ME4SE

A Pure Java Emulation of the Mobile Information Device Profile (MIDP)

Stefan Haustein

(now) TPP / ICM MP PD SW 2 2 KLF 2

Overview

- Motivation:
  Why a “pure Java” Emulation?
- Architecture / Features
  - ME4SE, SCM
  - XME4SE, Skins
- Possible Applications At Siemens
- Future Plans

Motivation (1/2)

1. SW Development Support

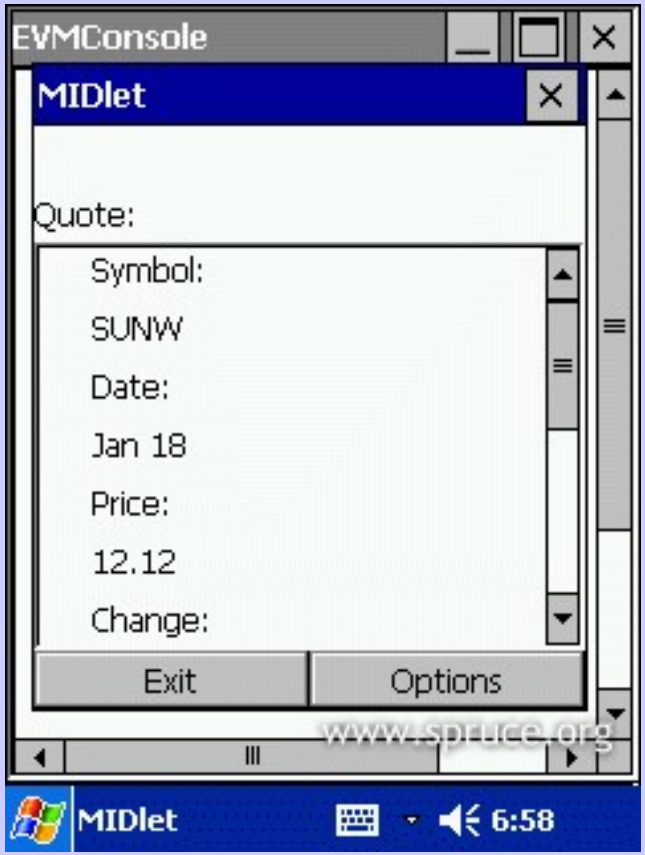
- **Platform Independence:**
  Initially, the WTK (containing an MIDP emulation) was not available for Linux and Mac OS

- **Simpler IDE Integration:**
  A pure Java emulation allows to use the built-in application launch and debugging options of any IDE without additional effort

- **Short Turnaround Times:**
  JAR creation and preverification steps can be skipped

Motivation (2/2)

2. **“Try before Buy”-Option**
   - A pure Java emulation allows to run MIDlets as Applets
3. **Ability to Run MIDlets on Personal Java Devices**
   - Also possible with a pure Java emulation



EMULATOR PROVIDED BY ME4SE

Best J2ME emulator is at http://www.me4se.org. If you see a grey rectangle and that nothing happens, it means that you must click here to install Java

HOW TO PLAY

Click with your mouse on the emulator, and after that press Space Bar to start playing.

GOAL OF THE GAME

The goal of this game is to break all the bricks of a level to advance to the next one. For this, you control a paddle that is used to hit the ball to destroy to the bricks. The challenge of the game is to not loose the ball.

KEY CONTROLS

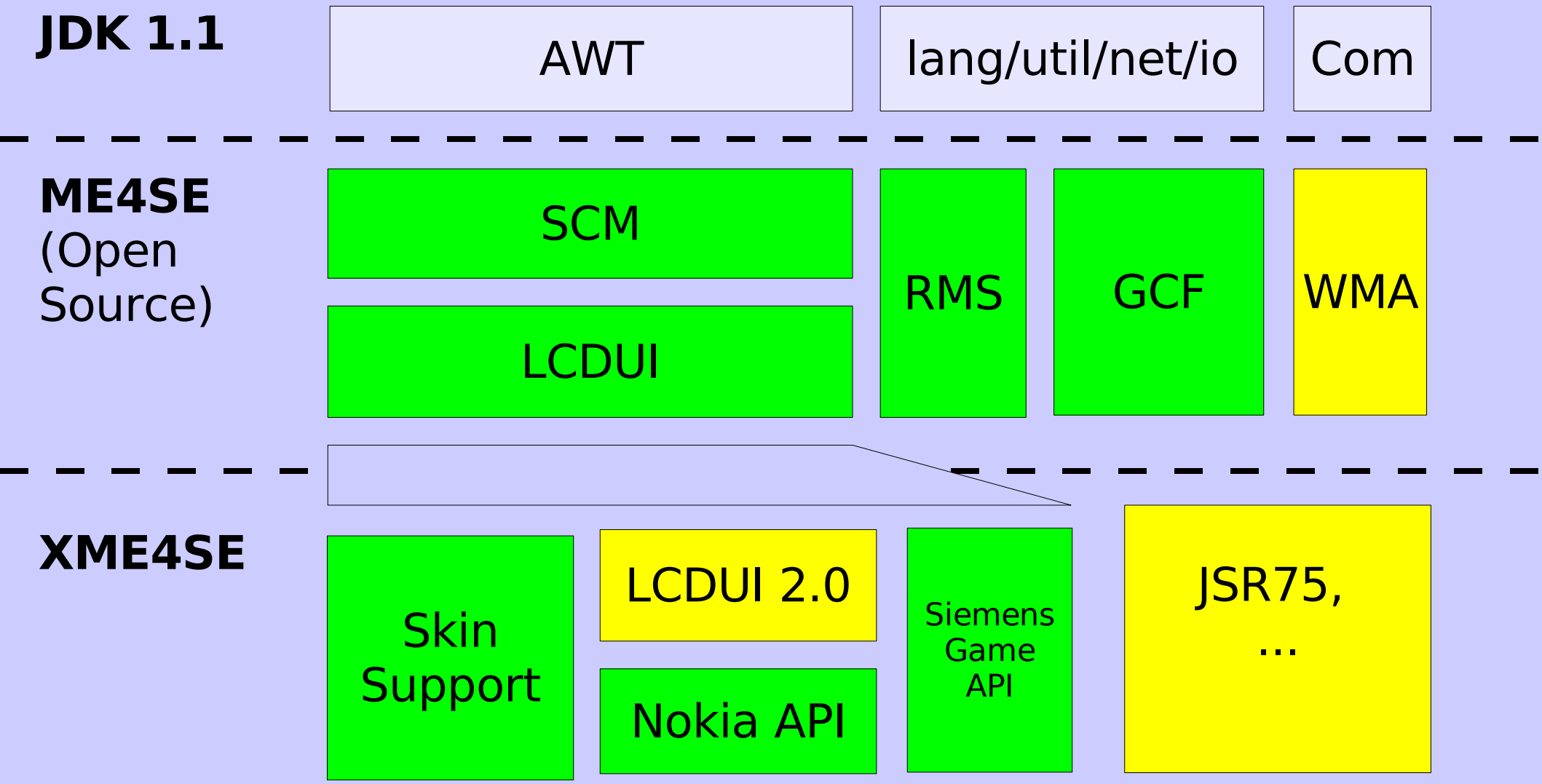
- ← move the paddle to the left
- → move the paddle to the right
- ↑ move the paddle up if you got the rocket engine spell
- ↓ move the paddle down if you got the rocket engine spell
- SPACE BAR release the ball or fire weapons if you have any weapons

www.javaground.be

ME4SE Architecture & Features

- Divided into an Open Source part and proprietary extensions
- Basic functionality covered in open part
  - MIDP 1.0
- Proprietary extensions (“XME4SE”) provide advanced functionality
  - Device skins
  - Additional MIDP 2.0 classes

Architekture



Why a “Simple Component Model”

- First Implementation was based directly on AWT components
- Problems
  - Platform dependent focus behavior; difficult to control
  - No overlapping components that are automatically restricted to parent frame (a pop up that is larger than the device screen looks strange)
  - Limitations: No icons in AWT lists, heavyweight Components too expensive
  - Widgets cannot be skinned
- SWING not an option (N/A for Applets, PJava)

SCM Architecture

- Really Simple
  - Event handling similar to AWT 1.0 (NOT 1.1: no listeners, simple recursive method calls with consumption indicator)
  - Interface similar to LCDUI CustomItem, but with fully controllable positions
  - Few classes, small size
- Allows Nesting and Overlapping Components
  - Components at the same level may overlap, but not become larger than the parent
  - Z-Order determined by index

But “leightweight == slow”!!

- No!
- Non-believers: Please try kAWT on a CX65...

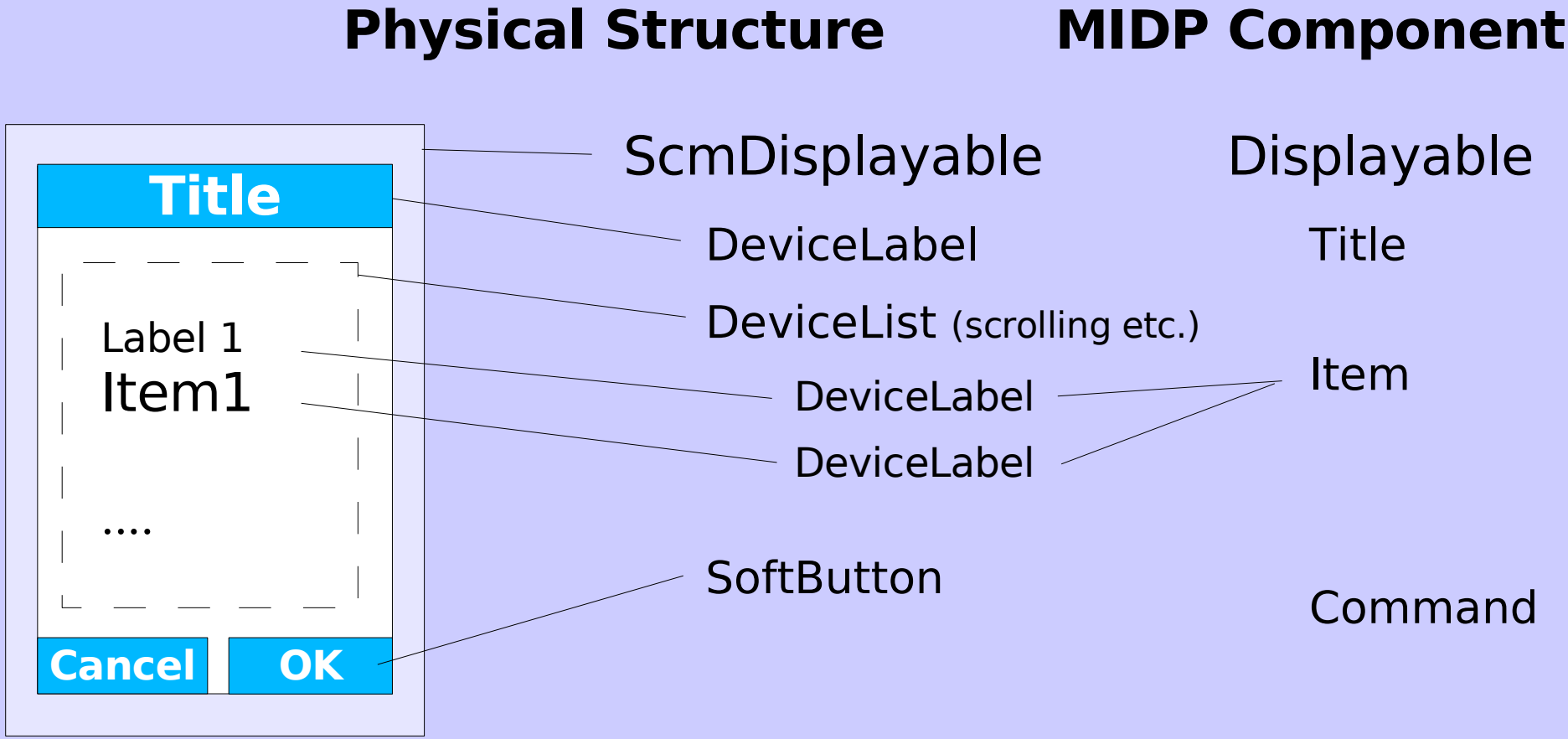
SCM / AWT Connection

- Class ScmWrapper
  - Single connection to AWT
  - Is an AWT component (*Canvas*)
  - Contains an ScmContainer
  - Registers listeners for all relevant events
  - Delivers events to corresponding SCM methods
    - *keyPressed*(), *keyReleased*(), pointer events...
  - Provides a backscreen buffer to avoid flickering
- The only “hard” AWT Dependecies are *Graphics* and *Image*
  - SWT connection
  - or native connection...?

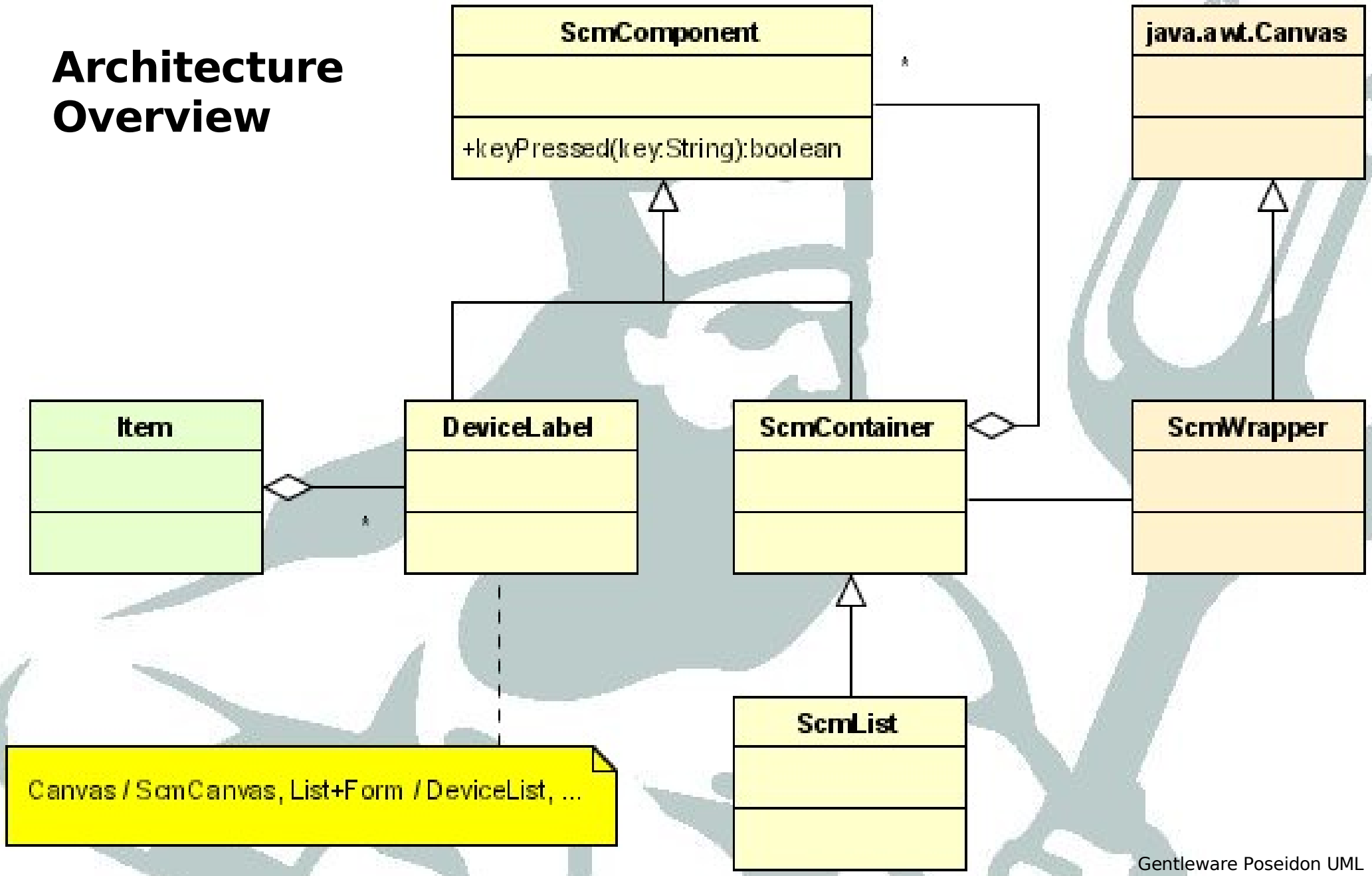
SCM / LCDUI Connection

LCDUI objects are mapped to one or more SCM objects

- e.g. StringItem has a label and a text string....



Architecture Overview



Gentleware Poseidon UML

Customization / Skins

- ME4SE is able to read WTK property files
  - Reduces documentation needs
    http://me4se.org/doc/properties.html
  - Helps to distinguish bugs in the skin file and in ME4SE
- Additional features
  - Support for scroll bars instead of up/down indicators (support for Siemens devices)
  - Support for the Siemens “implicit” full Canvas
  - Items may be completely be replaced by customized versions
  - Supports “physical” key codes
    (e.g. SoftButtons 1=-1; new feature)
  - Item specific colors, fonts etc.

MIDP 2.0 Support

- Not finished; parts added as needed / requested
  - Mostly to be able to support proprietary APIs on a “solid” base
  - Full support for Image operations available
- MIDP 2 Game package is currently missing
  - But very similar to existing Siemens Game API emulation
- API Status Overview generated via JavaDoc Tags
  - http://me4se.org/doc/status/index.html

Nokia Game API emulation mostly based on MIDP 2.0

- Advantage:
  - Emulation Layer could also be used on real MIDP 2.0 devices
  - Port Nokia specific games to Siemens w/o significant effort

AVAILABLE GAMES

- BreakM
- Space Mission
- StripM Poker Girls
- BreakM Candy
- StripM Poker Boys

download our catalog

COMING SOON

- Rock Paper Scissors
- Split-M
- Assembly Mind

EMULATOR PROVIDED BY ME4SE

Best J2ME emulator is at http://www.me4se.org. If you see a grey rectangle and that nothing happens, it means that you must click here to install Java

HOW TO PLAY

Click with your mouse on the emulator, and after that press Space Bar to start playing.

GOAL OF THE GAME

The goal of this game is to break all the bricks of a level to advance to the next one. For this, you control a paddle that is used to hit the ball to destroy to the bricks. The challenge of the game is to not loose the ball.

KEY CONTROLS

- ← move the paddle to the left
- → move the paddle to the right
- ↑ move the paddle up if you got the rocket engine spell
- ↓ move the paddle down if you got the rocket engine spell
- SPACE BAR release the ball or fire weapons if you have any weapons

EMULATOR PROVIDED BY ME4SE

Best J2ME emulator is at http://www.me4se.org. If you see a grey rectangle and that nothing happens, it means that you must click here to install Java

HOW TO PLAY

Click with your mouse on the emulator, and after that press Space Bar to start playing.

GOAL OF THE GAME

The goal of this game is to break all the bricks of a level to advance to the next one. For this, you control a paddle that is used to hit the ball to destroy to the bricks. The challenge of the game is to not loose the ball.

KEY CONTROLS

- ← move the paddle to the left
- → move the paddle to the right
- ↑ move the paddle up if you got the rocket engine spell
- ↓ move the paddle down if you got the rocket engine spell
- SPACE BAR release the ball or fire weapons if you have any weapons

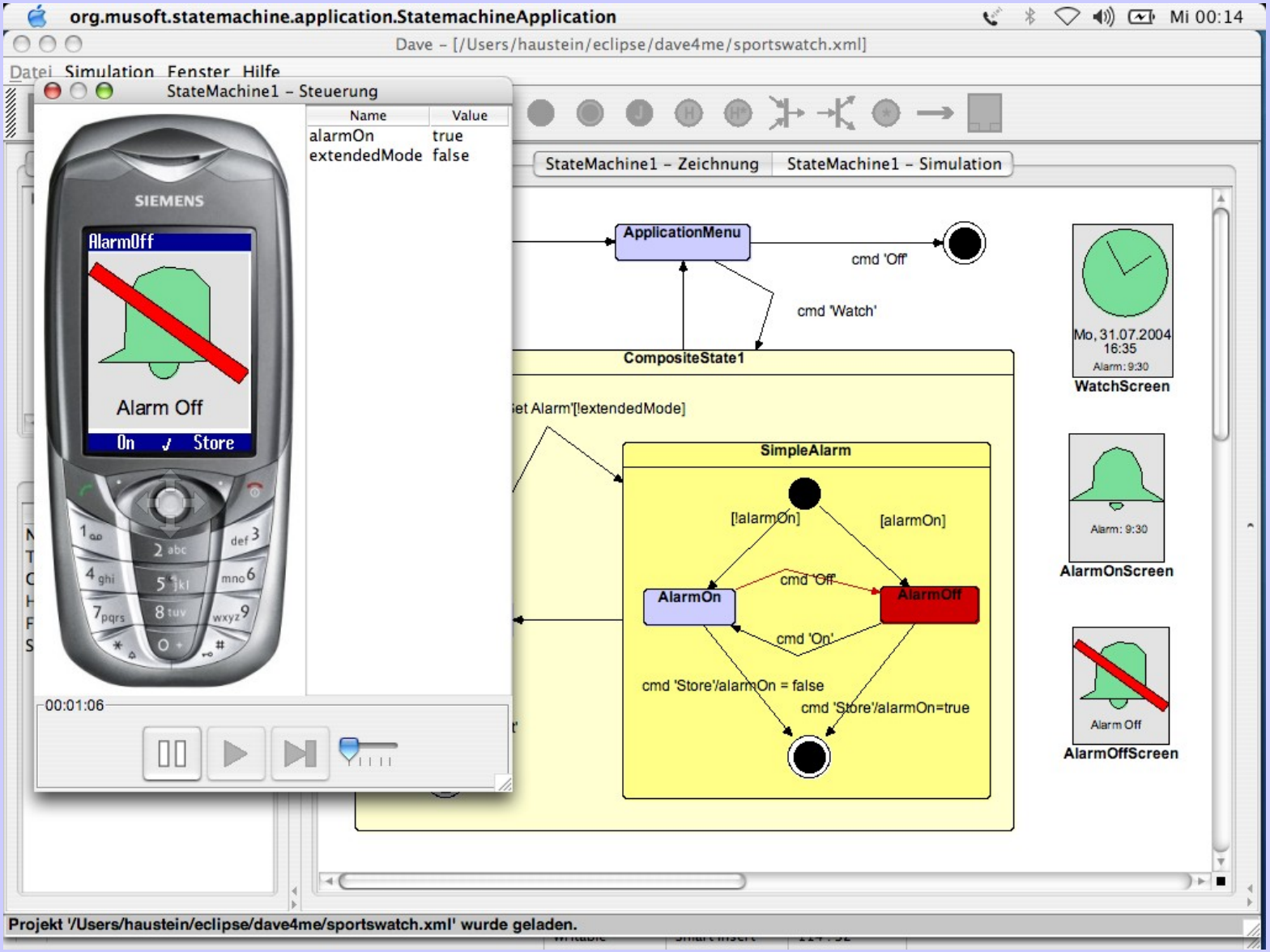
Non-GUI parts

- RMS based on Files (RAM for Applets)
- Additional JSRs partially available:
  - JSR 75 Files (full)
  - JSR 75 Adressbook (based on VCard Files)
  - WMA 1.0 (Need to connect a C35 or similar to the serial port to send an SMS)

Possible Applications at Siemens

- Development support for the next generation of devices
  - Functionality easy to extend (pure Java)
  - In many cases only simple wrappers required (for functionality already provided by J2SE)
  - Simple customization (#SoftButtons, Screen Resolution, Fonts...)
  - Has already been used for R65 development
- Support for porting Games
- Rapid UI Prototyping
  - It is easy to “control” ME4SE from other Java Applications
  - Observation: MIDP GUI design can be modeled with statecharts --> Demo: Dave4Me

Demo: DAVE (www.musoft.org) + ME4SE



Open Issues / Plans

- Licensing
  - What are the implications if I work on ME4SE at Siemens? To be clarified...
- Eclipse Integration
  - Learned today that this should be easy with 3.0
- MIDP 2 Game API Support
  - No rocket science, just need to find a few “free” hours to get it done...
- Currently missing: new / hidden Siemens Extensions
  - Center Key Icon support
  - Headline decorations...

Recent Contributions

Thanks to

Michael Kroll (fh do)

- GCF / RMS parts, WMA

Jörg Pleumann (uni do)

- JAM

Xaviar Kral (www.javaground.be)

- Nokia Skin, Bugfixes

Omry Yadan (telmap.com)

- Lots of Bugfixes

Thank you!

Please visit

http://me4se.org