Tizen 3 IVI Architecture
Multi User App FW

Dominig ar Foll
(Intel Open Source Technology Centre)
dominig.arfoll@fridu.net

02 Feb 2014
Tizen 3

- What is an In Vehicule Infotainement (IVI)
- Tizen Project Open development
- Tizen Architecture Overview
- Multi-user challenge
- Tizen 3 AppFW
- Tizen 3 User Mgt
FOSDEM'14 Automotive devroom – Tizen Multi User Application Framework
Tizen 3 Open Project

- Architects
- Maintainers
- Integrators
- Reviewers
- Developers

- Release Engineer
- QA Engineer

TSG Arch Forum

API Forum

Domain
- Architects
- Maintainers
- Integrators
- Reviewers

Domain
- Architects
- Maintainers
- Integrators
- Reviewers

package
package
package
package

FOSDEM'14 Automotive devroom – Tizen Multi User Application Framework
Tizen 3 : Code contribution Flow

Developer
- Write code
- Submit to Gerrit

Reviewer
- Offer suggestions
- Review
  - Not OK
  - OK
- Approve contribution
- Integrate contribution
- Release to build system

Maintainer [Integrator]
- Maintain can override code acceptance at any step
- Approve contribution

Release engineer
- Create image
- Smoke Test
  - Not OK
  - OK
- Release image
General Architecture

- Fastboot
- Embedded Security
- Connected
- HTML5
- Specialised IVI Middleware
Architecture Overview (Mobile Profile)

- Applications
  - Web Applications
  - Native Applications

Web Framework
- W3C/HTML5
  - Video
  - Touch
  - CSS3
  - WebGL
  - Worker
- Device APIs
  - Push
  - Contact
  - Noti
  - NFC
  - SystemInfo
- Web UI F/W
- Web Runtime

Native Framework
- Social/Content
- Locations
- Uix
- Media
- Web/Xml
- Net/Telephony/Messaging
- Graphics/UI
- Base/IO/Text/Locales
- App/Security/System Services

Core
- Application Framework
- Security
- System
- Base
- Connectivity
- Telephony
- PIM

Core Applications
- Graphics & UI
- Multimedia
- Web
- Messaging
- Location

Kernel
- Linux Kernel & device drivers

Manufacturer Adaptation Interface

FOSDEM'14 Automotive devroom – Tizen Multi User Application Framework
Architecture Overview (IVI profile)

Applications

Web Applications

Web Framework

Web Runtime

Core

Application Framework

Multimedia

Web

Messaging

Location

Security

System

Base

Connectivity

Telephony

PIM

Kernel

Linux Kernel & device drivers

FOSDEM'14 Automotive devroom – Tizen Multi User Application Framework
Tizen Multi-user system
Tizen Multi-user requirement

- Guest log in by default
  - Start Generic Application (e.g. rear cam, radio, ...)

- User logging in shall not stop running applications
  - ID user is added on top of Guest.
  - Multiple user can share the same Display (e.g. passenger and drivers)
  - Users can exchange seats (and so Display)

- Security must protect the user data and the system data.
Disociate Seat and User

- **General Linux**
  - Before Login → No use
  - Seat = Display

- **IVI**
  - Before Login → Guest
  - Login → add user to a seat
  - User can change seat
What needs to change

- Application Framework
- Login Manager
- Startup procedure
- Security model
Application Framework

- **Launching Application**
  - Explicit or implicit information (Combination of Action, URI, and MIME) can be used to determine an app to launch and the control backend.
  - Allowed to launch different type of app (i.e. Web to Native and Native to Web)

- **Application life cycle management**
  - Install – delete – update
  - List all, recently launch – Search – swap active

- **Event Handler**
  - Screen orientation size and number, dim, off
  - Critical resources RAM, disk, batterie, ...

- **Inter App communication**
  - Service Req
  - Message exchange
  - Copy/Paste
  - Drag/Drop
Multiple back-ends

![Diagram of Application Framework]

- **Tizen App Run Time**
- **Communications**
  - WRT backend
  - Core Backend
  - Cloud Backend
- **Security**
- **Event handler**
Tizen 3.x : Launching App (general model)

- App svc
- D-Bus Request

AMD
Check all App Info

Launchpad
Set environment
Forked as user

User env

Hybrid pre-launch
WRT
Native-core

Launched
Tizen 3: Single Launcher – Multi User

**AMD**
- Accumulate environment information for each logged user (a PAM module could be a good entry point for initial env. collection)
- Serve launch request (by name, ID, Mime type, ...)
- Ask user for choice when multi application can serve the request.
- Verify if Launch is authorised for the user (profile) in current context (car in movement, battery status, data roaming, ....)
- Subcontract the launch to a specialised launcher (launchpad) per App type (OSP, WRT, Core native)

**Launchpad**
- Three launchpads can be foreseen (OSP, WRT, Core Native).
- Receive all the context (sécurity, environment, ...) from AMD.
- Insert the Application at launch in the user context
- Manage lib and info caching to speed launch (preload, pre-link, ...)
- Manage App Live cycle (could also be in AMD) (kill, pause, slowdown, garbage collection)

---

**AMD & Launchpad run with privilege**
Launching Sequence | (1/2)

- Preload daemon on system session
  - CoreApp / NativeApp

```
UID 7001
Request

Application Management Daemon

SO_PEERCRED
- pid
- uid
- gid

/tmp/alaunch/-1

set_app_privilege(uid,
set_env(user-env)

set_app_privilege(uid)
set_env(user-env)

set_dac(uid)
setgroups()
setgid(APP_GID)
setuid(uid)

Launchpad
```

```
Launchpad
```

```
libprivilege-control.so
set_app_privilege(uid)
set_dac(uid)
```

```
{uid, user-env}

/system session
```

```
/user session
```

FOSDEM'14 Automotive devroom – Tizen Multi User Application Framework
• Preload daemon on system session
  - WebApp
AppFW with TLM

SystemD

SystemD --user

Display Server

TLM

PAM

SystemD --user

Home user(s) shell

AMD

Launchpad(s)
Impact

● Single AMD/Launchpad running with privileges
  • User session ENV must be known
  • UID is extracted from Launch request
  • App Launch control is simple
  • Resource optimisation
  • Simplification to Tizen DB access

Issues

• Apps are added in user session from outside
• Some ENV variables are created only in user session, e.g. d-bus, display, ...
• AMD or Launchpad need to manage App live cycle and do the App garbage collector
Soft coded paths

- All positions are platform defined
  - Position defined by a unique source code
  - Apps user Env variables for all location
  - XDG names are use when defined by Freedesktop.org

- All user dir are relative to $HOME
  - Created at run time (no pre-installation)
  - Music, Photos, videos, ...
  - Apps (private and shared)
  - Vconf (equivalent) private Data
  - Apps data remain private
  - Documents (e.g. Media) shared or public

- Tizen DB
  - Position are absolute
  - User add/remove utilities
Add / remove / Activate User

- Support creating, editing and deleting users and groups through a D-Bus API
- Support GUI-less session management
- Support auto-login of guest and default user accounts
- Support clean-up of guest environment
- Support via PAM plug-ins multiple authentication credentials per user (key fob, NFC, password/pin code)
- Support auto-relogin after short power outages (IVI)
gumd / libgum
(user management daemon / lib)

Daemon providing D-Bus and GObject-based client APIs to
- Create, modify and delete users
- Create, modify and delete groups
- Assign permissions to users through groups
- Support shared group folders
- Support additional user information like avatar images and nick names

API access controlled through SMACK - caller doesn’t need root privileges or execute setuid binaries

Better implementation model compared to running scripts from a GUI
tlm (tizen login manager)

- Creating default sessions at startup
- GUI-less user switching through hardware authentication (key fob, fingerprint, etc)
- Support simultaneous sessions on multiple seats
- Suspend/resume of sessions on different seats
- Cleanup of guest data on logout
- Manages session namespaces
Pointers
Links – Tizen

- Main site: tizen.org
- Tizen Association: www.tizenassociation.org
- Applications Development: developer.tizen.org
- Platform Development: source.tizen.org
- Snapshots: download.tizen.org
- Documentation: developer.tizen.org/documentation
- Wiki: wiki.tizen.org
- Bugs: bugs.tizen.org
Links – Tizen IVI

- General info: wiki.tizen.org/wiki/IVI
- GENIVI: genivi.org
Links – Tizen build tools

- Development tools: download.tizen.org/tools
- GBS: source.tizen.org/documentation/reference/git-build-system
- MIC: source.tizen.org/documentation/reference/mic-image-creator
- OBS: openbuildservice.org
- GERRIT: code.google.com/p/gerrit
- GIT: git-scm.com
Upstream projects

- Linux Kernel, SMACK, systemd, dbus
- OpenSSL, Sqlite
- X, Wayland, EFL, Enlightenment, Cairo
- Connman, BlueZ, oFono, wpa_supplicant
- Gstreamer, PulseAudio
- Webkit
- Eclipse (SDK)
- Qemu, U-boot (emulator)
- GCC, llvm, cmake, git (build)
- ... and more ...