IVI-EG4

4.Feb.2021

Table of contents

- Discussion Plan (minor update)
- Introduction
- PR software structure should be more composable and selectable
- remained Topics
  - Security Module
  - Package Management & deployment
  - API
  - HMI
  - IPC

Discussion Plan (minor update)

- Plan

#	date	Discussion Topics
1	Dec. 8, 2020	Kickoff, LifecycleManagement,
2	Jan. 7, 2021	LifecycleManagement, HelathMonitoring, + *“HAL”, Yocto Recipe Commit*
3	Jan. 21, 2021	~~HelathMonitoring, PowerManagement,~~ Commit Review, HAL(CAN)
4	Feb. 4, 2021	~~PowerManagement~~, AppFW related, Quick introduction to TestFW from Jan-Simon,
5	Feb. 18, 2021	PowerManagement, Previous topics follow up
	…	
	TBD (within trial)	AGL TestFW adoption Error Management / Logger service DEMO/Presentation for AMM

- No other update.

Introduction

Application Framework Topics related with PR

- ~~Continued use of systemd~~ *explained*
- Linux Security Module for AGL (SMACK, SELinux, AppArmor, or none). None implies leaving the LSM to the product developer. Choosing one effectively locks in the Product developer to a solution.
- Package management and deployment solution
- API mechanism. Continue to use OpenAPI or change to a different mechanism for platform services. Ideally we have an IDL that autogenerates the API code. (grpc, OpenAPI, or another)
- Long-term use of Qt
- IPC should be defined and selected.

https://wiki.automotivelinux.org/eg-app-fw/meetings

Today, we would like to have open discussion on these topics to understand problems and ideas. TOYOTA have not disclosed these implementation yet, and cannot show clear answers to all of them.

PR software structure should be more composable and selectable

- Today's point

It's difficult to adopt whole AGL to products without modification because OEM/Tier1 have their own requirements and assets that is not compatible to current AGL.
Through *Production Readiness* activity, Toyota have contributed our assets we have developed internally. However, it doesn't mean that we want to make whole AGL implemented by our assets.
While seeking "common" requirements and common implementation, *Production Readiness* structure should be more composable so that OEM/Tier1 can easily integrate some parts of AGL and their asset.

e.g.)

SMACK or SELinux or …?
Qt or Flutter or …? → **Select on uses needs**

What is needed when we make the feature implementation way **selectable**??
I hope you can make suggestion!

Security Module

- Linux Security Module for AGL (SMACK, SELinux, AppArmor, or none). None implies leaving the LSM to the product developer. Choosing one effectively locks in the Product developer to a solution.

(Toyota’s current thought)

Security is very important, and each OEM has a different concept (including requirements).
Toyota use SELinux and plan to continue to use it.

Current AGL is using smack. If OEMs adopt non-smack security modules, they cannot use some parts of AGL even if they want to.
We want *Production Readiness* to be used by various OEMs, so we don't want to restrict the security modules.

Package Management & deployment

- Package management and deployment solution

(Toyota’s current thought)

We can't give clear direction right now, because products do not treat software as a package unit.
An app store like Android is appealing to users. Since this is an applied feature, Toyota don’t focus on it from consideration in *Production Readiness* . We would like to discuss it in a few years when *Production Readiness* grows.

API

- API mechanism. Continue to use OpenAPI or change to a different mechanism for platform services. Ideally we have an IDL that autogenerates the API code. (grpc, OpenAPI, or another)

(Toyota’s current thought)

The more portable the API, the better. When we bring a service from *Production Readiness* into Toyota, we don't want to rework the service. We think it is a good idea to use IDL for the API. However, we do not have a design policy for IDL.

HMI

- Long-term use of Qt

(Toyota’s current thought)

Qt is changing its license in 2020, and has discontinued long-term support in OSS.
https://www.qt.io/blog/qt-offering-changes-2020

Toyota want to use other HMIs in *Production Readiness*. We propose Flutter. However, we do not want to restrict the HMI to a limited architecture in *Production Readiness*. If the API is implemented in IDL, the HMI will be flexible.
https://www.qt.io/jp/blog/qt-offering-changes-2020 (Japanese)

IPC

- IPC should be defined and selected

(Toyota’s current thought)

Basesystem has an IPC (as NSFrameworkUnified). We do not plan to adopt this as the IPC for all AGL services. If there is a better one, please let us know.