



# Mobile Service Architecture 2 (JSR 249) JCP EC Spec Lead Presentation

Presented by Kay Glahn, Vodafone

13 Jan 2009

**NOKIA**



**vodafone**

# Agenda

- Introduction
- The MSA EG
- Fragmentation
- The MSA Platform



# Agenda

- **Introduction**
- The MSA EG
- Fragmentation
- The MSA Platform



# The biggest defragmentation efforts in recent years

- To de-fragment the mobile Java platforms a project was founded: MOBILE SERVICE ARCHITECTURE (MSA)
- MSA 1 standards suite got delivered in Dec 2006
- MSA 2 standards suite is still work in progress
- MSA 1 compliant products have entered the market since 2007
- Vodafone and other operators are referencing MSA 1 in terminal requirements
- De-fragmentation effect on the market still needs to be seen



Java ME already is very little fragmented in comparison to native environments and web platforms on mobile phones



But there is still additional effort necessary to further reduce fragmentation



# Summary of characteristics of MSA

- Initiative of major industry players (operators, manufacturers and others)
- Lead by Nokia (Erkki Rysa) and Vodafone (Kay Glahn)
- TCK and RI implemented by Sun
- How:
  - \_ Selecting JSRs to form the MSA platform
  - \_ Specifying clarifications to reduce ambiguity and fragmentation
  - \_ Specifying additional requirements
  - \_ Providing compliance testing
  - \_ Providing a consistent licensing framework, increasing transparency



# The Purpose of MSA

- Reduce fragmentation in the Java space
- Create a standardized, high quality Java platform for mobile phones
- Provide an ongoing progress which keeps up with latest technologies
- Feeding industry and developer requirements into MSA and reduce proprietary requirements
- Make the Java platform equivalent to the native platform in terms of available functionality
- Make the promise “Write Once Run Anywhere” come true in the Java ME space



- Reducing terminal costs
- Reducing development costs
- Leveraging application development and service usage



# Agenda

- Introduction
- **The MSA EG**
- Fragmentation
- The MSA Platform



# Current MSA 2 EG Members

## Operators

- AT&T (SBC)
- China Mobile Communications Co. Ltd
- NTT DoCoMo, Inc.
- Orange France SA
- Sprint
- T-Mobile
- Telefonica
- TeliaSonera AB
- Vodafone Group Services Limited

## Device Manufacturers

- LG Electronics Inc.
- Motorola
- Nokia Corporation
- Research In Motion, LTD (RIM)
- Samsung Electronics Corporation
- Sony Ericsson

## Others

- Aplix Corporation
- BEA Systems
- Ericsson AB
- Esmertec AG
- Intel Corp.
- ProSyst Software GmbH
- Siemens AG
- Sun Microsystems, Inc.



## How the MSA EG works

- Regular conf calls
- F2F meetings (every 6 to 8 weeks)
- Teamroom as a collaboration platform
- EG mailing list



# How the Community can get Engaged

- Observer list
- Discussion Forum
- Blog
- Feedback in Public Reviews
- Engagement with EG and Spec Leads at public events like JavaOne



- Any feedback on the MSA specification is highly appreciated and can be provided through the following email address:

[jsr-249-comments@jcp.org](mailto:jsr-249-comments@jcp.org)



# Agenda

- Introduction
- The MSA EG
- **Fragmentation**
- The MSA Platform



# Fighting Fragmentation

- Reducing optionalities by additional clarifications
- Adding interoperability requirements
- Reducing the optionalities in API sets by providing three consistent stacks
- Making as many JSRs as possible conditional mandatory



# Fragmentation Challenges in Java ME

- TCK coverage is still limited (not 100 percent)
- Quality testing is missing in TCKs → Implementation bugs still persist on too many devices
- TCKs are black boxes and the tests are not publicly available
- Fragmentation and implementation bugs are two different things
- Technology fragmentation across device portfolio is a problem for operators
- A significant effort is necessary for developers to roll out clients across a large terminal portfolio



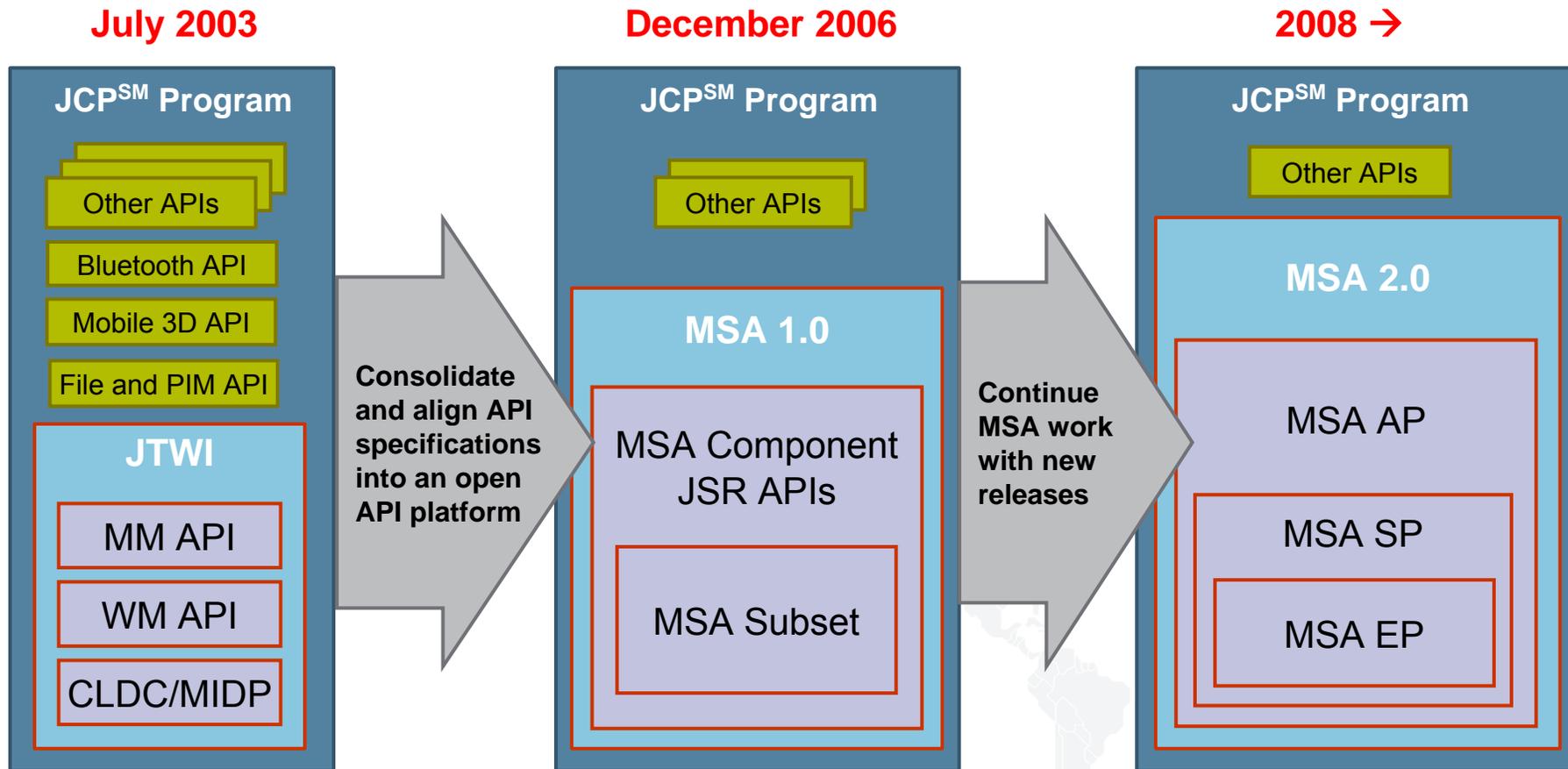
# Agenda

- Introduction
- The MSA EG
- Fragmentation
- **The MSA Platform**



# MSA Initiative

GROUP R&D



JCP = Java Community Process

JTWI = Java Technology for the Wireless Industry



# MSA 1 has been successful

- Devices are available on the market

## \_ Nokia

- = Series 40 5<sup>th</sup> Edition
- = Series 40 5<sup>th</sup> Edition Feature Pack 1
- = S60 3<sup>rd</sup> Edition, Feature Pack 2

## \_ Sony Ericsson

- = Java Platform 8

## \_ Motorola

- Development tools are available (Sun WTK, Eclipse, Netbeans, Emulators)
- Developers start developing MSA compliant applications



# MSA 2 Overview

- Expert Group has been extended
- New name: MSA Advanced → MSA 2
- Changed Focus :
  - \_ Originally MSA Advanced focused on CDC only
  - \_ MSA 2 covers the CDC and CLDC Platform
  - \_ Development goes in line with MIDP 3 where also CDC and CLDC are supported
  - \_ Will be based on MIDP 3, MIDP 2.1 as alternative for low-end devices
  - \_ CDC compliancy provides a migration path towards Java SE
  - \_ Most of the EG members are focusing on CLDC
- Scope:
  - \_ All devices from ultra low-end to high-end are covered
  - \_ Also covers emerging market devices which haven't been addressed in the past



## MSA 2 Key Goals

- Build on the success of MSA 1
- Further defragment the mobile Java platform
- Integrate latest technologies and APIs
- Build a consistent Java platform around MIDP 3 as a basis
- Add interoperability requirements → Adds interoperability testing to specification and TCK (Input from GCF)
- Provide a competitive environment which can keep up with native and web environments in terms of functionality and user experience. → As much JSRs as possible will be conditional mandatory to provide the same functionality to Java as to native applications



# Interoperability Testing

- MSA 2 (JSR 249) will now also cover interoperability testing by specifying the interaction between Java platform and other systems in the phone and thus provide an additional instrument to reduce fragmentation between MSA implementations
- Examples:
  - The behavior of the Java technology system in presence of an incoming phone call or priority message
  - The behavior and appearance of the Java technology security with respect to other trusted or important messages of the rest of the phone
- Interoperability requirements will be tested by the MSA TCK or by the corresponding component JSR TCK



# MSA Migration Path

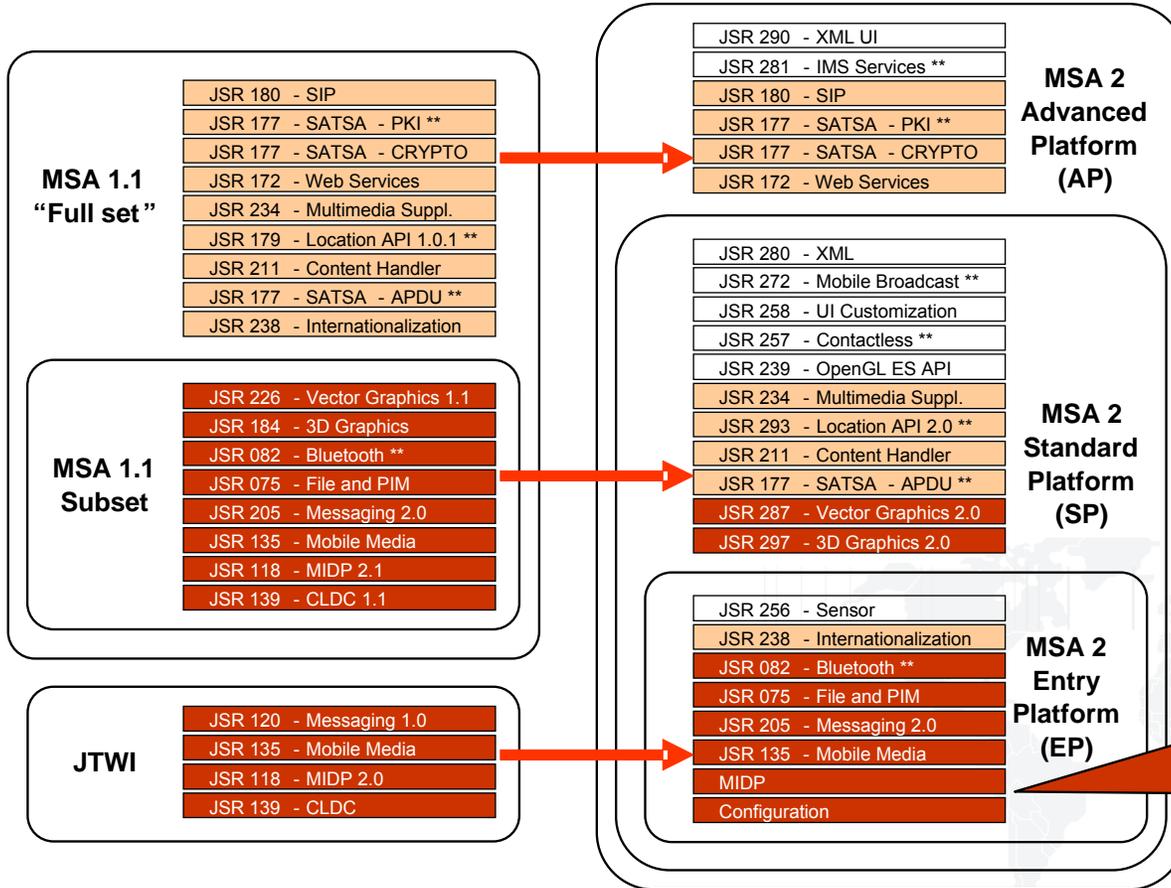
## MSA 1 and JTWI

## MSA 2

High Device Segment

Mid Device Segment

Low Device Segment



New API in MSA 2
API from MSA 1.1 full set
API from MSA 1.1 Subset

\*\* Conditionally Mandatory API

**MIDP**

- EP: JSR 118 - MIDP 2.1
- SP/AP: JSR 271 - MIDP 3.0

**Configuration**

- EP: JSR 139 - CLDC 1.1
- SP/AP: JSR 139 - CLDC 1.1.1 or JSR 218 - CDC 1.1.2



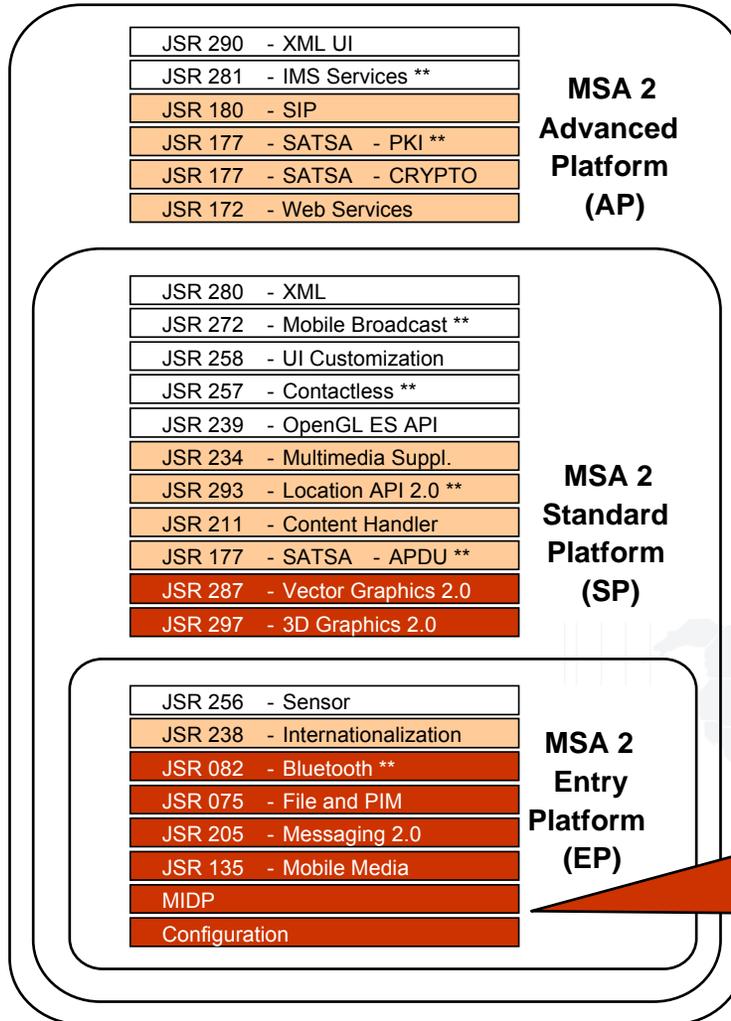
# MSA 2 Architecture

## MSA 2

High Device Segment

Mid Device Segment

Low Device Segment



New API in MSA 2
API from MSA 1.1 full set
API from MSA 1.1 Subset

\*\* Conditionally Mandatory API

- MIDP**
- EP: JSR 118 - MIDP 2.1
  - SP/AP: JSR 271 - MIDP 3.0

- Configuration**
- EP: JSR 139 - CLDC 1.1
  - SP/AP: JSR 139 - CLDC 1.1.1 or JSR 218 - CDC 1.1.2



# Challenges of MSA 2

- Cover a broad spectrum of devices from highest end to lowest end:
  - \_ Different requirements for different device types but the goal is a consistent platform over the whole range
  - \_ MIDP 2 for low-end devices and MIDP 3 for high-end devices
  - \_ New features of MIDP 3 will not be available in MSA 2 low-end devices → Which feature/clarification should go into MSA 2 and which one into MIDP 3?
- Dependencies between different JSRs:
  - \_ MSA 2 → Component JSRs → MIDP 3
  - \_ MIDP 3 has to provide TCK/RI first
  - \_ Component JSRs have to provide a CDC compliant TCK
  - \_ Component JSRs have to pass the TCK on top of MIDP 3 (both CLDC and CDC)
  - \_ MSA TCK/RI can be finalized
- Selecting the right set of APIs in order to accommodate everybody without blowing up the footprint too much → Current API set is still under discussion



# Status and Timeline



- JSR 248 (MSA 1.0)
  - \_ Available since December 21, 2006
- JSR 248 Maintenance Release (MSA 1.1)
  - \_ Available since February 21, 2008
  - \_ JSR 229 has been removed
  - \_ Other minor Changes and bug fixes to the TCK
- JSR 249 (MSA 2.0)
  - \_ Early Draft Review: Q1/2008
  - \_ Public Review: Q4/2008
  - \_ Updates to the Public Review during Q1/2009
  - \_ Proposed Final Draft: Q1/2009
  - \_ Final Approval Ballot: Q2/2009 (Depending on schedule of TCK/RI provided by Sun)



# What's next?



- MSA was established as an ongoing activity and not as single specification:
  - \_ More up to date specification by regular maintenance releases (6 month) is being considered
  - \_ Fragmentation has been significantly reduced but still hasn't been completely eliminated
  - \_ New technologies and APIs have to be adopted and integrated into a consistent platform
- MSA 2 paves the way towards CDC devices
  - \_ MIDP 3 runs on top of CLDC as well as CDC
  - \_ MSA 2 works with CLDC and CDC
  - \_ The configuration becomes less relevant for future Java ME environments
- Will Java SE become relevant for mobile devices?
  - \_ Java SE security model has been adopted by MIDP 3 and MSA 2
  - \_ Generic connection framework is available for Java SE (JSR 197)





**Kay Glahn**

**Consultant Mobile Service Architecture**

Vodafone Group R&D

Vodafone Group Services GmbH

Chiemgaustr. 116

81549 Munich

Germany

T + 49 89 95410 0

F + 49 89 95410 111

[kay.glahn@vodafone.com](mailto:kay.glahn@vodafone.com)

[www.vodafone.com](http://www.vodafone.com)

