Java and the Internet of Things

Henrik Stahl
Vice President
Java Product Management
9 Million+ Java developers worldwide

#1 Choice for developers

#1 Development platform

5 of Top 5 OEMs ship Java ME

3 Billion mobile phones run Java

100% of Blu-Ray players ship with Java

97% of enterprise desktops run Java

89% of desktops in USA run Java

5 Billion Java Cards in use

125 Million TV devices run Java
The Path to IoT Services

Reduce Complexity

- Interoperability & Standards
- Provisioning & Management

Drive Innovation

- Developer Productivity
  - Time-to-Market
- Device Lifecycle

Create Opportunity

- Data Capture
- Real-Time Analytics
- Reduced Human Interaction
- New Services
Challenges in the IoT Era

- **Intelligent Devices**: Always-on connected to variety of sensors and running multiple software applications
- **Big Data**: Generates high-frequency Fast Data analysis for instant decision making and automation of information flows
- **Responsiveness**: Enables customer service differentiation from automated, real-time responsiveness

*Fueling New Services*
Early IoT architecture

Black box
Hardcoded functionality

Applications, Middleware, Analytics/BI, Databases, Big data, ...

Enterprise datacenter and/or Cloud
Requirements on intelligent devices

1. Application Platform
   Streamline how IoT applications are developed, secured & deployed

2. Distributed Intelligence
   Make predictive decisions quicker and closer to the source of the data

3. Integration
   Connect intelligent devices to existing enterprise applications

4. Security
   Protect against malware and threats, manage security and identity of data and devices
## IoT Is Changing the Device

<table>
<thead>
<tr>
<th>Highly diverse and rapidly changing use cases &amp; technologies</th>
<th>Hardware capabilities &amp; connectivity evolving rapidly</th>
<th>Value is in software, but embedded software development is difficult</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expensive to reinvent and reintegrate must-have features (management, security, etc.)</td>
<td>Volume &amp; value of data is gaining importance as a business driver</td>
<td>Time-to-market and flexibility are key to success</td>
</tr>
</tbody>
</table>
Device Needs

- Always On
- Performant and Scalable
- Secure
- Remotely Manageable
- A Platform for New Services
- Provide Local Intelligence
Vendor Ecosystem Needs

<table>
<thead>
<tr>
<th>Manage BOM &amp; Profit</th>
<th>Innovation and Competitive Edge</th>
<th>Standards &amp; Regulatory Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to Market</td>
<td>Readily Available Resources</td>
<td>Reuse Across Markets</td>
</tr>
</tbody>
</table>

- €¥$  
- Innovation and Competitive Edge
- Standards & Regulatory Compliance
- Reuse Across Markets
IoT Architecture, take two

**Java enabled**
Open, intelligent, updatable

**Black box**

**Applications, Middleware, Analytics/BI, Databases, Big data, …**

**Enterprise datacenter and/or Cloud**
Business Value of Java in Embedded

- Extended Product Lifecycle
- Enhanced Experience
- Increased Market Reach/multiple UEs
- Proven, Reliable, Secure
- Control over BOM and Roadmap
- Strong Resource Availability
- Shorter Time-to-Market
- Reduced Support Costs
- Reduced Risk

Grow Revenue
New IoT Services
Increase Efficiency
Reduce Cost
Business Value of Java in Embedded

- **Control over BOM and roadmap**: Common Java platform across broad choice of hardware and OS delivering portability

- **Strong resource availability**: Widest, most diverse eco-system fueled by over 9m developers globally

- **Shorter Time-to-Market**: Standard commercial platform reducing QA cycles and reinvention – invest more cycles to innovate

- **Reduced Support costs** via remote management and update

- **Reduced Risk**: Widely deployed, secure, standards based platform – deployed globally – backed by Oracle.
Business Value of Java in Embedded

- **Extended Product Lifecycle**: In-market update to deliver new and manage services powered by industry standards
- **Enhanced Experience**: Increased device level interoperability and integration, fewer silo’s to manage
- **Increased Market Reach/Multiple UEs**: Re-use common modules across multiple domains.
- **Proven, Reliable, Secure**: From the SIMcard to the enterprise data center
Java Embedded Overview

Footprint

- 50KB-1MB: Java Card
- 1MB-10MB: Security
- 10MB-100MB: Small
- LARGE: Java ME
- MEDIUM: Java SE
Java IoT Vision

Any Device...
Any Market...
Any Size...
Today

APIs

Java SE 7

CDC 1.1

CLDC 1.1

Language

Java SE 7

CDC 1.1
(based on SE 1.4.2)

CLDC 1.1
(based on SE 1.3)
Java 8

APIs
Java SE 8
Java ME 8

Language
Java SE 8
Java ME 8
Stripped Implementations
New in Java ME 8 and Java SE 8

- Use cases: IoT devices and App Store deployments
- Users will be permitted to bundle an application with a subset of Java SE or Java ME, removing all unused portions for the smallest possible size
- Licensees must still ship complete and compatible implementations
- Changes to licensing terms and/or TCK Rules will be needed to ensure that end users creating stripped implementations do not fragment the platform or introduce incompatibilities
- The details are still being worked out - we will keep you informed
Requirements on Stripped Implementations

- Must be derived from a complete compatible implementation
- Cannot be changed once created
- Must be "closed" (not expose APIs - cannot load new code)
- Must function identically to the pre-stripped application
Future direction

- Unify language between ME/SE completely (if possible)
- Modularity in SE 9 & OSGi interop
- Investigate Java for smaller form factors (extreme low power sensors)
- Functionality for IoT
  - CoAP, DTLS (ARM)
  - MQTT (IBM)
  - Device I/O API (Oracle)
  - Update to sensor, bluetooth, USB, location, …?
Questions (for another day?)

- **Oracle’s** current path
  - Java ME/SE and supportive functionality in JCP
  - OSGi as “large embedded” app container
  - Follow protocol/interop work in other standards bodies
  - Industry specific standards in industry alliance groups (HGI)

- Do you agree with this structure? Would you prefer another?
- What is your view on Java as an IoT “device platform”? Do you support it? Prefer another solution? If so, which one and why?
- What would you like to contribute?