

Virtual Workshop from IVI-EG

15.Jun.2021

TOYOTA MOTOR CORPORATION
Woven Alpha, Inc.

■ Plan / History

#	date	Discussion Topics
1	Dec. 8, 2020	Kickoff, LifecycleManagement,
2	Jan. 7, 2021	LifecycleManagement, HelathMonitoring, + “HAL”, <i>Yocto Recipe</i>
3	Jan. 21, 2021	HelathMonitoring, PowerManagement , + α
4	Feb. 4, 2021	PowerManagement , AppFW related, Quick introduction to TestFW from Jan-Simon,
5	Apr. 1, 2021	PowerManagement(10min), Feature plan task(related with Virt-EG)(10min), + α (10min)
6	Apr. 15, 2021	Requirement Spec Status Update
7	May 13, 2021	Requirement Spec Review 1(PowerStateManagement), Flutter
8	May 27, 2021	Requirement Spec Review 2(PowerStateManagement + α), Flutter? , TBD
9	June 10, 2021	Requirement Spec Review 3(Service Launch/Termination, System Logging Support)
10	June 15, 2021	Requirement Spec Review 4(Use cases for each function), Flutter
11	June 24, 2021	Requirement Spec Review 5(Requirements for each function), ...

※10: Virtual Workshop

Time table

Today 4 items 16:00 - 18:00(JST)

- Overall introduction 10 min
- 100 min / 4 item = 25 min / item
 - (opt2 : 100 min / 6 item = 16 min / item)
- Wrap up 10 min

Tomorrow 2 items 15:00 - 16:30(JST)

- 50 min / 2 items
- 40 min additional discussion (if needed)

- The flow of 1 topic discussion (25min)
 - 5 min : Explain Abstraction • Use cases
 - 12 min: Brainstorm use cases for the item
 - Any ideas are welcome.(Duplicate OK, non-general OK)
 - 8 min : Summary
 - Compare with the use cases which I described before and select
 - Check whether each new use case is in the scope by comparing the abstraction
 - Add the new use cases based on the result (and about duplicate or non-general, they can be described as Option or Configuration item)

Goal:

- Put the use cases by participants
- Fix the use case by deciding whether it's needed or not

→Next, we'll describe the functional requirements based on the fixed use cases.

Requirement Spec activity

- Current activity

I'm currently creating a functional requirements document for services commonly required for IVI products as a Production Readiness activity for various functions. This document is being prepared based on a feature of the Basesystem software that we have previously disclosed, and is intended to describe what use cases exist for each feature as IVI and what requirements are needed to realize them, and to describe the functional requirements for the product as AGL.

Today's topic

- Today's topic

The following topics are the targets today.

- Services Launch / Termination(<https://confluence.automotivelinux.org/pages/viewpage.action?pagelId=44826721>)
- System Anomaly Detection(<https://confluence.automotivelinux.org/display/IVIPR/4.2+System+Anomaly+Detection>)
- System Logging Support(<https://confluence.automotivelinux.org/display/IVIPR/4.3+System+Logging+Support>)
- Power State Management(<https://confluence.automotivelinux.org/display/IVIPR/4.4+Power+state+management>)
- Backup data management(<https://confluence.automotivelinux.org/display/IVIPR/4.5+Backup+data+management>)
- Message notification and IPC(<https://confluence.automotivelinux.org/display/IVIPR/4.6+Message+notification+and+IPC>)

Discussion theme:

About the 6 topics above, there are some use cases as common function. So I'd like to discuss additional use cases needed for each function with you. The point of view is that the use cases should not be specific to each company's product, but should be common to all.

I'd like to explain each item and list the additional use cases required for each function.



Flutter (18:15 ~ 19:15)

1. Flutter for Embedded System [30min]
 - a. <https://elinux.org/images/f/f1/Jamboree77-Flutter-for-embedded-systems.pdf>
 - b. <https://www.youtube.com/watch?v=KN-ileJvorg>
2. Contribution Plan/Status by TOYOTA [10min]
3. Open discussion [15min]

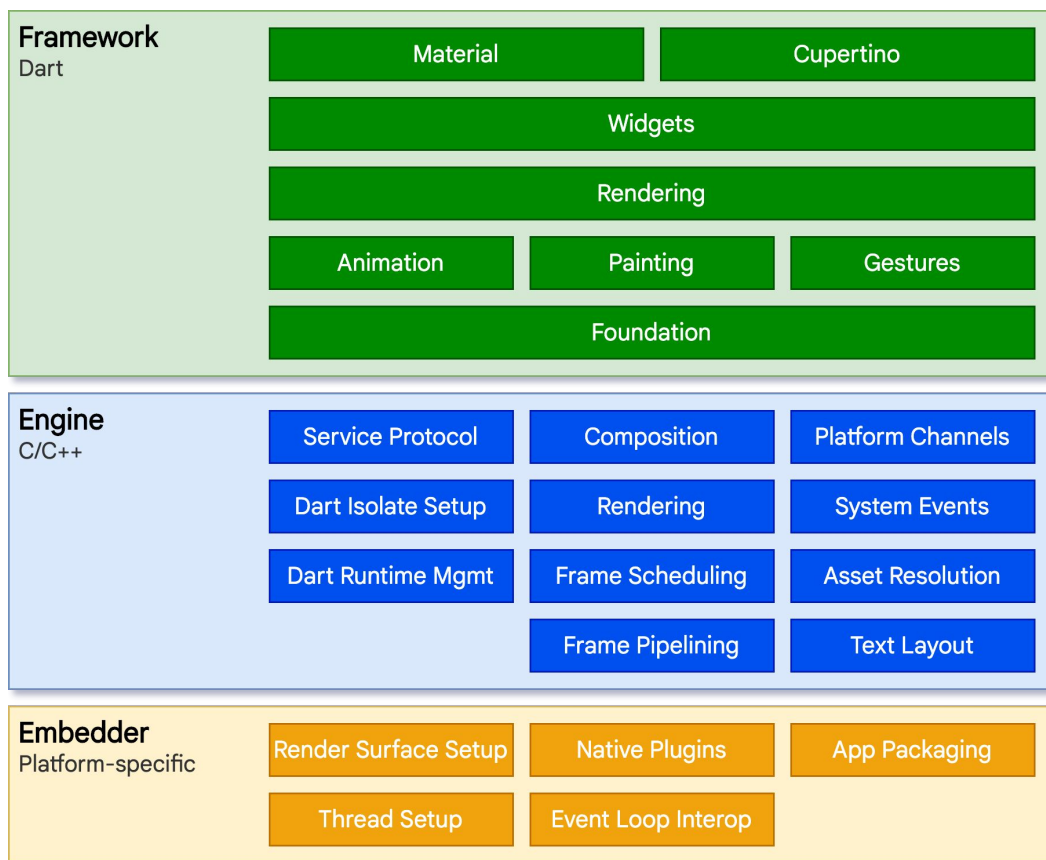
Flutter on AGL Trial

Flutter Components

- <https://flutter.dev/docs/resources/architectural-overview>

What can be contributed from TOYOTA?

- Flutter Embedder for AGL(agl-shell)
- Prototype of yocto recipes
 - Flutter build is based on GN + Ninja
 - Engine should not be modified from mainline



Flutter on AGL Trial : Original Plan as of April

- What TOYOTA have tested
 - Run against AGL Icefish(agl-shell)
 - Targeting newer agl-compositor but not updated yet
 - Sample Flutter Apps is running, some Open Source Flutter Apps can work
 - Recipes for minimum yocto image
- Goal of 2021 and (April - June)
 - April - June : Embedder is upstreamed to staging repository
 - Risk : Internal Legal Checks
 - April - June : Sample Flutter Apps can be demonstrated on AGL
 - Not targeting the integration with other AGL services
 - 2021 : Flutter can be an option of HMI FW in AGL
 - Architecture defined, integrated with AGL services
 - 2021 : With help from AGL community, other sample apps will be working

Flutter on AGL Trial : status in June

- Couldn't pass legal check by June :(ul> - hopefully that can be disclosed next month
- If someone wants to start the development, you can try related projects. They are not integrated against AGL, but they can work on embedded linux. And the core component is not so different from our project.
 - flutter-embedded-linux
 - <https://github.com/sony/flutter-embedded-linux>
 - sony's project
 - flutter_wayland
 - https://github.com/pub10cloud/flutter_waylan
 - simple wayland support for embedder

Rough comparison of Related Projects

- (This might not be correct because they are changing now)
- Key difference between (A) / (B)(C) is GTK
 - Minimise dependencies and footprint for embedded environments
- Key difference between (B) / (C) is the optimization for automotive use cases
 - (C) support plugins for 3D graphics for example
- We don't intend to compete each other. We plan to collaborate on contributing to upstream

	(A)Flutter Linux Desktop	(B) flutter-embedded-linux	(C) ivi-agl-flutter-embedder
Maintainer	flutter official (canonical)	sony	Toyota
Repository	https://github.com/flutter/engine/tree/master/shell/platform/linux	https://github.com/sony/flutter-embedded-linux	(not disclosed yet, based on https://github.com/jwinarske/flutter_wayland)
Target Environment	desktop	embedded system	automotive
Graphic shell support	GTK	wayland, DRM, (x11)	wayland (xdg, agl-shell for Jellyfish)

To support Flutter in AGL

- Just supporting “embedder” is not enough
 - Demo apps
 - Service integration
 - App development environment
 - packaging
 - security?
 - CI/CD
- Q: Flutter embedded supported xdg. Does agl-shell require a huge development?
 - Daniel : No, I guess.
 - TODO : Let's investigate in AppFW-EG and IVI-EG