kdbus in Tizen 3.0

Hyungjun Choi
Karol Lewandowski

Samsung Electronics
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
- ...

![Diagram of D-Bus message flow]

1 - Call App2 method
2 - Consult policy
3 - Deliver App1 request
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

<table>
<thead>
<tr>
<th>D-Bus (It’s inefficient)</th>
<th>kdbus (It’s efficient)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 copies</td>
<td>2 of fewer copies</td>
</tr>
<tr>
<td>4 complete validations</td>
<td>2 validations</td>
</tr>
<tr>
<td>4 context switches</td>
<td>2 context switches</td>
</tr>
<tr>
<td>Suitable only for control, not payload</td>
<td>Suitable for large data (GiB), zero-copy, optionally reusable</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>msg_size</th>
<th>default [s]</th>
<th>kdbus [s]</th>
<th>diff [s]</th>
<th>diff [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 B</td>
<td>0.371</td>
<td>0.268</td>
<td>0.103</td>
<td>27.763</td>
</tr>
<tr>
<td>8 B</td>
<td>0.355</td>
<td>0.266</td>
<td>0.089</td>
<td>25.070</td>
</tr>
<tr>
<td>4 KB</td>
<td>0.438</td>
<td>0.310</td>
<td>0.128</td>
<td>29.224</td>
</tr>
<tr>
<td>8 KB</td>
<td>0.546</td>
<td>0.351</td>
<td>0.195</td>
<td>35.714</td>
</tr>
<tr>
<td>16 KB</td>
<td>0.707</td>
<td>0.447</td>
<td>0.260</td>
<td>36.775</td>
</tr>
<tr>
<td>64 KB</td>
<td>1.937</td>
<td>0.977</td>
<td>0.960</td>
<td>49.561</td>
</tr>
<tr>
<td>512 KB</td>
<td>16.88</td>
<td>9.816</td>
<td>7.064</td>
<td>41.848</td>
</tr>
<tr>
<td>1024 KB</td>
<td>37.239</td>
<td>22.384</td>
<td>14.855</td>
<td>39.891</td>
</tr>
</tbody>
</table>
First attempts
Legacy D-Bus in Tizen

1. Call App2 method
2. Consult policy
3. Manually query SMACK for App1 permission to App2
4. Deliver App1 request

Implicit SMACK check(s)

Linux Kernel and Device Drivers
kdbus-enabled dbus-daemon

1 - Call App2 method
2 - Deliver App1 request
Manage bus (create, activate serv., etc.)

SMACK

kdbus

Linux Kernel and Device Drivers
## Legacy architecture

<table>
<thead>
<tr>
<th>Applications</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Web runtime</td>
<td>Native runtime</td>
</tr>
<tr>
<td>Security</td>
<td>Multimedia</td>
</tr>
<tr>
<td>Location</td>
<td>Messaging</td>
</tr>
<tr>
<td>Messaging</td>
<td>Web</td>
</tr>
<tr>
<td>Web</td>
<td>PIM</td>
</tr>
<tr>
<td>Linux Kernel</td>
<td></td>
</tr>
<tr>
<td>and Device Drivers</td>
<td></td>
</tr>
</tbody>
</table>

- **libdbus-1**
- **dbus-daemon**
- **glib**
- **libsystemd**
- **GNU Libc**
- **libkmod**
- **PAM**
- **udev**
- **journald**
- **systemd**
- **connman**
- **udisksd**
- **SMACK**
- **kdbus**
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
kdbus in Tizen 3.0
Tizen 3.0 architecture with kdbus
Tizen 3.0 architecture with kdbus
systemd

- systemd >= 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 (libraries) 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
• 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 >= 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

- **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](git://review.tizen.org/platform/upstream/dbus kdbus-dev)

- **libdbuspolicy-1 library**
  - [git://review.tizen.org/platform/upstream/dbus libdbuspolicy-dev](git://review.tizen.org/platform/upstream/dbus libdbuspolicy-dev)

- **kdbus-lsm patches**
  - [git://github.com/lmctl/linux kdbus-lsm-dev](git://github.com/lmctl/linux kdbus-lsm-dev)
  - [git://github.com/lmctl/kdbus 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

• 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/