Desktop on the Linux (and *BSD of course). . .

you’re doing it confused? weird? strange? wrong?

Who? Wolfgang ’datenwolf’ Draxinger

When? 27c3, 2010-12-27

DISCLAIMER

This talk is:

- highly opinionated
- biased
- born out of frustration
- . . . and anger

DISCLAIMER II

Linux is not Unix.

Nevertheless I'll mix the terms because I'm just to lazy to distiguish everytime.

I hope you're okay with that.

DISCLAIMER II

Linux is not Unix.

Nevertheless I'll mix the terms because I'm just to lazy to distiguish everytime.

I hope you're okay with that.

DISCLAIMER II

Linux is not Unix.

Nevertheless I'll mix the terms because I'm just to lazy to distiguish everytime.

I hope you're okay with that.

The situation

I work as a systems administrator:

- University's physics student computers.
- $\geq$ 3500 users!
- I'm the "problem solver" there.

My pleasure hacking projects are about:

- realtime graphics
- realtime simulation
- systems programming

a.k.a. *game engines*.⇒ highly optimized, resource aware code.

The situation

I work as a systems administrator:

- University's physics student computers.
- $\geq$ 3500 users!
- I'm the "problem solver" there.

My pleasure hacking projects are about:

- realtime graphics
- realtime simulation
- systems programming

a.k.a. *game engines.* ⇒ highly optimized, resource aware code.

The situation

I work as a systems administrator:

- University's physics student computers.
- $\geq$ 3500 users!
- I'm the "problem solver" there.

My pleasure hacking projects are about:

- realtime graphics
- realtime simulation
- systems programming

a.k.a. *game engines.* $\Rightarrow$ highly optimized, resource aware code.

The situation

I work as a systems administrator:

- University's physics student computers.
- $\geq$ 3500 users!
- I'm the "problem solver" there.

My pleasure hacking projects are about:

- realtime graphics
- realtime simulation
- systems programming

a.k.a. *game engines*. ⇒ highly optimized, resource aware code.

The situation

I work as a systems administrator:

- University's physics student computers.
- $\geq$ 3500 users!
- I'm the "problem solver" there.

My pleasure hacking projects are about:

- realtime graphics
- realtime simulation
- systems programming

a.k.a. *game engines*.$\Rightarrow$ highly optimized, resource aware code.

Linux desktop distributions have become evil!

With each and every new version of OpenSuSE, Ubuntu, Fedora problems got worse.

Most of the problems we encounter are attributed to automatisms.

It’s no longer “set and forget”.

Linux desktop distributions have become evil!

With each and every new version of OpenSuSE, Ubuntu, Fedora problems got worse.

Most of the problems we encounter are attributed to automatisms.

It’s no longer “set and forget”.

Linux desktop distributions have become evil!

With each and every new version of OpenSuSE, Ubuntu, Fedora problems got worse.

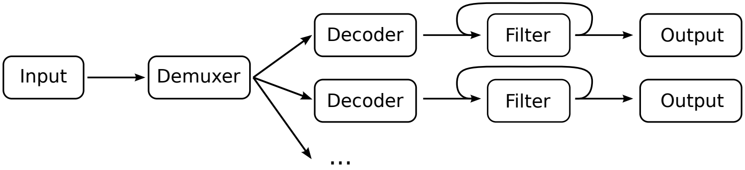
Most of the problems we encounter are attributed to automatisms.

It’s no longer “set and forget”.

Modern Desktops have Multimedia!

Your typical Multimedia Framework

Playback Module Graph



Input

Demuxer

Decoder

Decoder

...

Filter

Filter

Output

Output

file
http
rtmp
rtsp
...

avi
wav
mp4
mkv
ogg
mov
...

MP3
Vorbis
AAC
FLAC
h264
XviD
...

downmix
replaygain
deinterlaceing
brightnes
contrast
...

X11/XVideo
VDPAU
OpenGL
ALSA
PulseAudio
OSS
...

gstreamer

- Provides huge number of modules.
- "Fire and Forget" graph generator included.
- unfortunately not quite stable.

gstreamer

- Provides huge number of modules.
- "Fire and Forget" graph generator included.
- unfortunately not quite stable.

Phonon

- Multimedia-Meta-API – abstraction layer to access different multimedia frameworks through a single API.
- Part of the KDE project
- Builds filter graphs using capabilities of the current backend.
- Designed to allow switching the backend in mid-operation (why?)
- Available backends (Linux)
  - Xine
  - VLC
  - GStreamer (unmantained)
- Filter graph building logic must be provided for every backend!

Phonon

- Multimedia-Meta-API – abstraction layer to access different multimedia frameworks through a single API.
- Part of the KDE project
- Builds filter graphs using capabilities of the current backend.
- Designed to allow switching the backend in mid-operation (why?)
- Available backends (Linux)
  - Xine
  - VLC
  - GStreamer (unmantained)
- Filter graph building logic must be provided for every backend!

Phonon

- Multimedia-Meta-API – abstraction layer to access different multimedia frameworks through a single API.
- Part of the KDE project
- Builds filter graphs using capabilities of the current backend.
- Designed to allow switching the backend in mid-operation (why?)
- Available backends (Linux)
  - Xine
  - VLC
  - GStreamer (unmantained)
- Filter graph building logic must be provided for every backend!

Phonon

- Multimedia-Meta-API – abstraction layer to access different multimedia frameworks through a single API.
- Part of the KDE project
- Builds filter graphs using capabilities of the current backend.
- Designed to allow switching the backend in mid-operation (why?)
- Available backends (Linux)
  - Xine
  - VLC
  - GStreamer (unmantained)
- Filter graph building logic must be provided for every backend!

Phonon

- Multimedia-Meta-API – abstraction layer to access different multimedia frameworks through a single API.
- Part of the KDE project
- Builds filter graphs using capabilities of the current backend.
- Designed to allow switching the backend in mid-operation (why?)
- Available backends (Linux)
  - Xine
  - VLC
  - GStreamer (unmantained)
- Filter graph building logic must be provided for every backend!

PulseAudio

- Designed as a better ESD:
  - mix sound
  - provide audio capture to multiple clients simultanously
  - sound over network (e.g. alongside remote X11)
- Became sort of a media framework of it's own:

> *Things like transferring the audio to a different machine, **changing the sample format or channel count and mixing several sounds** into one are easily achieved using a sound server.*
>
> –[PulseAudio homepage]

PulseAudio

- Designed as a better ESD:
  - mix sound
  - provide audio capture to multiple clients simultanously
  - sound over network (e.g. alongside remote X11)
- Became sort of a media framework of it’s own:

  > *Things like transferring the audio to a different machine, **changing the sample format or channel count and mixing several sounds** into one are easily achieved using a sound server.*
  >
  > –[PulseAudio homepage]

Functionality Matrix

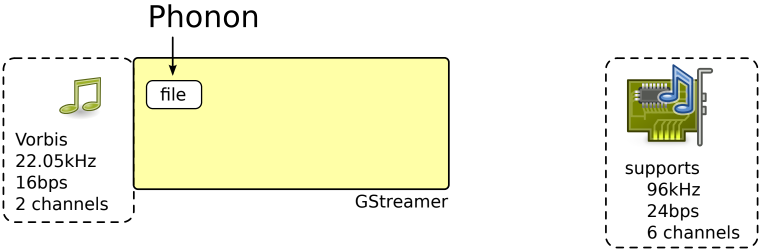
	Phonon	GStreamer	PulseAudio
graph building	✓	✓	
filtering		✓	✓
device access		✓	✓

Let’s hear some music

Vorbis
22.05kHz
16bps
2 channels

supports
96kHz
24bps
6 channels

Let’s hear some music



Phonon

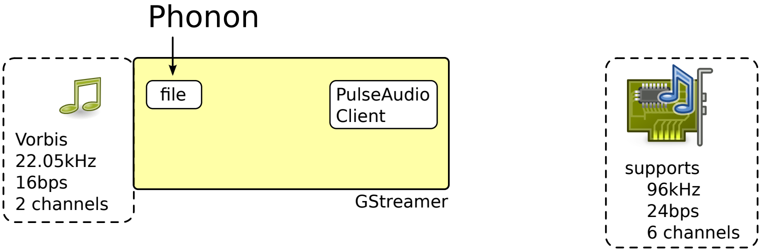
file

GStreamer

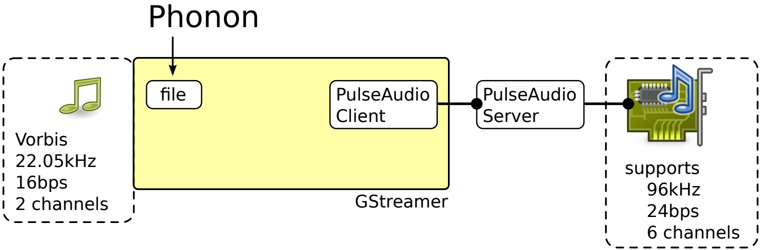
Vorbis
22.05kHz
16bps
2 channels

supports
96kHz
24bps
6 channels

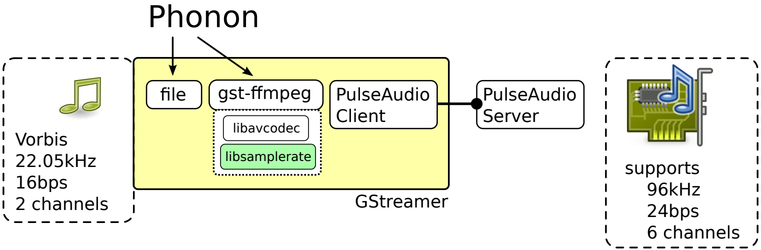
Let’s hear some music



Let’s hear some music



Let’s hear some music



Phonon

file

gst-ffmpeg

libavcodec

libsamplerate

PulseAudio Client

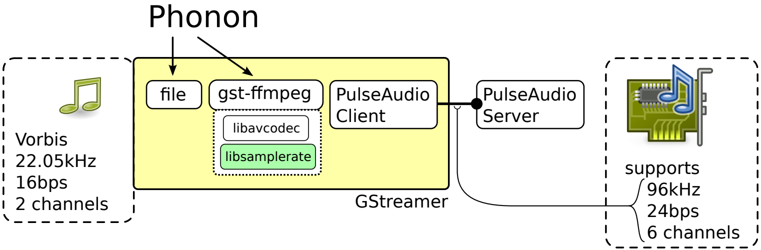
GStreamer

PulseAudio Server

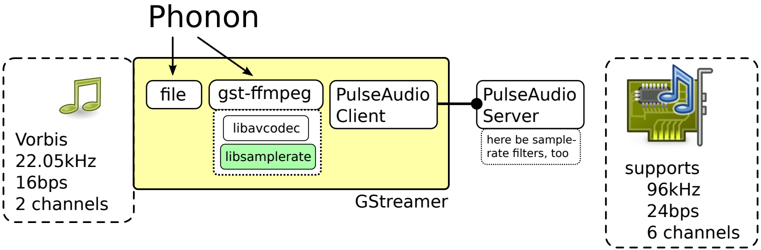
Vorbis
22.05kHz
16bps
2 channels

supports
96kHz
24bps
6 channels

Let’s hear some music



Let’s hear some music



Phonon

Vorbis
22.05kHz
16bps
2 channels

file

gst-ffmpeg

libavcodec

libsamplerate

PulseAudio Client

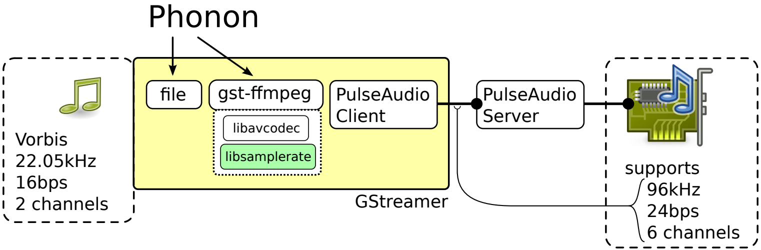
GStreamer

PulseAudio Server

here be sample-rate filters, too

supports
96kHz
24bps
6 channels

Let’s hear some music



Logins Complicated

Tasks of a X Display Manager

- Start X11 server, setup MIT-Cookie (XAUTHORITY)
- Show Greeter, Login Dialog
- (optional) Allow for choosing desktop environment and localization options
- (historically) provide XDMCP – don’t use this nowadays (insecure)

Tasks of a X Display Manager

- Start X11 server, setup MIT-Cookie (XAUTHORITY)
- Show Greeter, Login Dialog
- (optional) Allow for choosing desktop environment and localization options
- (historically) provide XDMCP – don’t use this nowadays (insecure)

User Interaction

- enter username
- enter password
- maybe set session type and localization

All in all a very short experience.
The less interaction, the better.

User Interaction

- enter username
- enter password
- maybe set session type and localization

All in all a very short experience.
The less interaction, the better.

User Interaction

- enter username
- enter password
- maybe set session type and localization

All in all a very short experience.
The less interaction, the better.

User Interaction

- enter username
- enter password
- maybe set session type and localization

All in all a very short experience.
The less interaction, the better.

GDM $\geq$ 2.21

- It’s modal (users tend to mistake it for a screen lock).
- Starts a full blown Gnome session for a simple login.
- Offers less configuration options than older versions.

GDM $\geq$ 2.21

- It’s modal (users tend to mistake it for a screen lock).
- Starts a full blown Gnome session for a simple login.
- Offers less configuration options than older versions.

GDM $\geq$ 2.21

- It’s modal (users tend to mistake it for a screen lock).
- Starts a full blown Gnome session for a simple login.
- Offers less configuration options than older versions.

GDM $\geq$ 2.21

- It’s modal (users tend to mistake it for a screen lock).
- Starts a full blown Gnome session for a simple login.
- Offers less configuration options than older versions.

GDM $\geq$ 2.21 – A Gnome session

```
gdm-binary
    /usr/lib/gdm/gdm-simple-slave
      /usr/bin/X
      /usr/bin/gnome-session
        metacity
        gnome-power-manager
        /usr/lib/gdm/gdm-simple-greeter
      /usr/lib/gdm/gdm-session-worker
  /usr/bin/dbus-launch
  /bin/dbus-daemon
  /usr/lib/libgconf2-4/gconfd-2
  /usr/lib/gnome-settings-daemon/gnome-settings-daemon
  /usr/lib/gvfs/gvfsd
  /usr/bin/pulseaudio
    /usr/lib/pulseaudio/pulse/gconf-helper
```

GDM $\geq$ 2.21 – A Gnome session highlighted

```
gdm-binary
    /usr/lib/gdm/gdm-simple-slave
      /usr/bin/X
      /usr/bin/gnome-session
        metacity
        gnome-power-manager
        /usr/lib/gdm/gdm-simple-greeter
      /usr/lib/gdm/gdm-session-worker
  /usr/bin/dbus-launch
  /bin/dbus-daemon
  /usr/lib/libgconf2-4/gconfd-2
  /usr/lib/gnome-settings-daemon/gnome-settings-daemon
  /usr/lib/gvfs/gvfsd
  /usr/bin/pulseaudio
    /usr/lib/pulseaudio/pulse/gconf-helper
```

GDM $\geq$ 2.21 – Sideshow Dependees

```
gdm-binary
  /usr/lib/gdm/gdm-simple-slave
   /usr/bin/X
   /usr/bin/gnome-session
    metacity
    gnome-power-manager
    /usr/lib/gdm/gdm-simple-greeter
   /usr/lib/gdm/gdm-session-worker
 /usr/bin/dbus-launch
 /bin/dbus-daemon
 /usr/lib/libgconf2-4/gconfd-2
 /usr/lib/gnome-settings-daemon/gnome-settings-daemon
 /usr/lib/gvfs/gvfsd
 /usr/bin/pulseaudio
  /usr/lib/pulseaudio/pulse/gconf-helper
```

GDM $\geq$ 2.21 – Why a Gnome session?

> *By default, GDM is shipped with files which will autostart the gdm-simple-greeter login GUI greeter itself, the gnome-power-manager application, the gnome-settings-daemon, and the metacity window manager. These programs are needed for the **greeter program** to work.*
>
> – [GDM documentation]

After *utmp* and *wtmp*, we proudly present. . .

ConsoleKit

What is ConsoleKit

> *ConsoleKit is a framework for keeping track of the various users, sessions, and seats present on a system. It provides a mechanism for software to react to changes of any of these items or of any the metadata associated with them.*
>
> –[ConsoleKit documentation (2010-12-25)]

Why do we need it?

Defining the Problem

To be written.

Relevant art

To be written.

–[ConsoleKit documentation (2010-12-25)]

`http://www.freedesktop.org/software/ConsoleKit/doc/ConsoleKit.html`

So what does it do?

- It's a Seat aware session manager.
- A Seat:
  - Input Devices
  - Output Devices
  - Permissions per User (Alice may play music, Bob may burn DVDs)
- Tracks the user
- Grants permissions dynamically
- **It uses *D-Bus*!**

So what does it do?

- It's a Seat aware session manager.
- A Seat:
  - Input Devices
  - Output Devices
  - Permissions per User (Alice may play music, Bob may burn DVDs)
- Tracks the user
- Grants permissions dynamically
- **It uses *D-Bus*!**

So what does it do?

- It's a Seat aware session manager.
- A Seat:
  - Input Devices
  - Output Devices
  - Permissions per User (Alice may play music, Bob may burn DVDs)
- Tracks the user
- Grants permissions dynamically
- **It uses *D-Bus*!**

So what does it do?

- It’s a Seat aware session manager.
- A Seat:
  - Input Devices
  - Output Devices
  - Permissions per User (Alice may play music, Bob may burn DVDs)
- Tracks the user
- Grants permissions dynamically
- It uses *D-Bus*!

So what does it do?

- It’s a Seat aware session manager.
- A Seat:
  - Input Devices
  - Output Devices
  - Permissions per User (Alice may play music, Bob may burn DVDs)
- Tracks the user
- Grants permissions dynamically
- **It uses *D-Bus*!**

I’m sorry to tell you, but it’s broken!

- Unix Philosophy: “Something’s either a process, or a file”.
- File permissions and ACLs only applied upon `open`.
- Once you got an FD, permissions and ACL don’t apply anymore.
- **ConsoleKit is easily circumvented**
- Oh, and when it fails, you’re borked.

(Live Demo)

I’m sorry to tell you, but it’s broken!

- Unix Philosophy: “Something’s either a process, or a file”.
- File permissions and ACLs only applied upon `open`.
- Once you got an FD, permissions and ACL don’t apply anymore.
- **ConsoleKit is easily circumvented**
- Oh, and when it fails, you’re borked.

(Live Demo)

I’m sorry to tell you, but it’s broken!

- Unix Philosophy: “Something’s either a process, or a file”.
- File permissions and ACLs only applied upon `open`.
- Once you got an FD, permissions and ACL don’t apply anymore.
- **ConsoleKit is easily circumvented**
- Oh, and when it fails, you’re borked.

(Live Demo)

I’m sorry to tell you, but it’s broken!

- Unix Philosophy: “Something’s either a process, or a file”.
- File permissions and ACLs only applied upon `open`.
- Once you got an FD, permissions and ACL don’t apply anymore.
- **ConsoleKit is easily circumvented**
- Oh, and when it fails, you’re borked.

(Live Demo)

I’m sorry to tell you, but it’s broken!

- Unix Philosophy: “Something’s either a process, or a file”.
- File permissions and ACLs only applied upon `open`.
- Once you got an FD, permissions and ACL don’t apply anymore.
- **ConsoleKit is easily circumvented**
- Oh, and when it fails, you’re borked.

(Live Demo)

My Advice:

Stick with pam_console and groups.

D-Bus

Applications sharing a desktop shall work together.

Several IPC methods over the years

- Inter Client Exchange
- Bonobo/CORBA (Gnome)
- dcop (KDE $\leq$ 3.x)

. . . and some more.

Lightweight things, like music player remote control.

Applications sharing a desktop shall work together.

Several IPC methods over the years

- Inter Client Exchange
- Bonobo/CORBA (Gnome)
- dcop (KDE $\leq$ 3.x)

. . . and some more.

Lightweight things, like music player remote control.

A unified IPC mechanism

D-Bus was originally intended to serve as a unified Desktop IPC.

Was soon expanded to serve as a system wide message passing system.

A unified IPC mechanism

D-Bus was originally intended to serve as a unified Desktop IPC.

Was soon expanded to serve as a system wide message passing system.

So, everything is fine, rainbows and unicorns, right?!

To me, the whole thing doesn’t look right.

So, everything is fine, rainbows and unicorns, right?!

To me, the whole thing doesn’t look right.

Java-esque naming

D-Bus uses names like

- `org.freedesktop.Hal.Manager`
- `/com/mycompany/TextFileManager`

– recommended to use domain name.

Java-esque naming

D-Bus uses names like

- `org.freedesktop.Hal.Manager`
- `/com/mycompany/TextFileManager`

– recommended to use domain name.

Narcistic Namespacing

- Names don't reveal the function
- Without functional grouping each *service* defines it's very own interface
- What if a Name get's changed?
  - Ethereal → Wireshark
  - wxWindows → wxWidgets

Just take a short look at Linux' *SysFS* for an example of usefull namespacing.

Narcistic Namespacing

- Names don’t reveal the function
- Without functional grouping each *service* defines it’s very own interface
- What if a Name get’s changed?
  - Ethereal → Wireshark
  - wxWindows → wxWidgets

Just take a short look at Linux’ *SysFS* for an example of usefull namespacing.

Narcistic Namespacing

- Names don’t reveal the function
- Without functional grouping each *service* defines it’s very own interface
- What if a Name get’s changed?
  - Ethereal → Wireshark
  - wxWindows → wxWidgets

Just take a short look at Linux’ *SysFS* for an example of usefull namespacing.

Narcistic Namespacing

- Names don’t reveal the function
- Without functional grouping each *service* defines it’s very own interface
- What if a Name get’s changed?
    - Ethereal → Wireshark
    - wxWindows → wxWidgets

Just take a short look at Linux’ *SysFS* for an example of usefull namespacing.

No *transparent* networking

Yes, D-Bus has TCP transport, but:

- no authentication
- no authorization
- no encryption

Srsly? A network transport no older than 5 years, without any means for security? And it's quite a mess to get to work nevertheless.

No *transparent* networking

Yes, D-Bus has TCP transport, but:

- no authentication
- no authorization
- no encryption

Srsly? A network transport no older than 5 years, without any means for security?

And it's quite a mess to get to work nevertheless.

No *transparent* networking

Yes, D-Bus has TCP transport, but:

- no authentication
- no authorization
- no encryption

Srsly? A network transport no older than 5 years, without any means for security?
And it’s quite a mess to get to work nevertheless.

Must be setup additionall to X11

- session bus is independent from X11
- ⇒ every GUI program has to do multiple bookkeeping
  - X11
  - D-Bus
- `ssh -X ...`, what about that?

Nothing particularily difficult to implement, but
that would add complexity, for only little gain.

Must be setup additionall to X11

- session bus is independent from X11
- ⇒ every GUI program has to do multiple bookkeeping
  - X11
  - D-Bus
- `ssh -X ...`, what about that?

  Nothing particularily difficult to implement, but that would add complexity, for only little gain.

Must be setup additionall to X11

- session bus is independent from X11
- ⇒ every GUI program has to do multiple bookkeeping
  - X11
  - D-Bus
- `ssh -X ...`, what about that?

  Nothing particularily difficult to implement, but that would add complexity, for only little gain.

Must be setup additionall to X11

- session bus is independent from X11
- ⇒ every GUI program has to do multiple bookkeeping
  - X11
  - D-Bus
- `ssh -X ...`, what about that?

Nothing particularily difficult to implement, but
that would add complexity, for only little gain.

Must be setup additionall to X11

- session bus is independent from X11
- ⇒ every GUI program has to do multiple bookkeeping
  - X11
  - D-Bus
- `ssh -X ...`, what about that?

  Nothing particularily difficult to implement, but that would add complexity, for only little gain.

D-Bus is FreeDesktop’s Hammer

Each and everything done by FreeDesktop
is tied to D-Bus somehow.

Even things where D-Bus makes no sense.

Case in Point: *Status Notifier Items* You know, SysTray.

D-Bus is FreeDesktop’s Hammer

Each and everything done by FreeDesktop is tied to D-Bus somehow.

Even things where D-Bus makes no sense.

Case in Point: *Status Notifier Items* You know, SysTray.

D-Bus is FreeDesktop’s Hammer

Each and everything done by FreeDesktop
is tied to D-Bus somehow.

Even things where D-Bus makes no sense.

Case in Point: *Status Notifier Items* You know, SysTray.

D-Bus is FreeDesktop's Hammer

Each and everything done by FreeDesktop
is tied to D-Bus somehow.

Even things where D-Bus makes no sense.

Case in Point: *Status Notifier Items* You know, SysTray.

XSystemTray

- Old method: SysTray is a special kind of sub-window manager.
- Each item a own X11 window ⇒ one could use everything X11 provides to draw it – serverside. (GPU acceleration FTW)
- It works for every X11 client, independent of host, transport and connection.

Status Notifier

- Status Notifier uses D-Bus for transport, graphical items are transported as raw pixmaps or SVG. (dynamic updates?)
- Status Notifier only available to programs having access to the D-Bus (remember, remote X11 vs. D-Bus).

If you care about common look and feel: Define user interface guidelies, provide a common library.

That’s actually done by GTK+ and Qt (the library thing).

If you care about common look and feel: Define user interface guidelies, provide a common library.

That’s actually done by GTK+ and Qt (the library thing).

That horse can carry only so much.

D-Bus doesn’t scale!

There’s actually been made the suggestion to give Linux a new special D-Bus socket type, to overcome routing bottlenecks.

That horse can carry only so much.

D-Bus doesn't scale!

There's actually been made the suggestion to give Linux a new special D-Bus socket type, to overcome routing bottlenecks.

There are better tools

Instead of D-Bus we could use *IPv6 * Local Multicast*.

- scales well
- can be versatilely routed (address rewriting)
- cryptographic batteries included (IPv6 mandates IPSec – Unicast)
- no single point of failure (D-Bus daemon) – well, the kernel may crash, but then you've got other problems.

This idea courtesy by Fefe.

There are better tools

Instead of D-Bus we could use *IPv6 * Local Multicast*.

- scales well
- can be versatilely routed (address rewriting)
- cryptographic batteries included (IPv6 mandates IPSec – Unicast)
- no single point of failure (D-Bus daemon) – well, the kernel may crash, but then you’ve got other problems.

This idea courtesy by Fefe.

There are better tools

Instead of D-Bus we could use *IPv6 * Local Multicast*.

- scales well
- can be versatilely routed (address rewriting)
- cryptographic batteries included (IPv6 mandates IPSec – Unicast)
- no single point of failure (D-Bus daemon) – well, the kernel may crash, but then you've got other problems.

This idea courtesy by Fefe.

There are better tools

Instead of D-Bus we could use *IPv6 * Local Multicast*.

- scales well
- can be versatilely routed (address rewriting)
- cryptographic batteries included (IPv6 mandates IPSec – Unicast)
- no single point of failure (D-Bus daemon) – well, the kernel may crash, but then you've got other problems.

This idea courtesy by Fefe.

There are better tools

Instead of D-Bus we could use *IPv6 * Local Multicast*.

- scales well
- can be versatilely routed (address rewriting)
- cryptographic batteries included (IPv6 mandates IPSec – Unicast)
- no single point of failure (D-Bus daemon) – well, the kernel may crash, but then you've got other problems.

This idea courtesy by Fefe.

There are better tools

Instead of D-Bus we could use *IPv6 * Local Multicast*.

- scales well
- can be versatilely routed (address rewriting)
- cryptographic batteries included (IPv6 mandates IPSec – Unicast)
- no single point of failure (D-Bus daemon) – well, the kernel may crash, but then you've got other problems.

This idea courtesy by Fefe.

There are better tools

Instead of D-Bus we could use *IPv6 * Local Multicast*.

- scales well
- can be versatilely routed (address rewriting)
- cryptographic batteries included (IPv6 mandates IPSec – Unicast)
- no single point of failure (D-Bus daemon) – well, the kernel may crash, but then you've got other problems.

This idea courtesy by Fefe.

PolicyKit

What is PolicyKit

> *PolicyKit is an application-level toolkit for defining and handling the policy that allows unprivileged processes to speak to privileged processes: It is a framework for centralizing the decision making process with respect to granting access to privileged operations for unprivileged applications. PolicyKit is specifically targeting applications in rich desktop environments on multi-user UNIX-like operating systems.*
>
> –[PolicyKit homepage]

PolicyKit

- Oftenly compared to *sudo*
  - *sudo* escalates
  - *PolicyKit* authorizes
- Uses D-Bus. . .

Authorizing means

- A program capable of privileged action is commaned to perform a task.
- Before this task is performed, PolicyKit is used to ask the user for permission
  - If the user itself has no permission ⇒ Deny
  - If the user authenticates the action ⇒ Execute it.
- ⇒ The privileged programm is running all the time, or started by *pkexec*

To me this sounds prone to logic errors on the privileged side..
Could we attack the privileged program through the action request?

Authorizing means

- A program capable of privileged action is commaned to perform a task.
- Before this task is performed, PolicyKit is used to ask the user for permission
  - If the user itself has no permission ⇒ Deny
  - If the user authenticates the action ⇒ Execute it.
- ⇒ The privileged programm is running all the time, or started by *pkexec*

To me this sounds prone to logic errors on the privileged side..
Could we attack the privileged program through the action request?

Authorizing means

- A program capable of privileged action is commaned to perform a task.
- Before this task is performed, PolicyKit is used to ask the user for permission
- If the user itself has no permission ⇒ Deny
- If the user authenticates the action ⇒ Execute it.
- ⇒ The privileged programm is running all the time, or started by *pkexec*

To me this sounds prone to logic errors on the privileged side..
Could we attack the privileged program through the action request?

Asking per task is a bad idea anyway

The whole thing is much like Windows UAC: The user gets nagged about authorizing this and that everytime.

Entering privileged realms itself should be protected.

Privileged stuff should not be required to be set so oftenly, that a convenient way to ask the user is required at all.

Asking per task is a bad idea anyway

The whole thing is much like Windows UAC: The user gets nagged about authorizing this and that everytime.

Entering privileged realms itself should be protected.

Privileged stuff should not be required to be set so oftenly, that a convenient way to ask the user is required at all.

Asking per task is a bad idea anyway

The whole thing is much like Windows UAC: The user gets nagged about authorizing this and that everytime.

Entering privileged realms itself should be protected.

Privileged stuff should not be required to be set so oftenly, that a convenient way to ask the user is required at all.

Automatisms

$\neq$

Things Just Work

NetworkManager

I think I invented it, or at least came up with that idea:
`http://forums.gentoo.org/viewtopic-t-163808-highlight-.html`

***Looking for program**... that is automatically setting the network interfaces, depending on the devices connected to. E.g. I'd like to configure my eth0 connection to either DHCP if it find's a certain host via MAC or to a static IP if it detects another host. Also I need something similair for WLAN, depending on the found ESSID and/or the strongest signal.*

Also it should work as a daemon, so that it a physical connection gets lost automatically the route tables and resolv.conf are adjusted, and vice versa. –[I in Gentoo forums 2004-04-20]

Sorry about that

Today's situation

- Either you're constantly roaming networks, then the network should provide the configuration and you don't care.
- Or your system is statically bound to a certain network, but then a user must not change anything.
- GSM/UMTS/LTE? Similary: About every 3G modem can be configured to act as a network interface. The rest, see above.

Sorry about that

Today's situation

- Either you're constantly roaming networks, then the network should provide the configuration and you don't care.
- Or your system is statically bound to a certain network, but then a user must not change anything.
- GSM/UMTS/LTE? Similary: About every 3G modem can be configured to act as a network interface. The rest, see above.

Sorry about that

Today's situation

- Either you're constantly roaming networks, then the network should provide the configuration and you don't care.
- Or your system is statically bound to a certain network, but then a user must not change anything.
- GSM/UMTS/LTE? Similary: About every 3G modem can be configured to act as a network interface. The rest, see above.

Sorry about that

Today's situation

- Either you're constantly roaming networks, then the network should provide the configuration and you don't care.
- Or your system is statically bound to a certain network, but then a user must not change anything.
- GSM/UMTS/LTE? Similary: About every 3G modem can be configured to act as a network interface. The rest, see above.

Ubuntu Desktop + NetworkManager

Your network connection will only come up,
after you log on. WTF?! . . . can be configured otherweise.

This doesn’t *just work*.

Ubuntu Desktop + NetworkManager

Your network connection will only come up,
after you log on. WTF?! . . . can be configured otherweise.

This doesn't *just work*.

Ubuntu Desktop + NetworkManager

Your network connection will only come up,
after you log on. WTF?! . . . can be configured otherweise.

This doesn’t *just work*.

Removeable Storage Media

USB Thumb drive get’s plugged in:

Many methods so far:

- automounters (until ca. 2002)
- fstab adjusters (I still prefer this)
- ivman (ca. 2004)
- pmount
- hal-mount
- **Currently: UDisks**

Removeable Storage Media

USB Thumb drive get’s plugged in:

Many methods so far:

- automounters (until ca. 2002)
- fstab adjusters (I still prefer this)
- ivman (ca. 2004)
- pmount
- hal-mount
- **Currently: UDisks**

None of these tackles the problem itself

It boils down to:

- A storage medium must be mounted to be accessible (easy)
- After its use it must be cleanly synched and unmounted before disconnecting, otherwise data is lost (hard).

Users don’t really understand about the need for synching/unmounting, they did click the “Save” button, so why’d not saved yet?

I understand my audience, or at least the majority understand the problem though, right?

`mount -o sync` not such a good solution, either.

None of these tackles the problem itself

It boils down to:

- A storage medium must be mounted to be accessible (easy)
- After its use it must be cleanly synched and unmounted before disconnecting, otherwise data is lost (hard).

Users don’t really understand about the need for synching/unmounting, they did click the “Save” button, so why’d not saved yet?

I understand my audience, or at least the majority understand the problem though, right?

`mount -o sync` not such a good solution, either.

None of these tackles the problem itself

It boils down to:

- A storage medium must be mounted to be accessible (easy)
- After its use it must be cleanly synched and unmounted before disconnecting, otherwise data is lost (hard).

Users don’t really understand about the need for synching/unmounting, they did click the “Save” button, so why’d not saved yet?

I understand my audience, or at least the majority understand the problem though, right?

`mount -o sync` not such a good solution, either.

None of these tackles the problem itself

It boils down to:

- A storage medium must be mounted to be accessible (easy)
- After its use it must be cleanly synched and unmounted before disconnecting, otherwise data is lost (hard).

Users don't really understand about the need for synching/unmounting, they did click the "Save" button, so why'd not saved yet?

I understand my audience, or at least the majority understand the problem though, right?

`mount -o sync` not such a good solution, either.

I don’t know of any good solution either.

But just providing nicer looking buttons won’t help.

Maybe this problem will silently go away? Everything stored in the Cloud . . . → has it’s own wealth of problems. Discussed on this congress.

I don’t know of any good solution either.

But just providing nicer looking buttons won’t help.

Maybe this problem will silently go away? Everything stored in the Cloud . . . → has it’s own wealth of problems. Discussed on this congress.

I don’t know of any good solution either.

But just providing nicer looking buttons won’t help.

Maybe this problem will silently go away? Everything stored in the Cloud . . . → has it’s own wealth of problems. Discussed on this congress.

I don’t know of any good solution either.

But just providing nicer looking buttons won’t help.

Maybe this problem will silently go away? Everything stored in the Cloud . . . → has it’s own wealth of problems. Discussed on this congress.

I don’t know of any good solution either.

But just providing nicer looking buttons won’t help.

Maybe this problem will silently go away? Everything stored in the Cloud . . . → has it’s own wealth of problems. Discussed on this congress.

One API to configure them all. . .

GConf

- Daemon and library providing unified interface to configuration data.
- Hierachical, key structured database
- Open to various storage backends, but so far
  - keys structured by directories
  - values in XML files (may also contain keys)
- Single point of failure
- Much like the Windows registry

GConf

- Daemon and library providing unified interface to configuration data.
- Hierachical, key structured database
- Open to various storage backends, but so far
  - keys structured by directories
  - values in XML files (may also contain keys)
- Single point of failure
- Much like the Windows registry

GConf

- Daemon and library providing unified interface to configuration data.
- Hierachical, key structured database
- Open to various storage backends, but so far
  - keys structured by directories
  - values in XML files (may also contain keys)
- Single point of failure
- Much like the Windows registry

XSettings

- X11 centric configuration system
- Colours, Mouse Pointers
- Input devices bahaviour

. . . eh, don’t we have Xrm for that?

XSettings

- X11 centric configuration system
- Colours, Mouse Pointers
- Input devices bahaviour

. . . eh, don’t we have Xrm for that?

What are the claimed problems of Xrm?

- All settings in one single property of the root window.
- No fine grained access to settings
- Changes to settings not easily detectible
- Large amount of data to process just to retrieve a very small subset from it.

Proposal of XSettings

- Settings managed by a XSettings daemon, providing a (invisible) settings window (remember, single point of failure).
- Serial numbers to identify changed settings
- Data stored in binary format, with no endianess enforced

  – *lolwut?* Sounds like fun:

  - Integer overflows
  - Buffer overruns
  - Shellcode injection

Proposal of XSettings

- Settings managed by a XSettings daemon, providing a (invisible) settings window (remember, single point of failure).
- Serial numbers to identify changed settings
- Data stored in binary format, with no endianess enforced – *lolwut?* Sounds like fun:
- Integer overflows
- Buffer overruns
- Shellcode injection

Do these people suffer from schizophrenia?

> *The Xrm database stores all information in a single text property on the root window. This makes it difficult to determine what settings have changed; it is necessary to parse the property and do string comparisons.*

And later on in the very same document:

> ***Why use a single property for all settings?***
>
> *Using a single property has several advantages. First, retrieving all settings takes only a single round-trip to the server instead of a round-trip for each settings. Second, it means that when multiple settings can be changed at once, only a single notification is received by clients, and clients will see interrelated properties changed in an atomic fashion.*

Do these people suffer from schizophrenia?

> *The Xrm database stores all information in a single text property on the root window. This makes it difficult to determine what settings have changed; it is necessary to parse the property and do string comparisons.*

And later on in the very same document:

> ***Why use a single property for all settings?***
>
> *Using a single property has several advantages. First, retrieving all settings takes only a single round-trip to the server instead of a round-trip for each settings. Second, it means that when multiple settings can be changed at once, only a single notification is received by clients, and clients will see interrelated properties changed in an atomic fashion.*

Zombies

. . . aim for the head.

HAL

Hardware Abstraction Layer

- A better backronym would be *Hardware Annotation Library*.
- Huge crapload of unreadable and unmaintainable XML files.
- **Officially deprecated!**
- Though still in use by some Distros – (*aim for the*. . ., well, you know what to do).

HAL

Hardware Abstraction Layer

- A better backronym would be *Hardware Annotation Library*.
- Huge crapload of unreadable and unmaintainable XML files.
- **Officially deprecated!**
- Though still in use by some Distros – (*aim for the*. . ., well, you know what to do).

HAL

Hardware Abstraction Layer

- A better backronym would be *Hardware Annotation Library*.
- Huge crapload of unreadable and unmaintainable XML files.
- **Officially deprecated!**
- Though still in use by some Distros – (*aim for the*. . ., well, you know what to do).

HAL

Hardware Abstraction Layer

- A better backronym would be *Hardware Annotation Library*.
- Huge crapload of unreadable and unmaintainable XML files.
- **Officially deprecated!**
- Though still in use by some Distros – (*aim for the. . .*, well, you know what to do).

HAL

Hardware Abstraction Layer

- A better backronym would be *Hardware Annotation Library*.
- Huge crapload of unreadable and unmaintainable XML files.
- **Officially deprecated!**
- Though still in use by some Distros
  – (*aim for the*. . ., well, you know what to do).

HAL

Hardware Abstraction Layer

- A better backronym would be *Hardware Annotation Library*.
- Huge crapload of unreadable and unmaintainable XML files.
- **Officially deprecated!**
- Though still in use by some Distros – (*aim for the*. . ., well, you know what to do).

I don’t want all this crap

In a organization’s network

- central software distribution
- central configuration
- users have no privileges at all
- custom terminal access solutions (provide access to localy mounted media on remotely accessed machine)

I, as an administrator, want the full control over my stuff.

In a organization’s network

- central software distribution
- central configuration
- users have no privileges at all
- custom terminal access solutions (provide access to localy mounted media on remotely accessed machine)

I, as an administrator, want the full control over my stuff.

You’ll end up creating your own distribution – or use Gentoo

- Customly compiled Desktops
- Alternate package sources, patched packages
- Also requires maintaining a custom configuration system

See your carefully crafted configurations break

So we were testing Ubuntu 9.04. . .

- University maintains a central authentication database for all students and employees
  - User Database accessed by LDAP/Active Directory
  - Kerberos-5 for authentication
- A carefully maintained set of Kerberos-5, LDAP nsswitch and PAM config files is provided
- Some of our older maintenance tools require SSH root access by public key, and only if from our IP range – yes, we know, you don't do this, but this is like using Duct Tape, it somehow works and then lasts.

The system passes all automated security tests.

See your carefully crafted configurations break

So we were testing Ubuntu 9.04. . .

- University maintains a central authentication database for all students and employees
  - User Database accessed by LDAP/Active Directory
  - Kerberos-5 for authentication
- A carefully maintained set of Kerberos-5, LDAP nsswitch and PAM config files is provided
- Some of our older maintenance tools require SSH root access by public key, and only if from our IP range – yes, we know, you don't do this, but this is like using Duct Tape, it somehow works and then lasts.

The system passes all automated security tests.

See your carefully crafted configurations break

So we were testing Ubuntu 9.04. . .

- University maintains a central authentication database for all students and employees
  - User Database accessed by LDAP/Active Directory
  - Kerberos-5 for authentication
- A carefully maintained set of Kerberos-5, LDAP nsswitch and PAM config files is provided
- Some of our older maintenance tools require SSH root access by public key, and only if from our IP range – yes, we know, you don't do this, but this is like using Duct Tape, it somehow works and then lasts.

The system passes all automated security tests.

See your carefully crafted configurations break

So we were testing Ubuntu 9.04. . .

- University maintains a central authentication database for all students and employees
  - User Database accessed by LDAP/Active Directory
  - Kerberos-5 for authentication
- A carefully maintained set of Kerberos-5, LDAP nsswitch and PAM config files is provided
- Some of our older maintenance tools require SSH root access by public key, and only if from our IP range – yes, we know, you don't do this, but this is like using Duct Tape, it somehow works and then lasts.

The system passes all automated security tests.

So what's the problem, then?

Well,

- ConsoleKit + PolicyKit have a set of own PAM rules installed
- These rules plus those of our Kerberos-5 auth plus the config for root-SSH were a bit unlucky

⇒root could SSH into those boxes without requiring a password, or a public key, but only if not from our IP range. Only good thing was: root doesn't get Kerberos tokens in our system, so no harm outside those test machines.

So what’s the problem, then?

Well,

- ConsoleKit + PolicyKit have a set of own PAM rules installed
- These rules plus those of our Kerberos-5 auth plus the config for root-SSH were a bit unlucky

⇒root could SSH into those boxes without requiring a password, or a public key, but only if not from our IP range. Only good thing was: root doesn’t get Kerberos tokens in our system, so no harm outside those test machines.

So what's the problem, then?

Well,

- ConsoleKit + PolicyKit have a set of own PAM rules installed
- These rules plus those of our Kerberos-5 auth plus the config for root-SSH were a bit unlucky

⇒root could SSH into those boxes without requiring a password, or a public key, but only if not from our IP range. Only good thing was: root doesn't get Kerberos tokens in our system, so no harm outside those test machines.

So what’s the problem, then?

Well,

- ConsoleKit + PolicyKit have a set of own PAM rules installed
- These rules plus those of our Kerberos-5 auth plus the config for root-SSH were a bit unlucky

⇒root could SSH into those boxes without requiring a password, or a public key, but only if not from our IP range. Only good thing was: root doesn’t get Kerberos tokens in our system, so no harm outside those test machines.

Morale

- Yes, it was a configuration error.
- But to set proper configurations one needs good documentation – **for sysadmins.**
- Distributions don't properly document their inner workings. **This must change.**
- Those convoluted interdependencies of current desktop systems do no good.

Morale

- Yes, it was a configuration error.
- But to set proper configurations one needs good documentation **– for sysadmins.**
- Distributions don't properly document their inner workings. **This must change.**
- Those convoluted interdependencies of current desktop systems do no good.

Morale

- Yes, it was a configuration error.
- But to set proper configurations one needs good documentation **– for sysadmins.**
- Distributions don't properly document their inner workings. **This must change.**
- Those convoluted interdependencies of current desktop systems do no good.

Morale

- Yes, it was a configuration error.
- But to set proper configurations one needs good documentation **– for sysadmins.**
- Distributions don’t properly document their inner workings. **This must change.**
- Those convoluted interdependencies of current desktop systems do no good.

We've seen only the tip of the iceberg so far.
There's a lot more to consider:

- Modern Unix Desktops depend on a number of system level services
- Some of these services aim at replacing core functionality, not even related to desktops
  - systemd (replaces SysV init, upstart, the like)
  - RealtimeKit (a whole story of its own).
- The more direct dependencies are created down to the system level, the harder it get's to install alternatives there.
- Eventually the whole development process may be only about fixing issues – probably by adding complexity instead of removing – and come to a standstill.

We’ve seen only the tip of the iceberg so far.
There’s a lot more to consider:

- Modern Unix Desktops depend on a number of system level services
- Some of these services aim at replacing core functionality, not even related to desktops
  - systemd (replaces SysV init, upstart, the like)
  - RealtimeKit (a whole story of its own).
- The more direct dependencies are created down to the system level, the harder it get’s to install alternatives there.
- Eventually the whole development process may be only about fixing issues – probably by adding complexity instead of removing – and come to a standstill.

Large, complicated set of immutable dependencies =

We’re getting **locked in!**

Conclusion

Conclusion

Fallacies of contemporary desktop development:

- Errection of huge and complex structures
- Features given more weight than simplicity and stability
- Problems oftenly not properly identified
- Problems tackled by throwing even more code at them, instead of fixing proper cause.

Quotes of famous people

Simplicity is the highest form of sophistication.
– unattributed (Leonardo da Vinci?)

Complexity has nothing to do with intelligence,
simplicity does.
– Larry Bossidy

Make things as simple as possible – but not simpler.
– Albert Einstein

Those who don't understand Unix
are doomed to reinvent it, poorly.
– Henry Spencer