Java Technology in the Small IoT Devices

Eclipse IoT Day, Grenoble, March 9th 2017

https://projects.eclipse.org/projects/iot.edje
ABOUT THE PRESENTER

Laurent Lagosanto
Senior Architect at MicroEJ®
18 years of “in the Small” activities, mostly about Java Technology

laurent.lagosanto@microej.com

The information contained herein is not warranted to be error-free.
MicroEJ® and all relative logos are trademarks or registered trademarks of IS2T S.A. in France and other Countries.
Java™ is Sun Microsystems’ trademark for a technology for developing application software and deploying it in cross-platform, networked environments. When it is used in this site without adding the "™" symbol, it includes implementations of the technology by companies other than Sun. Java™, all Java-based marks and all related logos are trademarks or registered trademarks of Sun Microsystems Inc, in the United States and other Countries.
Other trademarks are proprietary of their respective owners.
ECLIPSE IOT : THREE VERTICALS

CONSTRAINED DEVICES

GATEWAYS AND SMART DEVICES

IOT CLOUD PLATFORM

Checkout Eclipse IoT Whitepaper from 2016 Q4
JAVA TECHNOLOGIES ARE PERVERSIVE, REALLY?

Monolithic Firmwares
RTOS + Apps
Native Protocol Stacks
Eclipse IDEs

Java Applications
Java OSGi Containers
Java IoT Protocol Stacks
Eclipse IDEs

Java Web Services
Java EE Engines
Java IoT Protocol Stacks
Eclipse IDEs

March 9th, 2017
Eclipse IoT Day Grenoble
JAVA TECHNOLOGIES IN THE SMALL, WHY NOT?

« Too big »
« Too complex »
« Too slow »
« Not secure »
« I don’t need dynamic loading »
« I don’t need a Java Virtual Machine, I have Linux »
LET’S DEBUNK SOME MYTHS...

Size can be controlled, if you pick the right features & APIs (ever heard of Java Card ?)

You are NOT forced to interpret bytecodes, you know !

Linux cannot go NOT everywhere ! (Cortex-M anyone ?)

No dynamic loading ? So your product software will never be updated ?

A Virtual Machine provides sandboxing, even if you don’t have an MMU

Bytecodes can be downloaded safely : they can be verified

An automatically managed memory is safer, more efficient, than if statically allocated

March 9th, 2017

Eclipse IoT Day Grenoble
AND SOME OTHER ADVANTAGES:

Java Language is Object Oriented and helps build Services Oriented architectures (think about what you can do with Class.forName(...) or, instanceof ...)

Java APIs are well defined, and cover a wide functional range (Language level APIs, I/Os, Collections, Net, TLS, Junit, Crypto, ...)

Java Technologies are part of a huge ecosystem: developers, tools, trainings, open source projects and communities
MICROEJ AND ECLIPSE : THE EDJE PROJECT

A Java library at the edge of the IoT for sensor hubs and devices
Edje focuses on the following aspects:

- Controller Communication Interfaces
- Serial Port
- CAN
- Analog Input
- GPIO
- USB Devices
- LCD
- Digital and Analog I/O
- Peripheral Management
EDJE IS ALSO ABOUT DEFINING A COMMON API

List of minimum Java API

The minimum execution environment provided by an Edje-compatible device

Intersection between Java SE, Java SE Embedded, MicroEJ and Android
List of minimum Java API

The minimum execution environment provided by an Edje-compatible device
Intersection between Java SE, Java SE Embedded, MicroEJ and Android
How do you define such an API set:

- By looking at what customer applications need

- By trying to run useful open source libraries / services
  (Eclipse Paho, Eclipse Leshan, Eclipse Californium, cf the next presentation)

- By looking at other APIs on the market
  - Android (Things)
  - JDK Device I/O (Kura)

- It’s an ongoing effort
  MicroEJ 4.1 is due next month, Embedded World is next week, EclipseConverge is next.... expect some news...
ABOUT MICROEJ

Independent Software Vendor, global player in the embedded IoT industry

• http://www.microej.com/about
• Software tool & runtime licenses
• Professional services, training & consulting

KEY FINANCIALS

• $5M Series B funding end 2011 – $20M R&D total investment

Offices in France, Germany, USA

Expertise in embedded, virtualization, software engineering & process

Partnerships with key IoT, silicon, embedded SW and HW (EMS) vendors
PARTNERS & CONSORTIA – STRONG BUSINESS PARTNERS

HARDWARE

SOFTWARE

CLOUD

SERVICES

ALLIANCES
OPERATING SYSTEMS FOR ENABLING THE IOT

75%
IoT = “sub-gig” things:
• Processor < GHz
• Memory < GB

25%

Volume

SW Foundation

RTOS

MCU ($1-6 ASP)

MPU (>$10 ASP)

Android

microej®

March 9th, 2017

Eclipse IoT Day Grenoble
AN EXAMPLE OF FOOTPRINT REDUCTION

63% Bill of Materials savings
87% power consumption savings

<table>
<thead>
<tr>
<th></th>
<th>RTOS (Linux) + Java platform</th>
<th>RTOS (Any/None) + Java platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>410 mAh – batteries recharge every day</td>
<td>Batteries recharge every two or three weeks</td>
</tr>
<tr>
<td>Processor</td>
<td>Qualcomm Snapdragon 400 MSM8226, ARM based Quadcore 1.2 GHz</td>
<td>ARM Cortex-M4 based MCU, 100 MHz</td>
</tr>
<tr>
<td>RAM (data)</td>
<td>256 MB</td>
<td>Less than 0.5 MB</td>
</tr>
<tr>
<td>FLASH (Code + Resources)</td>
<td>2,000 MB</td>
<td>2 MB</td>
</tr>
<tr>
<td>Boot Time</td>
<td>35,000 ms</td>
<td>50 ms</td>
</tr>
</tbody>
</table>
MICROEJ FLAGSHIP PRODUCT LINE

SDK: OS & TOOLS FOR DEVICE SOFTWARE DEVELOPMENT

STORE FOR APPLICATION DEPLOYMENT

STUDIO FOR APPLICATION DEVELOPMENT
MICROEJ SDK COMPONENTS

**Firmware**
- **RESIDENT APPLICATIONS**
  - App 1
  - App 2
  - App 3
- **OPERATING SYSTEM**
  - Libraries
  - Core
  - BSP
  - Legacy

**Workbench**
- **GUI Tools**
- **Application Tools**
- **OS Builder**
- **Virtual Device Builder**
- **Simulator**
- **IDE**

**TARGET**

**HOST**
SINGLE-APP DEVELOPMENT WORKFLOW

MicroEJ SDK
- Platform development
- App development

MicroEJ Studio
- App development

MicroEJ SDK
- Firmware development

Standalone Application
- MicroEJ OS + Libraries + BSP

Monolithic MicroEJ Firmware

Program

EXECUTABLE

Sandboxed App binaries (built for a specific firmware)

Deployment to Infrastructure

Dynamic Load & Install

MULTI-APP DEVELOPMENT WORKFLOW

MicroEJ Studio
- App development

MicroEJ SDK
- Firmware development

App

App

App

MicroEJ Store

MicroEJ Firmware

MicroEJ Firmware

MicroEJ Firmware

March 9th, 2017
WHAT TYPES OF PRODUCTS CAN YOU BUILD?

Sowee was announced by EDF at CES 2017

The offer includes a remote controller that runs a mix of MicroEJ and Eclipse IoT Java Technologies:
LET’S SEE THIS IN ACTION : LIVE DEMO

MicroEJ & Eclipse IoT Java Technologies:
• running on Renesas PEACH  
  (Cortex A9, 10MB RAM, Arduino connector, Ethernet)
• a LWM2M client connected to leshan.microej.com
• a MQTT client connected mqtt.microej.com
• an example app playing with Leds and publishes on a topic
• a heartbeat service that publishes uptime on a topic
THANK YOU
FOR YOUR ATTENTION!

https://projects.eclipse.org/projects/iot.edje
https://developer.microej.com/
HOW DO YOU TEST OUT EDJE ?

• Get a MicroEJ Compatible board
e.g. STM32F746G-DISCO board from ST Microelectronics (~50€)
  (more boards are listed on developer.microej.com)

• Get a free copy of MicroEJ Studio at https://developer.microej.com
  follow the instructions in https://developer.microej.com/getting-started.html

• Clone the developer branch at https://github.com/MicroEJ/edje
  (it will be soon in https://github.com/eclipse/edje)

• Build Edje libraries from MicroEJ Studio
  (right-click « build with EasyAnt » on the two libraries projects you’ll get from github)

• Run the sample applications from MicroEJ Studio
  (right click « Run As MicroEJ Application » and the app will be remotely installed and started
  on the board)
NEED MORE THAN JUST EDJE?


• MicroEJ GitHub: (for examples, libraries, tools) https://github.com/MicroEJ