

Introducción a la accesibilidad en GNOME

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

```
attribute */  
uint64  
CODE,  
code.",  
code",  
0,  
64,  
/*  
/  
E
```



The logo consists of four vertical lines of different colors: red, yellow, green, and blue, positioned in the top-left corner of the slide.

GNOME

GNOME?

- Entorno de escritorio
- Compuesto por Free Software y otros tipos de software de fuentes abiertas
- Gran énfasis en hacer el entorno lo más amigable (user-friendly) posible
- Abrir ficheros, editar documentos, reproducir música

GNOME es ...

- Ampliamente usado: la mayoría de las distribuciones Linux lo usan
- Free Software: parte del proyecto GNU
- Usable: uno de los primeros que definieron HIG
- **Accessible**: la accesibilidad es un elemento central en GNOME

Accesibilidad

Accesibilidad

- Accesibilidad significa ayudar a la gente con discapacidades a participar en las tareas significativas de la vida diaria
- Esto incluye el trabajo, y el uso de servicios, productos y el acceso a la información
- GNOME incluye librerías y mantiene un framework que permite a la gente con discapacidades a utilizar toda la funcionalidad del entorno de usuario de GNOME.

Aspectos fundamentales de la accesibilidad

- Proporcionar acceso programático a las distintas aplicaciones del entorno
- Navegación por teclado
- Temas visuales

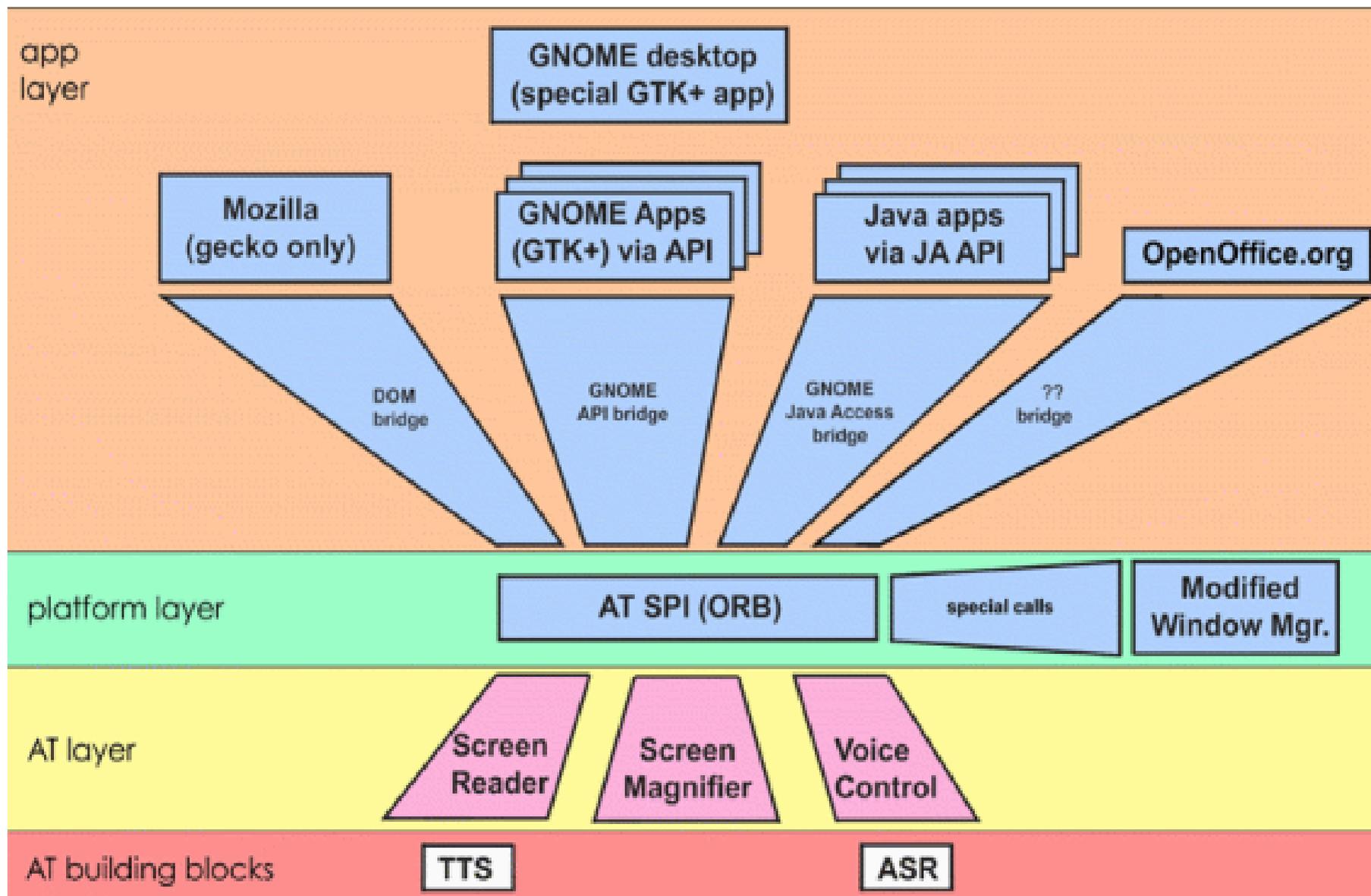


Accesibilidad: detalles técnicos

Terminología

- **AT: Tecnologías Asistivas** (Assistive Technologies)
 - Hardware, software o una combinación
 - Adapta las interfaces visuales a usuarios con distintas necesidades
- **AT-SPI: AT Service Provider**
 - Interfaz de accesibilidad inter-proceso utilizada por ATs para pedir y manejar información accesible a las aplicaciones
- **ATK: Toolkit de Accesibilidad**
 - Define una API de accesibilidad que puede ser implementada por widgets o objetos
- **GAIL:** Implementación de ATK para GTK+ y otros componentes de GNOME

Arquitectura en GNOME 2.0



ATK

- ATK es la API abstracta básica de GNOME
 - GtkWidget hereda de GtkAccessible
- Una representación de los widgets en árbol
- Puede ser útil que las aplicaciones establezcan valores de las propiedades de ATK (ej: glade)
- Dos aproximaciones para su implementación:
 - Directamente en la librería (ej: GtkIconView)
 - Usar una librería de implementación externa (ej: GAIL)

Otros usos: testing automático

- Una aplicación en desarrollo necesita ser testada continuamente para comprobar si sigue funcionando después de un cambio
- Se puede usar el soporte de a11y para testar la interacción visual
- Hay varios frameworks de testeo:
 - Dogtail
 - LDTP (used on **Mago**)
 - Strongwind

Como funciona (GNOME 2)?

- Estableciendo a “true” la variable de gconf: `"/desktop/gnome/interface/accessibility"`
- Gnome-settings-daemon establece `"GTK_MODULES = gail:atk-bridge"`
- Cuando arranca una aplicacion GTK se cargan esos módulos
- Atk-bridge registra la aplicación y llama a la implementación de AtkUtil para pedir el objeto raíz

Funciona perfectamente?

- Muy pensado desde el POV de un solo toolkit en cada momento
- Firefox utiliza Gtk para los menus, pero Gecko para el renderizado html
- Hace un workaround, al que se le debe dar soporte (basicámente un hack):
 - NO_AT_BRIDGE
 - NO_GAIL

Funciona perfectamente (2)?

- En palabras de Mark Doffman:
 - “I personally think that the ATK API is broken. (The AtkUtil API in particular)”
- Propone cambios en el registro del objeto raíz y la gestión de eventos.
- ATK tiene más de 10 años !!
 - Las necesidades de entonces y ahora no son las mismas
- Distintos bindings de python para at-spi
 - Dogtail pyspi
 - Pyatspi, etc => intenta ser el final

Por ahí viene
GNOME 3.0!!

ORBit/bonobo

- ORBit/bonobo llevan bastante tiempo marcados como deprecados
- Pero GNOME 3.0 tiene entre otros objetivos eliminarlos completamente
- At-spi y otros componentes están basado en CORBA
- En los últimos tiempos el soporte de accesibilidad se ha convertido en el último mayor usuario de ORBit

Clutter

- Clutter es una librería para crear interfaces de usuario
- Algunas aplicaciones han empezado a usarlo directamente
 - Hildon-desktop
 - gnome-shell
- Distintos toolkits han empezado a aparecer:
 - NBTK->MX
 - Candies, Glitter
- GAIL no les proporciona el soporte a11y

Gnome-shell

- Gnome-shell pretende redefinir la interfaz de usuario GNOME
- Toolkit visual ST, fork de MX (basado en clutter)
- Único tema visual
- gnome-mag deja de funcionar (también por bonobo)

GSettings

- Reemplazo de Gconf
- Es necesario migrar los esquemas de configuración
- Es necesario migrar también las aplicaciones
 - Algunas además quieren seguir siendo GNOME 2 compatibles

Buscando soluciones

at-spi

- Mark Doffman empezó la reescritura de at-spi utilizando DBUS
- Proyecto soportado inicialmente por **Nokia**
- Proporciona bindings de python, ahora sólo hay unos bindings
- En vez de reimplementar el puente Java, se está implementando un wrapper.

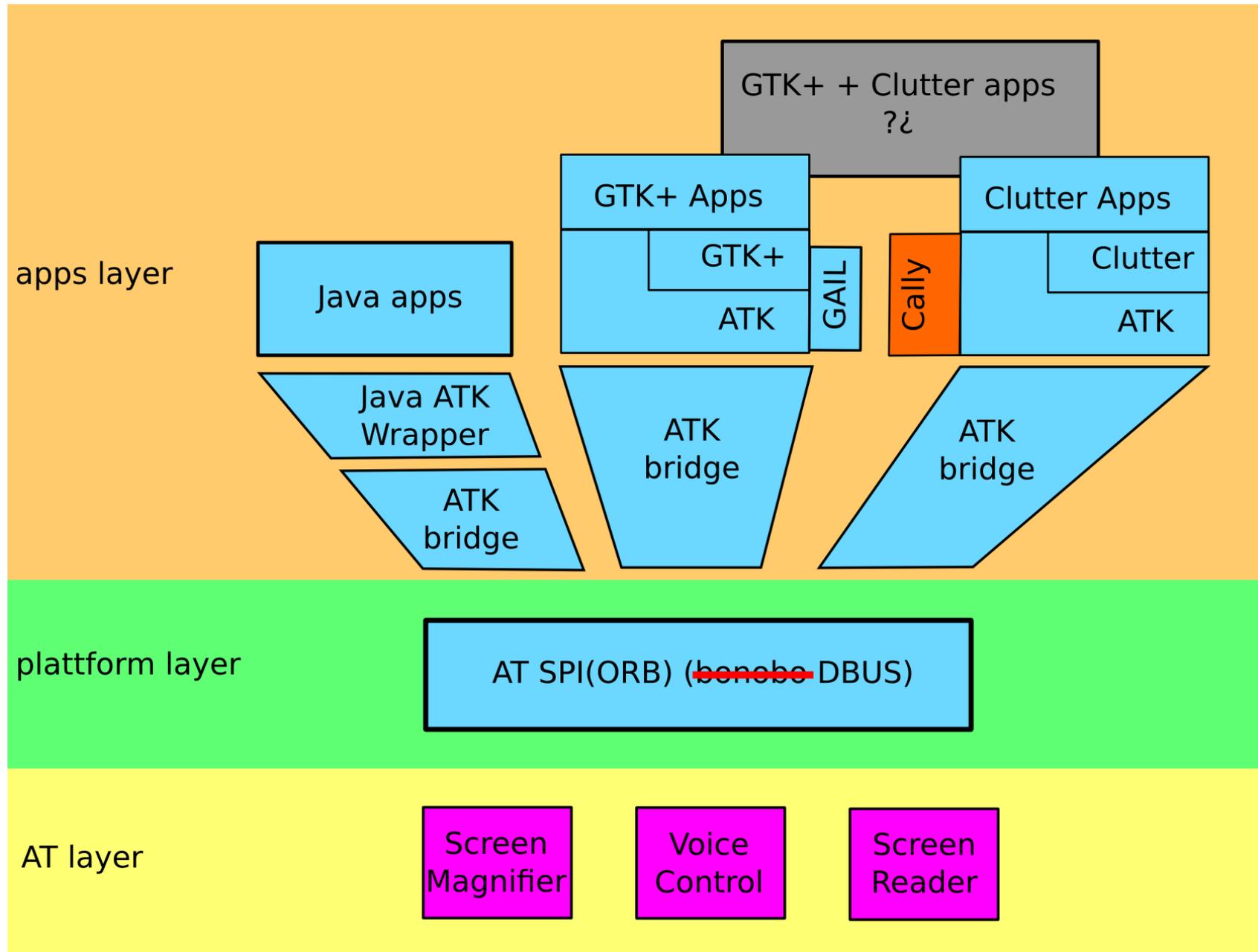
Cally

- Implementación de ATK para Clutter
- Primera implementación soportada por **Nokia** para proporcionar un soporte básico de testing para hildon-desktop.
- Desde entonces ha continuado evolucionando.
- En Julio del 2010 se ha integrado en Clutter, dejando de ser una librería independiente

gnome-shell

- Orca ha empezado a funcionar con gnome-shell
- Joseph Scheuhammer ha implementado un magnificador integrado directamente en gs
- Los desarrolladores de gnome-shell han incluido accesibilidad en su roadmap

Arquitectura en GNOME 3.0



GNOME 3 ya ha sido
publicado!!



GNOME 3.0 status

- GNOME Shell no es suficientemente accesible
- Fallback-mode no ha sido suficientemente testado
- La release 3.0 fue un desastre desde el POV de la accesibilidad
 - 3.0.2 era sustancialmente mejor (TestingDistro)

at-spi2

- Rendimiento: CORBA vs DBUS
- Compatibilidad con C-SPI descartada
 - Usa libatspi
- Tests de regresión de Orca
- Testeo por parte de los usuario
 - Aún no ha sido integrado por ninguna distribución

bonobo

- Aún queda pendiente la migración a DBUS de varias aplicaciones (gnome-mag, Dasher, GOK):
- Pero las distribuciones aún son reticentes al cambio a at-spi2.

Fue esto una sorpresa?

- No realmente, el objetivo principal para GNOME 3.0 era tener listas las funcionalidades principales
- El objetivo de la accesibilidad es GNOME 3.2
- Ahora ya tenemos un verdadero entorno de prueba

Conclusiones

Conclusiones

- GNOME se ha preocupado desde sus inicios de mantener un soporte de accesibilidad
- GNOME 3.2 representa un gran desafío
- Los mayores problemas de la accesibilidad son:
 - Ser considerada una tarea secundaria
 - Falta de recursos
- Pero a pesar de todos los problemas, es usable, y al nivel de otras soluciones comerciales

Referencias

- <http://live.gnome.org/Accessibility/>
- <http://live.gnome.org/Accessibility/GNOME3>
- <http://projects.gnome.org/accessibility/>



DEMO



Preguntas?