The GNOME™ Conference

GUADEC

Slow progress on app save/restore support

Philip Withnall
Endless
philip@tecnocode.co.uk

July 28, 2023

Rīga

What is app save/restore support?

- Reopening an app in the same state as it was when it was closed
- Closing by user action or by system policy
- gnome-session used to support a limited version of this in the olden days

Use cases

- Restore apps to largely the same state when logging out and in again
- Restore apps to largely the same state if they are killed and later restarted by the session manager due to resource constaints
- Restore apps to largely the same state if they are updated and restarted due to an update (e.g. flatpaks)
- Allowing apps to choose *not* to be restored (even if the data is available) if it would not be appropriate

Use cases

- **Not**: Allowing apps to checkpoint their state or provide an undo stack
- **Not**: Allowing apps to be restored if they are killed unexpectedly, crash, or power is lost

Main components

App Decide what to save and restore

Session manager Store the list of apps to restart, manage saving and restoring them

- (While we're there: might as well store the apps' data too)

UI toolkit Serialise/Deserialise the app's UI

Shell Decide when to kill/restart an app due to lack of resources

Proposed architecture

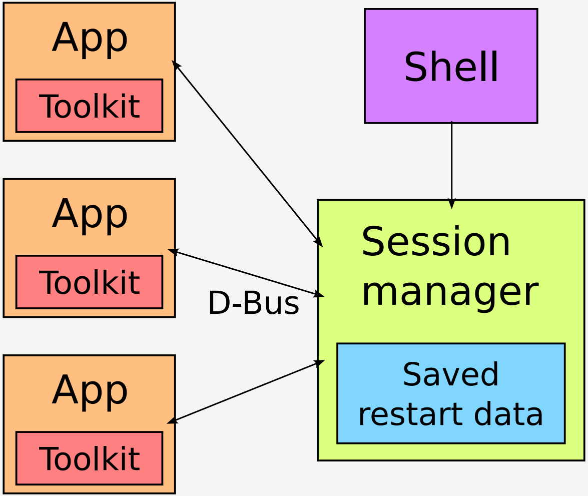
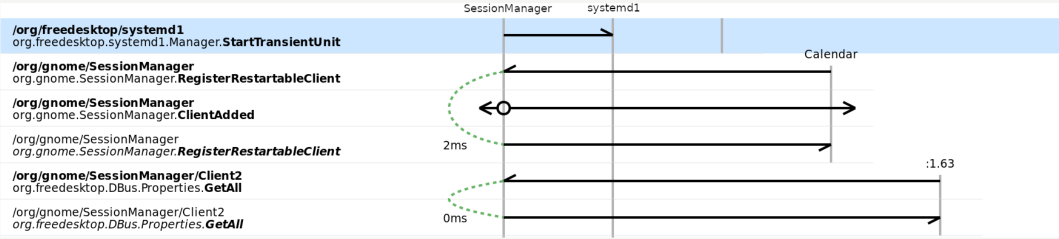


Figure: Proposed system architecture

Proposed architecture

- Extend the session manager protocol
- Apps get their restore data when they register with the session manager
- Save their data when the session ends (if they're still running)
- Expose their data on `org.gtk.Application` as well, so it can be queried any time
- Data is an arbitrary `GVariant`

Proposed architecture: restoring



/org/freedesktop/systemd1
org.freedesktop.systemd1.Manager.**StartTransientUnit**

/org/gnome/SessionManager
org.gnome.SessionManager.**RegisterRestartableClient**

/org/gnome/SessionManager
org.gnome.SessionManager.**ClientAdded**

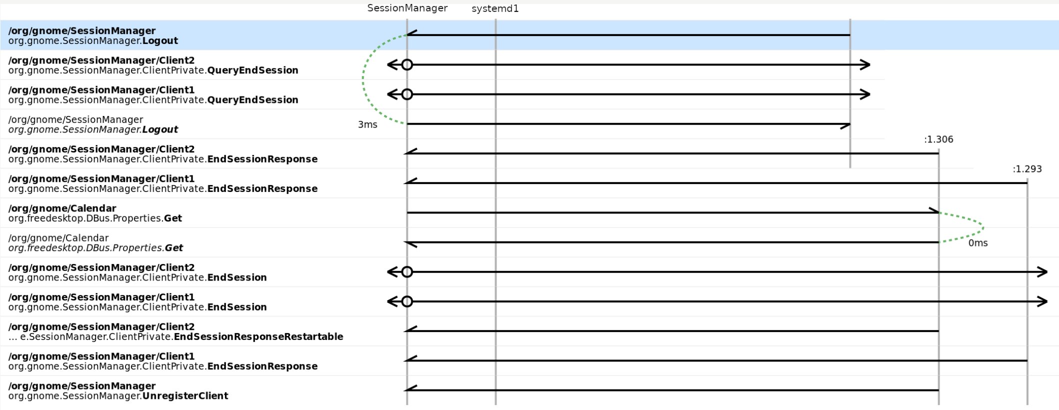
/org/gnome/SessionManager
*org.gnome.SessionManager.**RegisterRestartableClient***

/org/gnome/SessionManager/Client2
org.freedesktop.DBus.Properties.**GetAll**

/org/gnome/SessionManager/Client2
*org.freedesktop.DBus.Properties.**GetAll***

Figure: Registering a restartable client on session start

Proposed architecture: saving



/org/gnome/SessionManager
org.gnome.SessionManager.**Logout**

/org/gnome/SessionManager/Client2
org.gnome.SessionManager.ClientPrivate.**QueryEndSession**

/org/gnome/SessionManager/Client1
org.gnome.SessionManager.ClientPrivate.**QueryEndSession**

/org/gnome/SessionManager
*org.gnome.SessionManager.**Logout***

/org/gnome/SessionManager/Client2
org.gnome.SessionManager.ClientPrivate.**EndSessionResponse**

/org/gnome/SessionManager/Client1
org.gnome.SessionManager.ClientPrivate.**EndSessionResponse**

/org/gnome/Calendar
org.freedesktop.DBus.Properties.**Get**

/org/gnome/Calendar
*org.freedesktop.DBus.Properties.**Get***

/org/gnome/SessionManager/Client2
org.gnome.SessionManager.ClientPrivate.**EndSession**

/org/gnome/SessionManager/Client1
org.gnome.SessionManager.ClientPrivate.**EndSession**

/org/gnome/SessionManager/Client2
… e.SessionManager.ClientPrivate.**EndSessionResponseRestartable**

/org/gnome/SessionManager/Client1
org.gnome.SessionManager.ClientPrivate.**EndSessionResponse**

/org/gnome/SessionManager
org.gnome.SessionManager.**UnregisterClient**

Figure: Saving app state when ending the session

Application interface

```
struct _GApplicationClass
{
  GVariant *(* build_restart_data)  (GApplication  *application,
                                     char         **out_tag);

  void      (* consume_restart_data)(GApplication  *application,
                                     const char    *tag,
                                     GVariant      *data);
};
```

What's implemented?

- Application interface is ready for review in GLib
- Session manager API changes ready for initial review
- GTK session management changes are the right shape
- GTK widget tree serialisation/deserialisation not done
- Shell integration not thought about

Current problem points

- Little thought given to non-GNOME desktop, non-GTK apps
- Little thought given to XSMP apps
- Prototype for GTK serialisation is a few years old and I haven't incorporated it
- API has to be stable, so it has to be right

What's next?

- Help to validate the architecture
- Come to the GTK BoF session (Saturday, 10:00)!
- Landing some initial bits
- gnome-session changes
- Future work on GTK and Shell

Miscellany

Slide source `https://gitlab.gnome.org/pwithnall/guadec-app-save-restore-presentation-2023`

GLib merge request `https://gitlab.gnome.org/GNOME/glib/-/merge_requests/683`

gnome-session branch `https://gitlab.gnome.org/pwithnall/gnome-session/-/tree/session-restore`

GTK branch `https://gitlab.gnome.org/pwithnall/gtk/-/tree/save-state`

Beamer theme: `https://gitlab.gnome.org/GNOME/presentation-templates/tree/master/GUADEC/2023`