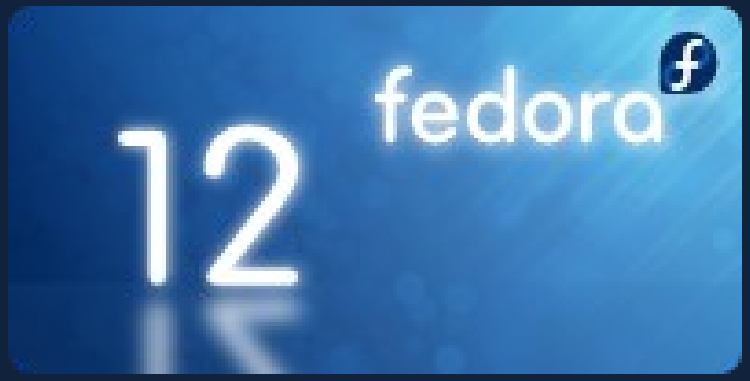
Fedora System Configuration overview



Jaroslav Řezník <jreznik@redhat.com>
Roman Rakus <rrakus@redhat.com>

revision 20091203-1

Agenda

- **Introduction**
- **System configuration tools overview**
- **System configuration tools cleanup project**
- **How to use Policy Kit**

Introduction

- **Various system parts needs to be configured**
- **Various admin knowledge of system configuration**

System configuration tools overview

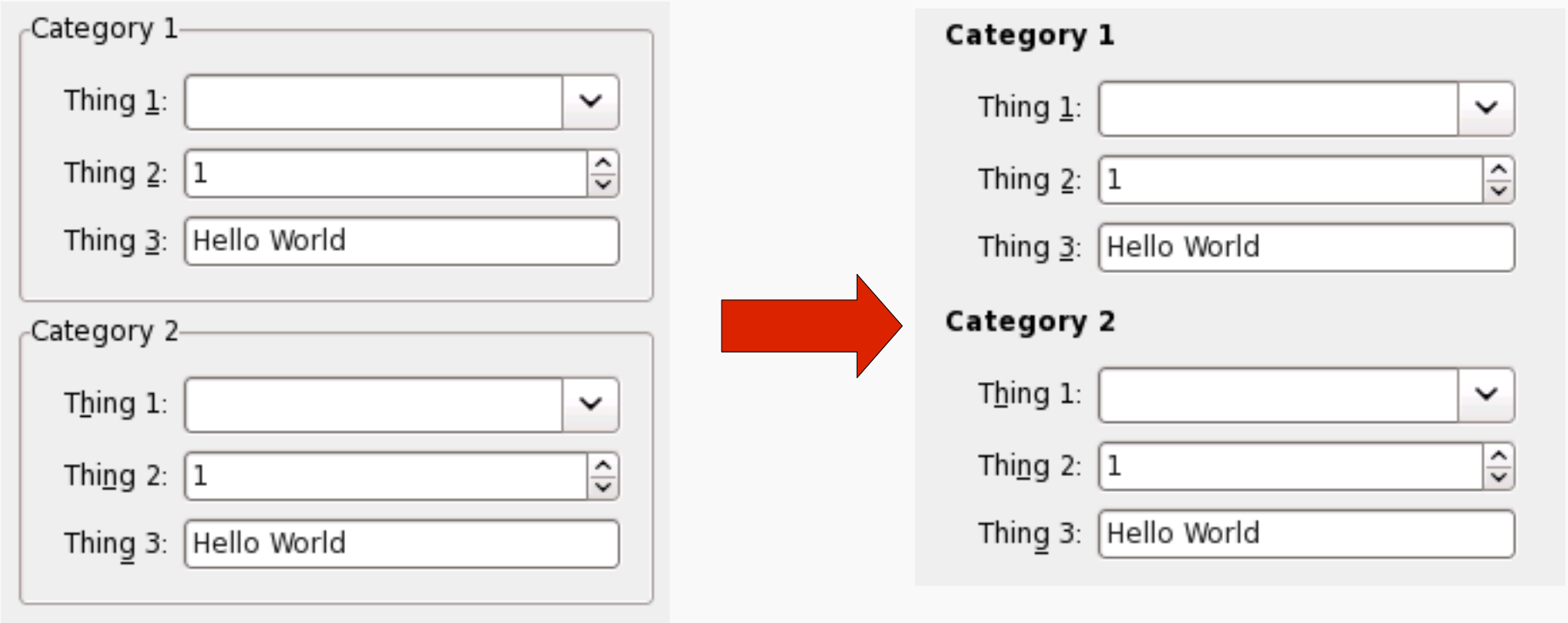
- **Manual editing**
  - Configuration files
  - Manage services
  - etc
- **System config tools**
  - GUI tools (Gtk), some have TUI or CLI
  - Mostly monolithic applications
- **Gnome/KDE settings**
  - User session only configuration
- **Application specific settings**
  - Firefox, Thunderbird, X-Chat, ...

Cleanup – The Idea

- **Consistent look & feel**
  - Follow Gnome Human Interface Guidelines (not 100%!)
  - http://library.gnome.org/devel/hig-book/stable/
- **Eliminate tools not used any more**
  - Outdated tools
  - Obsoleted by another tool
  - Autodetection
- **More/better functionality**
  - Augeas
  - Backend/frontend separation + PolicyKit
  - Troubleshooting
- **See Red Hat's Bugzilla tracker bug**
  - https://bugzilla.redhat.com/show_bug.cgi?id=480902

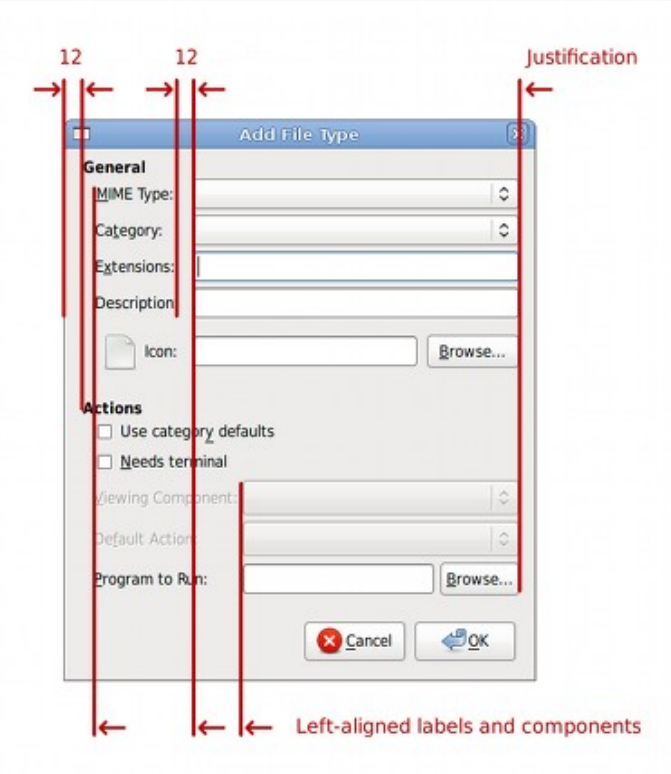
Some usability tips

- **Do not use frames and separators, use bold label with proper alignment**



Some usability tips

- **Dialogs spacing and positioning**
  - http://library.gnome.org/devel/hig-book/stable/design-window.html.en

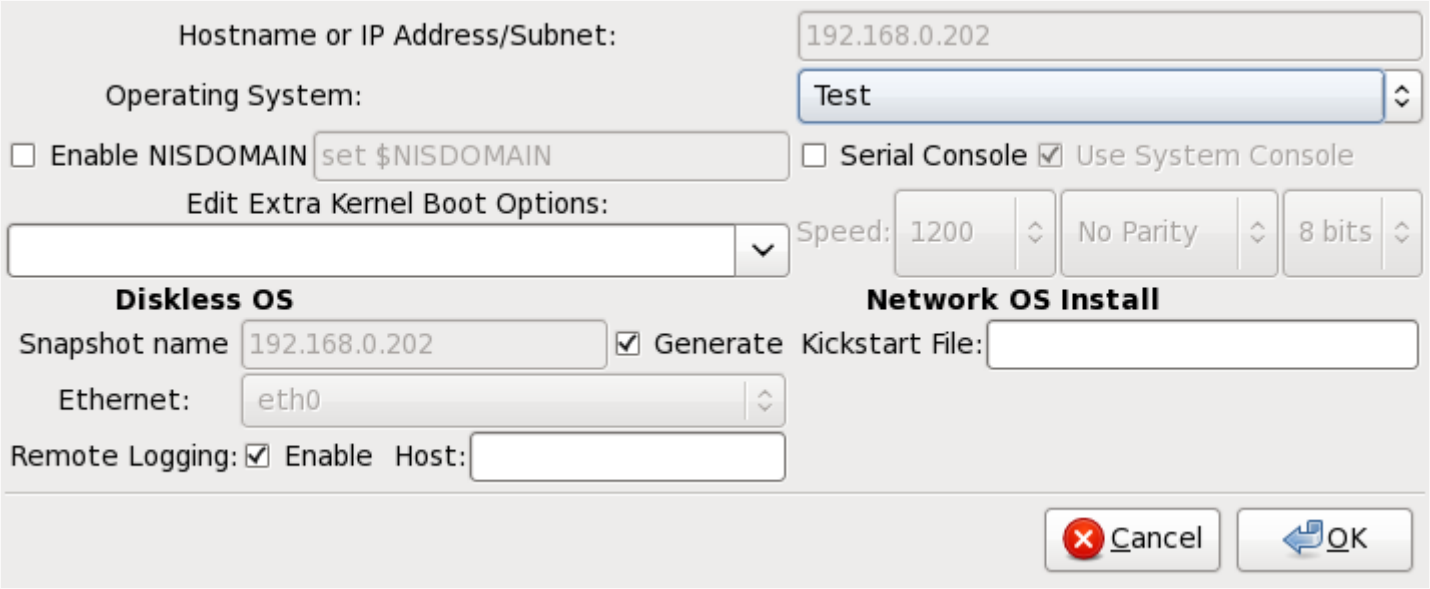


Some usability tips

- **Use standard about dialog**
  - With authors, license etc.
- **Child windows centered on parent window**
- **"OK" button sensitivity and controls sensitivity**
- **Missing tooltips**
- **Help (DocBook)**
- **Inputs checking**
- **Be more verbose, status, progress bar**

Example

- **system-config-netboot**

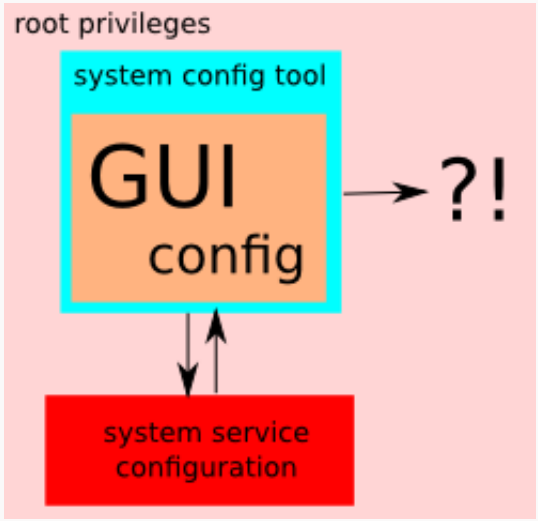


Separation, PolicyKit and Fedora

- **Bugs like**
  - S-c-tools cleanup: port to PolicyKit
  - Deprecate consolehelper and switch apps to use PolicyKit 1 for Fedora 12
  - port XXX to PolicyKit 1.0

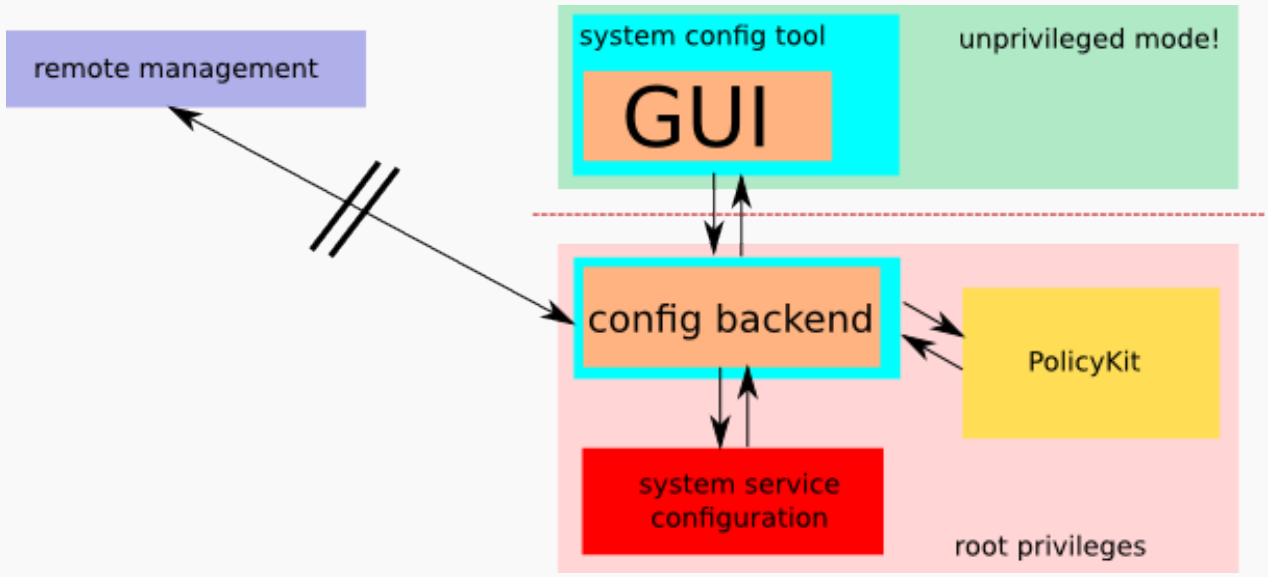
Current state 1/2

- Root environment
- Inconsistent
- No remote access



Separation + PolicyKit 1/2

- **2 separate modes (root and non-root)**
- **Communication (dbus)**
- **Authentication (polkit)**



Solutions

- **Separate frontend and backend**
  - Frontend is just a UI
  - Backend makes the configuration
- **Backend's communication**

Plan

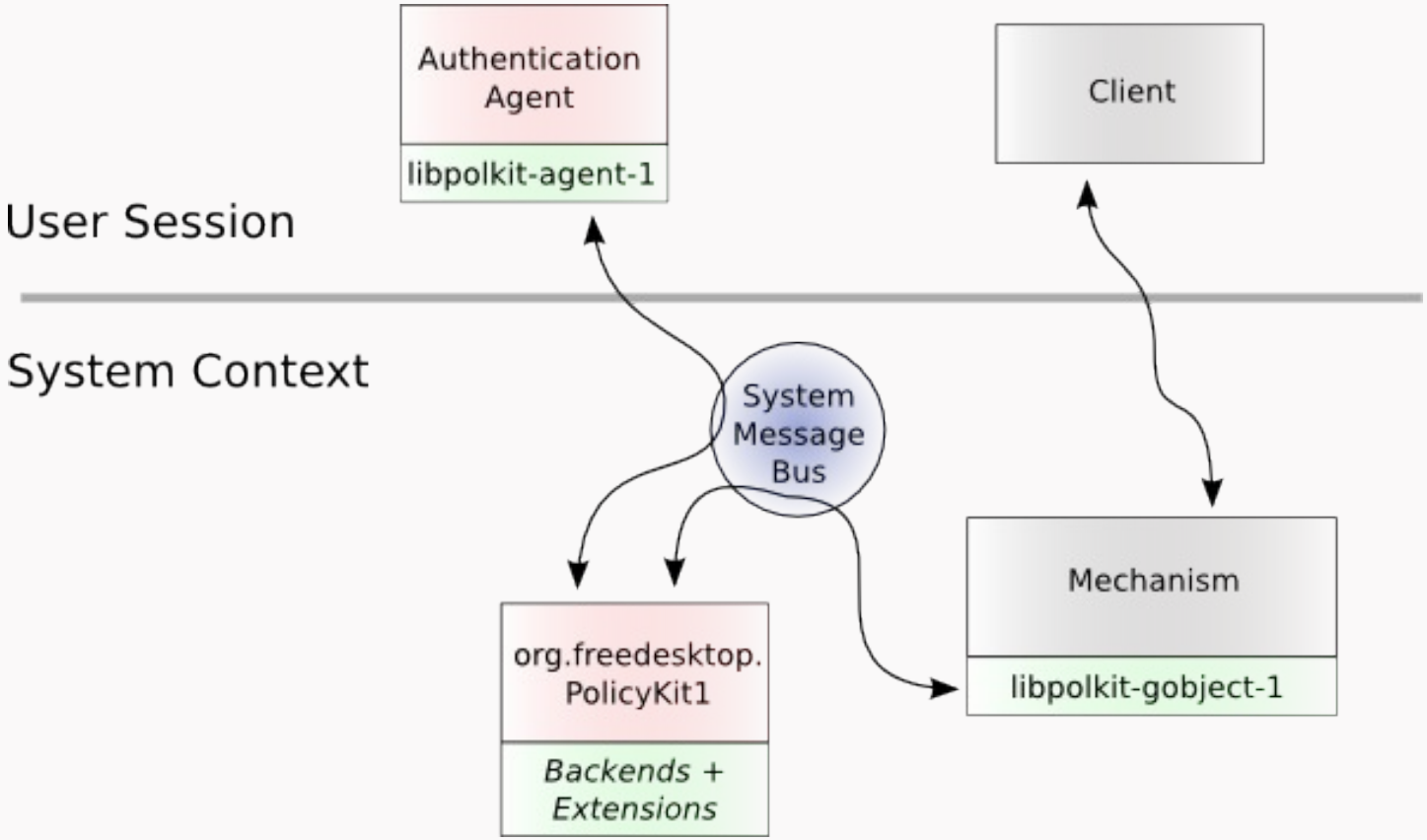
- **Review current system configuration tools**
  - Use cases of these tools
  - Contact and ask maintainers for help
- **Define interface for selected use cases**
  - Based on use cases
  - Again we need help from authors
- **Implement it ;-)**

Implementation

- **No framework...**
  - D-Bus
  - PolicyKit
  - Python-SLIP
- **...but guidelines**
  - same/similar behavior

Policy Kit 1 - architecture

- **Policy Kit daemon**
- **Authentication agent**



Usage

- **Backend / frontend separation**
  - Dbus preferred
  - Possible TCP/IP etc...
- **Backend runs in privileged mode**
- **Frontend runs in unprivileged mode**
  - GUI, CLI...
- **PolicyKit authentication in backend**
  - Frontend does not care...

Authentication agent

- **Takes care about authentication**
  - PAM, simple Yes/No dialog etc.
- **org.freedesktop.PolicyKit1.AuthenticationAgent interface**
- **Authentication agent has to be registered**
  - RegisterAuthenticationAgent in org.Freedesktop.PolicyKit1.Authority interface
  - Own authentication agent can be registered
  - Solves problem with choosing right agent in user session
- **Currently Gnome Authentication Agent in Fedora 12**
- **KDE Authentication Agent for Fedora 13, under kdereview process**

Usage

- **DBus interface**
  - Direct communication over DBus
- **GObject based library**
  - On top of DBus interface, glib like interface
- **PolicyKitQt-1**
  - Ongoing work on Qt like interface on top of GObject interface, later DBus
- **Python-slip**
  - Used in System Configuration Tools
  - Make using DBus and PolicyKit simple
- **KAuth library with PolicyKit backend**
  - New KDE library for authorizations
  - Multiplatform

Example .policy file

- **XML file, stored in /usr/share/polkit-1/actions**

```
<policyconfig>
  <vendor>Red Hat, Inc.</vendor>
  <vendor_url>http://www.redhat.com</vendor_url>

  <action id="com.redhat.devconf.examples.cry">
    <description>Cry</description>
    <message>Prevents me from crying</message>
    <defaults>
      <allow_inactive>no</allow_inactive>
      <allow_active>no</allow_active>
    </defaults>
  </action>
```

Default policies

- **Defaults – implicit authorizations for**
  - any client (allow_any)
  - clients in inactive sessions on local consoles (allow_inactive)
  - clients in active sessions on local consoles (allow_active)
- **With the following values**
  - no - not authorized
  - yes - authorized
  - auth_self - owner of the session
  - auth_admin - administrative user
  - auth_self_keep – like auth_self but for a brief period
  - auth_admin_keep - like auth_admin but for a brief period

Example – PolicyKit 1 auth. check

- **polkit-gobject-1 simple example – PolicyKit 1 check only, no separation & DBus involved**
- **Includes**
  - #include <polkit/polkit.h>
  - #include <glib-object.h>
- **Authority object**
  - PolkitAuthority *authority;
  - authority = polkit_authority_get();
- **Subjects – who we are going to authorize**
  - PolkitSubject
  - PolkitUnixProcess, PolkitUnixSession, PolkitSystemBusName

Example

- **Subject – cont. code**
  - PolkitSubject *subject;
  - subject = polkit_unix_process_new(getpid());
- **Authorization**
  - PolkitAuthorizationResult *result;
  - GError *error = NULL;
  - result = polkit_authority_check_authorization_sync(authority,
  - subject,
  - action_id,
  - NULL,
  - flags,
  - NULL,
  - &error);

Example

- **Checking result**
  - Challenge – more action needed
  - polkit_authorization_result_get_is_challenge(result)
  - polkit_authorization_result_get_is_authorized(result)
- **Flags**
  - No flags set
  - *POLKIT_CHECK_AUTHORIZATION_FLAGS_NONE*
  - Authentication through Authentication Agent
  - *POLKIT_CHECK_AUTHORIZATION_FLAGS_ALLOW_USER_INTERACTION*

Usage (DBus & PolicyKit)

- **Client task**
  - Call action method
- **Mechanism task**
  - Listen on DBus for method call from Client
  - Authorize action
- **Python-SLIP example**

Example – Python-SLIP

- **Example from Python-SLIP**
  - In source package – doc/dbus/example
- **Mechanism part**
  - Create slip.dbus.service.Object
  - And implement actual DBus method
  - For example org.fedoraproject.slip.example.mechanism.read

```
@slip.dbus.polkit.require_auth ("org.fedoraproject.slip.example.read")
@dbus.service.method ("org.fedoraproject.slip.example.mechanism",
                      in_signature='', out_signature='s')
def read (self):
    print "%s.read () -> '%s'" % (self, self.config_data)
    return self.config_data
```

Example - Python-SLIP

- **Client part**

```
class DBusProxy (object):
    def __init__ (self):
        self.bus = dbus.SystemBus ()
        self.dbus_object = self.bus.get_object ("org.fedoraproject.slip.example.mechanism",
"/org/fedoraproject/slip/example/object")

    @polkit.enable_proxy
    def read (self):
        return self.dbus_object.read (dbus_interface = "org.fedoraproject.slip.example.mechanism")
```

Questions?

- **THANK YOU!**

References

1) **http://www.fedoraproject.org/wiki/Features/SystemConfigCleanup**
2) **http://www.fedoraproject.org/wiki/SystemConfig**
3) **http://hal.freedesktop.org/docs/polkit/**
4) **https://fedorahosted.org/python-slip/**