

GNOME 3 (3.20)

We've come a long way

Tobias Mueller

tobiasmue@gnome.org



“GNOME”?

“GNOME 3”?

“GNOME 3.0”?

“GNOME 3.20”?



Elegant design



GNOME <https://www.gnome.org/>

[GNOME 3](#)
[Applications](#)
[About Us](#)
[Get Involved](#)
[Translation](#)

GNOME 3: Ease, comfort and control

GNOME 3 is an easy and elegant way to use your computer. It is designed to put you in control and bring freedom to everybody. GNOME 3 is developed by the GNOME community, a diverse, international group of contributors that is supported by an independent, non-profit foundation.

[Discover GNOME 3](#)
[Get GNOME 3](#)

[Make a donation and become a Friend of GNOME!](#)

Your donation will ensure that GNOME continues to be a free and open source desktop by providing resources to developers, software and education for end users, and promotion for GNOME worldwide.

[Get involved!](#)

The GNOME Project is a diverse international community which involves hundreds of contributors, many of whom are volunteers. Anyone can contribute to the GNOME!

Latest news

[March 29, 2016](#)
[March 29, 2016](#)
[February 19, 2016](#)

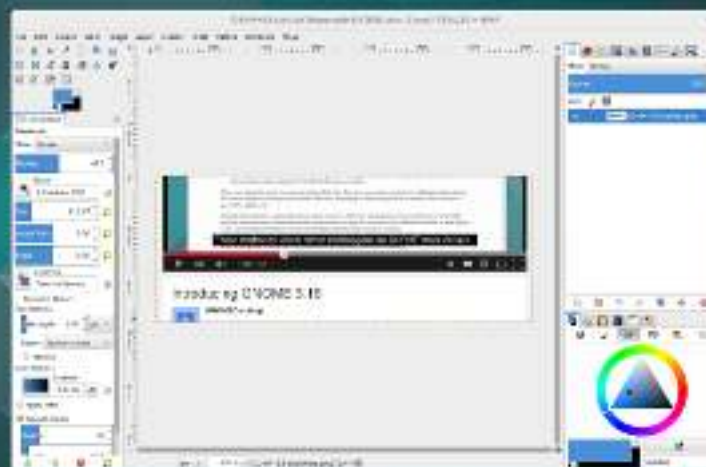
Activities overview



Type to search...



Edit Post | Barbican - Barbican Messages - WordPress



[01-04-15 subtitle] (Imported)-3.0 (RGB color, 1 layer) 730x382 - GIMP



Notifications



rupert: could I borrow your car for a while?

HowDoI/GtkApplication - GNOME Wiki!

<http://wiki.gnome.org/HowDoI/GtkApplication>

`GtkApplication` is the base class of a `Gtk Application`. Its primary purpose is to separate your program from `main()`.

`main()` is an operating system implementation detail that is really uninteresting to applications. The philosophy of `GtkApplication` is that applications are interested in being told what needs to happen, when it needs to happen, in response to actions from the user. The exact mechanism by which the operating system starts applications is uninteresting.

To this end, `GtkApplication` exposes a set of signals (or virtual functions) that an application should respond to.

- `startup`: sets up the application when it first starts
- `shutdown`: performs shutdown tasks
- `activate`: shows the default first window of the application (like a new document). This corresponds to the application being launched by the desktop environment.
- `open`: opens files and shows them in a new window. This corresponds to someone trying to open a document (or documents) using the application from the file browser, or similar.

When your application starts, the `startup` signal will be fired. This gives you a chance to perform initialisation tasks that are not directly related to showing a new window. After this, depending on how the application is started, either `activate` or `open` will be called next.

`GtkApplication` defaults to applications being single-instance. If the user attempts to start a second instance of a single-instance application then `GtkApplication` will signal the first instance and you will receive additional `activate` or `open` signals. In this case, the second instance will exit immediately, without calling `startup` or `shutdown`.

For this reason, you should do essentially no work at all from `main()`. All startup initialisation should be done in `startup`. This avoids wasting work in the second-instance case where the program just exits immediately.

The application will continue to run for as long as it needs to. This is usually for as long as there are any open windows. You can additionally force the application to stay alive using `g_application_hold()`.

Loading "wiki.gnome.org"...

... you receive a `shutdown` signal where you can do any necessary cleanup tasks (such as saving files to disk)

Notifications

rupert x

Bastian 13:59
 Release Beers March the 28th

Sunday
February 22 2015

February

S	M	T	W	T	F	S
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

Add world clocks...

to actions from

To this end, GtkA

- startup: se
- shutdown: p
- activate: s
- being laun
- open: opens
- documents

When your applic

not directly relat

open will be calle

GtkApplication d

instance applicat

In this case, the

For this reason, you should do essentially no work at all from `main()`. All startup initialisation should be done in `startup`. This avoids wasting work in the second-instance case where the program just exits immediately.

The application will continue to run for as long as it needs to. This is usually for as long as there are any open windows. You can additionally force the application to stay alive using `g_application_hold()`.

On shutdown, you receive a `shutdown` signal where you can do any necessary cleanup tasks (such as saving files to disk).

GtkApplication does not implement `main()` for you. You must do so yourself. Your `main()` function should be as small as possible and do almost nothing except creating your `GtkApplication` and running it. The "real work" should always be done in response to the signals fired by `GtkApplication`.

Integrated messaging



rupert

do you have some music recommendations?

sure, what genre? 14:01

could I borrow your car for a while?

what if

GtkApplication is the base

am from `main()`.

`main()` is an operating system implementation detail that is really uninteresting to applications. The philosophy of `GtkApplication` is that applications are interested in being told what needs to happen, when it needs to happen, in response to actions from the user. The exact mechanism by which the operating system starts applications is uninteresting.

To this end, `GtkApplication` exposes a set of signals (or virtual functions) that an application should respond to.

- `startup`: sets up the application when it first starts
- `shutdown`: performs shutdown tasks
- `activate`: shows the default first window of the application (like a new document). This corresponds to the application being launched by the desktop environment.
- `open`: opens files and shows them in a new window. This corresponds to someone trying to open a document (or documents) using the application from the file browser, or similar.

When your application starts, the `startup` signal will be fired. This gives you a chance to perform initialisation tasks that are not directly related to showing a new window. After this, depending on how the application is started, either `activate` or `open` will be called next.

`GtkApplication` defaults to applications being single-instance. If the user attempts to start a second instance of a single-instance application then `GtkApplication` will signal the first instance and you will receive additional `activate` or `open` signals. In this case, the second instance will exit immediately, without calling `startup` or `shutdown`.

For this reason, you should do essentially no work at all from `main()`. All startup initialisation should be done in `startup`. This avoids wasting work in the second-instance case where the program just exits immediately.

The application will continue to run for as long as it needs to. This is usually for as long as there are any open windows. You can additionally force the application to stay alive using `g_application_hold()`.

Desktop search with search providers



4 * 4



4 * 4 = 16



g8443



g8054



CIS224: Software Projects: Software Engineering and Research Methods



Search the Web for 4 * 4

And more!

- 🐾 Topic-oriented help
 - 🐾 Intuitive system settings
 - 🐾 ...
-
- 🐾 Many small features to improve your experience!





Under the hood

- 🐾 Fast and powerful platform, usable from several languages: C, C++, Python, JavaScript, Vala, ...
- 🐾 Easy theming with CSS - overhauled
- 🐾 Symbolic icons



How it all started

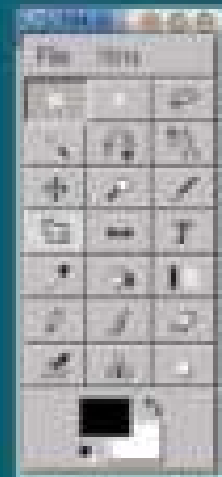
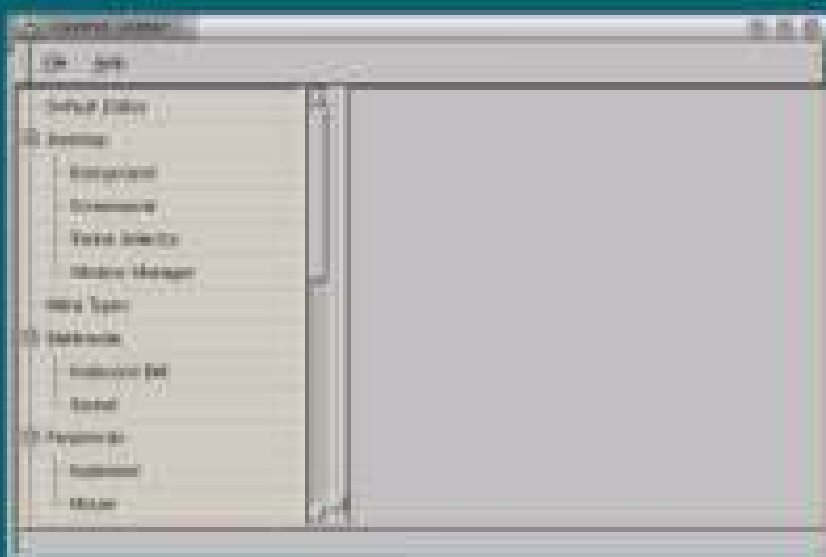
- 🐾 Proposal at GUADEC in Istanbul 2008
- 🐾 Hackfest in Boston
- 🐾 Mockups



```

<title>#aprendiendo a #aprender Linux/</title>
<subtitle>Guías para colegios con plataformas de referencia &#160;PLATAFORMA;</subt
itle>
<bookinfo>
  <title>#aprendiendo a #aprender Linux/</title>
  <subtitle>Guías para colegios con plataformas de referencia &#160;PLATAFORMA;</subt
itle>
  <author><firstname>Jaime Irving</firstname><surname>Dávila</surname></author>
  <author><firstname>Pablo</firstname><surname>Chamorro</surname></author>
  <author><firstname>Igor</firstname><surname>Támara</surname></author>
  <author><firstname>Vladimir</firstname><surname>Támara</surname></author>
</bookinfo>
<para>
  TRABAJO CEDIDO AL DOMINIO PÚBLICO
</para>
<para>Nosotros Jaime Irving Dávila, Pablo Chamorro, Igor Támara y
  Vladimir Támara hemos escrito
  "Aprendiendo a aprender Linux: Guías para colegios con plataforma
  de referencia &#160;PLATAFORMA;", que asisten al uso, la instalación y
  la administración de redes Linux en colegios. Por esta webdo
  cedamos esta obra al dominio público, renunciando a todos los derechos
  patrimoniales con lo que esperamos facilitar su adaptación y uso en
  colegios.

```





Butterfly

Search | Text | Tools | Delete

Butterfly

Kingdom: Animalia
 Phylum: Arthropoda
 Class: Insecta
 Order: Lamiaceae

- Superfamily
 - **Lycaenoidea**
- Superfamily
 - **Lycaenoidea**
- Superfamily
 - **Lycaenoidea**

Hedylidae

Search | Text | Tools | Delete

Skippers differ in several important ways from the remaining butterflies, which are classified in the superfamily **Lycaenoidea** and the neotropical superfamily **Hedylidae**. Collectively, these three groups of butterflies share several characteristics especially at the egg, larval and pupal stage (Ackery et al, 1998). However, skippers have the antennae clubs hooked backward like a crochet, whilst butterflies have club-like tips to their antennae and hedylids have feathered or pectinate antennae giving them an even more 'moth-like' appearance than skippers. Skippers also have generally stockier bodies than the other two groups, with stronger wing muscles. **Hesperioidea** is very likely the sister group of **Lycaenoidea**, and together with **Hedylidae** constitute a natural group or clade.

0724-094430.jpg

File Edit View Image Go Help

Previous Next In Out Normal

720 x 480 pixels 88.8 KB 61%

3 / 1

Applications



Firefox Web Browser

Browse the World Wide Web



Evolution Mail

Read and write email



Text Editor

Edit text files



Empathy Instant Messe...

Send and receive instant



Movie Player

Play movies and songs

More...

Recent Documents



M.I.A.

Kala



UME Guide

mobileguide



Tube Map

Transport for London



Aigo MID Running RF_2

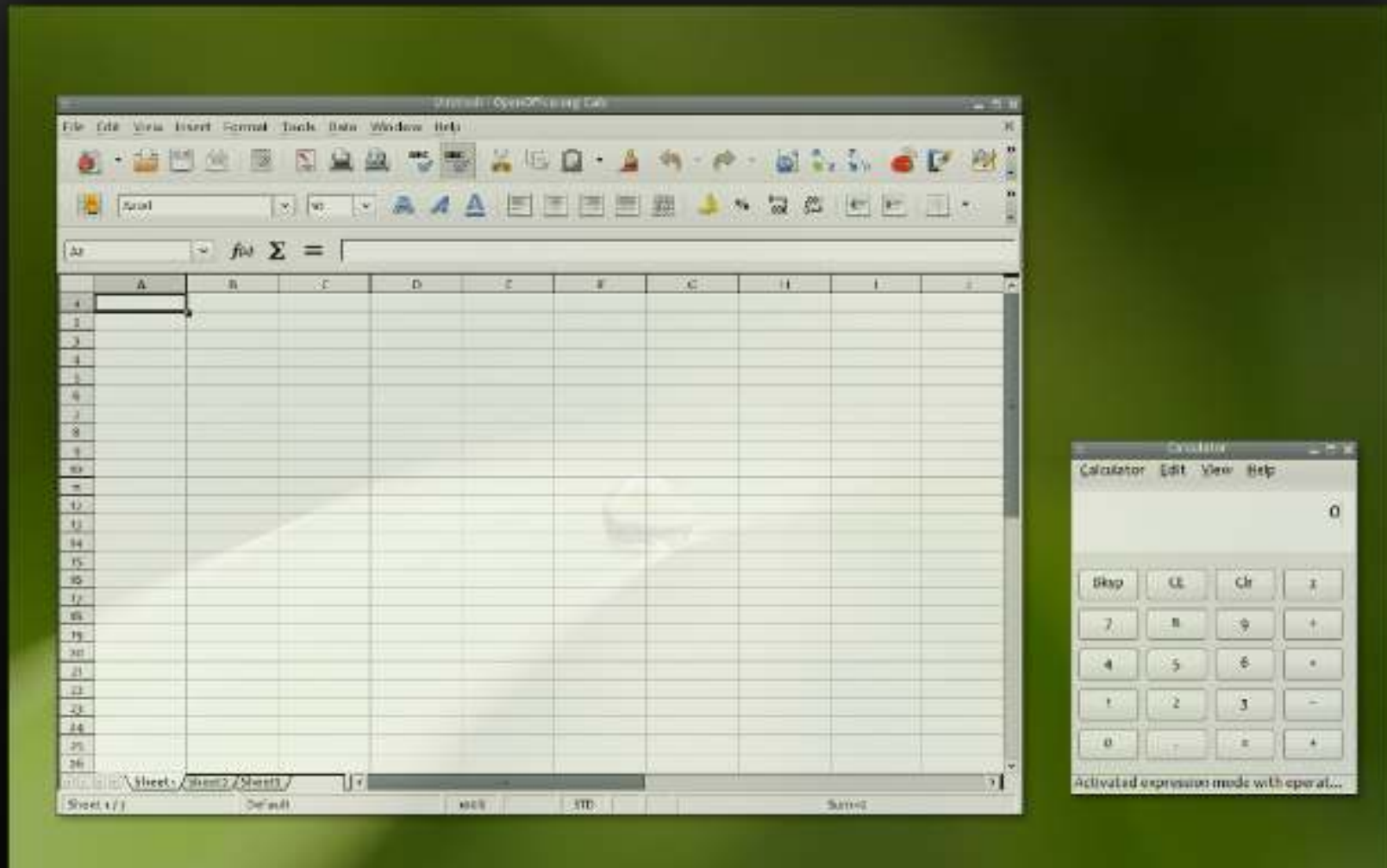
Mid Videos



Earth Horizon
















Wallpapers

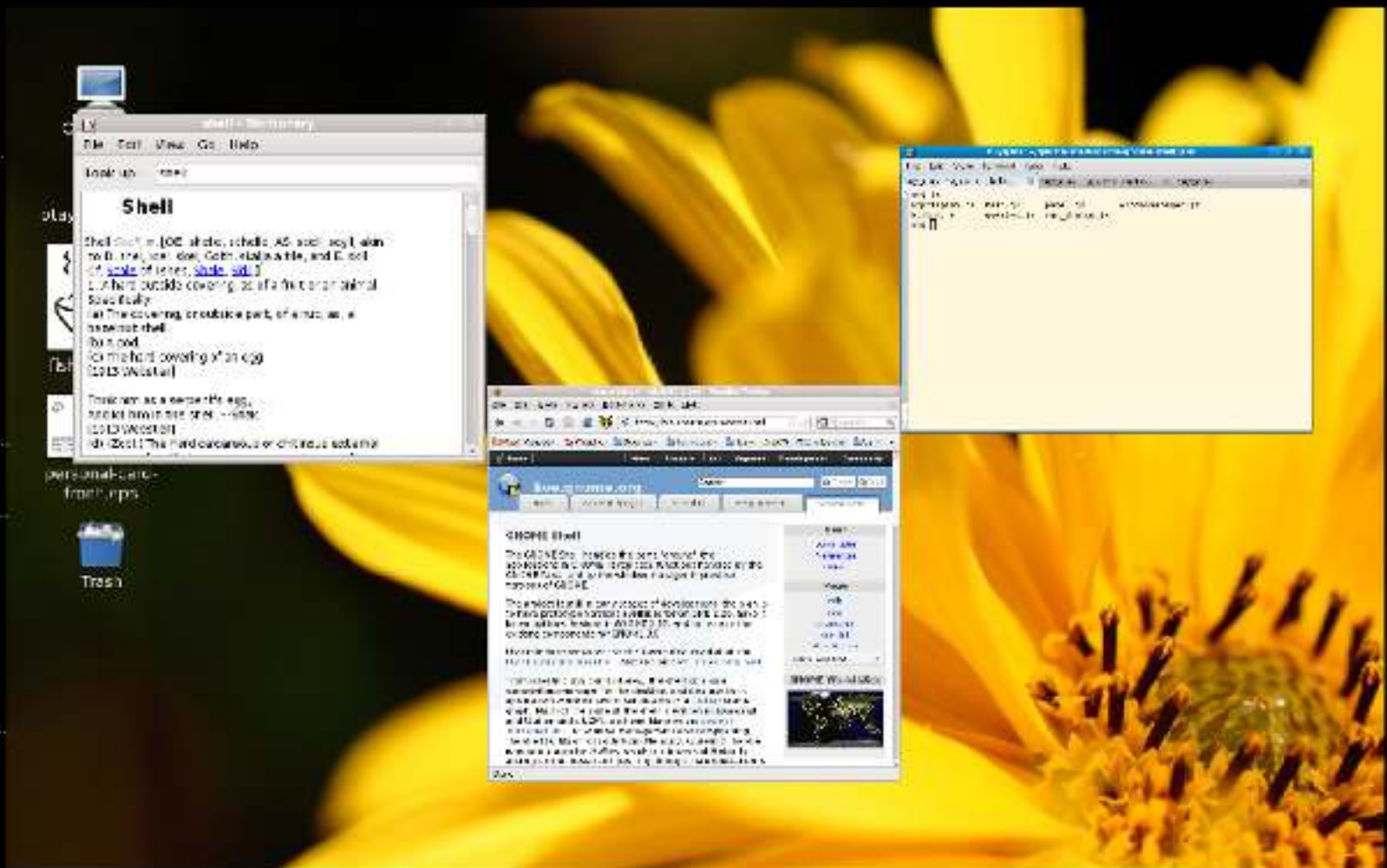
More...



Actual code



-  Applications
-  Software Updates
Change software update preferences
-  Help
Get help with GNOME
-  Phone Manager
Receive and send text messages from y...
-  Autorun Prompt
-  Emacs Text Editor
Edit text
-  OpenJDK Policy Tool
Manage OpenJDK policy files
-  Disk Usage Analyzer
Check folder sizes and available disk spa...
-  Input Method
Choose the input method to be used or ...
-  Control Center
-  About Me
Set your personal information
-  Evolution Mail and Calendar
Manage your email, contacts and sched...
-  Bluetooth
Configure Bluetooth settings
-  Devhelp
Developer's Help program
-  Personal File Sharing
Preferences for sharing of personal files



Find a name ;-)

“Current leader for a project name is the exciting “gnome-shell”

gnomesh

gnomecc



Make it a brand



Make it a brand



Make it a brand



2011-04-06 Release 3.0!



Party!!1



~150 Parties in 48 (sic!) countries

Promotional material produced and shipped



Evolving GNOME 3

<http://extensions.gnome.org>

GNOME Tweak Tool



From GNOME 3.0 to 3.20

1811 days

339189 changes



GNOME releases

 2.2: 2003-02-06

 ...

 2.26: 2009-03-18

 2.28: 2009-09-23

 2.30: 2010-03-31

 2.32: 2010-09-29

 3.0: 2011-04-06

 3.2: 2011-09-28

 3.4: 2012-03-28

 3.6: 2012-09-26


 3.8: 2013-03-27


 3.10: 2013-09-25

 3.12: 2014-03-26

 3.14: 2014-09-24

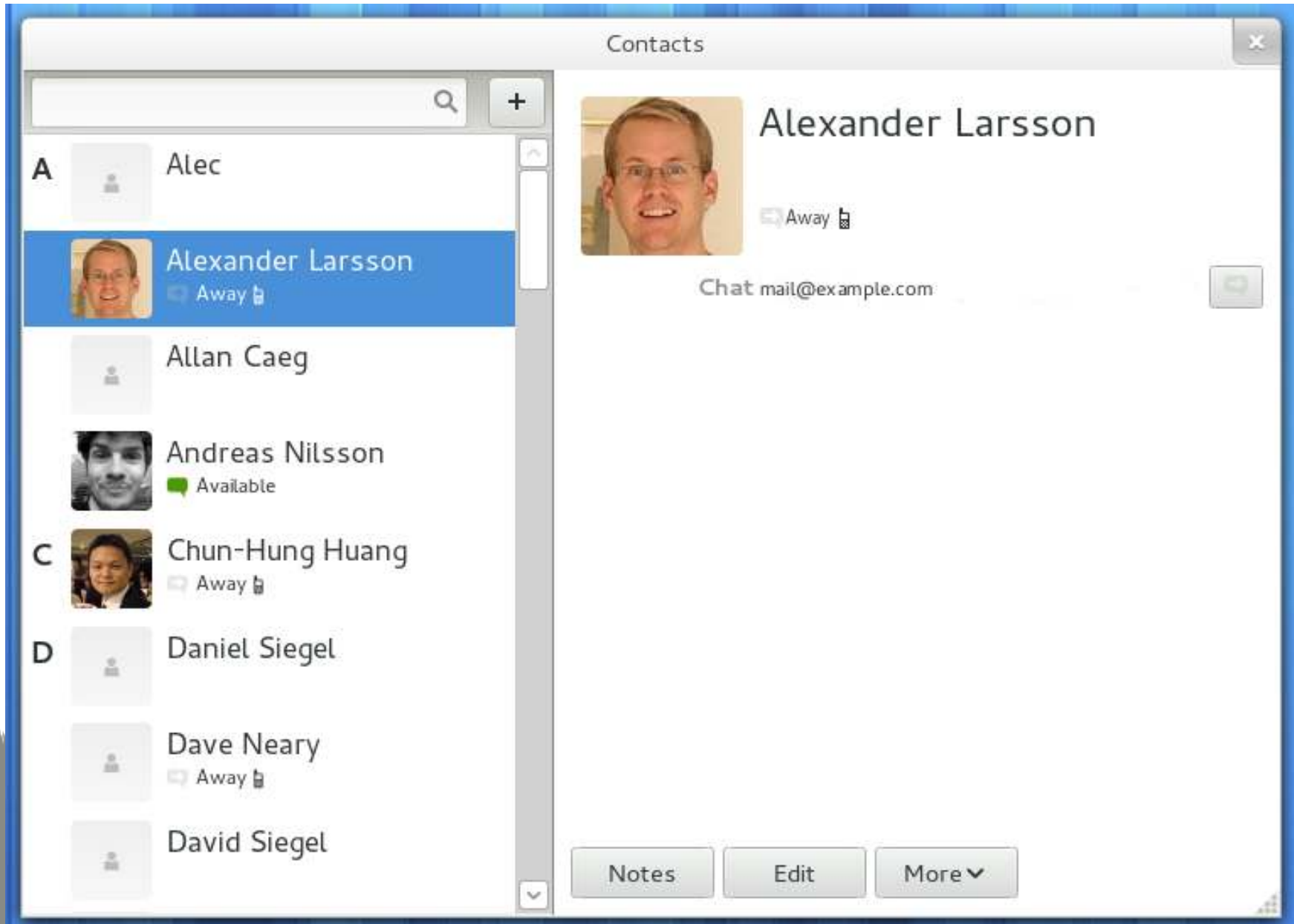
 3.16: 2015-03-25

 3.18 (G'burg): 2015-09-21

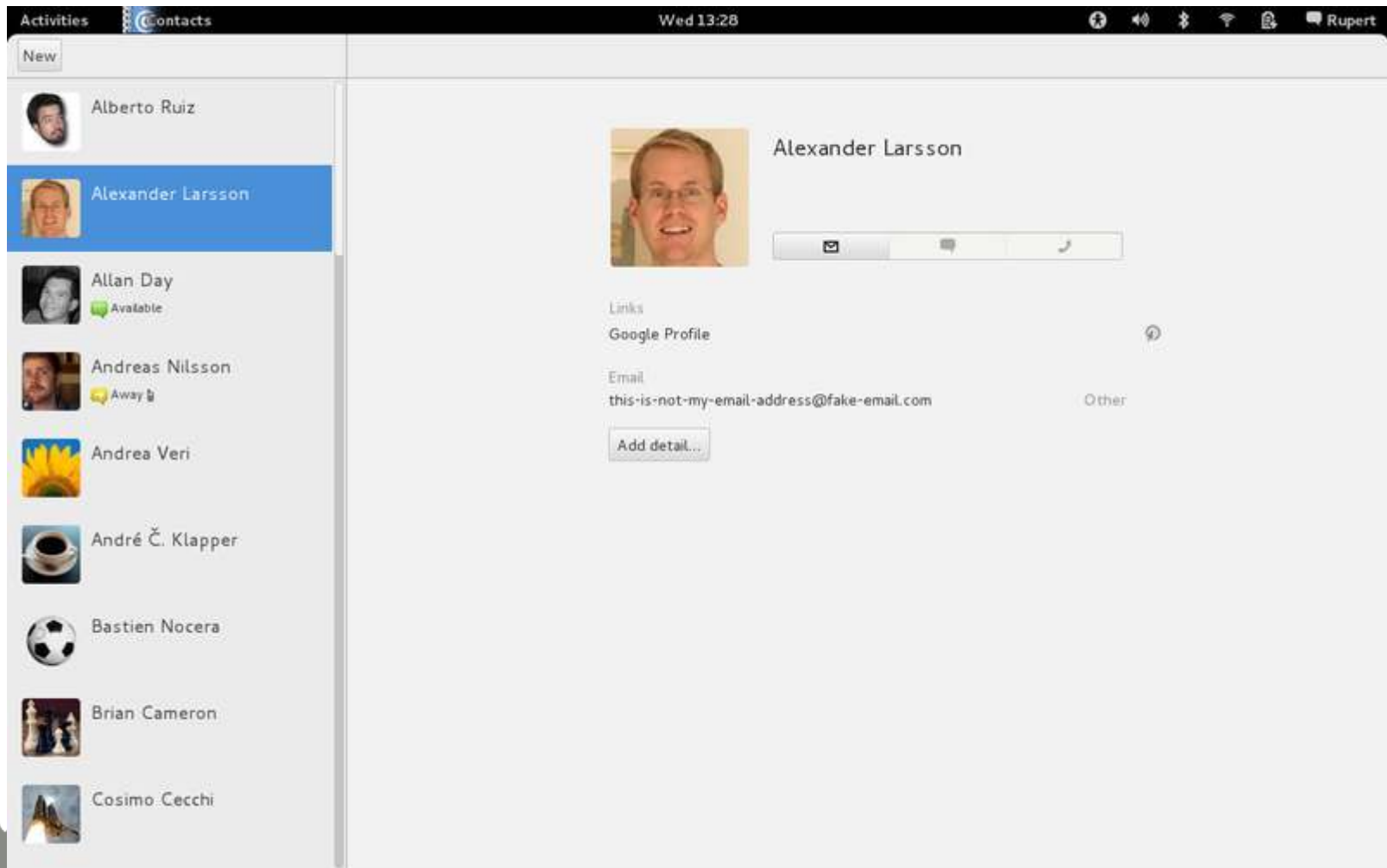
 3.20 (Delhi): 2016-03-21



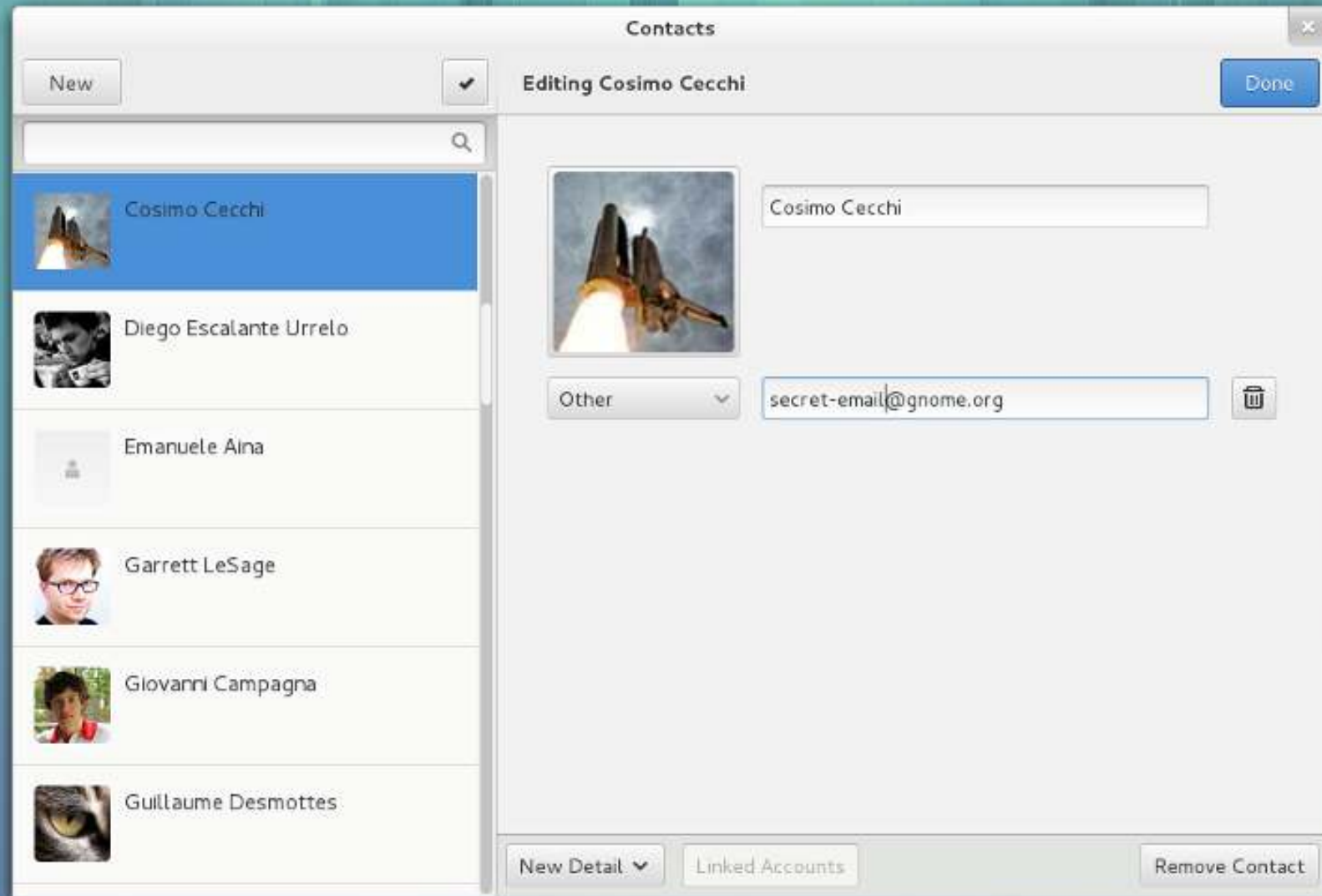
The story of contacts (3.2)



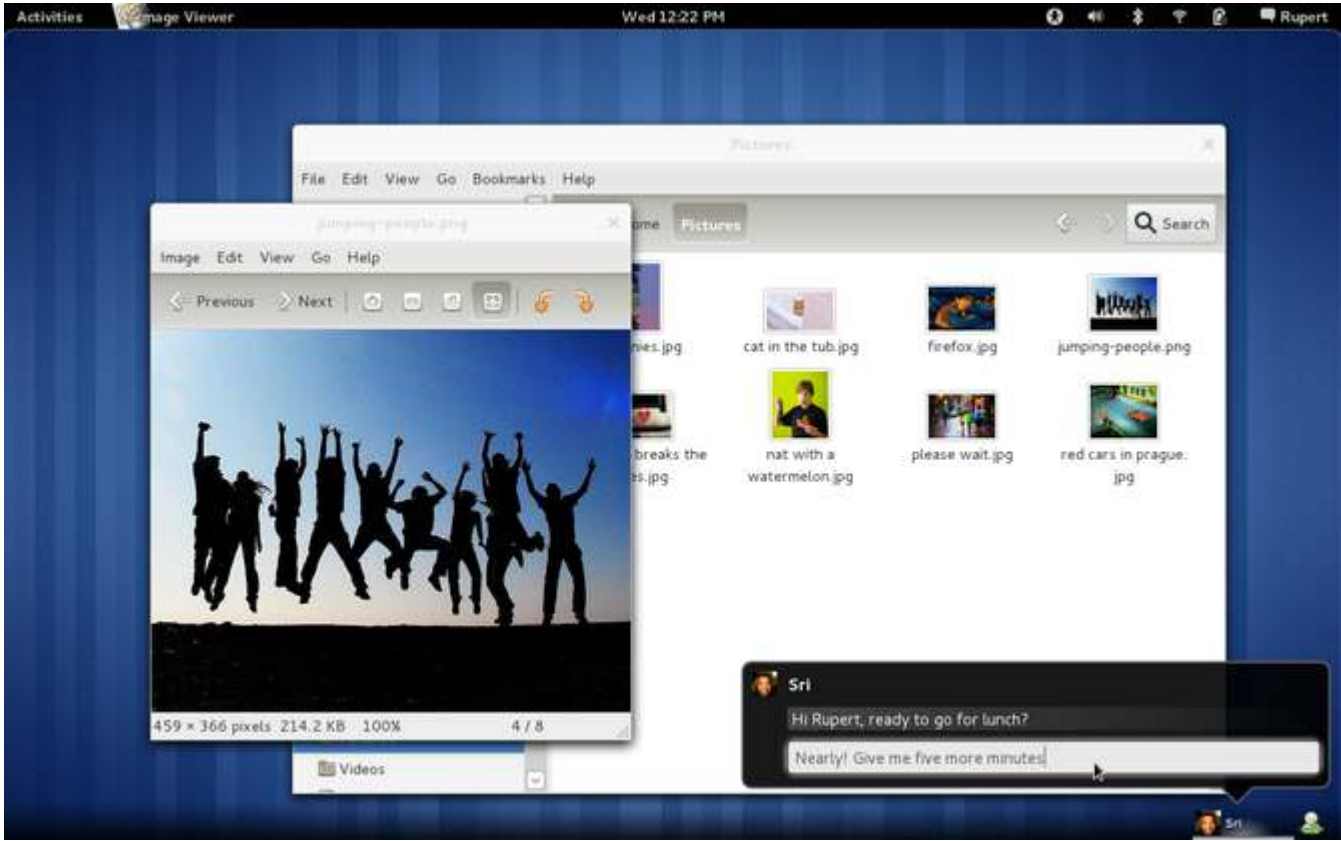
The story of contacts (3.4)



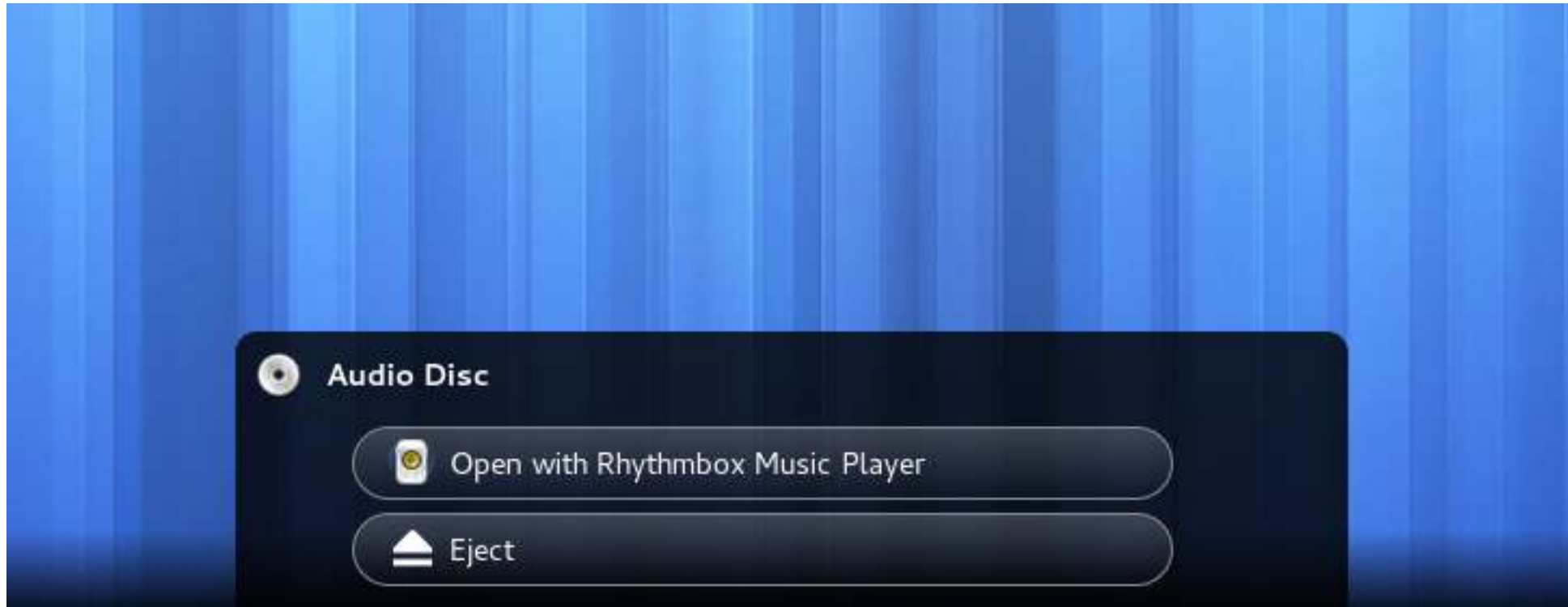
The story of contacts (3.8)



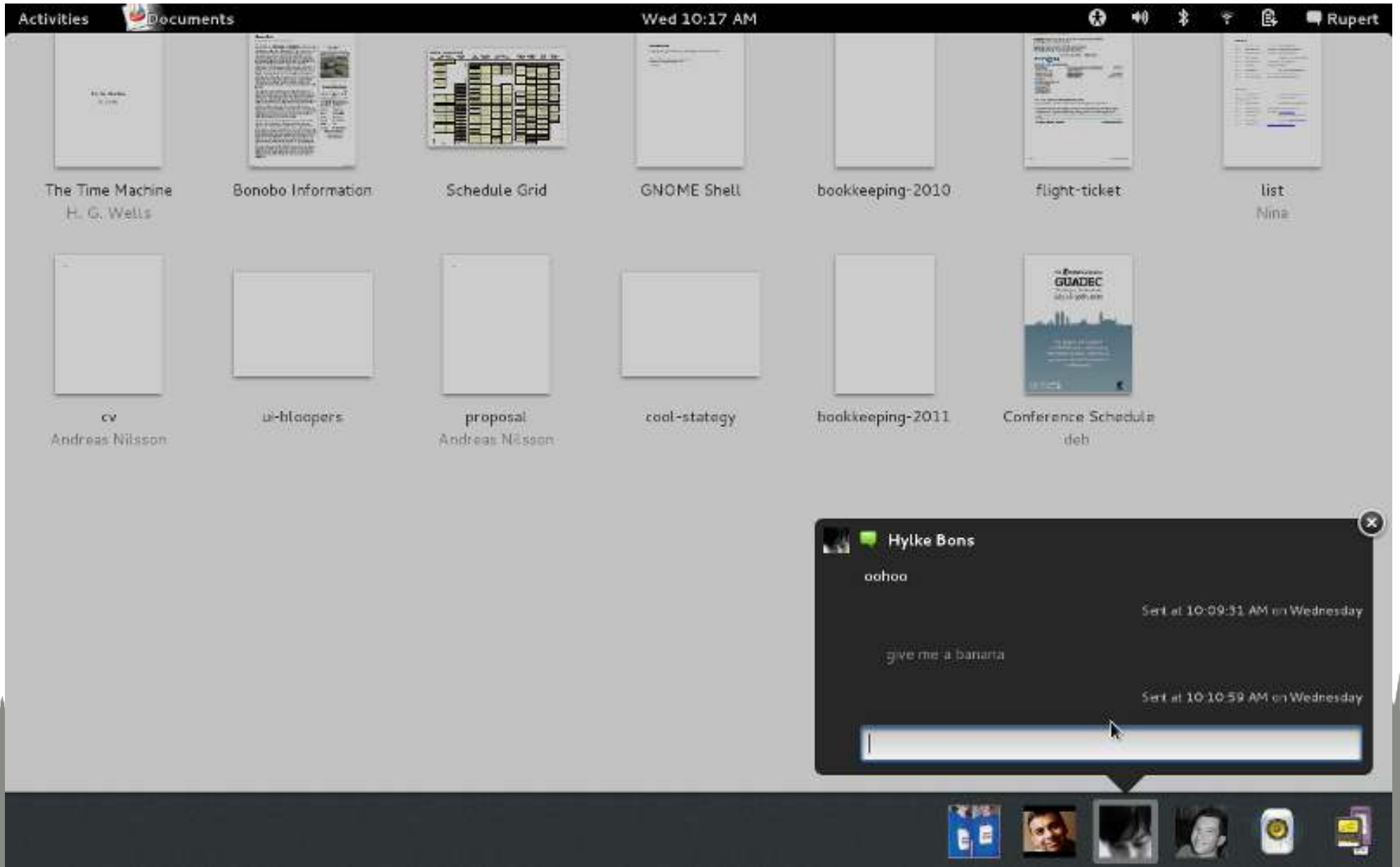
The story of notifications



The story of notifications



The story of notifications



The story of notifications



The story of notifications

Tue 14:50

Media



Jim Guthrie
The Prettiest Remix



Tuesday

March 15 2016



March



S	M	T	W	T	F	S
28	29	01	02	03	04	05
06	07	08	09	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	01	02
03	04	05	06	07	08	09

Add world clocks...

GNOME 3.20 on March 21th



GNOME 3.20

Approximately 772 people made about 25112 changes to GNOME

- 🐾 New notification system
- 🐾 Updated shell visuals
- 🐾 Improved developer experience
- 🐾 ...





GNOME is people







GNOME™