Introduction of Accessibility Infrastructure on GNOME

Li Yuan *Li.Yuan@Sun.Com*

Sun China Engineering and Research Institute

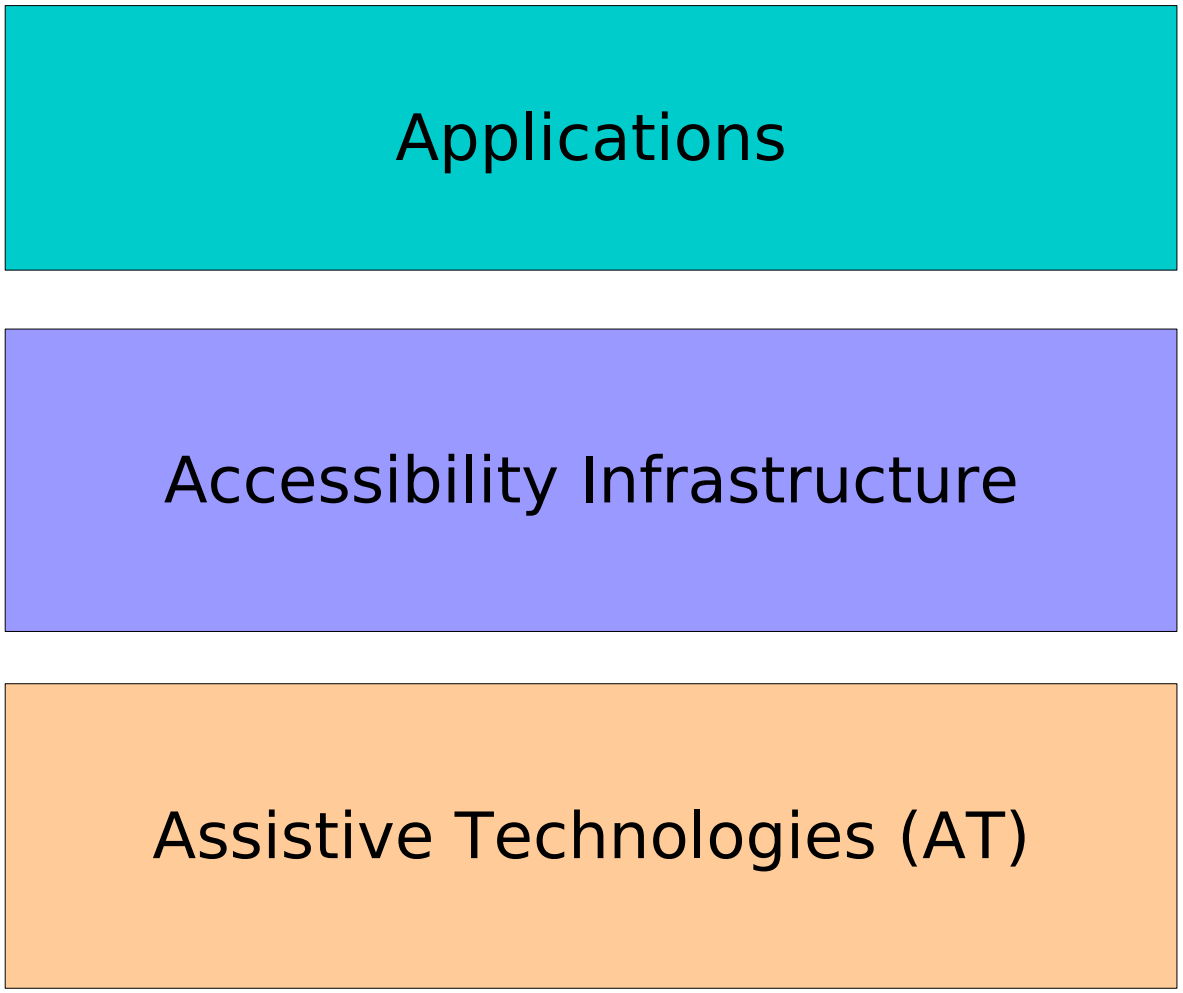
Agenda

- What is Assistive Technologies
- Architecture of Assistive Technologies
- Other Usage of Assistive Technologies
- The Future

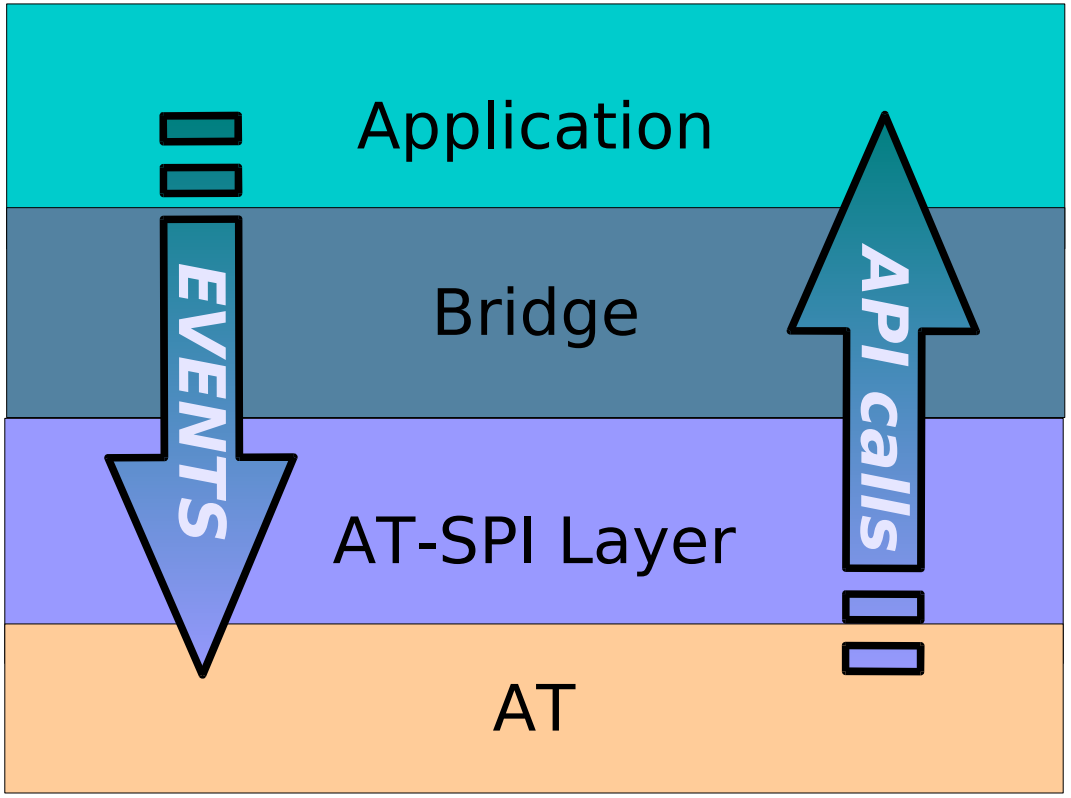
What is Assistive Technologies

- Assistive technologies (AT) enable individuals to make full use of computer-based technology despite variability in physical or sensory abilities due to illness, aging or disability.
- There are some kinds of ATs: Screen Reader, Magnifier, On-Screen Keyboard.

3 Large Components in AT World

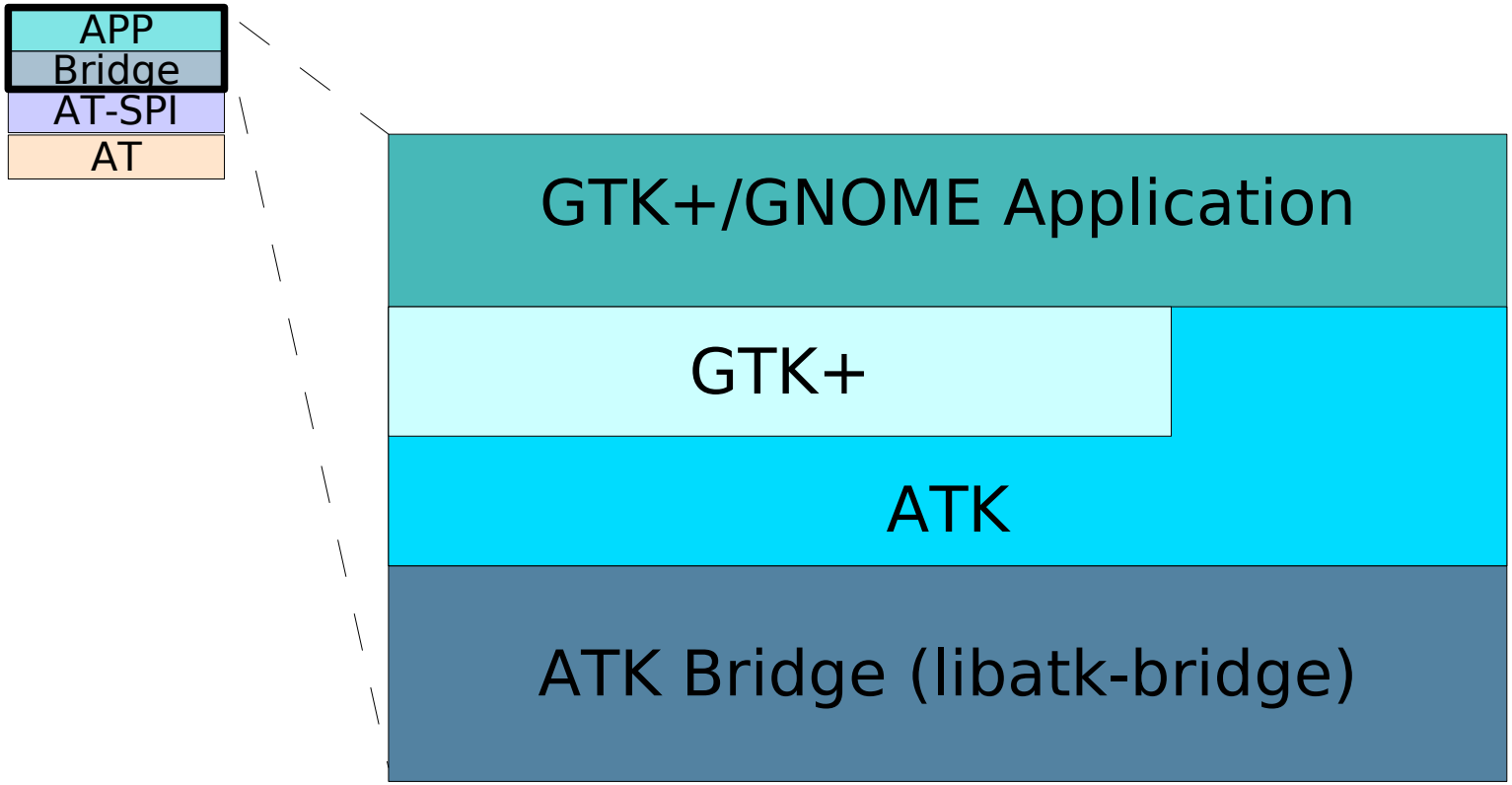


Applications and ATs communicate via a layered architecture

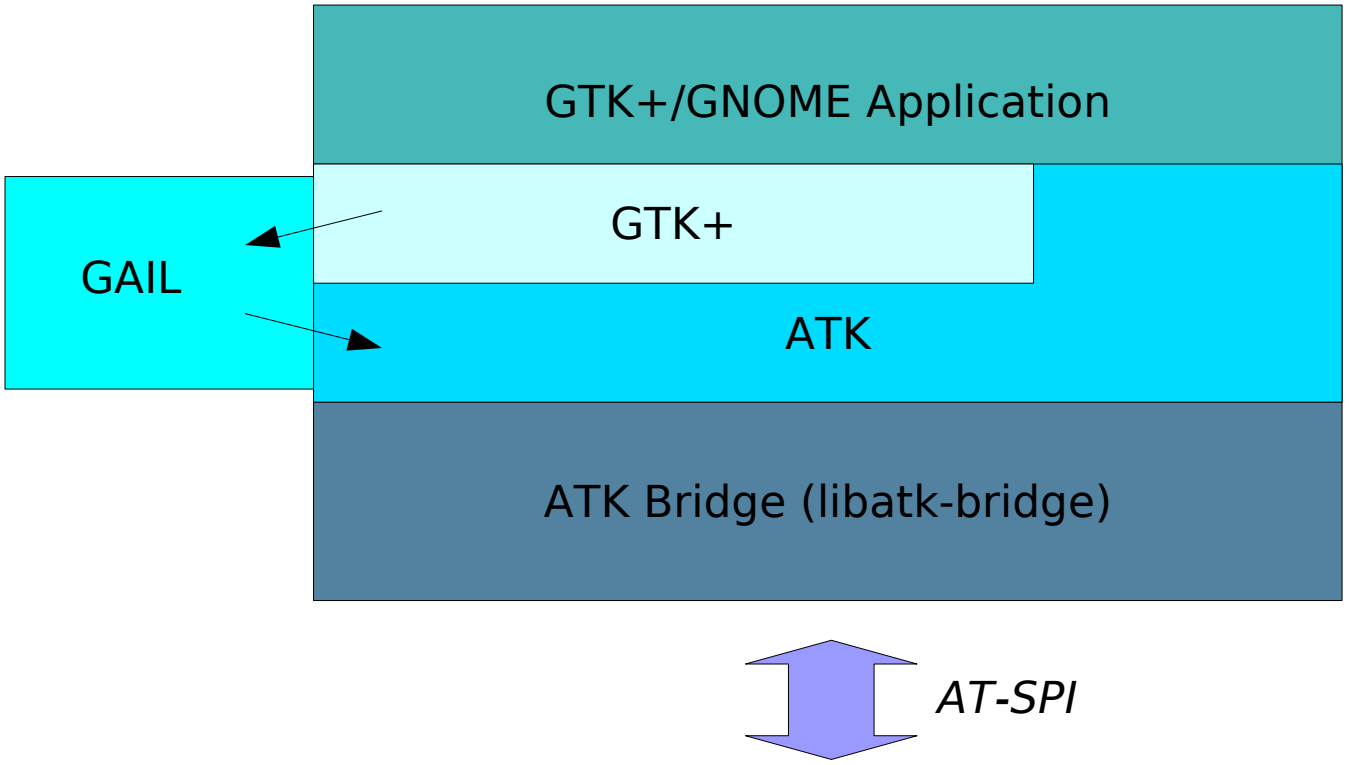


- SPI == “Service Provider Interface”
- Applications communicate with the AT-SPI via a “bridge” layer
- The bridge layer may be different for different applications.

How GTK+/GNOME Apps connect to AT-SPI



How ATK is implemented for GTK+ : the *GAIL* module



- *GAIL hooks into ATK and implements ATK interfaces on behalf of GTK+.*

(This relies on AtkFactory and AtkUtil classes which we will examine later.)

- *GAIL knows about ATK and GTK+, but neither ATK nor GTK+ know about GAIL.*
- *GAIL is dynamically loaded via glib's Gmodule API*

ATK and AT-SPI have a 1:1 mapping (almost)

AtkObject:
atk_object_get_name
atk_object_get_description
atk_object_get_role
atk_object_get_child_count
atk_object_ref_child
etc.

ATK

AtkValue:
atk_value_get_min_value
atk_value_get_max_value
atk_value_get_current_value
atk_value_set_current_value
atk_value_get_min_increment

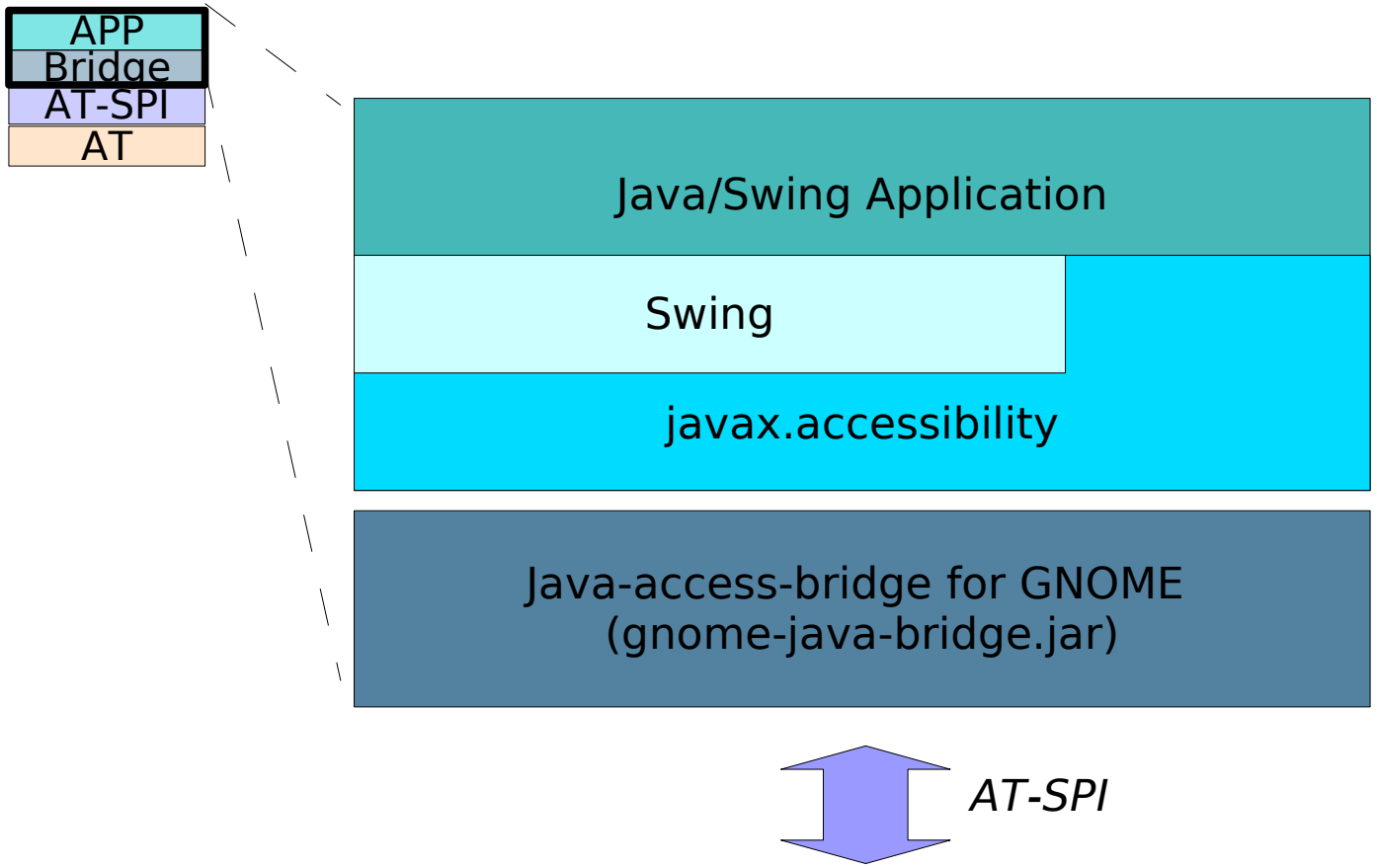
Accessibility::Accessible
Accessible:name
Accessible:description
Accessible:role
Accessible:childCount
Accessible:getChildAtIndex
etc.

AT-SPI

Accessibility::Value
Value:minValue
Value:maxValue
Value:minIncrement
Value:currentValue *(Read-write*

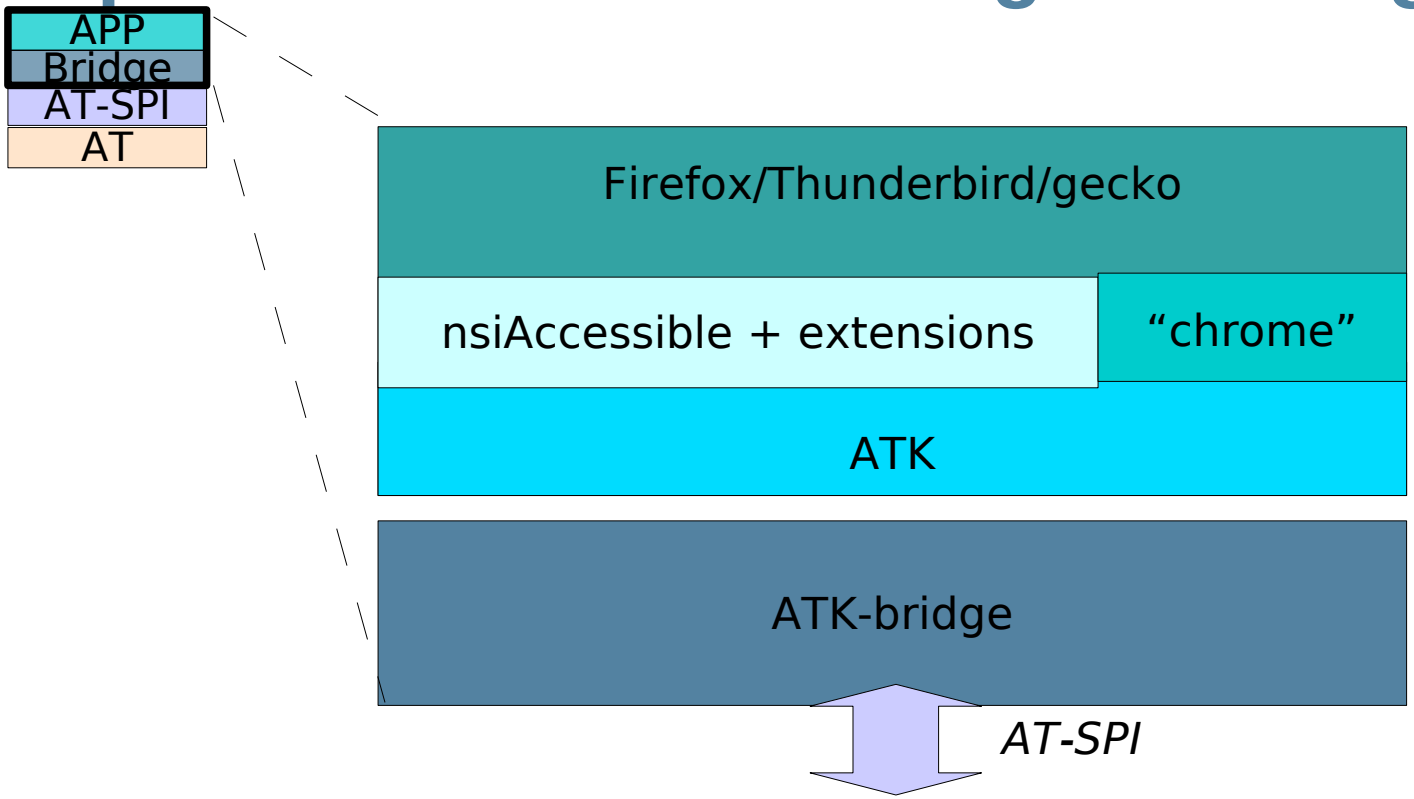
- *There are a few convenience/efficiency methods in AT-SPI that ATK doesn't have yet*
- *AT-SPI includes an object called the 'Registry' which has no ATK counterpart.*
- *ATK includes a 'Factory Model' which is useful to implementors but not in AT-SPI.*

Java applications use a different bridge



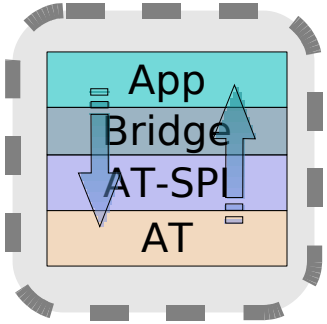
- *Swing defines a java-specific accessibility API in javax.accessibility*
- *The end result is AT-SPI, just as for GNOME apps (interoperability)..*

Mozilla uses ATK, but must provide its own implementations for XUL/gecko widgets

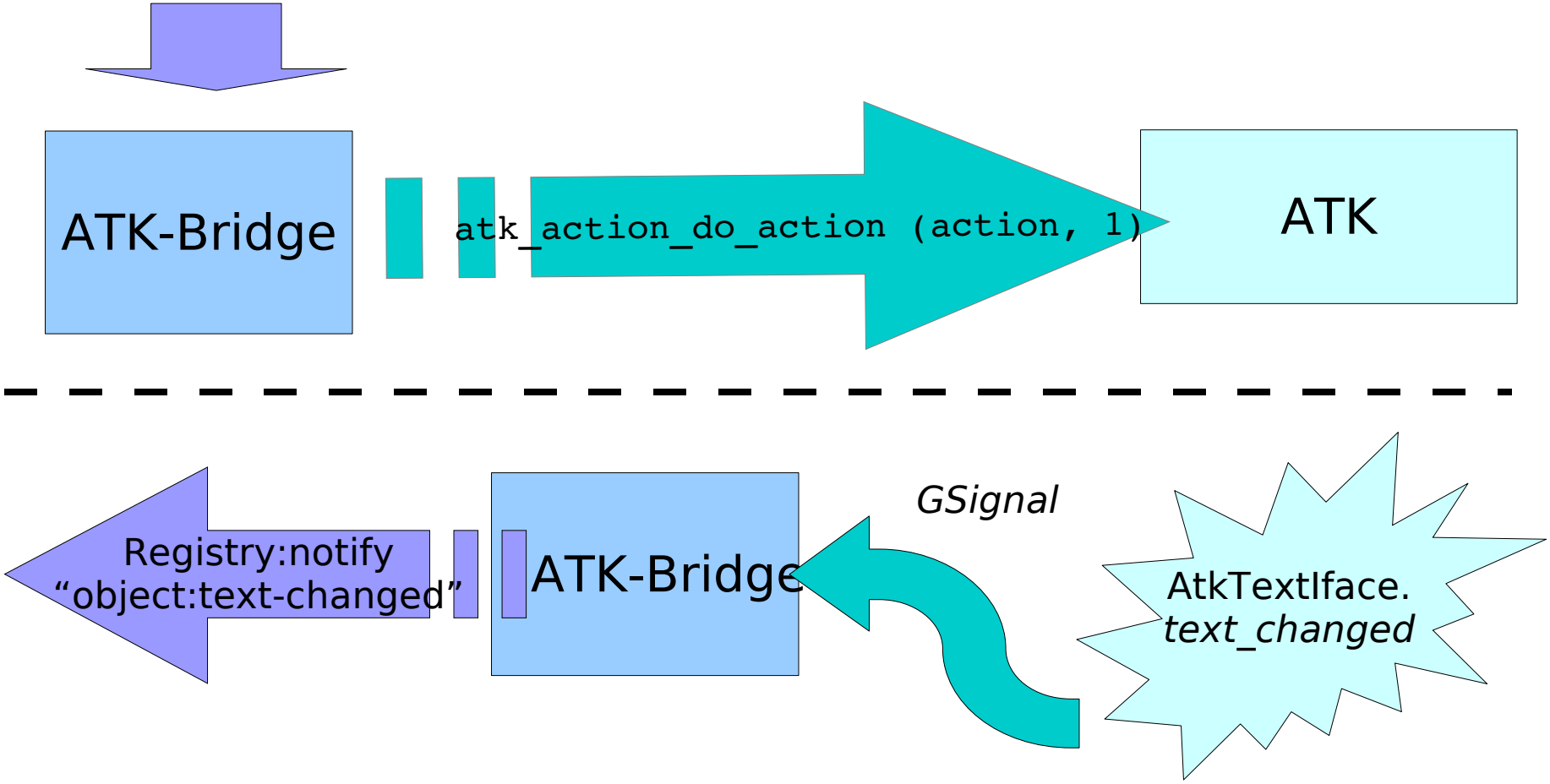


- *Both gecko (HTML content) and ‘chrome’ (GUI widgets) must provide ATK implementations*
- *nsiAccessible is used to implement MSAA too, but must be extended.*

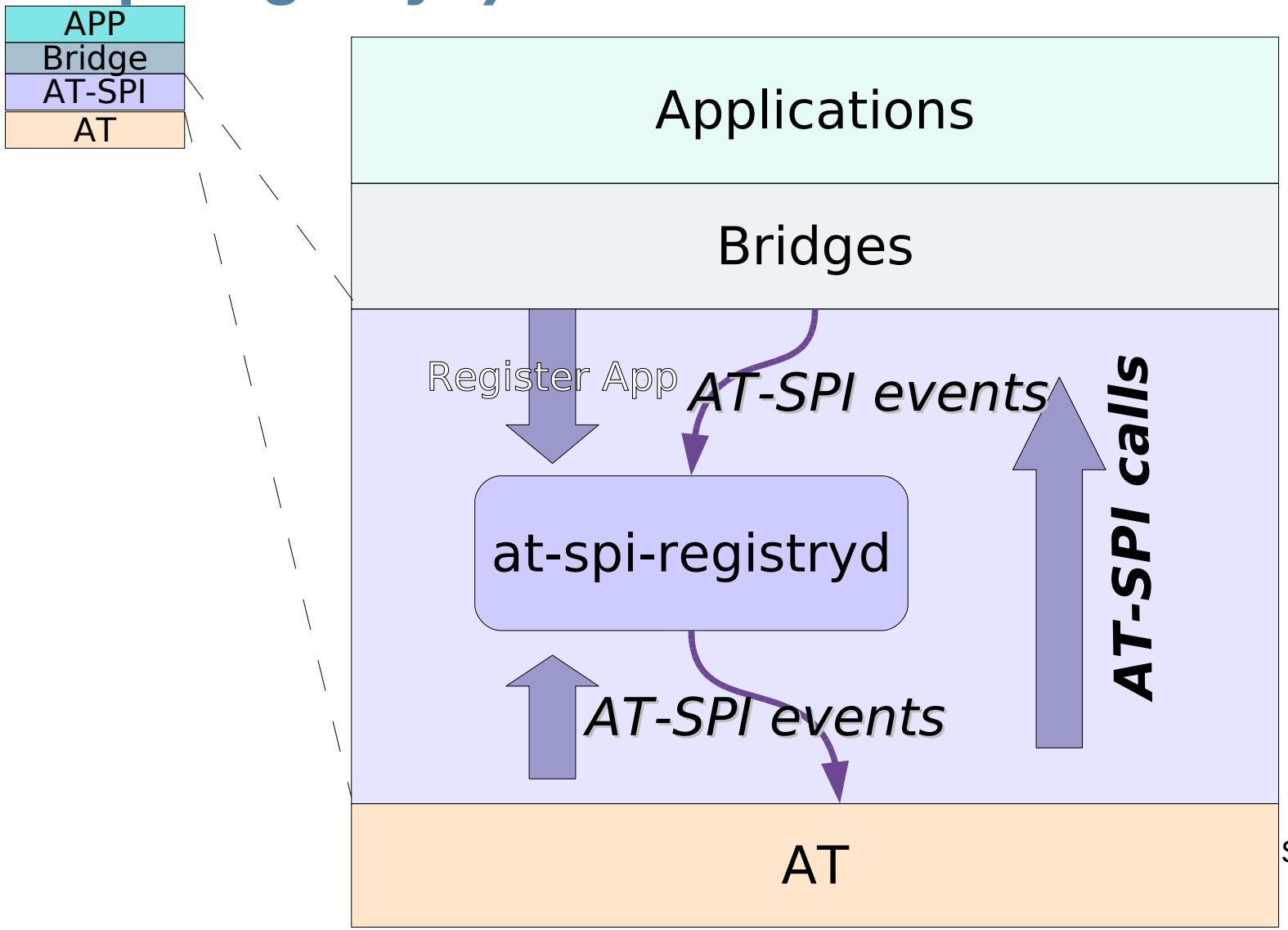
AT-SPI requests are relayed to ATK via atkbridge; ATK events are relayed to AT-SPI for dispatch



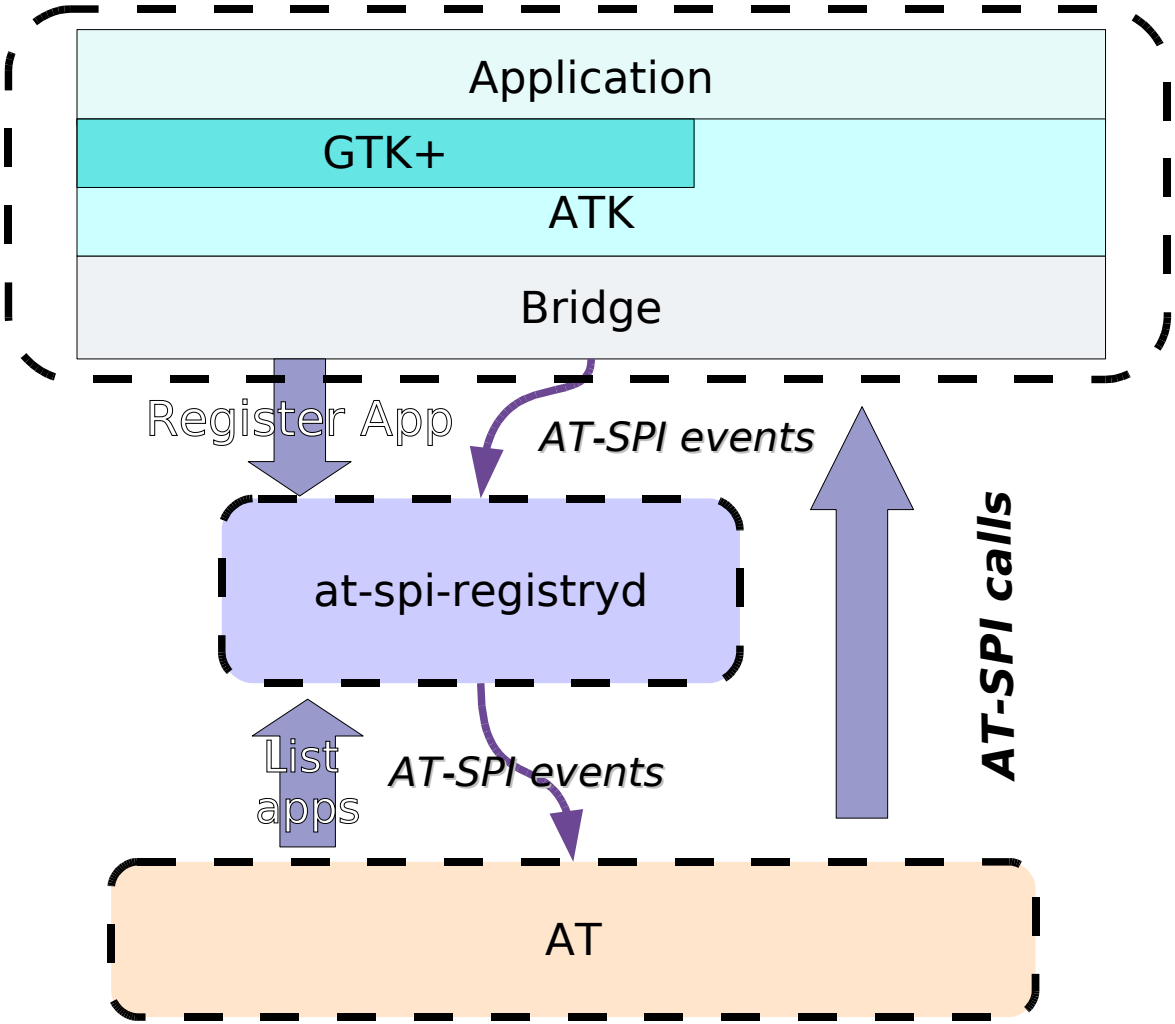

```
Accessibility::Action:doAction(1);
```



AT-SPI layer: Registry service (at-spi-registryd) and IPC

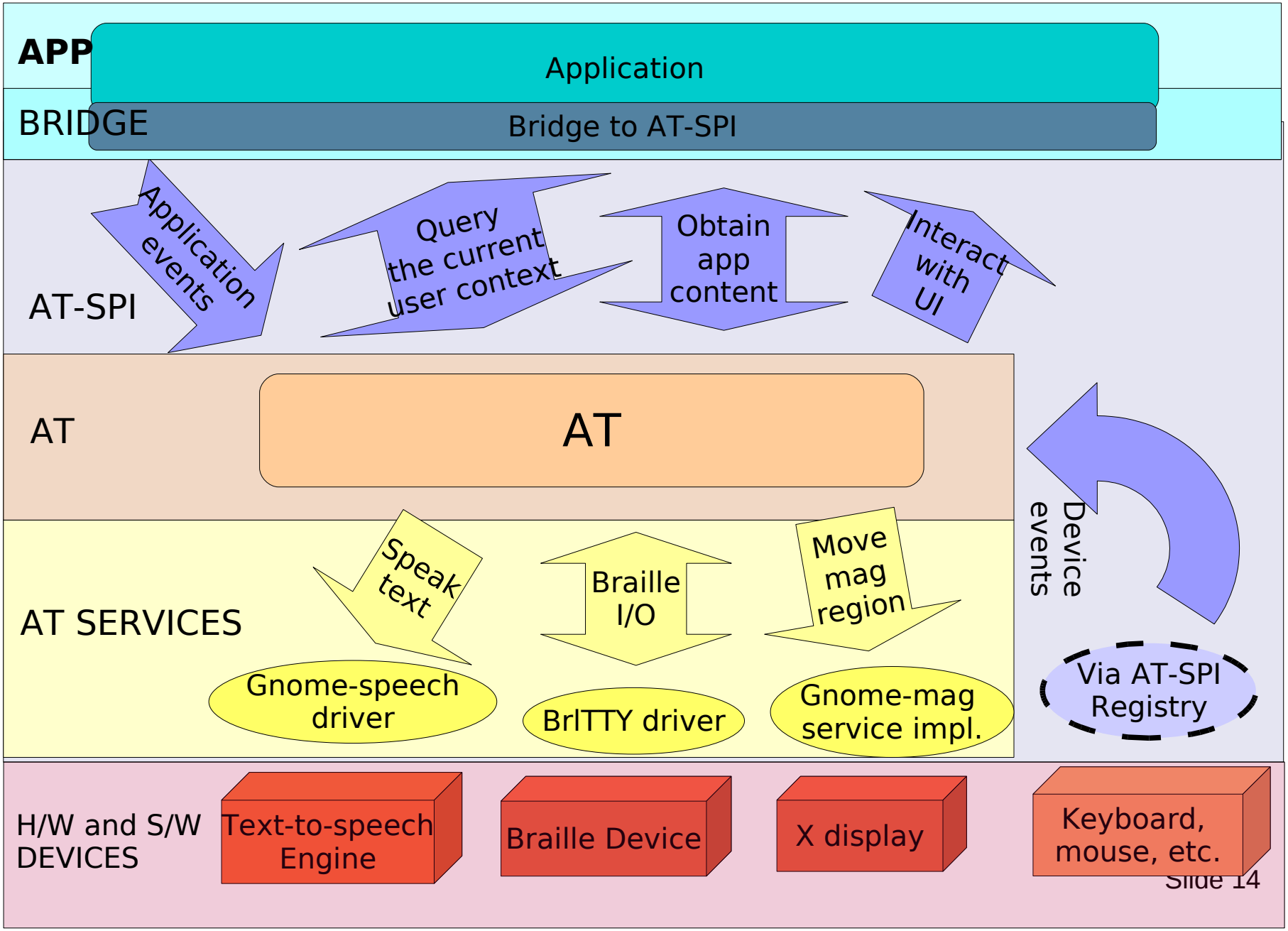


Process space diagram



- AT gets app handles from registry, and registers for event notifications
- Other AT communications go direct to the application's bridge code

The Big Picture



Other Usage of Assistive Technologies

- Assistive Technologies provides a way from which a program can query and control other application
  - Automation tests: LDTP, Dogtail
  - Applications like desktop dictionary

The Future of Assistive Technologies

- Current Assistive Technologies on GNOME uses CORBA to do the communication work
  - > will deprecated
  - > a little heavy
- Will move to D-Bus
  - > can work on both GNOME and KDE
  - > light-weight
    - > possibility to work on mobile device

Thanks!

Li Yuan Li.Yuan@Sun.Com

Sun China Engineering and Research Institute