

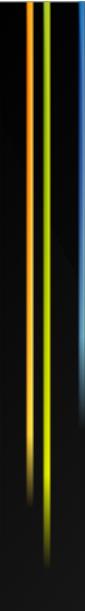
```
static void  
properties(GObjectClass  
*gobject_class)  
{  
    mSpec *pspec;
```



Light content sharing from the desktop

Enrique Ocaña González

eocanha@igalia.com

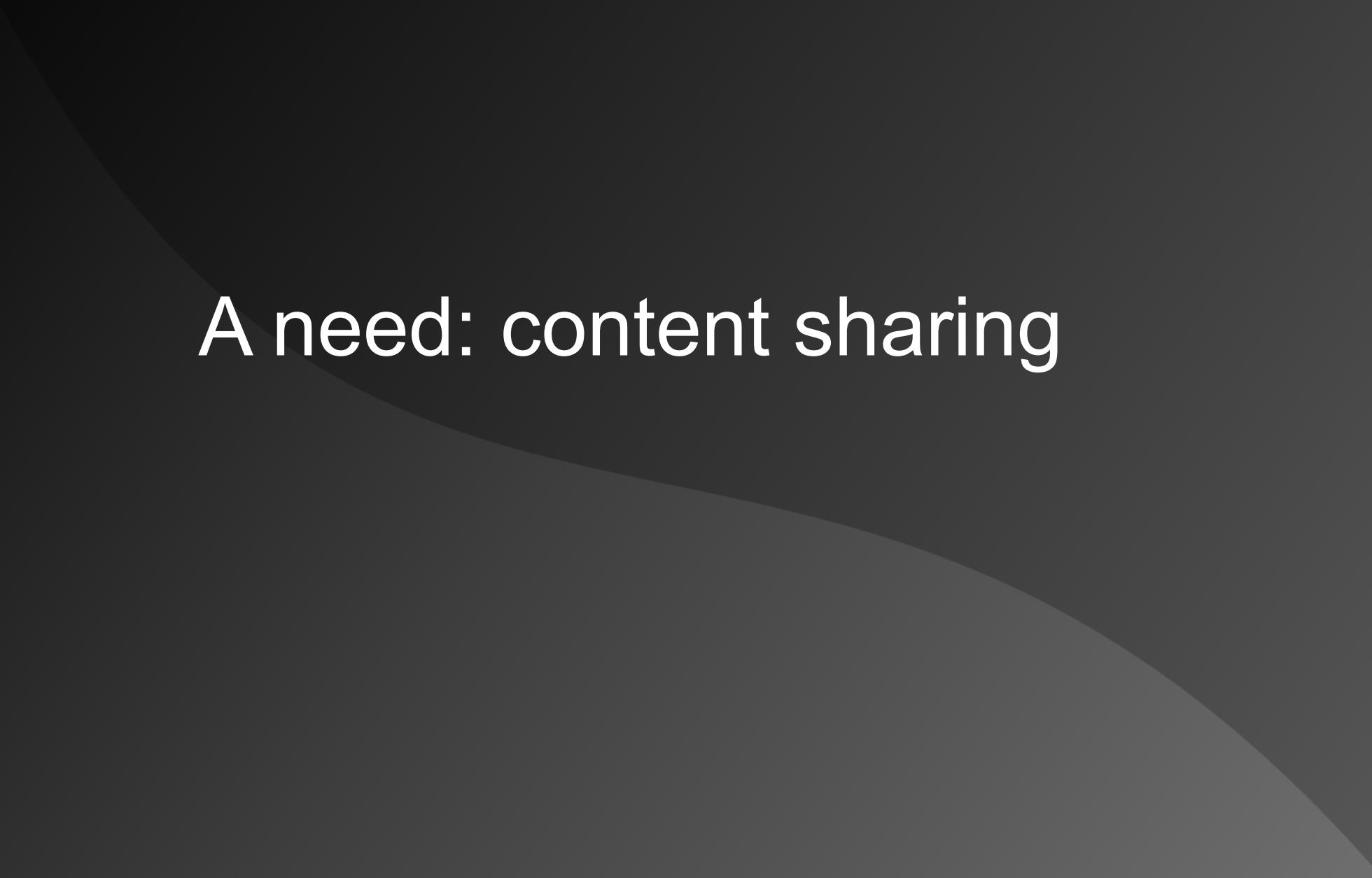


Index

- A need: content sharing
- Development process
- The result
- How it works
- Ideas for the future
- References



A need: content sharing



Motivation

Many desktop applications **use** web services and online contents

Few **offer** contents to other applications (P2P, servers)

Use cases:

- Share files or pictures in an convenient way
- Be an agent that generates XML or RSS contents which can be consumed by an external aggregation service (planet, Yahoo pipes, etc.)
- Remotely command applications

Alternatives

Dedicated server technologies:

Apache+PHP, Java/Tomcat, Mono/ASP...

Desktop utilities:

Gnome-user-share

KDE File server

Opera Unite

Disadvantages

Dedicated server technologies:

- Heavy, complex to set up

- Not integrated with Gnome mainloop

Desktop utilities:

- Gnome-user-share: depends on Apache (heavy)

- KDE File server: Kicker applet

- Opera Unite: not free software

Solution

Lightweight GPL application using technologies friendly with Gnome desktop:

Based on GObject

Easy integration with main loop

Possibility of DBUS services to third parties

Simplified network configuration

Existent libraries:

libsoup

libgupnp

Libsoup 2.4

Asynchronous, uses Glib main loop

SSL support using GNU TLS

Server basic and digest authentication support

SOAP and XML-RPC basic client support

Libgupnp 0.6

Asynchronous, uses Glib main loop

UPnP allows to configure port dedirection if supported by the router

Initial evolution plan



Stage 1

- Evaluate libraries
- Implement a little file server similar to KDE Public Fileserver
- Code a DBUS service to allow other programs to register/unregister files for sharing

Stage 2

- Generate RSS feeds
- Register more complex things: remote calls to functions, etc.

Stage 3

- Extend some existing Gnome applications to implement functionalities such as:
 - “Offer all your pictures as RSS”
 - “Offer the playlist of songs you've listened today to as RSS”

Meiga está ahora
aproximadamente
aquí

Development process

Programming language

Vala was the chosen language

Advantages:

- Easy to use → Less development time

- Execution speed

- Previous experience

Disadvantages:

- Evolving syntax

- There's no binding for all the libraries → Pure C

- More complex autotools integration

Real project evolution



File server module

Real project evolution



File server module

DBUS interface for external applications

Real project evolution



File server module

DBUS interface for external applications

RSS server module

Real project evolution



File server module

DBUS interface for external applications

RSS server module

Graphic interface

Real project evolution



File server module

DBUS interface for external applications

RSS server module

Graphic interface

Autotools integration

Real project evolution



File server module

DBUS interface for external applications

RSS server module

Graphic interface

Autotools integration

UpnP redirection utility

Release 0.1.0

Real project evolution



File server module

DBUS interface for external applications

RSS server module

Graphic interface

Autotools integration

UpnP redirection utility

Release 0.1.0

Real project evolution



Port from Glade to GtkBuilder

Real project evolution



Port from Glade to GtkBuilder

Log exposing to the graphic interface

Real project evolution



Port from Glade to GtkBuilder

Log exposing to the graphic interface

Spanish and Galician internationalization

Real project evolution



Port from Glade to GtkBuilder

Log exposing to the graphic interface

Spanish and Galician internationalization

Bug correction

Real project evolution



Port from Glade to GtkBuilder

Log exposing to the graphic interface

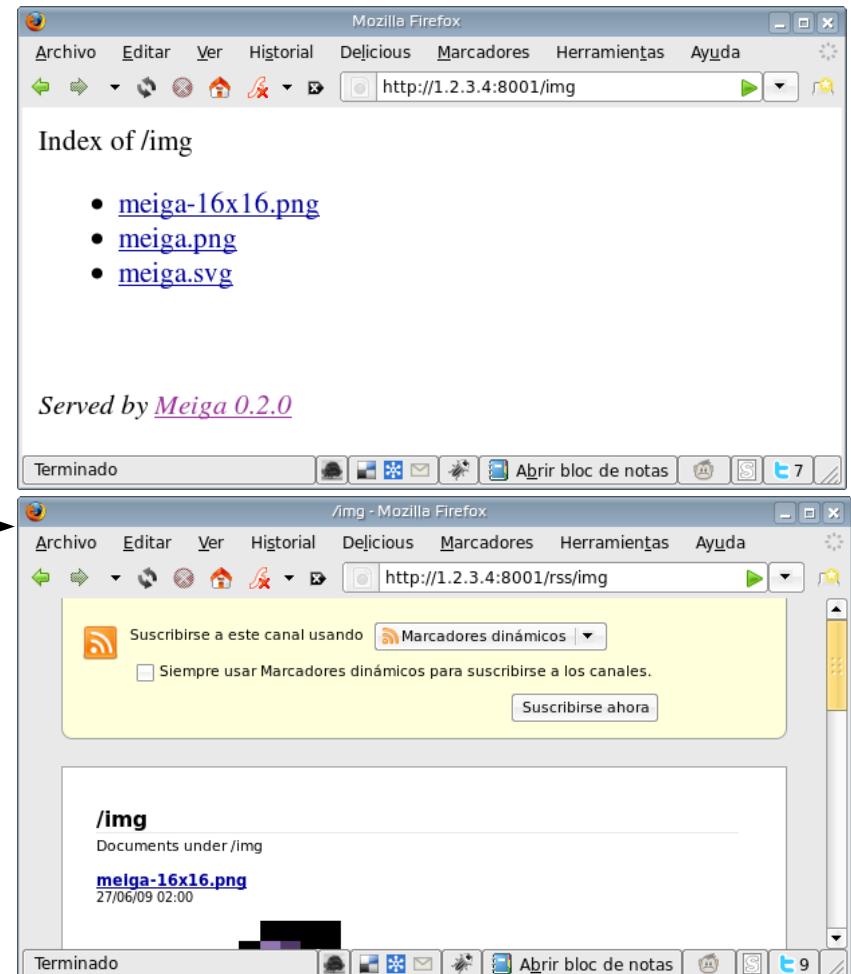
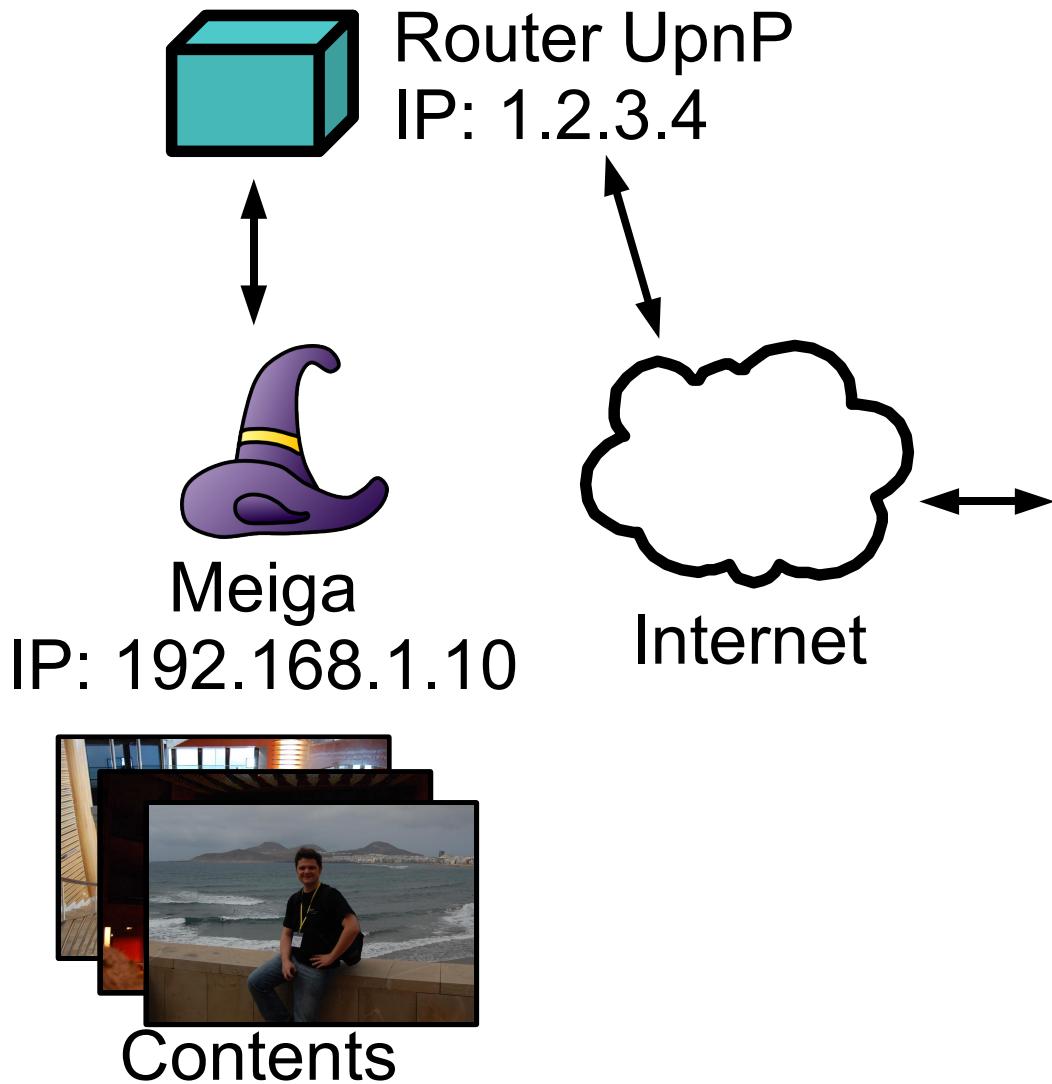
Spanish and Galician internationalization

Bug correction

Release 0.2.0

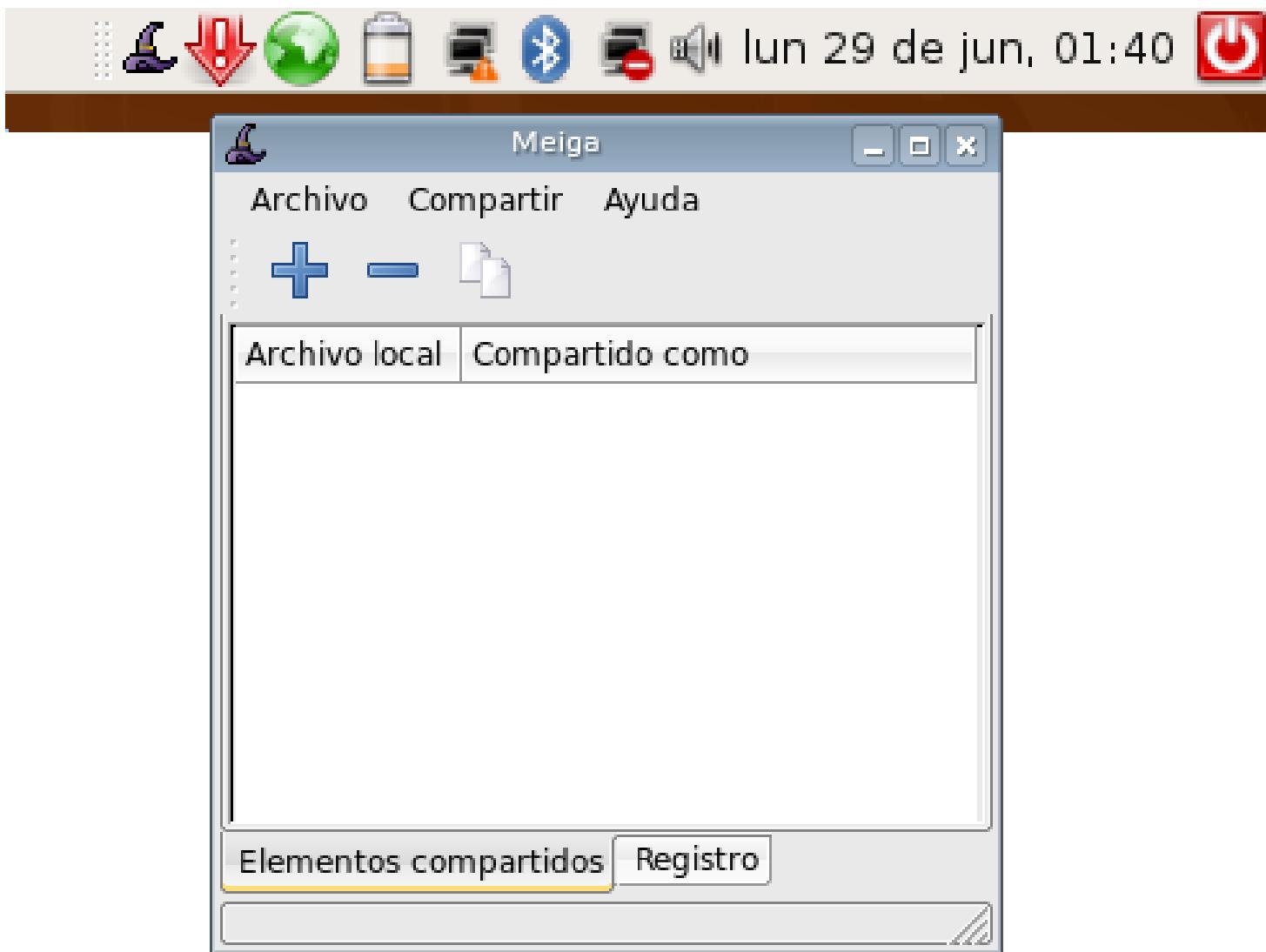
The result

Home network

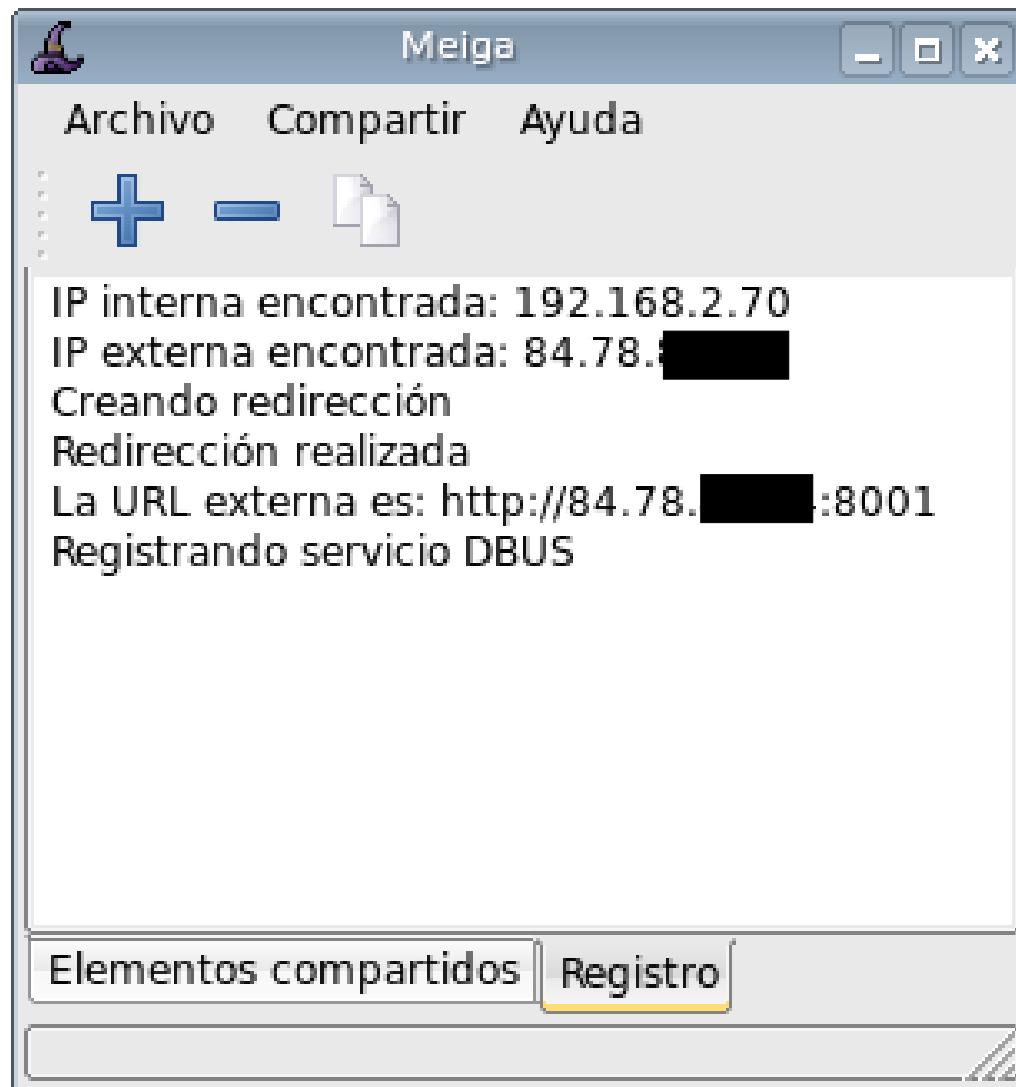


Remote user

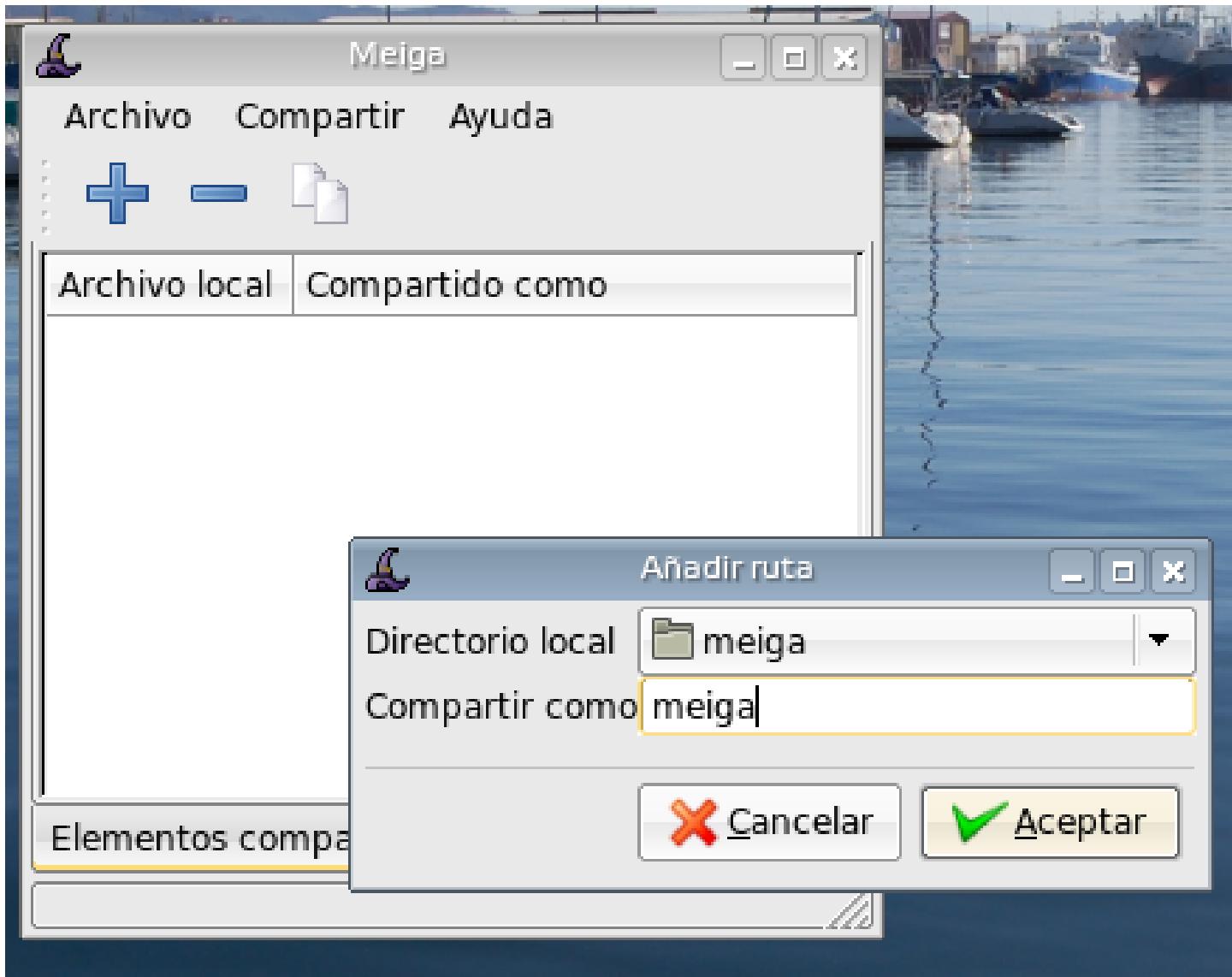
Main window and system tray



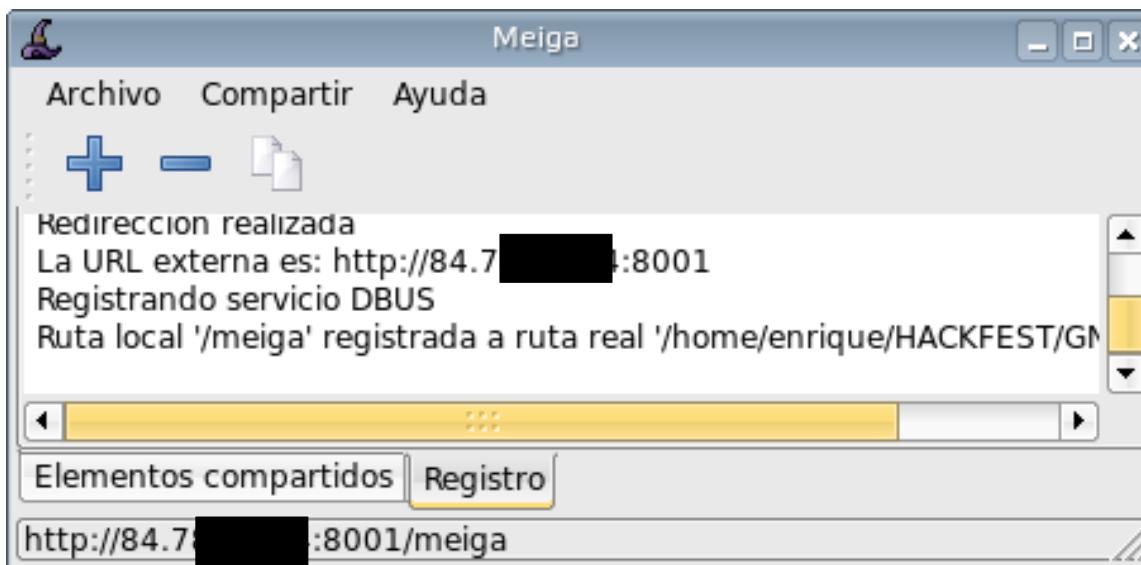
Log and port redirection



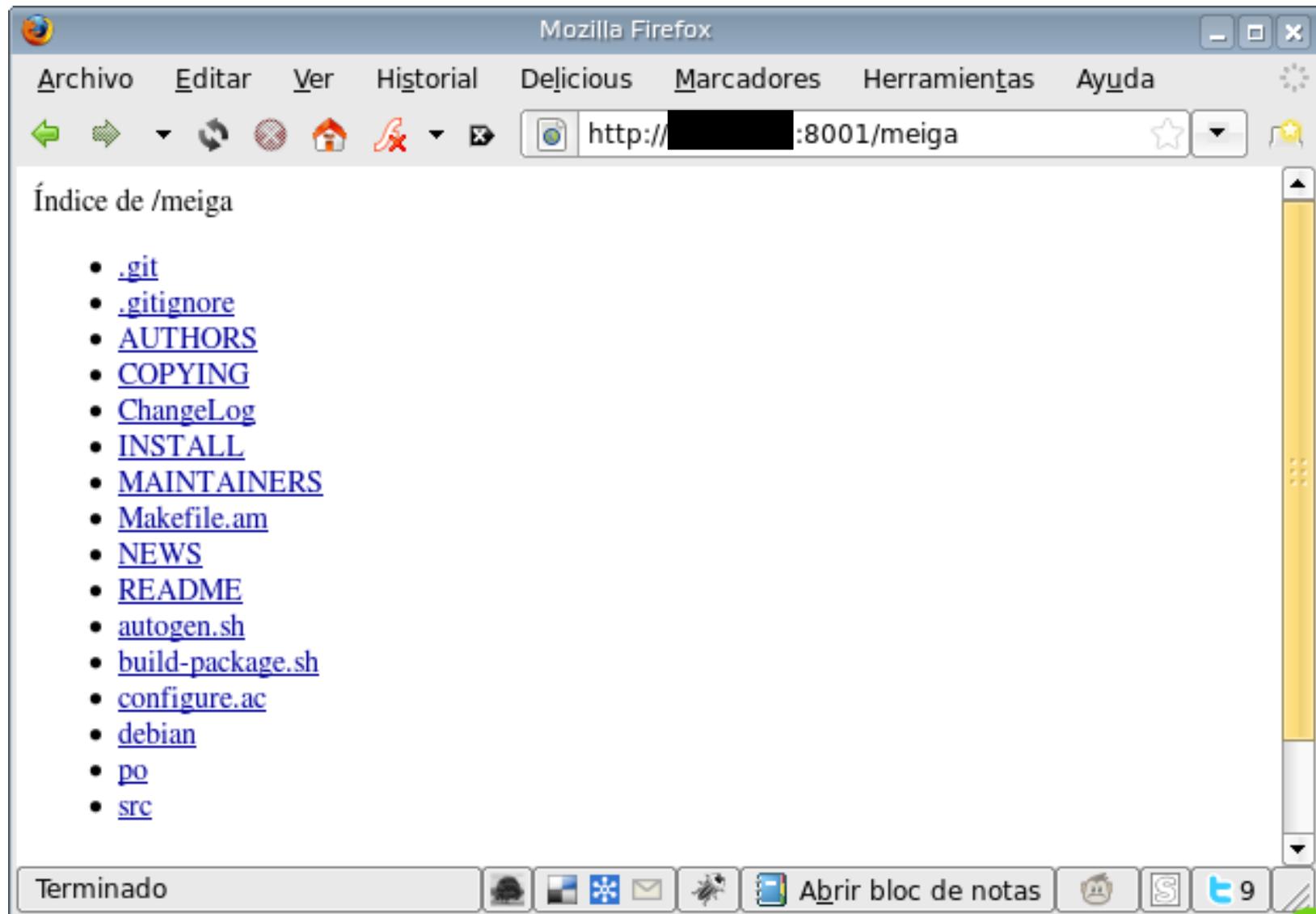
Sharing a folder



Sharing a folder



Remote access



Remote access

A screenshot of a Mozilla Firefox browser window. The title bar says "Mozilla Firefox". The menu bar includes "Archivo", "Editar", "Ver", "Historial", "Delicious", "Marcadores", "Herramientas", and "Ayuda". The toolbar includes standard icons for back, forward, search, and refresh. The address bar shows "http://[REDACTED]:8001/meiga/INSTALL". The main content area displays text related to the compilation of Meiga:

```
Compilation dependencies
*****
To compile Meiga you need the following packages installed:

gnome-common
pkg-config
libgtk2.0-dev      (>= 2.10.0)
libsoup2.2-dev     (>= 2.2.0)
libsoup2.4-dev     (>= 2.4.0)
libdbus-1-dev       (>= 0.74)
libdbus-glib-1-dev (>= 0.74)
libgupnp-1.0-dev   (>= 0.6)

In addition, the Vala 0.7.2 compiler is needed (0.5.2 is also reported
to work). If you have Ubuntu Jaunty or Debian Sid, you can install the
package. If you're using an older distribution, you must install it
by hand.

You can download and compile it yourself from here:
```

The status bar at the bottom shows "Terminado" and several icons for system notifications, including a speech bubble with "9", a green "e" icon, and a blue "k" icon.

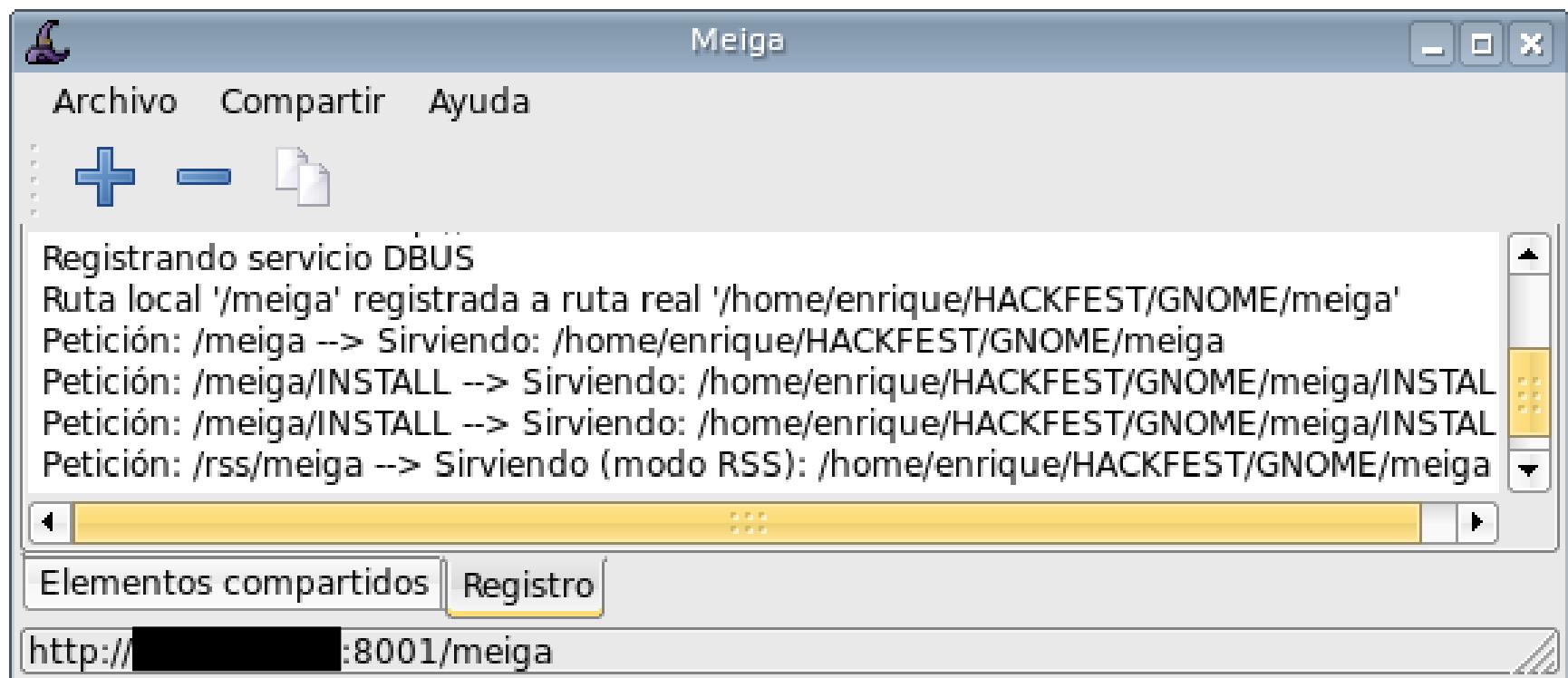
Contents as RSS

The screenshot shows a Mozilla Firefox window with the title bar reading "/meiga - Mozilla Firefox". The menu bar includes Archivo, Editar, Ver, Histórial, Delicious, Marcadores, Herramientas, and Ayuda. The address bar shows the URL [http://\[REDACTED\]:8001/rss/meiga](http://[REDACTED]:8001/rss/meiga). A yellow sidebar on the left contains a "Suscribirse a este canal usando" section with an RSS icon, a dropdown menu set to "Marcadores dinámicos", and a checkbox for "Siempre usar Marcadores dinámicos para suscribirse a los canales". A "Suscribirse ahora" button is also present. The main content area displays a list of files under the heading "/meiga":

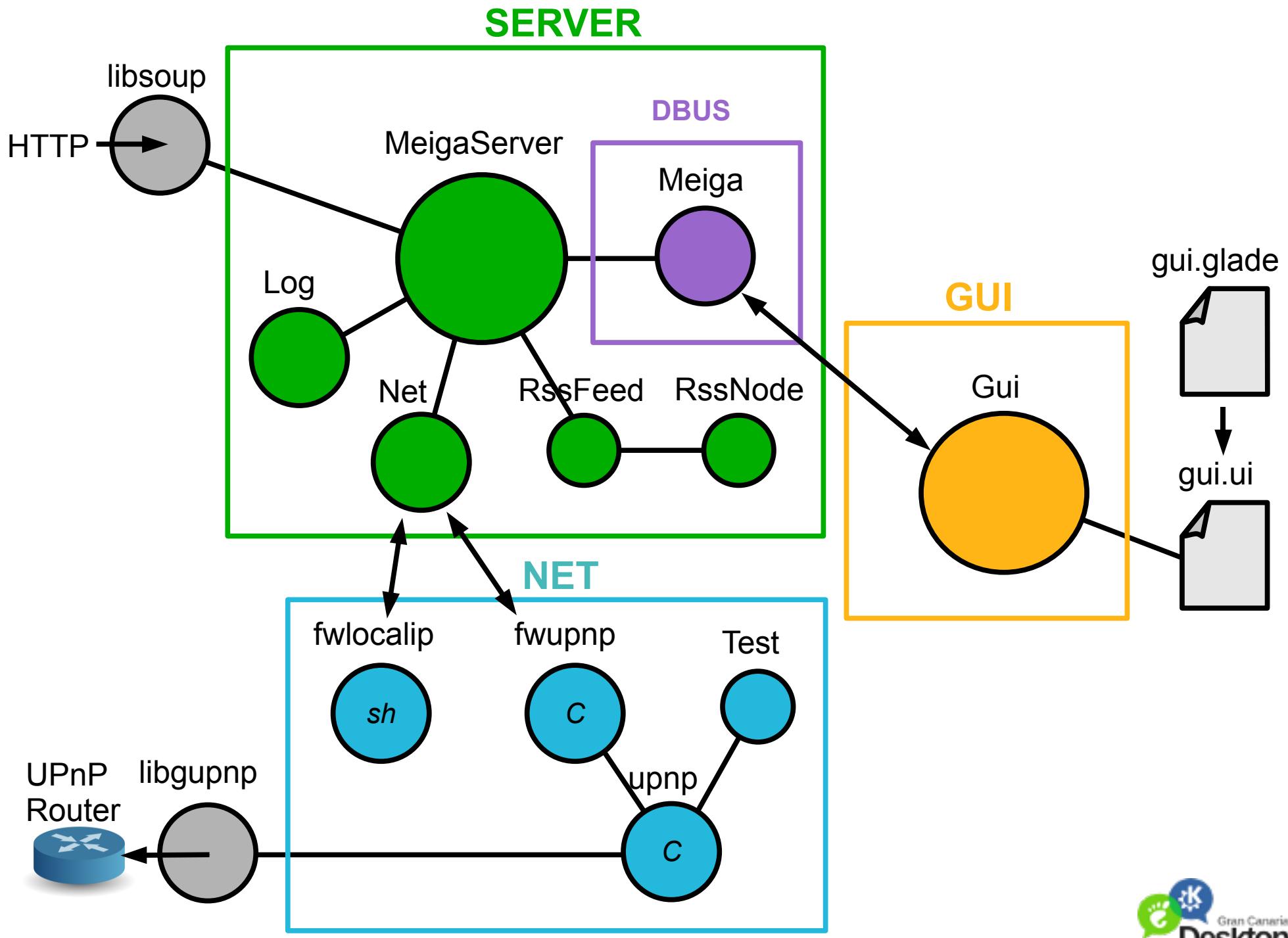
- [configure.ac](#)**
27/06/09 02:00
[Descargar configure.ac](#)
- [.gitignore](#)**
27/06/09 02:00
[Descargar .gitignore](#)
- [AUTHORS](#)**
27/06/09 02:00
[Descargar AUTHORS](#)
- [COPYING](#)**
27/06/09 02:00
[Descargar COPYING](#)
- [Changelog](#)**

The status bar at the bottom indicates "Terminado".

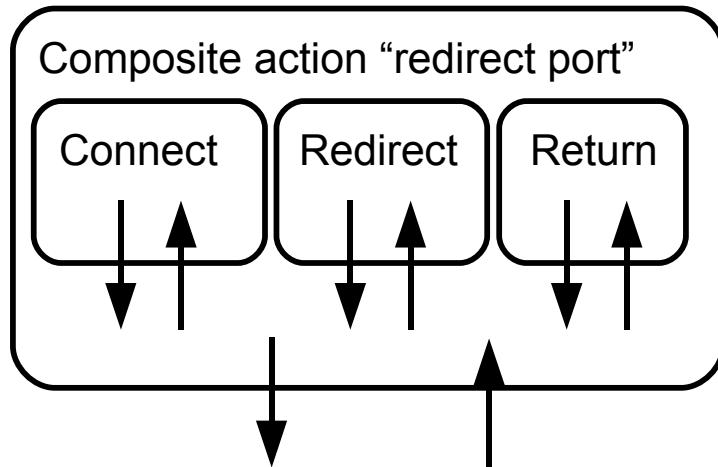
Log



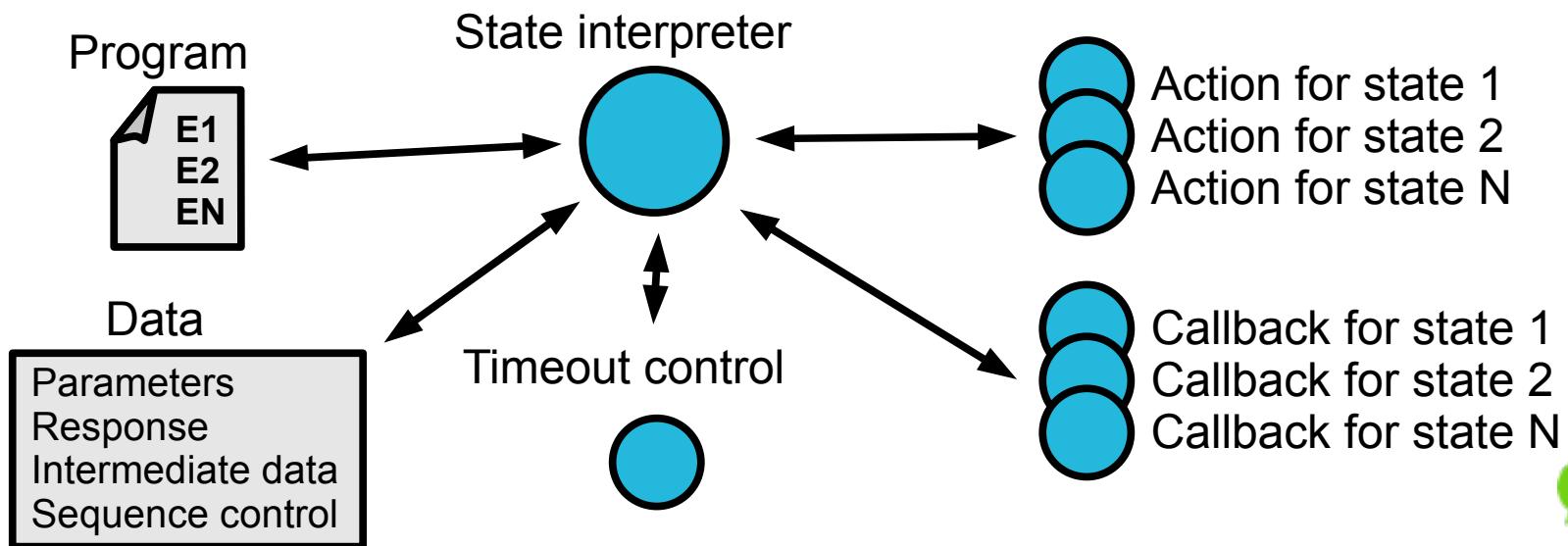
How it works



Programmable serial asynch events management



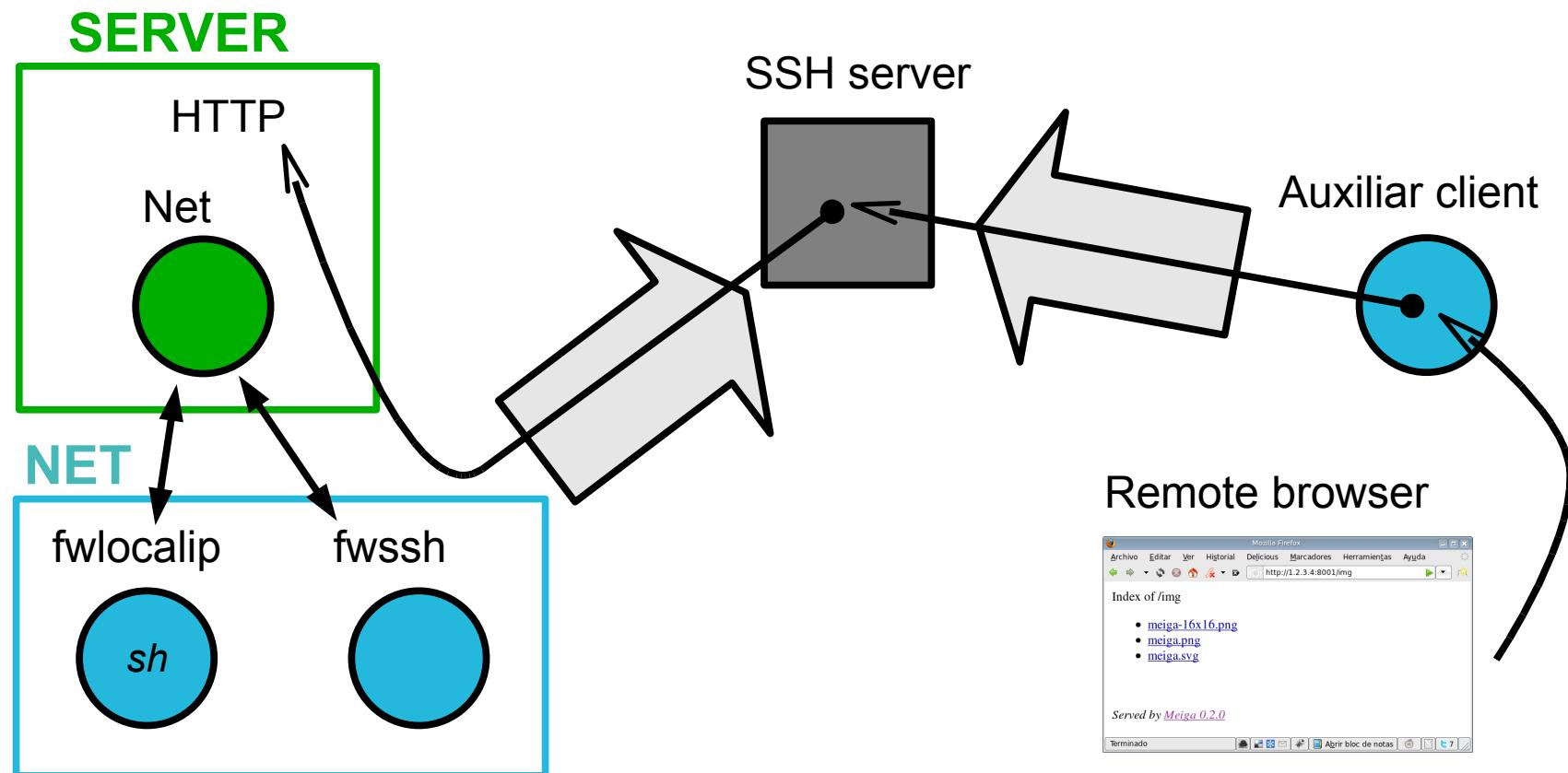
Some states are reused for other composite actions



Ideas for the future

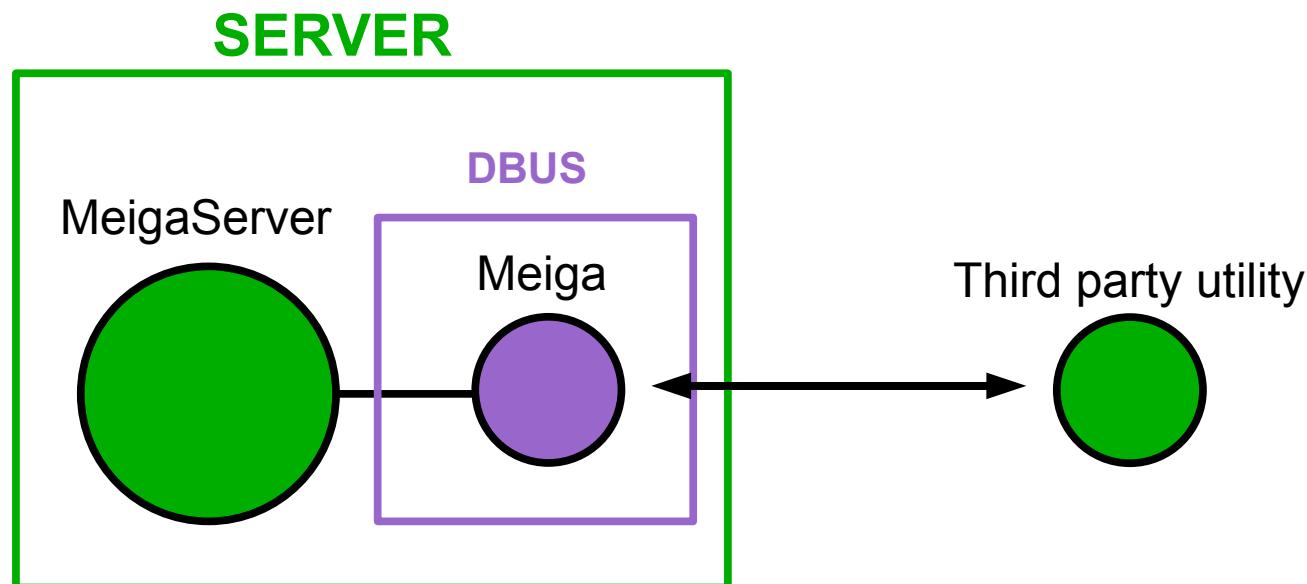
More redirection modules

Redirection by SSH



Integration with other desktop applications

Nautilus menu: “Share on Meiga”



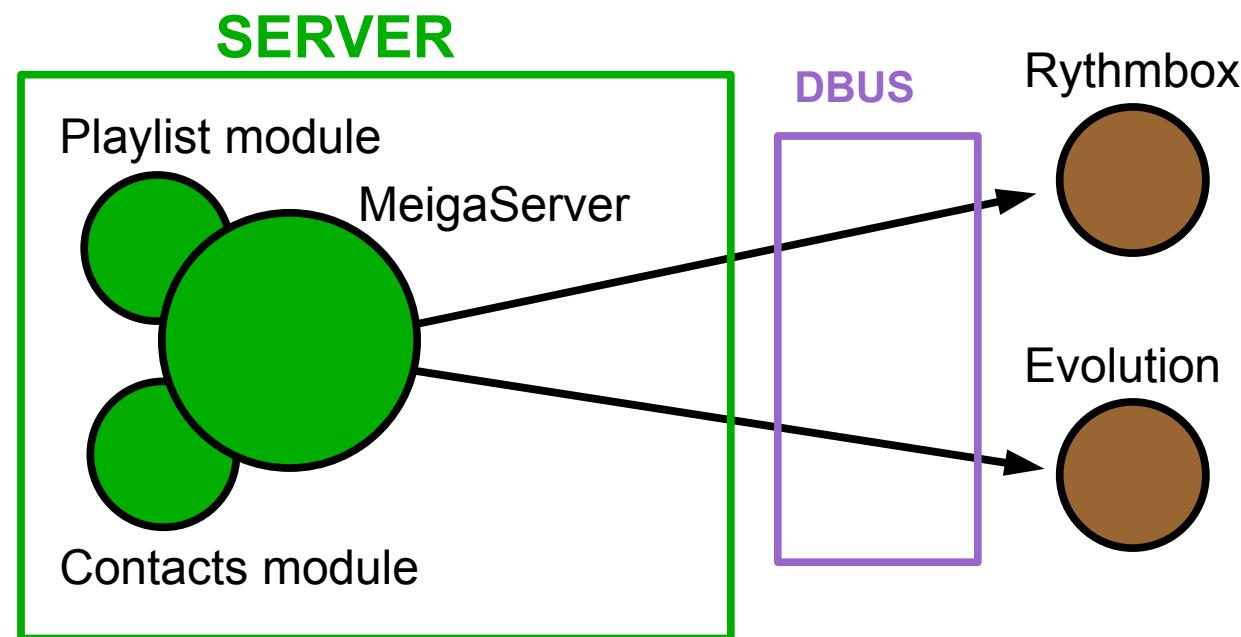
More server modules

Function calls

URL

/playlist/...

/addressbook/...



Security

Password access

Cyphering

Restriction by IP

etc.

Situations to avoid

Meiga shouldn't become a comprehensive interface for DBUS through HTTP (“serve everything”)

- Security holes
- There are currently more advanced utilities for network object sharing. Eg: CORBA



References

References

- <http://meiga.igalia.com>
- <http://live.gnome.org/Vala>
- <http://library.gnome.org-devel/libsoup/stable/libsoup-server-howto.html>
- http://dbus.freedesktop.org/doc/api/html/group__DBusBus.html
- <http://raphael.slinckx.net/blog/documents/dbus-tutorial>
- <http://www.gupnp.org/docs/gupnp/client-tutorial.html>
- <http://www.upnp.org/standardizeddcps/igd.asp99>
- <http://www.lrde.epita.fr/~adl/autotools.html>
- <http://live.gnome.org/Vala/GameDevelopmentSeries/Setup>



<http://meiga.igalia.com>

Thank you very much

Questions, comments, suggestions?