



# **The embedded Rich Client Platform (eRCP): Effective Collaboration on Common Tools and Platforms**

Mark Rogalski, IBM Lotus



## Topics

- In the Beginning - Eclipse
- What is eRCP?
- Why is eRCP important?
- Industry uptake
- Other Eclipse embedded projects
- Thoughts on open, collaborative development



## In the beginning - Eclipse

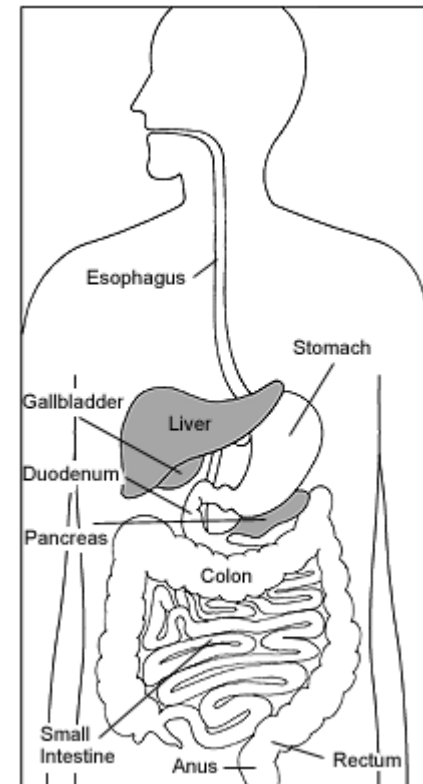
- Consortium working on open source projects
- Started when IBM contributed the Eclipse IDE
- Now a collaboration of over 100 companies working on hundreds of projects
- IDE has morphed to RCP
- Java centric framework



## What is eRCP?

- **Google “eRCP” – top hit is:**
- **ERCP (Endoscopic Retrograde Cholangiopancreatography)**

Endoscopic retrograde cholangiopancreatography (en-doh-SKAH-pik REH-troh-grayd koh-LAN-jee-oh-PANG-kree-uh-TAH-gruh-fee) (ERCP) enables the physician to diagnose problems in the liver, gallbladder, bile ducts, and pancreas...





## What is eRCP?

- eRCP is an embedded version of the Rich Client Platform
- Project was launched by IBM, Motorola, and Nokia
- Reduces RCP size/function to fit on devices
- Utilizes same OSGi Service Platform (a community standard)
- Pushes requirements/patches back to core components
  - Enable running core components on JME CDC/Foundation Profile
  - Keep dependencies in check
- Adds components to enable application binary compatibility across a range of devices

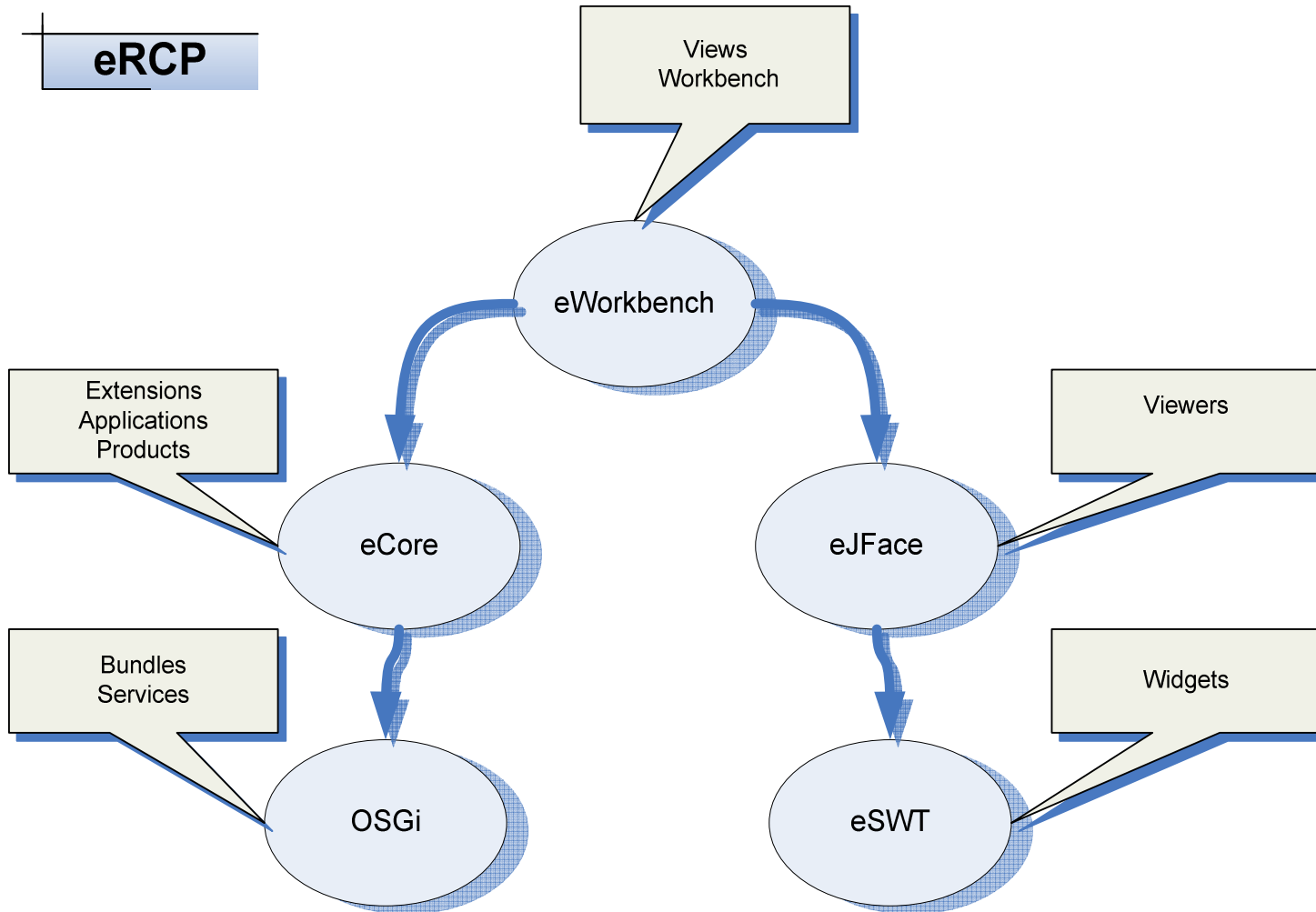


## eRCP Architecture

- OSGi underpinnings
- Applications and services run in the same JVM
  - Consumes fewer resources than separate JVM processes
  - Allows sharing of common services
  - Enables variety of life cycle choices
- eRCP applications can be launched from a workbench or be integrated into the native desktop
- Deployable to a range of devices
  - Desktops, handhelds, cell phones, etc.
  - eRCP apps can also run on RCP

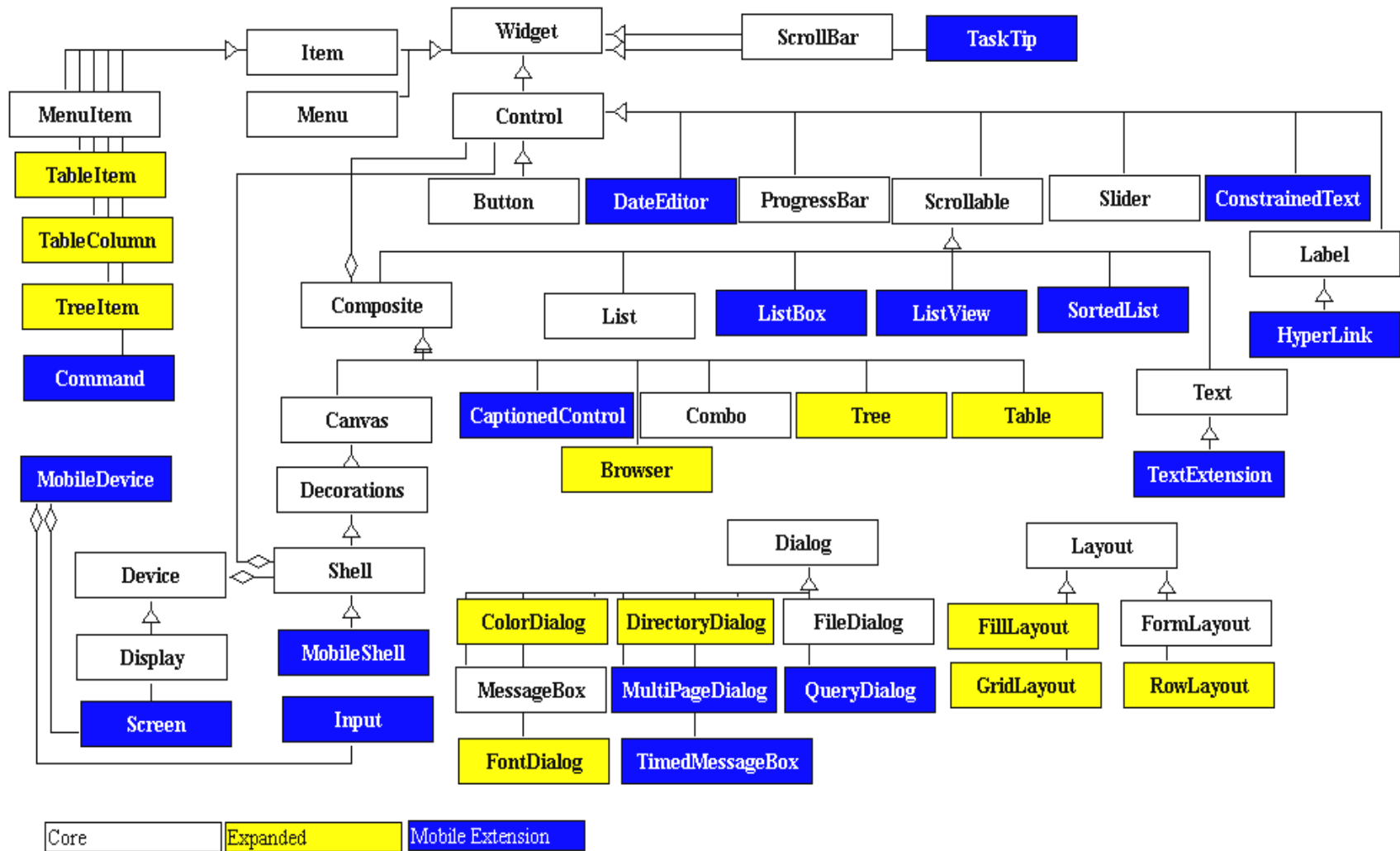


# eRCP Components





# Lots of eSWT Widgets





## Native Look and Feel

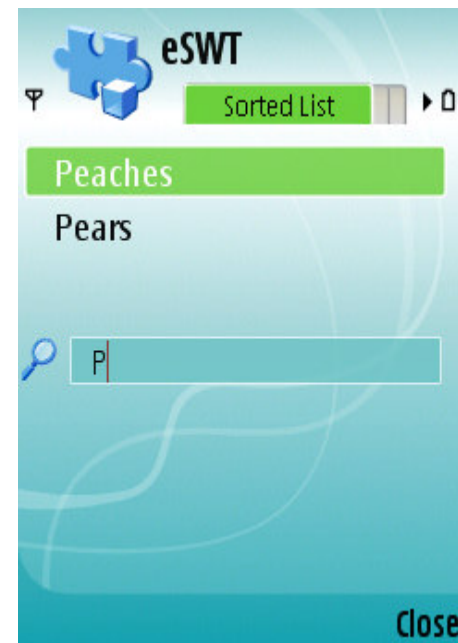
A screenshot of a mobile application interface. At the top right, there is a search icon and the text "Abc". The interface is divided into two main sections: "Manufacturer" and "Inspector". Under "Manufacturer", there are three input fields: "Company" with the value "Hapaq Lloyd", "Contact" with "Buzz Lightyear", and "Order No" with "1001-2006". Under "Inspector", there are three input fields: "Name" with "Douseau", "Reference No" with "3287", and "Status" with "Completed". At the bottom left, there is a button labeled "More Layouts", and at the bottom right, there is a button labeled "Exit".

- Java apps can blend in with pre-installed applications
- Use the same navigation controls and soft-keys as native apps
- Exploit user's familiarity with native GUI operations



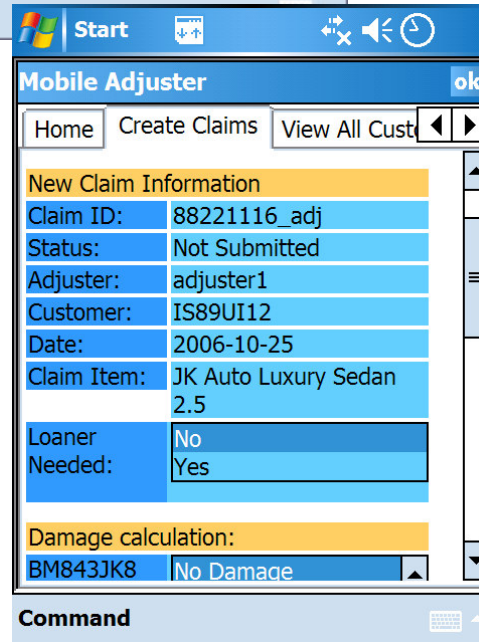
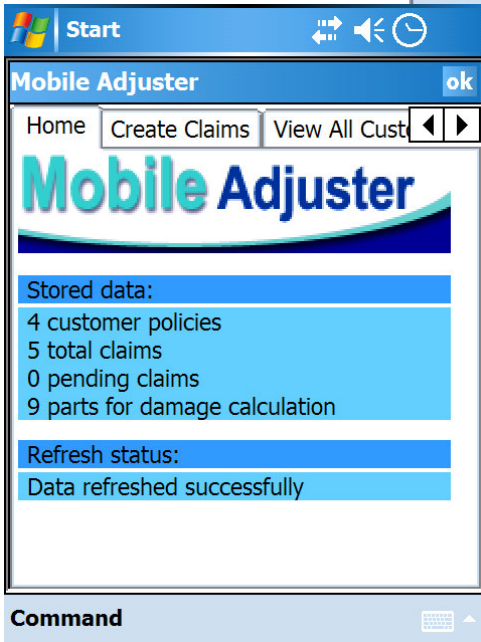
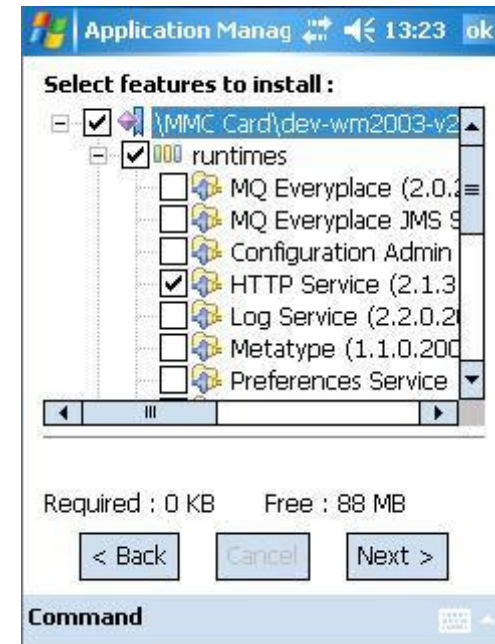
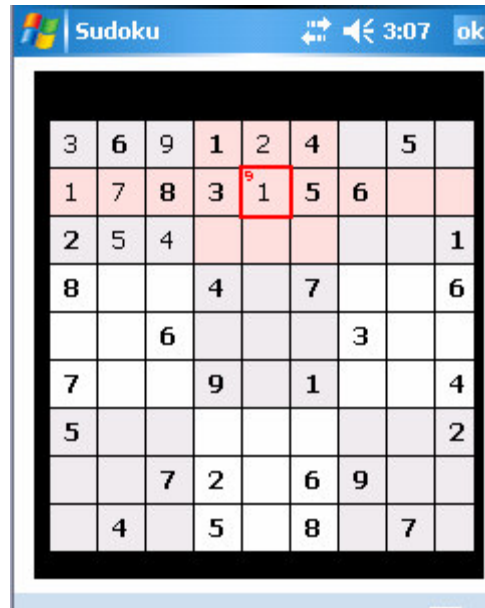
## Mobile oriented widgets

- DateEditor
- CaptionedControl
- MultiPageDialog
- SortedList
  
- HyperLink
- TextExtension
- MobileDevice
  - Discover device capabilities and characteristics





# Screenshots





## Tools

- Eclipse IDE
  - Developers can apply desktop development skill to devices
  - Reuse code already written
  - Take advantage of plentiful books and articles
- eRCP Test Harness
  - Sends tests to device and reports back on the results
- Instantiations WindowBuilder
  - Visual UI building
  - Code generation with ability to edit
  - Beta release this week



## Platforms

eRCP is mostly Java code, but eSWT needs to be implemented for specific platforms

- Windows™ Desktop
- Windows Mobile 2003 / 5 / 6
- WinCE 5.0
- Nokia S60 platform (Coming up!)
- Nokia Series 80 platform
- Others being considered (GTK, Qte, ...)



## Why is eRCP important?

- There are a **HUGE** number of devices out there that are becoming quite capable Java™ application platforms
- eRCP is the next step up from MIDP in Java platforms
- Brings OSGi service oriented capabilities to devices
  - Applications can reuse other applications, plugins and services
  - Platform for mobile mash-ups
- Extensive rich UI capabilities that are lacking in MIDP
- Higher level of device abstraction and integration
- Over-the-Air update of applications



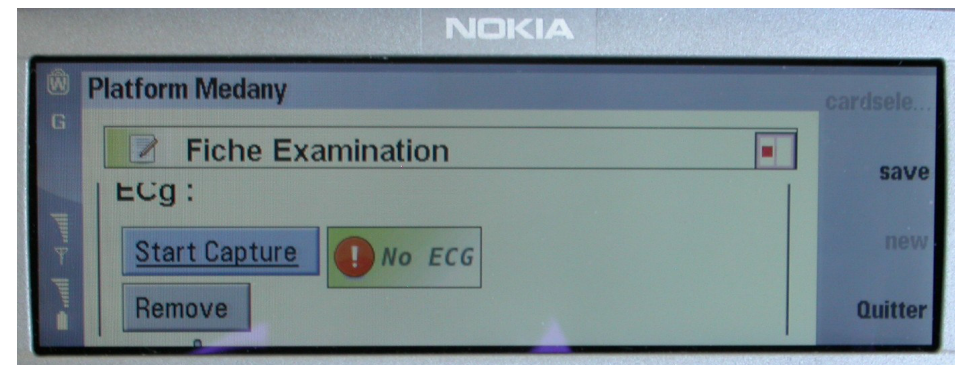
## Industry Uptake - Medany

- Medical field usage of eRCP, EMF, and GMT
- Developed by Anyware Technologies
- Acquires data on mobile device:
  - texts, numerics, multimedia (pictures, sounds...)
  - binary data (glucometer...).
- Easy and secure (access restrictions):
  - usage : for nurses, doctors...
  - update : interfacing with other devices must be easy.
  - synchronization : automatic, two ways



# Medany Solution

- Tools and runtime for eRCP based applications
- Tools are based on
  - Eclipse Modeling Framework (EMF)
  - Open ArchitectureWare
- Tools generate
  - eRCP and RCP applications
  - Synchronization servlet
  - Web/struts application
- Runtime deals with
  - Data acquisition
  - Synchronization
  - Application updates



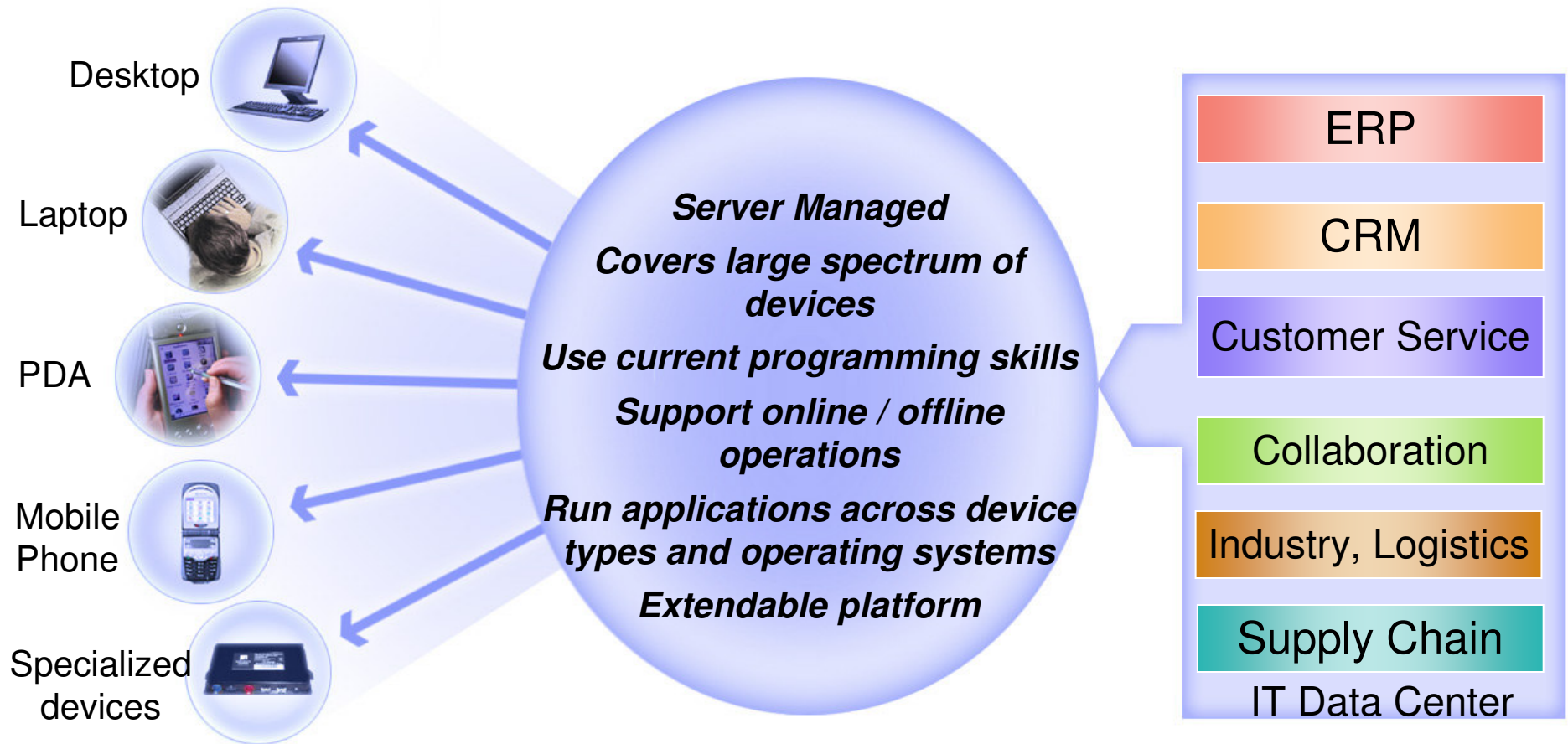


## Industry Uptake - IBM

- IBM® Lotus® Expeditor®
  - IBM's managed client solution
  - Desktop version is based on RCP
  - Device version is based on eRCP
  - Delivers significant middleware services not provided by Eclipse
  - Remote management for large volumes of devices



# IBM's Managed Client Solution





## Industry Uptake - Nokia

- eRCP is a add-on software for Nokia E90s
  - Available in conjunction with IBM Lotus Expeditor
- Runs on top of mobile optimized OSGi engine
- Mobile JSRs planned to work with eRCP
  - Wireless Messaging API (JSR 120)
  - Location API (JSR 179)
  - File and PIM API (JSR 75)
  - Scalable 2D Vector Graphics API (JSR 226)
  - Mobile Media API (JSR 135)





## Industry Uptake - Sprint

- eRCP part of Sprint's Mobile Java Service Platform
  - For more capable PDA type phones
  - CDC/Foundation 1.1 capable JVM
  - Continues to run Midlets
  - Runs eRCP applications
- Future possibilities
  - OSGi aware Midlets
  - Extending platform to feature phones



## Future eRCP Plans

- Release 1.1 - August 2007
  - Currency with Eclipse 3.3
  - Windows Mobile 6 Professional support
  
- Release 1.0.3 - October 2007
  - Maintenance Release
  
- Release 1.2 - July 2008
  - Currency with Eclipse 3.4
  - Window Mobile 6 Standard support
  - More support for multiple screens



## DSDP – Device Software Development Platform

- Projects specialized for devices
  - Device Debugging
  - Target Management
  - Native Application Builder
  - Tools for Mobile Linux
  - Mobile Tools for Java
  
- Some companies involved
  - Fujitsu
  - IBM
  - Motorola
  - Nokia
  - WindRiver
  - MontaVista
  - Mentor Graphics
  - PalmSource
  - Symbian
  - Texas Instruments
  - QNX
  - Freescale



## Device Debugging (DD)

[www.eclipse.org/dsdp/dd](http://www.eclipse.org/dsdp/dd)

- Mission: ***Build enhanced debug models, API's, and views that augment the Eclipse Debug Platform in order to address the added complexities of device software debugging.***
- The Eclipse 3.2 Debug Model
  - APIs to enable customized embedded debugger implementations
  - Model driven view updates with asynchronous interactions between UI and debug model
  - Flexible view wiring (e.g. input to variables view)
  - The ability to debug multiple sessions simultaneously
  - Enhance the platform memory view with embedded-specific renderings
- The Debugger Services Framework (DSF)
  - Concurrency – ensures thread-safety and fast responsiveness for stepping, etc.
  - Services – provides plugability of individual components like registers, memory, breakpoints, etc.
  - Data Model – for retrieving data and populating views
- IP-XACT editor and debugger views (from SPIRIT consortium)



# Target Management (TM)

[www.eclipse.org/dsdp/tm](http://www.eclipse.org/dsdp/tm)

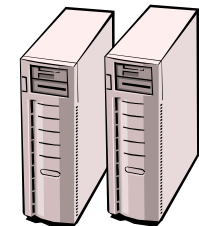
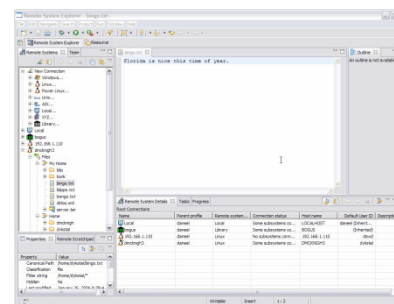
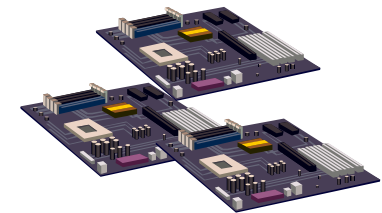
- Mission: **Create data models and frameworks to configure and manage embedded systems, their connections, and their services.**

## Remote Computer Systems...

- Targets (Locally connected, shared, fielded)
- Hosts (Grids, farms, nodes)

## ... and developing software on them

- Build, connect, get status
- Download, run, debug, test
- Upload





## TM Features

- Remote System Explorer Framework
- Support for Dstore, FTP, ssh connection types
- Integrate Jakarta Commons Net library for FTP access
- CDT remote launch capabilities
- Ansi terminal view & serial connection
- Support for Windows, Linux, Solaris, Mac



## Native Application Builder (NAB)

[www.eclipse.org/dsdp/nab](http://www.eclipse.org/dsdp/nab)

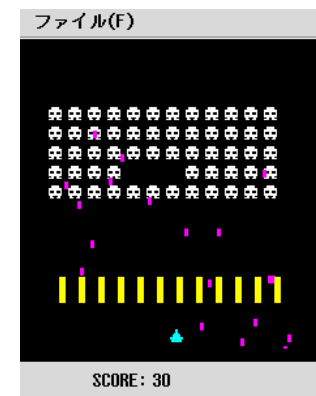
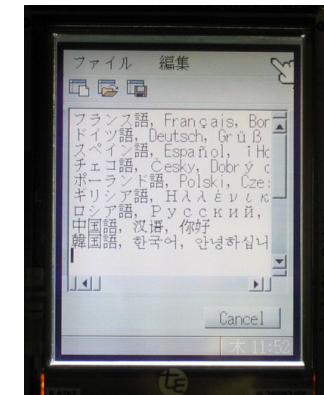
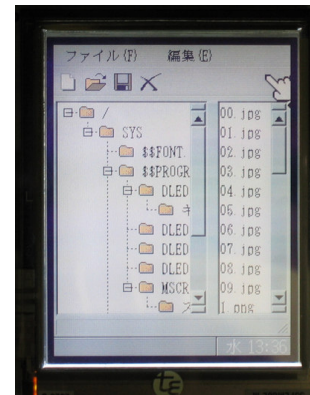
- Mission: **Create a C++ GUI builder for embedded operating systems, and widget runtime similar to eSWT for Java.**
- Features
  - Visual editor plugin to Eclipse for building GUI's
  - C++ application framework
  - MWT (Multiplatform Widget Toolkit) – separate runtime for each supported platform

Supported Platforms of MWT	
OS	Windows, Linux, MacOSX, FreeBSD, Solaris, T-Engine, ITRON, BTRON
CPU	IA32, ARM, SH3/4, FRV, MIPS, PPC, SPARC
Graphics	X11 Server, DirectFB, Frame Buffer, Win32, WinCE, T-Engine(T-Shell), MacOS



## NAB Features

- Large set of widgets
- Mostly wrapping native widgets
- High degree of customizability
- Good for games as well as rich GUI
- Applications easily ported across OS and GUI system by re-compiling





## Tools for Mobile Linux (TmL)

[www.eclipse.org/proposals/tml/](http://www.eclipse.org/proposals/tml/)

- Mission: ***Create frameworks and tools for the entire life-cycle of C/C++ application development targeted at mobile Linux platforms.***
- Design
  - Focus on modeling
- Development
  - Cross-compilation of OS, middleware, and applications
  - Simulate mobile device services
  - Emulate mobile devices and end-to-end environment
- Debug
  - Cross debugging
  - Support device emulators



## TmL continued

- Deployment
  - Application testing
  - Code Signing
  
- Mobile Linux Emulator Framework
  - Generic framework to support different device emulator architectures
  - VMware, User-mode Linux (UML) emulators, QEmu emulators, etc.



## Mobile Tools for Java (MTJ)

[www.eclipse.org/dsdp/mtj](http://www.eclipse.org/dsdp/mtj)

- Mission: ***Support mobile device Java application development, including application creation wizards, UI design tools, device emulator framework, deployment framework, generic build processes, and debugging.***
- Provide platform that device vendors can extend to support their devices
  - Runtime management framework to manage emulators + real devices
  - Build framework for customized packaging and signing
  - Deployment framework
- Provide tools to develop mobile Java applications
- Provide localization, optimization, and security
- In need of committers interested in completing this project



## Open, collaborative development

- Not everything developers create adds value
- Need to put limited resources where it really matters
- Just having an open OS is not enough

Open source development enables...

- Sharing cost of infrastructure
- Community providing on-going requirements, feedback, solutions
- Interoperability – the whole is greater than the sum of the parts



## Further Information

- Visit our websites
  - <http://www.eclipse.org/ercp>
  - <http://www.eclipse.org/dsdp>
  
- Join newsgroup
  - <news://news.eclipse.org/eclipse.dsdp.ercp>
  
- Contribute to eRCP!
  - Bring eRCP to your favorite device!



## Questions & Answers





## Legal Notices

- IBM and Lotus are trademarks or registered trademarks of International Business Corp. in the United States and other countries
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both
- Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Other company, product, or service names may be trademarks or service marks of others