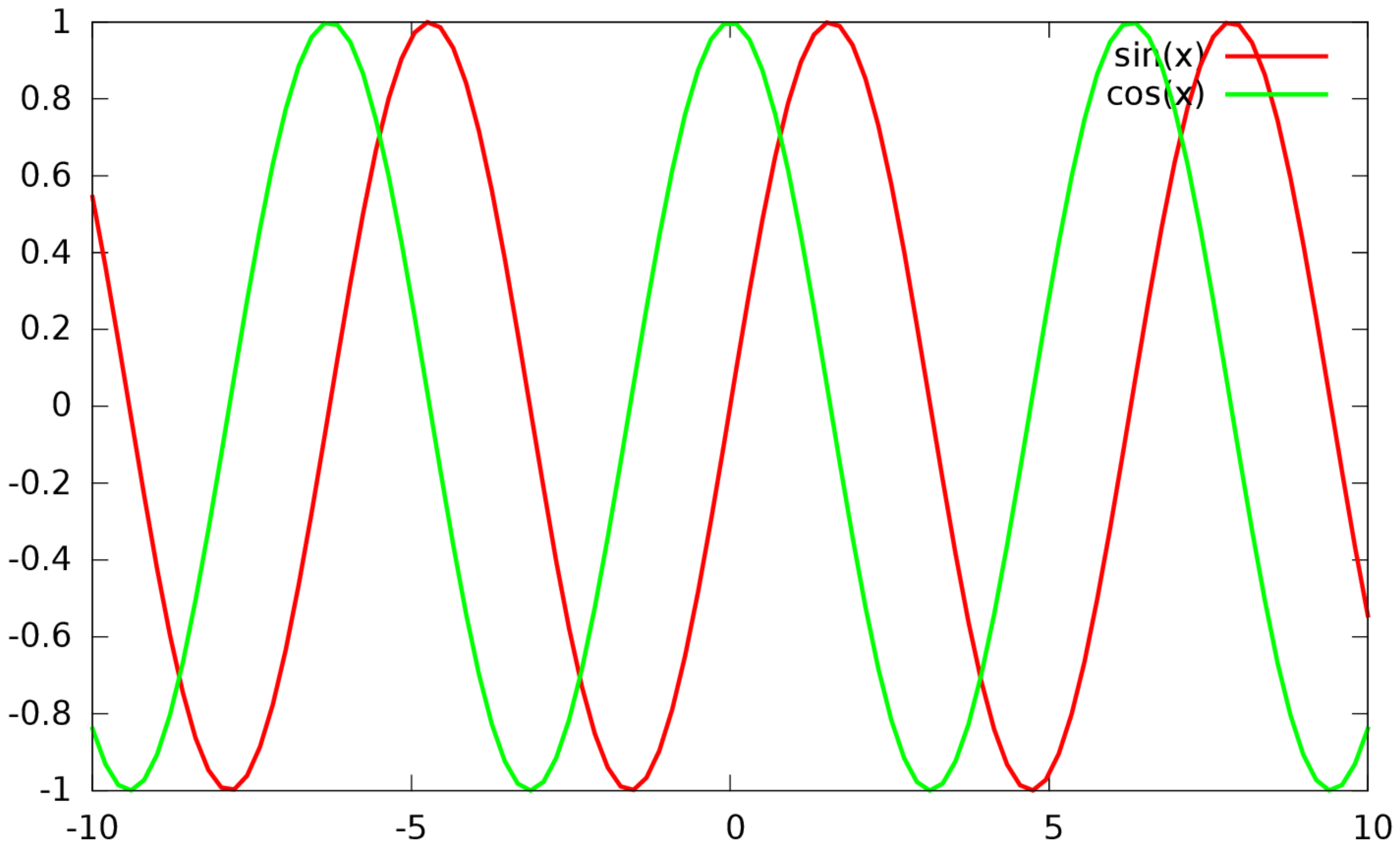
Designing accessible applications

Samuel Thibault
Slides & stuff on
http://brl.thefreecat.org/

http://liberte0.org/

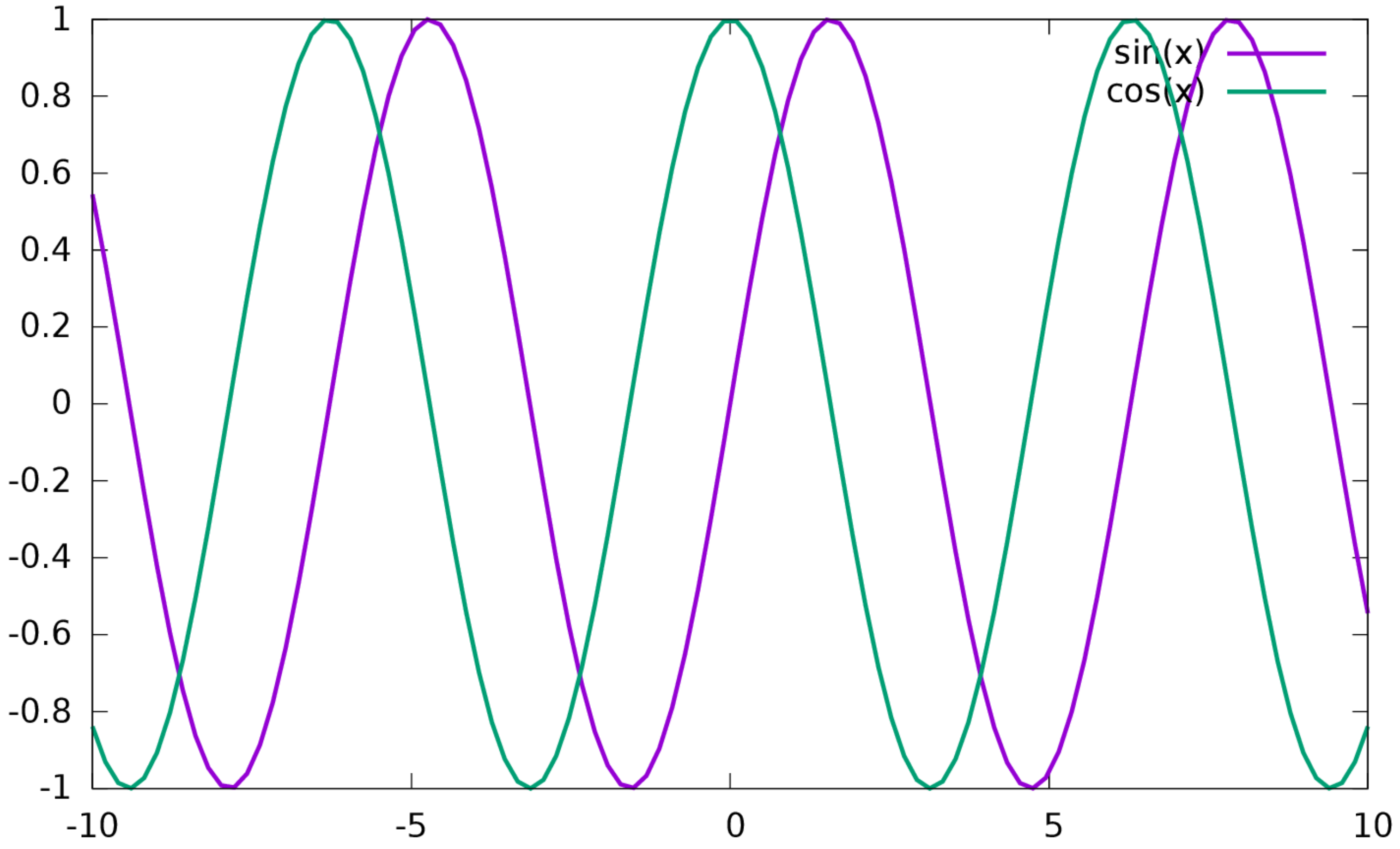
Gnuplot



Color blindness: 8% male, 0.5% female

⡙⠑⠃⠊⠁⠝

Gnuplot 5!!



Color blindness: 8% male, 0.5% female

What is accessibility?

AKA a11y

Usable by people with specific needs

- Blind
- Low vision
- Deaf
- Colorblind
- One-handed

- Cognition (dyslexia, attention disorder, memory, ...)
- Motor disability (Parkinson, ...)
- Elderly

See Accessibility HOWTOs

- You

“Handicap” depends on the situation and is not necessarily permanent

10% handicapped – 20% limited

This is all about freedom #0

“The freedom to run the program, for any purpose”

What about being *able to use* the program?

- RMS said a11y was just a “desirable feature”.
  - “Desirable” only, really?
- RMS said “this is free software, you can modify it” (freedom #1)
  - Can. Not. Happen.

UNO rights of persons with disabilities

"Discrimination on the basis of disability" means any distinction, exclusion or restriction on the basis of disability which has the purpose or **effect of impairing** or nullifying the recognition, enjoyment or **exercise**, on an equal basis with others, **of all human rights and fundamental freedoms** in the political, economic, social, cultural, civil or any other field. It includes all forms of discrimination, **including denial of reasonable accommodation**

"Reasonable accommodation" means necessary and appropriate modification and adjustments **not imposing a disproportionate or undue burden**, where needed in a particular case, to ensure to persons with disabilities the enjoyment or exercise on an equal basis with others of all human rights and fundamental freedoms;

A question of priority

- Should be prioritized
  - Just like internationalization

A question of who doing it

- Concerns only a small fraction of population
    - Already a hard time using computers...
    - Almost nobody with both disabilities and programming skills (and very difficult to work)
    - Even fewer people with awareness and programming skills
        → “This is free software, you can modify it” can not work.
- Support has to be integrated
    - Distributed among maintainers themselves
    - Not borne by the tiny a11y community

Why making GUI accessible?

(when textmode seems so easier to make accessible)

- A lot of stuff is not available in textmode
  - e.g. real javascript support
- Business applications
- Non-tech people need to get help from non-tech people around

Dedicated software?

- e.g. edbrowse, a blind-oriented editor/browser
- Generally a bad idea!
  - Oriented to just one disability
  - Lack of manpower
    - e.g. Web browser
      - javascript/flash/table/CSS support?
    - e.g. An office suite
      - MSOffice/OpenOffice compatibility?
  - Disabled & non-disabled working together
    - Better use the same software

➔ Better make **existing** applications accessible

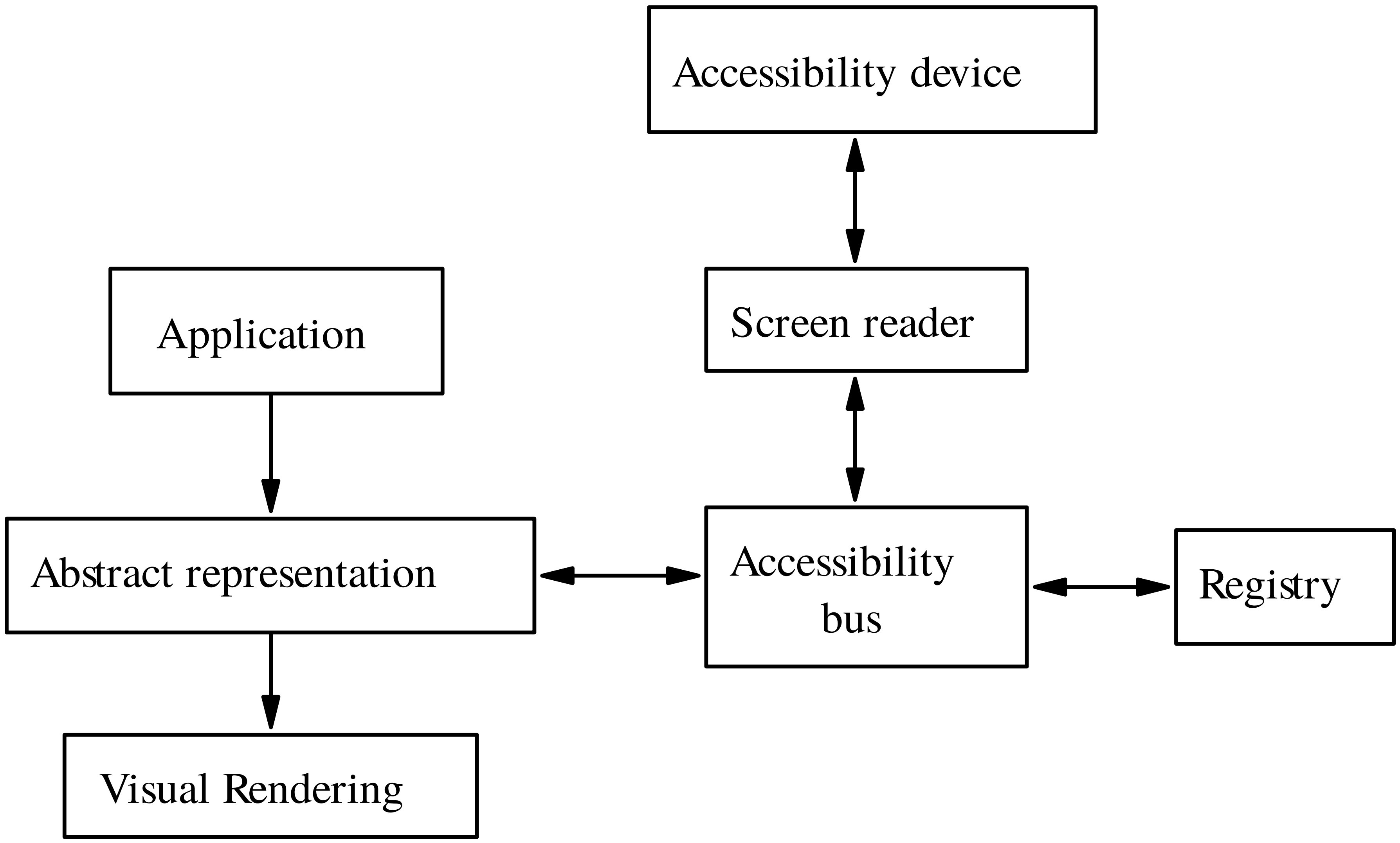
Design principles

- Same software, made accessible
  - Understand each other, get help, etc.
- Synchronized work
  - Just alternate input/output
  - Being able to work together
- Pervasive
  - Shouldn't have to ask for software installation / configuration

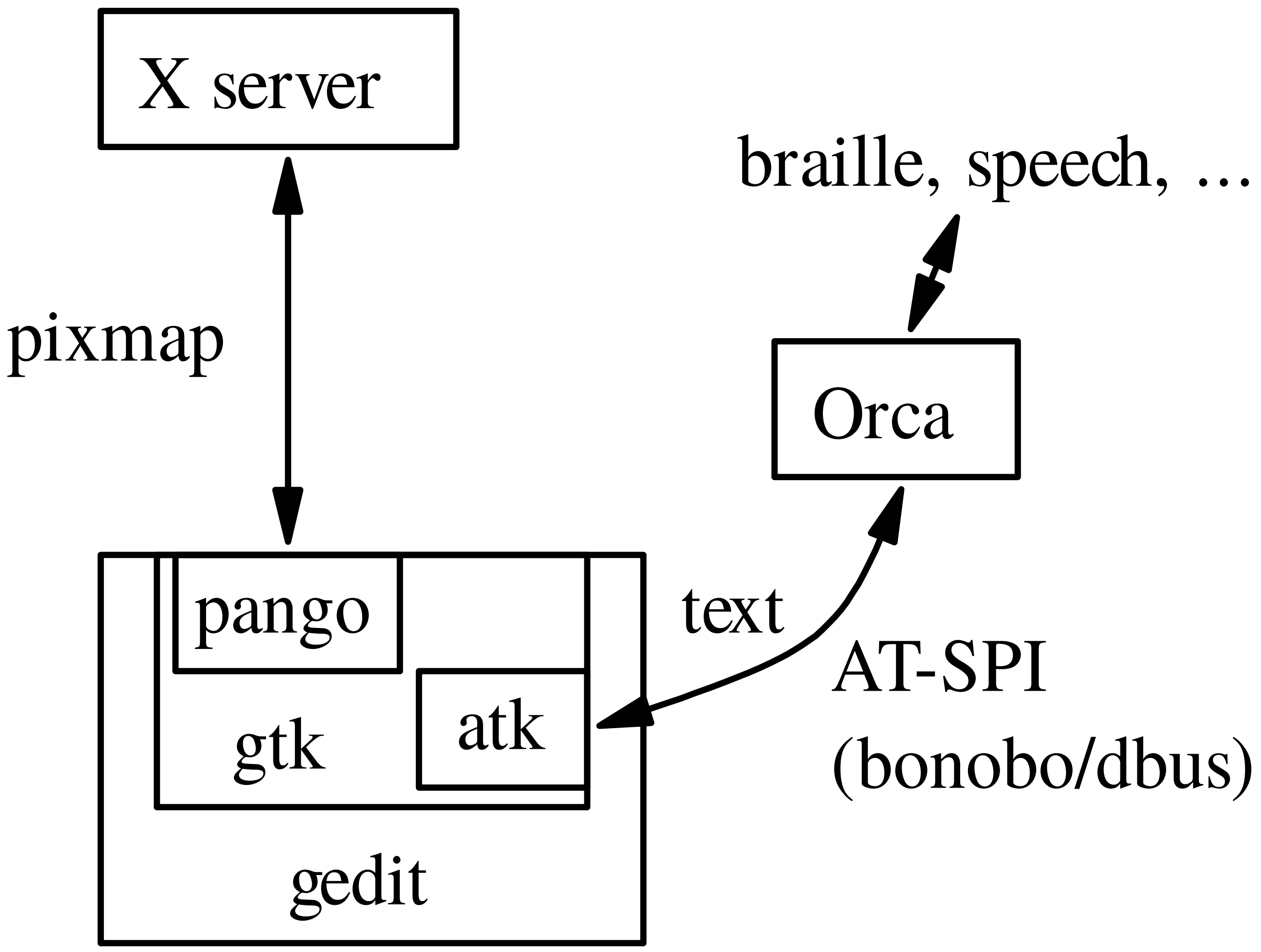
Status in a few words

- Text mode is generally quite well accessible
  - But not so well suited to beginners
- Gnome quite accessible
  - Gnome 3 was however almost a restart-from-scratch
- We're late compared to the Windows world
  - We started less than a dozen years ago
  - They started a couple of decades ago
- We're Stone Age compared to the Apple world
  - Really *good* and *integrated* support

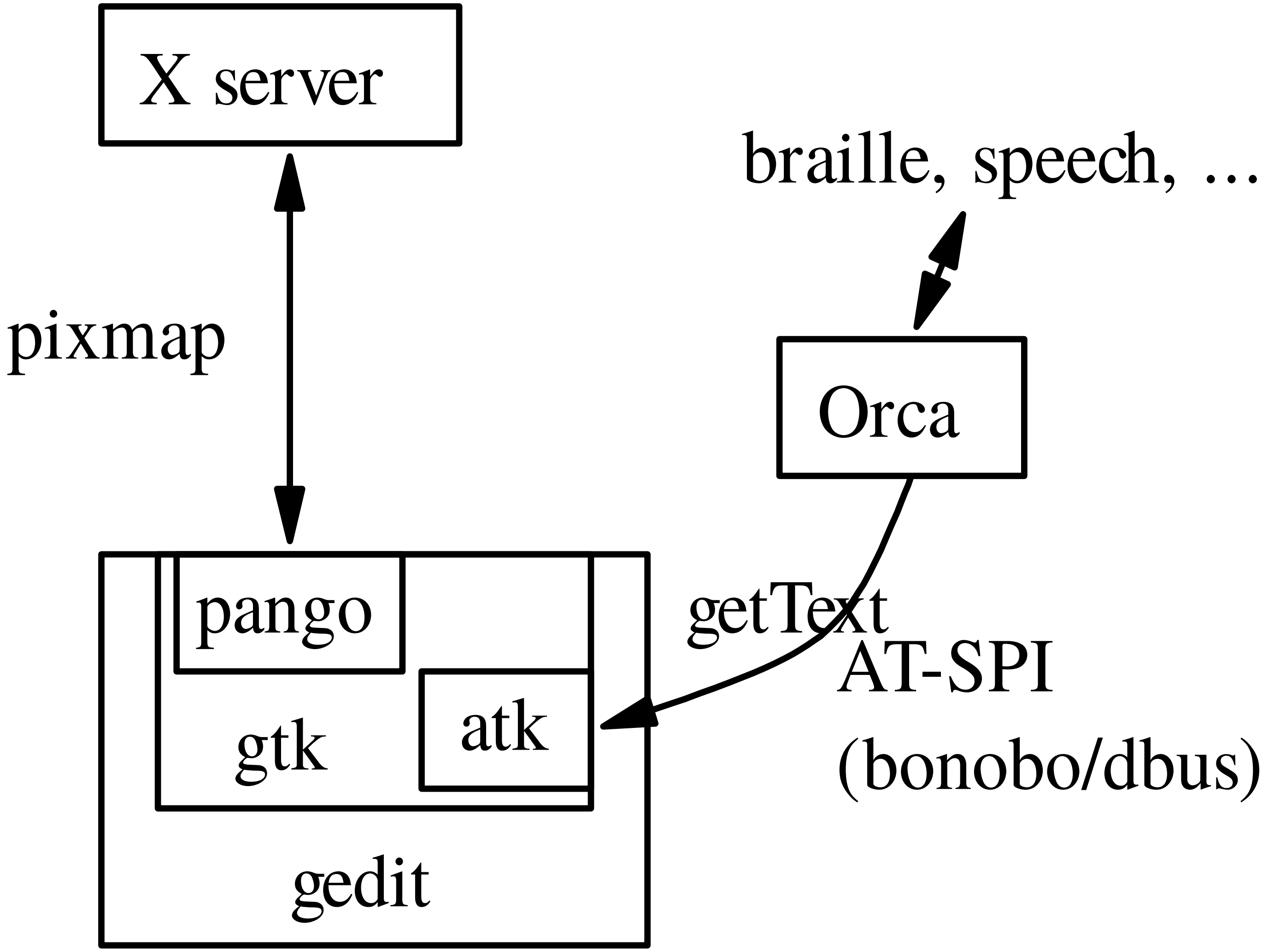
Generic methodology



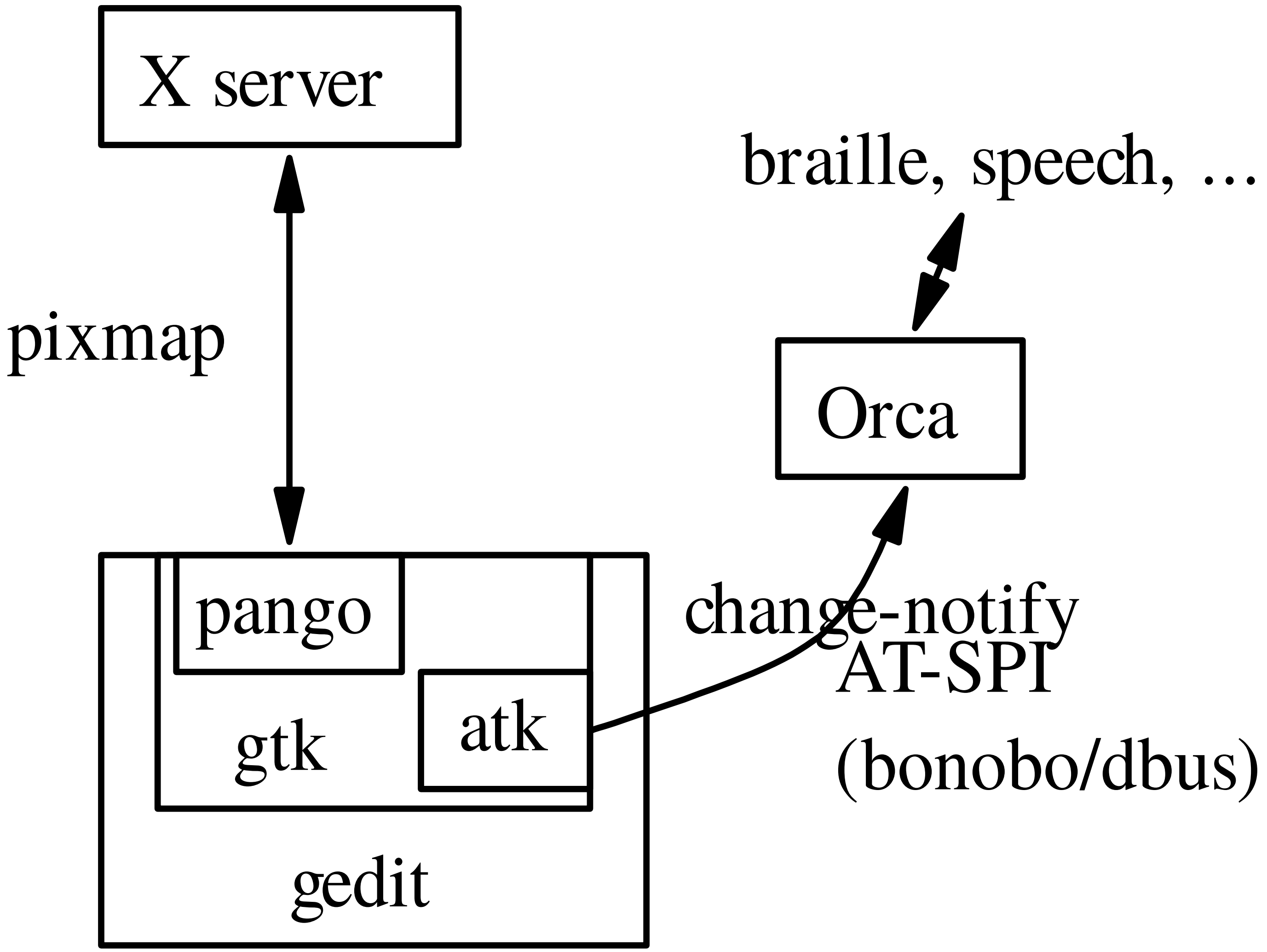
X accessibility, AT-SPI RPCs



X accessibility, AT-SPI RPCs



X accessibility, AT-SPI RPCs



Abstract representation

- Window
  - Vertical container
    - Menu bar
      - File Menu
        - Open Menu Item
        - …
      - ...
    - Horizontal container
      - Text area
      - Ok button

Technically speaking

A lot of applications already *technically* accessible

- Console
- GTK2/3
- KDE-Qt4 sketchy, Qt5 improving
- Java Swing
- Acrobat Reader

A lot are not

- Mono?
- KDE-Qt3
- Xt
- Self-drawn (e.g. xpdf)

⡙⠑⠃⠊⠁⠝

In practice

- A lot of technically-accessible applications actually aren't really usable
  - A visually-organized mess of widgets...

First name:	Foo
Last name:	Bar
Password:	baz

In practice

- A lot of technically-accessible applications actually aren't really usable
  - A visually-organized mess of widgets...

First column
- Label First Name
- Label Last Name
- Label Password

Second column
- Text Foo
- Text Bar
- Text baz

In practice

- A lot of technically-accessible applications actually aren't really usable
    - A visually-organized mess of widgets...

- Label First Name for Text Foo
- Label Last Name for Text Bar
- Label Password for Text baz

In practice

- A lot of technically-accessible applications actually aren't really usable
  - A visually-organized mess of widgets...

First column
- Label First Name
- Label Last Name
- Label Password

Second column
- Text Foo
- Text Bar
- Text baz

In practice

- A lot of technically-accessible applications actually aren't really usable
  - A visually-organized mess of widgets...

First column
- Label First Name
- Label Last Name
- Label Password

Second column
- Text Foo
- Text Bar
- Text baz

➔ Screen reader “Script” for each application

⡙⠑⠃⠊⠁⠝

Don't try to make applications accessible,
just make accessible applications

Quite often just a matter of
common sense from the start

Not a reason for not fixing
your existing apps of course,
it will just be a bit harder :)

Text applications

- Usually work really great for braille output
- Always provide such equivalent of graphical applications, e.g. based on same shared lib
  - Useful for servers via ssh too!
- The default output of screen readers is what the cursor is on
  - Works great with shell, editor, etc.
  - Doesn't work so great with semigraphical apps

➔ Put the cursor appropriately!

- Even when invisible, e.g. mutt, aumix

Graphical applications

- Design your application **without** gui in mind first
  - Logical order, just like CSS ☺
- Use standard widgets
  - e.g. *labeled* text fields
  - Avoid homemade widgets, or else implement atk yourself for them
  - Always provide alternative textual content for visual content
- Keep it simple!
  - Not only to make screen reading easier, but to make life easier for all users too!

⡙⠑⠃⠊⠁⠝

Some pitfalls and advices

(from the accessibility howtos)

- Shouldn't *have* to use the mouse for anything
- Care of contrasts, configurable colors
- Avoid timing-based actions, or make them configurable
- No 2D organization, logical organization
- Keep it simple and obvious
- ...

⡙⠑⠃⠊⠁⠝

Tools

⡙⠑⠃⠊⠁⠝

Test it yourself! (textmode)

Brltty + gnome-terminal

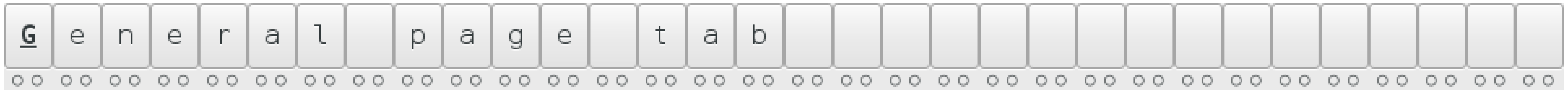
- see doc on http://brl.thefreecat.org

Documentations

- Accessibility HOWTOs
  - Quite old, but still very useful advices
- Gnome Accessibility devel guide
  - For GTK applications

Test it yourself! (GUIs)

orca -e braille-monitor


General page tab

- Then work as usual
- Only using keyboard
- Checking text appears there

And crash-test

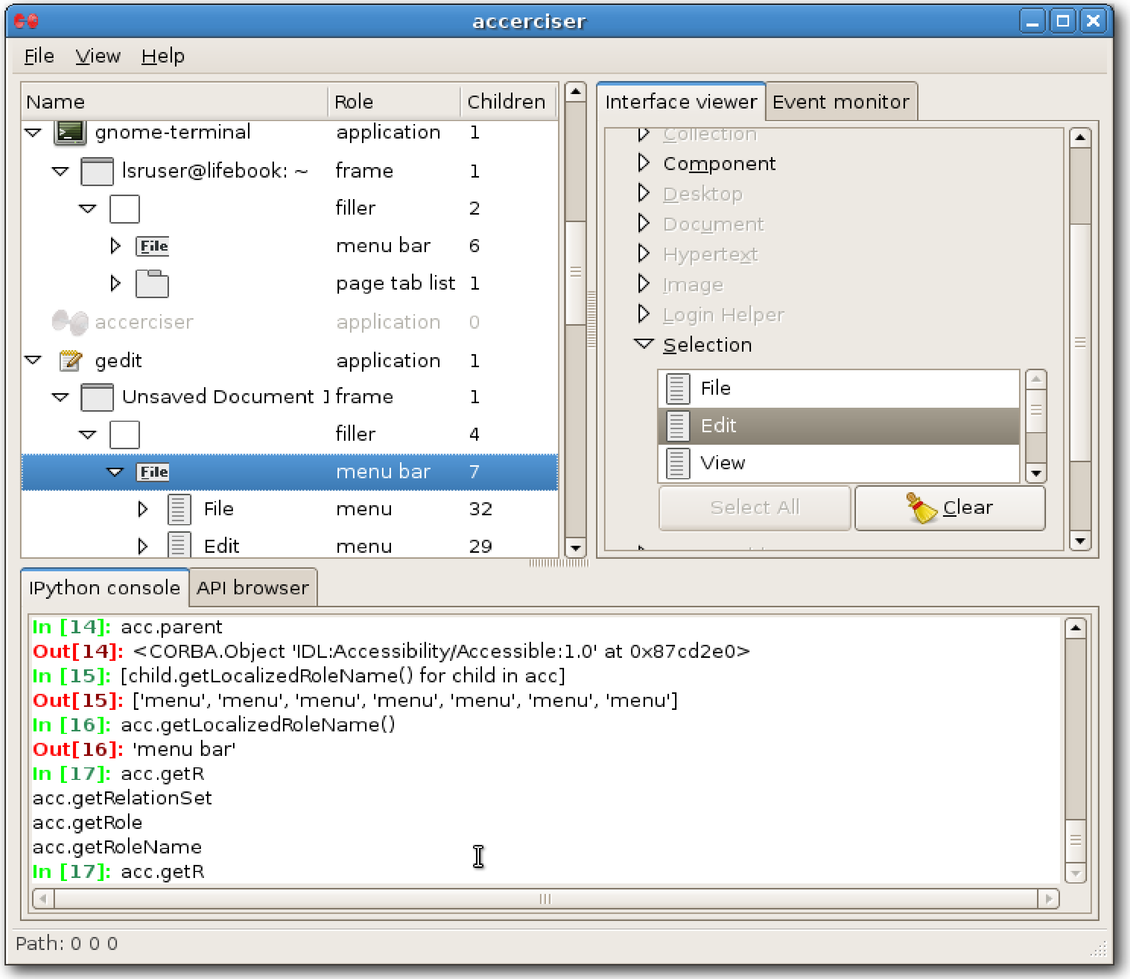
- Turn on speech, switch off the screen

https://developer.gnome.org/accessibility-devel-guide/stable/

Test it yourself! (GUIs)

Accerciser

- Sort of debugger
- Tree of widgets
- Properties



Name	Role	Children
gnome-terminal	application	1
lsruser@lifebook: ~	frame	1
	filler	2
File	menu bar	6
	page tab list	1
accerciser	application	0
gedit	application	1
Unsaved Document 1	frame	1
	filler	4
File	menu bar	7
File	menu	32
Edit	menu	29

```
In [14]: acc.parent
Out[14]: <CORBA.Object 'IDL:Accessibility/Accessible:1.0' at 0x87cd2e0>
In [15]: [child.getLocalizedRoleName() for child in acc]
Out[15]: ['menu', 'menu', 'menu', 'menu', 'menu', 'menu', 'menu']
In [16]: acc.getLocalizedRoleName()
Out[16]: 'menu bar'
In [17]: acc.getR
acc.getRelationSet
acc.getRole
acc.getRoleName
In [17]: acc.getR
```

Path: 0 0 0

https://developer.gnome.org/accessibility-devel-guide/stable/

About bugs

- Take users suggestions into consideration
  - E.g. bracketed links in text web browsers
- Be patient with disabled people
  - It's not easy for them to use your software
  - It's even more difficult for them to explain their problems in an understandable way
    - e.g. “braille doesn't follow”

➔ Discuss!

About bugs (2)

- Try to keep in mind their disability and their consequences
  - Yes, blind users don't care that the framebuffer doesn't show up properly!
- You could even contact your local institutes for disabled people, to discuss directly with users

Conclusion

- Accessibility is a concern for a lot of people
  - 10% have major concerns
  - 20% have minor concerns
- Dealing with it usually boils down to common sense
- It very often actually also helps other users
- But we need to raise awareness of this