2021 IVI-EG and TOYOTA Activity Plan
Content

- Objectives of this Discussion
- Activities in 2020
- Plan for 2021
- 2022 and Beyond
- Finalize Feature list
Objectives of Discussion Today

- Agree on the scope of TOYOTA’s contribution
  - list up what’s missing
- Agree on the next steps of IVI-EG
  - update feature list
Disclosed production-ready functions as planned
- basesystem
- RBA

Kooky Koi release (Feb/2021)
- Succeeded in Kicking off Production Readiness
- [AGL/meta-agl-devel.git] / meta-oem-production-readiness /

Areas of Improvement
- We should be more compliant to rules of yocto recipes
- We should avoid huge commits
2020 Activity : IVI-EG

- Successfully launched bi-weekly IVI-EG
  - Where we can focus on Production Readiness
  - Where OEMs can disclose Production Requirements

- Areas of improvement
  - Discussion Topic was too focused on basesystem and review comment for committing basesystem
    - Less attractive
  - Activities related information are not properly maintained in JIRA or Confluence
    - Meeting minutes are not sufficiently described.
      - Less appealing
2020 Activity: Product Requirements

- Production Readiness Requirement Specification v0.10 Draft
  - Draft version for BaseSystem was published

- Areas of improvement
  - Too implementation specific and lack of sufficient Requirements
  - Couldn’t held meetings among OEMs as planned
Plan for 2021 Overview

- **IVI profile operation**
- **Production Readiness profile operation**
- **Toyota PF development**

### 2020 Production Readiness profile operation
- 1-6: KK
- 7-12: LL

### 2021 Production Readiness profile operation
- 1-6: MM
- 7-12: LL

### 2022 Production Readiness profile operation
- 1-6: LL
- 7-12: LL

**Contribution**
- **AGL community operation**
  - **Trial operation**
  - **Official operation**
- **Source Code Contribution**
- **Additional Contribution (New Feature)**
  - **Flutter Embedder**
  - **Policy Management**
  - **SPEC : Production Requirements**

**Fig.) Production Readiness 2021 plan**

**Fig.) Components to Contribute in 2021**
- **HMI Framework (Flutter)***
  - Flutter engine
  - Flutter embedder
- **Application (Policy) Management**
  - Arbitration Manager
  - Property Manager
  - Application Manager
  - Window Manager
  - Audio Manager
- **Basesystem**
  - Vehicle service
  - Native service
  - System service
  - Other service
  - Peripheral service
- **HAL**
- **Hardware**

**Next target**
- **Already contributed**
Flutter on AGL Trial (1/4)

- What is Flutter?
  - Flutter is Google’s portable UI toolkit for crafting beautiful, natively compiled applications for mobile, web, and desktop from a single codebase.

- Why Flutter for IVI?
  - High performance
  - Smartphone-tier touch mechanics
  - Developer ergonomics
  - Faster iteration from customer feedback
  - BSD 3-Clause "New" or "Revised" License
  - more thoughts in our presentation
    - https://www.youtube.com/watch?v=zSbsliluixw&t=1963s
Flutter on AGL Trial (2/4)

- **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 (3/4)

- 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 (4/4)

- What’s missing?
  - (would like to Ask Community members)
  - (We need someone who can kindly lead this activity from community side)
  - Fund needed?

- Rough schedule
  - 21’ April. Define the (initial) Architecture and the scope of contribution
  - other options
    - [https://github.com/sony/flutter-embedded-linux](https://github.com/sony/flutter-embedded-linux)
    - [https://github.com/jwinarske/flutter_wayland](https://github.com/jwinarske/flutter_wayland)
    - canonical flutter embedder gtk backend (xdg-backend) <- this could be the option
  - 21’ June. Complete internal legal check and push to staging
What is Policy Management?
- Manage which apps to show on display and starting/stopping sound, judges priority and switchability based on arbitration rule.
- We internally called this “Application Management” due to some historical reason, but actually this is Policy Management. Not directly related to agl AppFW.

Why Policy Management again? What’s the relationship with RBA?
- Arbitration Controller that manage both of Window and Sound
- Purpose : showcase production use cases and the implementation
- option1 of https://wiki.automotivelinux.org/_media/agl-distro/rulebasedarbitrator_a02.pdf
  - option2 for the integration with agl-compositor
What TOYOTA plan to contribute
- Arbitration Manager
- Detail is under planning

Goal of 2021 and 2021 (April to June)
- April to June: Define what to disclose and complete internal refactoring
- April to June: Complete internal legal process
- April to September: Push source to staging repository
- April to September: Design Yocto recipes
- 2021: Merged to meta-agl-devel/meta-oem-production-readiness
Product Requirement (1/2)

- We are Code First Community! Still, Production Requirements are important for AGL, especially for OEMs.

- Why?
  - Without the definition of product requirements, we cannot understand the importance of each functions
  - We cannot talk about what's missing in AGL

- Product specific requirements vs (common) Product requirements
  - Many requirements are product specific
    - ex. show OEM logo on display,
  - There are common requirements among OEMs
    - ex. keep log data when failure occurs,

- AGL should focus on common requirements
Product Requirement (2/2)

- Goal of this activity
  - Mid Term (~2022): Product Requirements are defined as part of AGL Specifications
  - Short Term (2021): Product Requirements for basesystem and related functionalities are documented

- Looking back on the Trial Period
  - We wanted to start this activity during Trial Period, but we couldn't.

- Approach toward the goal
  - 1st step: Toyota exemplifies what kind of Requirements we need. Toyota will summarize use cases for basesystem and why these functions are required for products.
AppFW

- Closely work with AppFW-EG
- Show our requirements
### Toyota PF Development

#### IVI Profile Operation

- **Trial Operation**
  - **Source Code Contribution**
    - RBA, Health Monitoring, Power Management, etc.

#### Production Readiness Profile Operation

- **Official Operation**
  - **New Feature Contribution**
    - Flutter Embedder
    - Policy Management
  - HMI Framework
  - Policy Management Base System (update)

#### AGL Community Operation

- **Remain**
  - All of Toyota’s publishable product codes

---

**Fig.) Production Readiness Future Plan**

1. Toyota will try the following topics.
   - Basesystem source codes from staging to src from one selected function.
   - Basesystem recipes from meta-agl-devel to the suitable layer (meta-×××) in accordance with source codes.

2. Toyota will try the following topics.
   - Policy management and flutter functions source codes from staging to src from one selected function.
   - Policy management and flutter recipes from meta-agl-devel to the suitable layer (meta-×××) in accordance with source codes.

---

After considering dependencies and importance as a function, we will contribute them into the IVI-Profile starting from the necessary ones.