



# **OSGi Alliance Community Event**

## **OSGi on the Ricoh Office Device Platform**

**Abdalla Bushnaq**



- **Short Introduction of Ricoh**
  - Ricoh's Product Line-up, Global Network, Consolidated Sales, MFP Market Share.
- **The Ricoh Office Device Platform**
  - Platform Evolution, Embedded Software Architecture, SDK/J Architecture, Types of Applications.
- **OSGi**
  - The OSGi Platform, What is provided? Why OSGi? Difficulties, Pros and Cons of using OSGi, Would Ricoh decide to use OSGi again? A Problem Example from Real Life.



## Abdalla Bushnaq

Is an Assistant General Manager at Ricoh.

Heads a team of software engineers at Ricoh's European Development and Support Centre in Germany.

The team develops tailored solutions for European customers that help them to manage larger numbers of Ricoh multifunctional office devices. This may include solutions to control costs, maintenance management, as well as embedded software to optimize the MFPs functionality to the customers' need.

His team also provides Europe-wide support to 3rd party developers that want to develop solutions for the Ricoh embedded Java platform.





# Ricoh's Product Line-up

## *Image Processing Device*



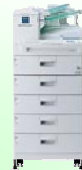
MFP(Aficio1105)



Printer (Aficio AP 3800C)



Digital Duplicator



Fax



Scanner



Digital Camera

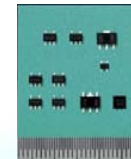


DVD+RW/+R



CD-R/RW

## *Others*



Semiconductors



Thermal products



Watches



Software

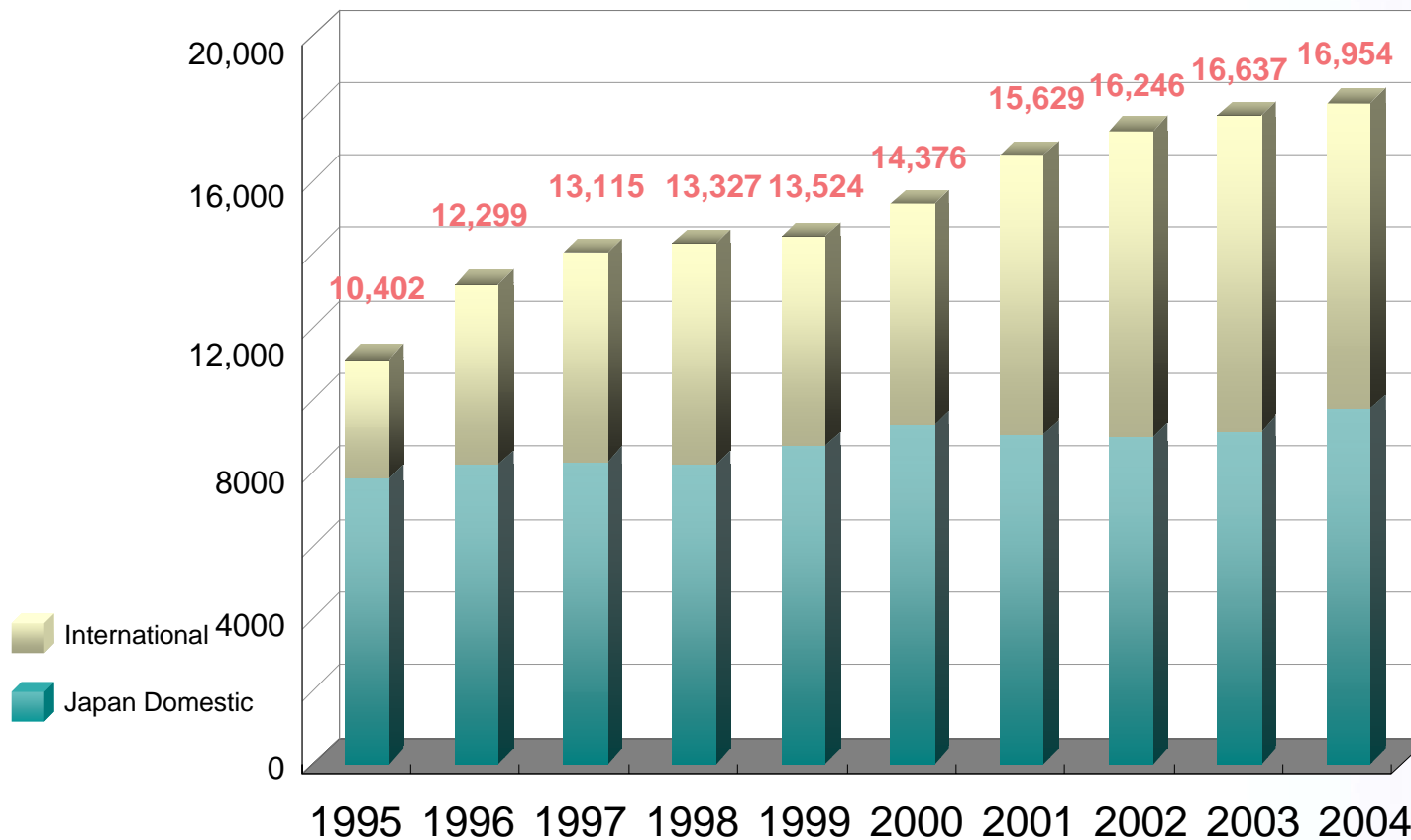




# Consolidated Sales

## 11th Consecutive Year of Increased Sales

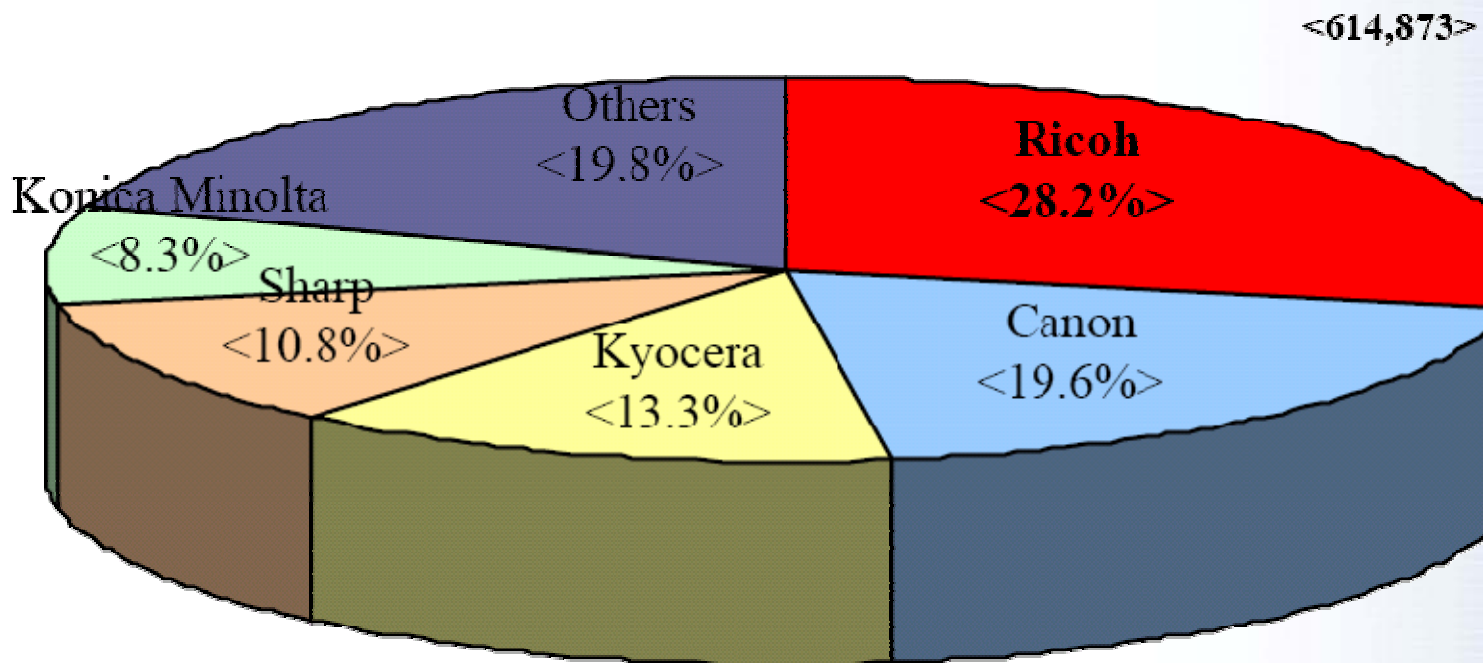
(US\$ million)





## Ricoh MFP Market Share

### Black & White (&CC) Office Copier Placement (1<sup>st</sup> 6 months 2006 by Manufacturer)





# Platform Evolution

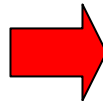
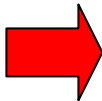
1965

1994

2000

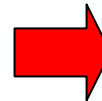
2004 

Printer  
Scanner  
Copier  
Fax



**GW Architecture**

Copier/Printer/Scanner/Fax



**Embedded SA.**

Copier/Printer/Scanner/Fax





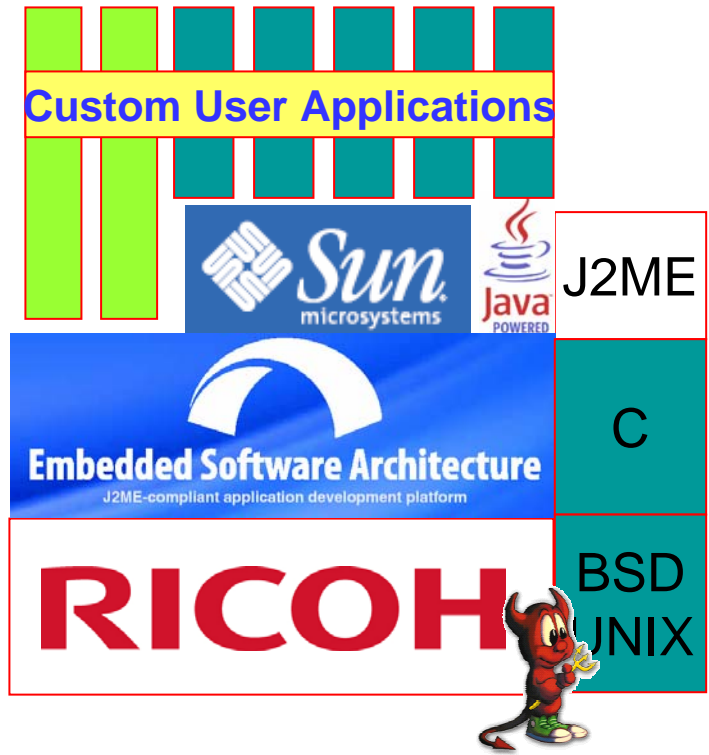
# Embedded Software Architecture

2004 

Open Platform

Embedded S.A.

Copier/Printer/Scanner/Fax



## Functions controlled by SDK.

- LCD Panel
- Scanner
- Printer
- FAX
- etc.

## Network functions:

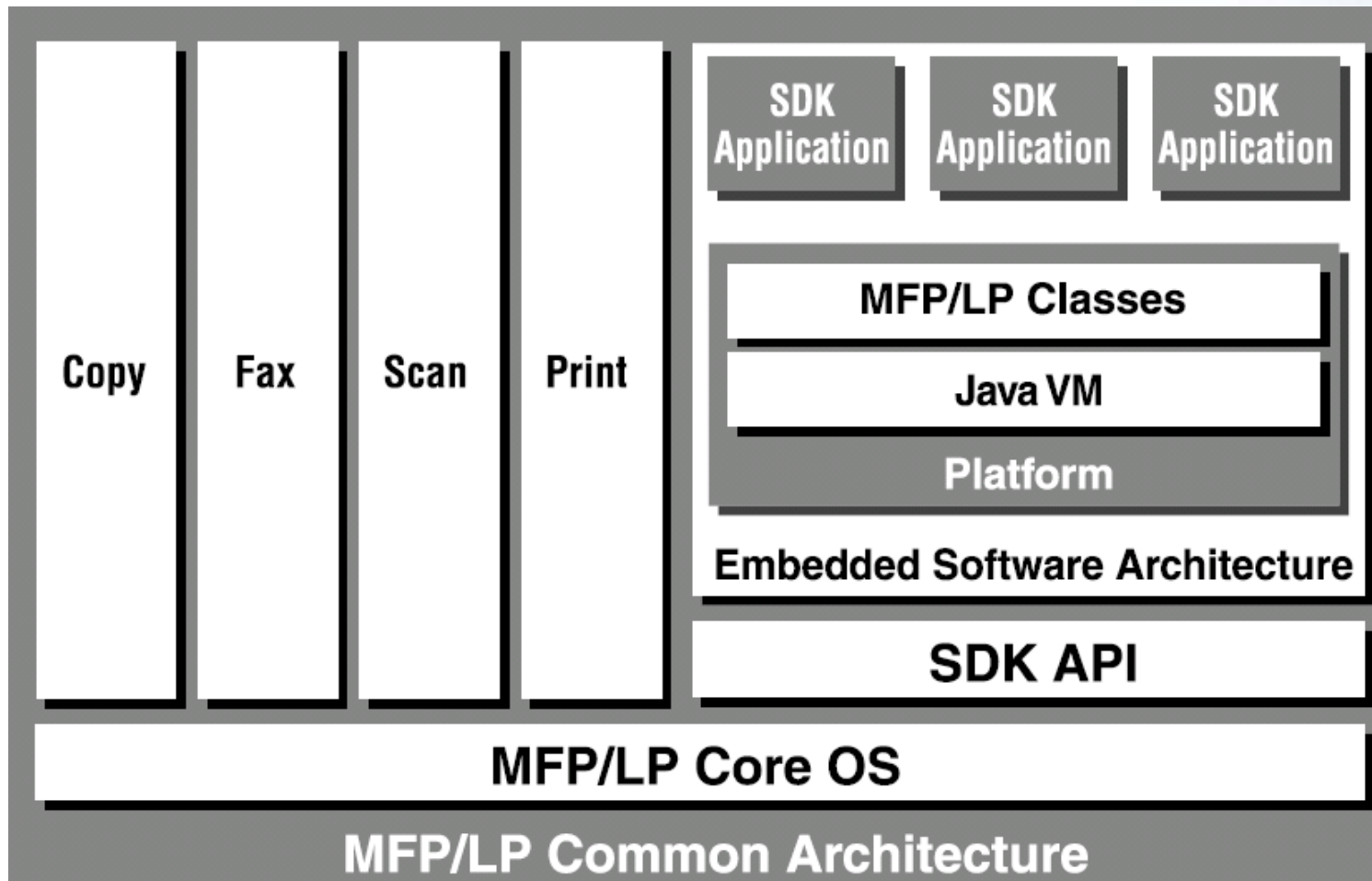
- HTTP, ftp, SMTP, SMB, etc.
- Web Services (SOAP WSDL)
- etc.

## Supported devices:

- MFP (color/B&W)
- LP (color/B&W)
- Wide Format



# SDK/J Architecture





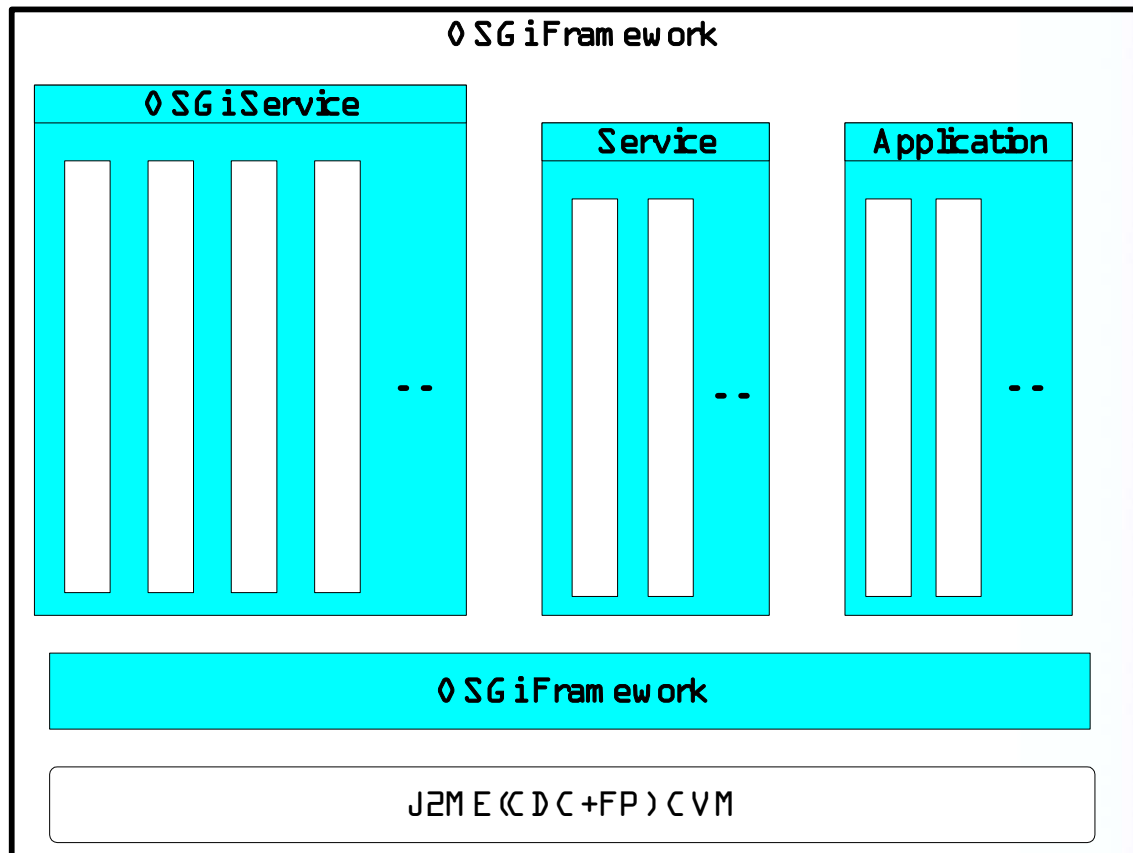
# Types of Applications

- **Panel Operation-Oriented "Thin Client" (Xlet)**
  - This type of Java technology application uses the MFP as focal point for paper-based workflows.
  - Examples include scan-to-server and print on demand.
- **Java Servlet Based Web Application**
  - Web server applications such as device monitoring.
- **Background Application**
  - Daemon-type of applications such as a web service print server.

**And many more...the possibilities are only limited by your imagination!**



# The OSGi Platform





# What is provided?

- **OSGi version 3**
  - This was the latest version available at development start.
  - Knopflerfish version 1.3.3 .
- **Provided Bundles**
  - ACR Manager Service
  - ApplicationManager Service
  - Server Service
  - Util Service
  - Net Service
  - Au Manager Service
  - MultiXletManager Service
  - PanelManager Service
  - FunctionManager
  - SystemEventManager Service
  - Taskbar Service
  - SasManager Service
  - ComManager Service
  - RemoteInstManager Service
  - Activation Service



## Why OSGi?

- **Enable dynamic install/uninstall/update the embedded apps**
  - Office equipments requires automatic update to reduce maintenance
  - Need deploying custom applications to large number of devices
- **Easy to add system level bundles**
  - Add/Remove optional APIs easily
  - Publish/Subscribe required APIs
- **Light Servlet support**
  - Tomcat is too heavy.
  - JETTY requires more memory and xlet support was more difficult to realize.
  - Decided to use Knopflerfish.



# Difficulties

## Features that Ricoh wanted to use but was not able to.

- It is not possible to use the same bundle with a different version.
  - Knopflerfish limitation.
- OSGi supports memory or HD cache method, but it can only be turned on or off for the whole platform. It is not possible to choose for separate bundles.
- **Using a bundle as a service was not possible to get running.**
- **Difficult to use the OSGi advantages**
  - You have to learn a lot.
  - Example: security implementation in http service.  
Org.osgi.service.http.httpcontext handleSecurity method.



# Pros and Cons of using OSGi

- **Pros**

- Independencies of bundles.
- Updating without reboot.
- Support for embedded development.
- Creating an independent class loader for every bundle.

- **Cons**

- It does not match the lifecycle of a bundle on a MFP.
  - Pause, resume (Xlet)
  - Various Sleep-modes
- It is not possible to get all bundle status.
  - Pause, sleep mode, memory consumption...



# A Problem Example From Real Life

- **Bug in Knopflerfish**
  - Remote updating a bundle will unzip the included jar files to create a cache.
- **Real Life**
  - Whole embedded Java platform is running from a 64MB SD card.
- **Effect**
  - SD card was full after about 3 updates.
- **How?**
  - Cluster size of the SD card was 16KB.
  - This caused the unzipping of a relative small jar file with a lot of small files to fill much more space on the card than expected.
- **Solution**
  - Added functionality to the RemoteInstManager Service to clear the bundle cache after rebooting the platform.



# Would Ricoh decide to use OSGi again?

- **Yes!**
  - Actually doing so in the next embedded platform generation.



**Thank you!**