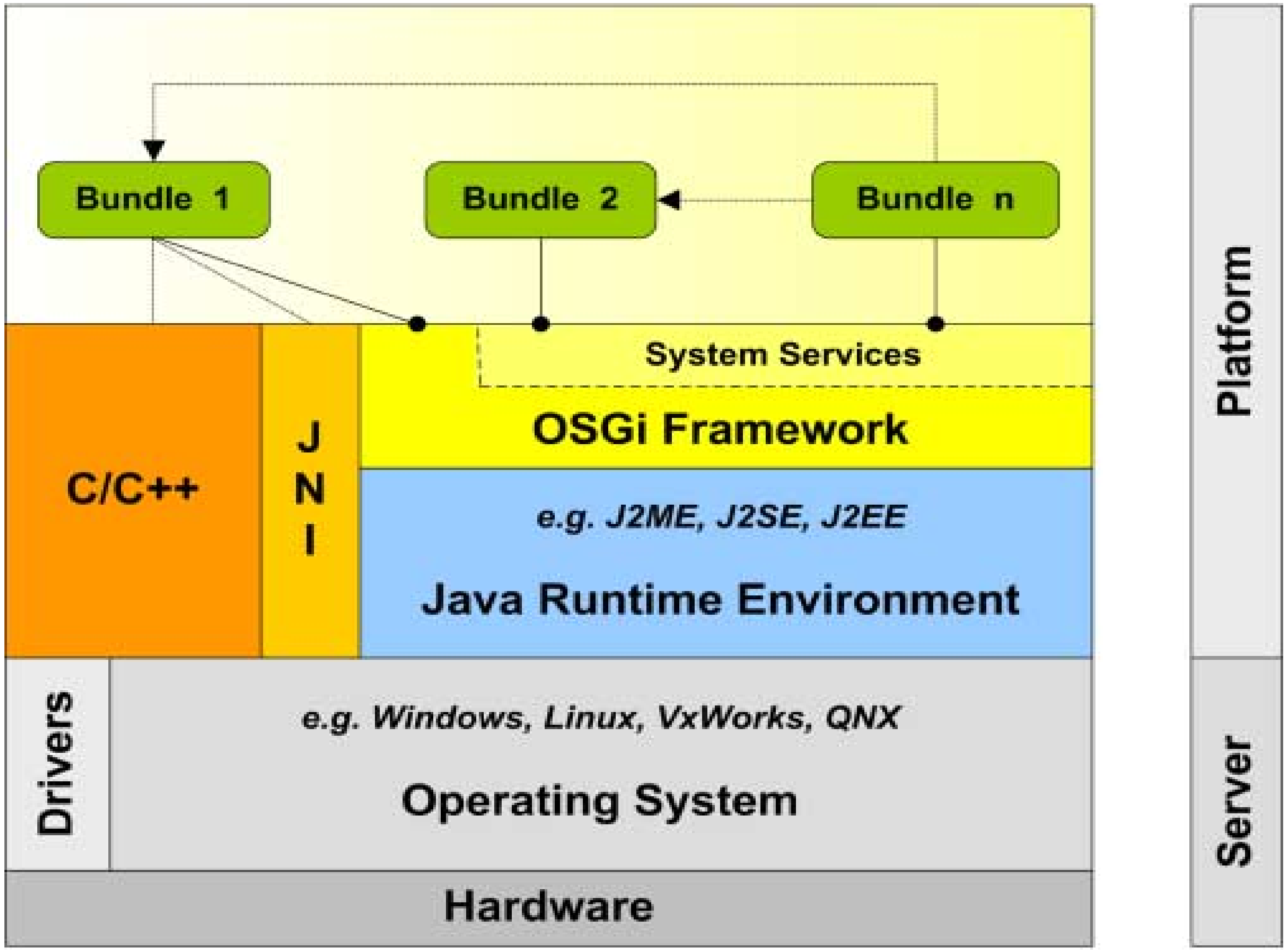


## What is OSGi Technology?

- OSGi technology is a dynamic module system for Java™
- OSGi technology is component-based
- OSGi technology is service-oriented
- OSGi offers standardized ways to dynamically manage the lifcycle of software on the Java™ platform
  - updates of software no longer require a restart of the JVM!

- OSGi technology is *Universal Middleware*

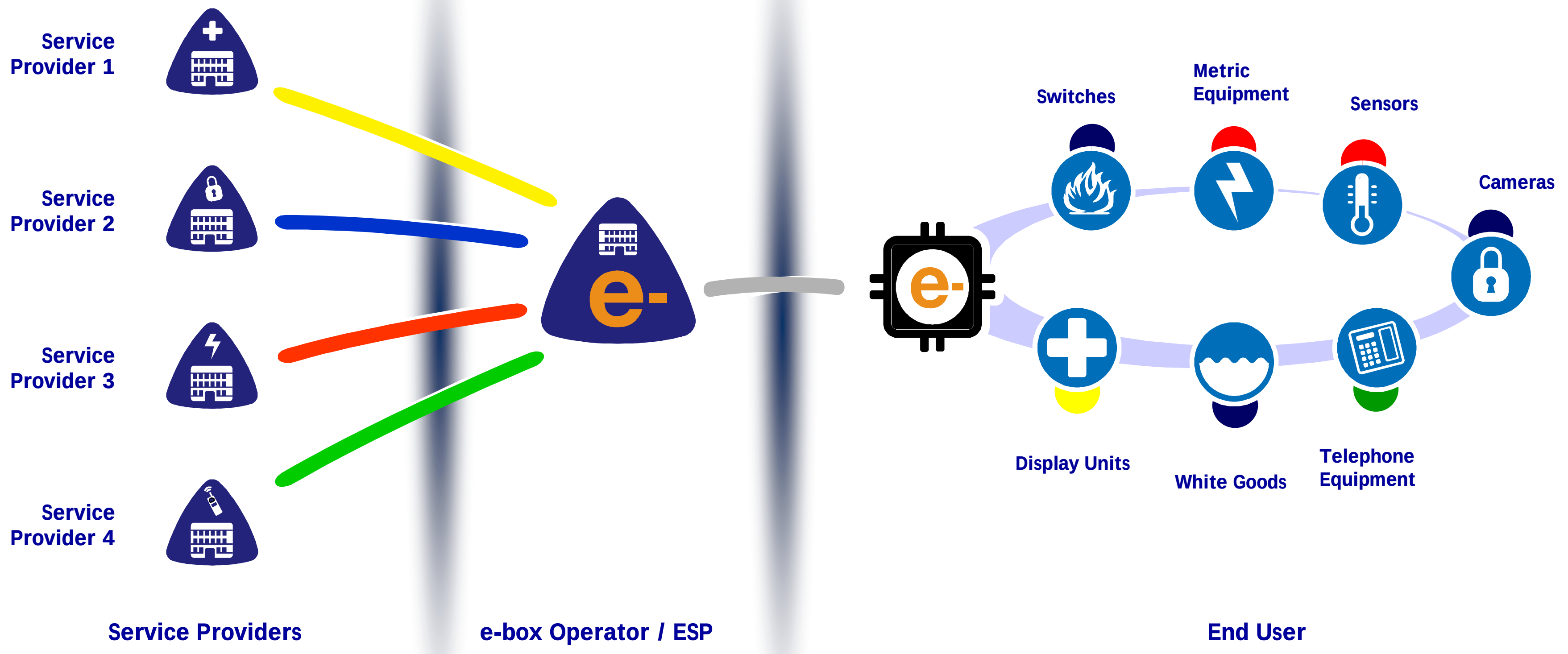
## OSGi Architecture



## Lite om historien bakom OSGi

- Några ord om Makewave
- Introduktion till OSGi
- Lite om historien bakom OSGi
- Genomgång av OSGi teknologin
- Exempel på hur OSGi används
- Ytterligare fördjupning

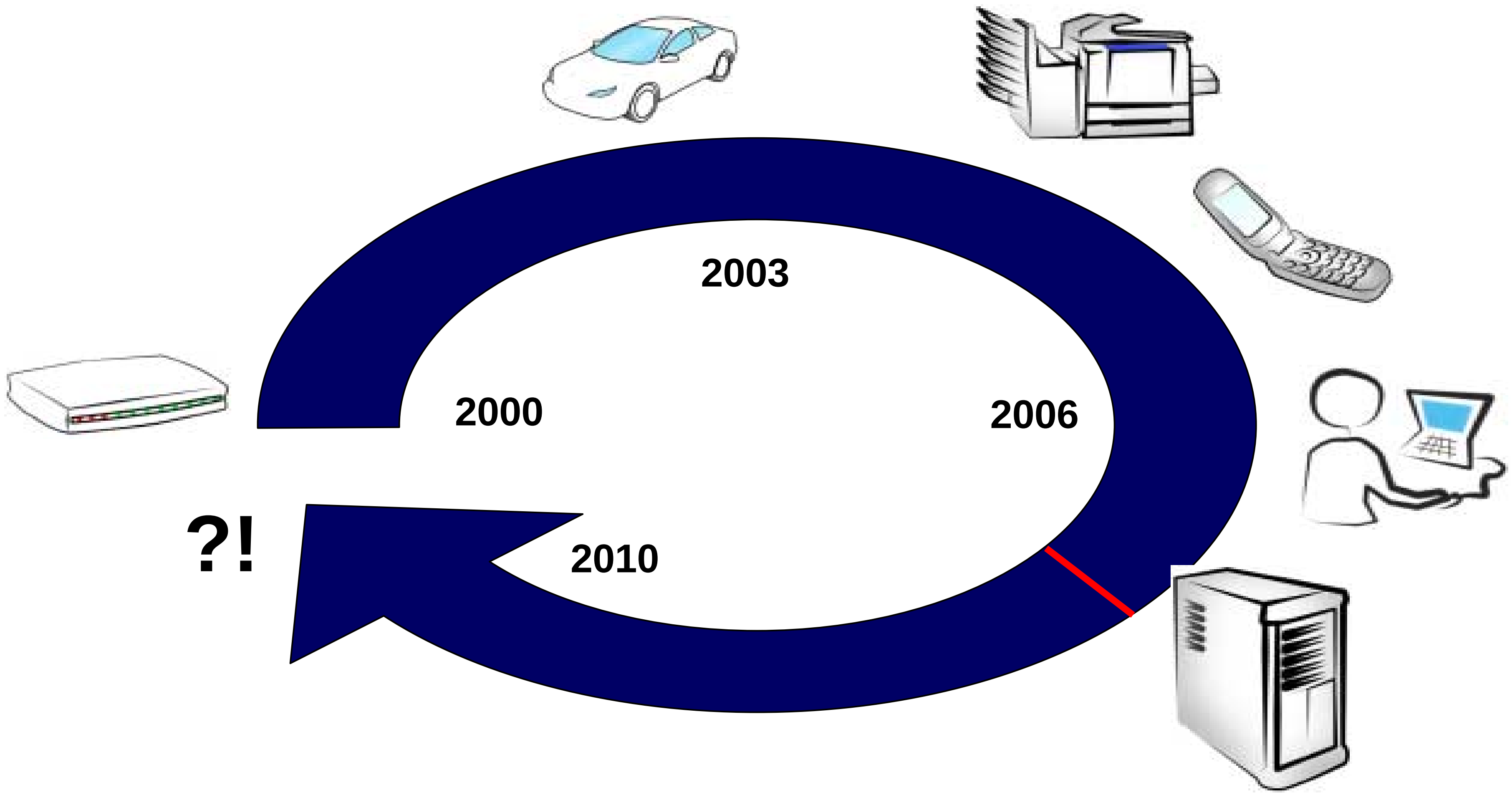
# Ericsson e-services Architecture / Business Model (1998/99)



## One slide from the OSGi Congress 2003:

- As usual, new technology takes more time to get adopted than the enthusiasts want to believe:
- Example:
  - First 802.11 work group meeting was held in 1990,
    broad market acceptance was not until around 2001
- Corollary:
  The OSGi World Congress in 2009 will be spectacular! :-)

## The evolution of the OSGi technology



## The residential OSGi market

- Three reasons why the residential OSGi market can resurrect:
- BOM for a R/G is now 1/6 of the price
- Wireless devices / sensors
- Broadband penetration

## The e-ebox set the original requirements

- In the e-box we wanted a:
  - Platform
  - Java based
  - Load software / services / applications dynamically
  - Multiple Service Providers can co-exist
    - Integrity between applications, no sneaking
  - Remotely managed
- And the result was OSGi R1 !
- So the platform designed for the intelligent fridge is now becoming a corner stone in JEE app servers.
  - Without any major makeover / redesign

## Genomgång av OSGi teknologin

- Några ord om Makewave
- Introduktion till OSGi
- Lite om historien bakom OSGi
- Genomgång av OSGi teknologin
- Exempel på hur OSGi används
- Ytterligare fördjupning

## What is OSGi Technology?

- OSGi Technology is currently based on Java
- Java provides a high degree of platform independence
  - Abstracts Hardware Architectures
  - Abstracts Operating Systems
- JNI links Java and the native platform
- Java lacks a good module/component system!
- Java lacks dynamism - updates of software require a restart of the JVM!

## What is OSGi Technology?

- OSGi technology is a 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.
- Addresses the lack of a good module system and lack of dynamism in the Java™ platform

- *OSGi – Class Loaders on Steroids*

## OSGi Remote Manageability

- OSGi is specifically designed with an eye towards remote management of networked services
- OSGi is management protocol agnostic – meaning it fits in several different markets that each prefer their own protocols
- This is achieved by providing access to neccesary resources and functionalities inside the framework to Management Agents
- Management Agents implement management protocols

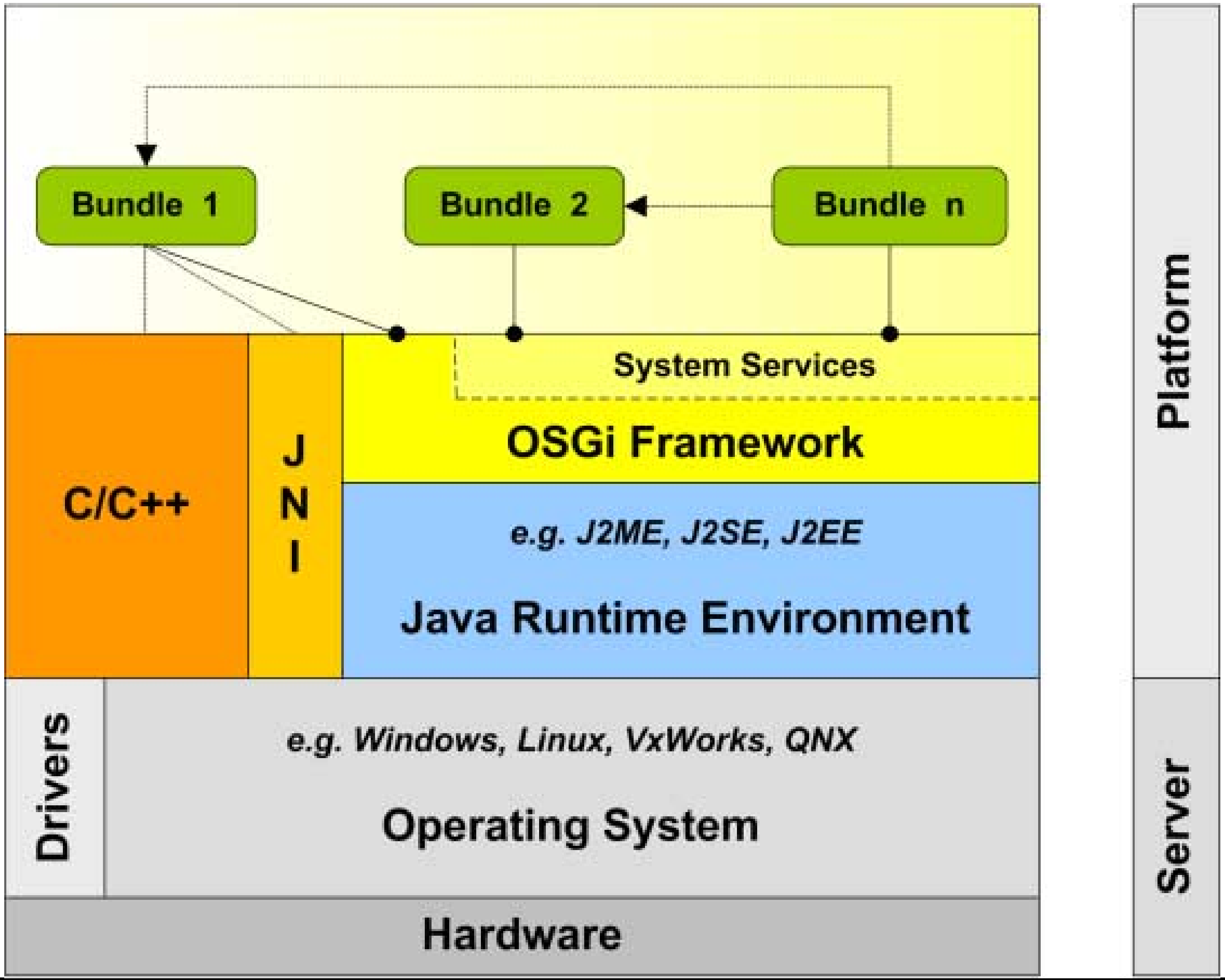
## What does OSGi specify?

- Core - The framework and system services
  - A general purpose middleware software platform
  - Service-oriented, component-based
  - Security and platform management related system services
- A Compendium of Services
  - Different subsets of these services are relevant for different markets
  - A subset can be formalized as a profile E.g. Mobile Profile, Vehicle Profile

## Where is OSGi used?

- It can be used almost everywhere Java is used
- Used in desktop applications
  - Eclipse
  - Lotus Notes
- Used in embedded environments
  - Mobiles
  - Printers
  - Telematics
- Used on server side
  - Enterprise Application containers (J2EE)

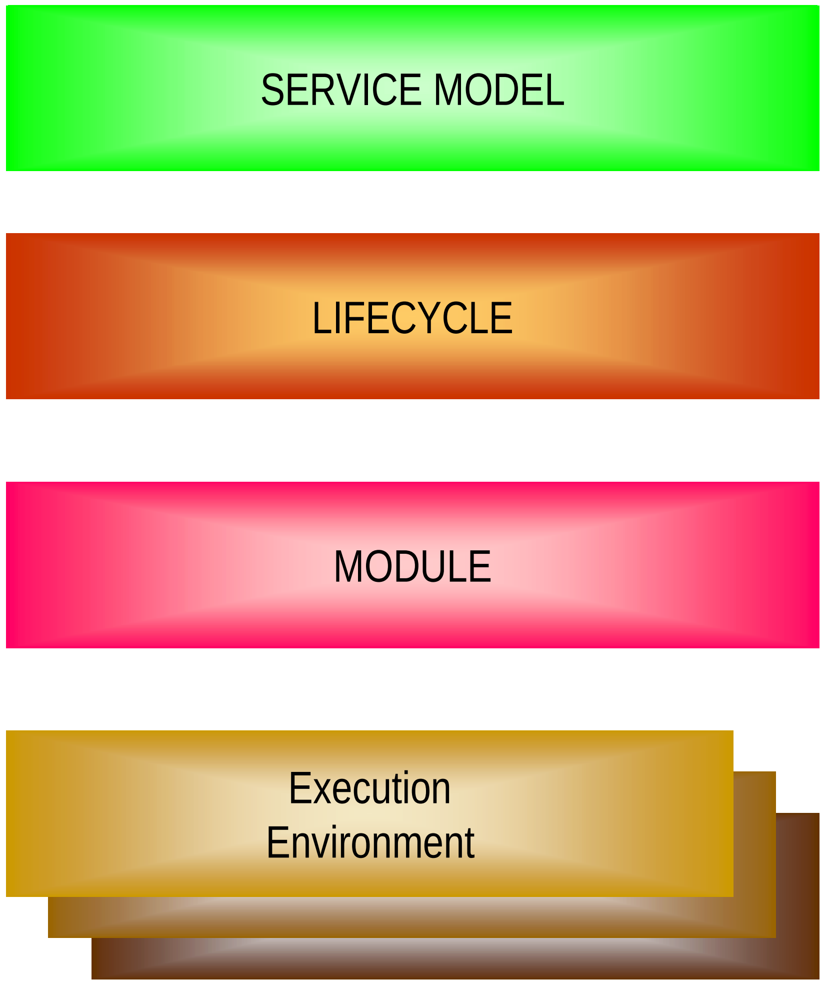
## Software Architecture



## The OSGi framework

- The base - runs multiple applications and services
- Single VM instance
- Separate class loader per bundle
  - Class loader network
  - Independent namespaces
  - Class sharing at the Java package level
- Lifecycle management
  - Install, Start, Stop, Uninstall, etc.
- Versioning
  - Multiple versions of same packages/classes can co-exist
- Intra VM publish/find/bind service model
- Operational Control
- Java Permissions to secure framework

## Framework Layering



**L3** – Provides a publish/find/bind service model to decouple bundles

**L2** - Manages the lifecycle of bundle in a bundle repository without requiring the VM be restarted

**L1** - Creates the concept of modules (aka. bundles) that use classes from each other in a controlled way according to system and bundle constraints

**L0** -
•OSGi Minimum Execution Environment
•CDC/Foundation
•J2SE

## Concepts of OSGi - bundles

- Bundles
  - A plain jar-file with a special format manifest file
  - Java code
  - Resources
  - Native code
  - Internationalization
  - Loaded into the framework

## Concepts of OSGi - collaboration

- Bundle collaboration
    - Sharing Java packages
        - Exporting packages
        - Importing packages
    - Services
    - Ways to get notified with bundles are installed, started, etc
    - Isolation, own name space per bundle

## Concepts of OSGi - class loaders

- Each bundle has its own class loader to:
  - Provide integrity beween bundles
  - Avoid namespace collisions
- A bundle class loader can only load classes from:
  - The system class loader
  - Imported packages
  - The bundle’s jar file

## Concepts of OSGi - class loaders

- A bundle can export Java packages.
  - Framework determines the actual exporter if several bundles offer the same package.
  - The exporter with the highest specification version for the package will be the first candidate.
  - A bundle with no specification version will be the last export candidate.
- A bundle can import Java packages from other bundles.
  - If the package is not exported, the importing bundle cannot be resolved and is not started.
  - To use classes imported from another bundle, use the regular import clause in the Java code.
- Imports and exports are stated in a bundles manifest.

## Concepts of OSGi - Manifest file

```
Manifest-Version: 1.0
Ant-Version: Apache Ant 1.6.5
Created-By: 1.4.2_11-b06 (Sun Microsystems Inc.)
Bundle-Category: service
Bundle-ManifestVersion: 2
Export-Package: com.makewave.bundle.example.hello_world2
Bundle-Vendor: Makewave
Bundle-SubversionURL: https://www.knopflerfish.org/svn/
Bundle-Activator:
com.makewave.bundle.example.hello_world2.Activator
Bundle-SymbolicName: org.knopflerfish.bundle.hello_world2
Bundle-DocURL: http://www.knopflerfish.org
Bundle-APIVendor: Makewave
Import-Package:
org.osgi.framework,com.makewave.bundle.example.hello_w
 orld2
Bundle-Version: 0.0.1
Bundle-UUID: org.knopflerfish:hello_world2:0.0.1
Bundle-Description: A Hello World Bundle 2
Bundle-ContactAddress: http://www.knopflerfish.org
Bundle-Name: Hello World 2
Bundle-Classpath: .
Build-Date: Mon January 28 2008, 18:38:38
Built-From:
C:\x\knopflerfish_osgi_2.0.3\knopflerfish.org\osgi\example
 s\hello_world2
```

## Concepts of OSGi - Manifest

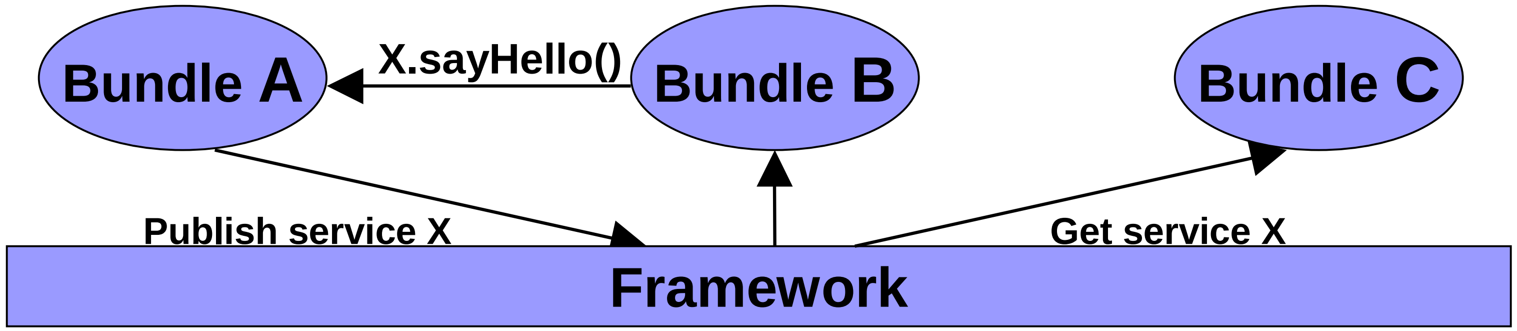
- The framework guarantees that only classes that are declared in the manifest as imported or exported can be used as such.
  - No sneaking
  - No unvoulantary leaking
- The manifest can be inspected prior to installing by an administrator
  - Bring in 3rd party lib => inspect it first!

## Concepts of OSGi - bundle management

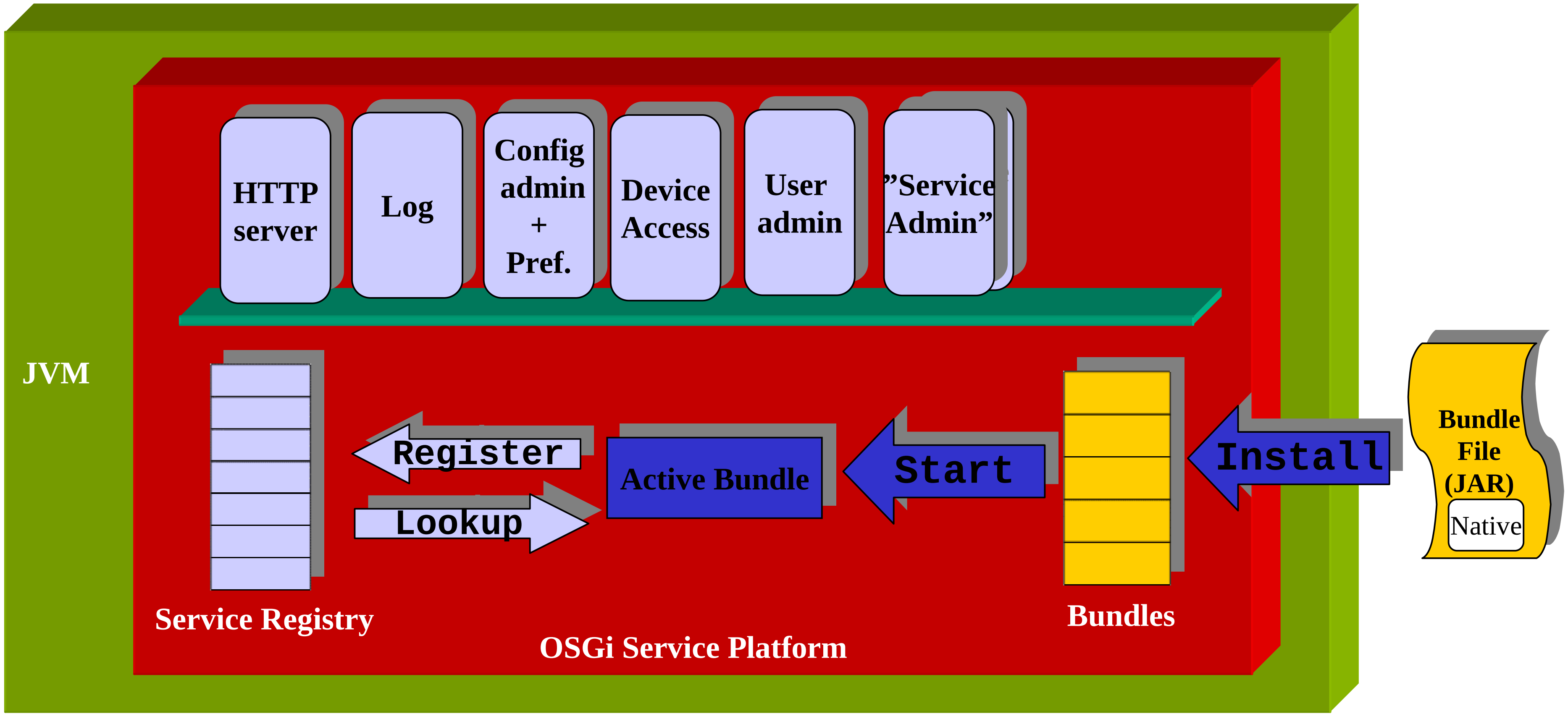
- Bundle primitives
  - Install <=> load the bundle into the framework
  - Start <=> “start” the bundle by calling a predefined method
  - Stop <=> “stop” the bundle by calling a predefined method
  - Uninstall <=> remove the bundle from the framework
  - Update <=> update the code of the bundle (without restarting JVM)

## Concepts of OSGi - Services

- Services
  - A means for inter-bundle communication
  - A means for providing functionality
  - Dynamic publish/find/bind service model
  - A service registered under one or more Java interfaces
  - OSGi defines a number of standard services



# OSGi Service Platform Overview



## OSGi Compendium Services (R4)

- Defines the compendium of OSGi defined services
- Will be updated from time to time as new services are finalized

  - Log Service
  - Http Service
  - Device Access
  - Configuration Admin
  - Preferences Service
  - Metatype
  - Wire Admin
  - User Admin
  - IO Connector
  - Initial Provisioning
  - UPnP Device
  - Declarative Services
  - Event Admin
  - Service Tracker
  - XML Parser
  - Position
  - Measurement and State
  - Execution Environments

## OSGi & Remote Management

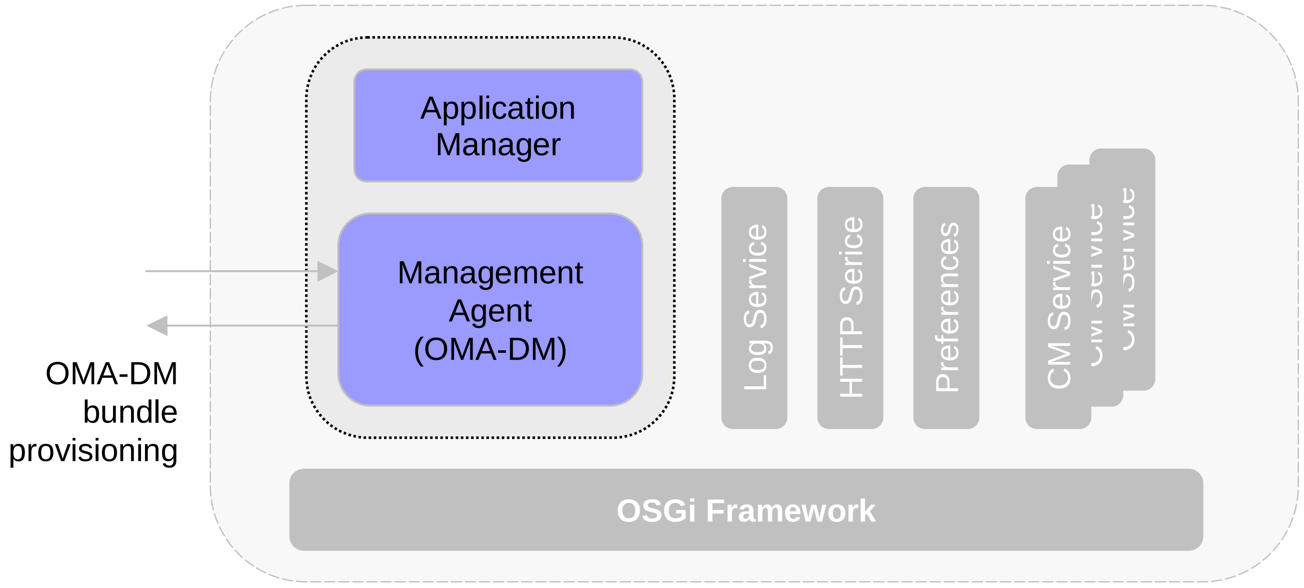
- The OSGi technology enables remote management
- But it does not dictate how to do it (so far)
  - I.e. there is no management protocol defined (except OSGi MEG)
- Instead the model assumes a Management Agent
- This is a bundle that is the link between the OSGi platform and its management system

## OSGi & Remote Management

**OSGi MEG defines an OMA-DM based provisioning through a management agent**

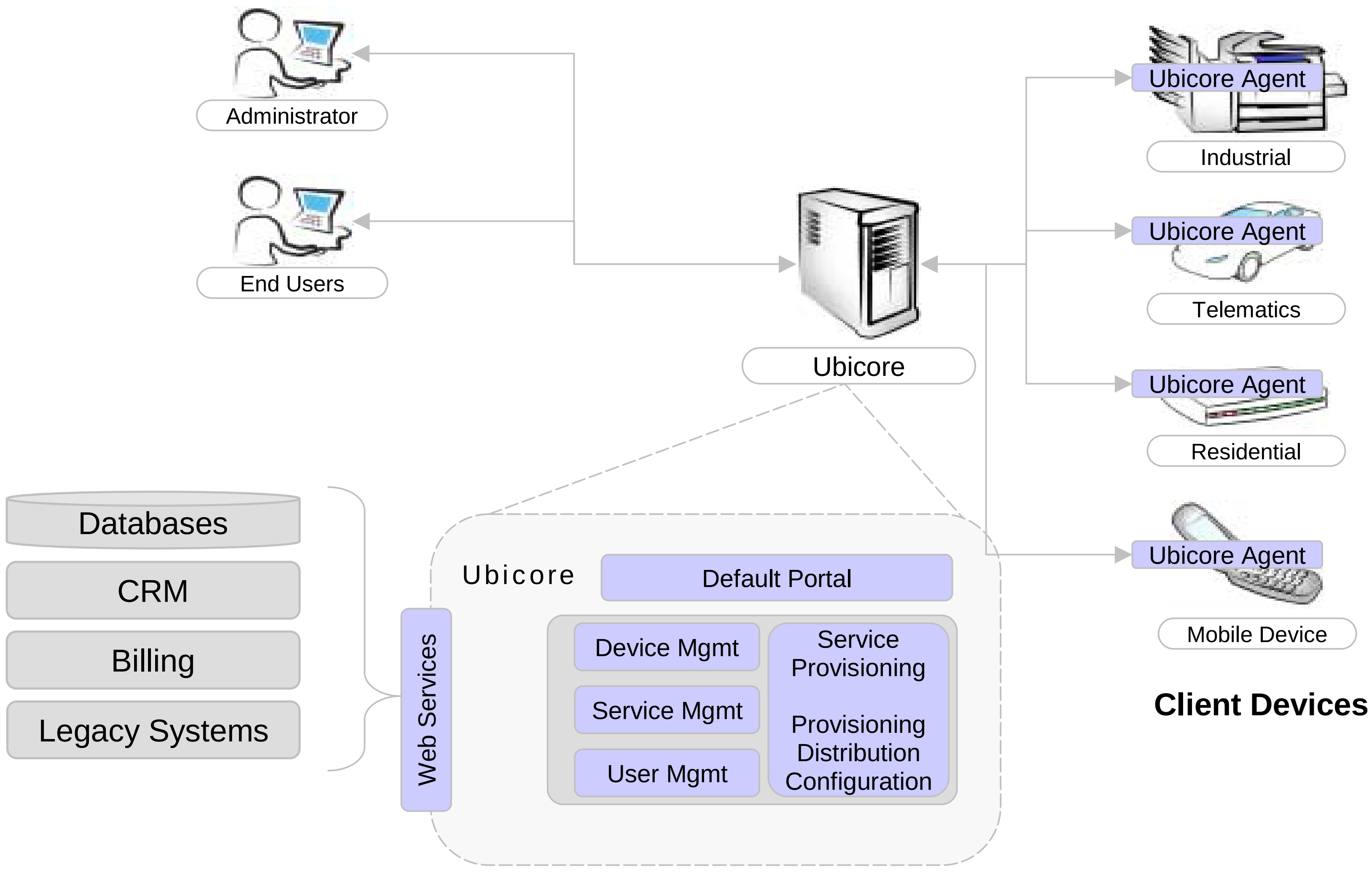
**But it is not enforced, and the model is completely flexible.**

**Arbitrary agents can be installed supporting any protocol!**

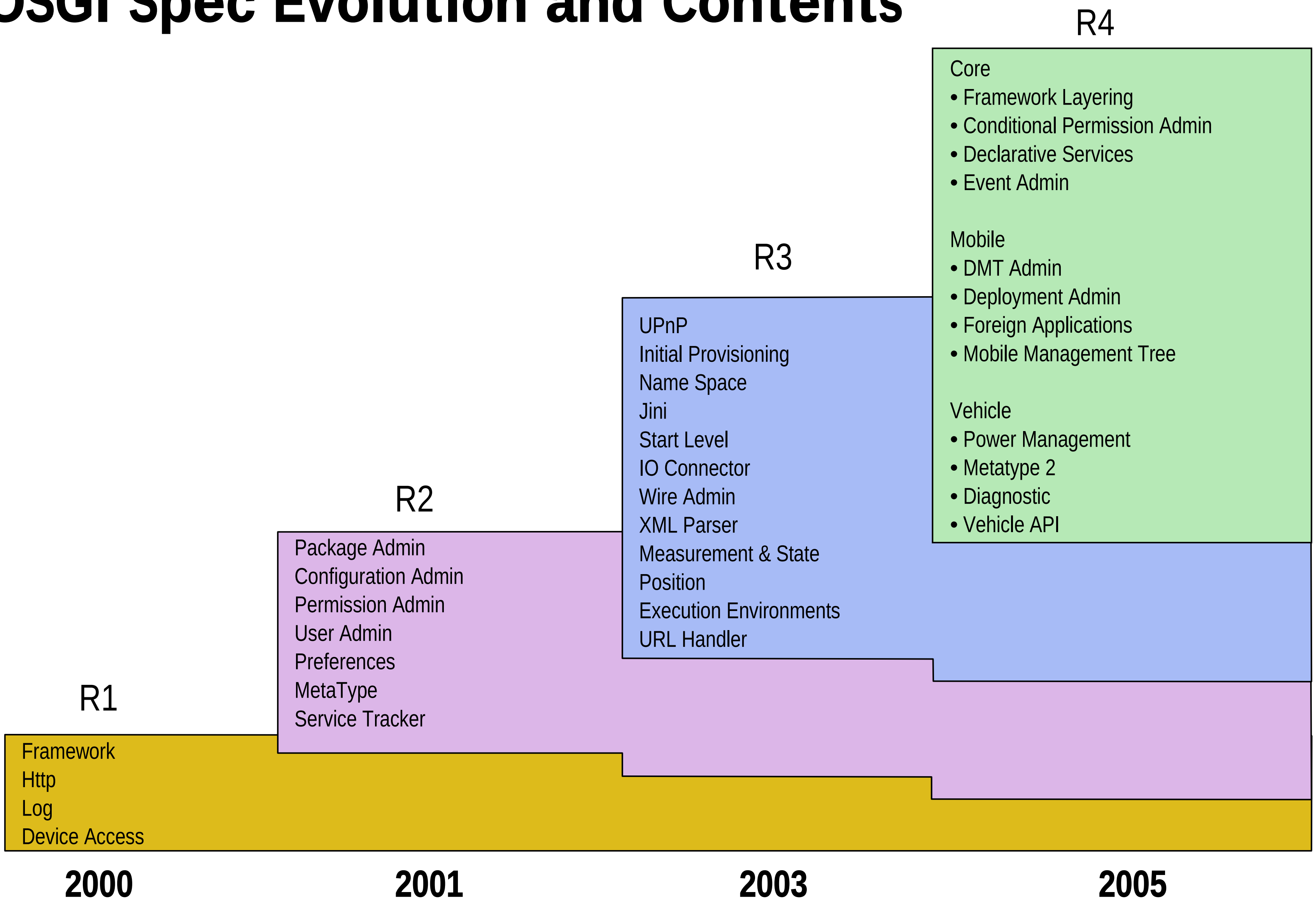


**But voices have been raised saying that a std protocol is a missing feature!**

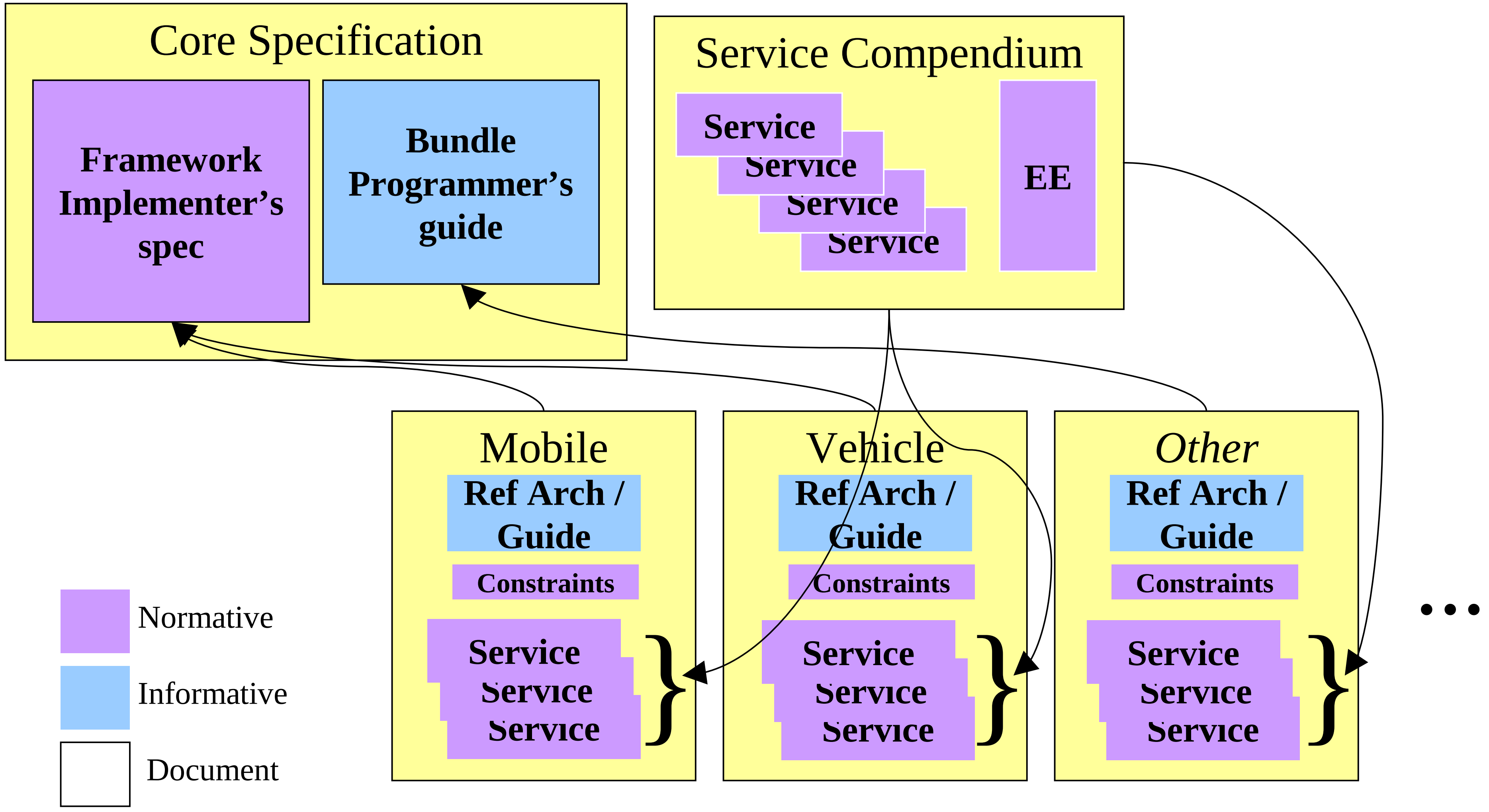
## Typical Remote Management Architecture



## OSGi Spec Evolution and Contents



## The OSGi Specifications



## The OSGi Expert Groups are creating the specs

- Core Platform Expert Group (CPEG)
  - Responsible for the Framework and core services as well as overall architecture
- Vehicle Expert Group (VEG)
  - Responsible for vehicle related requirements and designs
- Mobile Expert Group (MEG)
  - Responsible for mobile device related requirements and designs
- Enterprise Expert Group (EEG)
  - Responsible for enterprise/server related requirements and designs
- Residential Expert Group (REG)
  - Responsible for residential related requirements and designs

## OSGi – open source adoption

- Significant open source adoption
- Three open source projects / implementations of the OSGi specification (R4)
  - Equinox / Eclipse
  - Felis / Apache (Oscar)
  - Knopflerfish / Makewave
- Used by several other open source projects
  - Spring DM
- Important to understand
  - The OSGi Alliance maintaines the spec, and certifies for compliance
  - The open source projects are implementations of the spec.
  - I.e. the implementation is not the spec!!

## Programming with OSGi

- Pros
  - Easy to get going
  - Dynamic
  - Non-stop applications
  - Basic functionality for free (standard services)
  - A new level of modularization (bundles)
- Cons
  - Dynamic

## Programming with OSGi

- What if I don’t care about dynamics, service lookup etc.?
  - One application, one vendor, one installation

- By using OSGi in any Java based system the modules/services often just falls into the right place.

- See the OSGi design philosophy as your guide to designing modular java applications.

## Some examples – Hello World

```
package com.makewave.bundle.example.hello_world;

import org.osgi.framework.*;

public class HelloWorld implements BundleActivator {

  /**
   * Bundle is starting
   */
  public void start(final BundleContext bc) {
    System.out.println("\n*** Hello World!\n\tHere I am\n");
  }

  /**
   * Bundle is stopping
   */
  public void stop(BundleContext bc) {
    System.out.println("\n*** Hello World!\n\tI bid you farewell\n");
  }

}
```

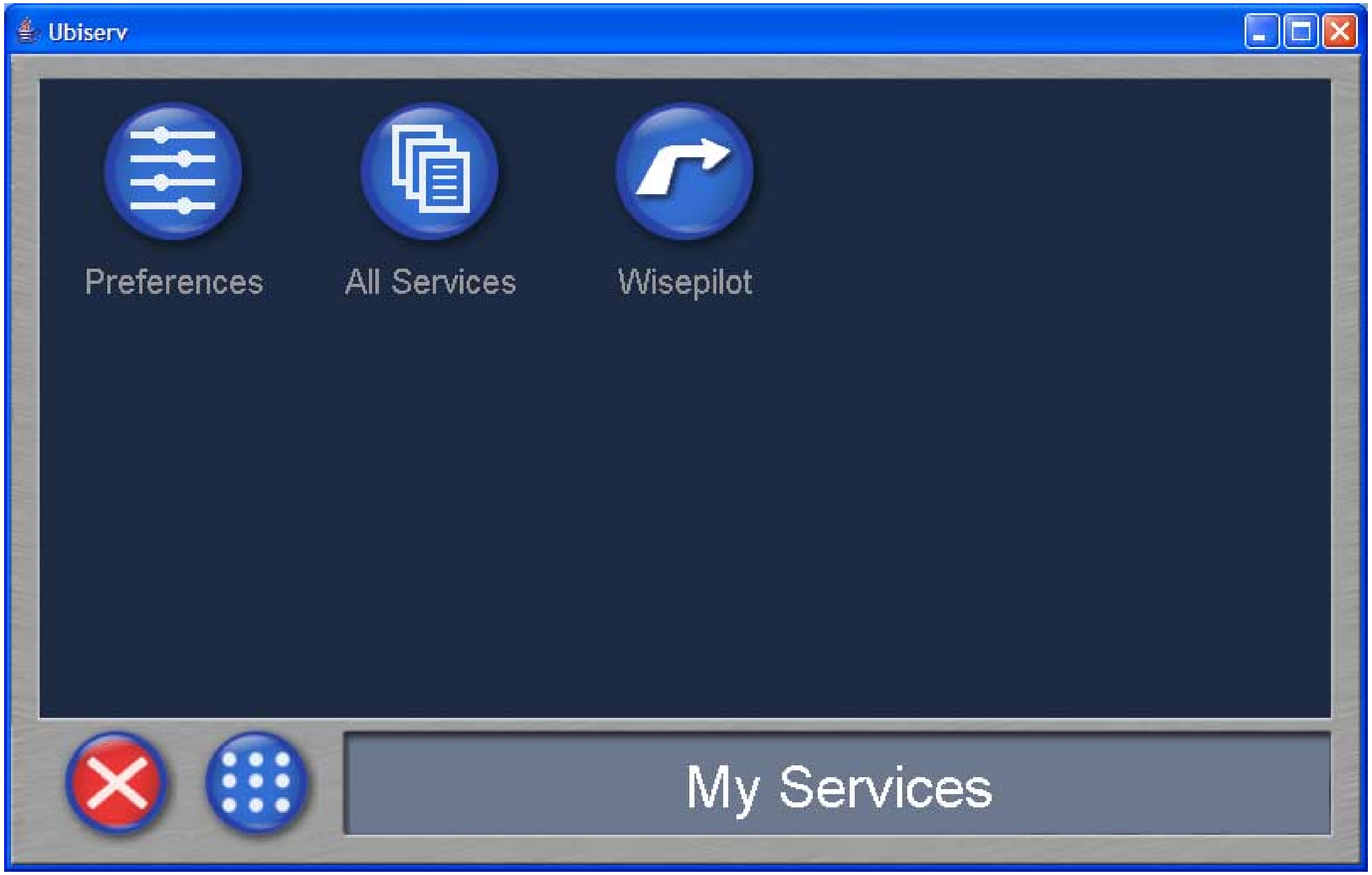
## What is the whiteboard model

- A service offers addListener and removeListener methods
- Instead define a listener interface that other bundles use to register in the framework to create a listener design
- Now the framework takes care of the house-keeping when bundles come and go and
- This can be thought of as inversion of control.

## Using the white board model

- The telematics R&D project called GST defined the concept of an Application Manager (Service Browser)
- The ”desktop” of the in-vehicle system
- The component / service through which the end-user can manage applications from within the CVIS host

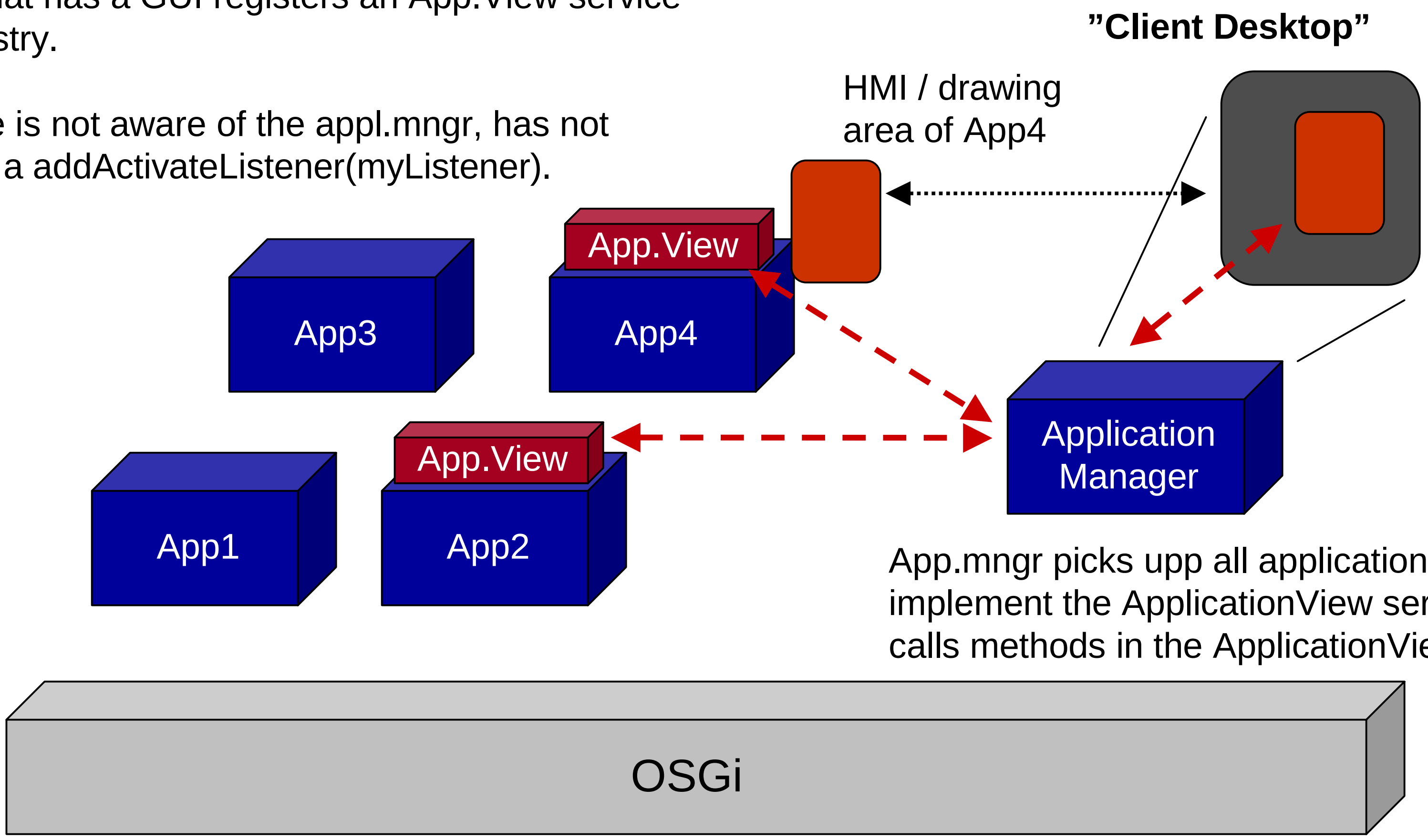
## A telematics application manager



## The white board model applied

Bundles that has a GUI registers an App.View service in the registry.

The bunde is not aware of the appl.mngr, has not registered a addActivateListener(myListener).



App.mngr picks upp all applications that implement the ApplicationView service and calls methods in the ApplicationView interface

## An example of a bundle using this

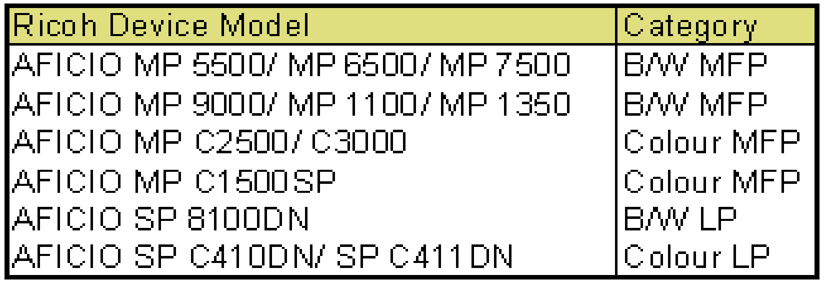
```
package org.cvisproject.service.hello_cvis.impl;

import java.util.Dictionary;
import java.util.Hashtable;

import org.osgi.framework.BundleActivator;
import org.osgi.framework.BundleContext;
import org.osgi.framework.ServiceRegistration;

import com.makewave.service.appmanager.ApplicationView;

public class Activator implements BundleActivator {

  private BundleContext context;
  private ServiceRegistration viewRegistration;

  private View view;

  public void start(BundleContext context) {
    this.context = context;
    view = new View(context.getBundle());
    Dictionary properties = new Hashtable();
    properties.put(ApplicationView.APPLICATION_ID, view.getViewId());
    viewRegistration = context.registerService(ApplicationView.class.getName(),
  view, properties);
  }

  public void stop(BundleContext context) {
    viewRegistration.unregister();
```

## Exempel på hur OSGi används

- Några ord om Makewave
- Introduktion till OSGi
- Lite om historien bakom OSGi
- Genomgång av OSGi teknologin
- Exempel på hur OSGi används
  - Embedded
  - Desktop
  - Server
- Ytterligare fördjupning

## OSGi in embedded - Ricoh

**Ricoh ships MFPs with Knopflerfish OSGi on all '05A and coming' Ricoh models**



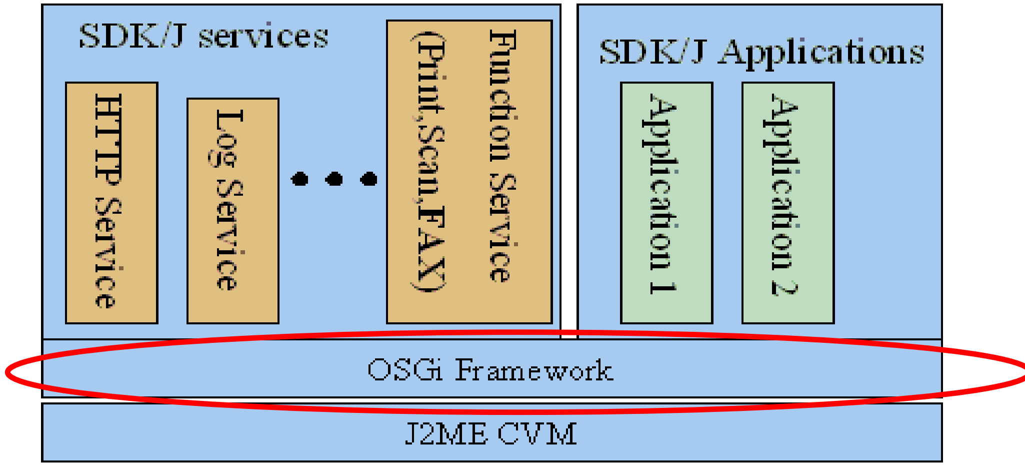
| Ricoh Device Model | Category |
|---|---|
| AFICIO MP 5500/ MP 6500/ MP 7500 | B/W MFP |
| AFICIO MP 9000/ MP 1100/ MP 1350 | B/W MFP |
| AFICIO MP C2500/ C3000 | Colour MFP |
| AFICIO MP C1500SP | Colour MFP |
| AFICIO SP 8100DN | B/W LP |
| AFICIO SP C410DN/ SP C411DN | Colour LP |

RICOH

Aficio™MP 5500/MP 6500

## OSGi in embedded - Ricoh

- OSGi enables a printer plug-in model.
- 3rd party apps can be loaded on the printer, extending the printer functionality.
- Core printer functions are available as OSGi services.
- Printer plug-ins are developed using an SDK.



## OSGi in telematics - BMW

- BMW 5 and 6 series ships with an OSGi framework
- Used as platform for telematics applications where multipe applications co-exist
- 3rd party applications can be enabled
- Possible to remotely manage.

## OSGi in desktop – eclipse

- Eclipse is fundamentally based on a plug-in model
- Up until release 2 of Eclipse they lacked dynamics
  - restart required!
- In release 3 of eclipse they moved to an OSGi based architecture
  - Runs an OSGi framework
  - All plug-ins are bundles
  - OSGi is still hidden inside eclipse, but is becoming more and more visible

## OSGi in the server side – Spring OSGi

- Spring has added support for OSGi such that a Spring application can be deployed in an OSGi environment.
- Benefits of using Spring OSGi for enterprise applications
  - Better separation of application logic into modules
  - The ability to deploy multiple versions of a module concurrently
  - The ability to dynamically discover and use services provided by other modules in the system
  - The ability to dynamically deploy, update and undeploy modules in a running system
  - Use of the Spring Framework to instantiate, configure, assemble, and decorate components within and across modules.
  - A simple and familiar programming model for enterprise developers to exploit the features of the OSGi platform.

## OSGi in the desktop / server side

- John Kellerman IBM product manager:
  - ”....IBM today consumes and redistributes 15 projects from Eclispe in over 350 products...”
- The whole Lotus suite is now running OSGi for instance....

## Server side OSGi is just beginning….

- There is really exciting work going on inside the Enterprise Expert Group
  - OSGi R5 will include new features for better supporting enterprise
- Companies involved:
  - Siemens (co-chair)
  - Iona (co-chair)
  - BEA
  - Oracle
  - RedHat (JBoss)
  - IBM
  - SpringSource
  - Makewave

## OSGi EEG work items includes

- Scaling including multi-container and multi-process environments
- Language bindings for enterprise services including, but not limited to the Java language
- Distributed and/or federated service model:
  - within multiple Service Platforms
  - with external, heterogeneous systems
- Requirements for extensions to the OSGi publish/find/bind service model
- bundle dependencies profiling and matching
- enterprise-class life-cycle and configuration management
  - from initial provisioning
  - through software and asset management, patching, etc.
  - focused on desktops, laptops, and servers
  - including reliability, availability, serviceability concerns

## Ytterligare fördjupning

- Några ord om Makewave
- Introduktion till OSGi
- Lite om historien bakom OSGi
- Genomgång av OSGi teknologin
- Exempel på hur OSGi används
- Ytterligare fördjupning

## Resources

- OSGi Alliance www.osgi.org
  - Peter Kriens blog: www.osgi.org/blog
  - Specification: www2.osgi.org/Specifications/HomePage
- Open source implementations
  - Knopflerfish, www.knopflerfish.org
  - Equinox, www.eclipse.org/equinox
  - Apache Felix, felix.apache.org
- User's Forums in
  - Japan, Korea, Spain, France
    - OSGi User Forum Japan has ~80 member companies!
  - and there is one in Sweden too!

## OSGi User’s Forum Sweden

- Det finns en svensk OSGi-användarförening
  - En ideel förening
- § 2 Föreningens syfte
  - Föreningen har till ändamål att främja medlemmarnas intressen genom att aktivt delta i diskussion och utveckling av OSGi teknologi, samt att främja dess användning.
- Öppen för alla, juridiska personer såväl som individer
- Webplats:
  - http://sweden.osgiusers.org/
  - maillista

## Articles on OSGi

- SD times named OSGi:
  - ”a quite contender for the title of most important technology of the decade”.
  - http://www.sdtimes.com/article/story-20070601-27.html

## Thank You!

**Christer Larsson**
**CEO**
**www.makewave.com**
**cl@makewave.com**