

Java ME Directions

JCP F2F - Austin

Florian Tournier - Oracle
May 9, 2017

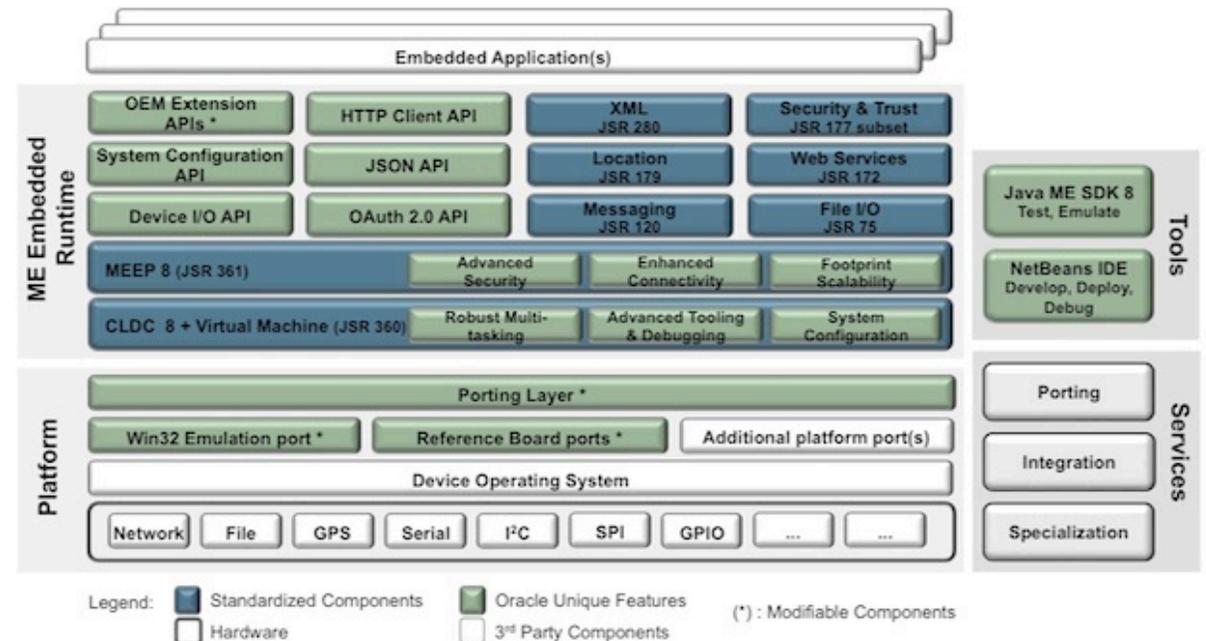


Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Java ME Offering

- Java ME 8
 - Released in 2014 - simultaneously with SE 8
 - JSR 360 CLDC 8 / JSR 361 MEEP
 - TCK / RI available for licensing through Oracle
- Oracle Java ME - Embedded
 - Oracle commercial implementation of Java ME 8
 - Includes proprietary APIs/enhancements
 - Highly optimized, secure, multi-tasking VM
 - On-demand, remote provisioning
 - Cellular connectivity, Device I/O integration
 - Regularly updated : 4 releases since 2014
 - Available on a variety of embedded platforms
- Java ME SDK available on OTN



Java ME 8 Adoption

Adoption Snapshot

- ME 8 products are only coming to market
 - Most products based on Oracle ME-E implementation are still in development phase
 - No request for stand-alone ME 8 JSRs licensing
- Oracle Java ME-E adoption / interest is focused on specialized IoT market segments
 - Wireless modules (inc automotive), Smart meters / energy, development boards, integrators
 - (To a lesser extent) edge gateways, smart sensors.
 - Critical to adoption : large volume potential & strong requirements for upgradability

Why has been slowing down adoption ?

- Potential roadblocks
 - Embedded development / certification / update cycles are very long – especially in low-end
 - No strong case yet for interoperability yet at the low-end of the embedded space
 - Commercial Model vs free/homegrown options
 - CLDC 1.1-based products continue to work
 - Volumes for updatable / programmable devices remain small outside of modules / meters
 - More and more silicon can support SE-E

Perspectives for Java ME 9

Current Course

- Java ME refresh is not part of the Java 9 Release in 2017
- Synchronized releases are desirable, not critical
 - Limited ME/SE developer community overlap
 - Embedded adoption cycles are longer
- No obvious functionality gap that cannot be addressed with existing releases
 - Most Java ME 8 products are still in development and have not yet reached commercial stages

Building a Case for ME 9

- The JCP EC ME Working Group has been discussing the need for a Java ME 9 release
 - Market opportunity & competition for Java in low-end segments
 - Technical requirements for ME/SE consistency and specific to the embedded market
 - Results will be presented at the JCP F2F
- Java ME customers and prospects want clarity on platform evolution
 - Public messaging on Java ME future desired at JavaOne 2017

ME 9 – Potential Technical Scope

- Drive toward greater alignment with Java SE
 - Increased language level compatibility, API semantic parity, etc
 - Decrease the language feature gap.
 - Candidates :
 - Collections
 - Reflection
 - Runtime Annotations
 - Concurrency utilities
 - Collections and Math API
 - JNI Access
- Add support for IoT standards and protocols
 - REST client, MQTT/CoAP support, expanded DIO
- Update old JSRs relevant to Embedded
 - SATSA in particular
 - JSR update could be incremental to ME8 and not necessitate a platform release
- Designed for high volume, constrained devices
 - Compact footprint: as low as 128 KB RAM, 1 MB Flash
 - Low Startup Time requirements

Going Forward Proposal – Platform (Spec and TCK)

Oracle's effort in the embedded space is focused on Java SE

- Java SE 9 will offer greater coverage of the embedded device landscape
 - Finer-grained control over an application's runtime footprint through JSR 376 (Jigsaw)
 - Cheaper/more powerful silicon implies a greater fraction of devices can run SE9

Alternatively : aim for Java 10 timeframe as a synchronized release

If there is critical mass of interest, Oracle would support an ME9 proposal in JCP

- Java ME 9 = update to CLDC & MEEP
- Oracle would support 3rd party spec lead
 - A suitable spec lead would have to be designated in agreement with the JCP EC
 - Oracle would be part of the EG
- Oracle may lead/participate to optional JSR updates in some cases
 - If relevant to SE context, or critical to cross-platform / version adoption
- Licensing model will be agreed with potential spec lead

