Smart Device Link

Integration into Linux systems

June 2017
Author: Jeremiah Foster
Open Source Technologist

LUXOFT

LXFT LISTED NYSE

Who am I?

- Jeremiah C. Foster, proud father of Hannah and husband of Annika
- Huge FOSS fanboi and Debian user
- Open Source Technologist for Luxoft
- GENIVI Community Manager

Purpose of this talk

- To let everyone know that it is possible to integrate iOS and Android smart devices into Linux based automotive systems
- To provide a route to collaboration and a justification of why collaboration is so important in the SDL case

What are we talking about?

- A standardized way to connect ‘smartphone’ devices, namely iOS and Android devices, to an in-vehicle infotainment system running Linux via ‘Smart Device Link’
- This requires software libraries on both the infotainment system and the smartphone

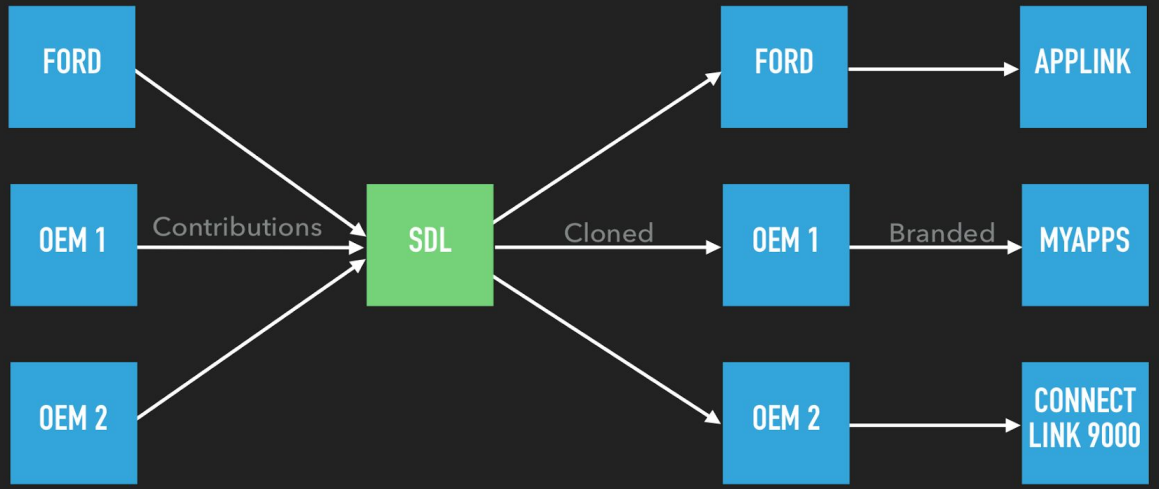
Smart Device Link is not AppLink

- “AppLink was originally a proprietary API created by Ford
- Ford announced they were contributing AppLink to the open-source under the name SmartDeviceLink in 2013. Purchased Livio
- Livio engineers are the project maintainers
- AppLink is now the branded version of SDL based off the open-source project”

SDL consortium

Created by Ford and Toyota the “SmartDeviceLink Consortium, is a nonprofit organization working to manage an open source software platform with the goal of giving consumers more choice in how they connect and control their smartphone apps on the road.”

SMARTDEVICELINK AND OEM BRANDS



SDL contribution model

- The consortium appears to be using a standard open source model using permissive licenses: “smartdevicelink/sdl_core is licensed under the BSD 3-clause "New" or "Revised" License
  A permissive license similar to the BSD 2-Clause License, but with a 3rd clause that prohibits others from using the name of the project or its contributors to promote derived products without written consent.”
- All SDL Code is on GitHub, all documentation is open
- Project evolution is based on Apple’s Swift Evolution and offers a high degree of transparency regarding decisions on what is included and what is rejected
- https://github.com/smartdevicelink/sdl_evolution

SDL USER EXPERIENCE

Project status

From the SDL GitHub README:

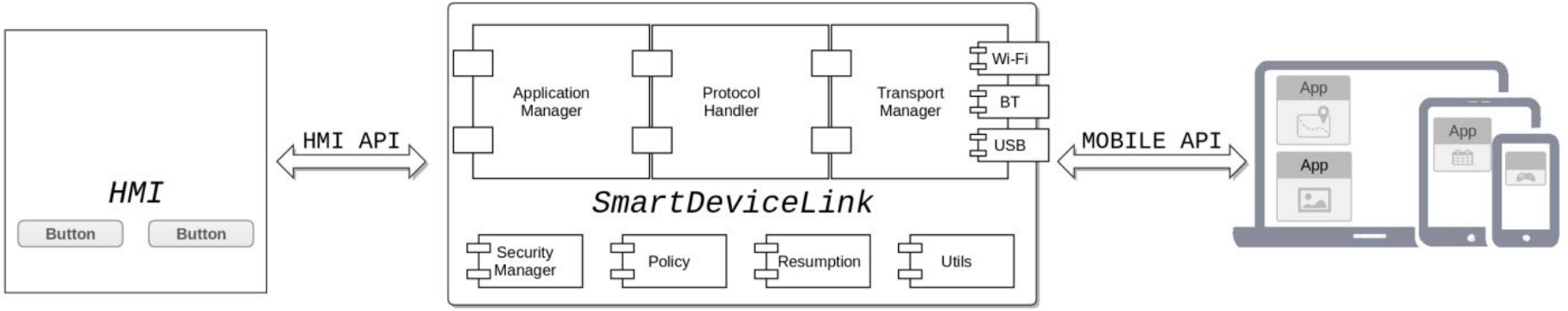
“We're ramping up our efforts to get SmartDeviceLink developed and maintained directly in the open. For the Mobile libraries, we're expecting better integration soon, SDL Core is slightly more complicated. We are currently working on generating documentation, creating a developer portal, an open forum, Mobile validation, and everything else that we've been asked for to renew the community's interest in this project. From a technical standpoint, SDL is stable, and the most work is being put into making it a more robust solution for app connectivity. We are, however, definitely looking for and interested in other people and company's contributions to SDL whether it be feature based, bug fixes, healthy conversation, or even just suggestions for improvement.”

Features of SDL

- Provides a Text to Speech (TTS) interface to allow drivers to keep their eyes on the road. Obviously this is a large safety benefit
- Provides the ability to control apps using SDL via the steering wheel buttons as well as TTS
- Attempts to be vendor agnostic with regard to device; supports both iOS devices and Android devices
- Allows for the control of user data on the head unit, including some analytics functions
- Provide a policy engine for apps using the head unit
- Allow the OEM to use their own interface and preserve their brand

Key requirements of SDL

- POSIX compliance for portability
- Transport protocol should be easy to replace, modifiable
- Well documented APIs



Architecture diagram

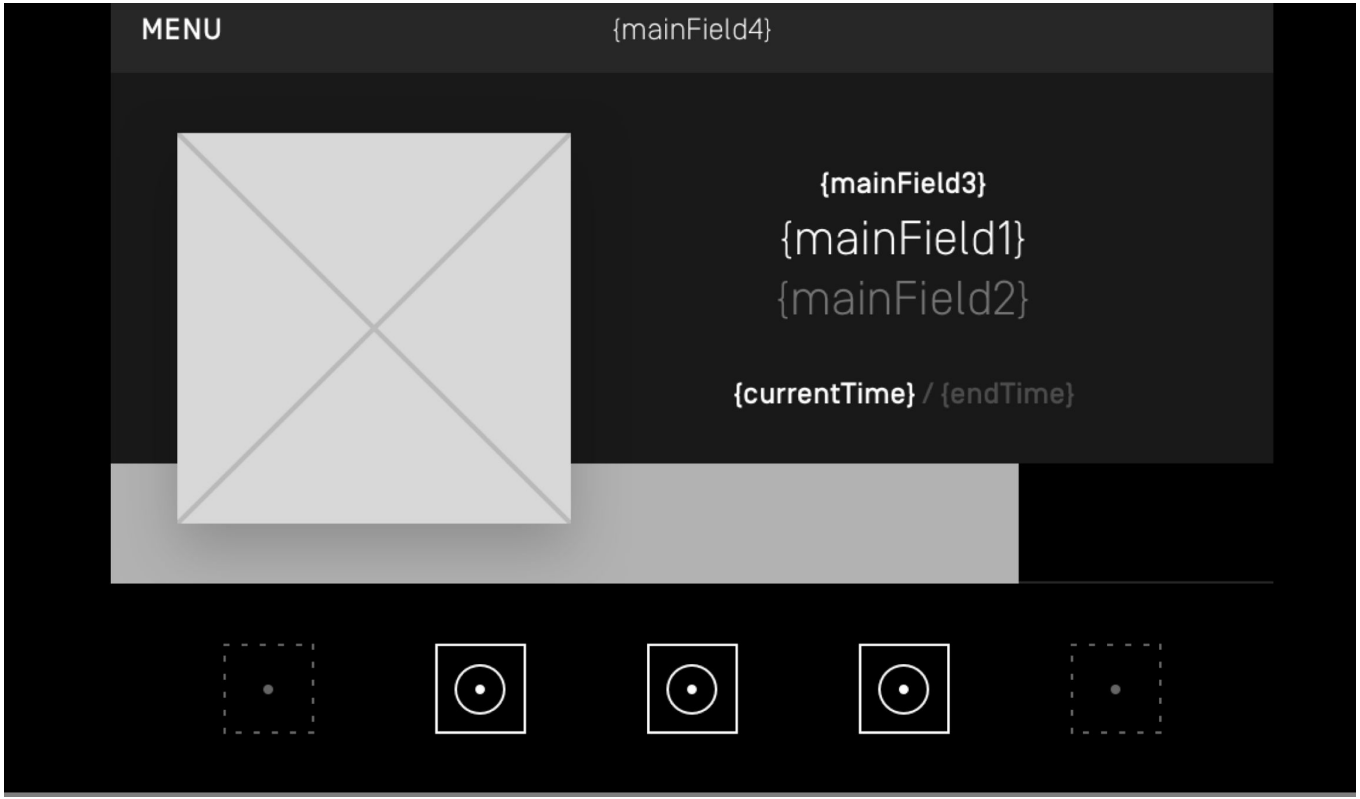
LXFT LISTED NYSE

LUXOFT

HMI

- SDL comes with a generic HMI
- Look and feel highly customizable
- One option is to use Qt for HMI; https://github.com/smartdevicelink/sdl_core/wiki/SDL-on-Linux-with-QT
  - Uses a rather old version of Qt (5.1), Qt now up to 5.9 with 5.10 due in November
  - Uses dbus for IPC as well as standard Qt modules like Qt Declarative
- Web based HMI
  - Depends on Chromium
  - Ember.js, Handlebars.js, jquery, native WebSocket libraries
  - Template based

Template example



MENU

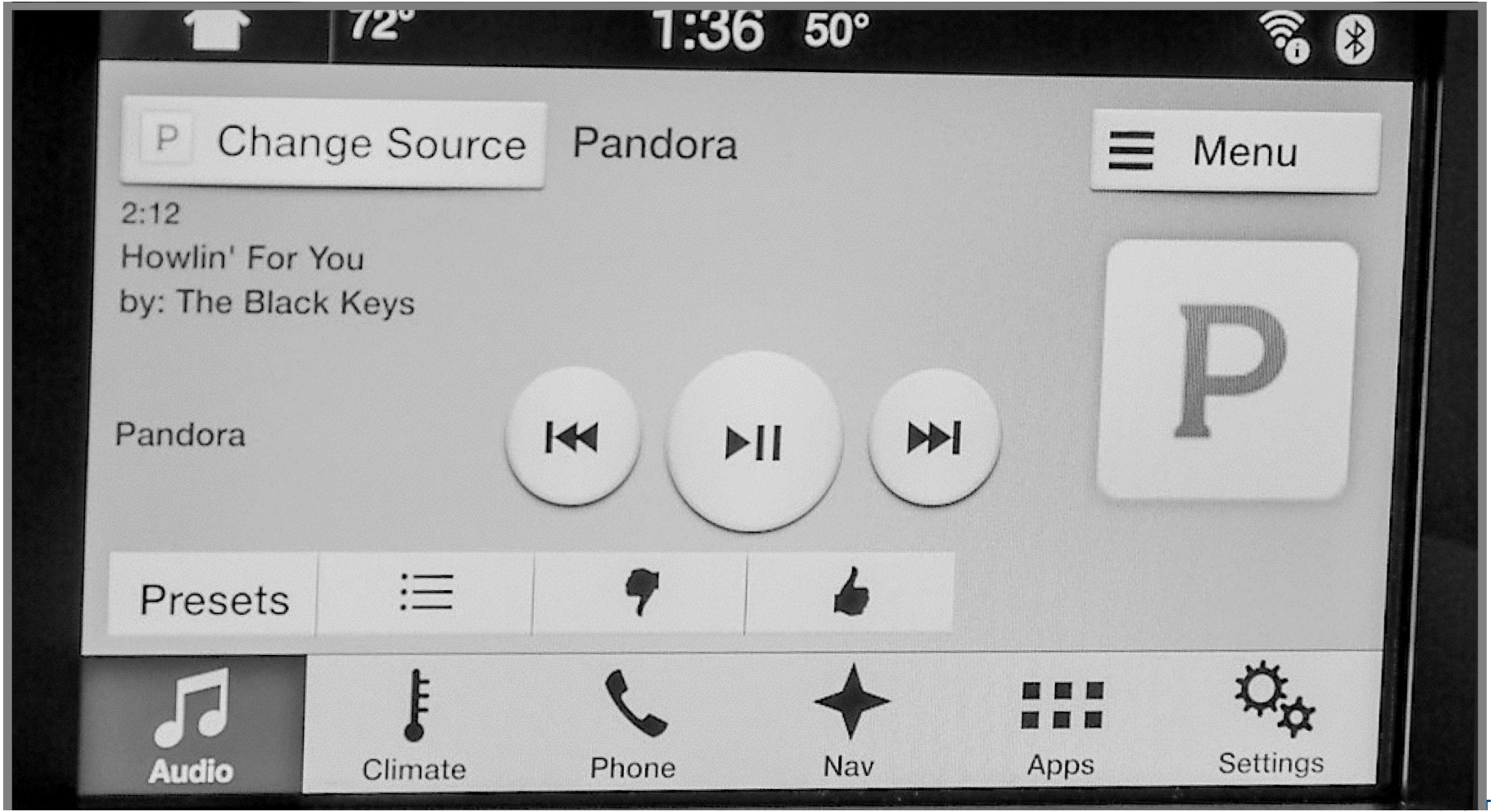
Media Player

St. Vincent

Every Tear Disappears

St. Vincent

1:36 / 4:09


72°
1:36
50°
P Change Source
Pandora
Menu
2:12
Howlin' For You
by: The Black Keys
Pandora
P
Presets
Audio
Climate
Phone
Nav
Apps
Settings

Dependencies

A quick note about dependencies

The dependencies for SDL Core vary based on the configuration. You can change SDL Core's configuration in the top level CMakeLists.txt. We have defaulted this file to a configuration which we believe is common for people who are interested in getting up and running quickly, generally on a Linux VM.

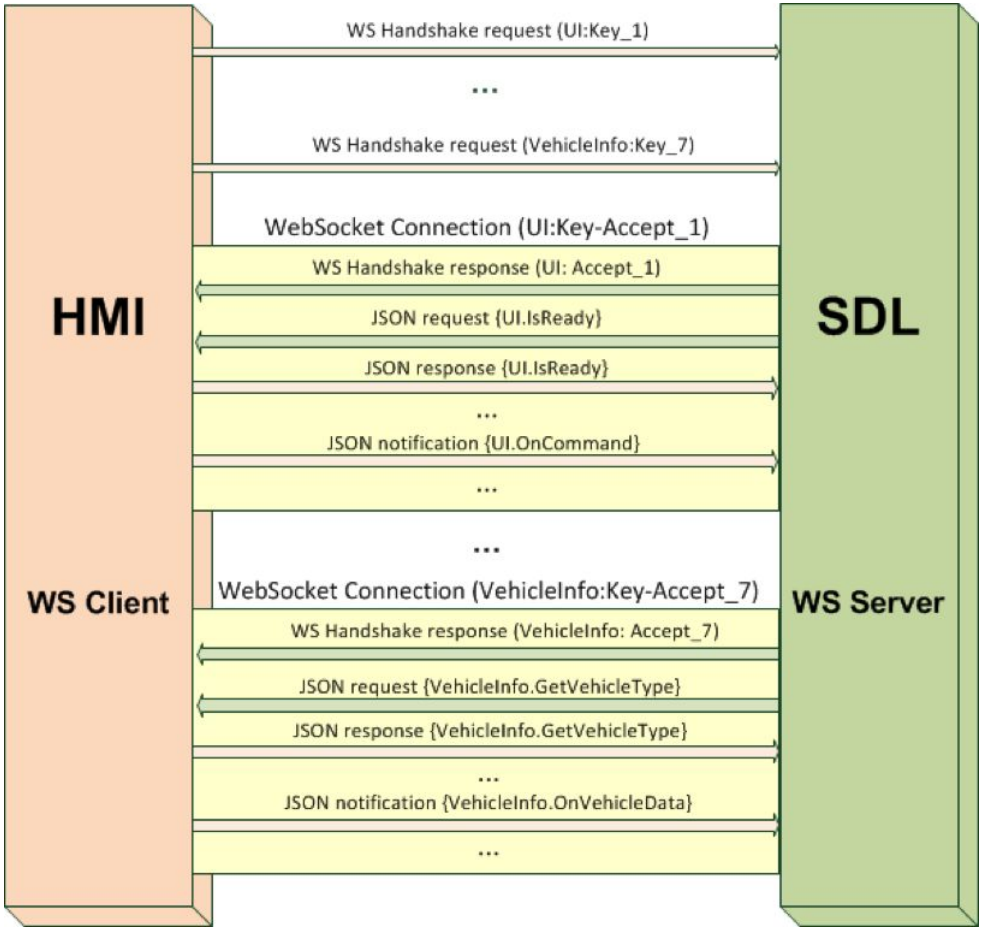
Dependencies list

Flag	Description	Dependencies
Web HMI	Use HTML5 HMI	chromium-browser
HMI2	Build with QT HMI	QT5, dbus-*dev
EXTENDED_MEDIA_MODE	Support Video and Audio Streaming	Opengl es2, gstreamer1.0*
Bluetooth	Enable bluetooth transport adapter	libbluetooth3, libbluetooth-dev, bluez-tools
Testing framework	Needed to support running unit tests	libgtest-dev
Cmake	Needed to configure SDL prior to compilation	cmake

Known Dependency Issues

- log4cxx - We know that the version of log4cxx on a linux machine can conflict with the one used, which is why it is provided in the repository. To avoid the conflict, we recommend removing liblog4cxx*.
- cmake - on some versions of linux, the included cmake package doesn't have the right version. If apt-get is your package manager, you can find the correct version using

Web based HMI using WebSockets



Competition

Apple's CarPlay

Developed originally with BMW

Widely used

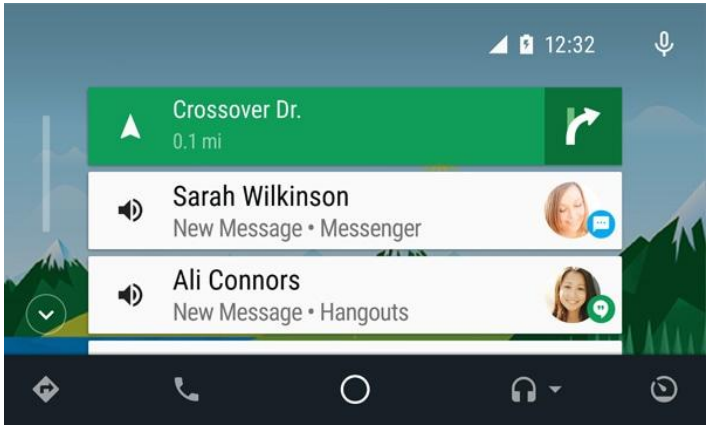
Proprietary and branded

Android Auto

Vast ecosystem

Google will sell services on top

Questions remain regard vehicle data



Competition

Baidu Carlife

About one year old

China only

- **MirroLink**
- **VNC Automotive**
  - See AGL talk, very good overview of current approaches to the smart device connectivity issue
- **Bosch my spin**

Why collaboration is key

- SDL as a more open source solution is a bit behind in terms of feature parity and adoption, collaboration will speed adoption and improve quality
- Large complex projects can only work with broad and deep collaboration
- This is non-differentiating ‘middle-ware’ or plumbing
- If users and OEMs don’t control access to their data they miss out on the opportunities that the data provides. This includes regulatory control, new business models, personalization, policy, etc.

SDL @ GitHub

smartdevicelink

Repositories | People 4

Pinned repositories

sdl_core
SmartDeviceLink In-Vehicle Software and Sample HMI
C++ ★ 118 ⑂ 136

sdl_ios
Get your app connected to the 🚙, make your users feel like a 🌟
Objective-C ★ 77 ⑂ 38

sdl_android
SmartDeviceLink mobile library for Android
Java ★ 82 ⑂ 59

generic_hmi
JavaScript ★ 2 ⑂ 3

sdl_evolution
Tracking and proposing changes to SDL's public APIs.
XSLT ★ 11 ⑂ 37

Search repositories... Type: All Language: All

sdl_shaid
Third party server to create, distribute, and manage Smart Device Link (SDL) Application IDs.
JavaScript ★ 6 Updated 5 hours ago

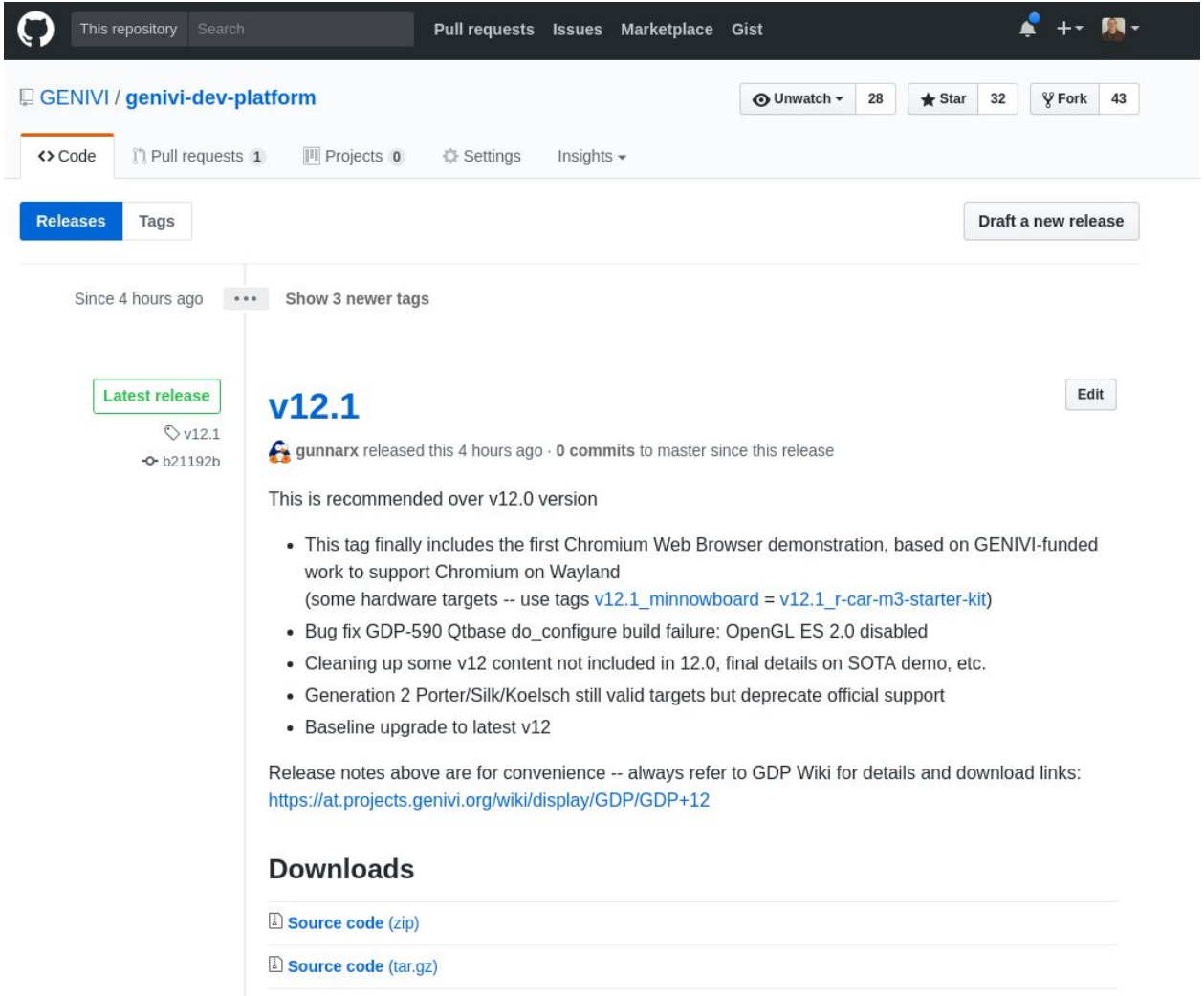
Top languages
JavaScript Objective-C Java Lua C++

Existing work

- There is an OpenEmbedded meta layer for integration called meta-sdl maintainer by Phong Tran.
  - Phong has contributed to GENIVI and the GDP
  - Code hosted at GitHub
  - Brings in changes to log4cxx, bluez-tools, sdl-core
  - Provides a systemd service file
- Adds a number of patches to sdl-core:
  https://github.com/phongt/meta-sdl/blob/release/4.1.0/recipes-automotive/sdl-core/sdl-core_4.1.0.bb

Chromium

- SDL's web HMI has a dependency on Chromium
- Chromium is coming into GENIVI's GDP and is largely complete. Igalia is responsible for this work and has done a lot of work on Chromium. [See their slides from their talk on porting Chromium to Wayland yesterday at ALS]
- Large project, huge code base
- https://github.com/OSSystems/meta-browser
  - Yocto layers for browsers
- Still relies on X11 to a large extent (again, see the Igalia slides)


This repository
Search
Pull requests
Issues
Marketplace
Gist
GENIVI / genivi-dev-platform
Unwatch 28
Star 32
Fork 43
Code
Pull requests 1
Projects 0
Settings
Insights
Releases
Tags
Draft a new release
Since 4 hours ago
Show 3 newer tags
Latest release
v12.1
b21192b
v12.1
Edit
gunnarx released this 4 hours ago · 0 commits to master since this release
This is recommended over v12.0 version
This tag finally includes the first Chromium Web Browser demonstration, based on GENIVI-funded work to support Chromium on Wayland
(some hardware targets -- use tags v12.1_minnowboard = v12.1_r-car-m3-starter-kit)
Bug fix GDP-590 Qtbase do_configure build failure: OpenGL ES 2.0 disabled
Cleaning up some v12 content not included in 12.0, final details on SOTA demo, etc.
Generation 2 Porter/Silk/Koelsch still valid targets but deprecate official support
Baseline upgrade to latest v12
Release notes above are for convenience -- always refer to GDP Wiki for details and download links:
https://at.projects.genivi.org/wiki/display/GDP/GDP+12
Downloads
Source code (zip)
Source code (tar.gz)

Summary for integrations

- Since both GENIVI and AGL are building Yocto based images an SDL recipe would have greatest code re-use
- To address the largest possible user base, the proposal would be to put the SDL recipe in meta-ivi-common which can feed into both AGL and GENIVI and even, potentially, AUTOSAR. Any Yocto or OE based source build

Thank you!

LUXOFT

LXFT
LISTED
NYSE