

Wicked Trip into Wicked Network Management

Matthias G. Eckermann

Senior Product Manager

mge@suse.com

LinuxCon 2013 NA
2013-09-12 15:40 UTC



Some Words of Encouragement

There is a theory which states, that if ever anybody discovers exactly what the Universe is for and why it is there, it will instantly disappear and be replaced by something even more bizarre and inexplicable.

There is another theory which states that this has already happened.

Douglas Adams

Back in 1992, the world was flat

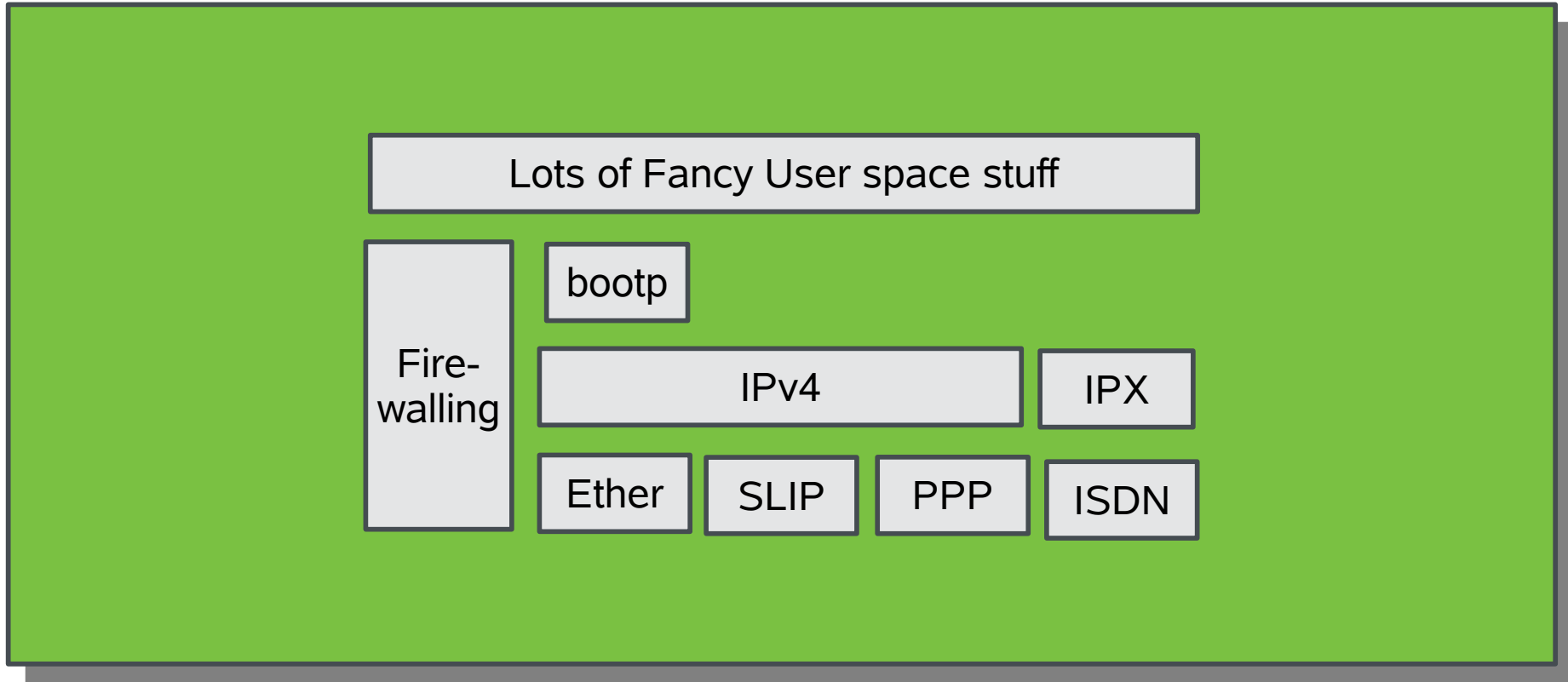
Mail

Usenet

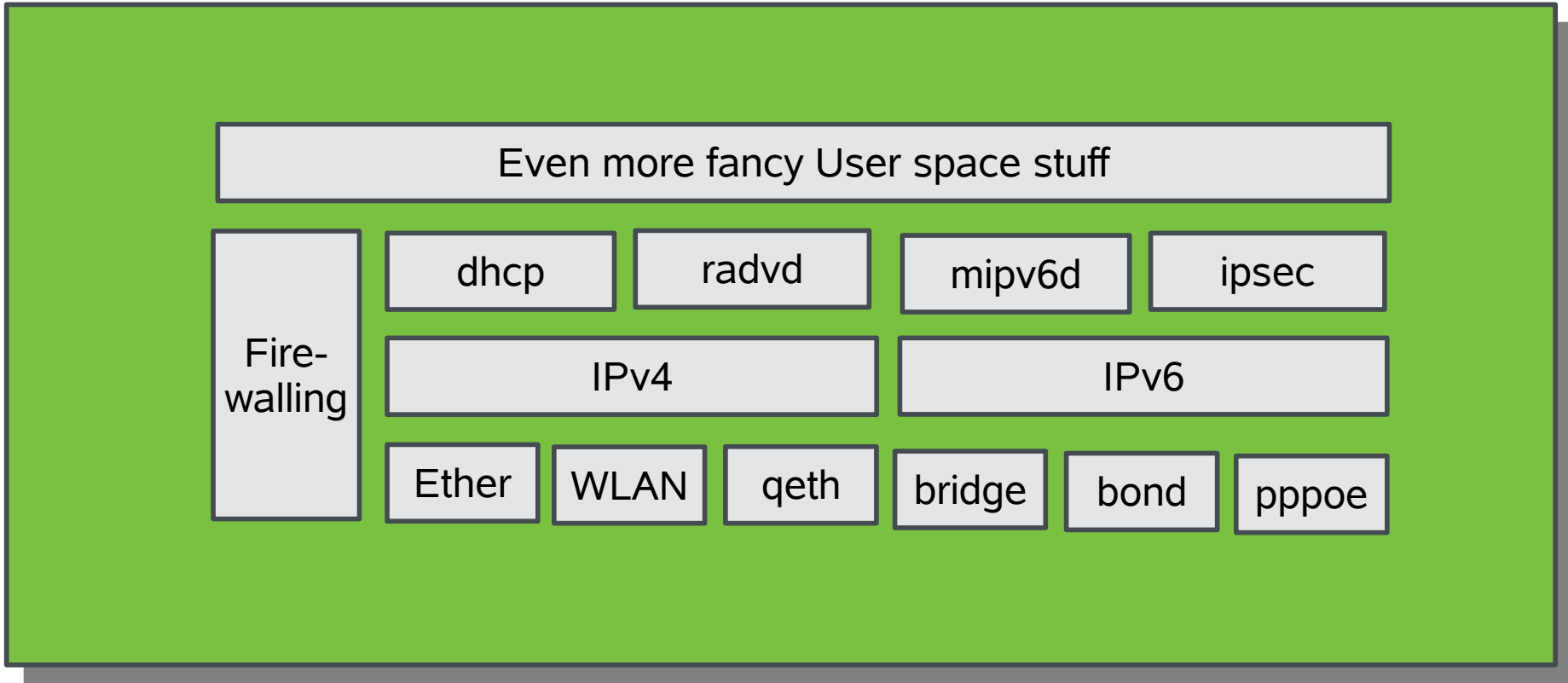
UUCP

Expensive Modem

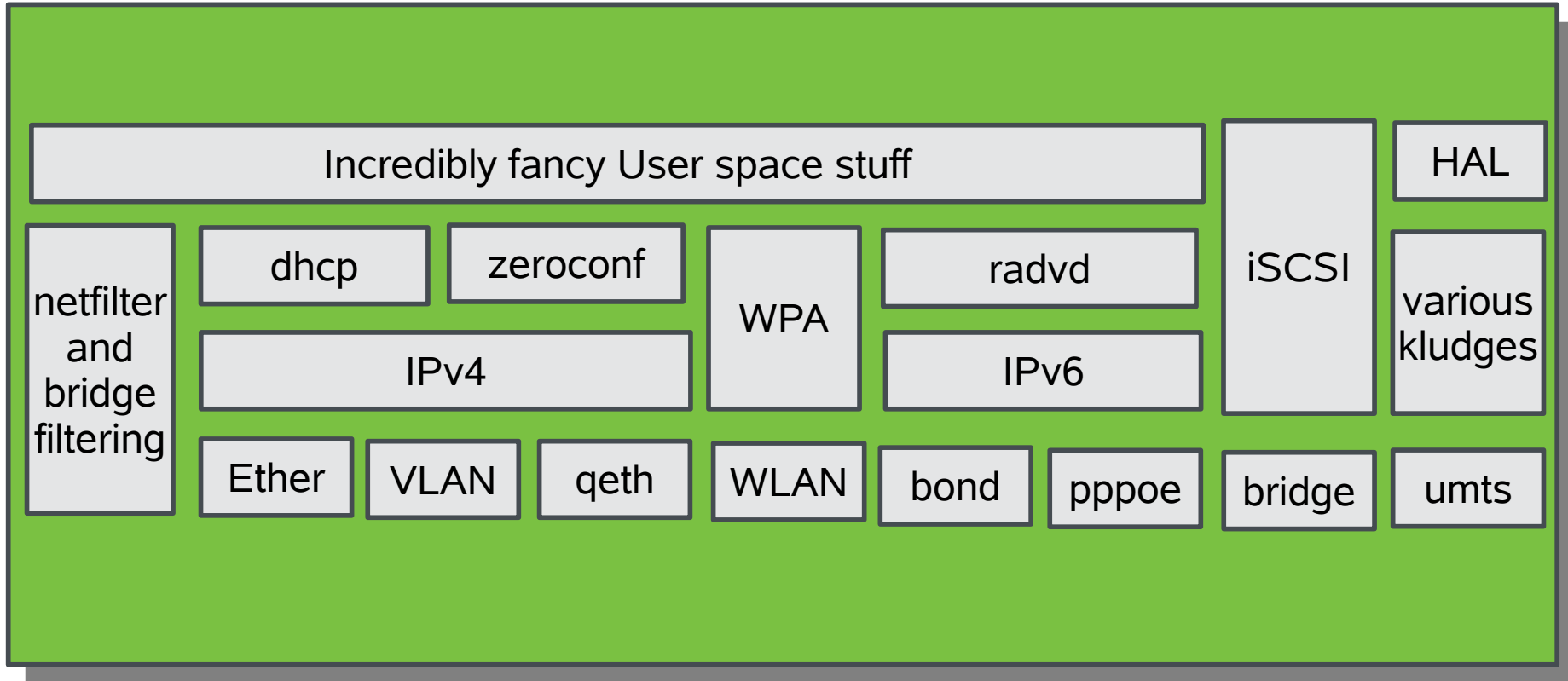
Five years later, we had IP networking



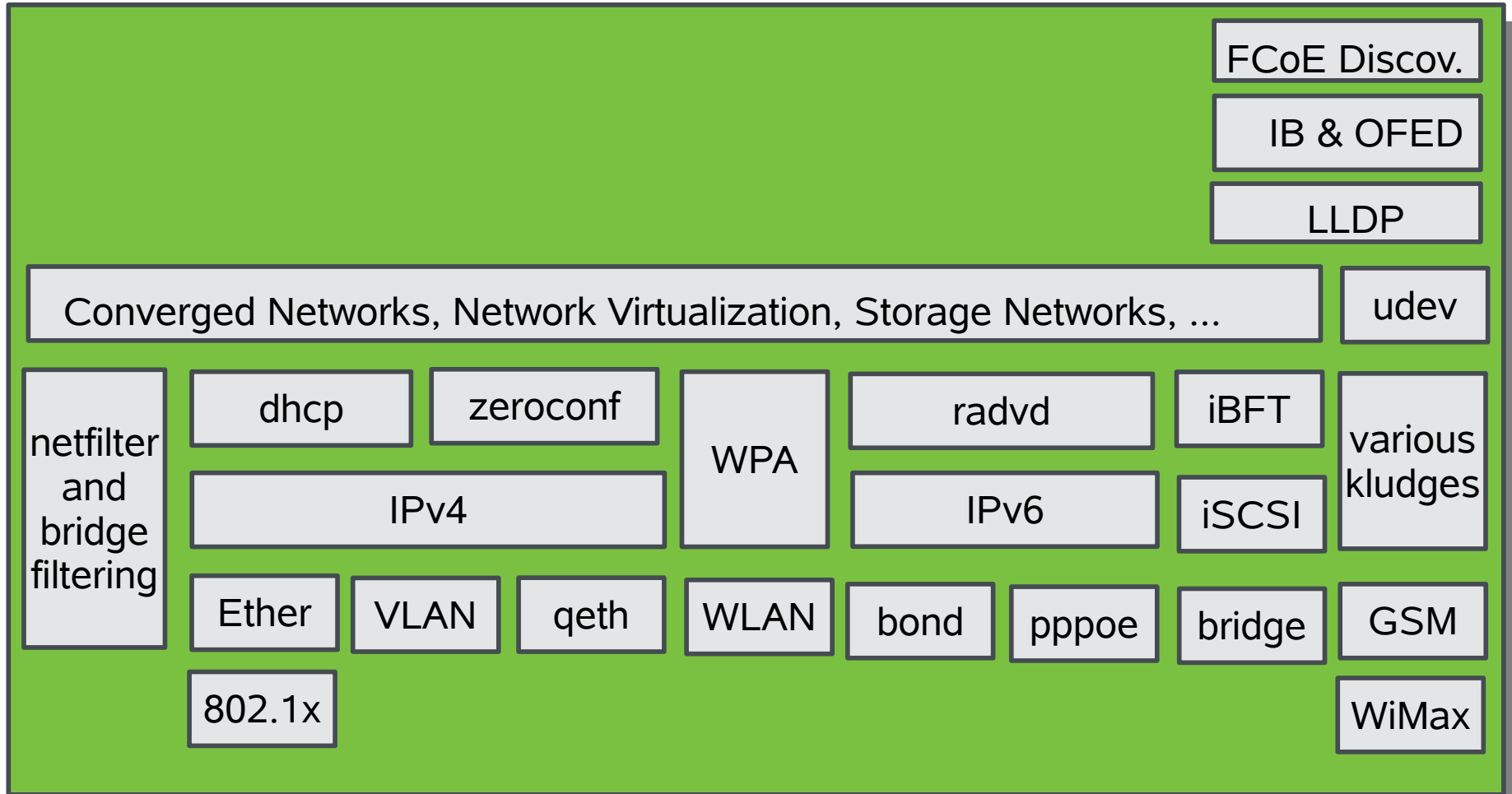
In 2002, we got IPv6



2007: Mobility and Data Center diverge



2012: The Universe changed again



Wicked Trip into Wicked Network Management

Matthias G. Eckermann

Senior Product Manager

mge@suse.com

LinuxCon 2013 NA
2013-09-12 15:40 UTC



Agenda

- What we want to achieve
- About Wicked
 - Wicked history
 - Implementation decisions
 - Compents
 - Supporting Migration
- Object Model and Layering

What we want to achieve

Goal

- Cope with increasingly complex configurations

Target Audience

- Data Center *and* End Users

Positioning

- Network configuration is a service

Usability

- Make adoption as smooth as possible



What we want to achieve (cont.)

Technical Attributes

- Architecture-independent
- Extensible
- Needs small footprint (initrd use)
- React flexibly to network changes
- Broadcast event notifications
 - interface comes up, IP address assigned, routing changed

What we don't want to achieve

- Replacing NetworkManager completely
- World domination
(aka locking users into a specific tool set)

Wicked history

- Started as a hack week project for network monitoring
- Morphed into “try to do better than ifup”
- Original design was based on a REST interface
 - Worked, but ...
- Second Iteration moved to a dbus interface
 - Much better, but a bit of a learning curve

Implementation Decisions

- Client / Server model
 - Dbus Service (provided by a daemon)
- Layered architecture
 - providing separate Dbus interfaces
- Structured configuration files
 - XML for now
- Stateless (mostly)
- Extensible
 - Server can be extended with scripts

Components

- wickedd, the server process
- wicked command line utility
- dhcp4, dhcp6 and other supplicants
- network-nanny (support interface hotplugging)
- Future: need help with writing a taskbar applet

Supporting Migration

Existing Configuration files

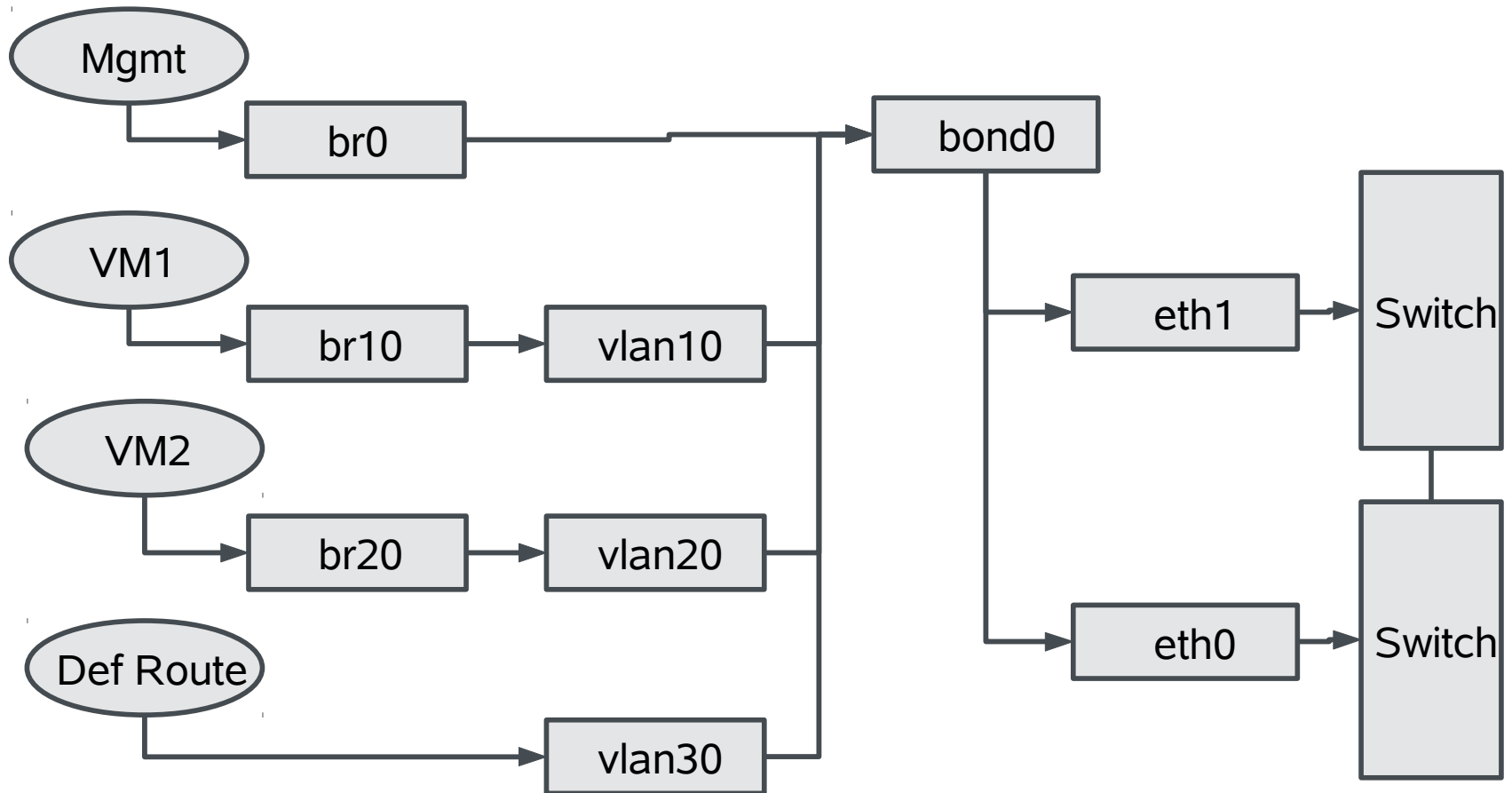
- wicked client can read ifcfg files (both the green and the red flavours)
- Plan to provide ifup wrapper scripts

Testing

- Conflict as little as possible with existing infrastructure
- Just drop in an test

Example – Virtualization

Virtualization Host, Simple Case



Object Model and Layering

Wicked object model

- The central DBus objects are network interfaces
 - /org/opensuse/Network/Interface/*
- Each object supports a set of DBus “interfaces”
 - Unfortunately, we're overloading of the term interface a bit
 - which is why we will refer to network *devices* subsequently
- Many DBus interfaces are generic, but some are specific to the device type
 - Ethernet, loopback, VLAN virtual device, etc

Wicked object model, cont'd

- Network device object naming based on the kernel's interface index
 - /org/opensuse/Network/Interface/<ifindex>
 - pro: invariant against interface renames
 - con: no object without existing device (you need factory functions to create virtual devices for VLANs, bridges, bonds)

Wicked object model, cont'd

- DBus interfaces are named `org.openSUSE.Network.*`
 - Heavy use of polymorphism to simplify the code
- `org.openSUSE.Network.<DeviceType>`
for device-specific ones, like Ethernet, VLAN
 - they all export a “changeDevice()” method,
taking a DBus dict as argument
- Generic interfaces supported by all network devices,
such as `org.openSUSE.Network.Interface`

Wicked object model, cont'd

- 1:1 correspondence between DBus interfaces and sections of a config file
 - Simplifies the client side code significantly
 - Simplifies extending the supported configuration options

Wicked object model, example

Ethernet device

DBus Interface

org.opensuse.Network.Ethernet

org.opensuse.Network.Firewall

org.opensuse.Network.Interface

...Addrconf.ipv4.static

...Addrconf.ipv6.static

...Addrconf.ipv4.dhcp

...Addrconf.ipv6.dhcp

DBus methods

changeDevice

firewallUp,firewallDown

linkUp,linkDown

requestLease,dropLease

requestLease, dropLease

requestLease, dropLease

requestLease,dropLease



Ethernet device configuration

```
<interface>
  <name>eth0</name>
  <ethernet>... </ethernet>
  <link>...</link>
  <firewall> ... </firewall>
  <ipv4:static>
    <address>...</address>
    <route>...</route>
  </ipv4:static>
  <ipv4:dhcp/>
</interface>
```

Summary

Current Status

- Implemented
 - Ethernet
 - VLAN
 - Bridging
 - Bonding
 - dhcp4
 - dhcp6
 - IPv4 zeroconf
 - Static addressing
- In implementation
 - Wireless (using wpa-supPLICANT)
 - ibft
- Documentation needs improvements

Thanks

- Olaf Kirch
- Marius Tomaschewski

Try it

<http://software.opensuse.org/package/wicked>

Clone it

<https://github.com/openSUSE/wicked>

Your questions!?





Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

+49 911 740 53 0 (Worldwide)
www.suse.com

Join us on:
www.opensuse.org

Unpublished Work of SUSE. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

