Openmoko Framework

DBus Services for the Masses

Charlie, Daniel, Jan, John, Mickey, Stefan

September'08 @ Openmoko HQ

Outline

1. Motivation
2. DBus Framework
3. freesmartphone.org
4. How to use
5. Hacking

Motivation

What do we want?

- Write software that people can tinker with
- Unleash the FOSS community's potential
- Leverage creativity and innovation

A look into the past

- Openmoko 2007
    - libgsmd
    - neod
    - eds
- Openmoko 2008
    - QtopiaComm
    - QtopiaPIM
    - QtopiaDevice

Problems

Service Worlds are greedy

- Mutually Exclusive
- Fighting for hardware access
- Modularization?

Library Interfaces are Limiting

- Security
- Concurrency
- Bindings

What developers (that's us!) want

Uniform way to access...

- Services
- Peripheral Devices

...without being locked in to...

- Languages
- Libraries

...so that they can solve their problems

- Simple Access
- Stability
- Multidevice
- Multivendor (not just Openmoko!)

How to get there?

- DBus as service abstraction line
- Competition on top
- Collaboration below

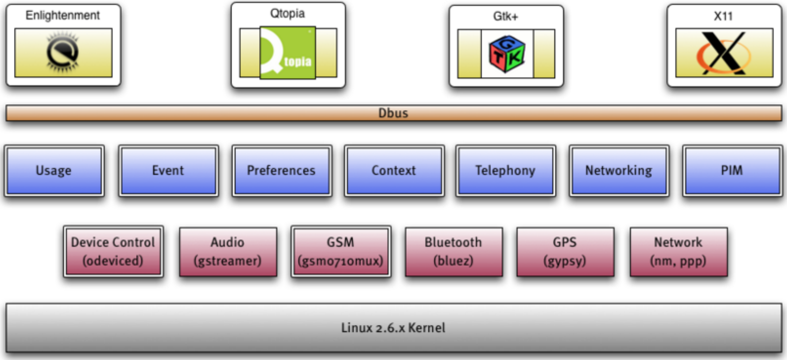
DBus 101

- DBus is Inter-Process-Communication (IPC)
  - Message Bus(es)
  - Inherently Asynchronous
- Object Model
  - busname
  - objectname
  - interfacename
  - method/signal
- Invented for desktop interoperability
  - Tons of Language Bindings
  - Simple Mainloop Integration
- Suitable for embedded devices
- Widely used (demo!)

freesmartphone.org?

- Free software standards collaboration portal
- Created 2006 after struggling for years against fragmentation
- Complementing freedesktop.org
- Chance to bring communities together
  - Maemo
  - Xanadux (HTC folks)
  - OpenEZX
  - GPE (Phone)
  - Treo/Chumby/etc.

Architecture



Statistics (Sep'08)

Subsystems

- Device
- Usage
- GSM
- GPS
- Events
- Preferences
- Phone

Numbers

- 9 Months
- 7 Subsystems
- 22 Interfaces
  - 126 Methods
  - 24 Signals
- 81 Python Modules
- 14857 LOC

Device

Device Control Service

- Info
- LEDs & Vibrator
- LCD
- Peripheral Power Control
- Battery Notifications
- Real Time Clock
- USB Mode Control
- Input (Buttons, Switches, Plugs)
- Idle Notification
- Audio
- Accellerometer

Usage

Resource Manager Service

- High Level Interface for Resource Management
  - Request
  - Release
  - Get State
  - Set Policy
- Resources
  - WiFi
  - Bluetooth
  - GSM
  - GPS
- Reference Counting
- Supports Suspend / Resume (cross layer issue!)

GSM

GSM Telephony Service

- GSM 07.10 Multiplexer
- Modem Abstraction
  - Generic SingleLine
  - Generic MuxedLine
  - TI Calypso
  - Freescale Neptune
  - Sierra
- Device (IMEI, Capabilities, etc.)
- SIM Access (Phonebook, Messagebook, IMSI, etc.)
- Network, Supplemental Services
- Voice Calls, Multiparty Call Handling
- SMS (PDU-Mode!)
- Cell Broadcasts
- GPRS

GPS

Global Positioning System Service

- NMEA
- UBX
- Persistent Ephemeris / Almanac
- fso-gpsd

Events

Rule-based Notification Handler

- Move policies from code to rules
- Triggers
- Filters
- Actions
- YAML
- Incredibly Powerful
- (Demo!)

Preferences

Configuration Service

- Integration with Profiles
- Transparent to Applications
- YAML

Phone

High Level Telephony Service

- Abstract Interface for voice-based communication
- GSM
- VoIP
- ...?
- What about Telepathy?

What can we do now?

- Dialer in 20 lines
- Remote SMS-accellerometer check in 10 lines
- Location-based shopping reminder in 10 lines
- Work together with testing team
- Hardware and Kernel testing with same code as applications are using
- ...

Using it!

- DBus can be used from virtually everything
- Documentation
  - Tutorials @ http://www.freesmartphone.org
  - Reference @ http://docs.freesmartphone.org
- Examples in git.freesmartphone.org
- mdbus
- cli-framework
- (Demo!)
- libframeworkd-glib

What's missing?

- PIM
- Networking
- Alarm
- Mouse & Accellerometer Gestures
- Security Considerations
- Performance Considerations

First Summary

- DBus Service Level Architecture
- Language and Library independence
- Subsystems covering the basics
- Flexible Rules engine
- Until end of this year
  - Tests, Tests, Tests!
  - Examples
  - Docs
- Let's build software on top of it!

Hacking on it!

- (Demo!)

Thanks

- Questions?