eQATor
testing Java™ phones all day and all night long

Anders Fornander & Nicolas Wettstein

JUGS
January 27, 2005

I – Introduction
II – Demonstration
III – Architecture
IV – Project
V – Q & A

Introduction –
eQATor Summarized

eQATor is a framework for automated software regression testing and quality monitoring.

eQATor has been built for testing of esmertec's J2ME™ CLDC products but is a generic framework.

eQATor has been implemented through heavy use of open source tools

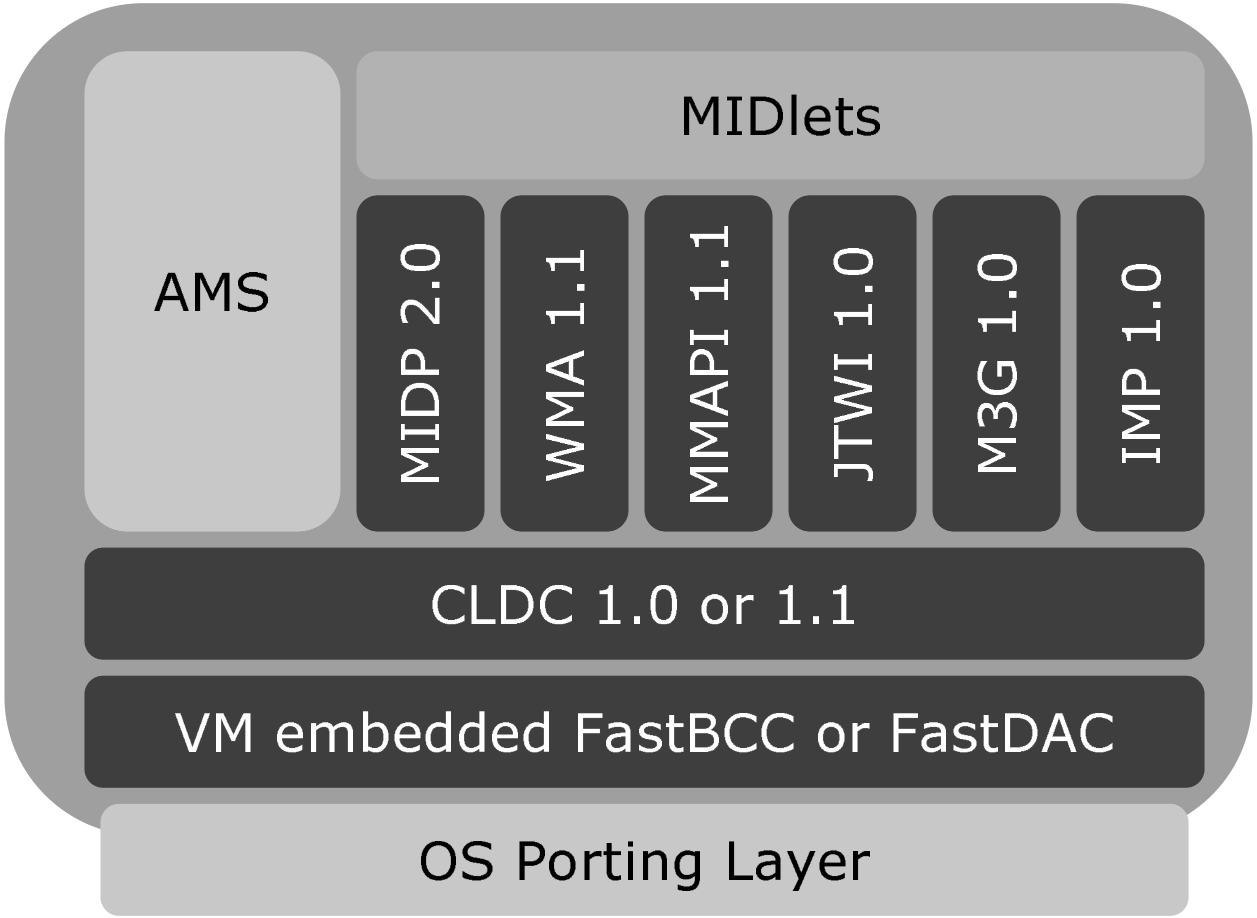
Introduction – Motivation for eQATor

Customers require it =>
no mobile-phone customer will accept a not validated Java™ on their telephone.

High frequency of Jbed™ CLDC releases =>
development code must be close to production quality all the time.

Reduction of production costs =>
automation is the only valid alternative due to many possible configurations and required regression test-cycle times (manual testing would be too expensive).

Introduction – Jbed CLDC



Jbed™ CLDC is esmertec's Java™ 2 Platform Micro Edition (J2ME™) CLDC technolgy implementation

Introduction – Technology Compatibility Kit (TCK)

A Java™ Specification Request (JSR) requires a Technology Compatibility Kit (TCK). A TCK is a suite of test cases that determines whether or not a product complies with a particular Java™ technology specification.

JSR / Technolgy:	Test Cases:	Number Automated:
JSR-30 / CLDC 1.0	4 560	4 560
JSR-139 / CLDC 1.1	11 471	11 471
JSR-118 / MIDP 2.0	782	657
JSR-205 / WMA 1.1	49	42
JSR-135 / MMAPI 1.1	122	122
JSR-185 / JTWI 1.0	180	175
JSR-184 / M3G 1.0	160	144
JSR-195 / IMP 1.0	104	104
	Total: 17 428	(99%) 17 275

Introduction – Benchmarks

In addition to the different TCK test suites a set of benchmark tests are performed. All the benchmarks are commercial, i.e. developed by 3rd party vendors and organizations. Different parts of the Jbed™ CLDC implementation are measured.

Benchmark:	Description:
CaffeineK	Measures the performance of basic KVM features
AMark	Measures the implementation in terms of its applicability for mobile phone games
Benchpress	Measures the graphical performance
EEMBC PNG decode	Measures PNG image decoding speed
EEMBC parallel	Measures performance of KVM threading capabilities
EEMBC kXML	Measures XML parsing and DOM tree manipulation
Mbooks	Measures basic math operations and method calls

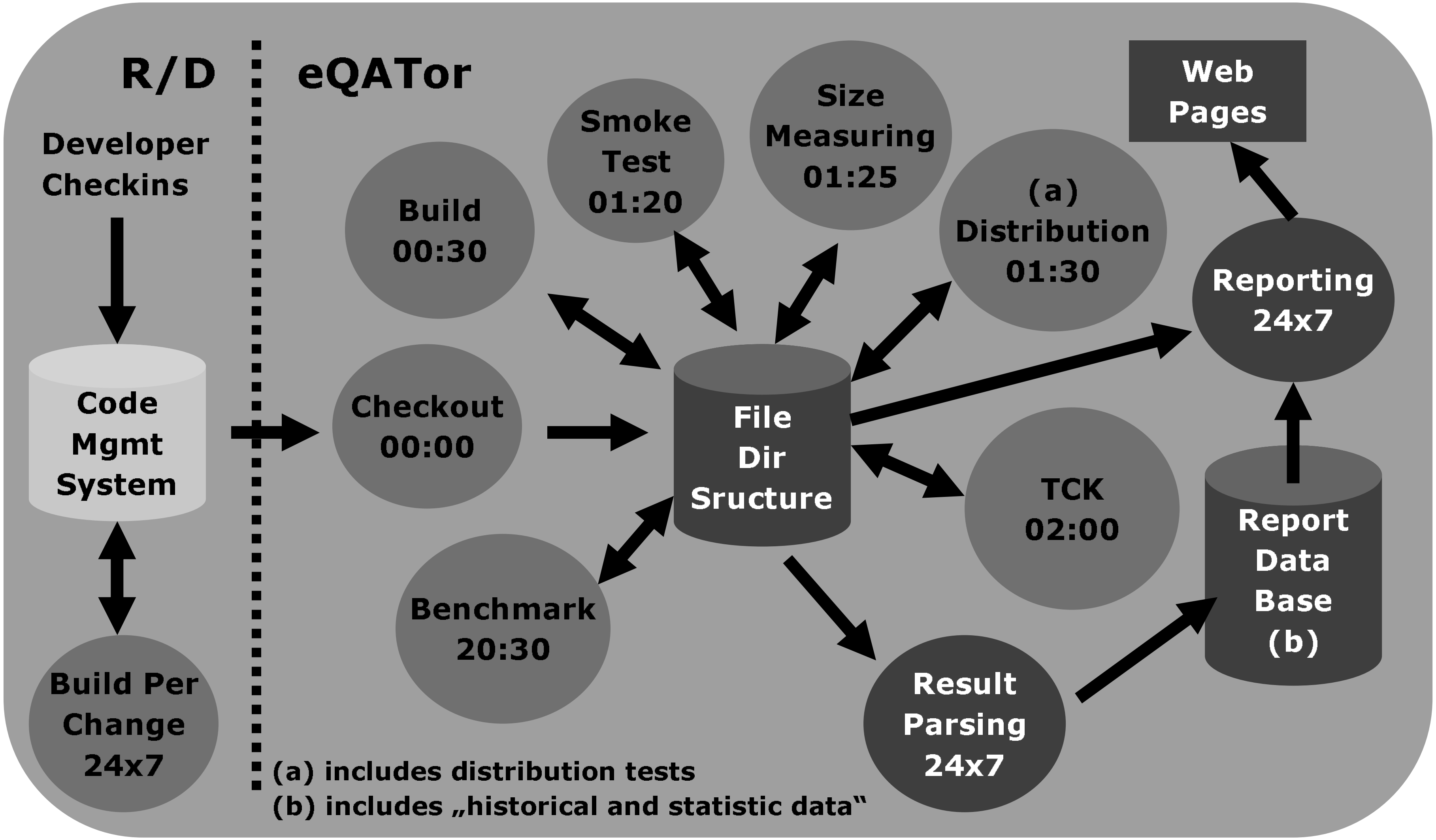
Introduction – Additional Special Tests

Smoke Testing – A smoke test is a “hello world” midlet that is down-loaded, (compiled), installed, executed, i.e. says hello over a network socket. This test checks that all functionality are working needed for running TCK.

Distribution Testing – A distribution test is the validation that a generated Jbed™ CLDC installer.exe contains the correct artifacts (sources, classes, docs, etc.) for that specific release.

Size Measurements – Two types of size measurements are performed: executable segment size measurement (data, bss, text) and Java™ package size measurement (byte code size).

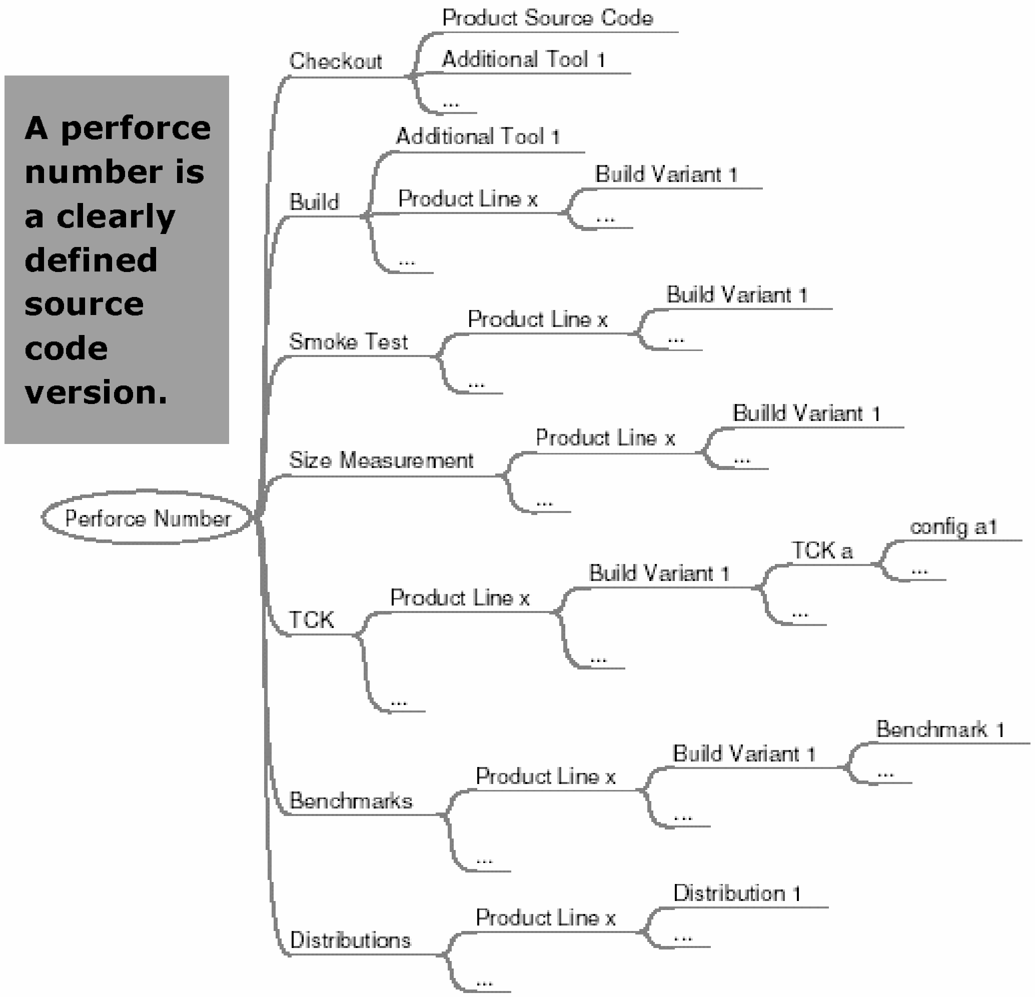
Introduction – Test Processes



(a) includes distribution tests
(b) includes „historical and statistic data"

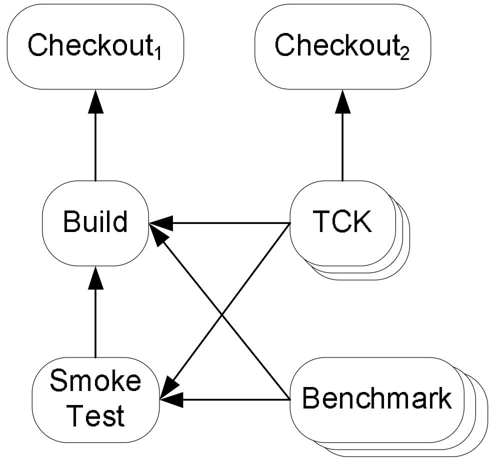
Introduction – Test Data Structure

A perforce number is a clearly defined source code version.



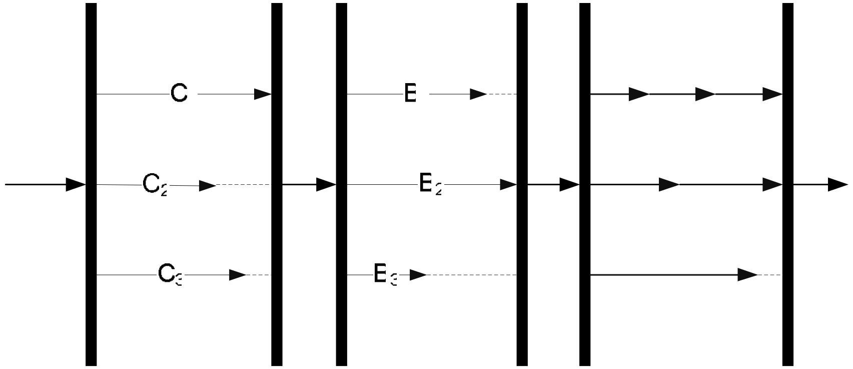
Demonstration

Architecture – Scheduling/Distribution



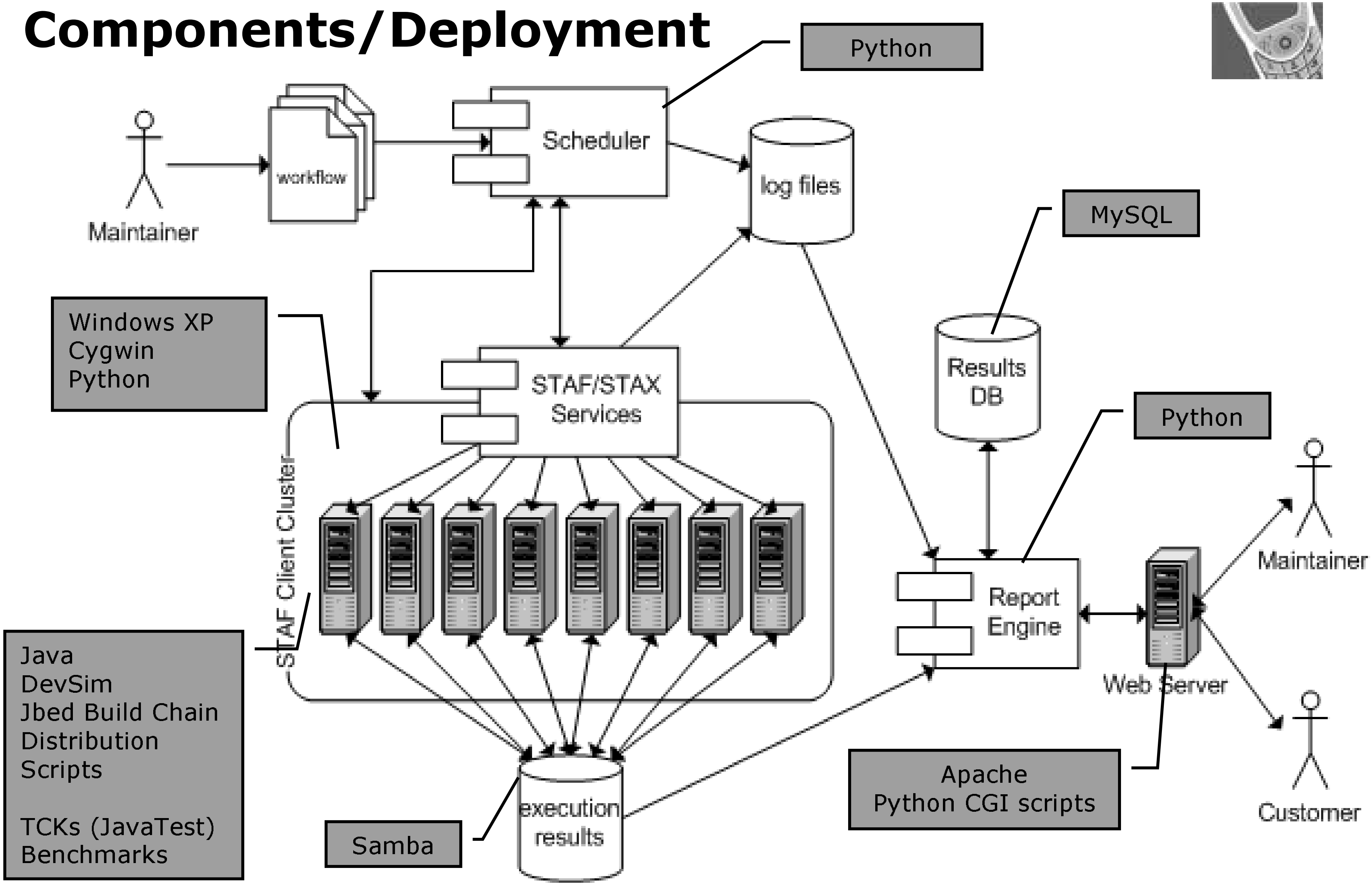
The scheduler is responsible for the distribution and execution of tasks on the client cluster. It Must takes into account:

- Task interdependencies
- Priorities, deadlines
- Easy scalability



The scheduler should try to Optimize the *client load,* i.e. minimize each client's *idle time*.

Architecture – Components/Deployment



Python

MySQL

Windows XP
Cygwin
Python

Python

Java
DevSim
Jbed Build Chain
Distribution
Scripts

TCKs (JavaTest)
Benchmarks

Samba

Apache
Python CGI scripts

Project – Team and Milestones

Project team:
- 3 engineers (full-time)
- 1 student (3 months internship)
- 1 project manager (part-time)

Project milestones:
- start March 2004
- v1.0 in production July 2004
- v2.0 in production October 2004 (improved web interface, more detailed information)

Size of current production environment:
- 12 PC clients (eQATor clients)
- 4 Linux Servers (DB, Web, Scheduler, Monitor)

Questions & Answers

List of URLs

esmertec

Jbed CLDC - www.esmertec.com/solutions/mobile_multimedia

J2ME - java.sun.com/j2me

JSRs - www.jcp.org/en/jsr/all

TCK - www.jcp.org/en/resources/tdk

STAF - staf.sourceforge.net/index.php

Python - www.python.org

Apache - www.apache.org

MySQL - www.mysql.com

J2ME Technologies –
Connected Limited Device Configuration (CLDC)

The Connected Limited Device Configuration (CLDC) defines the base set of application programming interfaces and a virtual machine for resource-constrained devices like mobile phones, pagers, and mainstream personal digital assistants. When coupled with a profile such as the Mobile Information Device Profile (MIDP), it provides a solid Java™ platform for developing applications to run on devices with limited memory, processing power, and graphical capabilities.

J2ME Technologies –
Mobile Information Device Profile (MIDP)

The The Mobile Information Device Profile (MIDP) is a key element of the Java™ 2 Platform, Mobile Edition (J2ME). When combined with the Connected Limited Device Configuration (CLDC), MIDP provides a standard Java™ runtime environment for today's most popular mobile information devices, such as cell phones and mainstream personal digital assistants (PDAs). MIDP provides: user interface capabilities (optimized for the small display size, varied input methods, and other native features of modern mobile devices), extensive connectivity (HTTP, HTTPS, datagrams, sockets, server sockets, and serial port), multimedia and game functionality, and secure over-the-air-provisioning capability.

J2ME Technologies –
Wireless Messaging API (WMA)

The Wireless Messaging API (WMA) is an optional package for the Java™ 2 Platform, Mobile Edition (J2ME) that provides platform-independent access to wireless communication resources like Short Message Service (SMS). Currently WMA can be used on top of CLDC and MIDP.

J2ME Technologies – Mobile Media API (MMAPI)

The Mobile Media API (MMAPI) extends the functionality of the J2ME platform by providing audio, video and other time-based multimedia support to resource-constrained devices. As a simple and lightweight optional package, it allows Java developers to gain access to native multimedia services available on a given device.

J2ME Technologies –
Java Technology for Wireless Industry (JTWI)

The Java™ Technology for the Wireless Industry (JTWI) specification defines the industry-standard platform for the next generation of Java™ technology-enabled mobile phones. JTWI specifies the technologies that must be included in all JTWI-compliant devices, to minimize API fragmentation and broaden the substantial base of applications already developed for mobile phones.

J2ME Technologies –
Mobile 3D Graphics (M3G)

The Mobile 3D Graphics specifies a lightweight, interactive 3D graphics API, which sits alongside J2ME and MIDP as an optional package. The API is targeted at CLDC class devices that typically have very little processing power and memory, and no hardware support for 3D graphics or floating point math. However, the API scales also up to higher-end devices featuring a color display, a DSP, a floating point unit, or even specialized 3D graphics hardware.

J2ME Technologies –
Information Module Profile (IMP)

When combined with the CLDC, the Information Module Profile (IMP) provides a Java™ 2 Platform, Mobile Edition (J2ME) application environment for networked embedded devices that do not have rich graphical display capabilities, or whose resources are limited in other ways: emergency call boxes, parking meters, wireless modules in home alarm systems, industrial metering devices, and the like.