IVI-EG 02

7.Jan.2021
TOYOTA MOTOR CORPORATION
Woven Alpha, Inc.

Agenda (from TOYOTA)

- What was discussed among TOYOTA / SUZUKI / MAZDA
  - How to establish common requirements
  - Difficulties to disclose requirements
- Plan / Schedule Update
  - AMM
- Lifecycle Management / Health Monitoring
- “HAL” Layer in meta-oem-production-readiness
- meta-oem-production-readiness review

Recap : 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?
  - **-> Had Offline discussion among OEMs (Dec. 17th/18th)**

- 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

OEMs discussion (Dec. 2020)

- How to distinguish common requirements and product-specific requirements
  - TOYOTA / MAZDA / SUZUKI can review whether disclosed “requirement / use case” is common to their product or not.
  - This common area could be the starting point for the Requirement of Production Readiness

- Start with High-level requirements / use cases.
  - Not implementation specification
  - Example of expected Level of Detail
    - (Use cases of Audio Service)

Last source management	When user turn on engine, IVI system start playing the last remaining source.
Start-up sound	During start-up sound, IVI system mute the last remaining source.

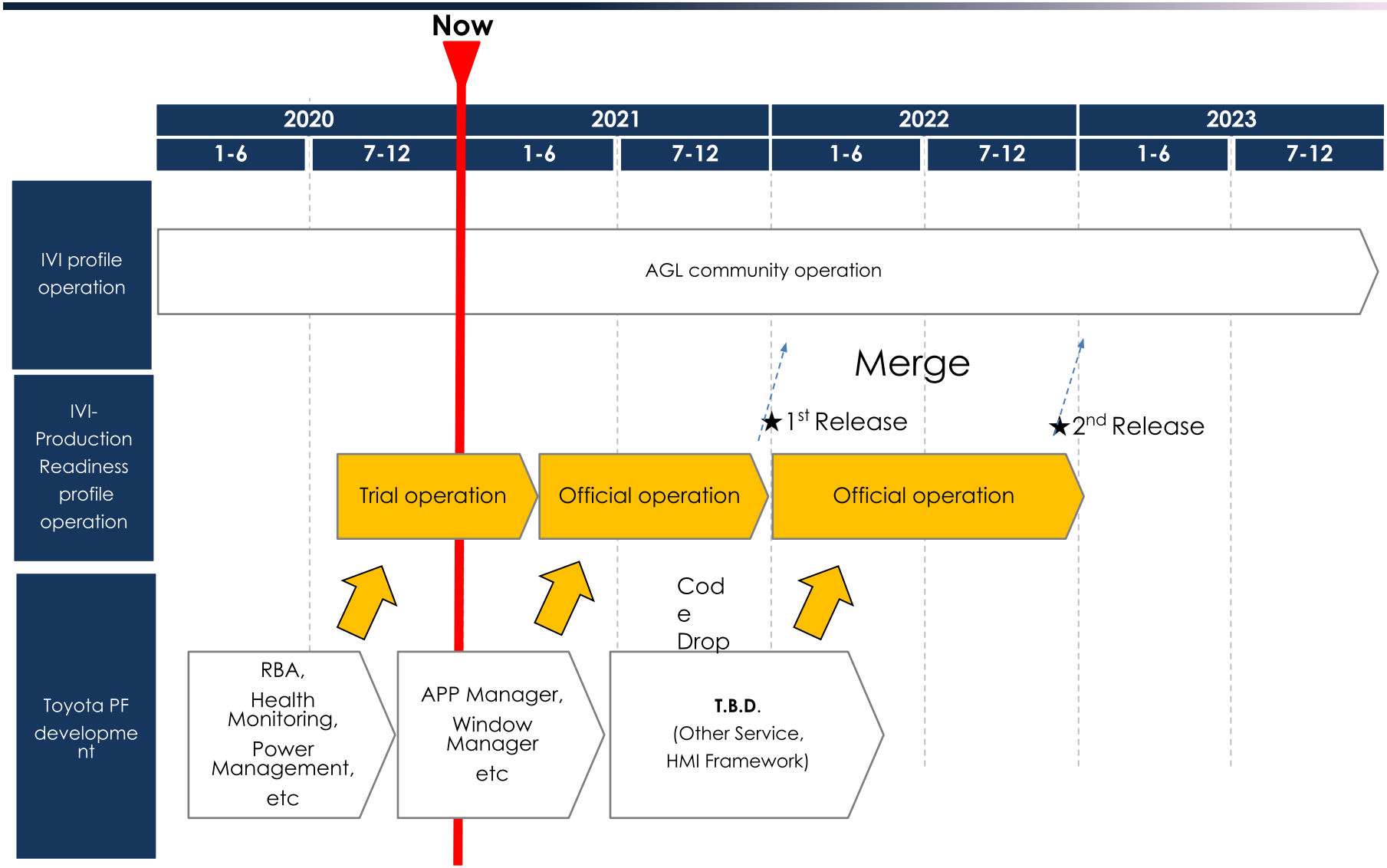
- https://confluence.automotivelinux.org/display/RS/Audio+Service

OEMs discussion (Dec. 2020)

- Difficulties when disclosing Requirements
  - Limited information available
    - Because of the development style (role of OEM/Tier1/…)
  - Sensitive information can be included
  - Company polices and Procedures
    - Procedures of disclosing IVI Requirement is not established yet

- Having regular OEM meetings during trial period
  - To speed up the process, to blur some sensitive information
  - Not intended to be closed activity. We will share the discussion result in IVI-EG

Production Readiness Profile Plan



Technical Discussion Topics (minor update)

- Discussion Topics
  - Trial Phase
    - LifecycleManagement(systemd)
    - HealthMonitoring
    - PowerManagement
    - **Logger / Error Management**
  - For Future Release
    - APP-FW, HMI-FW, Security, IPC, etc

- Goal of the Discussion
  - Phase1. Reach the consensus of
    - The Necessity of the Requirement for Product and IVI Profile
    - Good implementation for that Requirement
  - Phase2. How to merge / implement the function to IVI Profile

Technical Discussions Plan (minor update)

- Discussion Cadence
  - Start the discussion for a topic in by-weekly IVI-EG
  - Q&A in JIRA for 2~4 weeks
  - OEMs discussion about common requirements
  - ~~Conclude(or continue) the discussion in the next IVI-EG~~
- Plan

#	date	Discussion Topics
1	Dec. 8, 2020	Kickoff, LifecycleManagement,
2	Jan. 7, 2021	LifecycleManagement, HealthMonitoring, + *"HAL", Yocto Recipe*
3	Jan. 21, 2021	HealthMonitoring, PowerManagement, + α
4	Feb. 4, 2021	PowerManagement, + α
5	Feb. 18, 2021	TBD
	…	
	TBD (within trial)	- Agl TestFW adoption - Error Management / Logger service - DEMO/Presentation for AMM

DEMO in AMM (March 17 – 18)

- TOYOTA would like to Demonstrate the functionality of production readiness
- Planed Scenario and Environment (not concluded)
  - AGL app (HVAC) is monitored by HealthMonitor (system manager)
    - Cause malfunctions, detected by HealthMonitor, and restarted
  - AGL app receive the vehicle state change and update HMI
  - RBA is also integrated and demonstrated
  - meta-oem-production-readiness / basesystem.git (after merged) + AGL J.J. + additional development (for *integration)* on R-Car H3
  - … more detail will be shared
- Status
  - Developing based on contracts between TOYOTA / RENESAS and TOYOTA / DENSO
  - Health Monitoring part is working
    - Cannot disclose additionally developing code right now due to the constraint of the contract. Need some internal process ☹
  - RBA is integrated with AGL compositor
    - Upstreamed. Under review

DEMO in AMM (March 17 – 18)

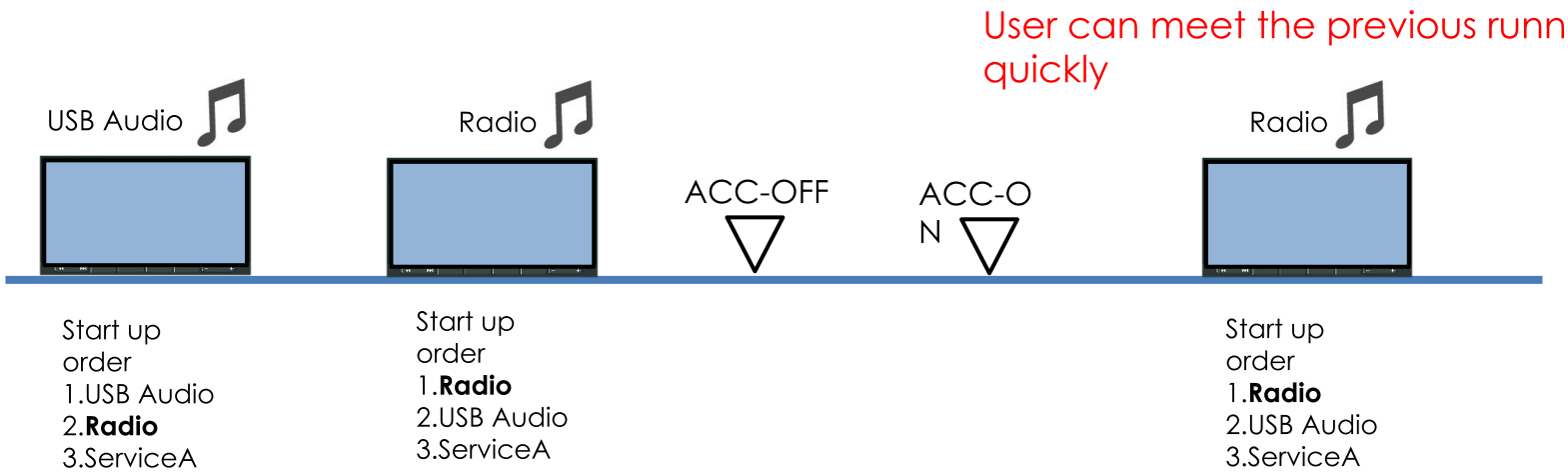
- Please let us know the registration / preparation process for DEMO in AMM
  - Ask DAN and send email.

Lifecycle Management and systemd (Answer to previous questions)

Lifecycle Management Requirement [1/2]

No update

- 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.

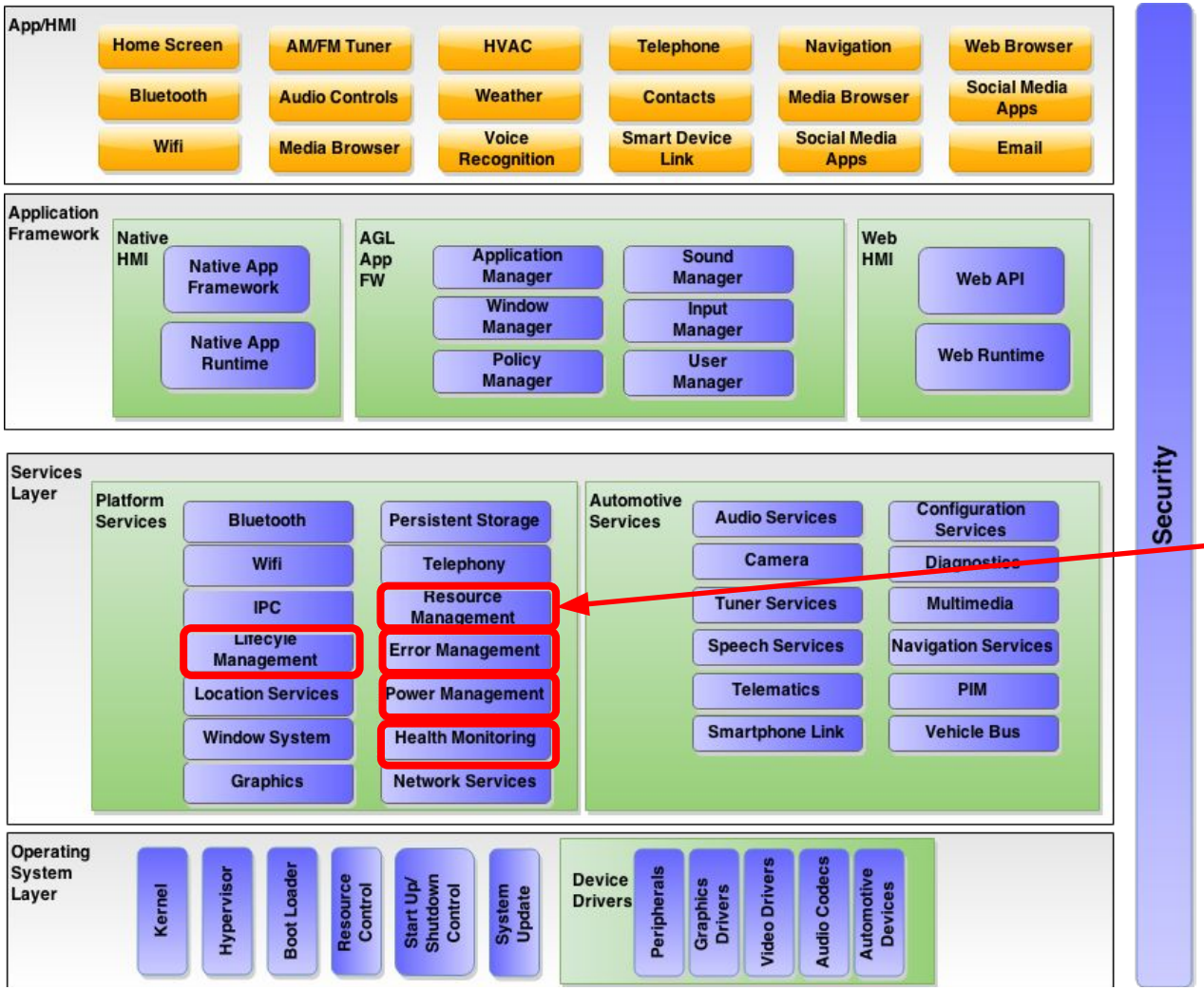


- Platform Requirements
  - System start other resident services according to the order set in the configuration file. And this order can be changed dynamically.

Implementation in Production Readiness [1/3]

No update

Architecture Diagram in AGL Spec ver1.0



System Manager

- Health Monitoring
- **Lifecycle Management**
- Power Management

Implementation in Production Readiness [2/3]

No update

- System Manager controls start / shutdown and monitoring of the resident service.

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

Lifecycle Management

Health Monitoring (Resource Manager)

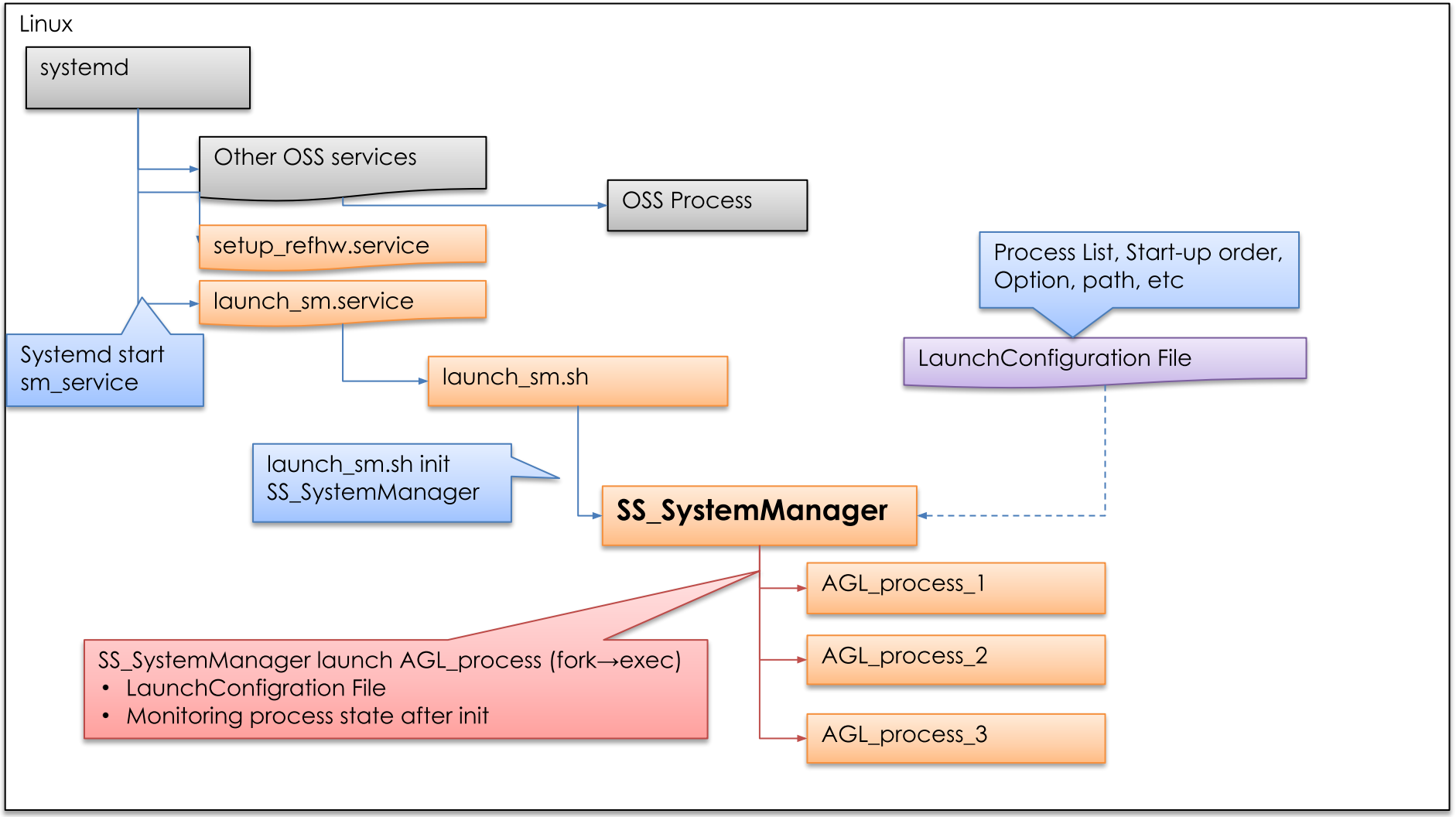
Logger Service

Power Management

Implementation in Production Readiness [3/3]

No update

- System Manager start services in proprietary manner



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

Received Some Questions here

Function	Description	Replaceable with systemd
System Start	Start resident services according to Config file.	yes
System Shutdown	Terminates services according to Config file.	yes
Malfunction Detection (HeartBeat)	Monitoring services with HeartBeat. On detecting Malfunctions, reset / restore services according to Config file.	Possible. Need to manage reset policy outside systemd
Malfunction Detection (process signal)	Detect process crash / exit. Reset / restore services according to Config file.	Possible. Need to manage reset policy outside systemd
Malfunction Detection (low memory)	Detect system memory shortage. Reset / restore services according to Config file.	No. Scope of Health Monitoring / Resource Manager.
LOG (abnormal state)	Save LOG of abnormal states	No. Scope of logger.
Change Model	Manage model specific processes and settings according to the configuration	No. Scope of other service.
Power State Management	Notify power state change to services.	No. Scope of power management.
RoB LOG	Store malfunction records as RoB log.	No. Scope of logger(Rob).

Future plan for Lifecycle Management

No update

- 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

Q&A

- Q: What is the meaning of yes / no? Why LOG is No?
- A:
  - "Yes" means the **same** functions can be implemented easily with systemd.
  - About LOG, "No" means support of other services are needed to collect information (e.g. system memory information). Want to discuss more when talking about Logger service.

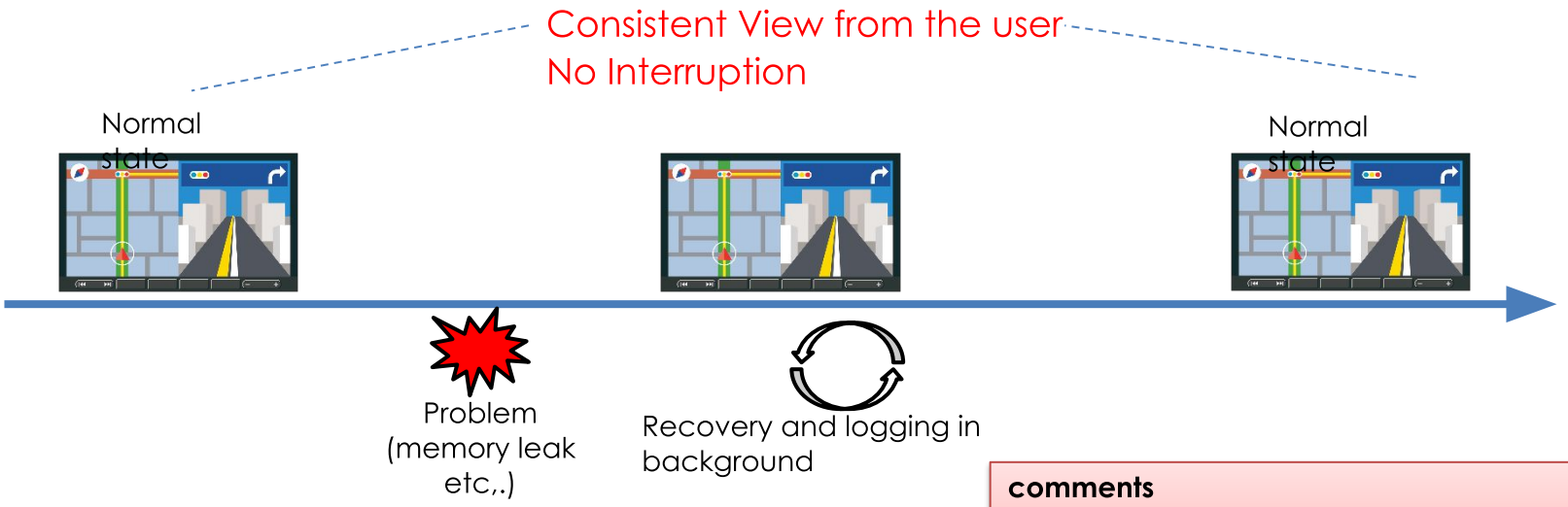
Function	Description	Replaceable with systemd
System Start	Start resident services according to Config file.	yes
System Shutdown	Terminates services according to Config file.	yes
Malfunction Detection (HeartBeat)	Monitoring services with HeartBeat. On detecting Malfunctions, reset / restore services according to Config file.	Possible. Need to manage reset policy outside systemd
Malfunction Detection (process signal)	Detect process crash / exit. Reset / restore services according to Config file.	Possible. Need to manage reset policy outside systemd
Malfunction Detection (low memory)	Detect system memory shortage. Reset / restore services according to Config file.	No. Scope of Health Monitoring / Resource Manager.
LOG (abnormal state)	Save LOG of abnormal states	(No.) Scope of logger
Change Model	Manage model specific processes and settings according to the configuration	No. Scope of other service.
Power State Management	Notify power state change to services.	No. Scope of power management.
RoB LOG	Store malfunction records as RoB log.	(No.) Scope of logger(Rob).

Health Monitoring

Health Monitoring Requirement

[Product Requirement / Use Case Example]

When the navigation system stops abnormally, the system shall restore to the original running state without the user's operation.



comments
-Session Management is required for this usecase.
-Systemd don't cover both.
-AGL don't have session manager.

[Platform Requirement]

- **Detect service abnormality** and recover to an operating state without user innervation.
- **Detect system memory leak**, and restore to normal state.
- Logging these anomalies as they occur.

Implementation in Production Readiness [1/3]

- Health Monitoring of services is part of system manager
  - System memory state is monitored and notified by Resource Manager

comments
-Before and After Systemd logging is required?
-Clarify requirement, scenario.

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

Lifecycle Management

Health Monitoring (Resource Manager)

Logger Service

Power Management

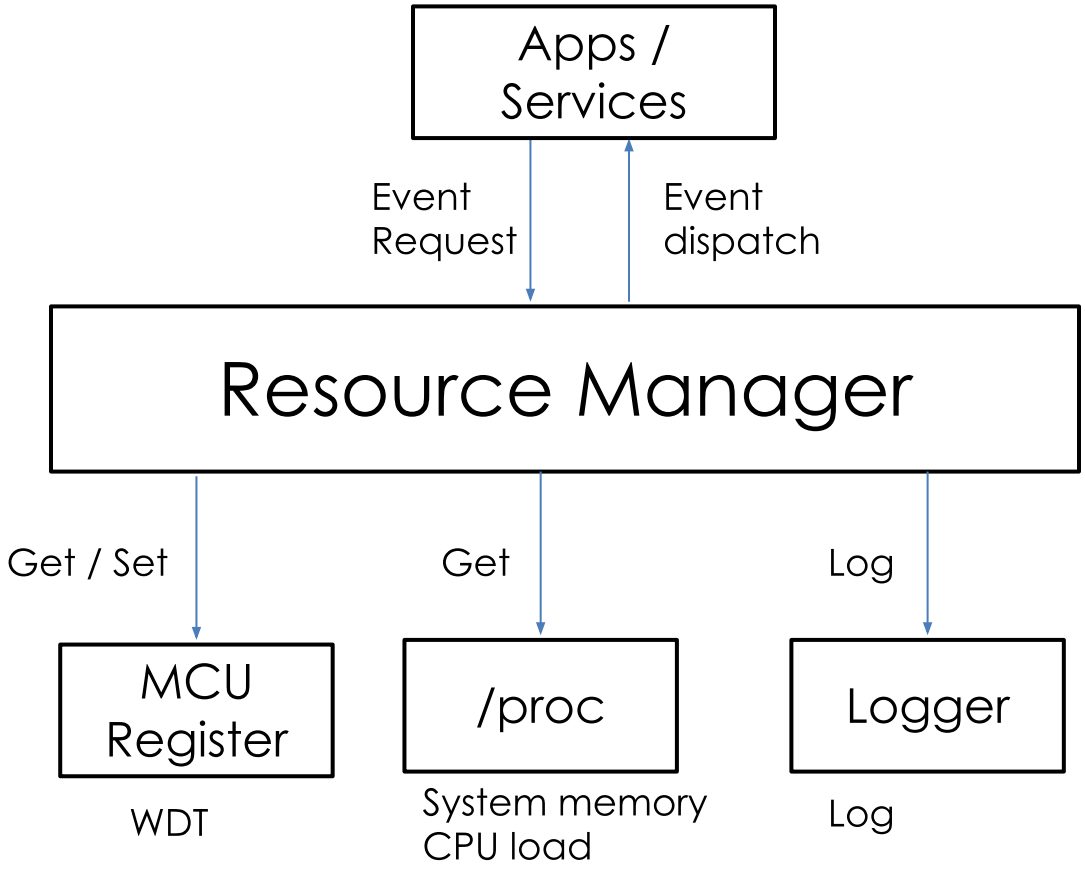
Implementation in Production Readiness [2/3]

- Resource Manager
  - Collect system resource information, notify low-memory, log output
  - Reset WDT

Function	Description
CPU Load Monitoring	Monitor CPU load and log high load information
Misc System Information collection	Collect memory usage information, write access status to NAND flash and network traffic after the power on.
System memory monitoring	Monitor system memory information. Notify low-memory event.
Watch Doc Timer (WDT) Update]	Reset WDT of microcontroller.
Provide system information (debug)	Provide misc system information to other services.
Log output of the minimum free memory information at ACC-OFF]	Output the minimum free memory information at ACC-OFF.

Implementation in Production Readiness [3/3]

- Collect system information via /proc and top
- Watch Dog reset functionality is not implemented in basesystem.git for now
  - Hardware specific implementation is needed



Future plan for HealthMonitoring / Resource Management

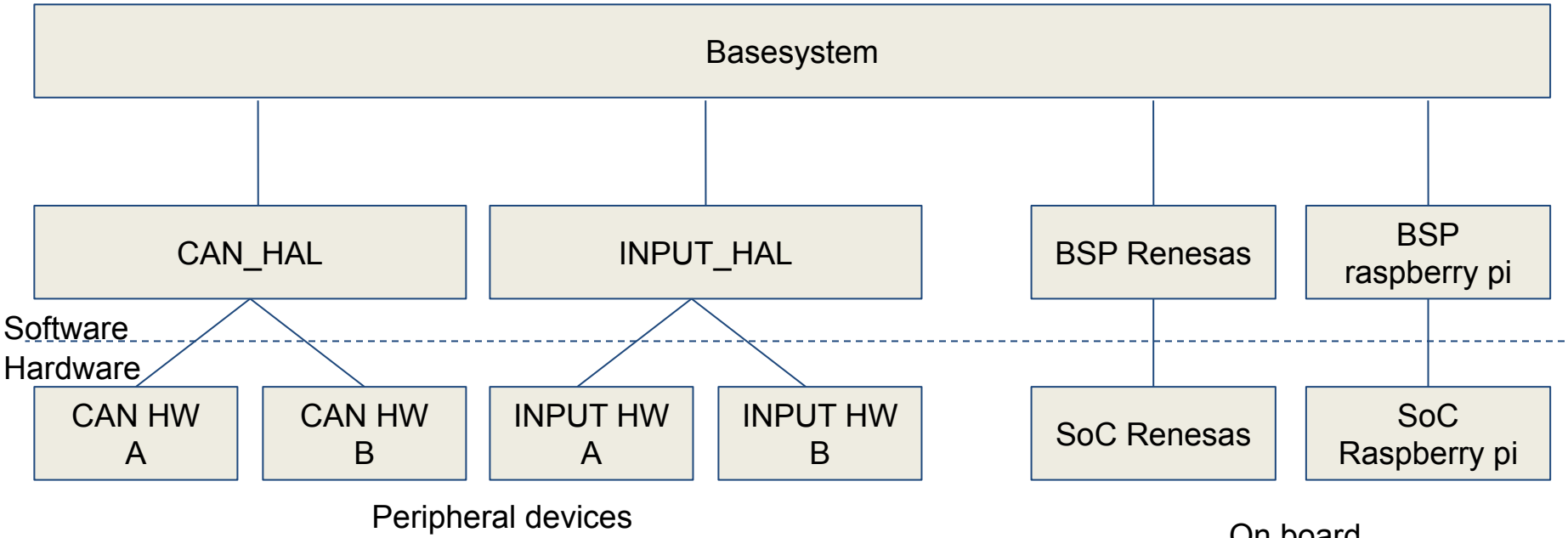
- Heart Beat might be substituted with Watch Dog Timer feature in systemd.
  - Reset / Restore method should be dynamic (change depend on the error state).
  - But systemd doesn't support that
  - Under investigation
- Various implementations are possible for Resource Management. Liable to change.

Next Step for Lifecycle / HealthMonitoring

- Summarize related Requirements and Use cases
  - Discuss with OEMs
  - Review in IVI-EG
  - … but more time needed
- Continue Q&A for the current implementation

Toyota's HAL example

1. What is the difference between BSP and HAL?
   a. BSP is for SoC.
   b. HAL is for Peripheral devices.
2. Why is HAL needed?
   a. If each hardware doesn't exist, test can be done as stub.
   b. Some vendor's devices can be supported without changing implementation of service.



HAL:
boot, can, clock, deck, input, nv, positioning, power, security, soc_temperature, usb, vehicle, video_in

Request for review of basesystem

- Have replied review comments so far.
  - Could you explain additional discussion point at this meeting? @JS

- Could you focus on meta-agl-basesystem recipes review?
  - Review any points due to meta-agl-devel.git correction.
  - Don’t review any points depending basesystem.git alone this time.
    - Toyota plan to migrate basesystem.git from staging/ to src/ step by step. We’ll refactor them and want you to review then.
  - e.g.)
    - Make same name files one.
    - Fix file which is not used.
    - Typo
    - And so on..