From bending metal to software company

Jeremiah C. Foster • 04.13.2018 • SFLC Building the Open Road

Jeremiah C. Foster

Digitalization

- Based on a transponder
- Some now use OCR technology to read your license plate

Payment Request API

W3C Candidate Recommendation 20 March 2018

- Transponder and payments become software based
- Toll itself is built into the cost of using the car
- One can imagine a shared mobility system where congestion tax is “surge” based

This will likely require significant infrastructure investment and the danger is that the shared mobility unicorns will privatize public infrastructure.

Connectivity

- Rich media, multiple networks (CAN, Flexray, Lin, EAVB, MOST)
- Bluetooth, Wifi, 4G, 5G
- Device connectivity
- Software updates via over-the-air
- 3rd party services
- Cloud computing
- Highly personalized
- Multiple high speed networks and compute nodes for AI, sensor data, media
- Highly accurate maps

Software defined vehicles

“As the automotive industry is transitioning from hardware- to software-defined vehicles, the average software and electronics content per vehicle is rapidly increasing. Software represents 10 percent of overall vehicle content today for a large, car. The average share of software is expected to grow at a compound annual rate of 11 percent, to reach 30 percent of overall vehicle content in 2030.”

-- Georg Doll, McKinsey

Electrification

Cars designed and purpose built with electric drive trains have more space and greater range. Software is also a key component of the drive chain.

Differentiation now driven by User Experience, software, and advanced hardware.

The dynamic of copyleft gives you an instant ecosystem

There already is a growing ecosystem of companies building software for rich media, connected devices. Vendor neutral, robust, with security built in. And based on Debian.

Copyleft is both an opportunity and a challenge

Now that there was an ecosystem of companies and software, compliance with the GPL was the new focus.

- Car makers are not software companies yes, they bend metal
- They don’t release early and often
- Complete and corresponding source code
- OSVs captured a lot of the process and know-how of software development

Fundamentally new view of Intellectual property. Car makers needed copyright policies, license policy, patch policy.

Process has been the response

Using the framework of an alliance, used successfully in other areas of automotive, the way to comply with copyleft requirements was process and policy creation.

- Copyright policy, Contributor License Agreements
- License policy with MPL 2.0 as the preferred license
- Supply chain management [This is where OpenChain, Community Compliance process, Common Cure, etc. will help]

Still some governance challenges;

- How to go from an alliance that shares the specification but competes on implementation to an alliance that shares the implementation but competes on specification?
- How do you graft process designed for developing safety-critical software development using standards like MISRA C onto the open source process? (See Mr. McGuire's talk)

Still some more draconian reactions such as 'blacklisting' GPLv3 likely won't keep pace with innovation in other areas.

Governance challenges

How does one keep the integrity of the ‘shell’ of protections that surround the vehicle software if the user is allowed to download arbitrary software.

How do you graft process designed for developing safety-critical software development using standards like MISRA C onto the open source process? (See Mr. McGuire’s talk)

Draconian reactions such as ‘blacklisting’ GPLv3 likely won’t keep pace with innovation in other areas.

"When designed and implemented correctly, software is often the first, and sometimes the best, hazard detection and prevention mechanism in the system."

-- NASA

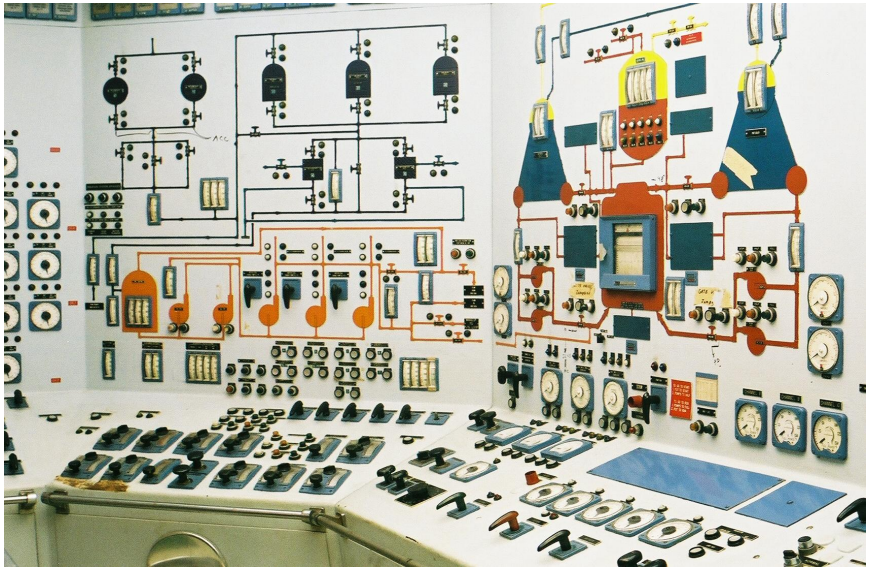


Photo credit: Wikipedia
https://commons.wikimedia.org/wiki/File:Nuclear_Ship_Savannah_-_Reactor_Control_Room_-_Center_and_Left_Panels.jpg

“Proprietary software is an unsafe building material. You can’t inspect it.”
-- Eben Moglen

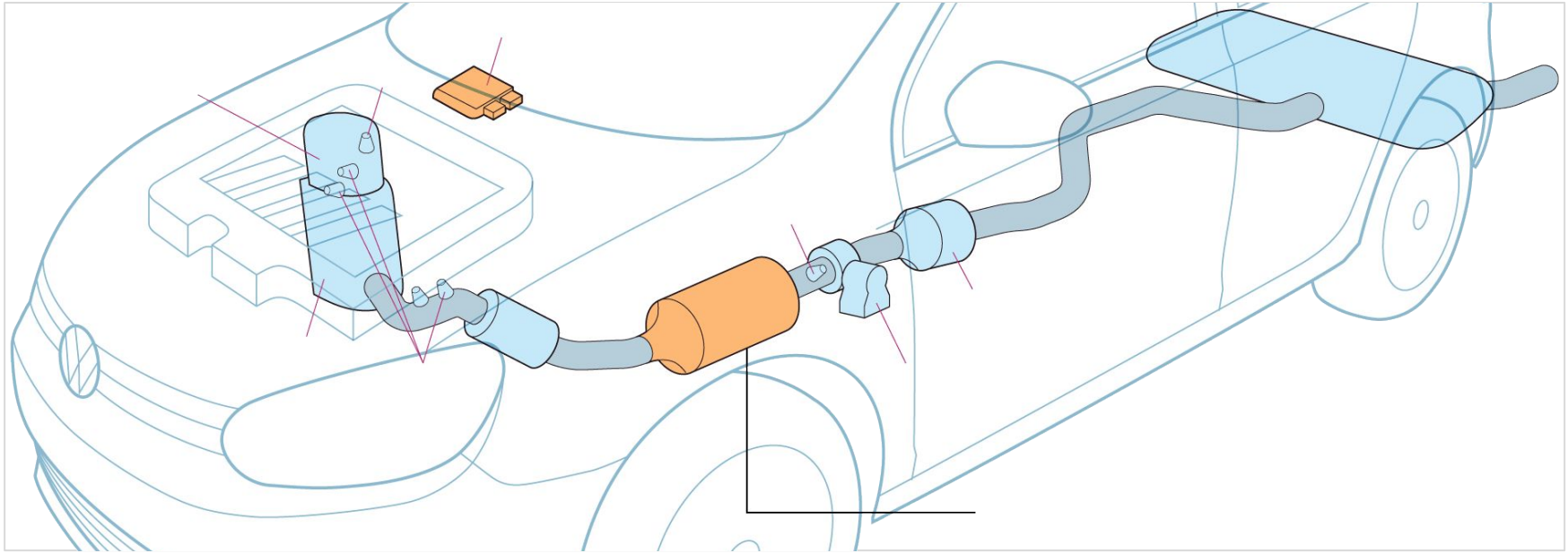


Illustration by Guilbert Gates|Source: Volkswagen, The International Council on Clean Transportation

meta-gplv2

This layer contains a set of recipes corresponding to old, obsolete versions of software that are GPLv2 licensed where the upstreams have moved to GPLv3 licenses. These were part of OE-Core until it was realised they are a ticking timebomb with regard to security updates and general maintenance. By splitting into a separate layer, it's hoped people realise these may not be the best solution to the "no GPLv3 problem" and it should also make it clear there is a different quality of service applied to these recipes. For now, they do continue to get minimal testing by the Yocto Project but this will eventually be stopped, with anyone wanting to use them taking up the maintainership.

Setup information

Git repository

git://git.yoctoproject.org/meta-gplv2 web repo

Last commit: 2 days, 9 hours ago (master branch)

Maintainer

- Ross Burton email

Dependencies

The meta-gplv2 layer depends upon:

- openembedded-core

Recipes | Updates

meta-gplv2 recipes (37)

Search recipes

Anti-anti-tivoization

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made. -- GPL

When I've asked automakers why they opt to simply ban the GPLv3 there is a common refrain, or set of refrains;

- Installation information must be made available
- Vehicle has to execute modified flash images
- Requires granting a patent license

In paraphrasing automaker's concerns in the above list, I don't feel I'm really representing some of the more thoughtful responses which is essentially; Anti-tivoization requirements create unacceptable safety risk.

Addressing Anti-Tivoization in Automotive Software

Exception to § 6 -- GPLv3 includes a provision that allows a copyright holder to use that license but to include “Additional Permissions” granting additional rights to the licensee:

> *“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law….*

You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.[14]

Addressing Anti-Tivoization in Automotive Software

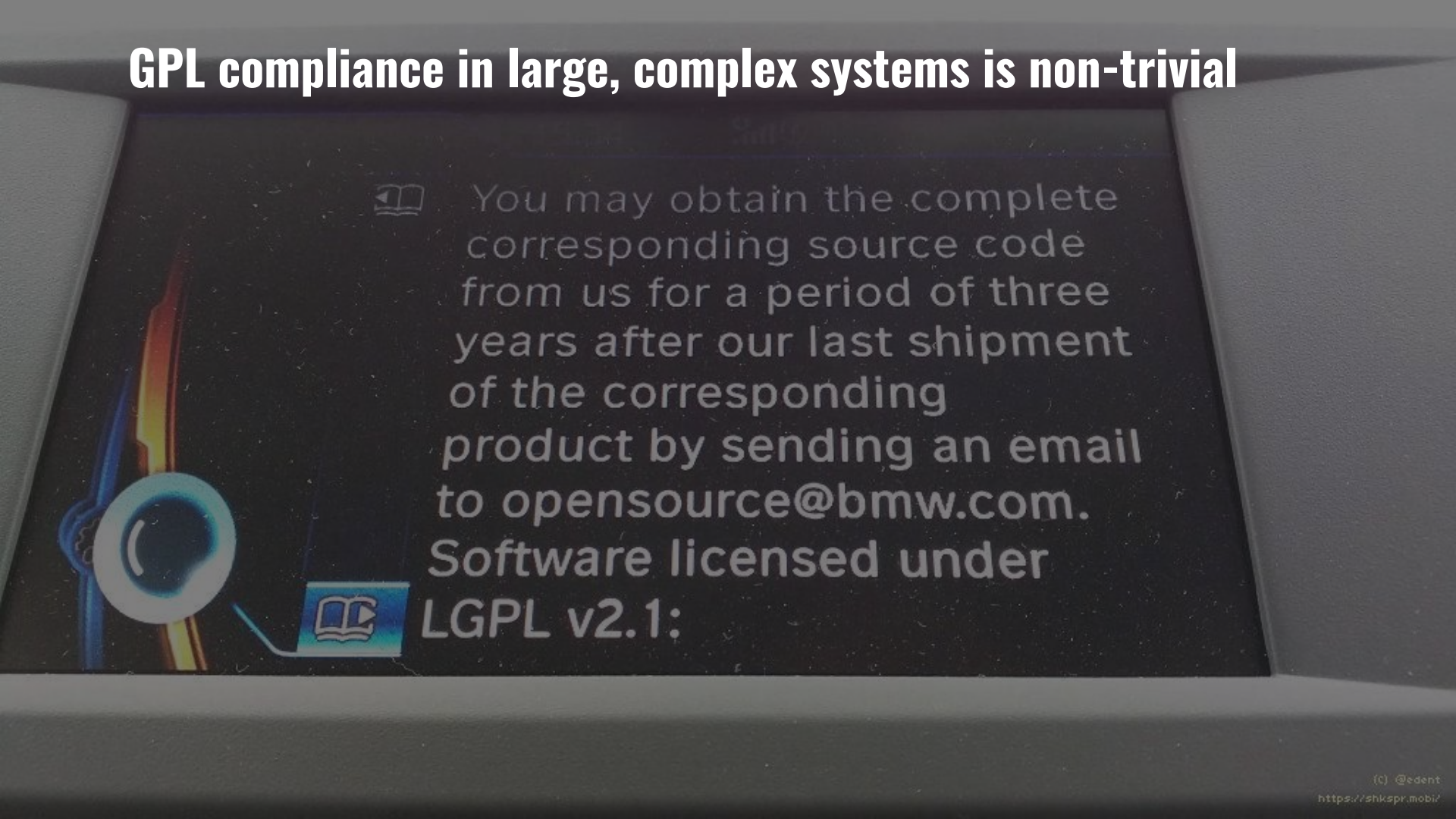
This provision of GPLv3 also allows downstream licensees to remove these additional permissions, if they so desire; When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.)[15]

This provision of GPLv3 provides a mechanism by which a copyright holder who prefers GPLv3 for their code, but is concerned about the effect of the Installation Information requirement on its downstream customers or end users, to grant an additional permission that does not obligate a licensee to follow the Installation Information requirement. At least one project has adopted such an additional permission, which could serve as a template:

> *The copyright holders grant you an additional permission under Section 7 of the GNU General Public License, version 3, exempting you from the requirement in Section 6 of the GNU General Public License, version 3, to accompany Corresponding Source with Installation Information for the Program or any work based on the Program. You are still required to comply with all other Section 6 requirements to provide Corresponding Source.*[16]

An additional permission under Section 7 of GPLv3 which exempts the licensee from the Installation Information requirement of that license, might allow for GPLv3 software to be used in automobiles while still locking down the software on the head unit to prevent the end user from changing and reinstalling the software.[17]

GPL compliance in large, complex systems is non-trivial



- Open Chain
- Common Cure
- The Principles of Community-Oriented GPL Enforcement

Do we need the AGPL3?

How do we continue to provide governance over our software when it no longer is distributed? When the car software producer is the car owner in a shared mobility system?

Who holds the liability?

Can we have full autonomous vehicles running free software?