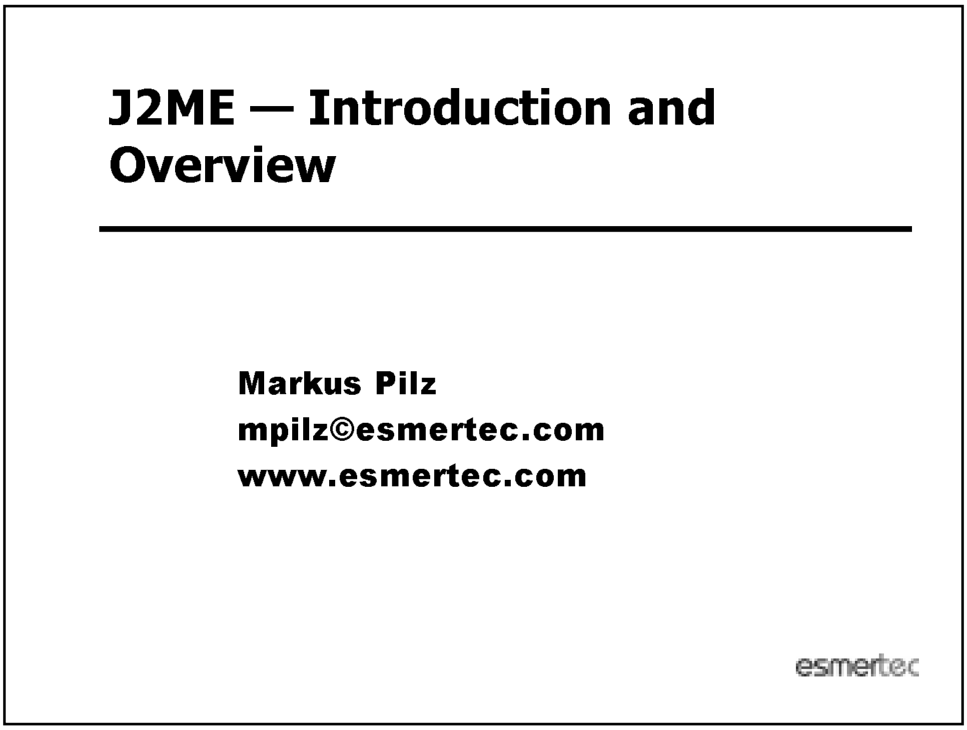
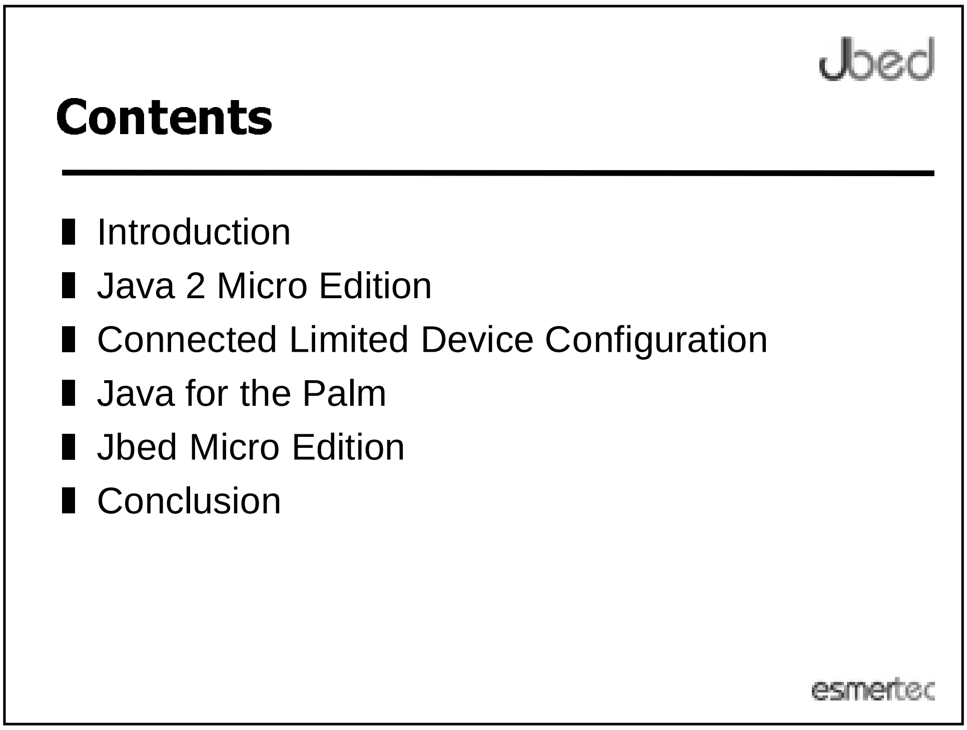


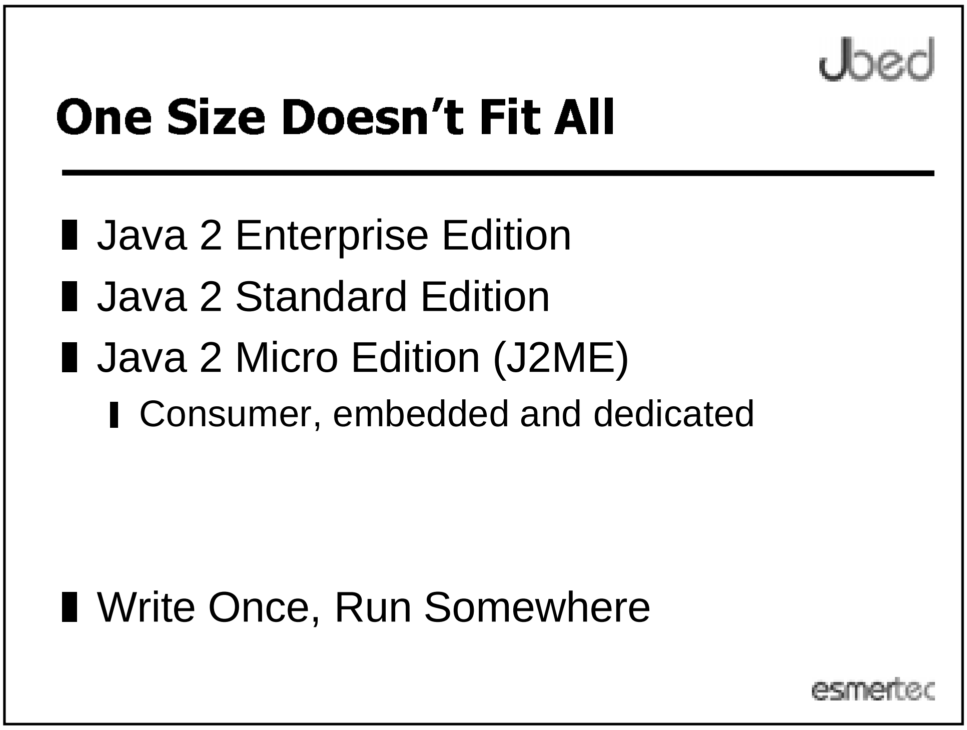
J2ME — Introduction and Overview

Markus Pilz
mpilz©esmertec.com
www.esmertec.com



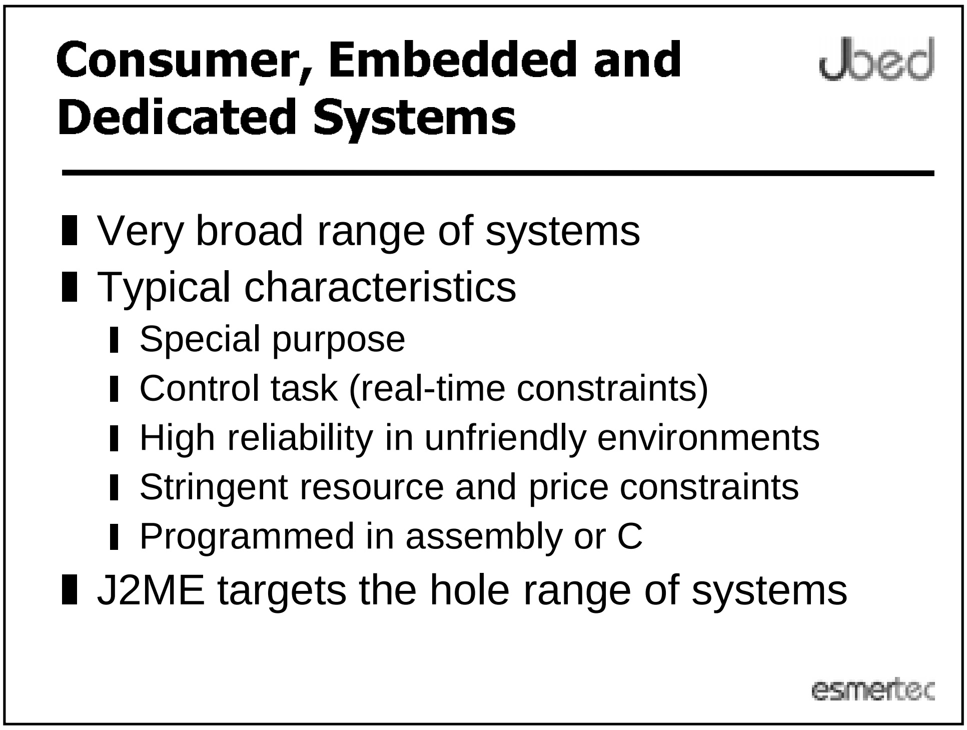
Contents

- Introduction
- Java 2 Micro Edition
- Connected Limited Device Configuration
- Java for the Palm
- Jbed Micro Edition
- Conclusion



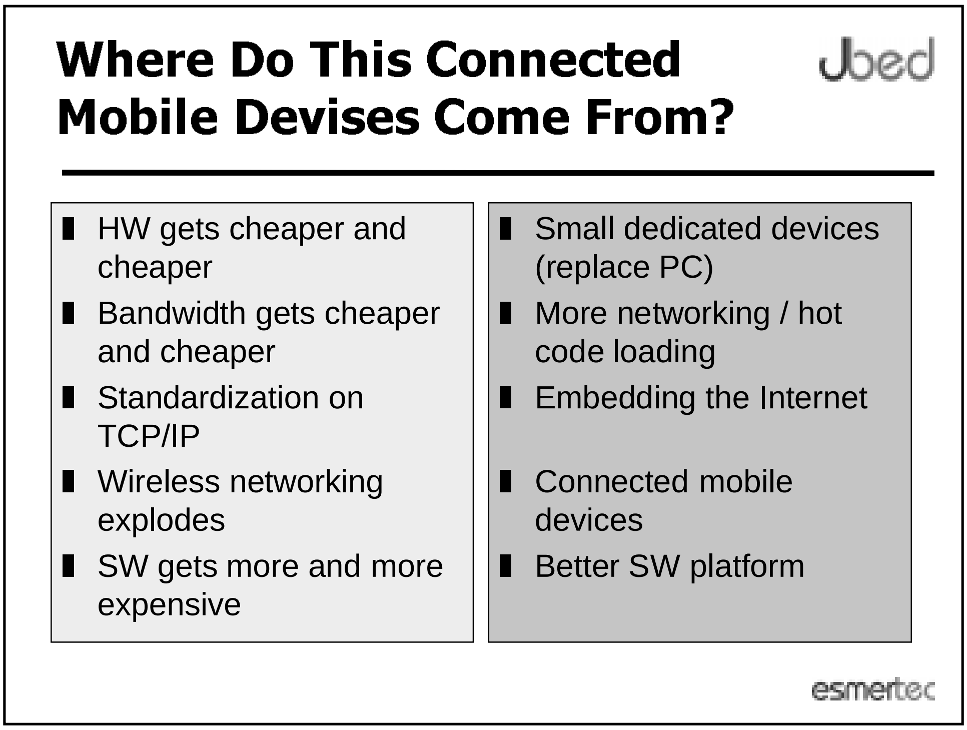
One Size Doesn’t Fit All

- Java 2 Enterprise Edition
- Java 2 Standard Edition
- Java 2 Micro Edition (J2ME)
  - Consumer, embedded and dedicated
- Write Once, Run Somewhere



Consumer, Embedded and Dedicated Systems

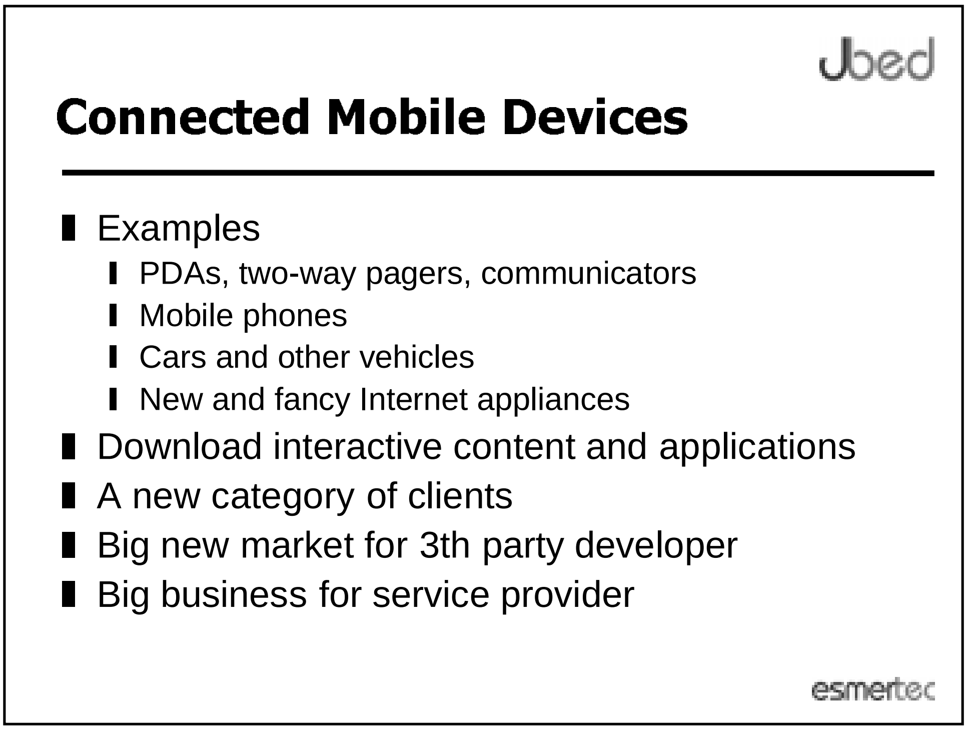
- Very broad range of systems
- Typical characteristics
  - Special purpose
  - Control task (real-time constraints)
  - High reliability in unfriendly environments
  - Stringent resource and price constraints
  - Programmed in assembly or C
- J2ME targets the hole range of systems



Where Do This Connected Mobile Devises Come From?

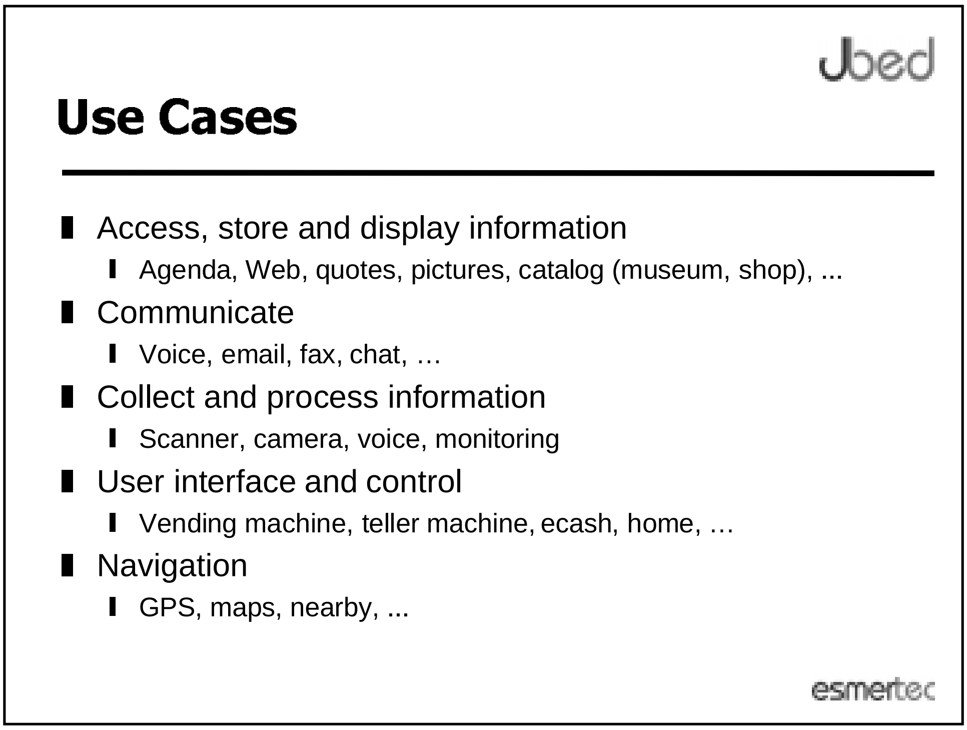
- HW gets cheaper and cheaper
- Bandwidth gets cheaper and cheaper
- Standardization on TCP/IP
- Wireless networking explodes
- SW gets more and more expensive

- Small dedicated devices (replace PC)
- More networking / hot code loading
- Embedding the Internet
- Connected mobile devices
- Better SW platform



Connected Mobile Devices

- Examples
  - PDAs, two-way pagers, communicators
  - Mobile phones
  - Cars and other vehicles
  - New and fancy Internet appliances
- Download interactive content and applications
- A new category of clients
- Big new market for 3th party developer
- Big business for service provider

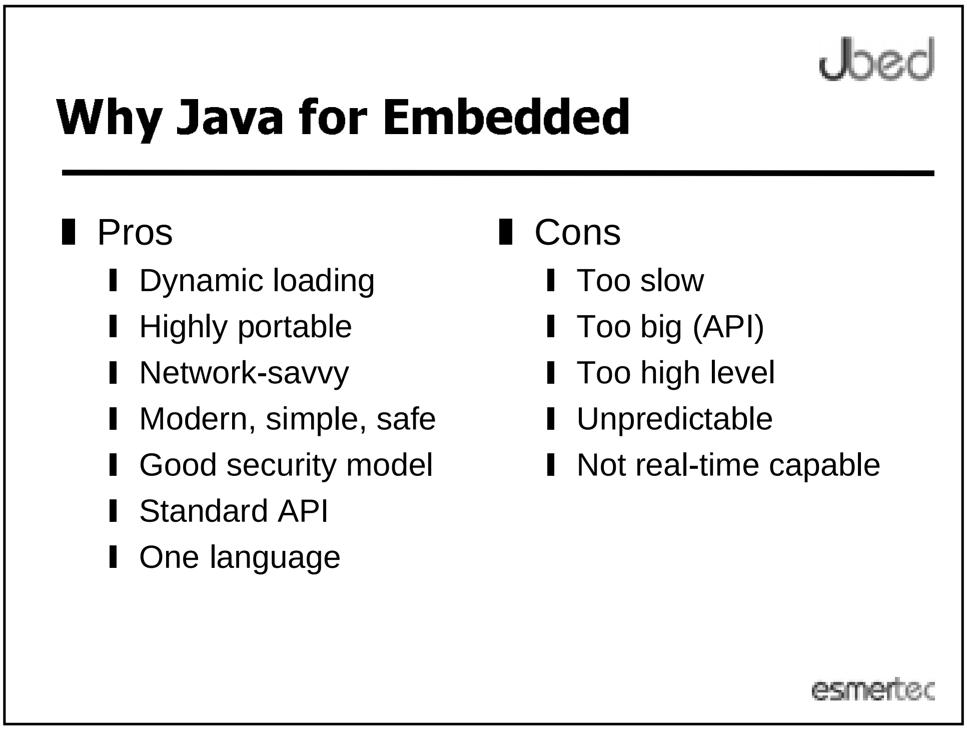


Jbed

Use Cases

- Access, store and display information
  - Agenda, Web, quotes, pictures, catalog (museum, shop), ...
- Communicate
  - Voice, email, fax, chat, …
- Collect and process information
  - Scanner, camera, voice, monitoring
- User interface and control
  - Vending machine, teller machine, ecash, home, …
- Navigation
  - GPS, maps, nearby, ...

esmertec



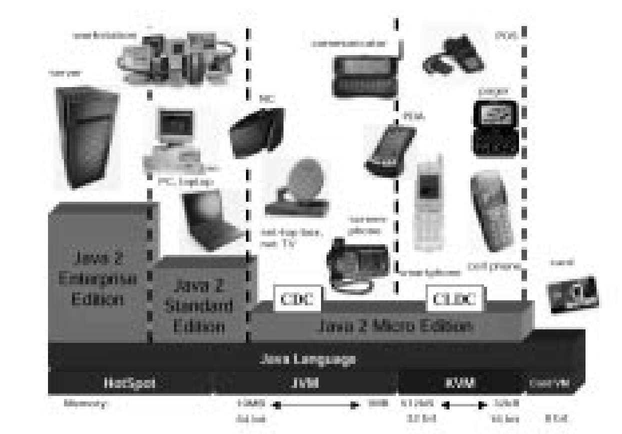
Jbed

Why Java for Embedded

- Pros
  - Dynamic loading
  - Highly portable
  - Network-savvy
  - Modern, simple, safe
  - Good security model
  - Standard API
  - One language
- Cons
  - Too slow
  - Too big (API)
  - Too high level
  - Unpredictable
  - Not real-time capable

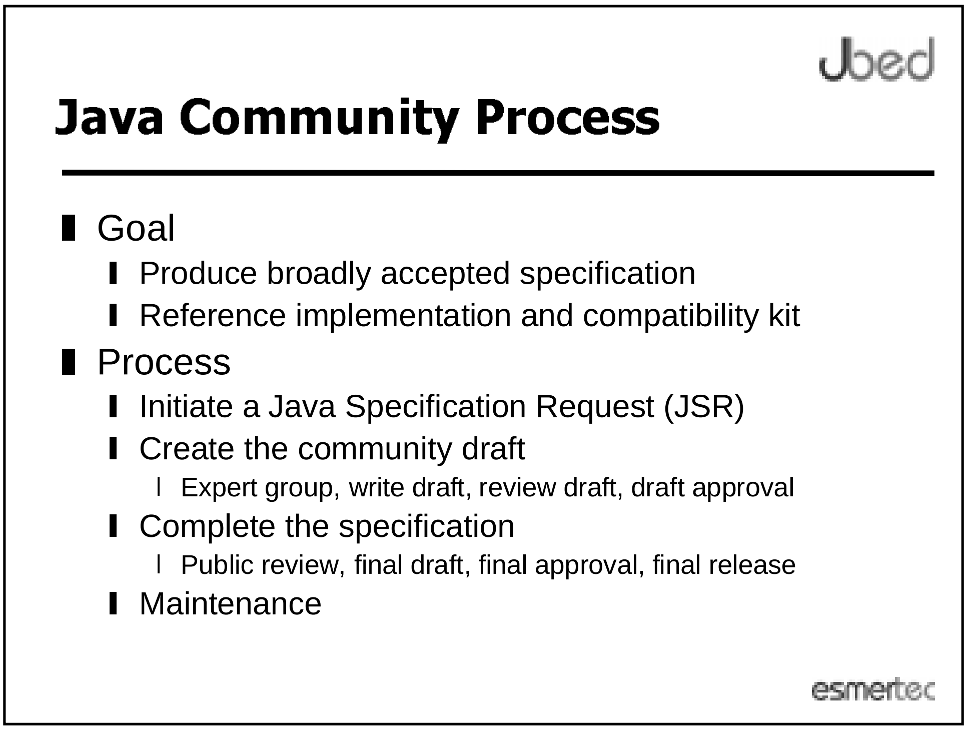
esmertec

Java 2 Micro Edition



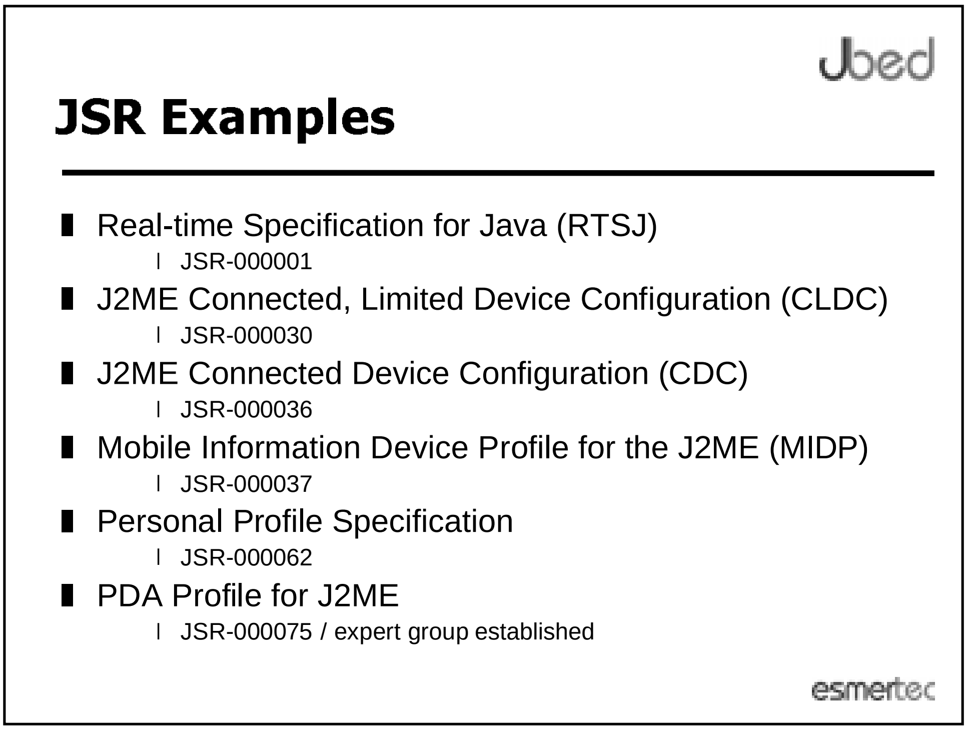
Configuration and Profile

- Configurations
  - A JavaVM and a set of core libraries
    - Connected, Limited Device Configuration (CLDC)
    - Connected Device Configuration (CDC)
    - …
  - Lowest common denominator
- Profiles
  - Set of APIs for a vertical market or device category
    - Mobile Information Device Profile (MIDP)
    - PDA Profile
    - Personal Profile (CDC, successor to PersonalJava)
    - ...



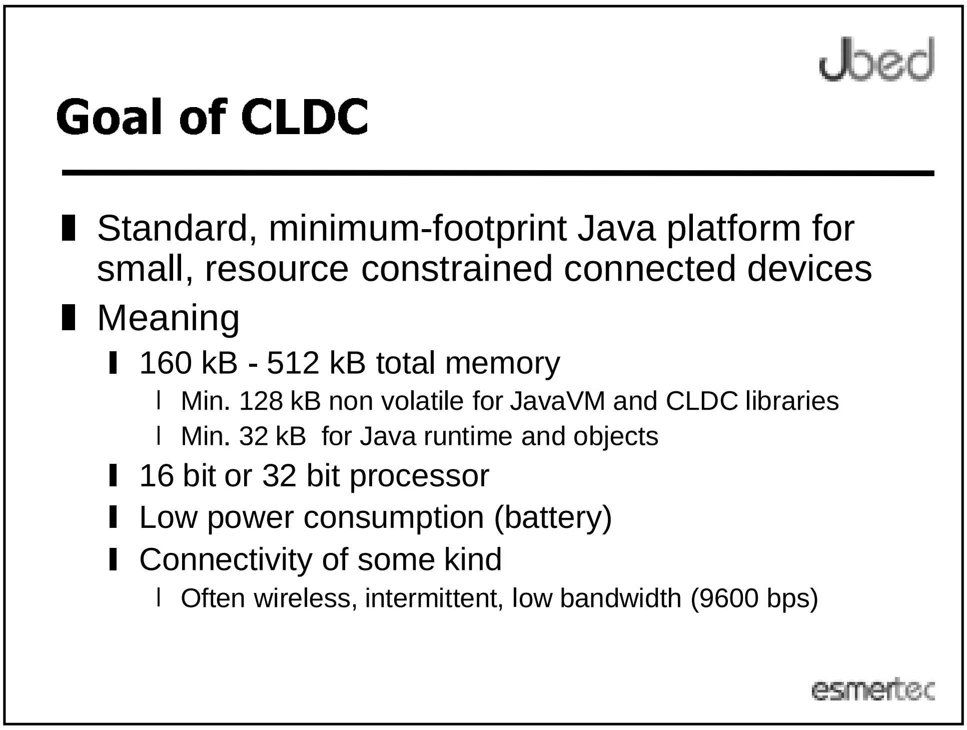
Java Community Process

- Goal
  - Produce broadly accepted specification
  - Reference implementation and compatibility kit
- Process
  - Initiate a Java Specification Request (JSR)
  - Create the community draft
    - Expert group, write draft, review draft, draft approval
  - Complete the specification
    - Public review, final draft, final approval, final release
  - Maintenance



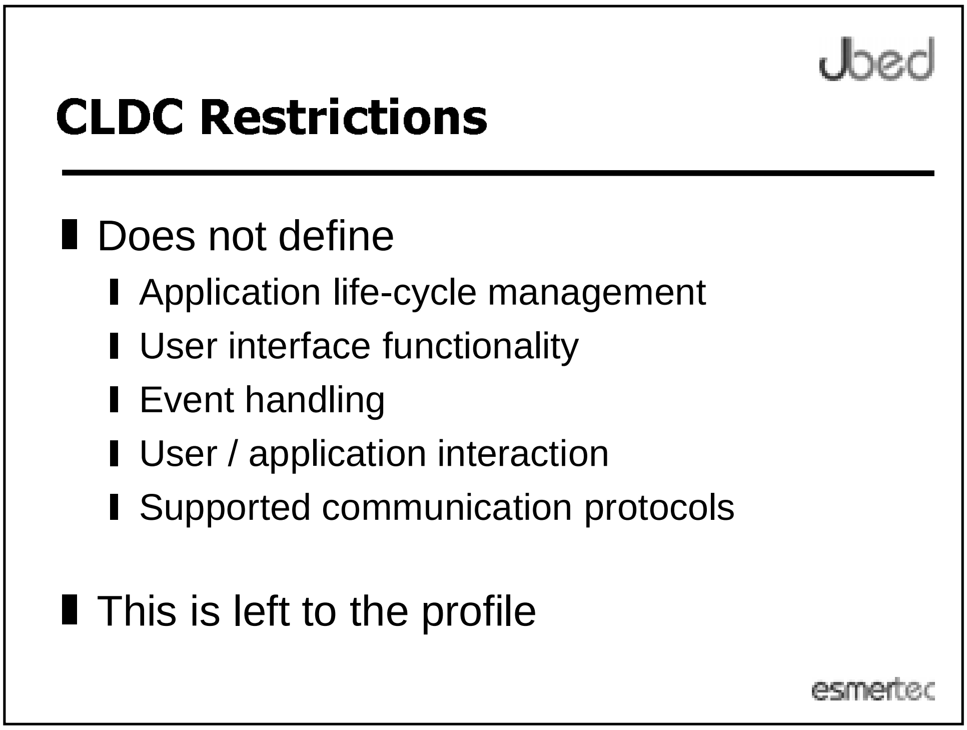
JSR Examples

- Real-time Specification for Java (RTSJ)
    - JSR-000001
- J2ME Connected, Limited Device Configuration (CLDC)
    - JSR-000030
- J2ME Connected Device Configuration (CDC)
    - JSR-000036
- Mobile Information Device Profile for the J2ME (MIDP)
    - JSR-000037
- Personal Profile Specification
    - JSR-000062
- PDA Profile for J2ME
    - JSR-000075 / expert group established



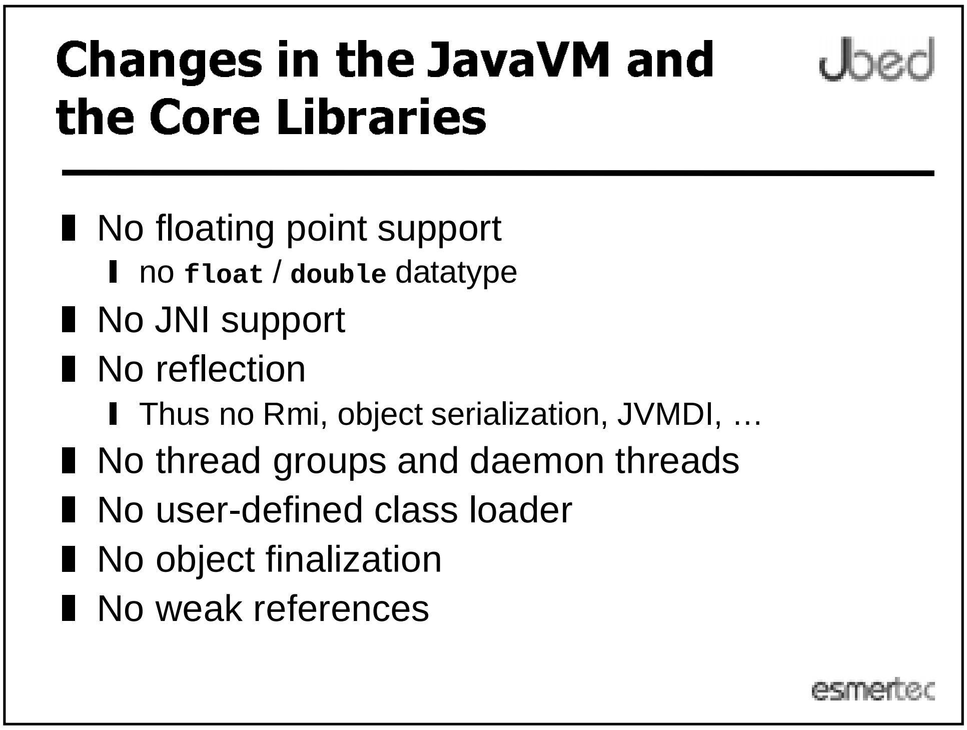
Goal of CLDC

- Standard, minimum-footprint Java platform for small, resource constrained connected devices
- Meaning
  - 160 kB - 512 kB total memory
    - Min. 128 kB non volatile for JavaVM and CLDC libraries
    - Min. 32 kB for Java runtime and objects
  - 16 bit or 32 bit processor
  - Low power consumption (battery)
  - Connectivity of some kind
    - Often wireless, intermittent, low bandwidth (9600 bps)



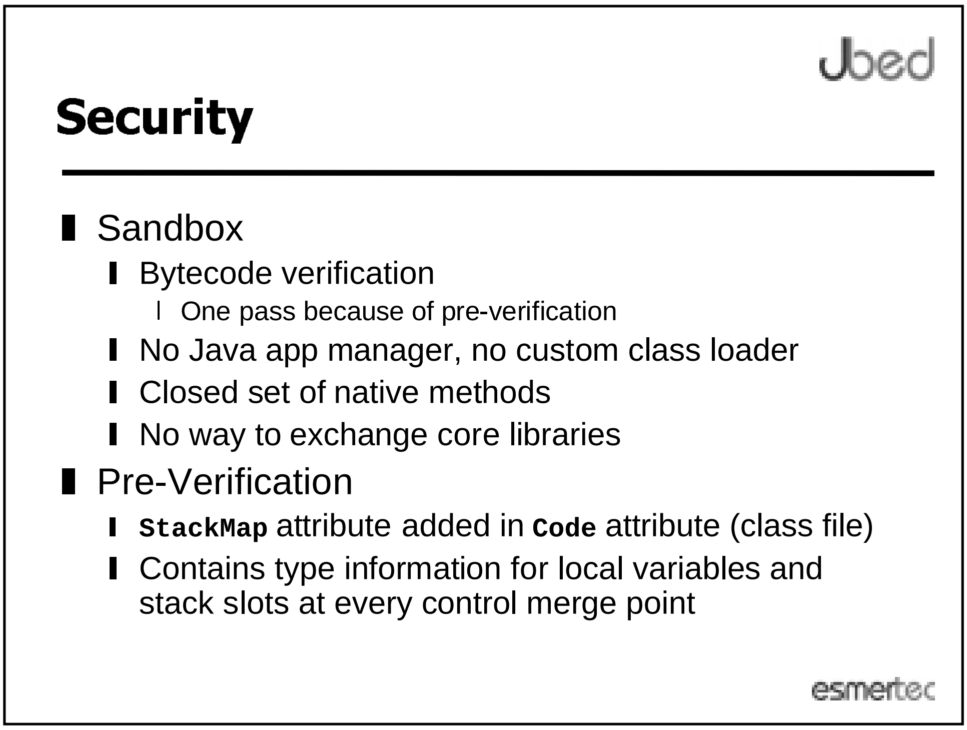
CLDC Restrictions

- Does not define
  - Application life-cycle management
  - User interface functionality
  - Event handling
  - User / application interaction
  - Supported communication protocols
- This is left to the profile



Changes in the JavaVM and the Core Libraries

- No floating point support
  - no `float` / `double` datatype
- No JNI support
- No reflection
  - Thus no Rmi, object serialization, JVMDI, …
- No thread groups and daemon threads
- No user-defined class loader
- No object finalization
- No weak references



Security

- Sandbox
  - Bytecode verification
    - One pass because of pre-verification
  - No Java app manager, no custom class loader
  - Closed set of native methods
  - No way to exchange core libraries
- Pre-Verification
  - `StackMap` attribute added in `Code` attribute (class file)
  - Contains type information for local variables and stack slots at every control merge point

Core Libraries (java.*)

`java.lang`

- `Boolean, Byte, Short, Integer, Long, Character`
- `Object, Class, Runtime, System, Thread, Runnable (I), String, StringBuffer, Throwable`

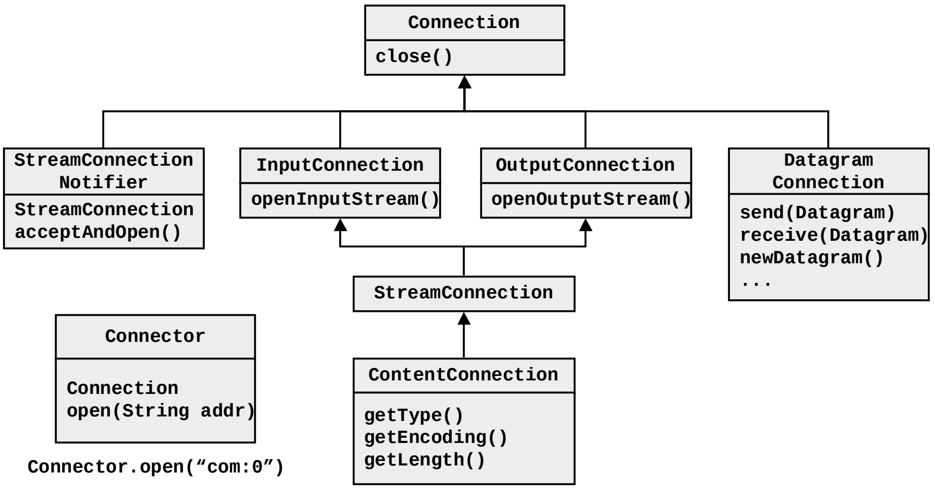
`java.util`

- `Vector, Stack, Hashtable, Enumeration (I), Calendar, Date, TimeZone, Random` (new), `Math` (new)

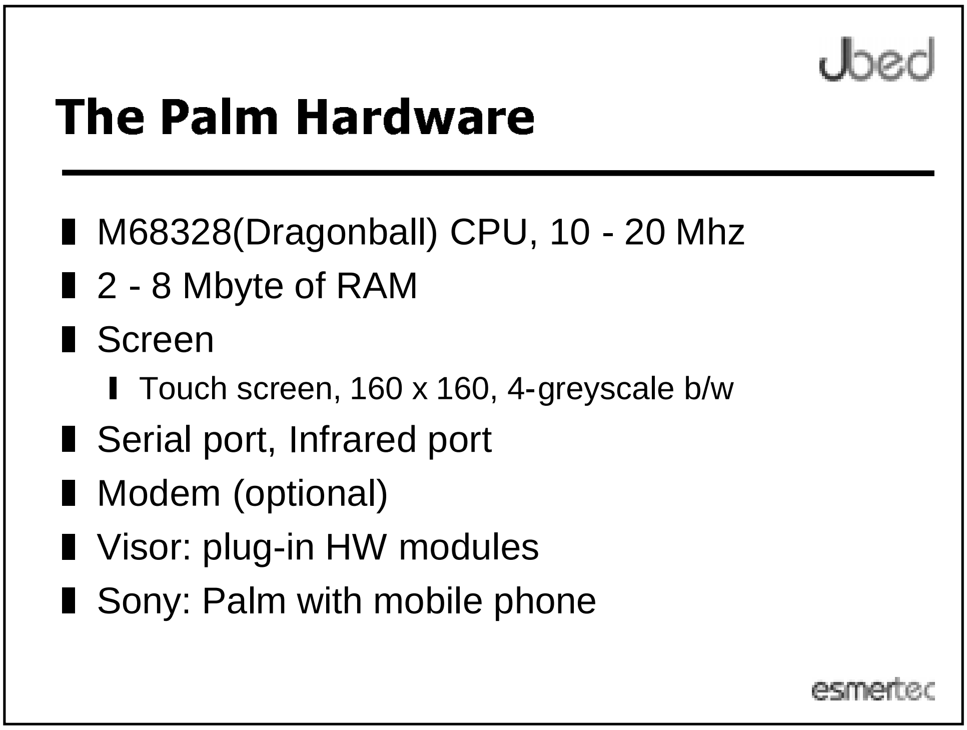
`java.io`

- `InputStream, OutputStream, ByteArrayInputStream, ByteArrayOutputStream, DataInput (I), DataOutput (I), DataInputStream, DataOutputStream`
- `Reader, Writer, InputStreamReader, OutputStreamReader, PrintStream`

Connection Framework (javax.microedition)



`Connector.open("com:0")`

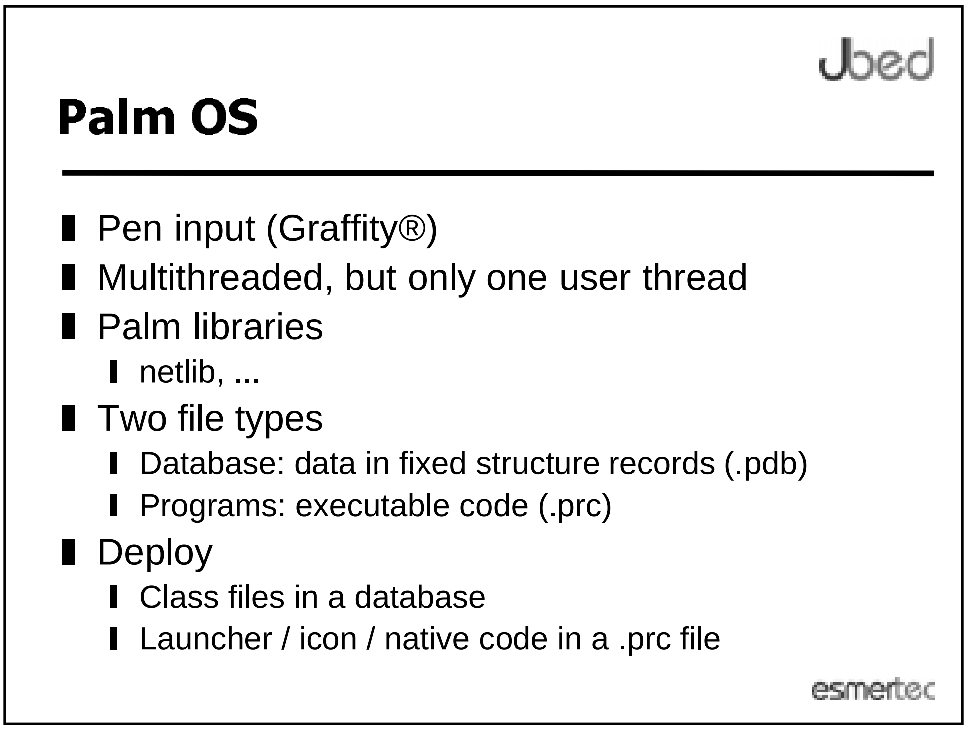


Jbed

The Palm Hardware

- M68328(Dragonball) CPU, 10 - 20 Mhz
- 2 - 8 Mbyte of RAM
- Screen
  - Touch screen, 160 x 160, 4-greyscale b/w
- Serial port, Infrared port
- Modem (optional)
- Visor: plug-in HW modules
- Sony: Palm with mobile phone

esmertec



Jbed

Palm OS

- Pen input (Graffity®)
- Multithreaded, but only one user thread
- Palm libraries
  - netlib, ...
- Two file types
  - Database: data in fixed structure records (.pdb)
  - Programs: executable code (.prc)
- Deploy
  - Class files in a database
  - Launcher / icon / native code in a .prc file

esmertec

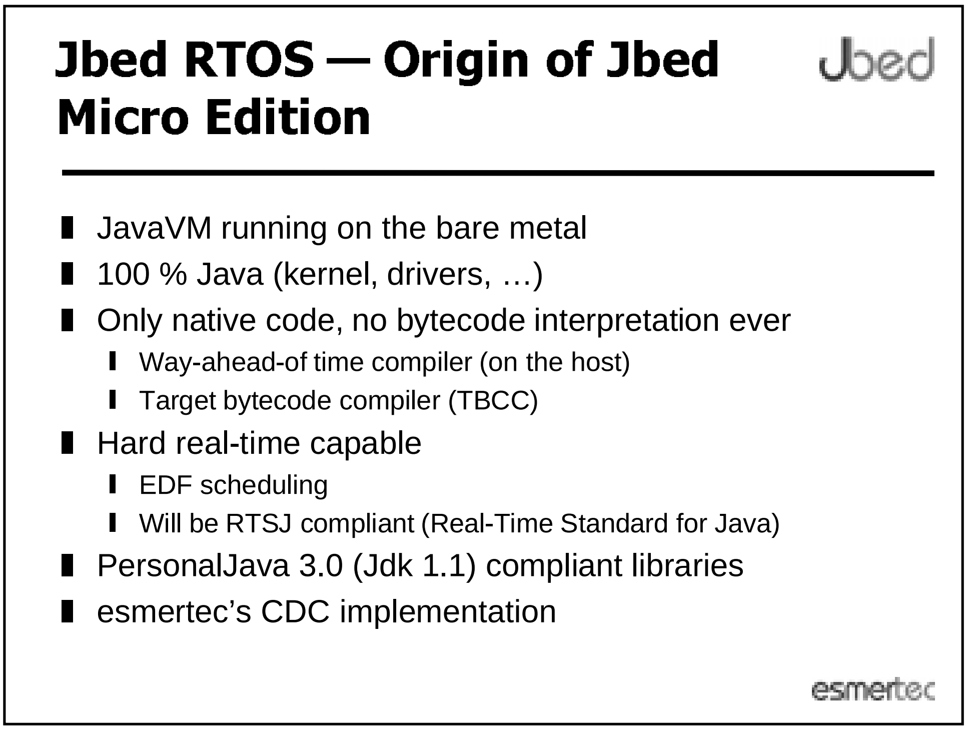
Java for the Palm

- KVM (Sun)
- Jbed Micro Edition (esmertec)
- J9 (IBM)
- Kada (Emwerks)
- …

- Waba (WabaSoft)

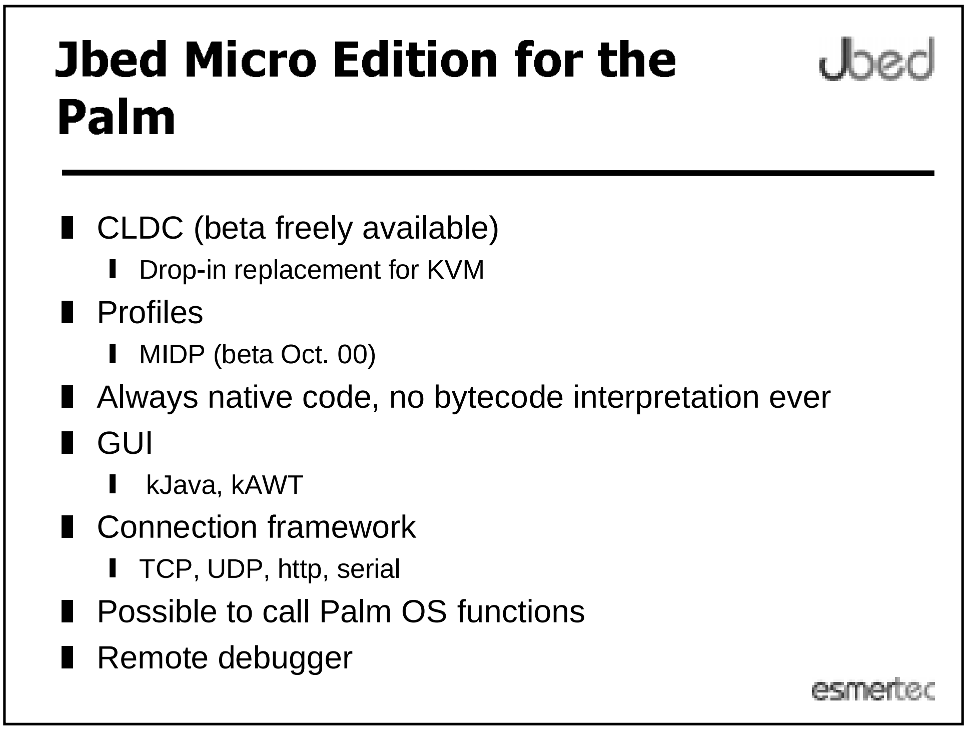
SUN's KVM (Kilobyte VM)

- Reference implementation for CLDC on Palm
  - Preview version 0.1 at JavaOne 1999
  - Stable (but not complete) version since Mai 2000
- Interpreter written in C
- KVM size: 240 kB
- Build cycle
  - Compile Java source to bytecode (`javac`)
  - Pre-verify the class files (`preverify`)
  - Creat .prc file and deploy (`MakePalmApp`)



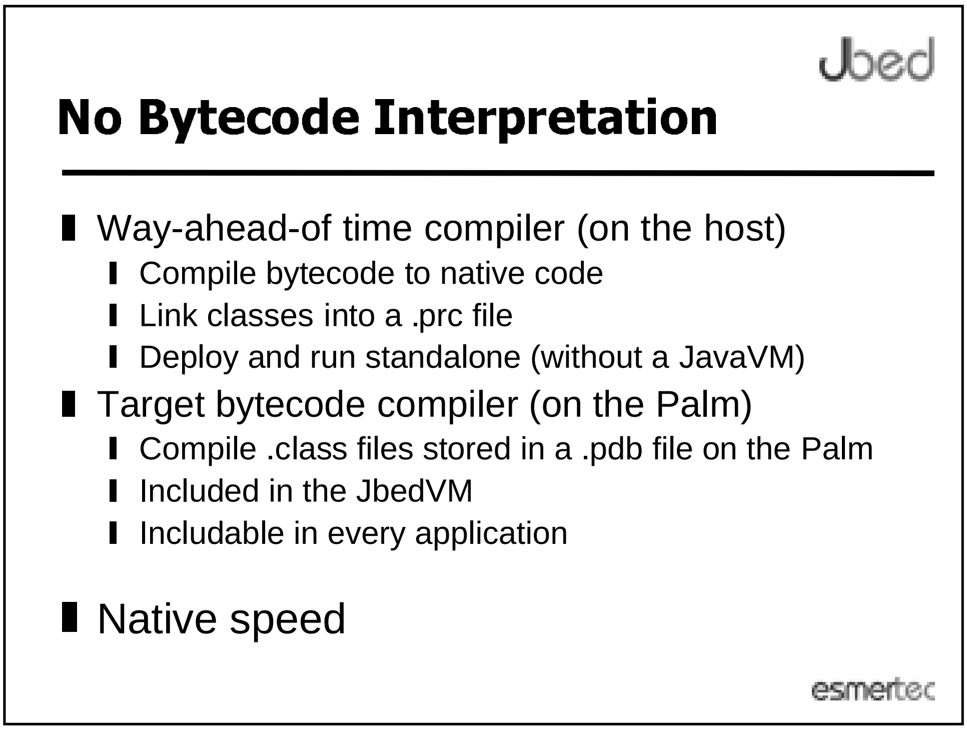
Jbed RTOS — Origin of Jbed Micro Edition

- JavaVM running on the bare metal
- 100 % Java (kernel, drivers, …)
- Only native code, no bytecode interpretation ever
  - Way-ahead-of time compiler (on the host)
  - Target bytecode compiler (TBCC)
- Hard real-time capable
  - EDF scheduling
  - Will be RTSJ compliant (Real-Time Standard for Java)
- PersonalJava 3.0 (Jdk 1.1) compliant libraries
- esmertec’s CDC implementation



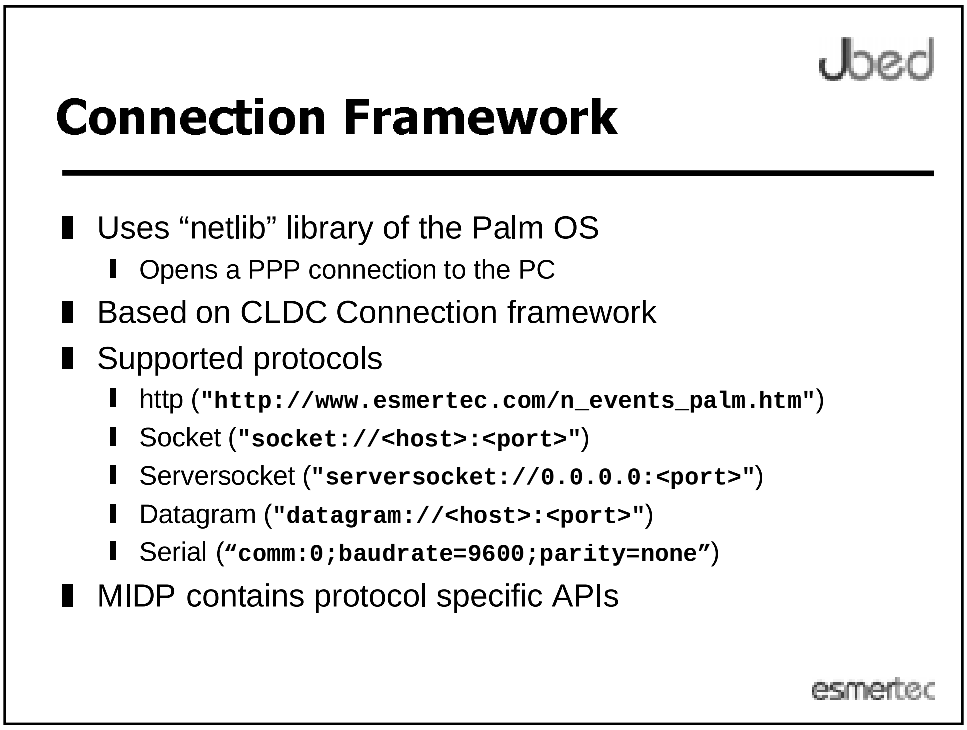
Jbed Micro Edition for the Palm

- CLDC (beta freely available)
  - Drop-in replacement for KVM
- Profiles
  - MIDP (beta Oct. 00)
- Always native code, no bytecode interpretation ever
- GUI
  - kJava, kAWT
- Connection framework
  - TCP, UDP, http, serial
- Possible to call Palm OS functions
- Remote debugger



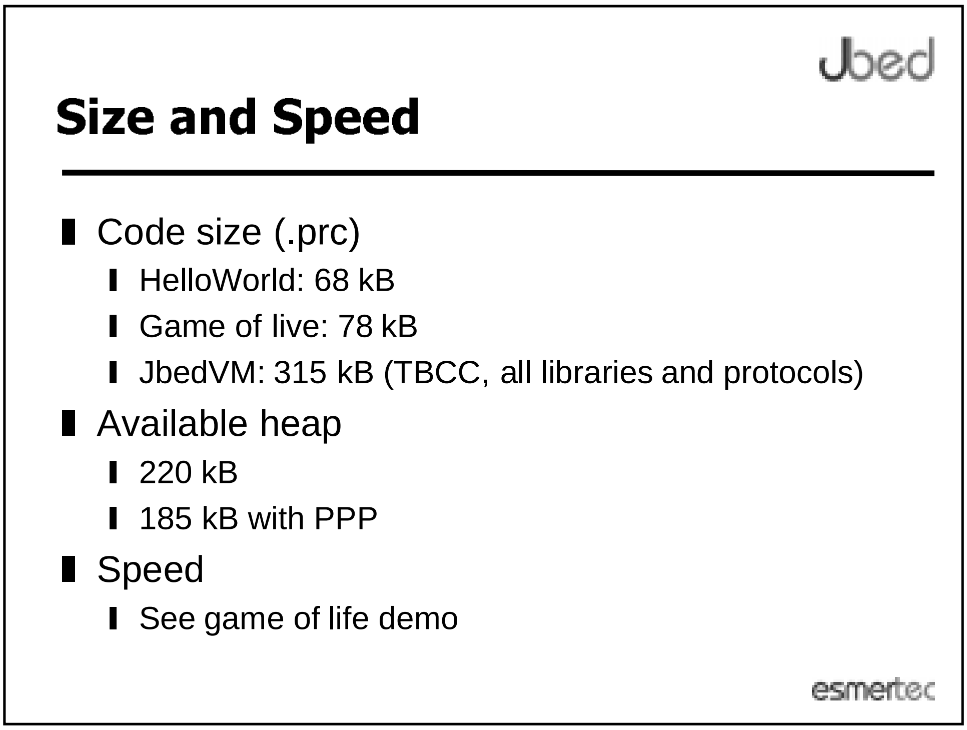
No Bytecode Interpretation

- Way-ahead-of time compiler (on the host)
  - Compile bytecode to native code
  - Link classes into a .prc file
  - Deploy and run standalone (without a JavaVM)
- Target bytecode compiler (on the Palm)
  - Compile .class files stored in a .pdb file on the Palm
  - Included in the JbedVM
  - Includable in every application
- Native speed



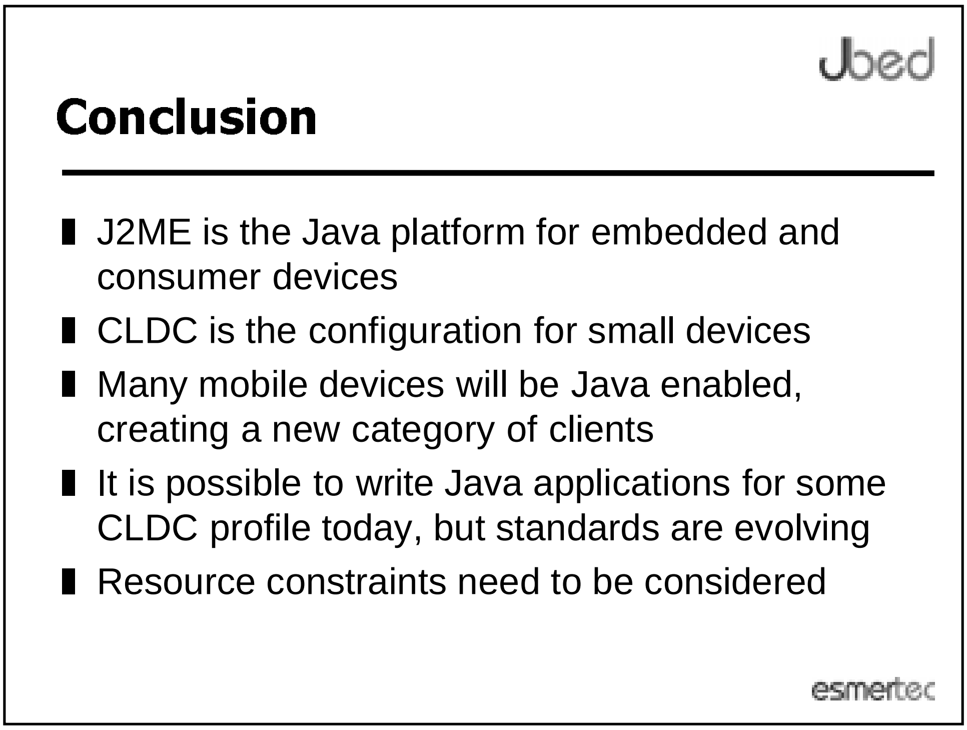
Connection Framework

- Uses “netlib” library of the Palm OS
  - Opens a PPP connection to the PC
- Based on CLDC Connection framework
- Supported protocols
  - http (**`"http://www.esmertec.com/n_events_palm.htm"`**)
  - Socket (**`"socket://<host>:<port>"`**)
  - Serversocket (**`"serversocket://0.0.0.0:<port>"`**)
  - Datagram (**`"datagram://<host>:<port>"`**)
  - Serial (**`“comm:0;baudrate=9600;parity=none”`**)
- MIDP contains protocol specific APIs



Size and Speed

- Code size (.prc)
  - HelloWorld: 68 kB
  - Game of live: 78 kB
  - JbedVM: 315 kB (TBCC, all libraries and protocols)
- Available heap
  - 220 kB
  - 185 kB with PPP
- Speed
  - See game of life demo



Conclusion

- J2ME is the Java platform for embedded and consumer devices
- CLDC is the configuration for small devices
- Many mobile devices will be Java enabled, creating a new category of clients
- It is possible to write Java applications for some CLDC profile today, but standards are evolving
- Resource constraints need to be considered