



# kdbus in Tizen 3.0

Hyungjun Choi  
Karol Lewandowski

Samsung Electronics

**TIZEN™**  
**DEVELOPER**  
**CONFERENCE**  
2014  
**SAN FRANCISCO**

# Agenda



# Agenda

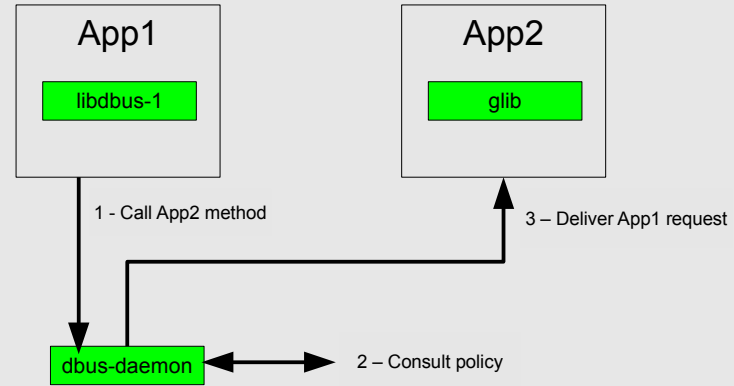
- **D-Bus vs kdbus**
- **Motivation and project goals**
- **First attempts**
- **kdbus in Tizen 3.0**
- **Challenges**

# D-Bus vs kdbus



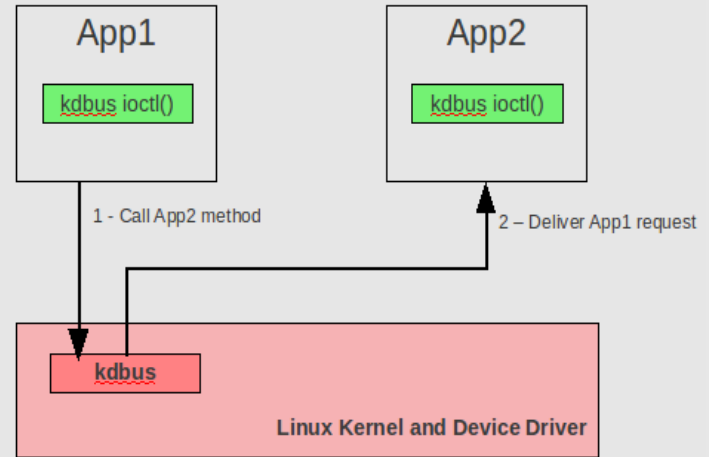
# D-Bus

- **Message bus system**
- **Method Call Transaction**
- **Signals**
- **Broadcasting**
- **Policy**
- **Activation**
- ...



# kdbus

- **Low-level, native kernel D-Bus transport**
  - All communication between processes take place over special character device nodes in /dev/kdbus.
- **Receiver buffers**
  - Single copy to destination(s)
- **memfds**
  - File descriptor for memory regions
  - Zero Copy!
  - At 512K zero copy is faster than single copy



# D-Bus vs kdbus

D-Bus (It's inefficient)	kdbus (It's efficient)
10 copies	2 of fewer copies
4 complete validations	2 validations
4 context switches	2 context switches
Suitable only for control, not payload	Suitable for large data (GiB), zero-copy, optionally reusable
...	...

Method Call Transaction (Remote procedure call and reply)



# Motivation and project goals





# Motivation and project goals



- **Motivation**

- More efficient, always available IPC mechanism
- kdbus may solve our issues with sharing large amounts of data (tested w/ prototypes)

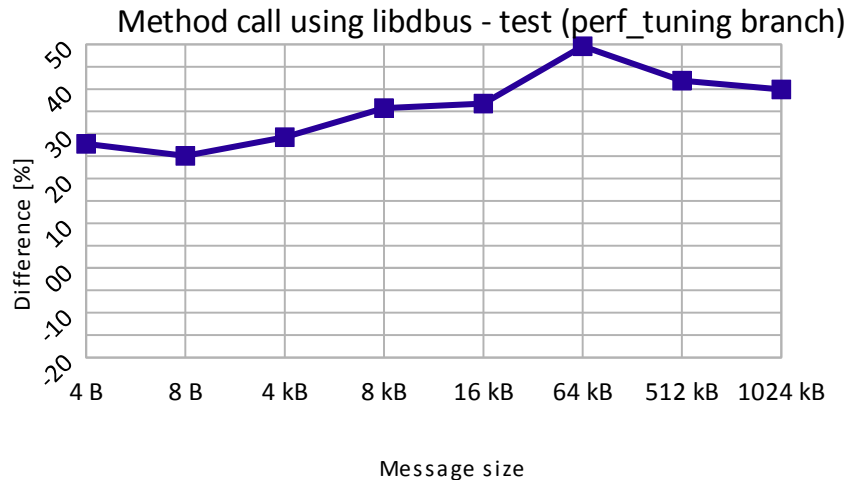
- **Project goals**

- Transparency  
(No need to modify D-Bus based codes)
- Compatible with native D-Bus
- Improve IPC performance of a Tizen product

# kdbus performance test results

The measurement was made by performing one thousand of calls and computing a sum of duration of every call.

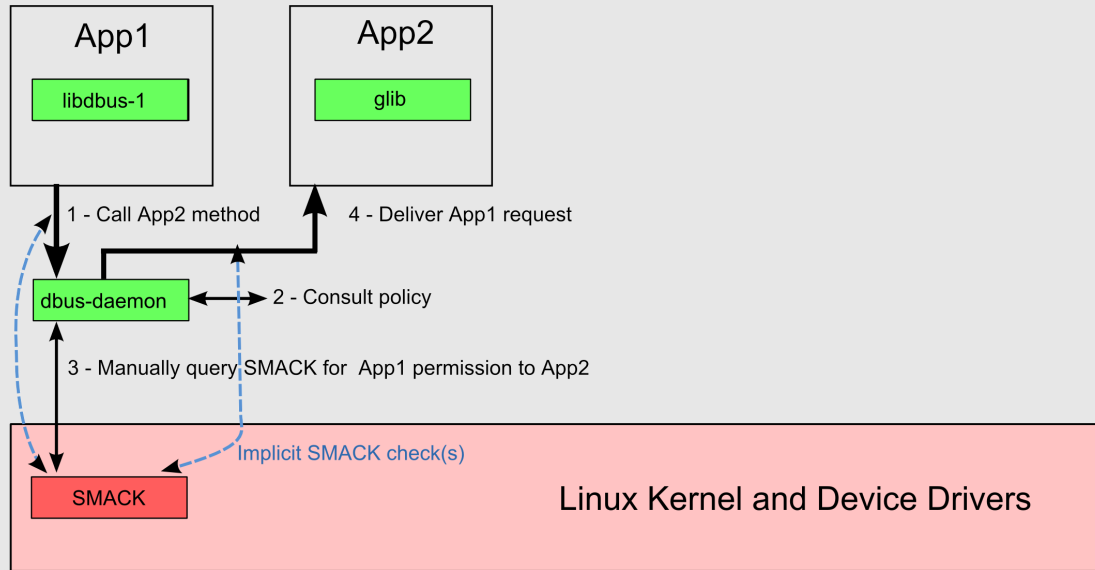
Method call using libdbus				
msg_size	default [s]	kdbus [s]	diff [s]	diff [%]
4 B	0.371	0.268	0.103	27.763
8 B	0.355	0.266	0.089	25.070
4 KB	0.438	0.310	0.128	29.224
8 KB	0.546	0.351	0.195	35.714
16 KB	0.707	0.447	0.260	36.775
64 KB	1.937	0.977	0.960	49.561
512 KB	16.88	9.816	7.064	41.848
1024 KB	37.239	22.384	14.855	39.891



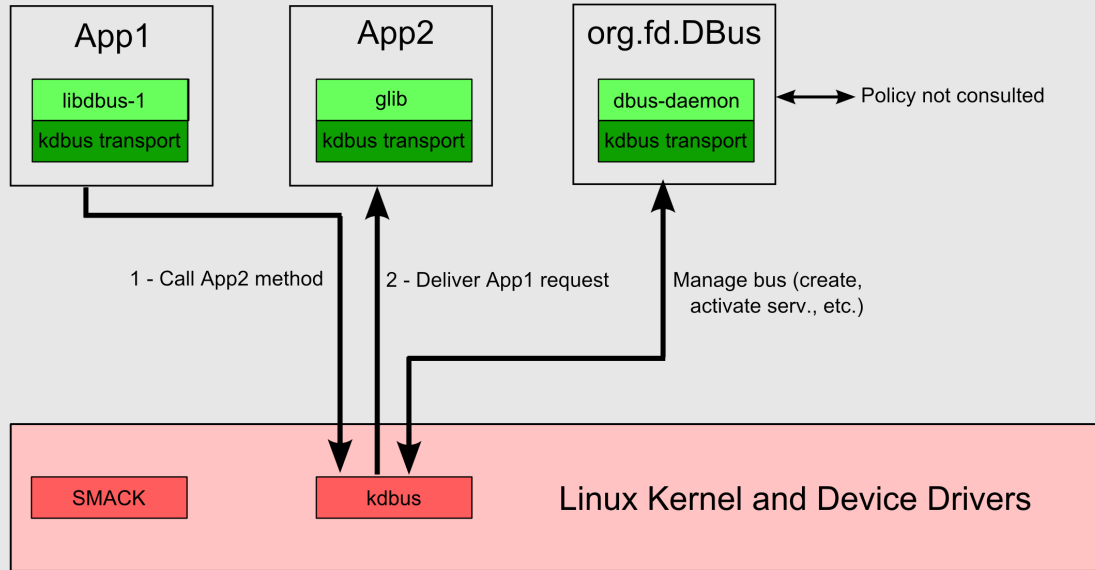
First attempts



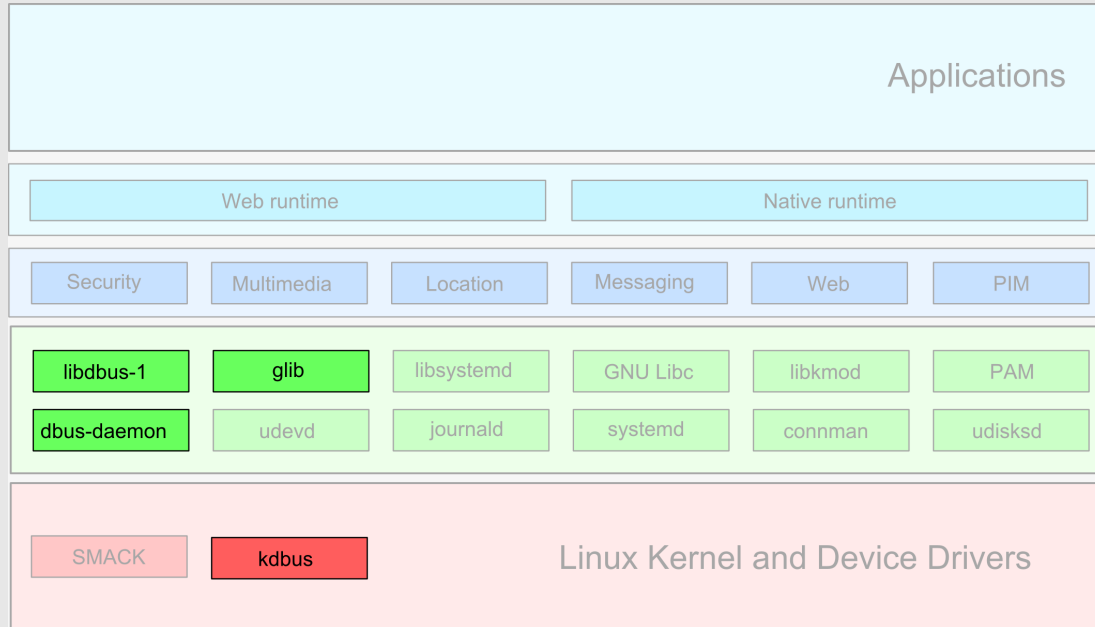
# Legacy D-Bus in Tizen



# kdbus-enabled dbus-daemon



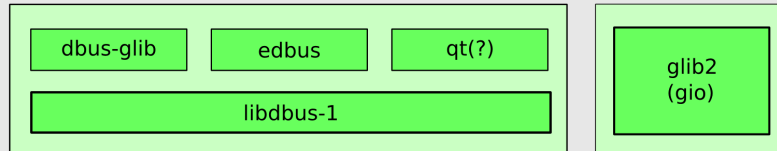
# Legacy architecture





# D-Bus libraries in Tizen

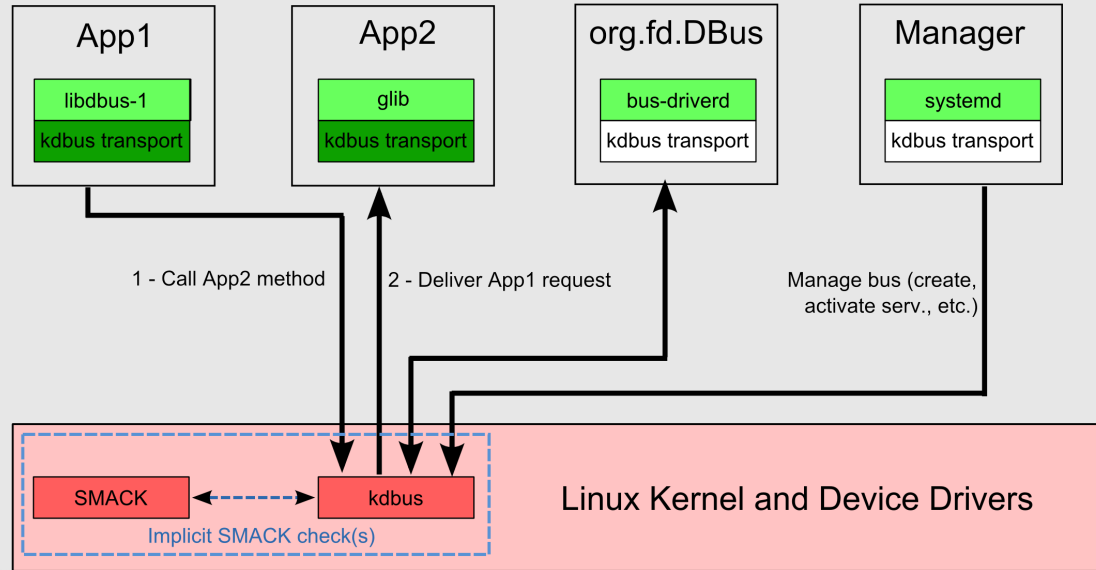
- **Tizen provides limited number of D-Bus binding libraries**
  - Most of these build on libdbus-1 foundation
  - Currently only libdbus-1 and glib2 need to be ported to kdbus



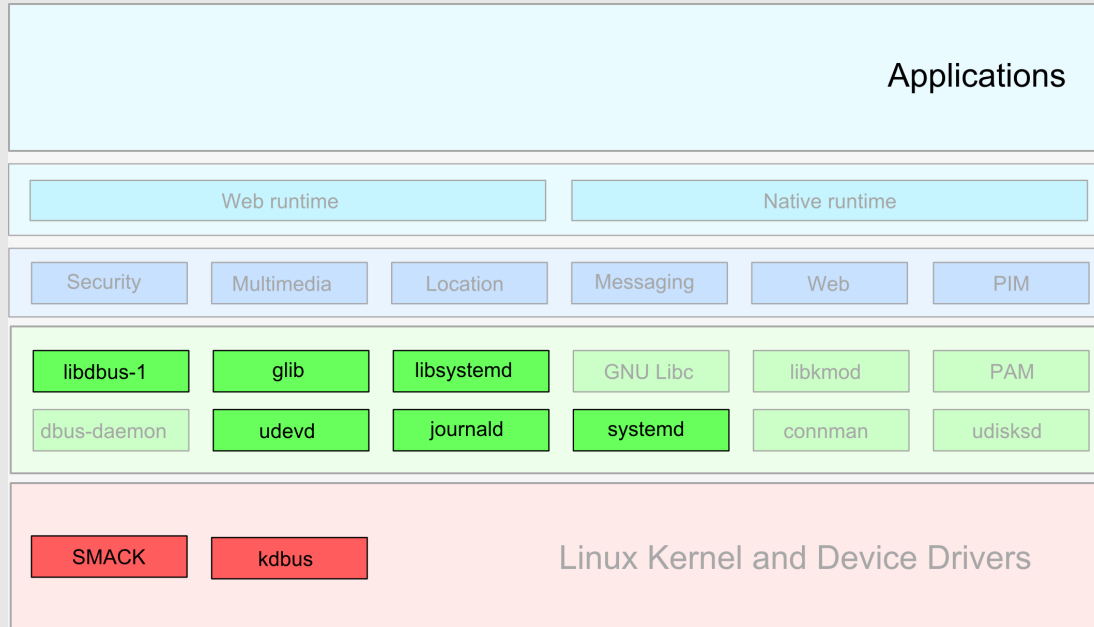
A decorative graphic in the top right corner of the slide. It features a large white circle, a smaller white circle above it, and a blue and white striped circle to the right. The background of this graphic is a stylized cityscape with a suspension bridge and buildings, set against a blue sky and water. The graphic is surrounded by scattered blue and green confetti dots.

kdbus in Tizen 3.0

# Tizen 3.0 architecture with kdbus



# Tizen 3.0 architecture with kdbus



# systemd

- **systemd  $\geq$  v209 handles kdbus natively:**
  - Creates and manages both system and user buses
  - Transparently handles transition from legacy dbus to kdbus (service generators, service masking) at boot time
- **Additionally provides:**
  - sd-bus (libsystemd) library supporting both AF\_UNIX and kdbus transports
  - bus-proxyd – compatibility AF\_UNIX socket for legacy clients
  - bus-driverd – “org.freedesktop.DBus” support (but see following slides)

# Native glib (gio) kdbus port

- **All major features present:**
  - Exchanging messages, broadcasting, signals, name reg., etc.
  - Makes use of kdbus-specific functionalities (memfd, bloom filters)
- **No changes in glib's API**
- **Development closely follows upstream**
- **Progress tracked on [bugzilla.gnome.org](http://bugzilla.gnome.org)**
- **glib RM agreed to include it in next dev. version of glib**



# Native libdbus-1 port

- **Originally designed to work with kdbus-enabled dbus-daemon**
- **Currently under active redesign & development**
  - Not up to date with current day kdbus
  - Uses “dbus-1” serialization on bus instead of GVariant
- **Requires a lot of work to become in shape for upstreaming**

# Security policy

- **Simplified policy architecture**
  - New Tizen services tend to perform policy checks by themselves (via policykit, cynara, etc.)
  - Dropping dbus-daemon allows us to kill overly complicated DBus policies
  - Per-destination policy checks fit perfectly in kdbus model
  - Existing Smack policies can be reused
- **Requires simple extensions to Linux LSM and kdbus**

# kdbus-lsm security hooks

- **New set LSM hooks**
  - security\_kdbus\_send()
  - security\_kdbus\_recv()
  - security\_kdbus\_talk()
  - security\_kdbus\_name\_acquire()
  - security\_kdbus\_name\_list()
  - ...
- **Preliminary SMACK implementation suggested**
- **Ongoing discussion with kdbus, SMACK, SELinux communities**

# kdbus to Tizen 3.0 (summary of changes)

- **Introducing kdbus requires:**
  - **Upgrading** systemd  $\geq$  209 (most likely - v212)
  - **Introducing** kdbus kernel module
  - **Patching** glib (gio) for native kdbus port
  - **Patching** libdbus-1 for native kdbus port
  - **Patching** kernel(s) and kdbus for kdbus-lsm security hooks
- **kdbus support to be enabled at build time**
- **Patches available on kdbus-integration branches**

# Challenges



# kdbus, systemd evolving rapidly

- **No API/ABI guarantees**
- **Major features still under active development:**
  - memfd moving to out of kdbus to generic kernel facility
  - **bus-driverd** dropped from systemd > 212
- **Integration with Linux kernel might drastically change landscape**



# Future work

- **Tizen is going to use systemd-based kdbus stack only**
  - kdbus-enabled dbus-daemon future is uncertain
- **Native libdbus-1 kdbus port still requires a lot of work**
- **kdbus-lsm patches:**
  - Long way to integration
  - kdbus' ability to “own a name” doesn't map well to SMACK model
- **No known good methods for comprehensive system-wide testing**

# Beyond Tizen

- **systemd >= 213 will drop bus-driverd**
  - org.freedesktop.DBus will no longer be available on kdbus (while retaining it on legacy socket)
  - Lack of org.freedesktop.DBus makes kdbus to not cover all D-Bus spec requirements
- **Upstream proposed changes to existing libraries**
  - Introduce **user** and **machine** bus types where kdbus might be available
  - Calling org.freedesktop.DBus would be explicitly disallowed on these buses
- **The above changes make kdbus opt-in, not a transparent replacement for D-Bus**

Thank you!



# References



# References

- **glib kdbus port**
  - [https://bugzilla.gnome.org/show\\_bug.cgi?id=721861](https://bugzilla.gnome.org/show_bug.cgi?id=721861)
- **Original kdbus-enabled dbus-daemon and libdbus-1**
  - <git://review.tizen.org/platform/upstream/dbus> kdbus-dev
- **libdbuspolicy-1 library**
  - <git://review.tizen.org/platform/upstream/dbus> libdbuspolicy-dev
- **kdbus-lsm patches**
  - <git://github.com/lmctl/linux> kdbus-lsm-dev
  - <git://github.com/lmctl/kdbus> kdbus-lsm-dev

# References

- **Rationale behind dropping bus-driverd**
  - <http://permalink.gmane.org/gmane.comp.sysutils.systemd.devel/18514>
  - [https://bugzilla.gnome.org/show\\_bug.cgi?id=721861#c24](https://bugzilla.gnome.org/show_bug.cgi?id=721861#c24)
- **D-Bus vs kdbus comparision**
  - <http://mindlinux.wordpress.com/2014/02/01/anatomy-of-kdbus-lenart-poettering/>
- **Tizen**
  - <http://en.wikipedia.org/wiki/Tizen>
  - <https://developer.tizen.org/>



**TIZEN**<sup>™</sup>  
**DEVELOPER  
CONFERENCE**  
2014  
**SAN FRANCISCO**