Debian is awesome to use in a 1000+ machines environment
- Automated deployment tools
- Customization: custom APT repositories
- Administration tools, and our famous reliability!

Workstations are a good use case, with GNOME as the desktop
- The easy way: leave users with self-administration permissions
  → But it doesn’t scale very well in terms of support
- The secure way: standard workstations with no specific permissions

In order to ship the best systems for users:
- How does GNOME actually work on the inside?
- Where are important places to look for a configuration / a problem?
- What can I tweak on my systems?
OUTLINE

1. The base plumbing for the desktop
   DBus, PolicyKit

2. Systemd services
   logind, journald…

3. User settings
   GSettings and dconf
   Menus and applications

4. Login and password management
   The GNOME display manager
   Accountsservice
   The keyring

5. Networking with GNOME
   NetworkManager
   The virtual filesystem stack

6. Hardware access
   PulseAudio
   Printing
   Power management

7. Miscellanea
   PackageKit
   Using the plumbing in custom scripts
   Deploying the configuration on workstations
GNOME 2.30 (squeeze)

GNOME 3.4 (wheezy)

GNOME 3.14 (jessie)
D-Bus is the basis for inter-process communications between GNOME applications and the underlying system

- Based on a typed messaging system over Unix sockets
- Implements an asynchronous RPC mechanism

Services can either

- Start by themselves and register a name, e.g. org.freedesktop.NetworkManager → systemd handles the case with Type=dbus
- Be auto-spawned by the DBus daemon
  → /usr/share/dbus-1/services/*.service
  → /usr/share/dbus-1/system-services/*.service

Basic permissions management for system services in /etc/dbus-1/*.conf

Most relevant daemons use PolicyKit instead
Examining your system with D-Feet
PolicyKit

- PolicyKit adds rich permissions management to a system D-Bus service
  - Can wrap any D-Bus call, invisible from the application

![Diagram showing the interaction between Application, PolicyKit, and Wrapped service]

User PolicyKit agent
- gnome-shell registers to org.freedesktop.PolicyKit1

Authentication required
- Is this user active?
  - Logind see later

Default policy
- /usr/share/polkit-1/actions/*.policy
- /etc/polkit-1

Which password is asked?
The root password or the current user’s?

- It depends on the configuration:
  - /etc/polkit-1/localauthority.conf.d
  - AdminIdentities=unix-group:admins;unix-user:joe

Debian default: the sudo group
Tuning the default policy

- Policy tuning is done either with JavaScript files or PKLA (ini-like) files → Depending on the distribution choices
- Debian uses PKLA. You can create `/etc/polkit-1/localauthority/30-site.d/my-config.pkla`

  - [Allow users to shutdown, even when someone else’s application asks not to]
    Identity=*  
    Action=org.freedesktop.login1.power-off-ignore-inhibit  
    ResultAny=no  
    ResultInactive=no  
    ResultActive=yes

  - [Let some users change the CPU frequency by hand]
    Identity=unix-group:benchmarks  
    Action=org.gnome.CPUFreqSelector  
    ResultAny=no  
    ResultInactive=no  
    ResultActive=yes

  - [Let a user install any package from the repository using PackageKit]
    Identity=unix-user:joss  
    Action=org.freedesktop.packagekit.package-install  
    ResultAny=no  
    ResultInactive=no  
    ResultActive=auth_self
Systemd services: logind

- Logind is the daemon that brings **reliable session management** on top of the existing kernel and system infrastructure.
  - Manages **seats** and their mapping with hardware components
  - Tells which session is active on which VT and which seat
    → Try the CLI interface: `logindctl`
  - Tells which session a process belongs to (using systemd cgroups)
  - Manages device permissions (see `/lib/udev/rules.d/70-uaccess.rules`)
    → Sets permissions dynamically on a number of devices like `/dev/snd/*`
    → Most specific groups (audio, video, netdev…) are obsolete.
Systemd services: the journal

- **adduser** joe systemd-journal
  → **gnome-logs**

- **systemd** (PID 1)
  → **cgroups** (kernel)
  → **journald**
  → **rsyslog**
  → **journald protocol**
  → **identify services**

- **system services**
  → **syslog**
  → **standard output/error**

- **User applications**
  → **GDM**
  → **systemd**
  → **cgroups (kernel)**

- **GDM**

- **User applications**

- **Important**
  - **udisksd**
    Error probing device: Error sending ATA command IDENTIFY PACKET DEVICE to /dev/sr0: ATA command f
  - **pulseaudio**
    Nous avons été réveillés avec POLLOUT actif, cependant un snd_pcm_avail() ultérieux a retourné 0 ...
  - **pulseaudio**
    Il s'agit très probablement d'un bogue dans le pilote ALSA « snd_ens1371 ». Veuillez rapporter ce ...
  - **pulseaudio**
    ALSA nous a réveillé pour écrire de nouvelles données à partir du périphérique, mais il n'y avait ...
  - **gdm-session-wor**
    pam_systemd(gdm-launch-environment:session): Failed to release session: Appel système interrompu
  - **minissdpd**
    setsockopt(udp, IP_ADD_MEMBERSHIP)(0.0.0.0): No such device
Other systemd services

- **Timedated and timesyncd**
  - Sets date/time
  - Switches time zones
  - Enables NTP support (systemd-timesyncd)

- **Hostnamed**
  - Sets the host name

- **Localed**
  - Sets the default system locale
  - Not directly used by GNOME (see later accountsservice)

- All of them are accessed using simple D-Bus services with PolicyKit authentication
User settings in GNOME 3.x: GSettings

- Schemas, defaults and overrides are managed by the client
- Dconf is optimized for speed: direct reads, binary database (GVDB)
- Changing a user setting:
  - `gsettings set org.gnome.desktop.sound event-sounds false`
- Listing all settings:
  - `gsettings list-recursively org.gnome.nautilus`
- There is also dconf-editor
Tuning GSettings in a package

- Ship an override file in `debian/package.gsettings-override`
  - `dh_installgsettings --priority=90`

  - `# Custom background
    [org.gnome.desktop.background]
    picture-options='zoom'
    picture-uri='file:///my/nice/picture.svg'
    
  - `# Squeeze-like icons on the desktop
    [org.gnome.desktop.background]
    show-desktop-icons=true``

  - `# I haz a theme
    [org.gnome.desktop.interface]
    gtk-theme='FabulousTheme'
    icon-theme='Wonderfullcons'
    [org.gnome.desktop.wm.preferences]
    theme='CoolBorders'

  - `# Default applications and extensions in the shell
    [org.gnome.shell]
    favorite-apps=['evolution.desktop', 'libreoffice-impress.desktop', '...']
    enabled-extensions=['apps-menu@gnome-shell-extensions.gcampax.github.com']

You can also use XML files for evolving backgrounds or multiple resolutions

The GTK theme needs to have the same name for GTK+ 2.0 and 3.0
Dconf: default and mandatory system settings

- Configure a system database: `/etc/dconf/profile`
  - user-db:user
  - system-db:local

- Default settings then go in `/etc/dconf/db/local.d/00_my_defaults`
  
  # Those users are too dumb, don’t let them do anything
  [org/gnome/desktop/lockdown]
  disable-applications-handlers=true
  disable-log-out=true
  disable-print-setup=true
  ...

- Make those defaults mandatory with `locks`: `/etc/dconf/db/local.d/locks/my_locks`
  
  /org/gnome/desktop/lockdown/disable-applications-handlers
  /org/gnome/desktop/lockdown/disable-log-out
  /org/gnome/desktop/lockdown/disable-print-setup
  ...

- To **update the database**: `dconf update`

Separator for dconf is `/` (instead of `. for GSettings`)
Menus and applications

- Available applications are described in .desktop files
  - **MimeTypes** describe file types the application can open
  - **Virtual x-uri-scheme/* MIME types** describe applications which can open URIs
- Found in /usr/share/applications
  - Overridden with $XDG_DATA_DIRS and ~/.local/share/applications
- Default MIME associations in Debian: /usr/share/gnome/applications/defaults.list
  - Overridden the same
- Adding/removing MIME associations: datadir/mimeapps.list
- Default menu (XDG standard): /etc/xdg/menus/gnome-applications.menu
  - Applications are affected in submenus using their Categories
  - Adding new sub-menus: /etc/xdg/menus/applications-merged/my-menu.menu
GDM: the display manager

- GNOME shell uses the same code:
  - in the login screen (minimal login session)
  - in the lock screen (formerly screensaver)

- Displays are started and closed dynamically
Configuring GDM

- Daemon configuration: `/etc/gdm3/daemon.conf` (Debian-specific)
  - Enabling autologin, debugging, VT configuration…
  - XDMCP

- The real configuration for the minimal session (Debian-specific)
  - `/etc/gdm3/greeter.gsettings` (GSettings format)
  - In a package: `/usr/share/gdm/dconf/50-my-settings` (DConf format)
    + `invoke-rc.d gdm3 reload`

AccountsService

- User defaults:
  - language, icon, selected session

  - Storage: `/var/lib/AccountsService`

  - Also provides a D-Bus interface to create and configure accounts
    → Used by the control center
Storing secrets: the GNOME keyring

- Keeps user secrets in AES-encrypted files
  - Several *keyrings*, each with its own password
  - Also acts as GnuPG and SSH agent
  - Special case: the *login keyring* uses the login password

- User interface: *seahorse*
  - Access user keys and passwords
  - *pam_gnome_keyring* also acts when **changing the password**
  - Infrastructure constraint: password change is on the same machine
The Network-Manager infrastructure

- **System connections**: started at boot time
  - Controlled by users with appropriate permissions (PolicyKit)
  - Preconfigured by the sysadmin

- **User connections**: started at login time / on-the-fly
  - Secrets stored securely in the keyring
  - Fast user switching: drops the connection (either wanted or buggy behavior)

- **System connections with user secrets**: e.g. 802.1x (WPA2 enterprise, NAC…)

---

Network-Manager agent: GNOME Shell

- GNOME keyring
- User secrets

Network-Manager daemon

GNOME control center

Kernel (netlink)

System and user connections (.ini-like files)
/etc/network-manager/system-connections

Secrets for system connections are stored directly here
Configuring system connections

- Real example: deploy TLS 802.1x authentication over your Ethernet network with a per-machine certificate users don’t know
  
- /etc/network-manager/system-connections/authenticated
  
- Other uses:
  - Pre-configuring Wi-Fi with a shared key
  - Pre-configured WPA2 enterprise using 802.1x with per-user credentials
  - Pre-configured VPN connection with per-user credentials
  - Pre-configured network with static IP that users are allowed to switch to
  - … (NM supports basically everything that ifupdown supports, in addition)

- Users with the appropriate PolicyKit permissions can still declare their own connections (e.g. WiFi roaming)

```
[802-3-ethernet]
duplex=full
mac-address=de:ad:be:ff:13:37

[connection]
id=NAC
uuid=b63b3cf5-4895-45e1-a5b6-3a4f38a20b99
type=802-3-ethernet

[ipv6]
method=auto

[802-1x]
eap=tls;
identity=Joe's machine
cia-cert=/etc/ssl/certs/nolcorp_ca.pem
client-cert=/etc/ssl/private/joe.pem
private-key=/etc/ssl/private/joe.key.pem
private-key-password=plop

[ipv4]
method=auto
```

Identifies the device
Random
Mount actions are explicit from an application
→ Done by **gnome-shell** when receiving a media notification
→ Done by **nautilus** when selecting a remote share

**Command-line:**
- See all mounted filesystems: `gvfs-mount -l`
- Mount a CIFS mount: `gvfs-mount smb://server/share/path`

**Gvfs-fuse:** nautilus redirects applications not supporting GIO to `/run/user/$uid/gvfs`
- Needs `fuse` group membership

**Note:** jessie is in the middle of a udisks → udisks2 transition
The gnome-disks interface
- Per-application software mixing for all sound providers
- Default Debian configuration is suitable for multiple users
  - Mute sound when switching users (using logind)
- Configuration needed only for people with specific needs
  - Sound over the network: RAOP/ZeroConf, EsounD, UPnP…
  - Pass-through
Printing

- CUPS DBus / PolicyKit interface: `cups-pk-helper`
  - gnome-control-center configures printers
  - gnome-settings-daemon notifies of print operations
  - Not very useful in a multiple-user, multiple-machine environment

- A CUPS server can hold thousands of printers
  → but the UI on the clients becomes unusable
  - No standard solution to filter printers out

Power management

- System DBus / PolicyKit interface: `upower`
  - The policy is applied by gnome-settings-daemon based on Gsettings.
  - Also queried by GNOME shell (in session and in GDM)
PackageKit

- A D-Bus interface to abstract package managers
  - Checking for updates: gnome-settings-daemon
  - Installing updates: gpk-update-viewer frontend
  - Adding/removing software: gpk-application
  - Distribution upgrades: not recommended
- Do you want users to play with packages?
  - Sometimes unattended-upgrades is more reasonable

Note: Debian jessie doesn’t use gnome-software
GNOME is scriptable

- **In Python:**
  ```python
  from gi.repository import Gtk, GnomeKeyring, …
  ```

- **In JavaScript:**
  ```javascript
  #! /usr/bin/seed
  Gtk = imports.gi.Gtk;
  ```

- **In shell with zenity**

- **Some real-world-examples:**
  - A daemon / applet to bypass an IE-only enterprise proxy
    Notification area / libnotify: display status
    Autostart with the session
    Store the password in the keyring
  - A script to create CIFS shortcuts accessible from “Places” menu
    Store the password in the keyring for GVFS
    `~/.gtk-bookmarks` → “Places” and the shortcuts for GtkFileChooser
  - A script to wrap a RDP / Citrix client
    Extract the same password as for CIFS
An infrastructure for Debian/GNOME machines

- Debian provides the desktop ready to use
  - But you need to **build your infrastructure** with the included bricks
- **Authentication**: OpenLDAP, Fedora directory server, Active Directory
  → Think about using **sssd**
- **Printing** is hard (see before)
- Network file systems: don’t forget about **NTP**!
- Need changes in packages? A Debian **mirror** and a custom APT **repository**
  → rsync / debmirror and reprepro / mini-dinstall / …
- Lots of machines? How about a custom **installation** media
- Remote management: you want a tool that works in **pull mode**, e.g. **Puppet**
  - Can be linked to inventory: GLPI + **FusionInventory**
- **Root password management** anyone?
- You encrypt partitions? Don’t forget about legal requirements (key escrow)
Thank you.