JavaOne℠

Java Card™ 3: a Platform for Embedded Systems

Saqib Ahmad, Sun Microsystems
Patrick Van Haver, gemalto
Laurent Lagosanto, gemalto

Goal of the presentation

Show why Java Card 3 technology may be the ideal choice for your embedded device!

Agenda

- Java Card 3 has been released !
  - Platform presentation
  - Architectural overview
- Java Card 3 as a platform for embedded systems
  - Exposes usual designs in embedded systems
  - Describe techniques used to reduce system footprint
- Highlight some key features that make the difference
  - Real life use cases
- Discuss possible evolutions for this technology

Java Card 3 Editions

- Two stand-alone "Editions"

Java Card Platform, Classic Edition

- Traditional smart card architecture
- Within the current JC 2.x memory constraints
- APDU – based

Java Card Platform, Connected Edition

- Faster CPU
- Larger memory
- Network-oriented
- High-speed interface

Core Java Card functionalities

Security (firewall, crypto, ...), Backward compatibility

Java Card Platform – Connected Edition

Targeted Hardware – Traditional vs. High-end

8/16 bit CPU	**32 bit CPU**
~2K RAM	**16K RAM**
48–64K ROM	**>256K ROM**
8–32K EEPROM	**>128K EEPROM, or several MB Flash**
External Clock: 1–5 Mhz	**Internal Clock—50 Mhz**
Serial I/O Interface ● 9.6–30K Baud ● Half duplex	**High Speed Interfaces** ● **1.5 Mb/s-12 Mb/s** ● **Full duplex**

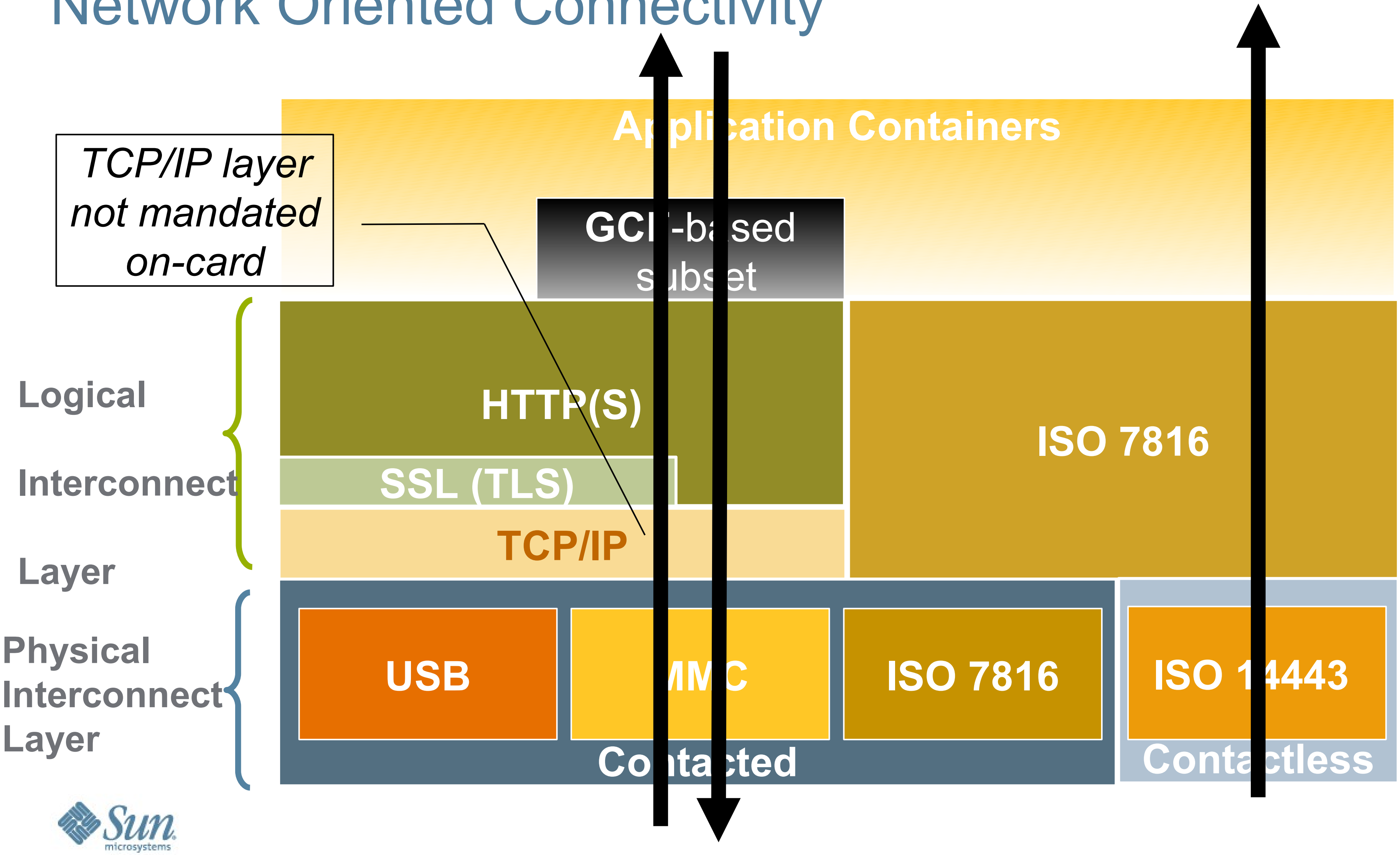
Java Card Platform – Connected Edition

Virtual Machine

- Based on a subset of CLDC
- Enhanced with Java SE 1.6 features
- Enhanced with Java Card-specific features
  - Firewall-based context isolation
  - Transaction Support
  - VM and object persistence

Java Card Platform – Connected Edition

Network Oriented Connectivity



TCP/IP layer not mandated on-card

Application Containers

GCF-based subset

Logical Interconnect Layer

HTTP(S)

SSL (TLS)

TCP/IP

ISO 7816

Physical Interconnect Layer

USB

MMC

ISO 7816

Contacted

ISO 14443

Contactless

Java Card Platform – Connected Edition

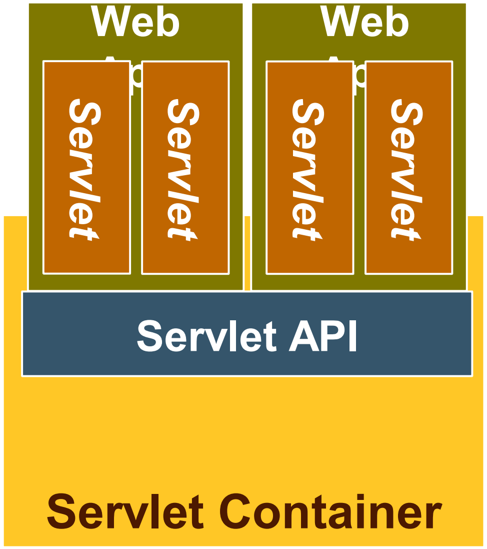
Enhanced Programming Model

- More developer-friendly, closer to main-stream Java
- Multi-threading
- Transaction demarcation with annotations
- Persistence by reachability
- Enhanced inter-application communication
- Generic Connection Framework
- Evolutive Cryptography
- 2 Application models

Java Card Platform – Connected Edition

Network Oriented Connectivity

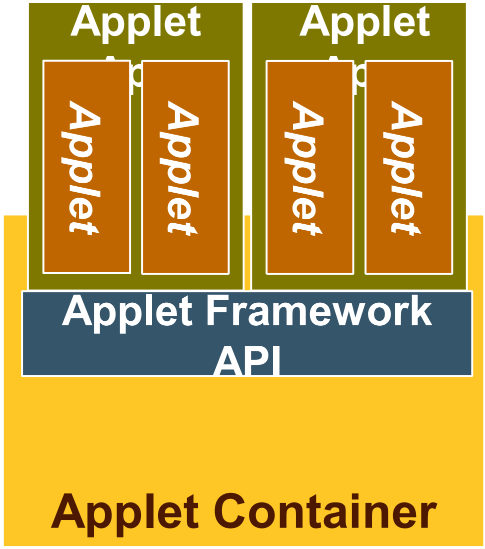
- Subset of Java EE Servlet Specification
- Application components are Servlets, Filters and event listeners
- Each application may be hosted on a dedicated secure port



Java Card Platform – Connected Edition

APDU-based Application Environment

- APDU-based applet model
- Application components are applets
- Two types of applets
  - Classic
  - Extended



Java Card Platform – Connected Edition

Security

- Code Isolation
- Context Isolation
- Access Control
  - Permission-based security
  - Role-based security
  - User Authentication
  - On-Card client authentication

Java Card Platform - Availability

- Specifications for classic and connected editions
- Development kits for classic and connected editions
  - Emulators
  - Tools
- Downloadable from http://java.sun.com/javacard
- Application development supported in NetBeans 7

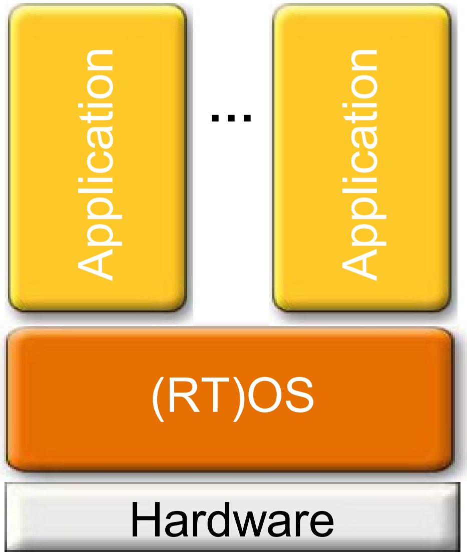
Agenda

- Java Card 3 has been released !
  - Platform presentation
  - Architectural overview
- Java Card 3 as a platform for embedded systems
  - Exposes usual designs in embedded systems
  - Describe techniques used to reduce system footprint
- Highlight some key features that make the difference
  - Real life use cases
- Discuss possible evolutions for this technology

Embedded systems

usual approaches

- Most of developers are only considering native OS to develop their embedded application



Rationale

- Porting effort made at OS level
- Benefit from OS services portfolio

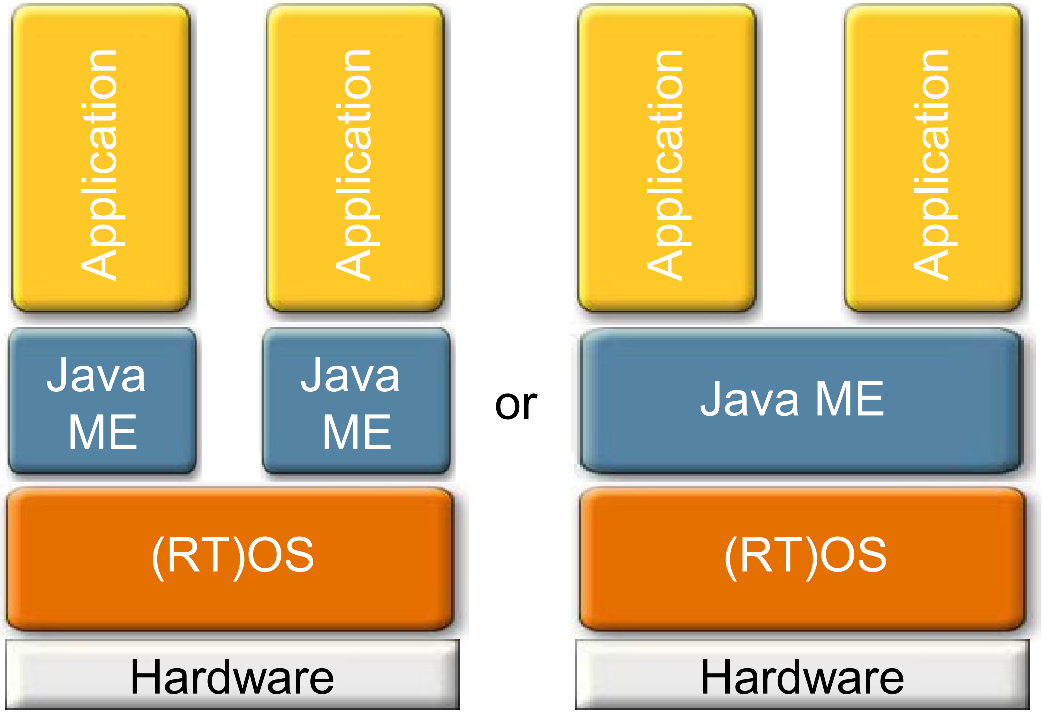
But

- Integration of multiple applications more complex
- Requires specific tools, specific skills, …
- Scalability & extensibility issue: difficult to switch to another OS

Embedded systems

usual approaches

- Adding Java in the stack solves some issues



- Focus on domain problems
- Better scalability and extensibility through profiles
- Integration within Java range (tools, community…)

But

- Higher cost (BOM)

Embedded systems

usual approaches

- These technologies are used in many devices

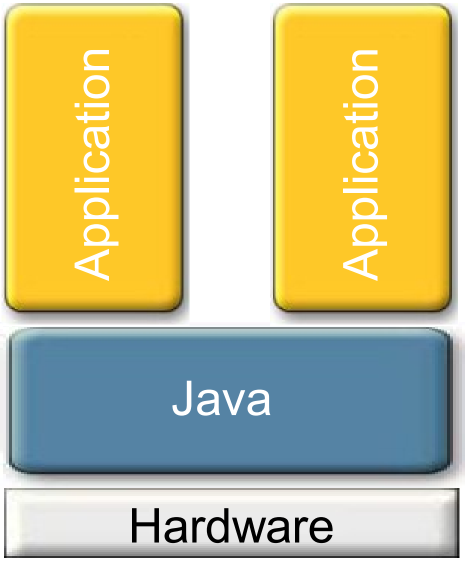
Which technology for Smart Objects ?

- More and more Small and Connected objects
- Require smaller platforms

Which technology for Smart Objects ?

Steps towards size reduction

- Java on the bare metal



- Keep Java benefits
- Smaller footprint

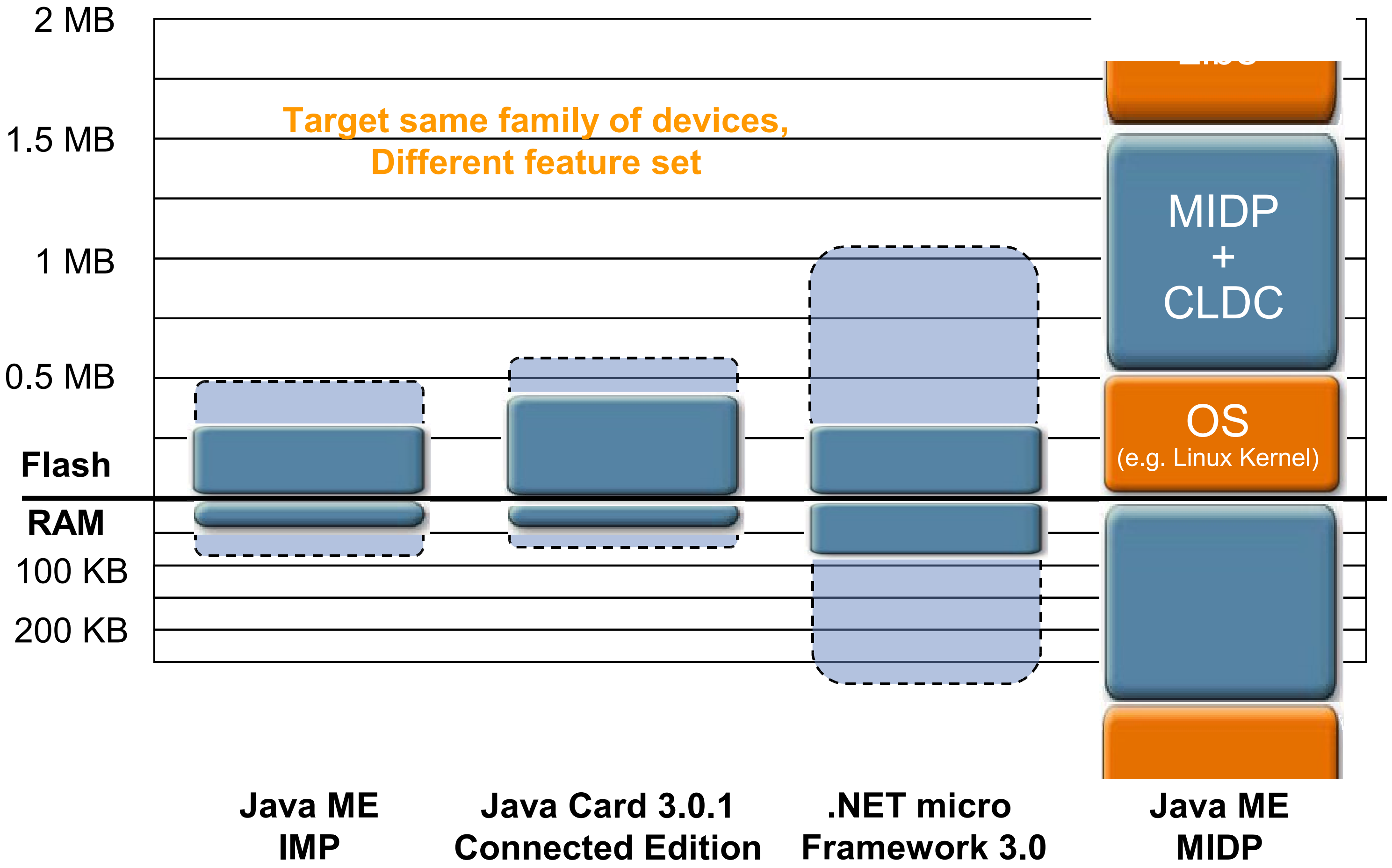
However

- Specific VM features are required to get the most of this architecture

Candidate Technologies

- Java ME stack
  - CLDC 1.1 with MIDP or IMP
  - Can be adapted to bare metal
  - Rich set of API
- .NET micro framework 3.0
  - Small footprint
  - Available either on host OS or directly on metal
- Java Card 3.0.1 Connected Edition
  - Originally designed for very constrained devices
  - Once smart-card centric system, now a more general purpose platform
  - Standalone, self contained and consistent specification

System footprint



Target same family of devices,
Different feature set

Target another
family of devices

Java Card 3.0.1 Connected Edition
Footprint reduction techniques

- Linking performed when code loaded on device
  - Code converted into its executable form and stored into Flash
  - Verification, factorization and optimizations performed **once**
  - ➢ Significantly reduce the stored size of the code

- Package sealing
  - Packages can be marked as non extensible
  - Gives opportunity to apply code optimizations
  - ➢ Reduce code size

Java Card 3.0.1 Connected Edition
Footprint reduction techniques

- In place execution (XIP)
  - Code executed directly from Flash (or ROM)
  - No need to copy it in RAM
  - ➢ Significantly reduce RAM consumption, hence budget

- Persistent Memory Model
  - Exact Garbage Collection and object promotion in NVM
  - ➢ Reduce RAM consumption

Java Card 3.0.1 Connected Edition

Footprint reduction techniques

- Single VM with Firewall to provide isolation between applications
  - Better use of system resources
  - Maintain a strict isolation of applications (controlled sharing)
  - Significantly reduce RAM consumption
  - May suppress the need for MMU or MPU

Agenda

- *Java Card 3 has been released !*
  - *Platform presentation*
  - *Architectural overview*
- *Java Card 3 as a platform for embedded systems ?*
  - *Exposes usual designs in embedded systems*
  - *Describe techniques used to reduce system footprint*
- *Highlight some key features that make the difference*
  - *Real life use cases*
- *Discuss possible evolutions for this technology*

Java Card 3.x unique features

at this memory budget (1/2)

- Powerful execution engine
  - CLDC-like VM with Java Card extensions
  - Java 6 language extensions (generics, annotations)
- Extensive connectivity capabilities with GCF
  - TCP client & server, UDP, TLS, Http client
- Strong security model
  - Crypto APIs
  - Isolation & Acces-control

Java Card 3.x unique features

at this memory budget (2/2)

- Web Server & Web Container
  - Web application support (Servlets, Filters, …)
  - Subset of Java Servlet 2.5 specification
  - Same deployment format (off-the shelf tools)

- Remote manageability
  - Most Java Cards come with Global Platform on-board
  - Global Platform Card Specification 3.0 almost there
  - Adapted to both APDUs and HTTP-based administration

Real life Use Case – Healthcare USBkey

- Experimentation: Electronic Health Records
  - *EHR data stored on a NAND-flash hosted DBMS*
  - *Accessed by applications through a JDBC subset*
  - *Hosted on a Java Card 3.0 powered web platform*
  - *Synchronized with remote servers over secure link (TLS)*
  - *Form factor: USB Key*
  - *HW: Smart-card Microcontroller + NAND Flash*

- BOF-4576: Demonstration of Electronic Health Records (EHR) on Java Card™ 3.0 Technology-Based Devices
  - Jean-Jacques Vandewalle, Gemalto
  - Nicolas Anciaux, INRIA

Real life Use Case – Sensors / M2M

- Advantages of Java Card 3 Connected:
  - From the start compatibility with battery-less devices
  - Connectivity at IPv4 or IPv6 level
  - Http client stack: devices can “push” information
  - Http server stack: devices can be remotely managed

- Additional benefits
  - Performance characteristics & multithreading capabilities may help writing device drivers & protocols directly in Java

Real life Use Case – Digital Home

- Digital Home Device requirements (UPnP/DLNA)
  - IP based communication
  - Http Server to offer description, UI and service
  - XML parsing / generation capabilities

- Java Card 3 Connected is a good candidate
  - For “enabling” an existing device, as an extension
  - For building new kind of devices, as a foundation

Agenda

- Java Card 3 has been released !
  - Platform presentation
  - Architectural overview
- Java Card 3 as a platform for embedded systems ?
  - Exposes usual designs in embedded systems
  - Describe techniques used to reduce system footprint
- Highlight some key features that make the difference
  - Real life use cases
- Discuss possible evolutions for this technology

Java Card 3 availability & ownership

- Specifications & tools
  - Java Card 3.0 available since march 31st 2008
  - Java Card 3.0.1 Release, JavaOne 2009
- Defined by the Java Card Forum & Sun
  - More info at www.javacardforum.org
- Owned & Licensed by Sun
  - More info at java.sun.com/javacard

Java Card 3.x evolution for the embedded space

- Opportunity of a “Static Edition” ?
  - Idea: removing the notion of dynamic loading
  - Built on the already well-known ROMization techniques

- Rationale:
  - Footprint reduction
  - Limited device lifetime
  - Focus on data update only (not application update)

Java Card 3.x evolution for the embedded space

- Removing APDUs & Applet Container
  - Useful in smart card targets only
- Rationale:
  - Footprint reduction
  - Complexity reduction

Java Card 3.x evolution for the embedded space

- Adding new Application Models
  - “main” entry point application model

- Leveraging on Servlets
  - javax.servlet.Servlet is independent from Http
  - Already supported in Java Card 3
  - Relevant if there’s a notion of “on demand service”

Java Card 3.x evolution for the embedded space

- Support for Real-Time Java
  - JSR-001 RTSJ claims compatibility with CLDC
  - It covers both timing aspects and “raw” memory access

- Areas for investigations
  - Garbage collection predictability in a persistent memory context
  - Collaboration/interactions between the Java Card Firewall and RTJS memory checks.
  - ...

Wrap Up

- There's a new Java Card in town
- It's packed with new features
- You may consider it when building those "things" in the "Internet of Things"

Thank You

Saqib.Ahmad@sun.com
Patrick.Van-Haver@gemalto.com
Laurent.Lagosanto@gemalto.com