IVI-EG 01 (in Virtual Workshop)
Agenda

- Kickoff of IVI-EG (TOYOTA’s idea) [~25min]
  - Objective of IVI-EG
  - Scope of IVI-EG
  - Discussion Topics
  - Production Readiness Profile

- Lifecycle Management and systemd [~20min]
  - Lifecycle Management Requirement
  - Current Implementation (in Production Readiness Profile)
  - Future Plan
  - Next Step

- Open Discussion
Kickoff of IVI-EG
Objective of IVI-EG

- Motivate more OEMs / Tier1s to contribute to AGL
- Fill the Gap between current AGL and IVI products in the market
  - Manage disclosed product codes and/or requirements
  - Define common requirements, support them in AGL
Scope of IVI-EG

Focus
- Production Readiness
- Requirement Specification
- Contributions from OEMs and Tier1s (next page)
  - How they can contribute to AGL?
  - What’s the Gap between their product and AGL?

Related but could be discussed in other EGs
- Miscellaneous Platform technologies for IVI
- RBA (especially if not specific to Production Readiness)
- App FW of Production Readiness
- Test FW of Production Readiness Profile
- Reference HW for Production Readiness
How OEM/Tier1 can contribute (easily)

- In September, we asked AGL members which features are important for Production Readiness
  - Some highly rated features are not covered yet
- We hope more members to lead topics
  - In your convenient way

### Table) Wish list for IVI-PR

<table>
<thead>
<tr>
<th>component</th>
<th>others(3)</th>
<th>Score(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SoundManager</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>PolicyManager</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>ErrorManagement</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>PowerManagement</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>ResourceManagement</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>Healthmonitoring</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>WindowManager</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>VehicleBUS</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>WebAPI(Chromium)</td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>WebRuntime(Chromium)</td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>ResourceTest (CPU, Memory etc.)</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>ApplicationFW</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>AFWBinder</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>LongTermSupport</td>
<td></td>
<td>14</td>
</tr>
<tr>
<td>InputManager</td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>Performance Test</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>UserManager</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>others=&gt;</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>BT</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>NetworkServices</td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>

- **MAZDA-san**
  - Is it still possible to contribute your Spec for Logging?

- **SUZUKI-san**
  - Can I ask your comment?
Technical Discussion

Discussion Topics

- Trial Phase
  - LifecycleManagement(systemd)
  - HealthMonitoring
  - PowerManagement

- For Future Release
  - APP-FW, HMI-FW, Security, IPC, etc

Goal of the Discussion

- Reach the consensus of
  - The Necessity of the Requirement for Product and IVI Profile
  - Good implementation for that Requirement
Technical Discussions Plan

Discussion Cadence

- Start the discussion for a topic in by-weekly IVI-EG
- Q&A in JIRA for 2~4 weeks
- Conclude(or continue) the discussion in the next IVI-EG

Plan

- TOYOTA can prepare these items for upcoming sessions
- We hope other members will lead their discussion topics

<table>
<thead>
<tr>
<th>#</th>
<th>date</th>
<th>Discussion Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dec. 8, 2020</td>
<td>Kickoff, LifecycleManagement,</td>
</tr>
<tr>
<td>2</td>
<td>Jan. 7, 2021</td>
<td>LifecycleManagement, HelathMonitoring, + α</td>
</tr>
<tr>
<td>3</td>
<td>Jan. 21, 2021</td>
<td>HelathMonitoring, PowerManagement, + α</td>
</tr>
<tr>
<td>4</td>
<td>Feb. 4, 2021</td>
<td>PowerManagement, (logger service? TBD)</td>
</tr>
<tr>
<td>5</td>
<td>Feb. 18, 2021</td>
<td>TBD</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Production Readiness Profile Plan (Recap)

<table>
<thead>
<tr>
<th></th>
<th>2020</th>
<th>2021</th>
<th>2022</th>
<th>2023</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-6</td>
<td>7-12</td>
<td>1-6</td>
<td>7-12</td>
</tr>
</tbody>
</table>

**IVI profile operation**

**IVI-Production Readiness profile operation**

**Toyota PF development**

- **Trial operation**
  - RBA, Health Monitoring, Power Management, etc

- **Official operation**
  - APP Manager, Window Manager etc

- **Official operation**
  - T.B.D. (Other Service, HMI Framework)

- **AGL community operation**

- **Merge**
  - ★1st Release
  - ★2nd Release

- **Code Drop**

- **Now**

- **T.B.D.**

- **Development**
Recipe

- Manage under *meta-agl-devel*
  
  Review by AGL Community

- Each company is free to add recipe.
  
  ```
  meta-agl-devel/
  |--meta-oem-production-readiness/
  |  |--meta-agl-basesystem/
  https://gerrit.automotivelinux.org/gerrit/gitweb?p=AGL/meta-agl-devel.git;a=tree;h=refs/heads/master;hb=refs/heads/master
  ```

Source code

- Starting from *staging/* is better we think.
- Toyota put our source codes under *staging/basesystem.git*.

To Jan-Simon

1. Would it be fine to manage under *staging/*?
2. In the future, do we need to move source codes to *src/*?
3. Or does it depend on us?
Status

Created `meta-oem-production-readiness` directory under `meta-agl-devel.git`.

Working for Toyota’s Basesystem recipe layer contribution to the directory.

- The layer is `meta-agl-basesystem`.
- Some contributed patches are reviewed one by one to be merged to master.
- Source codes are in `Staging/basesystem.git` repository without review.
- Now the review step for master merge is on the way which is about 10%.
- At this moment, Basesystem doesn’t work yet but you can check build test.

[https://confluence.automotivelinux.org/display/IVIPR/Base+System+Documents](https://confluence.automotivelinux.org/display/IVIPR/Base+System+Documents)

Goal

By the end of December

- Make `meta-agl-basesystem` contribution merged to master.
- Make build passes without any errors.

By the end of March

- Update Basesystem feature.
- Confirm an application operation through Basesystem feature.
About this EG, we want to use JIRA like any other project
- To exchange information
- To share Task status and problems

Request
- When creating JIRA ticket, we want to select "Production Readiness".
- To Walt
  - So could you make “Production Readiness” Component?
Lifecycle Management and systemd
Goal of the Discussion Today

- Share the requirement of Lifecycle Management
  - Managing Services startup, shutdown and state change
- Understand the current implementation of Production Readiness Profile (and IVI Profile)
  - Coupled with Health Monitoring / Power Management / Logging
  - Current code uses proprietary service start-up/shut-down.
  - But we also plan to replace them with systemd
- Discuss technical challenges and better implementation

- Create the discussion ticket
  - Continue the discussion and Q&A in JIRA
  - Try to reach the consensus in the next IVI-EG (Jan. 7, 2021)
What is Lifecycle Management

- Managing Services startup and shutdown order

Related Product Requirements (start up)

- Service which was active when system shutdown, shall start up earlier than other service at the next system start.

User can meet the previous running state quickly

Platform Requirements

- System start other resident services according to the order set in the configuration file. And this order can be changed dynamically.
Lifecycle Management Requirement [2/2]

- Related Product Requirements (shut down)
  - Every shutdown logs shall be saved.

  ![Diagram]

  - Service A
  - Service B
  - Service C

  Logger service
  
  Log files
  
  Save all logs properly. Logger service must be the Last to stop.

- Platform Requirements
  - System shall terminate other services according to the order set in the configuration file.
Implementation in Production Readiness [1/3]

Architecture Diagram in AGL Spec ver1.0

- System Manager
  - Health Monitoring
  - Lifecycle Management
  - Power Management
### System Manager controls start / shutdown and monitoring of the resident service.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Start</td>
<td>Start resident services according to Config file.</td>
</tr>
<tr>
<td>System Shutdown</td>
<td>Terminates services according to Config file.</td>
</tr>
<tr>
<td>Malfunction Detection (HeartBeat)</td>
<td>Monitoring services with HeartBeat. On detecting Malfunctions, reset / restore services according to Config file.</td>
</tr>
<tr>
<td>Malfunction Detection (process signal)</td>
<td>Detect process crash / exit. Reset / restore services according to Config file.</td>
</tr>
<tr>
<td>Malfunction Detection (low memory)</td>
<td>Detect system memory shortage. Reset / restore services according to Config file.</td>
</tr>
<tr>
<td>LOG (abnormal state)</td>
<td>Save LOG of abnormal states</td>
</tr>
<tr>
<td>Change Model</td>
<td>Manage model specific processes and settings according to the configuration</td>
</tr>
<tr>
<td>Power State Management</td>
<td>Notify power state change to services.</td>
</tr>
<tr>
<td>RoB LOG</td>
<td>Store malfunction records as RoB log.</td>
</tr>
</tbody>
</table>
Implementation in Production Readiness [3/3]

System Manager start services in proprietary manner

Linux

- systemd
  - Other OSS services
    - setup_rethw.service
    - launch_sm.service
  - OSS Process
    - launch_sm.sh
  - LaunchConfiguration File
    - Process List, Start-up order, Option, path, etc
  - SS_SystemManager
    - AGL_process_1
    - AGL_process_2
    - AGL_process_3

SS_SystemManager launch AGL_process (fork→exec)
- LaunchConfiguration File
- Monitoring process state after init
# System Manager and Systemd

- Systemd is enough for start up / shutdown services
- System Manager is needed for other related functions
- But, Power Management, Health Monitoring(Resource Management), Logging could be decoupled from start up / shutdown.
- HeartBeat(systemd Watchdog?) and reset are in a grey area

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Replaceable with systemd</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Start</td>
<td>Start resident services according to Config file.</td>
<td>yes</td>
</tr>
<tr>
<td>System Shutdown</td>
<td>Terminates services according to Config file.</td>
<td>yes</td>
</tr>
<tr>
<td>Malfunction Detection (HeartBeat)</td>
<td>Monitoring services with HeartBeat. On detecting Malfunctions, reset / restore services according to Config file.</td>
<td>Possible. Need to manage reset policy outside systemd</td>
</tr>
<tr>
<td>Malfunction Detection (process signal)</td>
<td>Detect process crash / exit. Reset / restore services according to Config file.</td>
<td>Possible. Need to manage reset policy outside systemd</td>
</tr>
<tr>
<td>LOG (abnormal state)</td>
<td>Save LOG of abnormal states</td>
<td>No. Scope of logger.</td>
</tr>
<tr>
<td>Change Model</td>
<td>Manage model specific processes and settings according to the configuration</td>
<td>No. Scope of other service.</td>
</tr>
<tr>
<td>Power State Management</td>
<td>Notify power state change to services.</td>
<td>No. Scope of power management.</td>
</tr>
<tr>
<td>RoB LOG</td>
<td>Store malfunction records as RoB log.</td>
<td>No. Scope of logger(Rob).</td>
</tr>
</tbody>
</table>
TOYOTA’s future plan for Lifecycle Management

- Fully utilize systemd as the core component of lifecycle management
  - Stop using proprietary service launcher

- Under investigation
  - Heart Beat might be substituted with Watch Dog Timer feature in systemd.
    - Reset / Restore method should be dynamic (change depend on the error state).
    - But systemd doesn’t support that
  - Interoperability with other services
Discussion / Conclusion / Next Step

- Want to discuss more with community
  - Requirements (or ideas) from other OEMs / Tier1s.
    - Do you think these requirements / use cases are common for your product?
  - What’s the current implementation of AGL? What’s the Gap?
    - Static config (start up order) only? or Dynamically changeable?
    - Relationship with HealthMonitoring / PowerManagement

Conclusion
- TBF

Next Step
- Continue the discussion and Q&A here
  - Xx
- Reach the consensus in the next IVI-EG (TBD Jan, 2021)