Mobile Service Architecture

Meeting the Growing Needs of Mobile Handsets

The Mobile Service Architecture (MSA) platform builds on the Java™ Platform, Micro Edition (Java ME) specifications that have come before it, including the Mobile Information Device Profile (MIDP), Connected Limited Device Configuration (CLDC), and Java Technology for the Wireless Industry (JTWI). As the wireless device market continues to evolve and incorporate new technologies and services — from Bluetooth to vector graphics to Web services — in mass-market handsets, there is demand to create a platform that standardizes on those new technologies. The broad adoption and success of MIDP and CLDC, followed by the success of the JTWI platform, in the mobile device market is now being followed by another wireless industry-defined standard — MSA.

MSA aims to create a predictable environment for application developers who utilize the latest handset technologies. By choosing to meet this specification, manufacturers benefit from a large number of compatible applications that take advantage of their devices' hardware and software functionalities. When creating products for this software and service environment, developers can access a broad range of devices that support their applications.

The MSA specification, Java Specification Request (JSR) 248, raises the bar for functionality for high-volume mobile devices while reducing variations in the platform environment and increasing the already substantial base of mobile applications. It defines the next mobile platform for applications and services.

The MSA specification includes a number of deliverables, such as:

- A roadmap of mobile handset-related JSRs and descriptions of their availability in various markets around the world. Historically, mobile devices in different geographic regions did not always support the same hardware and software. The MSA specification was created by wireless industry experts to specifically address this issue.
- A specification describing the essential client components of a mobile phone application environment plus a recommended combination of Java ME platform technologies. These requirements are designed to enhance end-to-end compatibility.
- A Technology Compatibility Kit (TCK) that verifies compatibility to the clarifications defined in MSA and an integrated implementation for the technologies described in the specification.

Functionality and mandatory JSRs

The goals of the MSA specification are to define a standard set of application functionality for mobile devices while clarifying interactions between various technologies associated with the MIDP and CLDC specifications. Because there is a wide range of variation in mobile handset hardware and software capabilities, the MSA specification offers two choices: to implement the predefined subset of the MSA specification or to implement the full MSA specification. MSA-compatible devices must implement either all of the predefined subset or all of the full MSA specification:

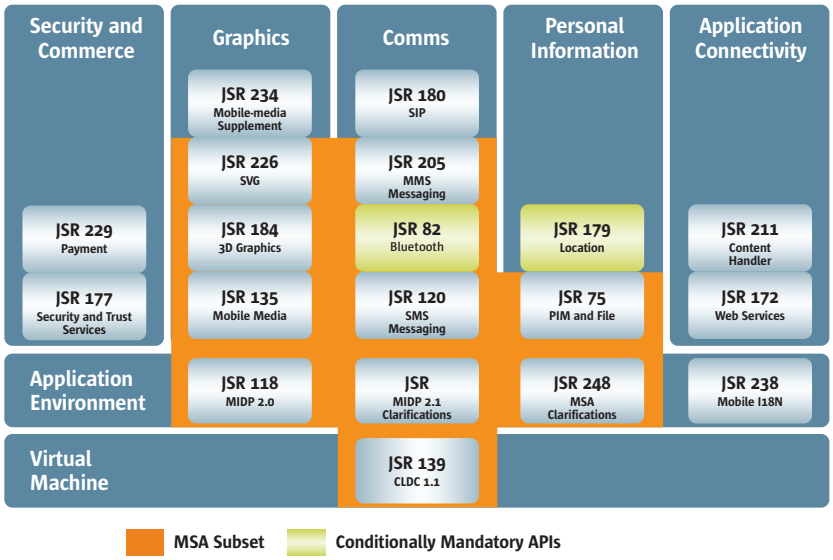
- The subset meets today's base common handset functionality.
- The full specification is targeted at feature-rich, leading-edge mobile devices.

Highlights

- Builds on the success of the Mobile Information Device Profile (MIDP), Connected Limited Device Configuration (CLDC), and Java Technology for the Wireless Industry (JTWI) specifications
- Defined by some of the world's largest mobile device vendors and wireless operators
- Specifies the standard service and application environment for Java technology-enabled mobile handsets

As today's leading-edge technologies become tomorrow's base functionality, the full MSA specification will become the next common base environment for most mobile devices.

The MSA subset includes several JSRs: 139, 118, 75, 135, 82, 184, 205, and 226. The full MSA specification includes all the JSRs in the subset plus additional ones: 172, 180, 211, 229, 234, 238, 177, and 179. While most JSRs are mandatory, a few JSRs are conditionally mandatory because they are dependent on the device providing the hardware capability to support the JSR. In addition, the MSA specification defines additional clarifications on top of the component JSR specifications, often mandating previously optional pieces in the specifications. The purpose is to create a more predictable mobile device environment.



JSR 248: Mobile Service Architecture — Java ME platform umbrella specification

MSA subset

To implement the MSA subset, devices must support CLDC (JSR 139) and MIDP (JSR 118) as well as JSR 75, 135, 184, 205, and 226. If the device supports Bluetooth, it must also support JSR 82. In addition, devices must support some clarifications to those JSRs as outlined in the MSA specification. With this set of specifications, developers can create a wide array of media-capable applications that seamlessly integrate with their devices' Personal Information Manager (PIM) data.

When MSA is first deployed, operators and device manufacturers may choose to implement the MSA subset as a common base across all devices, implementing the full MSA specification only on smart phones or feature phones. As hardware capabilities increase across all devices, the full MSA specification should become more prevalent than the MSA subset. Both the full MSA specification and the MSA subset can also be implemented using Connected Device Configuration (CDC) as long as the implementation complies with the specification.

JSR 139: CLDC 1.1 (mandatory)
The MSA base configuration is CLDC 1.1. This specification is based on a virtual machine designed with the constraints of inexpensive, resource-limited embedded devices in mind. It provides core technology for the Java ME platform runtime and libraries, and is used as the base for one or more profiles, including MIDP. For more information on CLDC, visit java.sun.com/products/cldc.

JSR 118: MIDP 2.1 (mandatory)
MIDP provides the core functionality for mobile applications, including user interface, network connectivity, local data storage, and application lifecycle management. It is targeted at mobile devices such as wireless handsets or PDAs. More information on MIDP may be found at java.sun.com/products/midp.

JSR 75: PDA Optional Packages for the Java 2 Platform, Micro Edition (J2ME™ Platform)
JSR 75 specifies two packages for the Java ME platform, both of which are mandatory. The PIM package enables developers to access PIM data, such as the calendar, address book, and to-do lists, that is resident on most mobile devices. The File Connection package enables developers to access various forms of data, such as images, sounds, videos, and more, on mobile device file systems. This includes removable storage devices, such as memory cards, that may be supported. Access to both PIM and file data enables applications to be tightly integrated with information on the device, allowing for more intelligent applications with easy-to-use interfaces.

JSR 135: Mobile Media API 1.2 (mandatory)
MMAPI allows for access and control, including playback and capture, of basic audio and multimedia resources on mobile devices. This delivers a media-rich, compelling user experience while extending the platform into multimedia. As a very small, lightweight package, MMAPI helps developers gain access to native multimedia services. More information on MMAPI may be found at java.sun.com/products/mmapi.

JSR 82: Java APIs for Bluetooth 1.1 (conditionally mandatory)
JSR 82 enables developers to create applications that utilize Bluetooth technology, a widely deployed standard for wireless communication. Bluetooth may be employed to exchange files, pictures, business cards, or other data between mobile devices. If the device supports Bluetooth, then it must also support JSR 82 in order to be MSA compliant.

JSR 184: Mobile 3D Graphics API for J2ME 1.1 (mandatory)
The Mobile 3D Graphics API specification enables the use of sophisticated 3D graphics in a lightweight, interactive environment with low read-only memory (ROM) and random access memory (RAM) footprints. 3D graphics are used in a wide variety of mobile applications, from games to animated screen savers and messaging. JSR 184 can be implemented on both high-end and low-end platforms across all mobile handsets in the marketplace today.

JSR 205: Wireless Messaging API 2.0 (mandatory)
The Wireless Messaging API (WMA) 2.0 is a superset of WMA 1.0 (JSR 120). It gives Java ME developers the ability to send and receive messages via the Short Message Service (SMS), Multimedia Messaging Service (MMS), and Cell Broadcast Service (CBS) formats. Because messaging is one of the more popular uses for mobile handsets, WMA allows developers to incorporate messaging into other applications on these devices.

JSR 226: Scalable 2D Vector Graphics API for J2ME 1.1 (mandatory)
The Scalable 2D Vector Graphics API specification defines an API for rendering 2D graphics in the World Wide Web Consortium (W3C) Scalable Vector Graphics (SVG) Tiny format. SVG makes it possible for developers to create interactive graphical content, with the ability to zoom and resize on displays with different resolutions and aspect ratios. JSR 226 also defines a subset of the Micro Document Object Model (uDOM) API to allow user interaction and dynamic manipulation of SVG content.

Compared with raster graphics, SVG is much more scalable because it allows for zooming without loss of quality, an important feature in applications such as mapping or for when looking at pictures on a small wireless device screen. SVG also enable users to search for text within a graphic, making application interfaces more interactive. A classic example of this capability is searching for a street name within a map. Applications that take advantage of JSR 226 can store, load, layer, manipulate, and render 2D graphics, creating a more dynamic and rich user experience.

With JSR 226, developers can take advantage of the large amount of available SVG content. Because SVG is based on the eXtensible Markup Language (XML), a developer accustomed to a scripting environment can also take advantage of the robust features offered by a full programmatic environment, such as the Java language.

Full MSA specification

In order to comply with the full MSA specification, implementations must comply with all of the MSA subset requirements as well as requirements and clarifications from the following JSRs. The full MSA specification is targeted at high-end mobile devices. It also provides a roadmap for a common platform to be employed by mass-market mobile devices in the not-so-distant future. By following the MSA specification, operators and handset manufacturers can deploy a robust, predictable base platform that allows developers to incorporate advanced handset hardware and software capabilities into sophisticated, interactive applications.

JSR 172: J2ME Web Services Specification 1.0 (mandatory)
The J2ME Web Services Specification allows developers to take advantage of existing Web services concepts and conventions when creating clients for enterprise services. It contains an XML parser — a subset of the Java API for XML Processing (JAXP) — as well as APIs that enable XML-based, Remote Procedure Call (RPC) communication (JAX-RPC subset). The J2ME Web Services specification follows the conventions and APIs of preexisting Web services specifications used in the Java Platform, Standard Edition (Java SE) and Java Platform, Enterprise Edition (EE). With JSR 172, developers may leverage the preexisting Web concepts and content to quickly develop and deploy Web services.

JSR 180: SIP API for J2ME 1.0.1 (mandatory)
The Session Initiation Protocol (SIP) API for J2ME JSR allows developers to initiate, respond, and handle SIP requests on memory-constrained devices such as mobile handsets. SIP APIs rely on the underlying CLDC Generic Connection Framework and follow the MSA security model. SIP applications allow functionality such as instant messaging, gaming, chatting, and presence. As the underlying mobile IP environment grows more sophisticated, the SIP API will allow MIDlets to take advantage of increased connectivity, allowing for constant real-time data updates.

JSR 211: Content Handler API 1.0 (mandatory)
The Content Handler API (CHAPI) allows Java ME applications to invoke another application to handle specified content type. The appropriate application to handle a content type is based on the application management system (AMS) on the device. Applications may register themselves with the AMS to handle various content types. With CHAPI, applications can be developed that hand off content and data at appropriate places. For example, if a message has a music file attached to it, the messaging application may use CHAPI to invoke a device's music player.

JSR 229: Payment API 1.1.0 (mandatory)
The Payment API allows Java ME applications to request and initiate payment transactions, such as when purchasing ring tones or wallpapers for mobile handsets. It also defines a syntax for describing provisioning data to support different payment schemes. This flexible API supports multiple methods for payment — subscription services charged by operators, single-time payments made to third-party providers, and stored payment accounts.

JSR 234: Advanced Multimedia Supplements 1.0 (mandatory)
Advance Multimedia Supplements is intended to augment MMAPI by providing access to features that have become more common on mobile devices such as cameras, 3D audio files, audio radio, image encoding, and image post processing. JSR 234 allows application developers to access a device's multimedia capabilities, enabling applications that control multimedia such as a radio software jukebox. Other applications may incorporate multimedia, such as a messaging application that allows users to send pictures from their cameras.

JSR 238: Mobile Internationalization API 1.0 (mandatory)
JSR 238 allows internationalization of Java ME applications, making it possible to isolate local resources and employ them during runtime in user- or device-selected locales. This enables a developer to easily create a single application for multiple languages and regions across the globe with the proper location-specific language, time conventions, currencies, and number formats.

JSR 177: Security and Trust Services API for J2ME 1.0 (mandatory, conditionally mandatory, optional)
The Security and Trust Services API (SATSA) provides access to security services for mobile devices, including secure storage to protect sensitive data, cryptographic operations to support payment transactions and data integrity, and a secure execution environment that allows deployment of custom security features. Developers may rely on these services to support a wide variety of applications, from banking and payment to user identification and authentication.

SATSA consists of four packages:

- SATSA-CRYPTO (mandatory) to support cryptographic operations such as message digest, signature verification, and encryption or decryption
- SATSA-APDU (conditionally mandatory) for communication with smart card applications
- SATSA-PKI (conditionally mandatory) to support digital signatures
- SATSA-JCRMI (optional) to support remote invocation of a Java Card™ device

Beyond the security model already supplied in Java ME applications, SATSA enables developers to easily add security services that support features such as user identification, data integrity, and payment transactions. These services facilitate digital rights management and commerce on mobile devices, potentially creating new business models.

JSR 179: Location API for J2ME 1.0.1 (conditionally mandatory)
JSR 179 allows Java ME technology devices to determine their physical locations, enabling location-aware applications on mobile handsets. Examples include mapping, messaging applications with presence, and treasure hunts.

Learn More
Get the inside story on the trends and technologies shaping the future of computing by signing up for the Sun Inner Circle program. You'll receive a monthly newsletter packed with information on the latest innovations, plus access to a wealth of resources. Register today to join the Sun Inner Circle Program at sun.com/joinic.

About Sun

A singular vision, The Network is the Computer™, drives Sun in delivering industry-leading technologies that focus on the whole system — where computers, software, storage, and services combine. With a proven history of sharing, building communities, and innovation, Sun solutions create opportunities, both social and economic, around the world. You can learn more about Sun at sun.com.

For more information

To learn more about the Mobile Service Architecture specification, visit jcp.org/jsr/detail/248.jsp. For more information on the Java ME platform, visit java.sun.com/javame. And to receive additional information on Sun software products, programs, and solutions, visit sun.com/software.

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA **Phone** 1-650-960-1300 or 1-800-555-9SUN **Web** sun.com

SunWIN# 498993 Lit.# SWDS12502-0 03/07