Moblin Application Development
Introduction

MMAE Team
Developer Relations Division (DRD)
Software and Services Group (SSG), Intel
Agenda

- Moblin V2 Application Framework
- Development Environment Setup
- App Development Technologies
Moblin V2 Application Framework
Moblin 2.0 Architecture Overview

- ‘Legacy’ UI (QT/GTK)
- Clutter UI (preferred)

Clutter Toolkit
- Cairo
- GStreamer
- GTK / QT
- COGL

UI Services
- QT
- GTK+

App Services
- PIM Server
- Web Svcs
- HTML Renderer
- Dev Mgmt
- GStreamer/Helix
- Sync
- BlueZ
- Pwr Policy
- GLib
- GObject
- GeoClue
- GUPnP
- Pango
- Cairo
- ConnMan
- D-Bus
- OpenGL
- X
- Trusted Boot
- Power Ext
- DRM

Linux Kernel & Drivers

Software and Solutions Group
Developer Relations Division
Component Overviews

- Clutter
- Multimedia Framework
- PulseAudio
- GeoClue
- GTK+
- D-BUS
- Qt
Clutter Overview

Clutter is an open source software library for creating sleek, fast and visually rich interfaces with 3D effects and animations, and new, intuitive, touch-screen paradigms

- Uses OpenGL without exposing the complexity
- Offers a simple, object-oriented API, tailored toward manipulating 2D objects in 3D space and time
- Built on GNOME technologies - glib, gobject, pango
- Developed in C with bindings for Python, Perl, C#
- Released under the GNU LGPL license
Demo
Clutter Architecture

Clutter Application

Clutter Toolkit
- GTK/Qt
- Box2D
- Gstreamer
- Webkit
- COGL

GLib
Pango
Cairo
OpenGL/ES
Clutter Key Features (1)

Supports manipulation of a scene graph of 2D multimedia layers in 3D space

Image Support
• Wide image format support including SVG, PNG, JPEG etc.
• Easily extended to import and integrate other image formats and pixel data sources.

Text Rendering
• Full UTF8 Anti Aliased text rendering using custom Pango based renderer
• Rich layout support, high quality scaling, support for markup with font styles, color and weights

Animation/Special effects
• Rich implicit animation building blocks
• Easily combined and extended
• Built in affects for fades, paths, wipes
• Limitless possibilities with Shader and FBO support for blurs etc.
Clutter Key Features (2)

Scriptable layout and animation file support

Web Rendering
- Integration with the Webkit web browser engine (Mozilla integration planned)
- Provides advanced browsing experience with kinetic panning, real-time content zooms, software keyboard and form support
- Extensible and configurable

Physics
- Integration with Box2D physics engine

Media playback
- Integration with Gstreamer (Helix integration planned)
GTK+ Overview

• **a library for creating GUI**
  - LGPL
  - Multiple platform including Linux/Unix/Windows/
  - Object-oriented architecture in C
  - Provides rich user experience
  - Hide the low level detail.

• **Widgets**
  - Basic widgets like button, check box and text entries
  - Powerful MVC API like tree views, multi-line text field
  - Numerous APIs for convenient uses like file chooser
    and font selection

• **Widgets in GTK+ are placed on windows using a box-packing model**
  - specify only how to pack widgets together in container
    boxes, rather than position them directly
Low level GTK+ Libraries

• **GLib**
  - A general-purpose utility library, not specific to graphical user interfaces
  - Some replacements for standard call like g malloc and additional helper functions for handling linked lists, strings and etc
  - Type system to form the foundation for GTK’s class hierarchy
  - Signal system which is used throughout GTK+
  - A thread API which abstracts the different native thread APIs of the various platforms

• **GDK**
  - It provides drawing and window system facilities on X11, Windows, and the Linux framebuffer device.
  - Use GDK for low-level access to the underlying windowing system, including low-level access to events, windows, and the clipboard
  - Don’t use it to render images. Uses Cairo.
Low level GTK+ Libraries(2)

• Cairo
  – 2D graphics library with support for multiple output devices
  – Features sophisticated API for drawing vector graphics, compositing images
  – It provides operations similar to the drawing operators of PostScript and PDF
  – Uses cairo to draw graphics in your application beyond the widgets provided by GTK+.

• Pango
  – A library for layout and rendering of text
  – Core of text and font handling for GTK+-2.0. Engine for GtkTextView, GtkLabel, GtkEntry and other widgets that display text.
  – Emphasize internationalized text handling.
Media Overview

Helix and GStreamer

Hardware Acceleration enabled for both

• Video Acceleration via LibVA for Menlow and Moorestown devices
• Audio Acceleration via Low Power Audio driver on Moorestown devices
• Codec bundles available from RealNetworks and Fluendo

Support for a Clutter based abstraction layer (ClutterMedia)

• Simple media interface that covers for 80% of the media applications
• More advanced applications will need to drop into the MMF level
• Support for Helix and GStreamer
Media Architecture Using GStreamer

Application

ClutterMedia -> Clutter

GStreamer

- SRC
- demuxer
- * Video Decoder
- * Texture Sink
- * Audio Decoder
- * PA Sink

XServer

PulseAudio

Linux Kernel & Drivers
Media Architecture Using Helix

- Application
  - ClutterMedia
    - HXClientKit Interface
    - Helix Client DNA
  - Clutter
    - Video Renderer
    - Codec
    - Audio Renderer
    - Codec
    - ClutterSite
    - PulseAudioSite
- XServer
- PulseAudio
- Linux Kernel & Drivers
Pulseaudio

- PulseAudio is a sound server for POSIX and Win32 systems.
- A networked sound server, similar in theory to the Enlightened Sound Daemon (EsounD).

**Features**
- Software mixing of multiple audio streams, bypassing any restrictions the hardware has.
- Network transparency, allowing an application to play back or record audio on a different machine than the one it is running on.
- Sound API abstraction, alleviating the need for multiple backends in applications to handle the wide diversity of sound systems out there.
- Generic hardware abstraction, giving the possibility of doing things like individual volumes per application.

- Support for low power audio playback on Moorestown devices
- Library licensed under LGPL and server daemon under GPL
GeoClue

Geoclue is a modular geoinformation service built on top of the D-Bus messaging system.

The goal of the Geoclue project is to make creating location-aware applications as simple as possible.

Geoclue is Free Software, licensed under GNU LGPL.

It is developed for Linux, but should be portable to any platform that uses D-Bus.

In Moblin2 Compliance Spec: v0.11.1
Main features of Geoclue

- Standardized D-Bus (and GObject) interfaces to various geographical data sources

- Easy and fast to implement: Need to get data from provider XYZ? Chances are, Geoclue is the easiest way to do that...

- **Geoclue Master provider** (experimental at the moment) makes things even easier: you don't even have to choose a data provider, just let Geoclue choose the best possible provider for you.

- Adding new providers (data sources) is straightforward: Even when the geoinformation source you want is not already in Geoclue, Implementing a Geoclue provider may well turn out to be the easiest way forward.
D-Bus Overview

• **Inter-process communication mechanism on same host**
  - Fast and lightweight

• **Architecture layers**
  - A library, `libdbus`, a general one-to-one message passing framework
  - A *message bus daemon* executable, built on `libdbus`, that multiple applications can connect to. A router to dispatch message.
  - *Wrapper libraries or bindings* based on particular application frameworks

• **Basic concepts**
  - System bus is a system wide daemon for events like USB is inserted into system
  - Session bus is a per-user login daemon for general IPC demands between apps

• **Moblin uses D-Bus as the primary IPC mechanism for applications.**
D-Bus Usage in the Moblin Application

• Launching application
  When registering an application to D-Bus service, the framework can allow applications to launch only once, running a single instance of an application is an optimized feature for small mobile devices. It is also used to make sure the application survives the task killing process.

• System notification
  The Moblin application can connect and listen to system events, like hardware state and device mode changes, and proceed accordingly.

• Separating applications, user interfaces, and engines
  It's easier to use D-Bus to separate applications, user interfaces, and engines. Then the engine can be used easily from different applications.
D-BUS Service Addressing Mechanism

• The "address" of a remote call has four parts.
  – well-known service name
  – path of the object that implements the remote method
  – interface name
  – function (method) name.

• All four components are needed to address a remote method.
Qt

Qt is a cross-platform application development framework, widely used for the development of GUI programs, and also used for developing non-GUI programs such as console tools and servers.

Qt uses C++ with several non-standard extensions implemented by an additional pre-processor that generates standard C++ code before compilation. Qt can also be used in several other programming languages; via language bindings.

Non-GUI features include SQL database access, XML parsing, thread management, network support and a unified cross-platform API for file handling.

Distributed under the terms of the GNU LGPL (among others), Qt is free and open source software.
Qt Features

- Intuitive C++ class library
- Portability across desktop and embedded operating systems
- Integrated development tools with cross-platform IDE
  - Qt Creator cross-platform IDE
  - Qt libraries
  - Qt Linguist translation & internationalization tools
- High runtime performance and small footprint on embedded
**Qt License**

**Commercial Version**
- For development of proprietary and/or commercial software with Qt where you do not want to share any source code.
- For regular desktop applications, there are no royalties, runtime licenses, or other additional costs.
- For fixed systems or devices where the main functionality relies on Qt, a runtime distribution license may apply.
- The license is sold on a per-developer basis and assigned to an individual. Qt licenses may be transferred, but only every six months and within the same organization.

**LGPL**
- Only can be used in Qt V4.5 and later version
- Support services are available separately for purchase 12 months Qt Standard Support $1970

**GPL**
- Support services are available separately for purchase 12 months Qt Standard Support $1970
Development Environment Setup
Development Environment

- 2 ways to setup a development environment
- Develop on your Linux workstation with an occasional remote validation and debugging task on a target device.
- Develop within a Moblin v2 environment using one of the following: a KVM image, a VMWare image, or on a platform such as a netbook.
## Comparison of the 2 ways

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Developing on Linux workstation</th>
<th>Developing in Moblin v2 environment (KVM, VMWare, or netbook)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benefits</td>
<td>Familiar desktop, configuration, network setup, etc.</td>
<td>A Moblin compliant distribution. No need to update libraries.</td>
</tr>
<tr>
<td></td>
<td>Standard development tools and environment</td>
<td>Application runs in actual target environment.</td>
</tr>
<tr>
<td></td>
<td>Running on powerful workstation</td>
<td>(On target device) Have platform-specific features and drivers.</td>
</tr>
</tbody>
</table>

| Drawbacks                                                                 | Must align (install or update) development packages with [Moblin v2 libraries](https://example.com/moblin-v2-libraries). | Slower performance                                            |
|                                                                          | UI experience is different than mobile device         | Must install necessary tools                                  |
|                                                                          | May not have platform-specific capabilities (such as touch screen, telephony stack, etc) | Smaller screen area for debugging                              |
|                                                                          |                                                                                                 | Netbook keyboard/mouse small or awkward                       |
Moblin Image Creator 2 Introduction

- Moblin Image Creator 2 (MIC2) is a tool for creating and manipulating Moblin images.
- MIC2 is a series of utilities.
  - `moblin-image-creator` for image creation
  - `moblin-image-converter` for image transformation
  - `moblin-chroot` for generating chroot environment from image and vice versa
- MIC2 is NOT based on MIC 1.0.
Moblin Image Creator 2 (cont.)

- Supports major Linux distributions and can be used on at least the following distributions:
  - **Fedora** (Fedora 9 and above)
  - **Opensuse** (> OpenSUSE 10.3)
  - **Ubuntu** (Ubuntu 8.10. MIC2 is not supported on Ubuntu 8.04 due to incompatible yum versions)

- Supports various types of images:
  - Live CD
  - Live USB
  - Loop images
  - KVM images
  - VMDK (Vmware)
MIC2 Installation Prerequisites

- The following packages **MUST** be installed before MIC2 installation:
  - Yum, rpm, kpartx, kparted, syslinux, isomd5sum, kvm, zlib-devel, python-devel

- Specific packages for Fedora and Moblin:
  - Pykickstart, device-mapper

- Specific packages for Ubuntu 8.10:
  - python-celementtree, python-elementtree, dmsetup

- Modules should be loaded
  - Squashfs, dm_snapshot, loop
MIC2 Installation

2 ways: From binary or From source

• From binary
  1. Check [http://repo.moblin.org/moblin/tools](http://repo.moblin.org/moblin/tools) and download binary packages (.deb or .rpm)
  2. Install 😊

• From source
  1. Check [http://repo.moblin.org/moblin/tools](http://repo.moblin.org/moblin/tools), download and unzip source codes
     or
     git clone git://git.moblin.org/moblin-image-creator-2
  2. cd moblin-image-creator-2
     make clean
     make
     sudo make install
Using moblin-image-creator2

```
sudo moblin-image-creator --config=netbook-core.ks --format=livecd --cache=mycache
```
Kickstart Configuration

- Configuration of images is based on kickstart, the format used for unattended installation in Fedora and Redhat.

- 2 existing kickstart configurations so far in MIC2
  - netbook-core.ks – Target image
  - netbook-core-developer.ks – For development

- User can customize the image by modifying the kickstart file.
Using moblin-chroot

- Two major usage models
  - One-shot image post processing
    ```
    sudo moblin-chroot -c livecd moblin-netbook-core-200903131337.iso
    ```
  - Multiple times changes
    ```
    sudo moblin-chroot -s chroot-fs-2 --unpack-only moblin-netbook-core-200903131337.iso
    ```
    ```
    sudo moblin-chroot chroot-fs-2
    ```
    ```
    sudo moblin-chroot -c livecd --convert-only chroot-fs
    ```
Using moblin-image-convertor

- This utility is used to transform a virtual machine image (KVM or VMDK) to a live image.

```
sudo moblin-image-convertor --srcimg=InputImage --dstformat=Targetformat
```
Application Development Tools
Supported Programming Languages

- C/C++
- Python
- Adobe* Flash*
- Adobe* AIR*
- Mozilla* App
- Widget Solution
Development Tools

• MIC2

• PowerTOP
  - A Linux tool that helps you find the programs that are misbehaving while your computer is idle, and provides tuning suggestions to achieve low power consumption.

• Linux Project Generator

• Anjuta DevStudio
  - An IDE which is easy to install and use and may be preferred over emacs or vi for those used to developing with graphical interfaces.

• Eclipse
  - Another very popular IDE
Linux Project Generator

- Linux Project Generator is a tool that can be used to quickly generate an autogen-enabled Linux project. It has a growing set of sample code that can be selected and included.
- It can generate projects in C, C++, or Python.

![Linux Project Generator Interface](Image)
## LPG Feature List

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>autogen enabled</td>
<td>Project can be built and installed with standard 3 commands: autogen.sh, make, (sudo) make install.</td>
</tr>
<tr>
<td>Customize project data</td>
<td>User enters name, path, language, license, etc.</td>
</tr>
<tr>
<td>C, C++, Python</td>
<td>Select which language you want your template project to be written in.</td>
</tr>
<tr>
<td>Project type: app, lib, or combo</td>
<td>Select if you want your project to generate an application, a library, or any combination (such as two app binaries and a library).</td>
</tr>
<tr>
<td>Dependent Libraries</td>
<td>Select the libraries to include. Uses pkg-config to generate appropriate build flags.</td>
</tr>
<tr>
<td>(Moblin) Compliance</td>
<td>Check whether to only display Moblin compliant libraries and versions to include as dependencies.</td>
</tr>
<tr>
<td>License</td>
<td>Pick from the most common open source licenses. Appropriate COPYING and code headers are generated. Supports GPL 2, GLP 2+, GPL 3, LGPL 2.1, LGPL 3, MIT, and BSD. New or custom licenses can be easily added.</td>
</tr>
<tr>
<td>gtk-doc</td>
<td>Check whether your library should auto-generate documentation from gtk-doc style comments.</td>
</tr>
<tr>
<td>I18N</td>
<td>Check whether to add support and examples for internationalization.</td>
</tr>
<tr>
<td>Sample Code</td>
<td>Select the samples you want included in your template. Ideally most tutorials and 'how-to' documentation also include a block of code that can be built into a template project.</td>
</tr>
<tr>
<td>automation</td>
<td>Supports most features as command-lined parameters.</td>
</tr>
</tbody>
</table>
Widget Overview

What is Widget
- A small plug-in which is usually shown as a small window or a dialog box
- Show on the web, desktop or mobile phone
- A small stuff which is attractive and easy to use
- Usually Plugged with HTML, Javascript, Flash or iframe
- Different widgets show different source (web service) in one page

Utility of widget
- Offer richer dynamic effect on the web or desktop
- Provide a service to other websites, and extend the brands and user groups by using virus-like sales mode
- Offer services for SNS users and gain more users fast
Moblin widget flash demo
The widget stack of the Product is built on Moblin OS
# Features overview (1)

<table>
<thead>
<tr>
<th>#</th>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Widget Standard</td>
<td>Widget based on W3C specification, and Compliance with Sohu/Apple/Nokia mobile widget</td>
</tr>
<tr>
<td>2</td>
<td>Plug-in</td>
<td>Support Plug-ins, such as Flash player.</td>
</tr>
<tr>
<td>3</td>
<td>Ajax Support</td>
<td>Provide API for sending and receiving data by means of Ajax.</td>
</tr>
</tbody>
</table>
| 4  | UI               | • Floating effect  
• Borderless  
• Irregularly shape  
• Translucent window |
| 5  | Event support    | The widgets can communicate with each other by event                                               |
## Features overview (2)

<table>
<thead>
<tr>
<th>#</th>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>SDK</td>
<td>Standard framework, open to the third party</td>
</tr>
</tbody>
</table>
| 7  | Widget manage                    | • Read Widget information from configuration file and register into Application Manager after downloading Widget  
• Delete Widget information from Application Manager. |
| 8  | Widget Desktop                   | • Special container to hold various kinds of widget applications          
• Different widget applications can be shown in full-screen or floating mode on the desktop |
| 9  | Local Resource Access API         | • Widget app can read and write local file                                 
• Camera control and get info                      
• Network status notification (TBD)                
• GPS control and get info (TBD)                    |
New experimental
Entertainment stack + Apps Store

ODM: BYD (BiYaDi)
OSV: Moblin OSV
Platform: Menlow Refresh

TTM: Q2’09
Entertainment stack ready by early Q109
Widget engine ready by early Q209
Online app store ready by Q209

Apps store owner: Sohu
Widget engine owner: Neusoft

About Neusoft and Sohu:
Neusoft is the largest offshore software outsourcing service provider in China. Sohu.com is No.2 portal website in PRC.
Moblin/MID Widget SDK

Sign NDA and get SDK from Neusoft before it is public
Based on Java eclipse as a plug-in
Running on Windows OS (Linux version is in plan, but not available)
Features:
• Develop guidance and help document which is integrated in Eclipse
• Create default project (default files)
• Coding language edit (has symbol highlight function)
• Debug with MID simulator
• Build widget package
Thanks!
BACKUP
BKM of keeping application UI style consistent in different OEM customized OS

- Design a set of GTK theme for application.
- Load the theme by gtk_rc_parse()
- Minor source code modification
- Can change UI style with no touch on source code
Moblin Application Template

What is prerequisites (including headers and libraries):

- X11
- Cairo
- OpenGL (usually via Mesa)
- Glib
- Clutter
- Standard Linux development tools (gdb, gcc, autoconf, and so on)
- Source code management utilities, including: subversion, git

You can obtain the latest project template from Moblin.org