



Connection Manager **for embedded Linux systems**

Linux-Kongress 2008

Marcel Holtmann
Open Source Technology Center



What is ConnMan?

- ConnMan is the Connection Manager for Moblin
- Handle Internet/Intranet IP connection setup
- Completely new architecture
- Written from scratch
- Targets at embedded distributions
- Network Manager version 0.6 was just not good and flexible enough
- Everything else is closed source

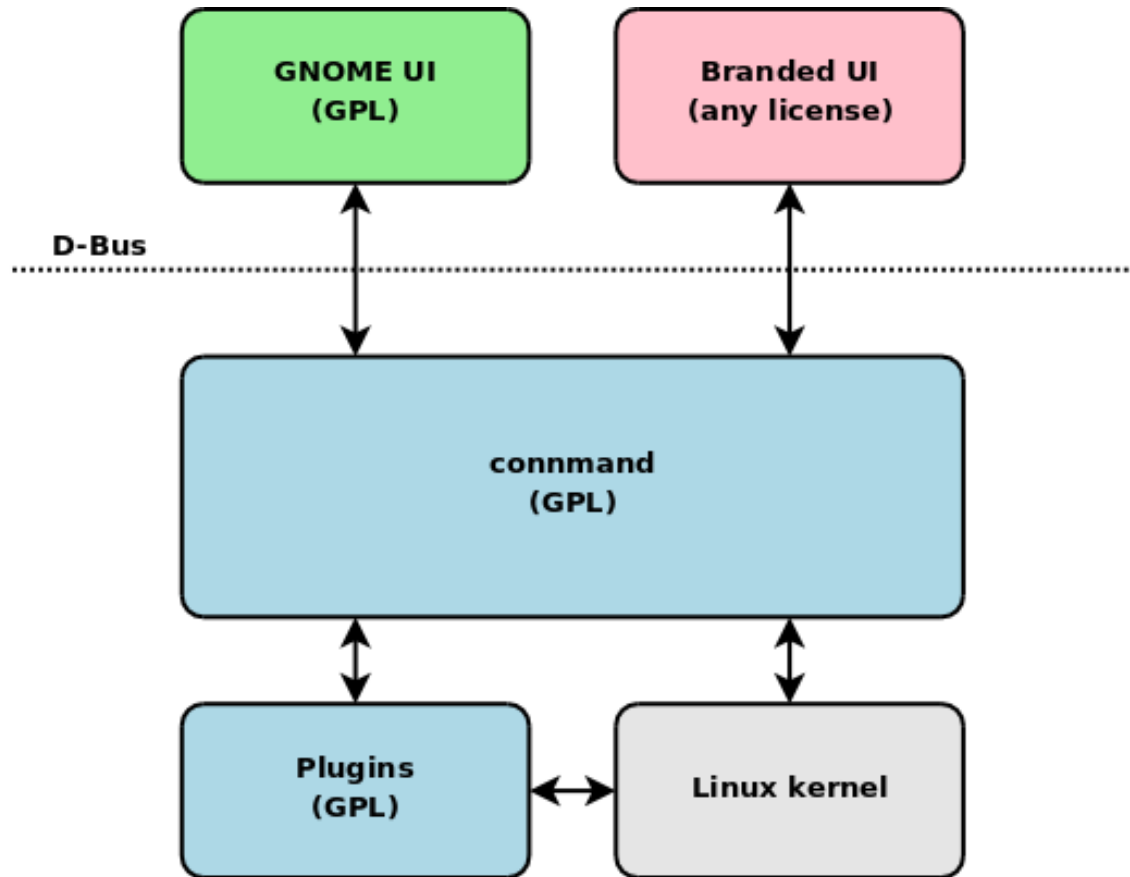
Design goals

- Full modular system
 - Core functionalities are separated into plugins
 - Plugins for various wireless technologies
- Separation between user interface and system daemon
 - Core daemon handles all networking functionality
 - Fully functional without any user interface
 - Make it really easy to replace UI components
 - Use D-Bus to give maximum freedom for license of UI
- Ready for embedded devices
 - Minimal dependencies
 - Small footprint
 - Component based model

License overview

- ConnMan core daemon is licensed under GPL v2
- All plugins require the GPL v2 license
- User interfaces or management tools can be licensed under any license
- Proprietary and closed sources user interfaces are explicitly allowed

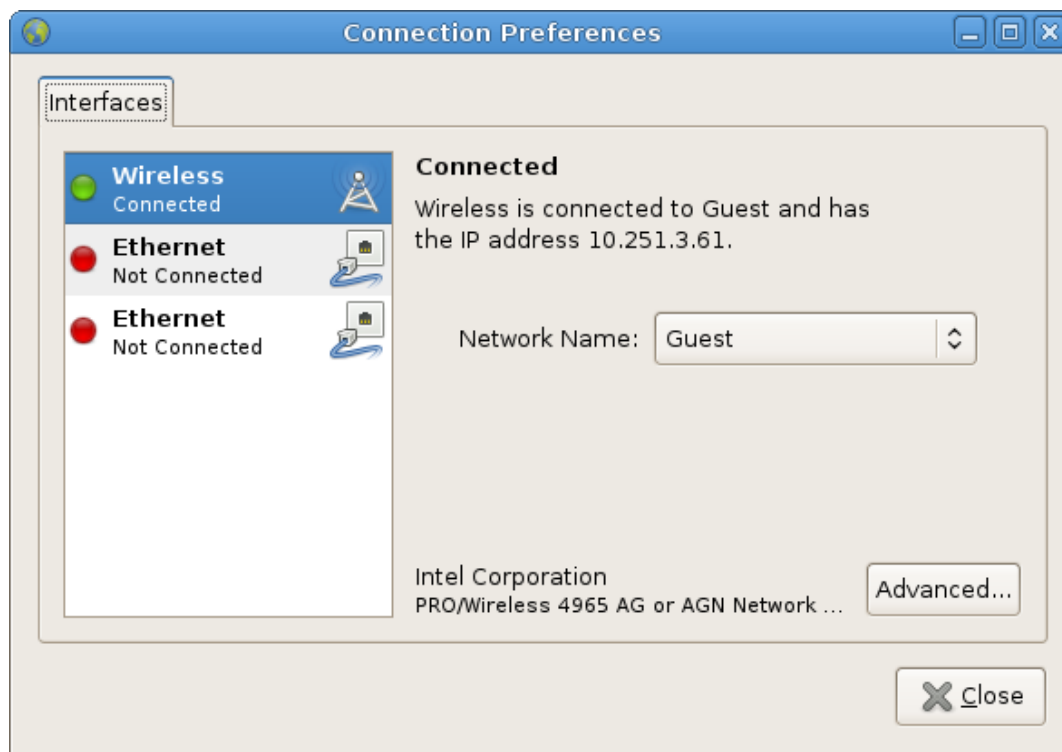
Architecture overview



User interfaces

- User interfaces communicate with the core daemon using the D-Bus system message bus
- Multiple interfaces can be running at the same time
- Possibility to split tasks into separate applications
- All state changes are propagated via D-Bus signals to support fully asynchronous applications
- Common split of user interfaces
 - System tray application for network status and network selection
 - Settings application for detailed configuration

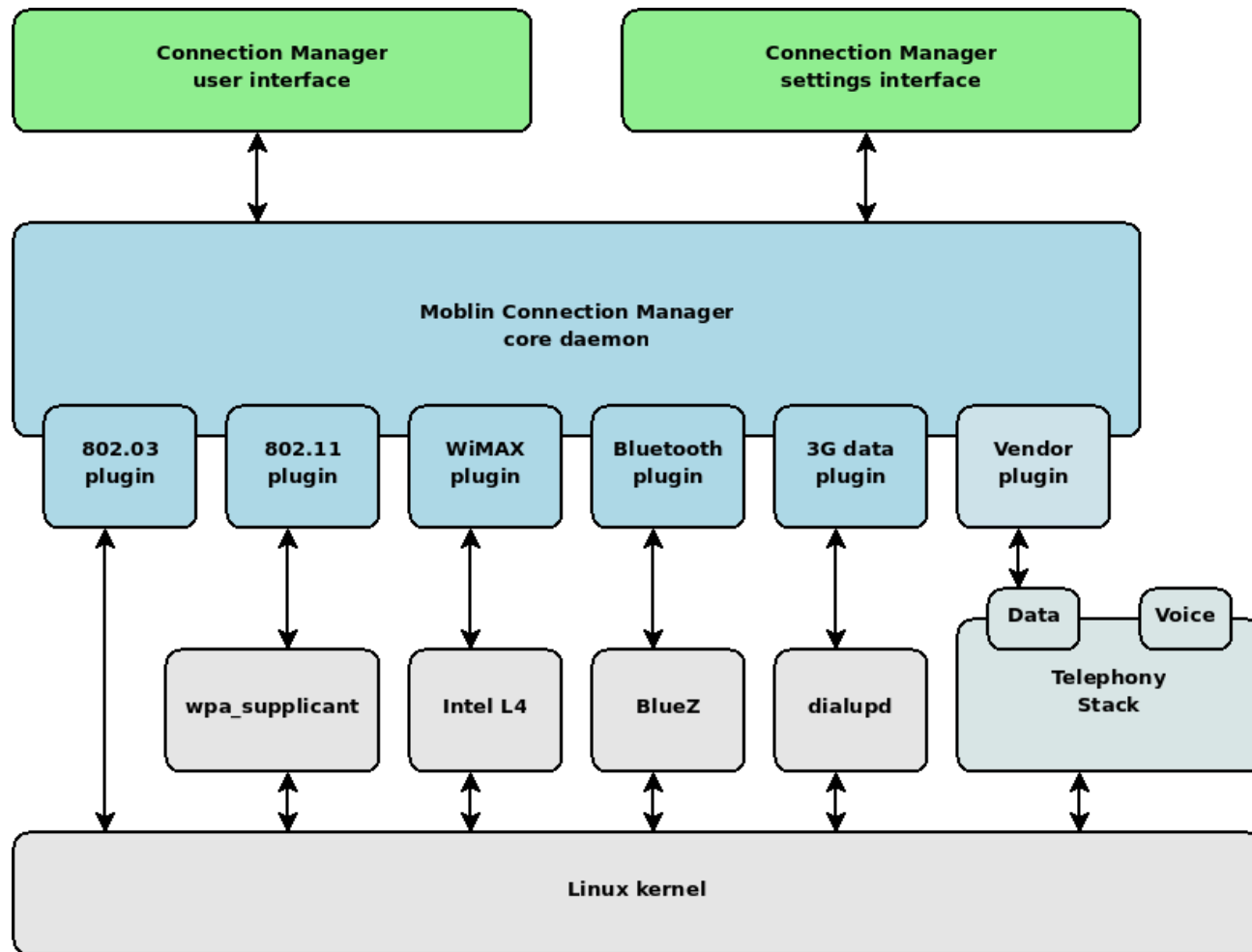
Example of a settings dialog



Use common infrastructure

- Built on top of common solution
 - wpa_supplicant for WiFi devices
 - BlueZ for Bluetooth PAN and DUN access
 - Intel L4 for accessing WiMAX networks
 - dialupd/pppd for GSM/UTMS and generic modem access
 - dhclient for DHCP transactions
 - Integrate with PolicyKit
- Well established solutions with a big developer community behind them
- Allows code and infrastructure sharing with other solutions like Network Manager

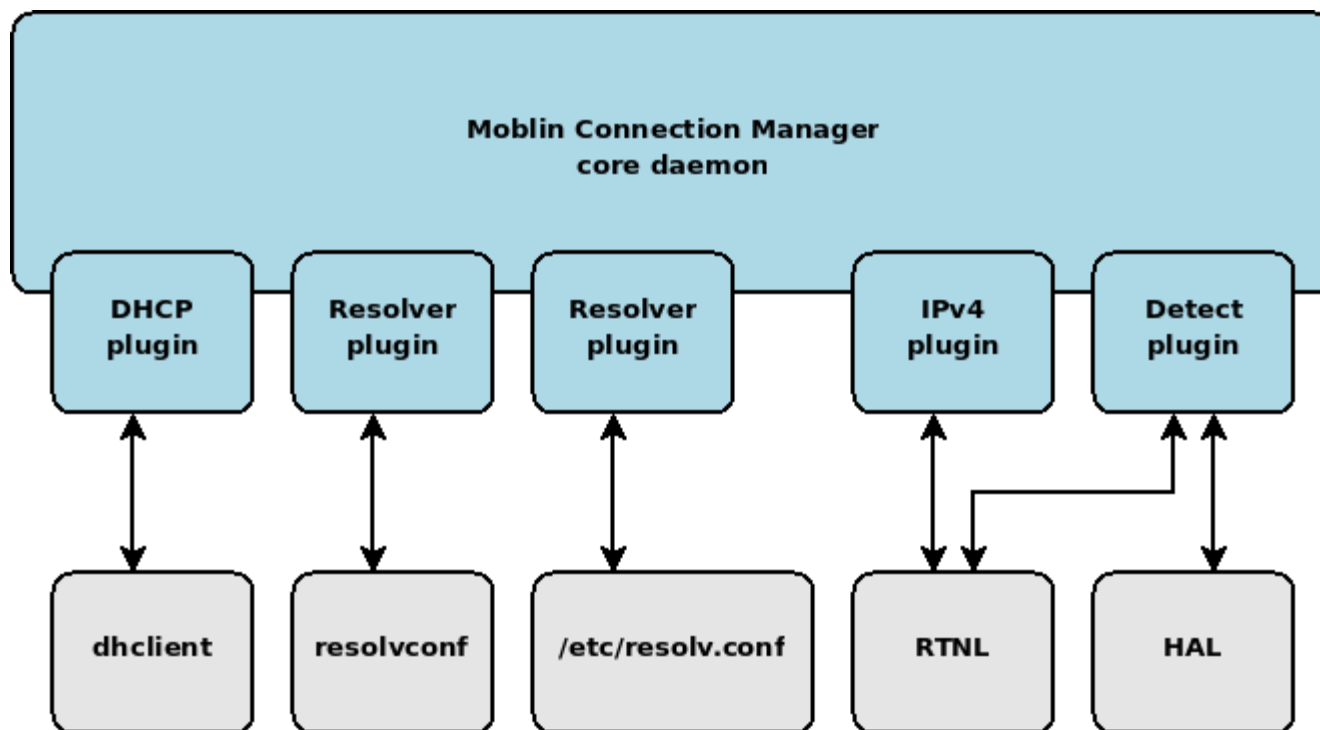
Support multiple technologies



New technologies

- Extending ConnMan with new technologies is simple
- Hardware detection can be done by HAL or also via vendor plugins if needed
- Plugins can register drivers for vendor devices
- Common elements like DHCP, IPv4 and resolver support can be easily re-used
- All plugins have to be licensed under GPL v2

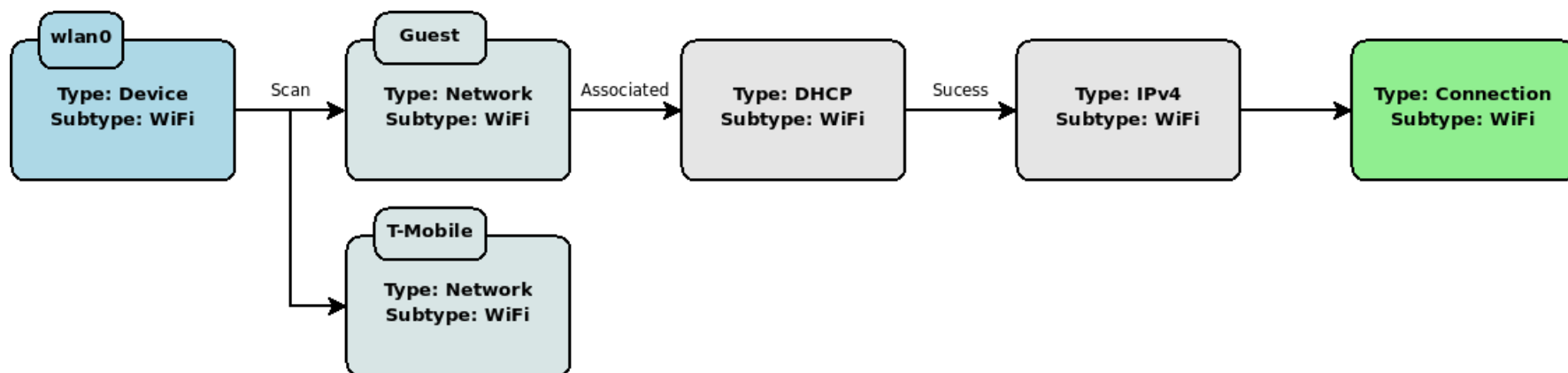
Other common elements



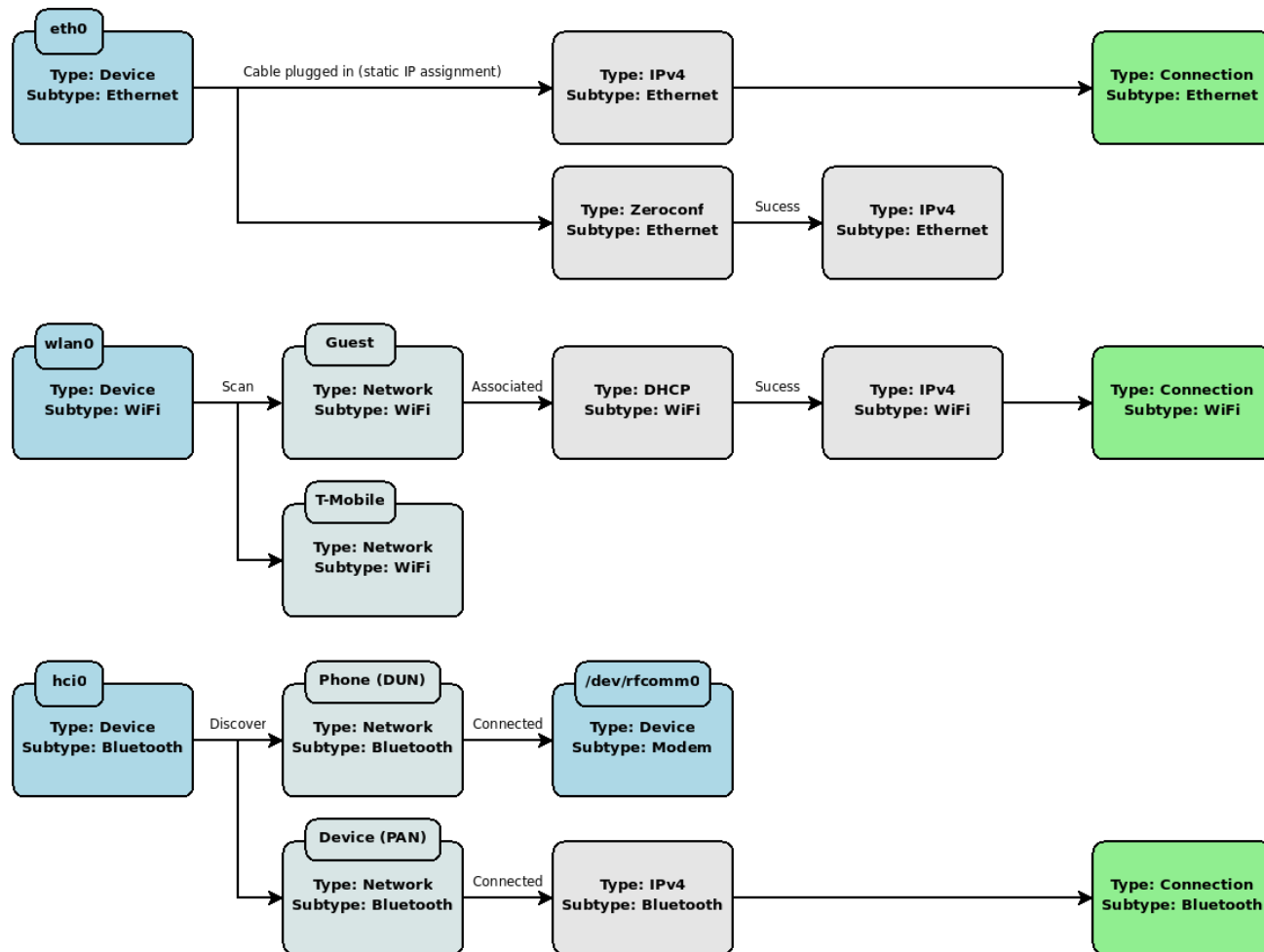
Element architecture

- Every piece of information inside ConnMan is encapsulated into an element
- Common elements for devices, networks, DHCP, Ipv4 etc. are defined by the core and can be used
- Vendor specific elements are possible
- All elements are connected via an N-ary tree
- Children can overwrite settings of parents
- Set of basic and advanced properties
- Full tree is exported via D-Bus for flexible configuration

Pieces of a network connection



Connecting elements



Output of list-elements

```
[ /net_00_1d_60_c8_d8_7f ]
  Subtype = ethernet
  Product = 88E8056 PCI-E Gigabit Ethernet Controller
  Vendor = Marvell Technology Group Ltd.
  Type = device
  Driver = sky2

[ /net_00_1d_60_c8_d8_7f/dhcp ]
  Parent = /net_00_1d_60_c8_d8_7f
  IPv4.Gateway = 192.168.1.1
  Subtype = ethernet
  IPv4.Netmask = 255.255.255.0
  IPv4.Address = 192.168.1.101
  Type = dhcp

[ /net_00_1d_60_c8_d8_7f/dhcp/ipv4 ]
  Subtype = ethernet
  Type = ipv4
  Parent = /net_00_1d_60_c8_d8_7f/dhcp

[ /net_00_1d_60_c8_d8_7f/zeroconf ]
  Subtype = ethernet
  Type = zeroconf
  Parent = /net_00_1d_60_c8_d8_7f

[ /hci0 ]
  Subtype = bluetooth
  Type = device
```

Driver architecture

- Drivers can be attached to elements
- All functionality is implemented by drivers
- Multiple drivers can be available, but only one can be attached to an element
- Driver selection is done via matching parameters and priority listing
- Similar to the definition of kernel drivers
- Plugins can register multiple drivers if needed

Plugin example

- Installed under */usr/lib/connman/plugins/* and loaded at start of the core daemon (no loading at runtime)
- Similar to kernel modules with *init* and *exit* callbacks

```
#include <connman/plugin.h>

static int example_init(void)
{
    return 0;
}

static void example_exit(void)
{
}

CONNMAN_PLUGIN_DEFINE("example", "Example plugin", VERSION,
                      example_init, example_exit)
```

Driver example

```
#include <connman/plugin.h>
#include <connman/driver.h>

static int ipv4_probe(struct connman_element *element)
{
    /* set IPv4 address */

    return 0;
}

static void ipv4_remove(struct connman_element *element)
{
    /* remove IPv4 address */
}

static struct connman_driver ipv4_driver = {
    .name      = "ipv4",
    .type      = CONNMAN_ELEMENT_TYPE_IPV4,
    .probe     = ipv4_probe,
    .remove    = ipv4_remove,
};

static int ipv4_init(void)
{
    return connman_driver_register(&ipv4_driver);
}

static void ipv4_exit(void)
{
    connman_driver_unregister(&ipv4_driver);
}

CONNMAN_PLUGIN_DEFINE("ipv4", "IPv4 plugin", VERSION, ipv4_init, ipv4_exit)
```

Future development

- Adding full support for profile management
 - Every profile will be the root element of its own tree
 - Switching profiles by selecting a tree
- Generic RTNL handling
 - RTNL is bound to events and not interfaces
 - Usage of tools like ifconfig, route etc. for IP configuration
- WiMAX and Bluetooth support
 - Integration with the Intel L4 WiMAX stack
 - Integration with BlueZ
 - To be merged back into the master ConnMan repository
- Finish the GTK+ based user interfaces

Release plans

- No release has been made so far
- That is on purpose
- Public accessible GIT trees
- Snapshots are available
- First release planned as Christmas present

Side projects

- **libgdbus**
 - Library for simple D-Bus integration with GLib
 - No need for GObject and dbus-glib
 - Also used for bluetoothd and obexd now
- **Kernel D-Bus**
 - Re-implementing dbus-daemon as kernel subsystem
- **dialpupd**
 - Modem Manager like daemon for GSM/UTMS access
 - Make it fully pluggable to support non-TTY device like PhoNet
- **libgdhcp**
 - Small DHCP library that integrates with the GLib mainloop

Questions?

- Website
<http://www.moblin.org/projects/connection-manager>
- Snapshots
<http://ftp.moblin.org/connman/snapshots/>
- Mailing list
connman@moblin.org
- #connman on freenode.net

Legal information

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel may make changes to specifications, product descriptions, and plans at any time, without notice.

All dates provided are subject to change without notice.

Intel is a trademark of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2008, Intel Corporation. All rights are protected.



