

The State of AGL Plumbing and Services

Scott Murray and Matt Porter
scott.murray@konsulko.com, mporter@konsulko.com



About Us

- Scott Murray
 - Linux user/developer since 1996
 - Embedded Linux developer starting in 2000
 - Principal Software Engineer at Konsulko Group
- Matt Porter
 - Linux user/developer since 1992
 - Embedded Linux became my full-time job starting 1999 at Motorola
 - CTO at Konsulko Group

Syllabus

- Overview of AGL
- Release history
- Current and planned features
- Build system and organization
- Key plumbing components
- AGL application framework
- AGL application APIs (bindings)
- Roadmap
- Community

Overview

Overview of AGL

- Automotive Grade Linux is an embedded Linux distribution targeting IVI and ADAS products
- Based on OpenEmbedded build system and Yocto Project Poky reference distribution
- Provides an application framework with a software cross-development kit
- Goal is to provide a secure application runtime environment and a uniform set of APIs meeting automotive and mobility use cases
- Intended to provide a common base distribution for products

History and Features

Release History

- Agile Albacore - January 2016
 - MOST driver and demo apps for HomeScreen/HVAC/Media/Navigation/Radio
- Brilliant Blowfish - July 2016
 - Application framework and Audio Routing
- Charming Chinook - January 2017
 - Bluetooth/WiFi/Radio Bindings, SDK, and additional BSP support
- Daring Dab - July 2017
 - Application framework v2, CAN/Mediascanner/Telephony Bindings, SmartDeviceLink
- Electric Eel - January 2018

Current and Planned Features

- Current
 - Application framework
 - Some core APIs (various connectivity APIs covered later)
 - Audio routing
 - Demo applications (in-tree Qt/QML based, out-of-tree HTML5)
- Future
 - Audio API
 - MediaPlayer API
 - WindowManager API
 - HomeScreen API
 - Storage API

Build System and Distribution Organization

Distribution Details

- Based on Yocto Project Poky Distribution, pyro release
- Layers
 - oe-core
 - meta-openembedded/*
 - meta-intel-iot-security
 - meta-security-framework (Cynara)
 - meta-security-smack (SMACK)
 - meta-agl
 - meta-agl-app-framework
 - meta-agl-bsp
 - meta-agl-distro
 - meta-agl-ivi-common
 - meta-qt5
 - meta-agl-demo
 - BSP Layers (vendor or community)
 - e.g. meta-freescale, meta-renesas-r-car-gen3, meta-ti

Plumbing and Services

Plumbing

- Service and application lifecycle
 - systemd
 - Each application is a service
 - AGL is considering moving to dynamic users
- Audio
 - ALSA
 - Pulseaudio
 - GENIVI AudioManager
 - Used for policy-driven audio routing in conjunction with an out-of-tree Pulseaudio module
- Graphics
 - Wayland
 - Weston with IVI shell

Plumbing (continued)

- Bluetooth
 - Bluez5
- Location
 - gpsd
 - geoclue
- Telephony
 - ofono
- Networking
 - connman
 - wpa_supplicant

Application Framework

What is the application framework?

- The AGL application framework provides a sandboxed application execution environment.
- Implements a complete application lifecycle for install and runtime control of applications
- Provides a secure environment using systemd cgroups, SMACK, Cynara, and a Cynara-enabled D-Bus daemon.
- Provides Websocket interface to bindings (application APIs)
- Applications are packaged according to W3C Widget guidelines
 - <https://www.w3.org/TR/widgets/>
- More information
 - http://docs.automotivelinux.org/docs/apis_services/en/dev/reference/af-main/0-introduction.html

Binding Overview

- The API binding mechanism abstracts an application's UI from its back end logic
- This allows re-using application logic with different UI implementations (e.g. Qt and HTML5)
- Allows the application framework to control access to APIs and resources in a fine-grained manner, effectively sandboxing applications based on their API permissions.
- End goal is to provide a complete and consistent API for AGL applications
- More information
 - http://docs.automotivelinux.org/docs/apis_services/en/dev/reference/af-binder/afb-overview.html

Binding Registration

- Binding implementation is a shared library
- A binding implementation:
 - Registers a unique binding **api** name
 - Registers a list of binding verbs to perform actions
 - Contains the binding verb/event backend implementation
 - Optionally registers **preinit** and **init** routines for the binding
 - Optionally registers a **specification** containing an OpenAPI v3 description of the binding
 - Optionally registers a text **info** description of the binding
 - Optionally registers an **onevent** callback for handling subscribed events
 - Optionally set **noconcurrency** flag to avoid concurrent verb calls

Application and Binding Initialization

- Application and binding packaging (widget) includes a config.xml file that:
 - Specifies the name, description, author, license
 - Lists any permissions that the package requires
 - Lists any bindings that the package requires
 - Lists any bindings that the package provides
- Application framework spawns an instance of afb-daemon
 - Loads and initializes the specified bindings
 - Executes the application, passing port number and authentication token arguments to it for binding access
 - Important to remember that each instance of the binding is separate
- More details
 - http://docs.automotivelinux.org/docs/apis_services/en/dev/reference/af-main/config.xml.html
 - http://docs.automotivelinux.org/docs/apis_services/en/dev/reference/af-binder/afb-bindings-overview.html

Application Binding Usage

- Submit requests in JSON format via HTTP or WebSocket
 - e.g. [2, "9999", "hvac/set", { "LeftTemperature" : 16}]
- Receive request status (success or failure) and any additional requested data
- Responses are also in JSON format
- Can subscribe / unsubscribe for events
- Events arrive asynchronously via WebSocket
- More details
 - http://docs.automotivelinux.org/docs/apis_services/en/dev/reference/af-binder/afb-bindings-writing.html

APIs

Upstream AGL Bindings

- Master Binding
- Bluetooth Binding
- WiFi Binding
- Radio Binding
- Telephony Binding
- MediaScanner Binding
- MediaPlayer Binding
- GPS Binding
- GeoClue Binding
- GeoFence Binding

WIP Bindings

- Audio Bindings
- New HomeScreen/WindowManager Bindings
- CAN Bindings

Master Binding

- Features
 - Application lifecycle facilities
 - Install
 - Uninstall
 - Start
 - Terminate
 - Pause
 - Resume
 - List
 - State

Master Binding API

- Verbs

- **afm-main/runnables** - list runnable apps
- **afm-main/detail** - info on app
- **afm-main/start** - start an app
- **afm-main/once** - start an app once
- **afm-main/terminate** - terminate an app
- **afm-main/pause** - pause an app
- **afm-main/resume** - resume an app
- **afm-main/runners** - list running apps

- Verbs (continued)

- **afm-main/state** - get state of an app
- **afm-main/install** - install an app
- **afm-main/uninstall** - uninstall an app

- Events

- None

Bluetooth Binding

- Features
 - Device discovery, pairing, connection, and settings
 - Device priority list
 - AVRCP Bluetooth binding controls
 - Media metadata, and position tracking
- Future
 - Clean up

Bluetooth Binding API

- Verbs (**Bluetooth-manager/***)

- **power** - set power on/off
- **start_discovery** - start device discovery
- **stop_discovery** - stop device discovery
- **discovery_result** - get discovered devices
- **remove_device** - remove a device
- **pair** - start pairing process
- **cancel_pair** - cancel pairing process
- **connect** - connect to device
- **disconnect** - disconnect from device

- Verbs (continued)

- **device_priorities** - get device priorities list
- **set_device_property**
- **set_property** - set bluetooth property
- **set_avrcp_controls** - avrcp control
- **send_confirmation** - confirm PIN
- **subscribe** - subscribe event
- **unsubscribe** - unsubscribe event

- Events

- **connection** - connection changed
- **device_added**
- **device_removed**
- **device_updated**

WiFi Binding

- Features
 - Discovers WiFi APs
 - Connect and Disconnect from APs
 - WPA2 passkey input
 - Connection status
 - Manages network connections
- Future
 - Clean up
 - Rewrite as a provider to a high level Network Management Binding

WiFi Binding API

- Verbs

- **wifi/activate** - Activate WiFi
- **wifi/deactivate** - Deactivate WiFi
- **wifi/scan** - Scan Wifi
- **wifi/scan_result** - Get scan result
- **wifi/connect** - connect to AP
- **wifi/disconnect** - disconnect from AP
- **wifi/status** - status of AP connection
- **wifi/insertpasskey** - supply AP passkey
- **wifi/subscribe** - subscribe event
- **wifi/unsubscribe** - unsubscribe event

- Events

- **wifi/passkey** - passkey requested
- **wifi/networkList** - AP list changed

Radio Binding

- Features

- Radio binding based on rtl-sdr SDR FM demodulation code previously used to build the QtMultimedia plugin from the Chinook release
- Additional hooks added to FM demodulation code to add scanning support
- Radio QML application reworked to use binding in place of QtMultimedia QRadio class
 - Only minimal changes were required, the QML interface for the binding emulates QRadio's interface to a large degree
 - Application enhanced to add scanning support

- Future

- Additional tuner hardware support
- Metadata support (e.g. RDS)
- HD Radio support

Radio Binding API

- Verbs

- **radio/frequency** - get/set frequency
- **radio/band** - get/set band
- **radio/band_supported** - check band support
- **radio/frequency_range** - get band frequency range
- **radio/frequency_step** - get band frequency step
- **radio/start** - start audio
- **radio/stop** - stop audio

- Verbs (continued)

- **radio/scan_start** - start scanning
- **radio/scan_stop** - stop scanning
- **radio/stereo_mode** - get/set stereo mode
- **radio/subscribe** - subscribe event
- **radio/unsubscribe** - unsubscribe event

- Events

- **radio/frequency** - frequency has changed
- **radio/station_found** - scanning has found a station

Telephony Binding

- Features
 - Bluetooth Hands-Free Profile (HFP) device support
 - Originate a voice call
 - Answer an incoming voice call
 - Provide status and information on voice call connections
 - Depends on ofono, bluez, and pulseaudio
- Future Development
 - In-call sending of dial tones (for conference bridges, etc.)
 - Call waiting/hold/forwarding
 - Voice modem support

Telephony Binding API

- Verbs

- **telephony/dial** - dial a phone call
- **telephony/hangup** - hangup an active phone call
- **telephony/answer** - answer an incoming phone call
- **telephony/subscribe** - subscribe event
- **telephony/unsubscribe** - unsubscribe event

- Events

- **telephony/callStateChanged** - state of a phone call has changed
- **telephony/incomingCall** - incoming call is ringing
- **telephony/dialingCall** - outgoing call is being dialed
- **telephony/terminatedCall** - call has been terminated

MediaScanner Binding

- Features
 - Media binding to report media insertion/removal
 - Media detection and path reporting
 - Receive metadata from Bluetooth binding
 - Depends on lightmediascanner

MediaScanner Binding API

- Verbs

- **mediascanner/media_result** - get all available multimedia
- **mediascanner/subscribe** - subscribe event
- **mediascanner/unsubscribe** - unsubscribe event

- Events

- **mediascanner/media_added** - media is attached to the device
- **mediascanner/media_removed** - media is removed from device

MediaPlayer Binding

- Features
 - Media audio playback and control
 - Depends on GStreamer
- Future
 - Video playback

MediaPlayer Binding API

- Verbs

- **mediaplayer/playlist** - get/set playlist
- **mediaplayer/controls** - playback controls e.g. play, pause, etc.
- **mediaplayer/metadata** - get metadata of current track
- **mediaplayer/subscribe** - subscribe event
- **mediaplayer/unsubscribe** - unsubscribe event

- Events

- **mediaplayer/metadata** - position/duration of current track
- **mediaplayer/playlist** - playlist changed

GPS Binding

- Features
 - Provides GNSS location data
 - Latitude
 - Longitude
 - Altitude
 - Speed
 - Time
 - Depends on gpsd

GPS Binding API

- Verbs

- **gps/location** - Get GNSS data
- **gps/subscribe** - subscribe event
- **gps/unsubscribe** - unsubscribe event

- Events

- **gps/location** - GNSS data updated

GeoClue Binding

- Features
 - Provides GeoClue location data
 - Latitude
 - Longitude
 - Altitude
 - Speed
 - Heading
 - Time
 - Supports gathering location data from multiple sources:
 - WiFi AP databases
 - 3g/4g 3GPP tower information
 - GeoIP database
 - GPS
 - Depends on GeoClue

GeoClue Binding API

- Verbs

- **geoclue/location** - Get GeoClue location data
- **geoclue/subscribe** - subscribe event
- **geoclue/unsubscribe** - unsubscribe event

- Events

- **geoclue/location** - GeoClue data updated

GeoFence Binding

- Features
 - Add/remove/list geographic bounding boxes
 - Generates enter/leave events when ingress/egress occurs in fenced bounding box
 - Generates event indicating that a user is “dwelling” at a location based on a configurable timeout
 - Depends on GPS binding for location data
- Future
 - Add support for GeoClue binding location data
 - Add support for per-fence dwell transition timing
 - Convert “dwell”, “entered”, and “exited” to separately subscribed events

GeoFence Binding API

- Verbs

- **geofence/add_fence** - add a geofence bounding box
- **geofence/remove_fence** - remove a geofence bounding box
- **geofence/list_fences** - list all geofence bounding boxes
- **geofence/dwell_transition** - get/set dwell transition time
- **geofence/subscribe** - subscribe event
- **geofence/unsubscribe** - unsubscribe event

- Events

- **geofence/fence** - geofence event occurred

Next Steps

Roadmap

- Bluetooth PBAP support
 - Integration with telephony API
- Completion of MediaPlayer binding
- Video support for MediaPlayer binding
- Speech recognition / TTS binding
- WWAN modem binding
- Audio Bindings
- New HomeScreen/WindowManager Bindings

Getting Involved

Community

- IRC: #automotive on Freenode.net
- Mailing list:
<https://lists.linuxfoundation.org/mailman/listinfo/automotive-discussions>
- Weekly developer call: <https://wiki.automotivelinux.org/dev-call-info>
- JIRA: <https://jira.automotivelinux.org>
- Gerrit: <http://gerrit.automotivelinux.org/gerrit>

Resources

- Wiki
 - <https://wiki.automotivelinux.org/start>
- Developer Docs
 - Getting Started
 - http://docs.automotivelinux.org/docs/getting_started/en/dev/
 - APIs
 - http://docs.automotivelinux.org/docs/apis_services/en/dev/

Questions?