

4.2 Services Launch / Termination

Abstraction

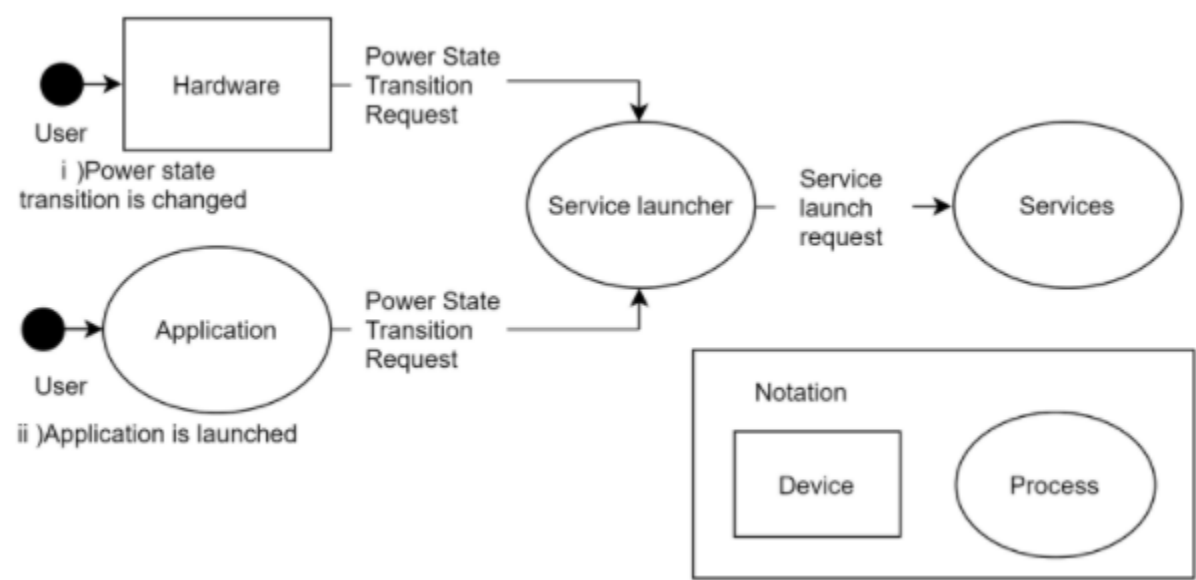
IVI realizes its functions through coordination of multiple services. Since resources such as CPU and memory are finite, it is necessary to launch and terminate specific services depending on the use case, rather than always launching and terminating all services.

There are two main triggers for launching services(Figure11): the first is when any IVI service is required to launch along with a power state transition(), and the second is when the application is launched and the service is requested to be launched along with the application launch(). When the service launcher receives information from each of these triggers, it will make a launch request to the required service.

Similarly, there are two main triggers for terminating services: the first is when any IVI service is required to terminate along with a power state transition, and the second is when the application terminates and the service is no longer needed (implementation depends on each company).

This chapter describes the use cases with launching and terminating services, the functional requirements to realize the use cases, and the current Basesystem design and implementation as a reference.

Figure 1) The trigger of launch service



Use cases

As described in the table2, use cases which need the system launch / termination function for services are described.

Table 2

#	Item	Description
UC. LT. 1	Service launch with power state transition	<div>There are multiple use cases accompanied by service launch.</div> <ul style="list-style-type: none">• The driver presses the button for ACC-ON, the IVI starts and the home screen is launched (the services required to launch the home screen are launched when the IVI starts).• IVI automatically updates software through OTA function while the driver is away from the vehicle.• IVI backs up data while the driver is away from the vehicle, etc.
UC. LT. 2	Service launch when using application	<div>The driver uses the navigation application and sets the destination. The user starts the application, enters the destination, and the time required and route are displayed.</div>
UC. LT. 3	The launch order of previously used service	<div>When the driver starts up the system, the driver uses the services that were running when the system was last stopped before any other services.</div>

UC. LT. 4	Service termination after finishing application	The driver makes a hands-free phone call while driving the car. After the user finishes the call, the service that is no longer needed is also terminated by being requested.
UC. LT. 5	Service termination with power state transition	<p>There are multiple use cases accompanied by service termination.</p> <ul style="list-style-type: none"> • The driver arrives at the destination and stops the car. The IVI system stops and the activated services are terminated. • After arrival at the destination, when the driver wants to continue a handsfree call after ACC-OFF, the user continues it although the display is off. • After the driver arrives at the destination, the driver presses the button for ACC-OFF and gets out of the car. When the driver gets out of the car, if there is danger behind the car, an alarm sounds and the driver is notified, etc.
UC. LT. 6	The order of service termination	The driver arrives at the destination and stops the car. When the driver presses the button for ACC-OFF, the system will exit with a log of the user's system usage at the end.

Functional Requirements

This table3 includes the functional requirements of the Services Launch / Termination module.

Table 3

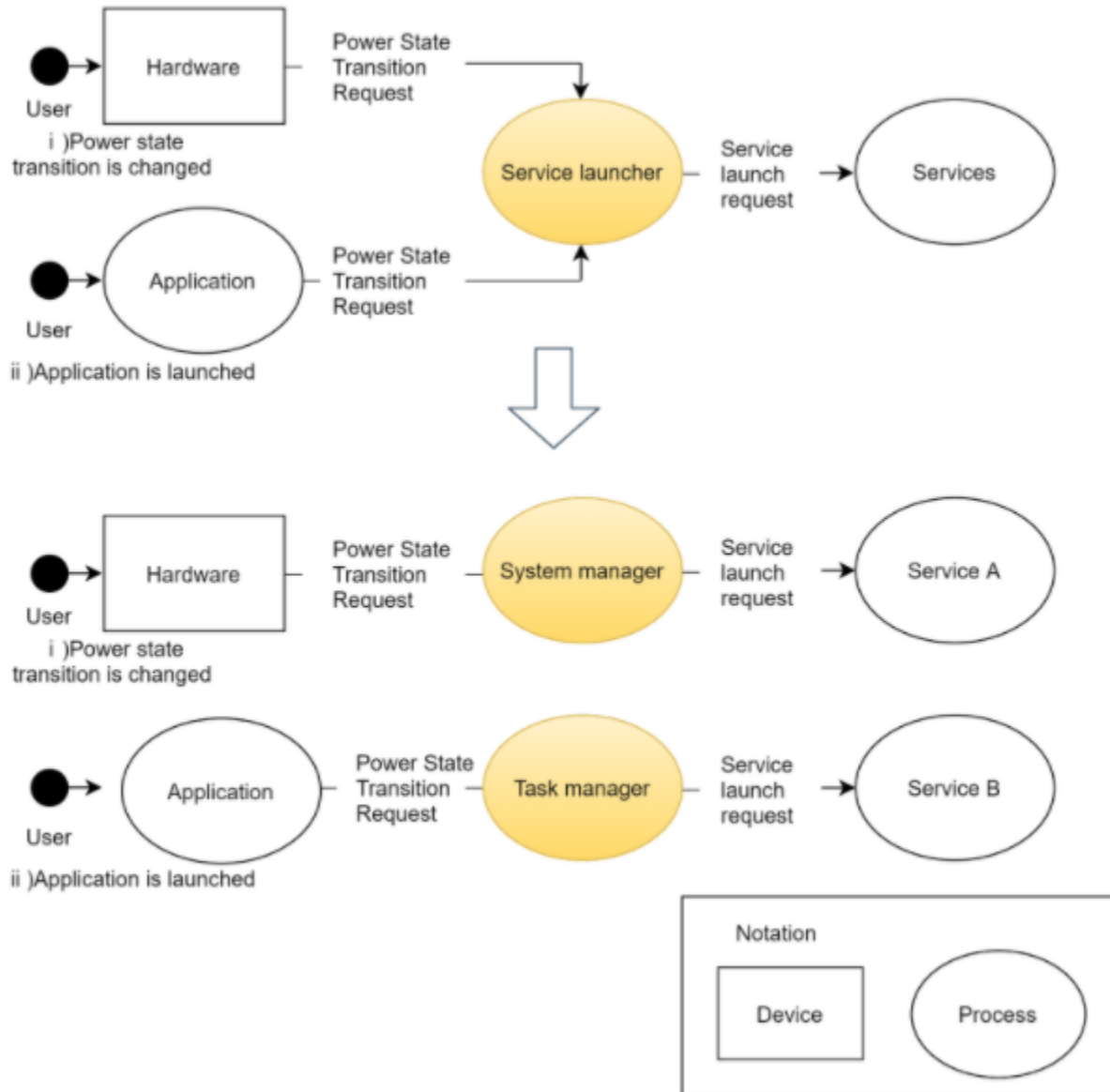
#	Item	Related use case	Description
RQ. LT.1	Service launch	UC.LT.1	Service launcher shall send launch requests to the required services according to the power state transition requests.
RQ. LT.2		UC.LT.1	Service launcher shall allow the configuration of which IVI services to launch.
RQ. LT.3		UC.LT.1	The order of the necessary services shall be configurable.
RQ. LT.4		UC.LT.2	Service launcher shall send launch request to the necessary services along with application launch.
RQ. LT.5		UC.LT.3	Services that were running when the system was last shut down shall be launched in priority.
RQ. LT.6	Service termination	UC.LT.4	Service terminator shall send a termination request to the service that provided functionality to the application when the application is no longer in use and there is a request to terminate the service.
RQ. LT.7		UC.LT.5	Service terminator shall send a termination request to the necessary service according to a power state transition request.
RQ. LT.8		UC.LT.6	Service terminator shall be able to set the order in which services are terminated during system shutdown.

Service Launch / Termination in Basesystem

Reference implementation in Basesystem

In the implementation of Basesystem, the function modules for launching and terminating resident services and non-resident services are separated. In Basesystem implementation, System manager launches and terminates resident services and Task manager launches and terminates non-resident services. As shown in the following figure 14, the Service launcher corresponds to (i) Systemmanager and (ii) Task manager for each trigger in the Basesystem module.

Figure 4



System manager

When IVI starts, System manager launches services according to the configured order. It needs to be launched first in the IVI system and it is responsible for launching and terminating other services. When IVI shuts down, it terminates the services in the order according to the settings in the same way as at startup.

Reference code : https://gerrit.automotivelinux.org/gerrit/gitweb?p=staging/basesystem.git;a=tree;f=service/system/system_manager;hb=refs/heads/master

Task manager

Task manager launches and terminates the non-resident service. This function is provided as a process and launched by System manager. Task manager provides the service launch interface to HMI application and launches the service if requested. Task manager monitors the running status of the launched service, and if it detects a hang-up, it forcibly terminates the service.

Reference code : https://gerrit.automotivelinux.org/gerrit/gitweb?p=staging/basesystem.git;a=tree;f=service/system/task_manager;h=0d795ded146125a0f28b363e8e0821210d1be209;hb=refs/heads/master