



Java ME – Wo bleiben die Applikationen?

JUGS

Dienstag, 12. Dezember 2006



glue

Software Engineering

Roland Loser

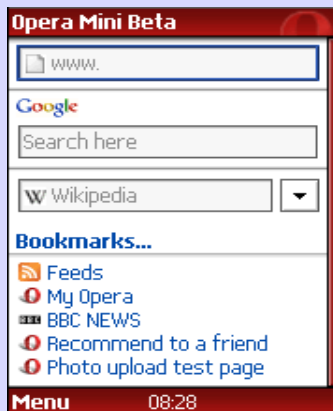
Leiter Mobile Applications

roland.loser@glue.ch



Was kennen Sie für Java ME Anwendungen ?

Opera Mini



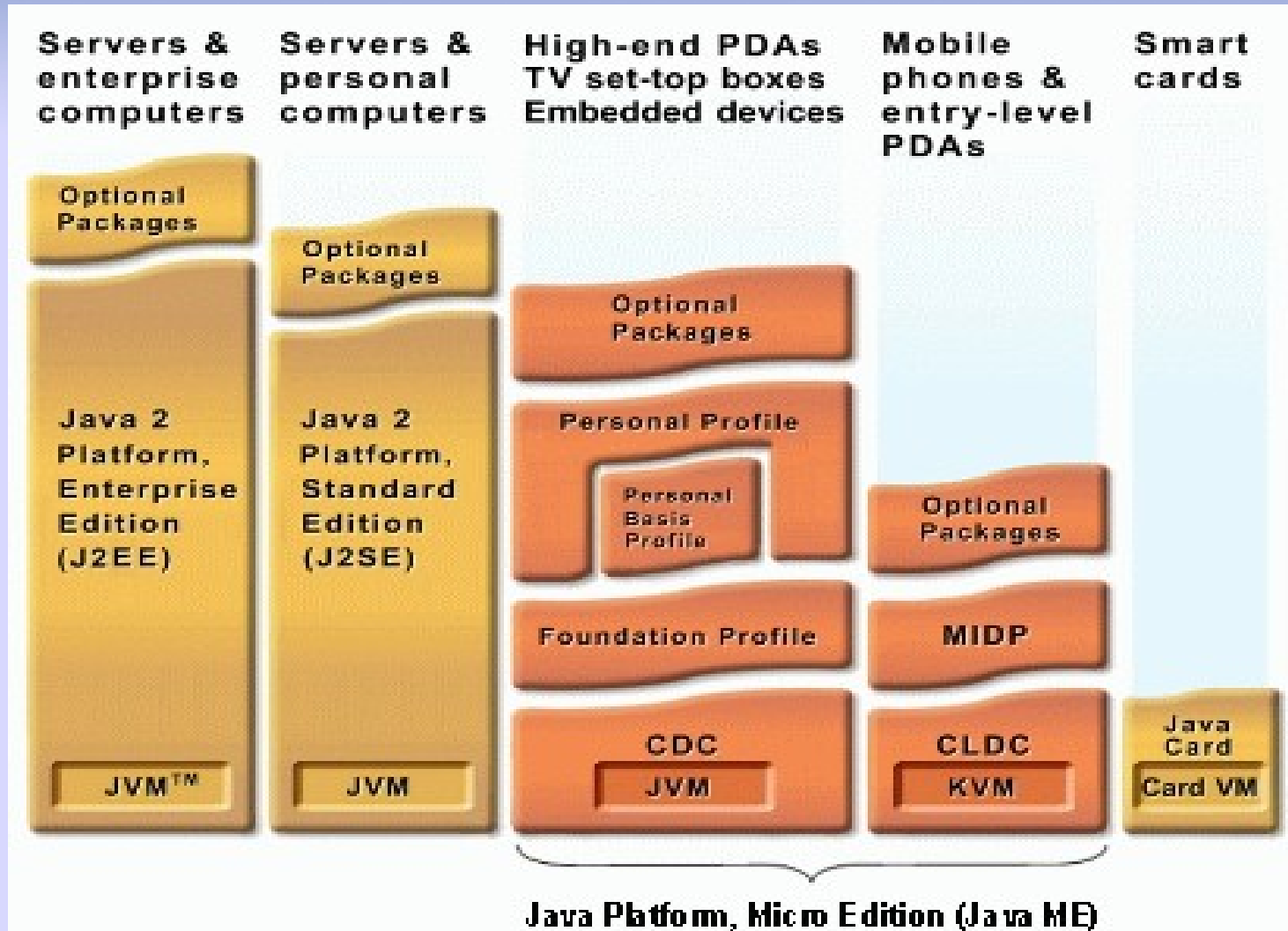
GMail



mig33



Java ME - Refresher





Java ME - Das Versprechen

- Überall verfügbar
- Offene Plattform
- Einfache Programmierung
- Sicherheit
- Massenmarkt



Java ME – Die Realität

- Heterogenität !
- Mobiloperatoren ?
- Usability ?
- Konnektivität ?
- Akzeptanz ?



Heterogenität

- Hardware: Modelle / Hersteller
- Handy OS: Native, Symbian, Windows Mobile, Linux
- Java ME: VM, MIDP 2.0/1.0, Optionale API's



Apropos Heterogenität...

2614 Downloads eines MIDlets

- 440 verschiedene User Agents
- 162 verschiedene Handymodelle
- 13 verschiedene Hersteller



Mobiloperatoren

- Anderer Fokus: Portale, HandyTV
- Kaum Java ME Know-How
- Kein Event Billing basierend auf GPRS
- Angst vor Usability/Support
- GPRS Gebühren!



Usability ?

- Installation
- Ablage
- Sicherheitsabfragen
- Verbindungseinstellungen



Konnektivität

- Separate GPRS Einstellungen
- Eigenheiten GPRS APN / WAP Gateways
- Fehlerhafte Implementationen
- Zunehmende Komplexität mit WLAN



Akzeptanz

- Technologie unbekannt beim User
- Bedenken: Viren / Kosten
- Aha Effekt



Und jetzt?



Applikationen sind cool !

- Rich & Responsive UI
- Zugriff auf Gerätefunktionalität
- Web2.0 ;-)
- On Device portal



Rich & Responsive UI



- Responsive da meistens lokal
- Komplexe Daten können auf die Bildschirmgröße angepasst, dargestellt werden
- Wenig Data Traffic, da Rendering lokal.



Zugriff auf Gerätefunktionalität

- PIM und Files
- Kamera / Multimedia
- Recordstore
- Bluetooth
- SIM Karte ...



Web 2.0 (Beta)

- Javascript (Web) ~ Java ME (Handset)
- Asynchrone Kommunikation
- User Generated Content (z.B. Foto upload)
- Mobile Browser sind Stand Web 1.0



On Device Portal

- Präsenz auf dem Endgerät des User
- Offline Komponenten



WAP/XHTML?

- Entwicklungsaufwand kleiner
- Ansprechende UI's
- Kein Deployment, leichter änderbar
- Immer online. Latenz
- Beschränkter Zugriff auf Device
- Authentifizierung?



Hybrider Ansatz (Mobrix)

- Framework mit Widgets, Engine
- MIDlet -> Engine + Beschreibung
- Beschreibung kann dynamisch nachgeladen werden.
- Erlaubt schnelle Entwicklung
- Eigenes Authentifizierungsprotokoll



Was wir gemacht haben



- WM Tippspiel
 - Tipps/Resultate für alle WM Spiele
 - Update aller Daten < 500 Byte



Was wir gemacht haben



- Miss Bern
 - Komplett offline
 - Bilder / Bio
 - Integration SMS Voting



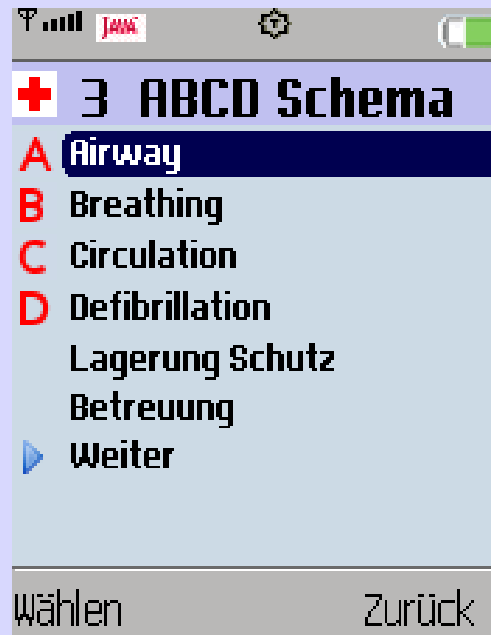
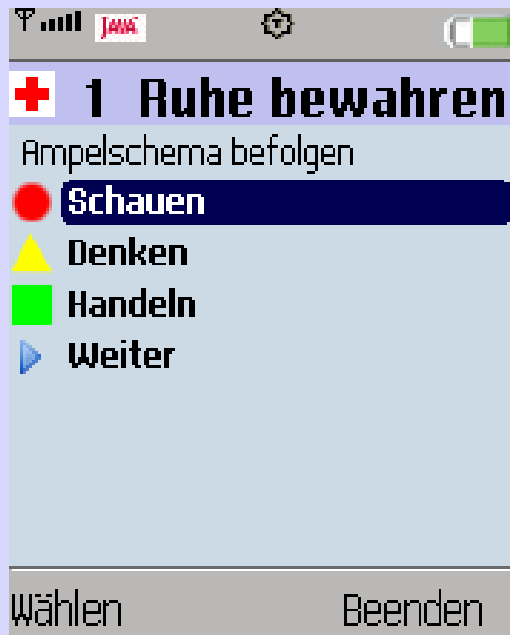
Was wir gemacht haben



- Orbit-iEX
 - 350 Aussteller
 - eTicket
 - SBB Fahrplan
 - Messeplaner



Was wir gemacht haben



- 123help (SRK)
 - Offline
 - Erste Hilfe Anleitung



Was wir gemacht haben



- Swiss Indoors
 - Spieler Bio's
 - Live Resultate via Socket-Connection
 - Inkrementelle Updates



Was wir gemacht haben



- telDirButler
 - Online
Telefonbuch
Abfragen
 - LCDUI basiert



At the end of the day

- Mobile Dienste/Anwendungen sind komplex!
- Technisch komplex
- Anspruchsvoll in der Kommunikation mit Kunden, Operator, Enduser
- Keine falschen Erwartungen



Tipps (1)

- Komplexität reduzieren. Device know how
 - wurfl.sourceforge.net
 - www.jbenchmark.com
 - www.j2mepolish.org



Tipps (2)

- Connectivity
 - XML, Bilder -> Achtung Datenkosten
 - Vorteile einer Anwendung ausnützen.
Datenvolumen kann massiv gesenkt werden.
 - Permanente Socketconnections
(Geheimtipp Port 25...)



Tipps (3)

- UI
 - LCDUI integriert besser in Handset Look&Feel. Mehr spezifisches Device know how nötig.
 - Canvas UI komplex zu implementieren. Besser kontrollierbar, portabler.



Tipps (4)

- Von anderen lernen
 - Wie kommunizieren andere ihre Anwendungen
 - <http://mini.opera.com>
 - <http://www.shozu.com>



Java ME – Die Applikationen kommen

- GPRS Flat-Rate in Sicht...
- Installierte Handset Basis wird immer besser
- Big Players wie Google setzen auf Java ME
- Neue Hardware Features (GPS, RFID ...)
fordern Java ME



Download

JAVA an **9234** (20 Rp./SMS)



Fragen?