

Wireless Java Applications

Auf Grund der inzwischen sehr guten Abdeckung des Marktes mit Mobiltelefonen und der zunehmenden Verbreitung der so genannten *Personal Digital Assistents* (PDAs) als Hardware-Plattform, gewinnt der Markt der mobilen Anwendungen immer mehr an Bedeutung. Einen zusätzlichen Schub bekommt diese Entwicklung durch die Verfügbarkeit der Java-Plattform in diesem Umfeld. Durch sie ist die Anwendungsentwicklung wesentlich einfacher und unabhängig von speziellen Mobiltelefonen oder PDAs geworden. Die Unterstützung beschränkt sich zudem nicht nur auf Mobiltelefone oder PDAs, sondern umfasst theoretisch alle - in der Regel mobilen Geräte - die unter anderem wegen ihrer Baugröße und der Stromversorgung durch Batterien mit relativ kleiner Leistung eine Beschränkung hinsichtlich Rechenleistung und Speicherkapazität haben.

Ein Grundstein für diese Entwicklung wurde im Zuge des *Java Community Process* (JCP) mit der Spezifikation einer Java-Plattform für kleine, tragbare und vernetzte Endgeräte gelegt, der *Java 2 Micro Edition* (J2ME). Man erhofft sich so eine vereinfachte Anwendungsentwicklung in diesem Umfeld, da eine standardisierte Plattform als Basis für die Entwicklung von mobilen Anwendungen geschaffen wurde. Und auch durch die Übertragung von Konzepten aus der Java-Plattform *Standard Edition* ist eine einfachere Integration mobiler Clients in Anwendungen möglich, die auf Basis der Java-Plattform *Enterprise Edition* entwickelt wurden. Als Beispiel sei hier das durchgängige Konzept der Streams genannt, das auch bei der Übertragung von Daten über Netzwerkverbindungen eine tragende Rolle spielt. Eine weiter wachsende Bedeutung der mobilen Anwendungen in naher Zukunft ist also zu erwarten.

Für *Kölsch & Altmann* ist es daher selbstverständlich sich auch mit den Technologien in diesem Umfeld vertraut zu machen. Im Folgenden wollen wir - nach einem kurzen Überblick über die Plattform *Java 2 Micro Edition* - unsere Erfahrungen in diesem Technologieumfeld darstellen.

Java 2 Micro Edition

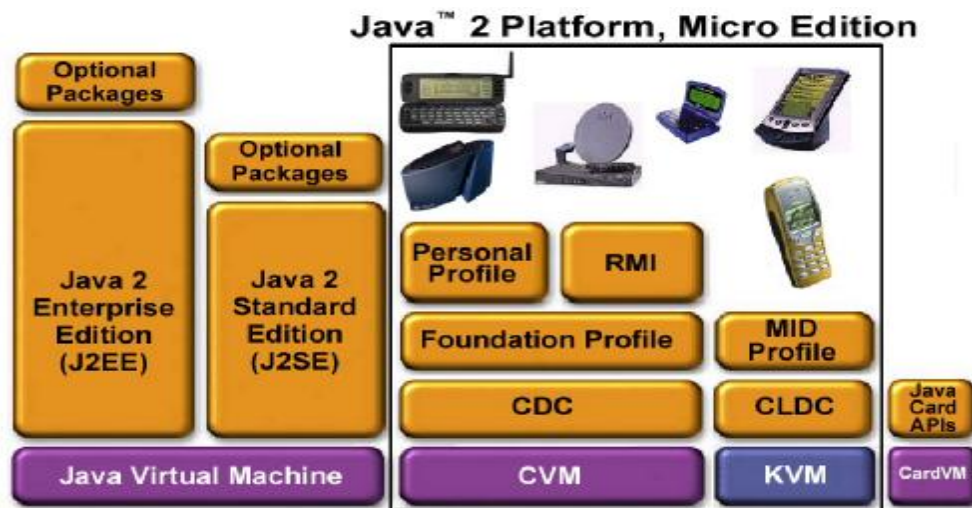


Abbildung 1: Die J2ME-Architektur (Quelle: Sun Microsystems)

In **Abbildung 1** ist die Architektur der J2ME und deren Einordnung innerhalb der gesamten Java-Plattform dargestellt. Um den verschiedenen Ausprägungen von mobilen Geräten gerecht zu werden, für die Anwendungen auf Basis dieser Plattform entwickelt werden sollen, beschreibt die Architektur von J2ME im wesentlichen die zwei folgenden Architekturbauusteine:

- § In einer so genannten *Configuration*, werden eine Java-Virtual-Machine (JVM), die den Geräten mit beschränkter Rechenleistung und Speicherkapazität angepasst wurde, sowie eine Sammlung von Java-APIs in Form der *Core Libraries* spezifiziert.
- § Ein *Profile* spezifiziert eine Erweiterung der APIs aus der *Configuration*, um besondere Eigenschaften einer Geräteklasse zu unterstützen.

Configurations und Profiles

Eine *Configuration* spezifiziert eine vertikale Klassifizierung von Geräten und beinhaltet dazu eine minimale Speicherkapazität, die Rechnerarchitektur (zum Beispiel 32bit Prozessor) oder auch die grundlegenden Eingabe- und Ausgabemöglichkeiten wie zum Beispiel eine vorhandene Netzwerktauglichkeit.

In der Spezifikation eines *Profile* werden auf der Basis einer bestimmten *Configuration* die Besonderheiten von verschiedenen Geräten berücksichtigt. Zum Beispiel die Eingabemöglichkeit über einen Touch-Screen, die minimale Größe des Displays oder die Netzwerkanbindung direkt über TCP/IP oder beispielsweise per Bluetooth. Die Entwicklung einer Anwendung wird also in der Regel auf der Basis eines *Profile* stattfinden.

CLDC und MIDP

Mobiltelefone und PDAs eignen sich auf Grund ihrer Verbreitung auf dem Markt besonders als Hardware-Plattform für mobile Anwendungen. Die Entwicklung von Anwendungen für diese Geräte geschieht in der Regel auf Basis der *Connected Limited Device Configuration* (CLDC) und dem *Mobile Information Device Profile* (MIDP), die speziell für diese Art von Geräten spezifiziert wurden.

In **Abbildung 2** wird die J2ME-Architektur daher noch einmal am Beispiel von CLDC und MIDP veranschaulicht. Auch in den weiteren Abschnitten wollen wir uns auf die MIDP-fähigen Endgeräte beschränken.

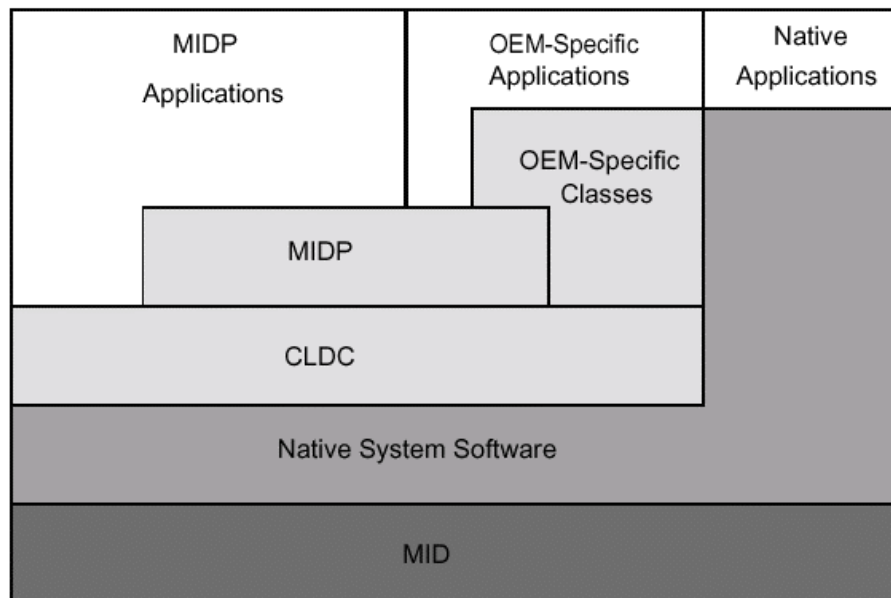


Abbildung 2: Die CLDC und das MIDP der J2ME (Quelle: Sun Microsystems)

Einsatzbereiche

Die größte Auswahl an MIDP-fähigen Geräten findet man derzeit bei den Mobiltelefonen und PDAs, bei deren Hersteller eine deutliche Akzeptanz der Java-Plattform vorzufinden ist.

Bei der Nutzung der mobilen Endgeräte als Anwendungsplattform gibt es grundsätzlich die folgenden beiden Möglichkeiten. Zum einen so genannte Stand-Alone-Anwendungen, und zum anderen Anwendungen in der Rolle eines Clients innerhalb einer verteilten Anwendung. Da diese zweite Möglichkeit sicher die für die Zukunft interessantere Nutzungsmöglichkeit darstellt, wollen wir einige Einsatzszenarien für mobile Client-Anwendungen aufzeigen:

- § Kleinere Geräte (zum Beispiel Mobiltelefone) können, wegen ihrer beschränkten Möglichkeit grosse Datenmengen komfortabel eingeben zu können, als Anzeigegeräte dienen. Die anzuzeigenden Daten werden entweder online oder bereits vollständig beim Start der Anwendung von einem Server empfangen. Es handelt sich hierbei also um eine Alternative zu den WAP-fähigen Mobiltelefonen, allerdings mit zusätzlichen Möglichkeiten, wie beispielsweise Daten lokal vorzuhalten und somit lange Telefonverbindungen zu vermeiden.
- § Bietet das Gerät komfortablere Eingabemöglichkeiten, beispielweise mit Hilfe eines Stiftes auf einem Touch-Screen, kann das mobile Gerät für die Erfassung von Daten vor Ort benutzt werden. Die lokal erfaßten Daten können anschließend an einen Server zur weiteren Bearbeitung oder Auswertung versendet werden.
- § Eine umfangreichere Anwendung auf dem mobilen Endgerät wäre eine zusätzliche Alternative zu den evtl. bereits vorhandenen HTML- und Java-Clients innerhalb einer verteilten Anwendung. Der Benutzer könnte in diesem Szenario also sowohl am Arbeitsplatz mit dem HTML- oder Java-Client, als auch unterwegs mit dem mobilen Endgerät auf den selben Datenbeständen arbeiten. Man kann somit noch einen Schritt weiter gehen als es bei den gängigen Organizer-Anwendungen derzeit der Fall ist. Statt einer notwendigen Synchronisierung der Daten könnte hier tatsächlich auf einen nur einmal vorhandenen Datenbestand auf dem Server zugegriffen werden.

Integration

Für die, aus den drei oben beschriebenen Szenarien hervorgehende, notwendige Integration mobiler Clients in verteilten Anwendungen bietet J2ME bereits das dazu notwendige Rüstzeug. Die Spezifikation der CLDC beinhaltet ein so genanntes *Generic Connection Framework*, bei dem das Konzept der Input- und Output-Streams der J2SE als Schnittstellen für Netzwerkkommunikation benutzt wird. Auf dieser Basis bietet dann zum Beispiel das MIDP die notwendigen Implementierungen dieser Schnittstellen für eine HTTP-Verbindung an. Bis zur Version 1.0 von MIDP war nur die Unterstützung von HTTP Pflicht, seit der Version 2.0 aber ist auch die Unterstützung weiterer Protokolle, bzgl. Datagramm- und Socketverbindungen, vorgeschrieben.

Auf dieser Basis bieten sich nun verschiedene Möglichkeiten der Integration von mobilen Clients an. Im Folgenden sollen beispielhaft einige davon aufgezeigt werden.

Integration in Java-Enterprise-Anwendungen

Für die Integration von mobilen Clients in eine Java-Enterprise-Anwendung, die auf Basis der klassischen Client-Server-Architektur aufgebaut ist, bietet sich auf der Seite der Server-Anwendung die Verwendung eines Servlets an. Mit diesem können die von einem Client empfangenen HTTP-Requests bearbeitet werden. Die Anwendungskomponente für den mobilen Client bildet ein Java-Client in Form einer MIDP-Anwendung. Diese kommuniziert über eine HTTP-Verbindung mit dem Servlet der Server-Anwendung.

Das Servlet erzeugt aber in diesem Fall als Antwort auf den Request keine HTML-Seite. Statt dessen werden vom Client serialisierte Daten per POST Request an den Server versendet, und evtl. mit der entsprechenden Antwort auf diesen Request wiederum serialisierte Daten an den Client verschickt. Zu

beachten ist hier, dass auf der MIDP-Plattform des Clients nicht der Standardmechanismus von Java für die Serialisierung von Objekten zur Verfügung steht. Hier ist eine eigene Lösung gefragt, oder es findet sich im Open-Source-Umfeld eine passende. In der Regel wird diese Lösung aber nicht kompatibel zum Standardmechanismus von Java sein, und so muss der selbe Ansatz auch auf der Seite des Servers beim Servlet verwendet werden.

Für den einfachen Datenaustausch sind die vorhandenen Lösungsansätze aber ausreichend. Die Erweiterung von MIDP 2.0 um die Socket- oder Datagrammverbindungen ist unter Beachtung der Spezifikationen der Java 2 Enterprise Edition (J2EE) nicht nutzbar. Dafür ist in der Spezifikation von MIDP 2.0 die für viele Anwendungen notwendige Unterstützung von HTTPS als Protokoll für die Datenübertragung vorgesehen.

Mobile Web Services

Auch Web Services können auf mobilen Endgeräten genutzt werden. Durch die Verfügbarkeit von APIs auf der MIDP-Plattform für XML-RPC und SOAP ist die Entwicklung von MIDP-Anwendungen, die als Client eines Web Service fungiert, bereits sehr vereinfacht worden. Auch für UDDI ist in naher Zukunft eine Implementierung für die MIDP-Plattform verfügbar.

Peer-to-Peer Anwendungen

Eine weitere Architektur für eine Anwendung ist die Peer-to-Peer (P2P)-Architektur. Hier kann eine Anwendung auf dem Endgerät direkt mit einer Anwendung auf einem anderen Endgerät kommunizieren, ohne dazu einen Server als dritte Instanz adressieren zu müssen. Ein Framework für die Entwicklung von Anwendungen auf Basis der P2P-Architektur wird zum Beispiel durch das Projekt JXTA (eng: juxtapose = nebeneinander) zur Verfügung gestellt. Dabei handelt es sich um ein P2P-Protokoll auf der Basis von XML. Durch eine Implementierung der JXTA-APIs für die MIDP-Plattform ist es möglich, Anwendungen für mobile Endgeräte zu entwickeln, die die Rolle eines Peer eines P2P-Netzwerks einnehmen. Allerdings bestehen hier noch einige Einschränkungen hinsichtlich der Funktionalität, zudem ist derzeit noch ein Desktop als Relay für die Einbindung von mobilen Endgeräten in ein JXTA-Netzwerk notwendig. In **Abbildung 3** ist ein JXTA-Netzwerk mit der Anbindung von mobilen Peers dargestellt.

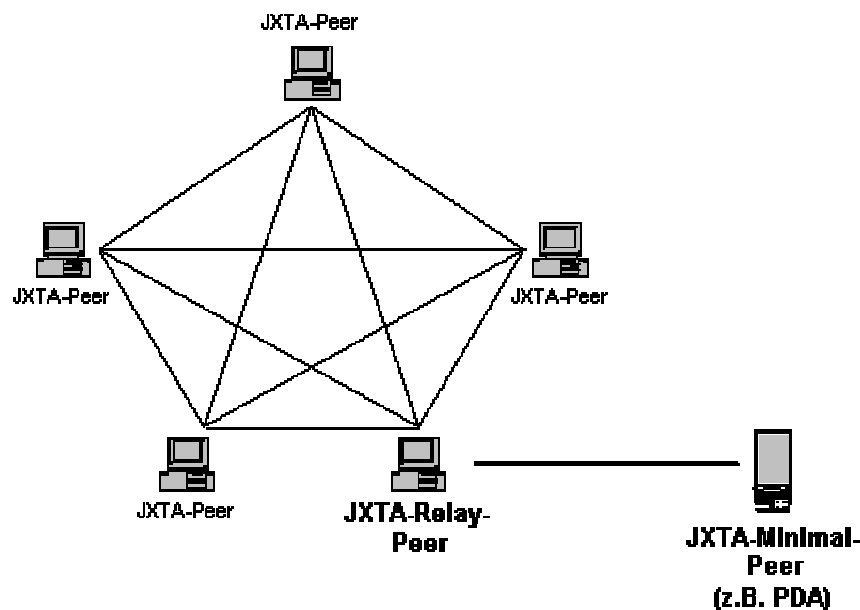


Abbildung 3: JXTA-Netzwerk mit mobilen Peers

Fazit

Mit J2ME (und mit den vielen verfügbaren Erweiterungen) steht bereits eine Plattform für die Anwendungsentwicklung zur Verfügung, mit der sich Anwendungen für eine große Palette von mobilen Endgeräte entwickeln lassen. Dies kann auch relativ effizient geschehen, da sich evtl. bereits vorhandenes Know-How im Java-Umfeld und die dort verwendeten Entwicklungsumgebungen gut nutzen lassen. Grenzen sind noch der Komplexität der auf dem Endgerät installierten Anwendung durch die derzeit vorhandenen Speicherkapazitäten der Endgeräte gesetzt.

Und auch bei der Gestaltung der Benutzerschnittstelle sind zum Beispiel die noch sehr kleinen Displays der Mobiltelefone zu berücksichtigen. Dagegen wären bei aktuellen PDAs bereits umfangreichere, und damit einfacher zu bedienende Benutzeroberflächen möglich. Trotz der breiten Unterstützung von Endgeräten durch die J2ME-Plattform kann es für eine Anwendung evtl. vertretbar sein die Endgeräte beispielsweise auf PDAs zu begrenzen, um dafür deren bessere Möglichkeiten bei der Gestaltung von Benutzeroberflächen zu nutzen. Beispielsweise wäre es bei einer internen Zeiterfassung vertretbar, die Benutzeroberfläche für PDAs zu optimieren, wenn alle Mitarbeiter der Firma über einen PDA verfügen.