Cally

Clutter Accessibility Implementation Library

Alejandro Piñeiro Iglesias

apinheiro@igalia.com

Introduction: Accessibility

(based on Guadec 2002 GAIL talk)

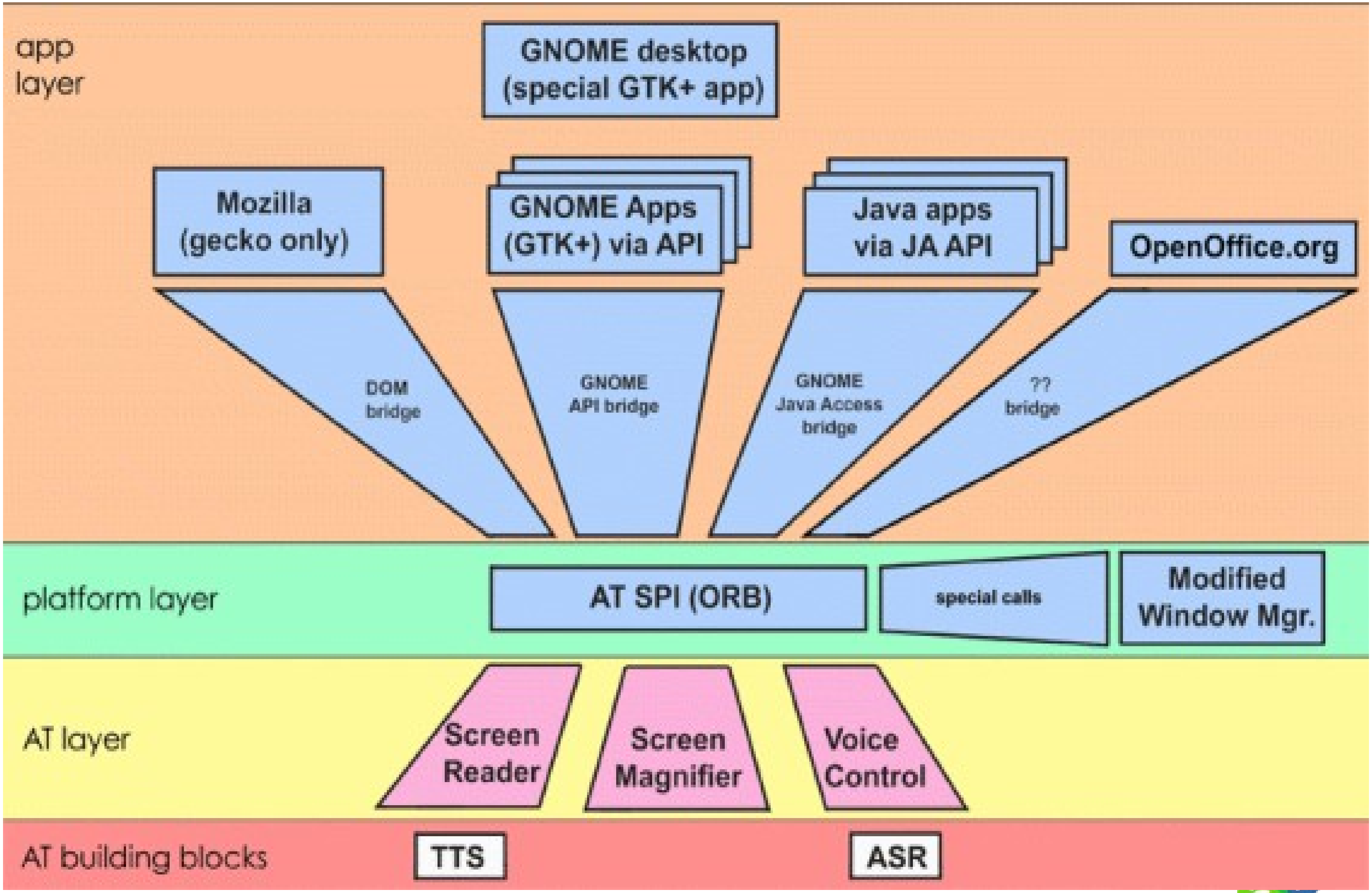
Accessibility

- Accessibility means helping people with disabilities to participate in substantial life activities
- That includes work, and the use of services, products, and information
- GNOME includes libraries and support framework that allow people with disabilities to utilize all the functionality of the GNOME user environment.

Terminology

- **AT**: Assistive Technologies
  - Hardware, software or a combination
  - Adapts UI to users with different needs
- **AT-SPI**: AT Service Provider
  - Interprocess accessibility API used by ATs to query and manipulate accessible objects
- **ATK**: Accessibility Toolkit
  - Defines an accessibility API that can be implemented in widgets or objects
  - May be implemented by non-GTK+ toolkits (abstraction on ATKUtil)
- **GAIL**: GNOME Accessibility Implementation

GNOME a11y architecture



A11y and ATK key aspects

- Provide programmatic access to the features and capabilities of application objects
- Keyboard navigation
- Theming
- ATK supports interface query
- Interfaces allow ATs to do functional heuristics.
- A11y implementation for a widget must decide which interfaces express his capabilities

Automatic testing

- An application in development state requires to test continuously if it still works after a change
- You can use a11y to test the GUI interaction
- After all, it is simply a custom AT application
- Several existing testing frameworks:
    - Dogtail
    - LDTP (used on **Mago**)
    - Strongwind
    - GATE

Clutter

Clutter: introduction

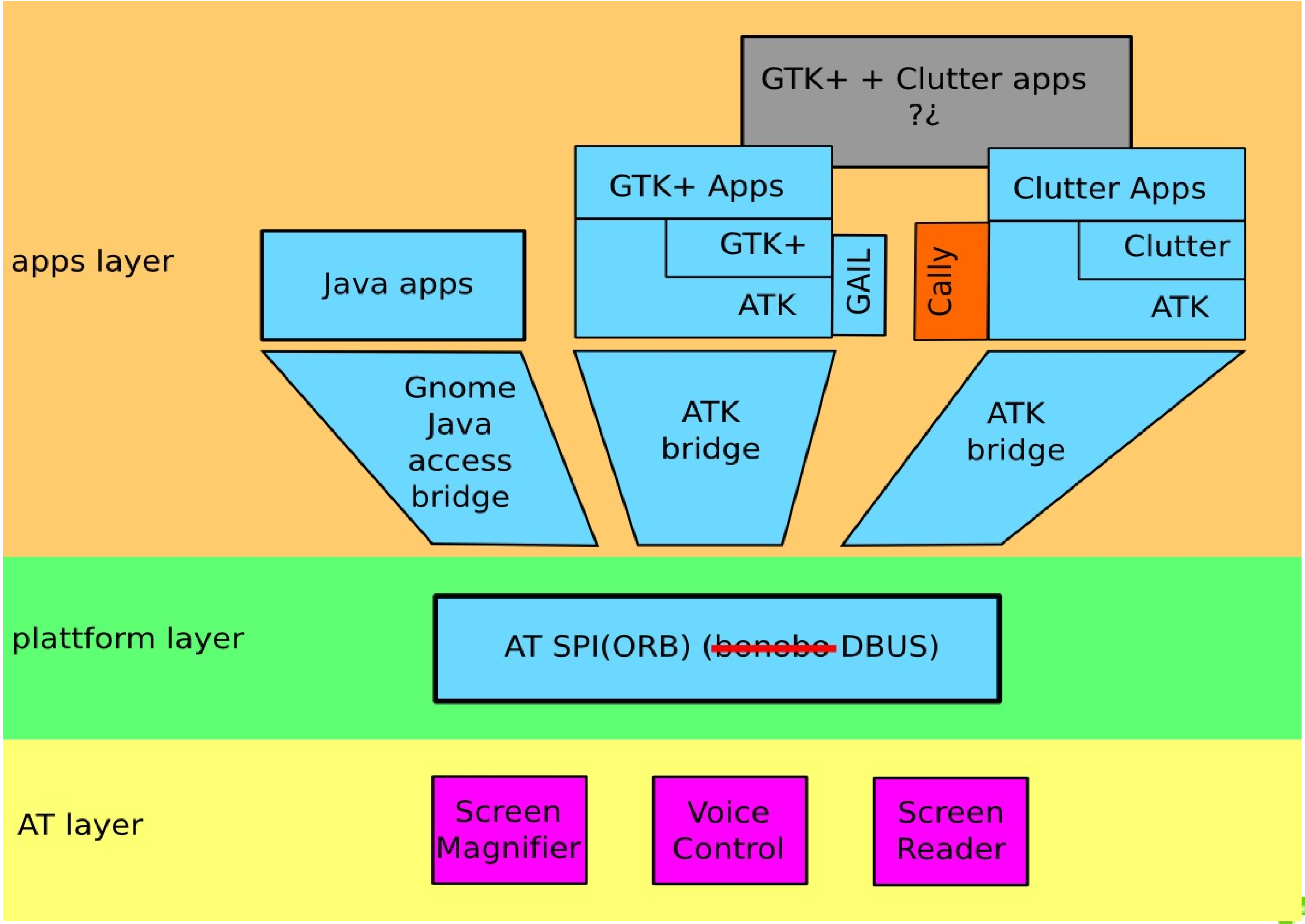
- Clutter is a library to create visually rich user interfaces
- Its purpose is being used as a drawing toolkit
- If a new toolkit uses Clutter as a drawing technology, theory says that a11y should be implemented over it rather than on top of Clutter (ie: GTK+ vs GDK)
- Tidy has appeared as a proof of concept

But ...

- Months ago we lacked information about any new complete toolkit
  - Tidy is a concept. Candies? NBTK is recent
- Some applications have started to use directly clutter for his interfaces
  - hildon-desktop
  - gnome-shell
- Basic direct a11y support on top of clutter is required.

Cally

Draft Architecture with Clutter



apps layer

GTK+ + Clutter apps ?¿

GTK+ Apps

GTK+

ATK

GAIL

Cally

Clutter Apps

Clutter

ATK

Java apps

Gnome Java access bridge

ATK bridge

ATK bridge

plattform layer

AT SPI(ORB) (~~bonobo~~ DBUS)

AT layer

Screen Magnifier

Voice Control

Screen Reader

Cally: implementation (I)

- Previous diagram is a naïve approach
- But, is there another solution?
- **CAIL** is born -> then renamed to **Cally**
  - 2009/02/16 -> 2009/05/20
- Funded by **Nokia**.
- First implementation has a specific target "to allow basic automatic testing support for hildon-desktop".
- Implementation using GAIL as mirror

Cally: implementation (II)

- Clutter 0.8 in mind
- Only main classes supported
- GtkAccessible equivalent not implemented
  - Based on AtkGObject, Clutter not modified at all
- A lot of functional information is missing
  - Clutter in general is a low-level library
  - Not a proper ATK_ROLE for most Clutter Objects
- Applications would have to add some a11y info
  - Ie: some apps are using ClutterTexture directly as buttons

Cally: implementation (III)

- An implementation over any final framework would give us full functional information
- The current implementation could be used as base
- The most abstract classes could be used:
  - CallyActor equivalent to GailWidget
  - CallyRoot equivalent to GailTopLevel
- Cally will still be useful for apps using only Clutter (hildon-desktop, gnome-shell)

CALLY: Issues

Issues: AtkComponent

- At this moment absolute window position calculation depends on the backend
- Right now only X11 is supported
- No multi-backend support on Cally
- Quick SDL API review
  - I haven't found the way to do that
  - Possible on any backend?
- Review if it this information is really dependent on the backend

Issues: AtkAction

- ClutterActor has signals for “press” and “release”
- It makes sense to implement AtkAction on CallyActor
- So we should find a way to extend AtkAction implementation on subclasses
- The solution chosen was the one used at GailCell
    - Maintain a list of action functions
    - Public methods to remove/add actions

Issues: ClutterContainer

- AtkObject uses a composite pattern
- GailContainer redefine these methods
- On Clutter, ClutterContainer is an interface
- Several AtkObject methods can be implemented using ClutterContainer
- The implementation of these methods are conditional (ATK interface heuristic)
- These methods can be refined (ie: CallyGroup)

Issues: AtkUtil

- AtkUtil reimplementation is required in a pure clutter environment
- AtkUtil::get_root is used by atk-bridge to get the root object
- CallyRoot manages the stage and display their children
  - No multi-stage supported
- The other AtkUtil features are not implemented yet

Issues: interaction with GTK+

- We want a11y support for pure-clutter apps
- But we want also a11y support for clutter + Gtk
- Two different mixed environments
  - Stage inside a GtkClutterEmbed
  - GTK widgets inside a clutter stage
- Three total different environments
- BUT, both CallyUtil and GailUtil redefines atk_get_root
- Probable future headache, not a clear solution, both tries to “get the control”

Issues: Objects exposed

- At this moment all the Clutter objects are exposed on the ATK tree representation
  - Has sense to expose Animations or Behaviours?
- A consequence of implement it at this low-level
- Not a big issue on a automatic testing environment
- Classic ATs applications should filter the relevant objects
- Would be good to investigate this on Cally itself

Issues: Public headers!

- GAIL is a isolated implementation, no development headers exported (except -util and -misc)
  - Indirect mechanism to extend GAIL objects on custom widgets (Anonymous inheritance)
  - Pro: avoid ABI dependency for the ATK interfaces implementation
  - Con: some pain if you want to extend it (ie: HAIL)
- Cally is a big candidate to be extended massively

Issues: Module loading

- We want a11y support to be optional
- Like GAIL, Cally is a module (Gmodule)
- With GAIL: `GTK_MODULES=gail:atk-bridge`
  - Modules loaded on `gtk_init`
- You shouldn't get the stage before `clutter_init` is called
- No `CLUTTER_MODULES` or equivalent
- For the moment the module should be loaded by hand by the application

More...

- Thread safe guards not managed
- Current module directory meaningless
- Missing AtkText support on ClutterText
  - Work in progress
- No gtk-doc support
- No i18 support

Finishing

Last development news

- Start to develop against trunk, waiting for Clutter 1.0
  - Cally 0.8 maintained as a branch
  - CallyCloneTexture, CallyLabel removed
- Library renamed to Cally on May
- Started to work on ClutterText a11y support
  - Based on GailEntry and similar
  - At first, GailUtil will be used
  - This will add a GAIL dependency
- A examples directory added

Future

- In general, complete the Cally implementation
- Check gnome-shell
- Check the new clutter based toolkits
  - nbtk
  - Glitter
  - etc.
- No plans for a full a11y support on Tidy
  - Anyway, some Tidy objects has a11y support implemented. Should be published (ToDo)

DEMO

Conclusions

- Is the a11y support for Clutter interesting?
  - Yes
- Is it working?
  - Yes
- Is it complete?
  - No
- Is it directly usable?
  - No, applications need to load the module by hand

We are open to patches!

- A lot of functionality missing
- How to get it:
  - git clone http://git.igalia.com/cally.git
- Someday it will be moved to the clutter repository (we hope)
- Someday we will use clutter bugzilla to track bugs down (we hope)
- Any comments: apinheiro@igalia.com

QUESTIONS?