

IVI-EG 01 (in Virtual Workshop)

9.Dec.2020

TOYOTA MOTOR CORPORATION

- 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

- 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

■ MAZDA-san

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

■ SUZUKI-san

- Can I ask your comment?

	component	others(👤)	Score(👤)
Trial	SoundManager		27
	PolicyManger		25
	ErrorManagement		25
	PowerMangement		25
	ResouceMangement		23
	Helthmonitoring		23
	WindowManager		20
	VehicleBUS		20
	WebAPI(Chromium)		17
	WebRuntime(Chromium)		17
	Resource Test (CPU,Memory etc.)		16
	ApplicationFW		15
	AFWBinder		15
	LongTermSupport		14
	InputManager		13
	Performance Test		12
	UserManager		10
	others=>		10
BT			10
NetworkServices			10

Table) Wish list for IVI-PR

■ 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

■ Discussion Cadence

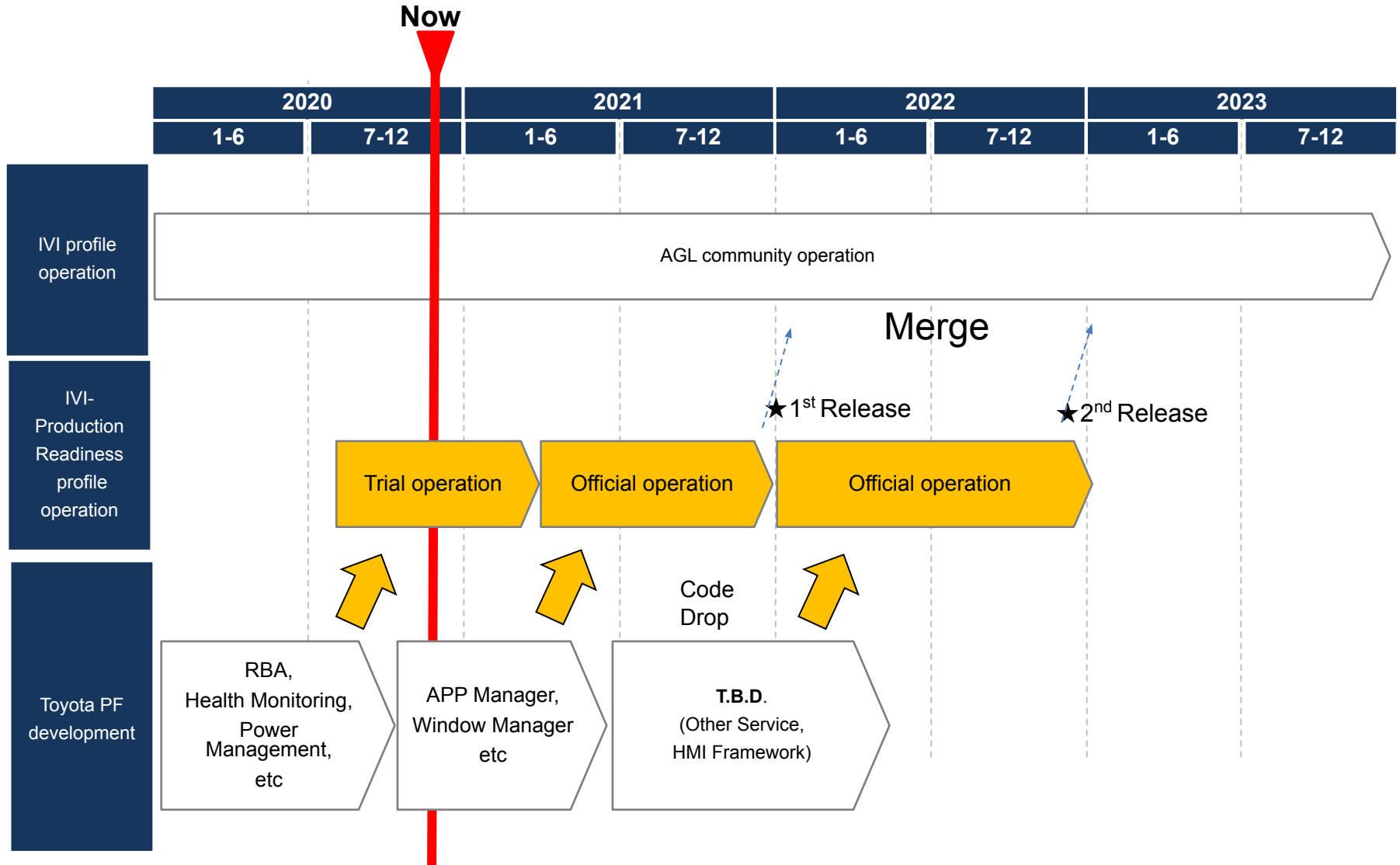
- Start the discussion for a topic in by-weekly IVI-EG
- Q&A in JIRA for 2~4 weeks
- Conclude(or continue) the discussion in the next IVI-EG

■ Plan

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

#	date	Discussion Topics
1	Dec. 8, 2020	Kickoff, LifecycleManagement,
2	Jan. 7, 2021	LifecycleManagement, HelathMonitoring, + α
3	Jan. 21, 2021	HelathMonitoring, PowerManagement, + α
4	Feb. 4, 2021	PowerManagement, (logger service? TBD)
5	Feb. 18, 2021	TBD
	...	

Production Readiness Profile Plan (Recap)



■ 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>

■ 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 Wait
 - 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

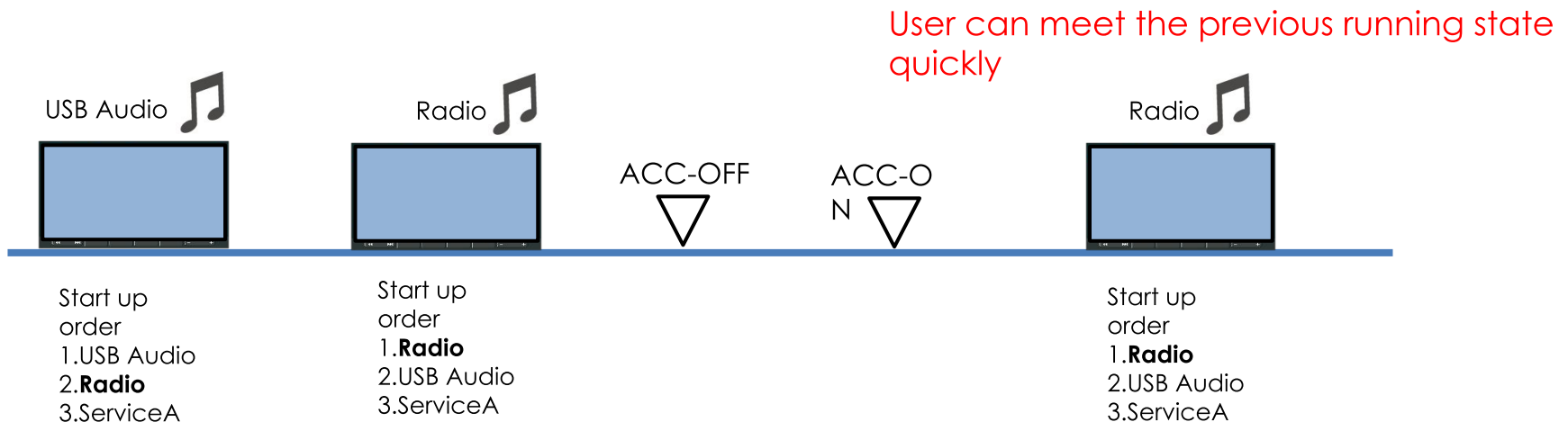
- Create the discussion ticket
 - Continue the discussion and Q&A in JIRA
 - Try to reach the consensus in the next IVI-EG (Jan. 7, 2021)

■ What is Lifecycle Management

- Managing Services startup and shutdown order

■ Related Product Requirements (start up)

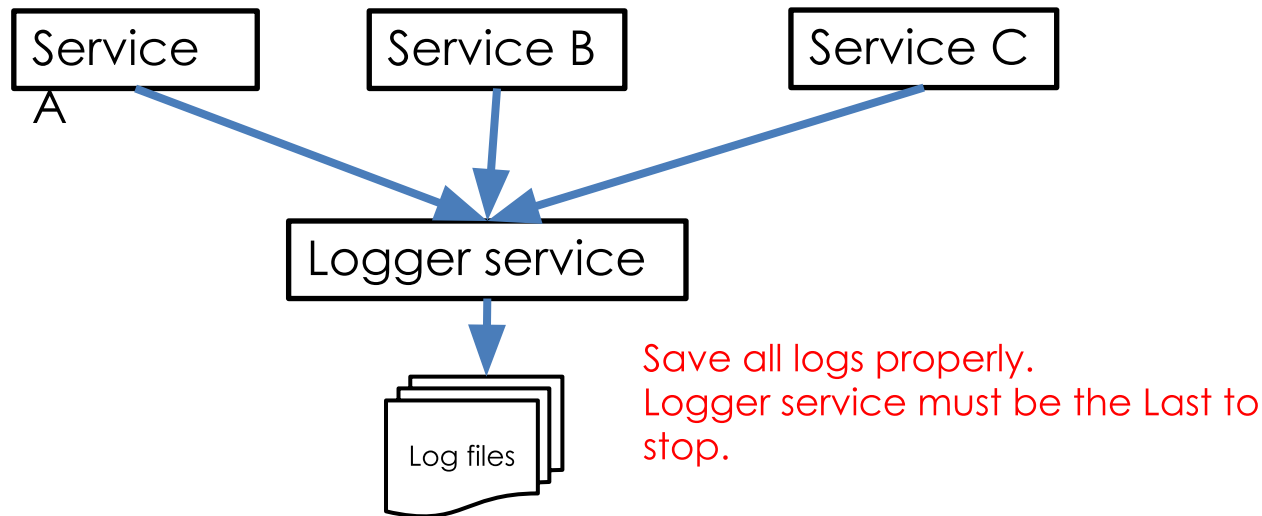
- Service which was active when system shutdown, shall start up earlier than other service at the next system start.



■ Platform Requirements

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

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

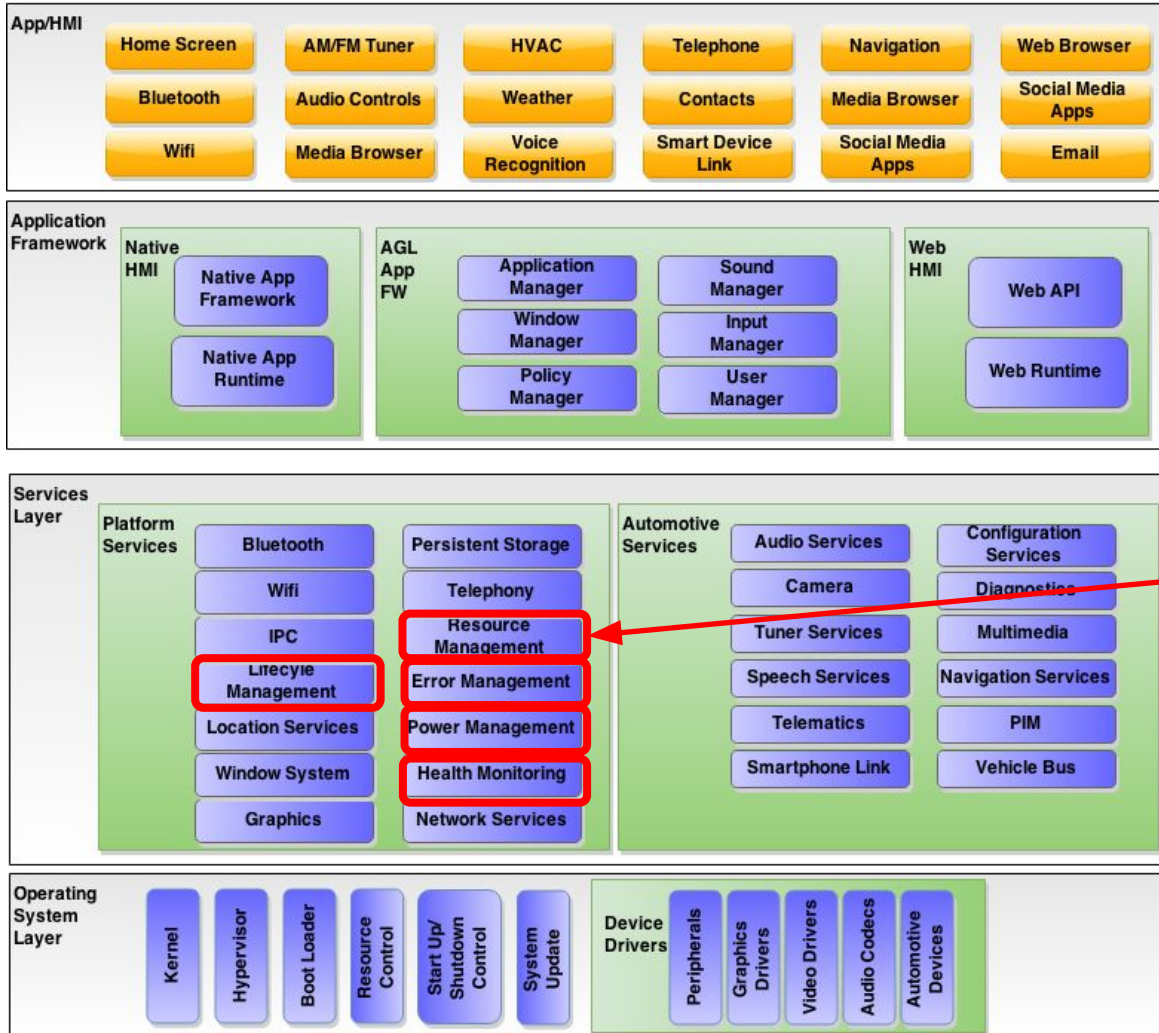


■ Platform Requirements

- 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.

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.

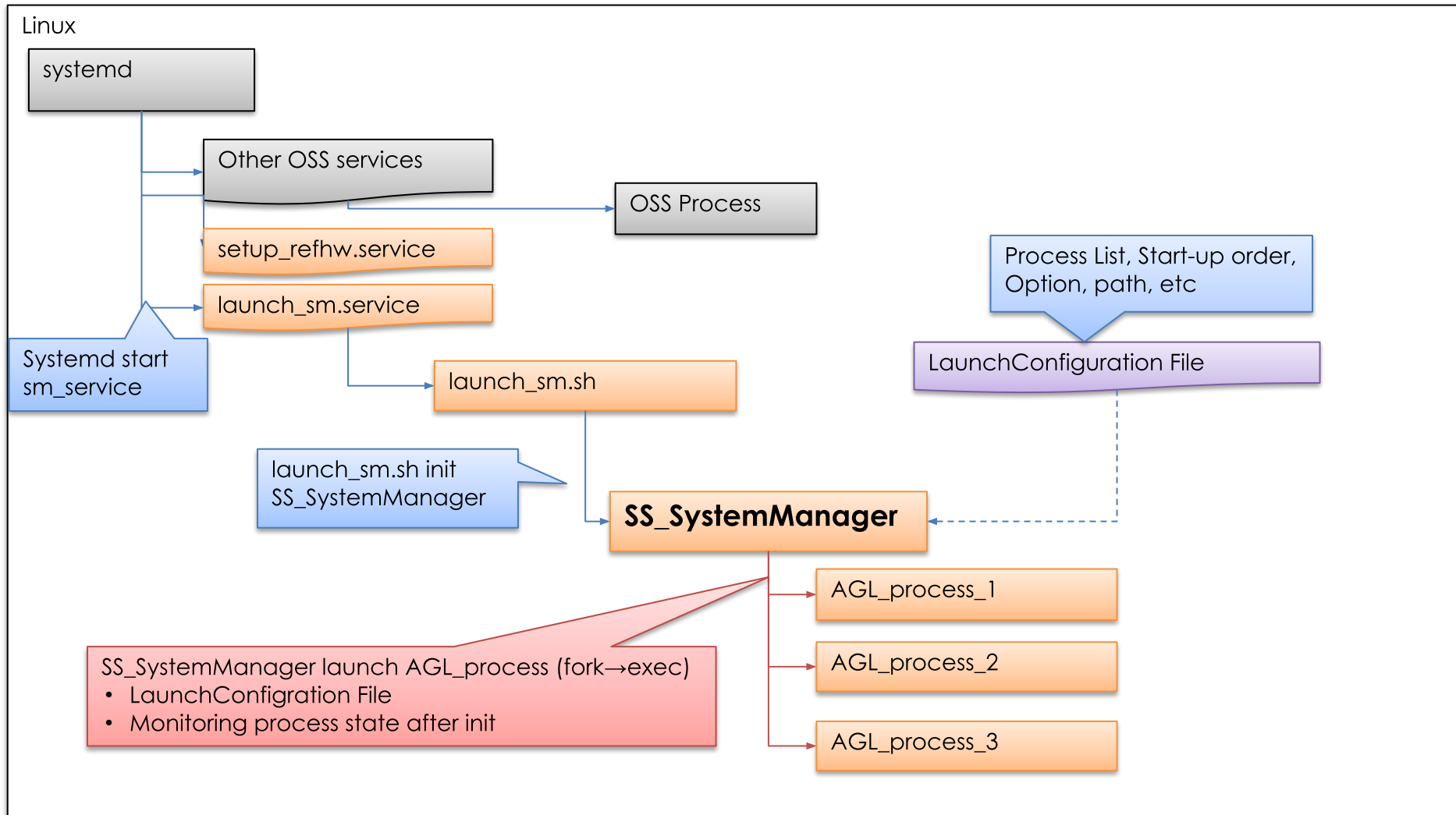
Lifecycle Management

Health Monitoring (Resource Manager)

Logger Service

Power Management

- System Manager start services in proprietary manner



- 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

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).

- 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

- 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)