Android goes reading
History
From first contact to Go Live

First contact 12/2011
Start pitch 06/2012
Contract 12/2012
Go Live 03/2013

Bernhard Hochstätter, Dominik Helleberg, Michael Krümming
Mobile Solutions

What we do…

- We offer solutions for application and software development
- We support all the way – from the first idea until go-live and even afterwards
- We assemble the best team according to your specific needs

We are hiring!

Our Products

App Development
„Software development of mobile applications (Apps) on all current platforms (iOS, Android, WP7) and platform independent with HTML5.“

App Deployment
„Deployment of mobile applications for the platforms iOS, Android, WindowsPhone and BlackBerry.“

Bernhard Hochstätter, Dominik Helleberg, Michael Krümmling
Android goes reading
Reader Application Development
Agenda

1. Android with changed conditions
2. Android with new challenges
3. Clean drawing
4. Android without animations
5. Android without Settings App
6. New challenges for testing
Android with changed conditions

- PagePlace Reader App as foundation
  - Supports all Android Smartphones and Tablets but no eInk Device

- Reader Application supports only one device → no device diversity
  - One resolution
  - One display size
  - One aOS version (Android 2.3.3)
  - No device specialities

- Unlimited rights and possibilities
  - Using any permission
  - Less restricted Android SDK
  - System Application
Android with new challenges

- challenges due to hardware
  - eInk display with only 16 grey shades, 4 fps and ghosting susceptibility
  - Specific performance characteristics
  - Limited key availability
  - Optimise battery lifetime up to 7 weeks

- challenges due to system environment
  - Being the only application on the device
  - Being Homescreen
  - Being Settings App
  - Handle all important system events
Clean drawing

- Drawing problems with eInk:
  - Switching between screens
  - Ghosting of dialogs
  - Page turning
  - Time delays between view and display

- System was enhanced with new functionality for screen refresh (Java and native)

- “The triangle of clean drawing”

- Final solution for the switch between the screens:
  - Custom view that handles drawing in every screen
  - Reduce ghosting through multistep process and screen interaction
Android without animations

- Custom view for lists and grids
  - No scrolling
  - Navigation through paging

- Web Shop
  - No scrolling
  - Enhanced interfaces (e.g. control the display from Website)

- Replace 2D and 3D animations from application
Android without Settings App

- Settings App not visible on the device
  - Reader has to offer important settings

- Handle WiFi
  - Adoption and customization of native implementation from Android 2.3.3

- Setting system time
- Setting brightness of frontlight
- Setting timeout delays
- Factory reset
- Managing battery
- Managing system updates
- Handling internal and external storage
New challenges for testing

- New test requirements coming up besides application- and integration tests

- Complex investigations and benchmarks were made
  - Current measurements on the device
  - Performance benchmarks and comparisons to competitors' products
  - Current measurements during over 10,000 page turns in the reader

- We had to find new testing procedures
Android goes reading
System Development
System development
How to work inside the Droid

- 7 GB Source
- Compile Time ~ 45 minutes
- „Deployment Time“ ~ 5-10 minutes
- Dependencies usually fail at runtime
- You can change everything... Maybe you shouldn’t ;)}
Downsizing/Boot-time tuning
How a droid comes to life....

bootloader

service manager

kernel

zygote

init

daemons

System Server

Package Manager

Activity Manager

Power Manager

launcher

...
Downsizing/Boot-time tuning
How a droid comes to life....
Downsizing/Boot-time tuning
How a droid comes to life....
Power Management
Let the droid sleep...
Power Management
Let the droid sleep...

```java
-deepSleepTime = SystemClock.elapsedRealtime() - (Σ (all cpu_states));

-deepSleepTime = SystemClock.elapsedRealtime() - SystemClock.uptimeMillis();
```
## Power Management

Let the droid sleep...

```
$ adb shell dumpsys batteryinfo

....
Statistics since last charge:
System starts: 0, currently on battery: false
Time on battery: 3h 13m 20s 916ms (0,2%) realtime, 1h 23m 39s 681ms (0,1%) uptime
Total run time: 81d 20h 13m 10s 836ms realtime, 33d 20h 15m 30s 182ms uptime,
Screen on: 21m 34s 334ms (11,2%), Input events: 0, Active phone call: 0ms (0,0%)
Screen brightnesses: dark 2m 46s 112ms (12,8%), dim 9m 53s 192ms (45,8%),
                   medium 8m 32s 673ms (39,6%), light 22s 356ms (1,7%) 
Kernel Wake lock "PowerManagerService": 1h 4m 50s 972ms (1180 times) realtime
Kernel Wake lock "main": 36m 28s 908ms (14 times) realtime
Kernel Wake lock "mipi_link": 33m 4s 670ms (2137 times) realtime
Kernel Wake lock "radio-interface": 16m 30s 724ms (808 times) realtime
Kernel Wake lock "gps-lock": 11m 1s 604ms (5 times) realtime
Kernel Wake lock "wlan_rx_wake": 9m 3s 70ms (186 times) realtime
Kernel Wake lock "alarm_rtc": 4m 30s 287ms (448 times) realtime
Kernel Wake lock "BTLowPower": 3m 13s 600ms (221 times) realtime
Kernel Wake lock "musb_autosuspend_wake_lock": 31s 998ms (594 times) realtime
Kernel Wake lock "wlan_wake": 31s 786ms (17901 times) realtime
...
```
Thanks for Attention!