BaseSystem Features

2019.10.17

TOYOTA MOTOR CORPRATION

Table contents

1 Overview

1.1 About This Document

The document is the summary of features that is disclosed Toyota's product-level codes (**Base System**).

AGL PF developers should refer to API specifications and source codes in addition to this document.

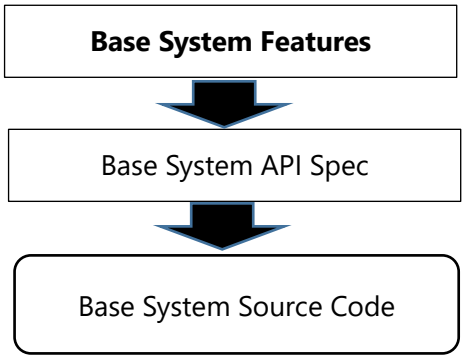


Fig. About this document

1.2 Software PF comparison

To understand the Toyota software PF, let's see the differences between AGL software PF and Toyota software PF.

Each of software PF has the following software layers,

① Application Framework
② Application Service
③ Vehicle Middleware
④ OSS(Linux) Middleware
⑤ Linux Kernel

The concept of the software layer is almost the same. However, the software components included are different.
In particular, necessary features of automotive software as generic them is shortage.
In particular, necessary features aren’t enough as generic automotive software.
The following figure shows the software structure of each PF.

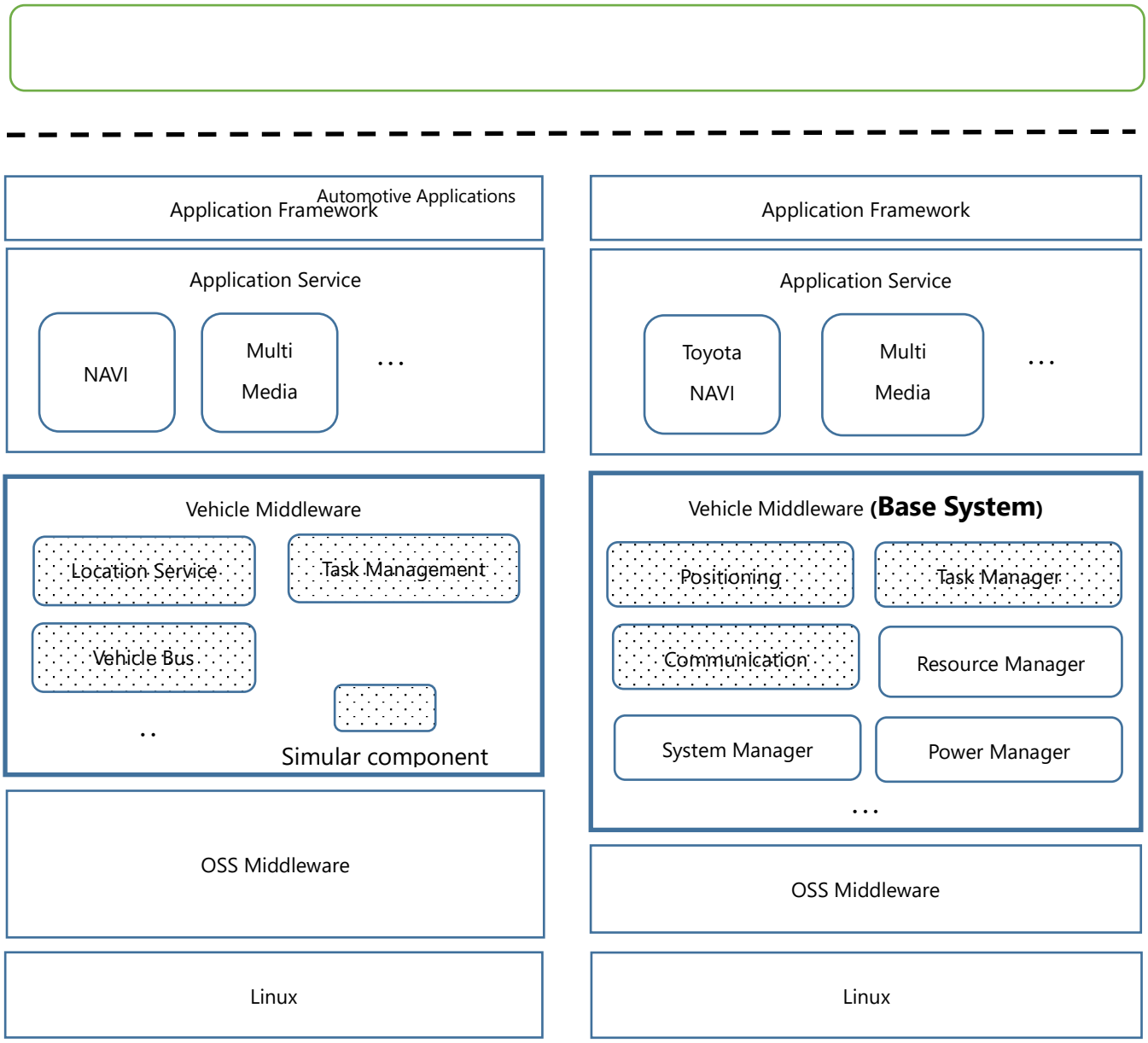


Fig. Comparison software PF

1.3 About Base System

Base System is the target this time, is a subsystem that is included in the Vehicle Middleware layer, and is software that realizes features unique to vehicles.

Base System has five software packages.

- System Service
- Native Service Framework
- Peripheral Service
- vehicle Service
- Other Service

Each package has several units, and each unit has several features.

The relation between package, unit and feature is as follows.

Sub-System(Base System) > Packages > Units > Features

The details of the units of Base System explain in Chapter 2 onwards.

Chapter 3 explains typical use cases of Base System.

2 Base System

The packages included in Base System show below.

- System Service ：
  - Provides and executes processing related to the entire system.
- Native Service Framework：
  - Provides a means of communication between units and data interchange.
- Peripheral Service ：
  - Provides control features for peripheral devices.
- Vehicle Service ：
  - Provides acquisition of vehicle network information and control it.
- Other Service ：
  - Provides common features which are not provided by other Base System units.

2.1 System Service

■Overview

System service is a service package that provides the following features.

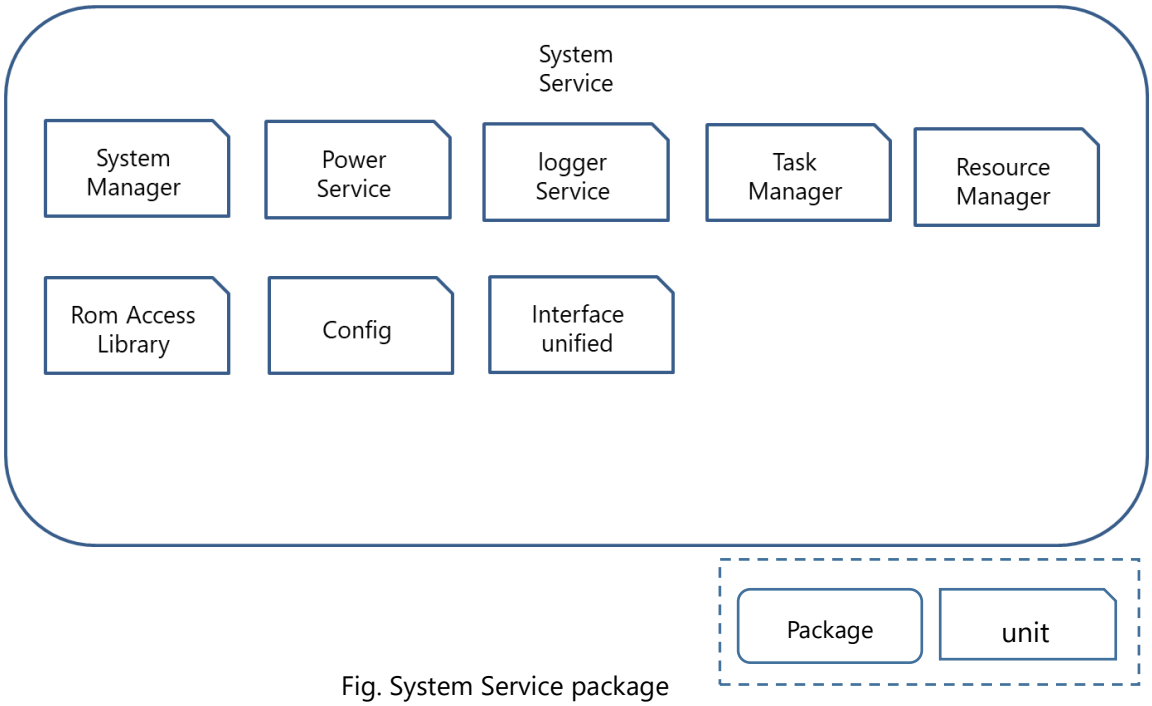


Fig. System Service package

2.1.1 System Manager

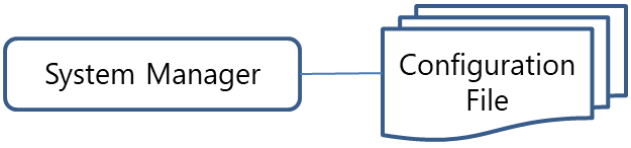
■Feature Overview

This service provides the features to process the system start and shutdown.

This provides the following features.

1) System start-up

When System Manager starts, it starts the other resident services in line with the order in Configuration File. The services can group and System Manager starts the services per the group.



2) System shut-down

This terminates the other services in line with the order in Configuration file. The system shutdown processes based on Power Supply Sub-System requests, the service requests and malfunction detections etc.

3) Malfunction detections

When System Manager detects every kind of the malfunctions, it executes the various failure processing. The contents of the failure processing statically prescribe in Configuration file in advance. The prescribed ones are the system reset, the restart (event occurrence processes) and the occurred event log storing.

3)-1. Heart Beat

This confirms the behaviors by the regular communication (Heart Beat request) to the controlled services. When it has no responses from the services, System Manager recognizes the malfunctions and restarts and terminates the services or resets the systems in line with the set-up in Configuration file.

3)-2. Services abnormally shutdown

This monitors the process status after start. When it detects the abnormal shutdowns, it restarts and terminates the services or resets the systems in line with the set-up in Configuration file.

3)-3. System memory shortage

This monitors the system memory in cooperation with Resource Manager. When the value gets lower than the certain level for the specified time, it recognizes the system memory shortage and resets the system.

* The Resource Manager is a feature within the system service. The Resource Manager describes later.

4) Abnormal log output

This provides the feature to output the abnormal log for the services. It stores the targeted log in USB storage device and SD card etc. due to the requests from the applications.

■Unit relation diagram

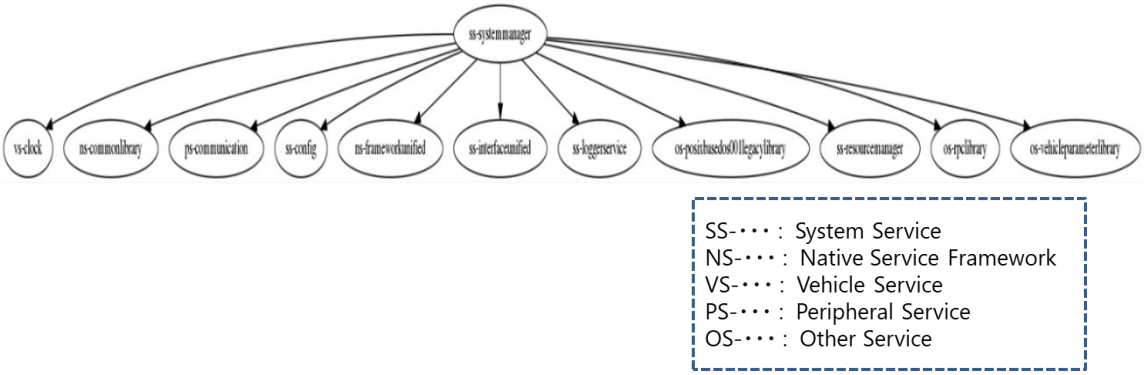


Fig. unit relation diagram of System Manager

■Use cases

[Main]

・Main (Start sequence)

・Main (Stop sequence)

2.1.2 Power Service

■Feature Overview

This service provides the feature to relay to System Manager.

Power Service complements the following features for System Manager.

1) System start-up

This notifies the start-up process completion to Power Service by the request from System Manager.

2) System shut down

This notifies the shutdown process completion to Power Service by the request from System Manager.

3) Malfunction detection

This detects the all types of the malfunctions.

3)-1. Heart Beat

This requests System Manager to monitor the malfunctions (Heartbeat) by the request from

Power Service.

3)-2. Service abnormal termination

This requests the system reset to Power Service by the malfunction notification from System Manager.

■Unit relation diagram

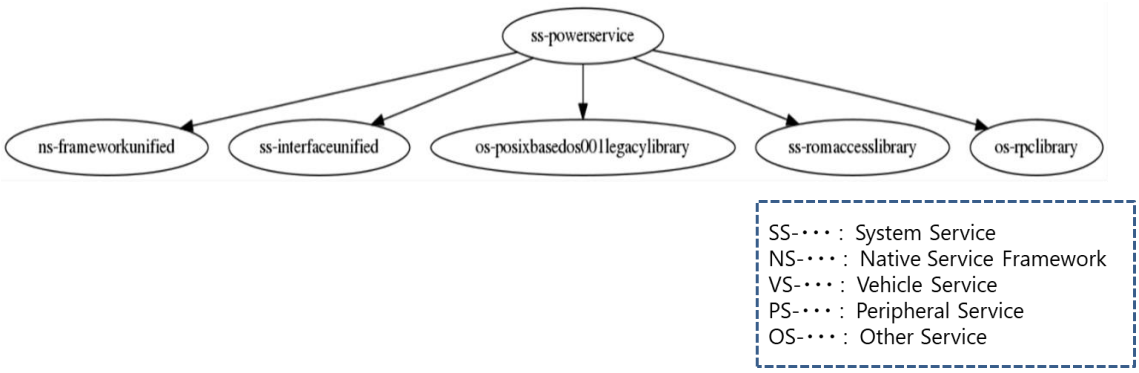


Fig. unit relation diagram of Power Manager

■Use cases

[Main]

・Main (Start sequence)

[Control]

・Start-up system

This notifies the start-up process completion to Power Service by the request from System Manager.

・Shut-down system

This notifies the shutdown process completion to Power Service by the request from System Manager.

[Notify]

・Detect malfunction (Heart Beat)

This requests System Manager to monitor the malfunctions (Heartbeat) by the request from Power Service.

・Detect malfunction (Service abnormal termination)

This requests the system reset to Power Service by the malfunction notification from System Manager ~~unit~~.

[Service]

・Start the session

・Terminate the session
・Request the system mode information
・Receive power level notification
・Receive shutdown POPUP request
・Receipt of Crank status notification
This receives Crank status notification.

2.1.3 Logger Service

■Feature Overview

This service provides the feature of collecting the log, saving in non-volatile area (USB/SD) or outputting to the network.

1) Request the log storing

On request of the receipt of the storing, this reads the log generated by other modules and saves it.

2) Periodically read

This periodically reads the log written by Logger.

3) Output log to the network

This outputs the log to the network by UDP.

■Unit relation diagram

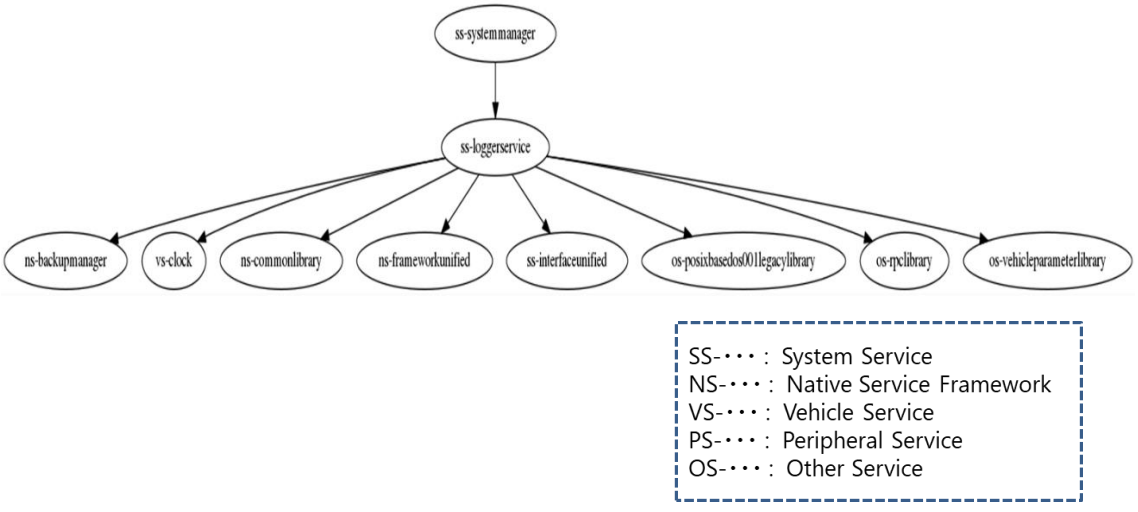


Fig. unit relation diagram of Logger Service

■Use case

[Main]

・Main (Start sequence)

・Main (Stop sequence)

[Control]

・Start session for the logging

・Terminate session for the logging

[Service]

・Callback against the each notifications from IPC

・Request output event log and emergency error log to USB/SD

・Read and log reset request of the event statistical log

・Request the start of CAN logging

[Event Data Send]

・Request notification to clear the event log

・Send CAN current date and time

・Send Vehicle identification number (VIN)

・Notify event to persist the event log

・Send information of the screen shot event response

・Send logging propriety information for the selected device

・Send logging propriety information for UDP

・Request notification to upload the event log

2.1.4 Task Manager

■Feature Overview

This service provides the features to play the role, as Launcher, to control the non-resident services.

Task Manager has the following features.

1) Service start and termination

This starts and terminates the non-resident services.

2) Malfunction detections

This detects the abnormal behaviors of the controlled services.

2)-1. Hang-up detections

This monitors the running status of the starting service and forces the corresponding service to

terminate at the hang-up detection.

2)-2. Service abnormal termination detections

When the service terminates abnormally, it assumes that the application using the applicable service detects this malfunction and executes the failure processing.

When the same service is continuously and abnormally until ACC-OFF status, Task Manager reboots the system.

2)-3. Malfunctions at the service start

When the start completion does not respond within the specified time after the service start, Task Manager recognizes the malfunctions and forces the applicable service to terminate.

Against the services, besides, Task Manager provides the processes at the start and termination, which shall be equipped by the non-resident service as the shared library (Primary Library).

The communication with Task Manager is hidden within Primary Library.

■Unit relation diagram

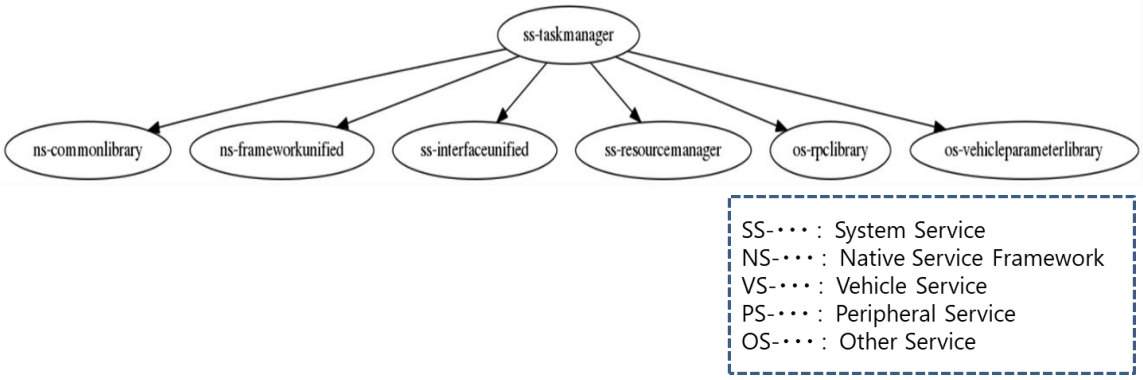


Fig. unit relation diagram of Task Manager

■Use case

[Main]

・Main (Start sequence)

・Main (Stop sequence)

[Task Manager]

・Initialize data before non-resident service start

・Start non-resident service process

・Terminate non-resident service process

・Detect service abnormal termination

・Start order maintenance

・Start non-resident service process

・Notify residual system memory decrease
・Release resource at the normal reboot
・Store log at the malfunction occurrence
・Acquire the specified service information
[primary]
・Notify start/termination event completion
・Set service timeout
・Acquire information of service started
・Acquire expanded information of service started
・Acquire the private data

2.1.5 Resource Manager

■Feature Overview

This service provides the feature to provide the abnormal status notifying feature in order to execute the failsafe process at the malfunction detections.
Resource Manager provides the following features.

1) CPU load monitoring

Monitor CPU load and execute LOGGING of the upper level process occupying CPU when the high load status is abnormally continued during the specified time.

2) System information provision

This obtains the residual memory information, write access status to NAND flash and communication volume in the network device after the power on.

3) System memory monitoring

This monitors the memory in the whole system. Issue an event when the residual memory falls below the certain level.

4) Watch Doc Timer (WDT) Update

This monitors the error detection for the software by WDT including MM microcomputer.
Recognizes the malfunction and triggers WDT when FIFO process and RR process keep occupying CPU for the long time.

5) Debug display information provision

This provides the residual memory information, CPU load information and residual memory

information for CMA for the debug display.

6) Log output of the minimum residual memory information at ACC-OFF

This maintains the minimum residual memory information after the start and output the log at ACC-OFF detection.

■Unit relation diagram

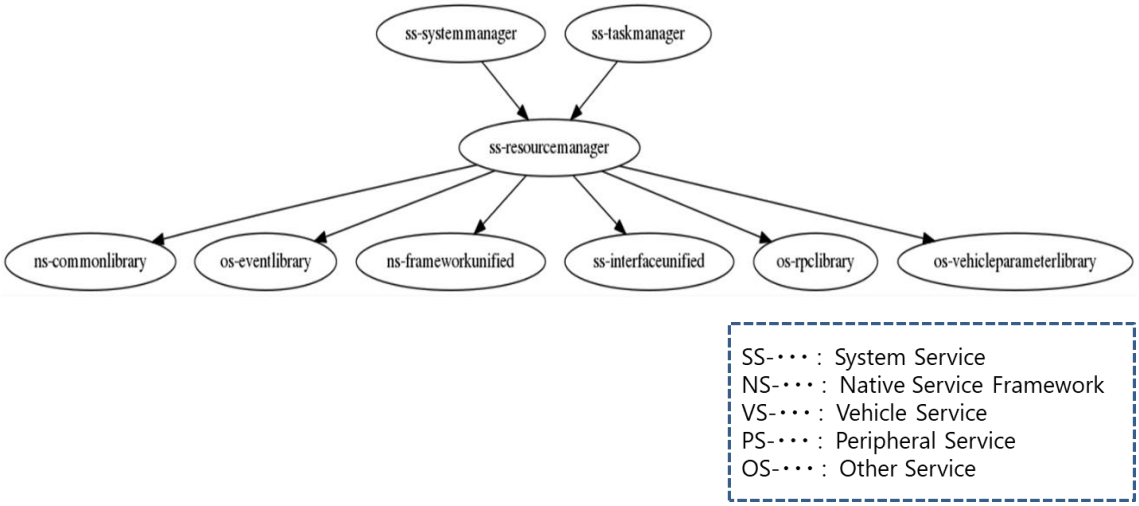


Fig. unit relation diagram of Resource Manager

■Use case

[Main]

・Main (Start sequence)

[Control]

・Initialize setting

・Monitor system memory

・Start event handler

2.1.6 Rom Access Library

■Feature Overview

This service provides the features of common process to control the external device.

1) libssaccess

This provides below features according to specified NOR-Flash device pathname and block ID.

1)-1. This reads data from specified main or backup block

1)-2. This writes data to specified main and backup blocks

2) libSS_RomAccessIf

This provides features of Read/Write data to non-volatile area according to defined ID. ID belongs to category. Category has follow properties.

- Boot area, RAM area, ROM area

■Unit relation diagram

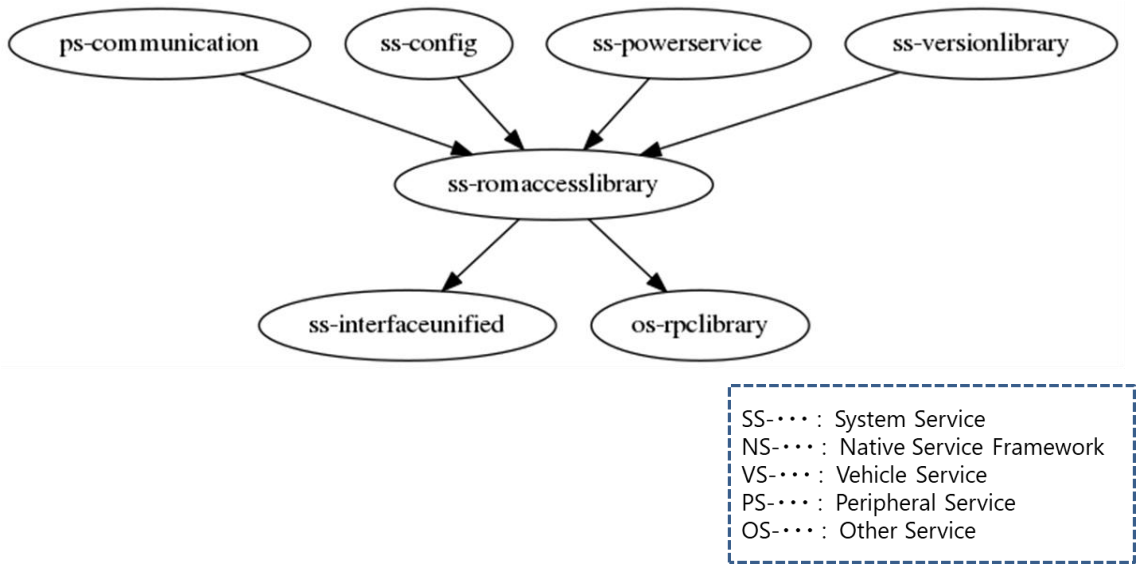


Fig. unit relation diagram of Rom Access Library

■Use cases

[libssaccess]

・Set boot Info
・Get boot Info
・Get Ram Info
・Set Ram Info
・Get the Boot Mode
・Set Boot Mode
・Get Data Reset Mode
・Set Data Reset Mode
・Get Data Reset Mode Faster
・Set the Active Flash loader
・Get Error Logging Count
・Set Error Logging Count

・Get Last Illegal Reset status
・Set Last Illegal Reset status
・Get the Last User Mode
・Set the Last User Mode
・Get the Limp Home Cut Off Request Mode
・Set the Limp Home Cut Off Request Mode
・Get Next Wakeup Type
・Set Next Wakeup Type
・Get the Production Mode
・Set the Production Mode
・Get product information data
・Set product information data
・Set Reset Count
・Set Reset Count
・Get the Signature
・Set the Signature
・Get the Transport Mode
・Set the Transport Mode
・Get Program Update Status
・Set Program Update Status
・initialize rom access

[libSS_RomAccessIf]

・read data
・write data

2.1.7 Config

■Feature Overview

This service uses for the registration of outside services targeted for collection of Out of Memory Killer, registration group re-launch services, and the setting environment variable of System Manager.

■Use cases

・initialize system environment

■Unit relation diagram

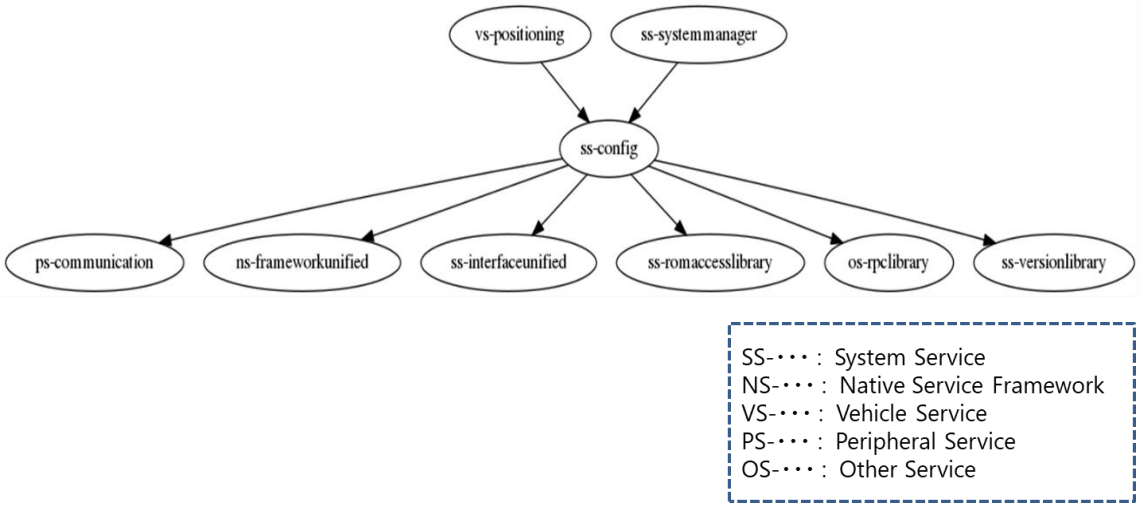


Fig. unit relation diagram of Config

2.1.8 Interface unified

■Feature Overview

This service provides the features for timer and the library of system manager.

1) Timer
2) System startup, system termination, error detection, and error log output features in System Manager

■Unit relation diagram

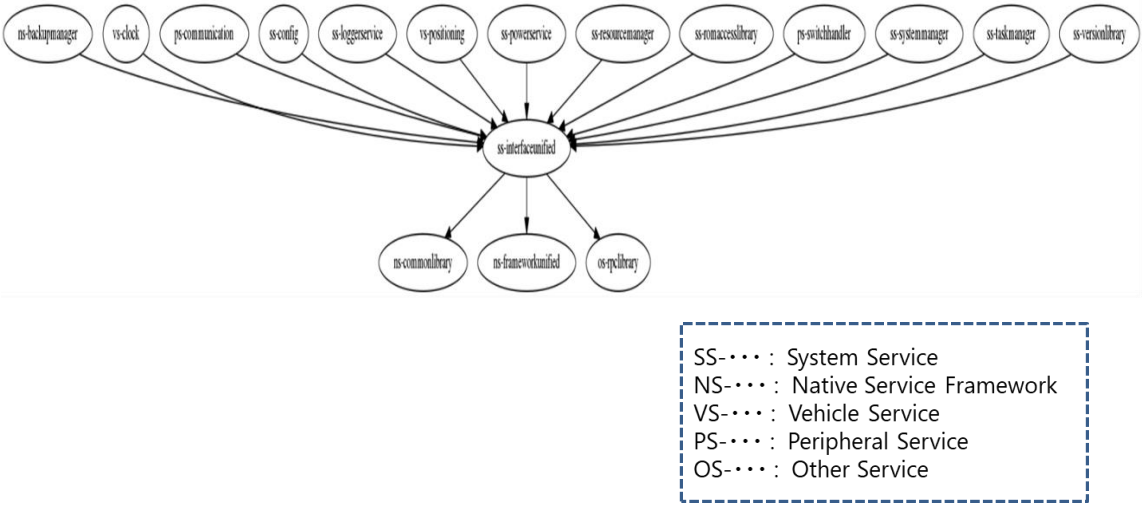


Fig. unit relation diagram of Interface unified

■Use case

[Timer]

・Register and terminate One-Shot Timer（without Start() parameter）

・Register and terminate One-Shot Timer（with Start() parameter）

[System IF]

・Start service

・Stop service

・Dump output for debug(Abnormal detection is included)

・Store Log in the internal non-volatile memory

・Copy all files in the log stored area in the internal non-volatile memory

・Output of specified character strings information at the log storing to the system information log

・Record the mode at the next boot to the non-volatile memory

・Record the data reset mode at the next boot to non-volatile memory

・Reset whole board

・Register the callback function invoked in the receiving of System Manager

・Clear(Delete) stored log in the internal non-volatile memory

・Record updated status in the non-volatile information

・Write request to the specified field of the shared memory with Boot Loader

・Acquire expanded information of the start and shutdown parameter

・Store the start order

・Dump data contents writing in OnDebugDumpResponseReceived*2 sending

*2 OnDebugDumpResponseReceived is API belonging

2.2 Native Service Framework

■Overview

Native service is a service package that provides the following features

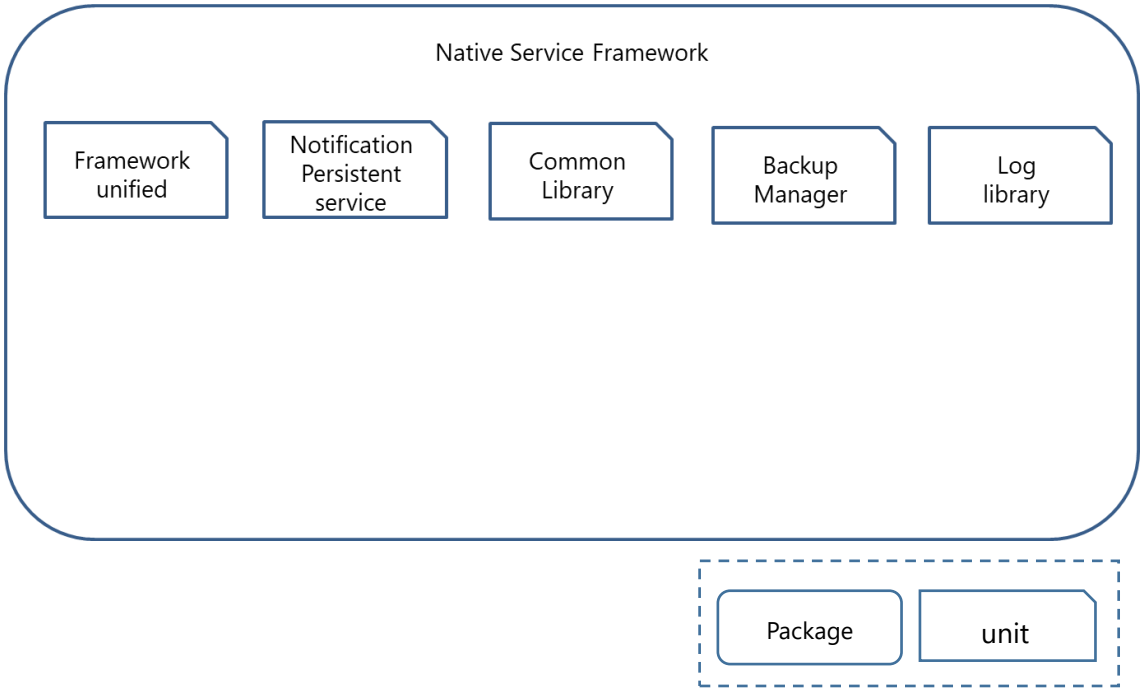


Fig. Native Service Framework Package

2.2.1 Framework unified

■Feature Overview

This service provides the following features.

1) Handle

This provides a type used to hide the internal structure of the instance.

2) Dispatcher

This provides a mechanism for receiving messages and registering callback features. If a service wants to receive message from other module, dispatcher is in need.

3) One-to-one inter-process communication

This provides a mechanism to communicate with any process.

4) One-to-many inter-process communication

This provides a mechanism to communicate with many processes. Execute one-to-one process communication by service name, command and session.

5) Session

This provides a mechanism to establish a session (communication path) with the service that wants to communicate.

6) Abnormal Monitor

This provides a feature to monitor abnormalities.

- Mutual abnormal monitor when session is build
- Abnormal monitor when calling sync API
- Abnormal monitor of multi cast service

■Unit relation diagram

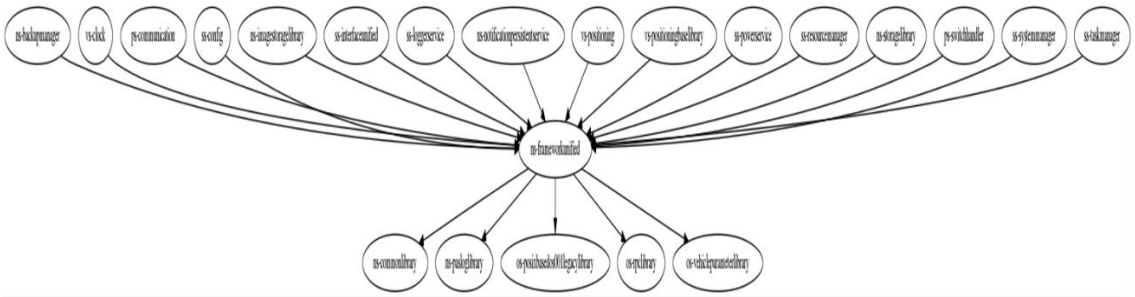


Fig. unit relation diagram of Framework unified

■Use cases

・Start-up service
・Register subscription event
・Register unsubscribe event
・Get service usage status
・Set / Get application data
・Defer Message
・Open / Close session open
・Notify session event
・Open / Close message queue
・Synchronize communication
・Asynchronize communication
・Transmit response data

・Receive message data
・Communicate by zero copy
・Receive ReNon-blocking message
・Discard received message
・Start sub thread(no loop)
・Start sub thread has loop (sub thread communication)
・Callback timer
・Read / Write ring buffer
・Read / Write configuration file
・Initialize
・Output log
・Clear force
・Set / Get log output control word
・Operate log file
・Set / Write log Level
・Change to real-time log output
・Send log event
・Session management(data pool)

2.2.1 Notification Persistent Service

■Feature Overview

This service provides the following features.

1) Notification feature

This provides a mechanism for message communication between processes in Publish-Subscribe format.

2) Persistent feature

2)-1. Release

This copies the file on the volatile memory file system specified by the client to the file on the nonvolatile memory file system.

2)-2. Load

This copies a file saved on the nonvolatile memory file system to a file on the volatile memory file system specified by the client.

■Unit relation diagram

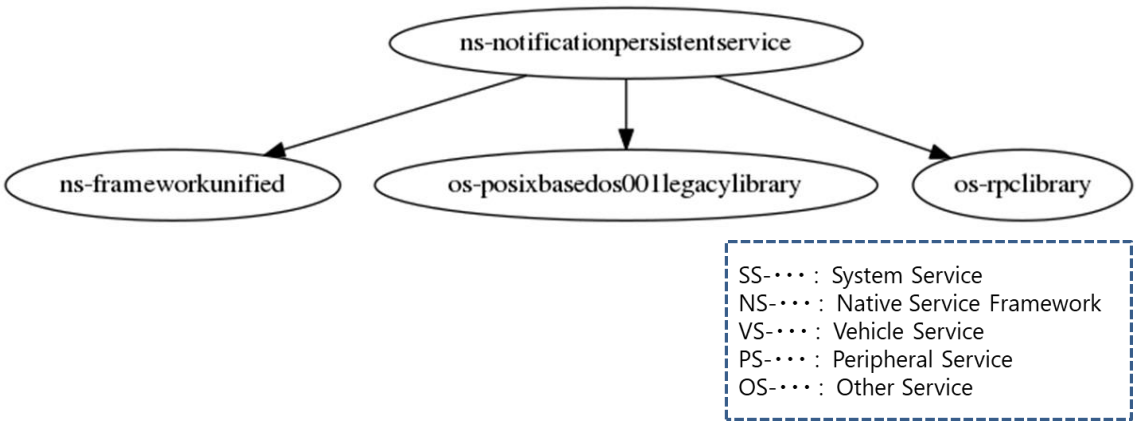


Fig. unit relation diagram of Notification Persistent Service

■Use cases

[Notification feature]

- Broadcast
- Register Immediate Notification Data
- Get Notification Data
- Unsubscribe notification
- Unsubscribe notifications
- Unregister notification
- Unregister notifications
- Set notification default value

[Persistent feature]

- Register Persistent file
- Set file persistent type
- Release persistent file
- Load persistent file
- Register persistent folder
- Set folder persistent type
- Release persistent folder
- Load persistent folder
- Synchronize persistent data
- Clear persisted data
- Release Persistent Data
- Set Persistent Notification Type

2.2.2 Common Library

■Feature Overview

This service provides common processing of process management function.

The main usage procedures for the process management process of this function show below.

1) Set process attributes.

2) Create a process from process attributes.

3) Register the process No. to Tls(Transport Layer Security).

4) Register process information in Monitor.

5) Create a process group.

6) Register the created process in a group.

7) Use mutex and semaphore for exclusion between processes.

8) Create a thread.

■Unit relation diagram

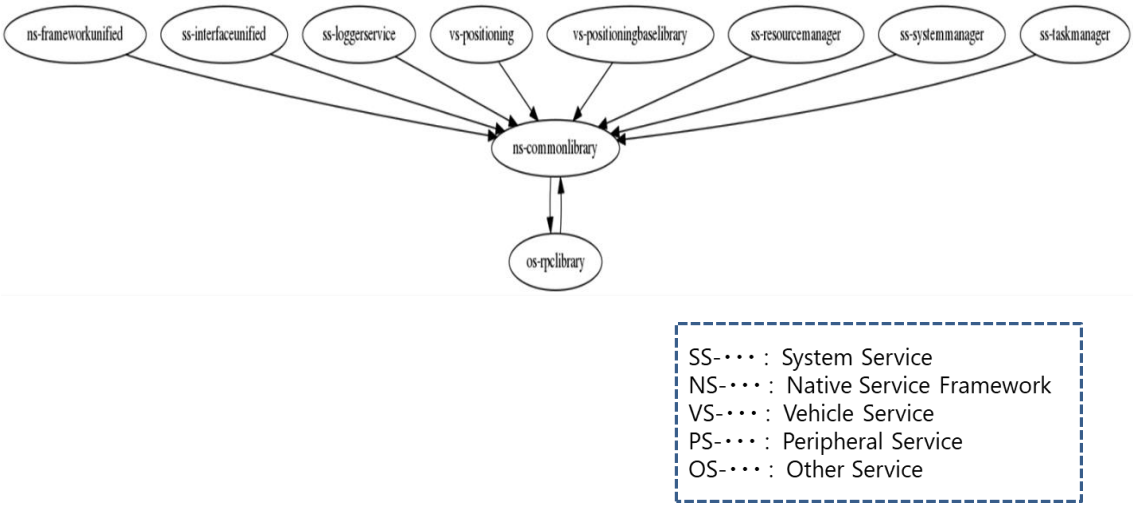


Fig. unit relation diagram of Common Library

■Use cases

- Set process attribute settings
- Create process
- Group process
- Exclusive control by mutex
- Create thread
- Exclusive control by semaphore
- Maintain the state of each process
- Control the identification information of the created process

2.2.3 Backup Manager

■Feature Overview

This service provides functions of Read/Write non-volatile date according to defined ID. ID belongs to category. Category has the following properties.

1) Use NAND
2) Use Cache DRAM
3) Use Backup DRAM
4) Real-time write
5) Backup cycle

Backup Manager parses and uses XML file where attribute information of backup information are described.
Category name, item name, ID(Area ID) and size of backup area are described in XML file. The item belongs to one category and the following attributes are set for each category.

■Unit relation diagram

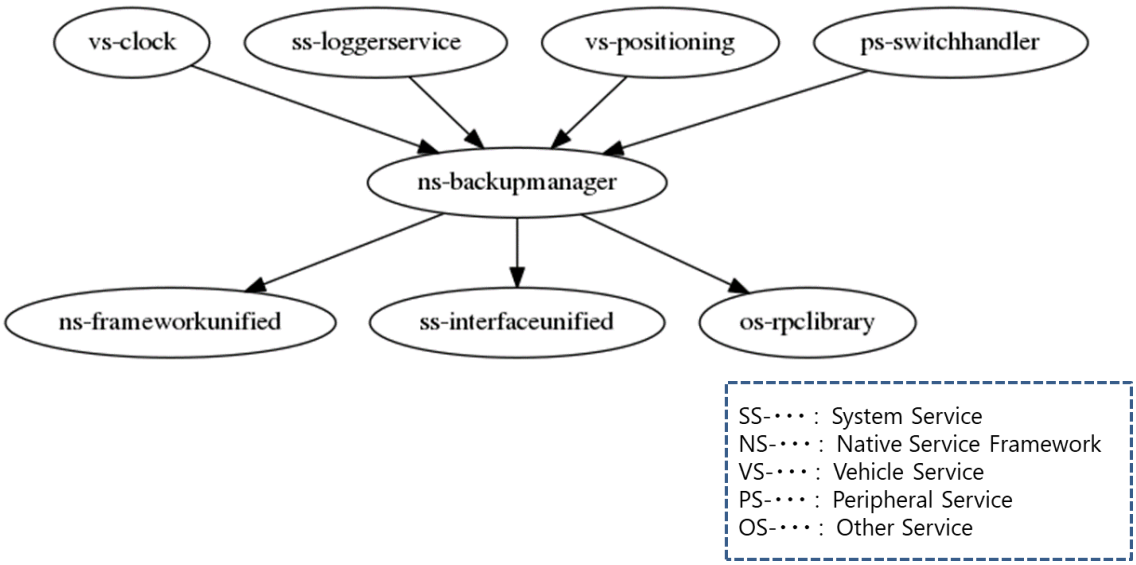


Fig. unit relation diagram of Backup Manager

■Use case

- Start Backup Manager service
- Read data from backup
- Write data to backup
- Fill data to backup
- Get data size of backup
- Check data's validity in backup
- Stop Backup Manager service

2.2.4 Log Library

■Feature Overview

This service provides the log freeze function.

■Use case

Freeze Log

■Unit relation diagram

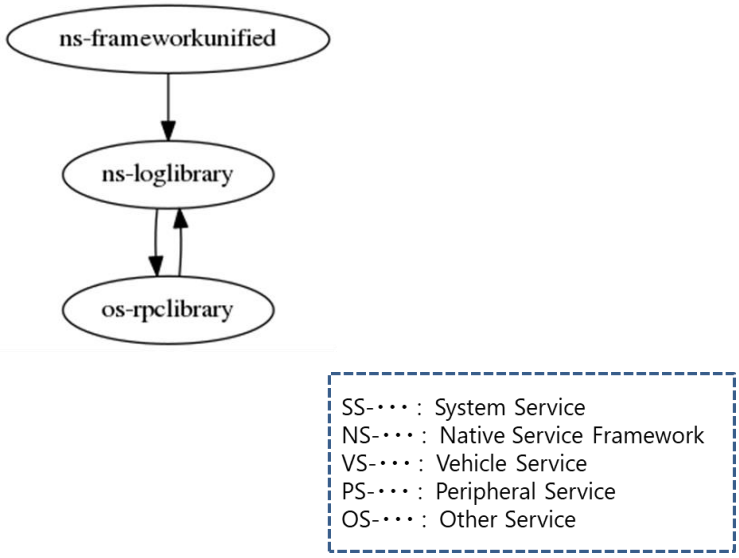


Fig. unit relation diagram of Log Library

2.2.5 Version Library

■Feature Overview

This service provides the features to set/get Version information.

■Unit relation diagram

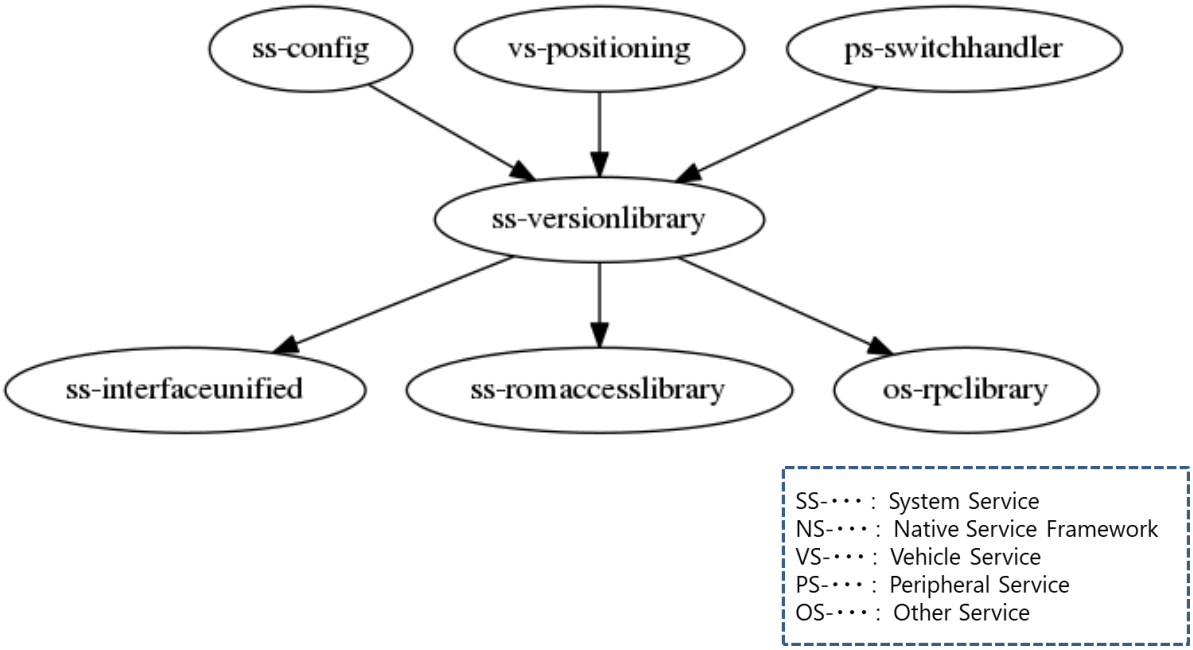


Fig. unit relation diagram of Version Library

■Use case

・Acquire version information and set-up

2.3 Peripheral Service

■Overview

Peripheral service is a service package that provides the following functions

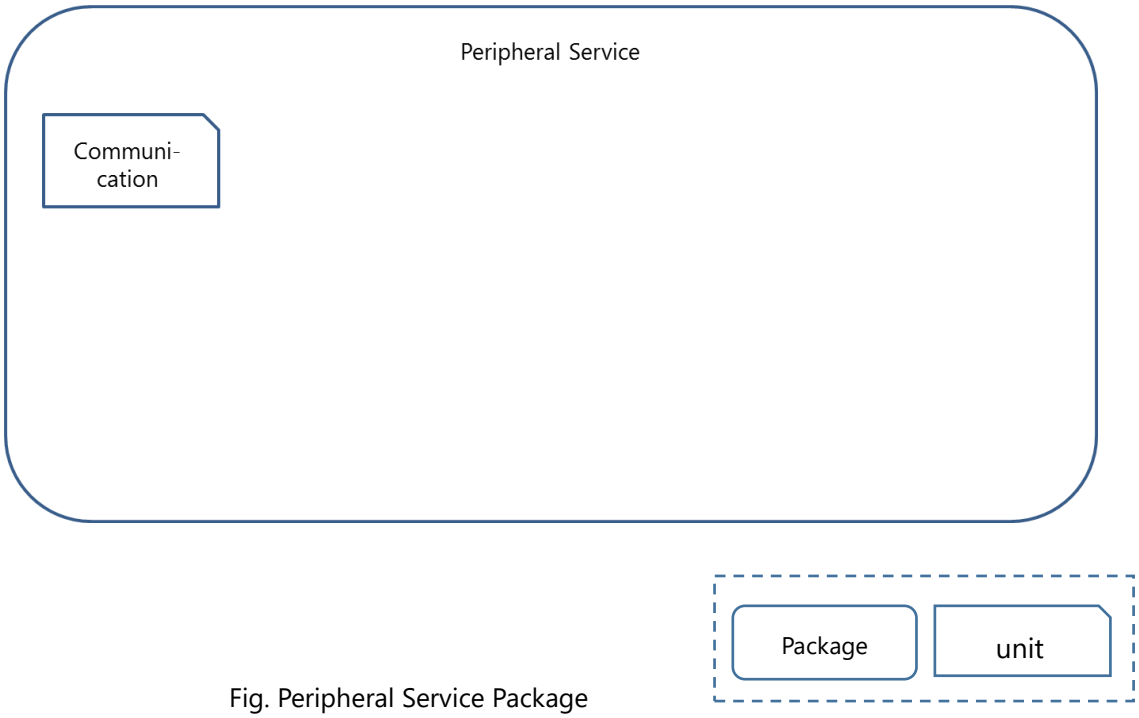


Fig. Peripheral Service Package

2.3.1 Communication

■Feature Overview

This service provides the following communication functions.

1) CAN

This supports the CAN data communication function by CAN HAL.

■Unit relation diagram

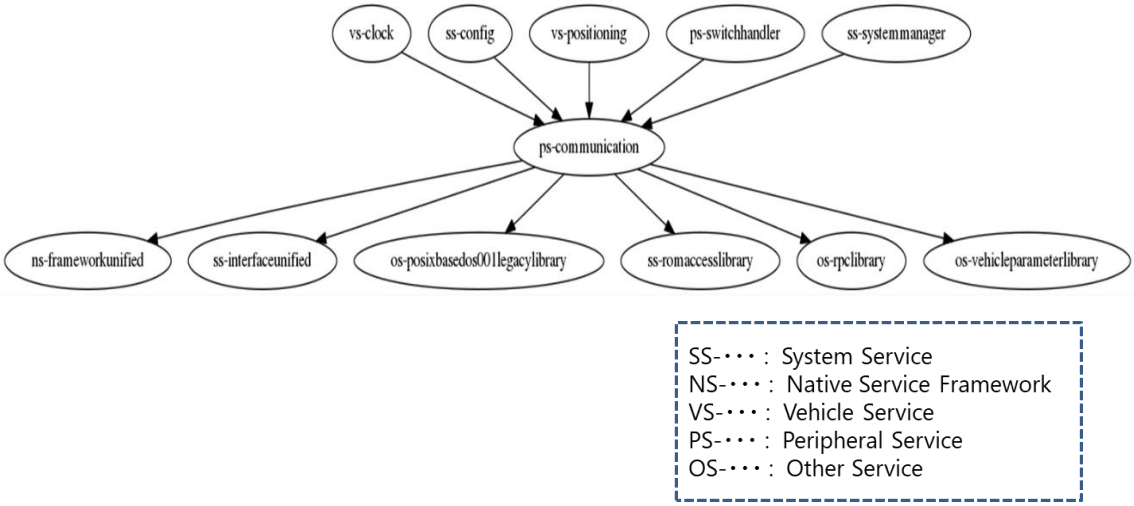


Fig. unit relation diagram of Communication

■Use cases

[CAN]

- Start to send CAN data transmission
- Start to send the CAN data
- Notify of the end of send sequence
- Deliver registration of CAN data
- Receive the CAN command
- Receive the CAN data
- Delete all the Delivery registration of CAN data
- Start to control CAN command
- Start to watch CAN data
- Suspend CAN communication

2.4 Vehicle Service

■Overview

Vehicle service is a service package that provides the following functions

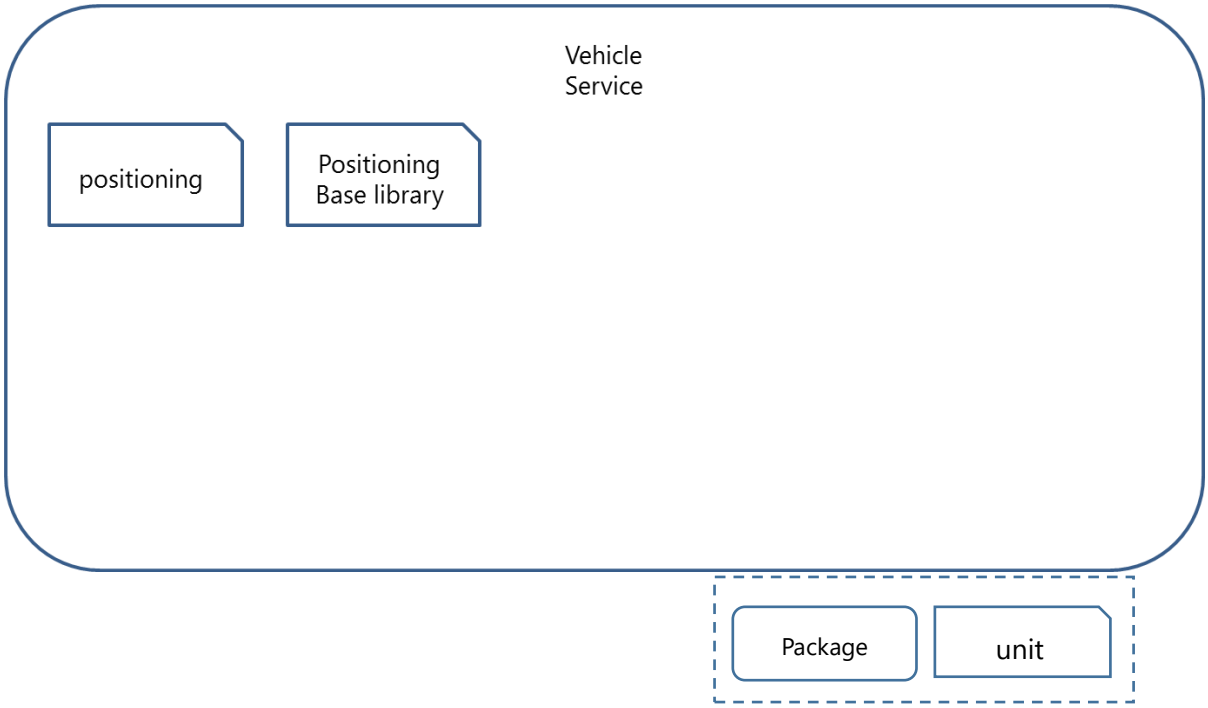


Fig. Vehicle Service Package

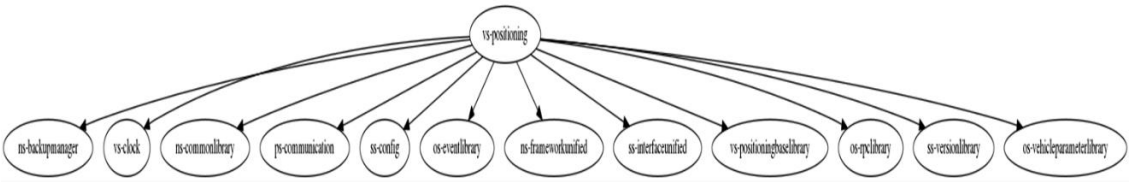
2.4.1 Positioning

■Feature Overview

This service provides GPS and Sensor functions.

1) GPS
2) Sensor functions

■Unit relation diagram



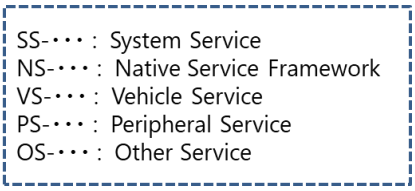


Fig. unit relation diagram of Positioning

■Use cases

[GSP]

- Get longitude and latitude by Sync method
- Subscribe longitude and latitude notify
- Subscribe altitude notify
- Subscribe speed notify
- Subscribe heading notify
- Get altitude by Sync method
- Get speed by Sync method
- Get heading by Sync method
- Set speed information
- Set location information(longitude, latitude, altitude, heading)
- Subscribe GPS time notify
- Request GPS setting
- Set GPS information
- Get GPS information
- Request GPS reset
- Get GPS version
- Set GPS time by Pub-Sub mode
- Set GPS time by diagnosis function by Sync mode
- Get GPS time by Sync mode
- Get Sensor data by Sync method
- Subscribe Sensor data notify
- Register listener that first send extension Sensor package
- Notify GPS information

[Sensor functions]

- Notify Line Sensor information of vehicle
- Notify first Line Sensor information of vehicle
- Notify speed of vehicle
- Notify reverse signal of vehicle

・ Notify speed pulse of vehicle

2.4.2 Positioning Base Library

■Feature Overview

This service provides the base function for positioning.

■Unit relation diagram

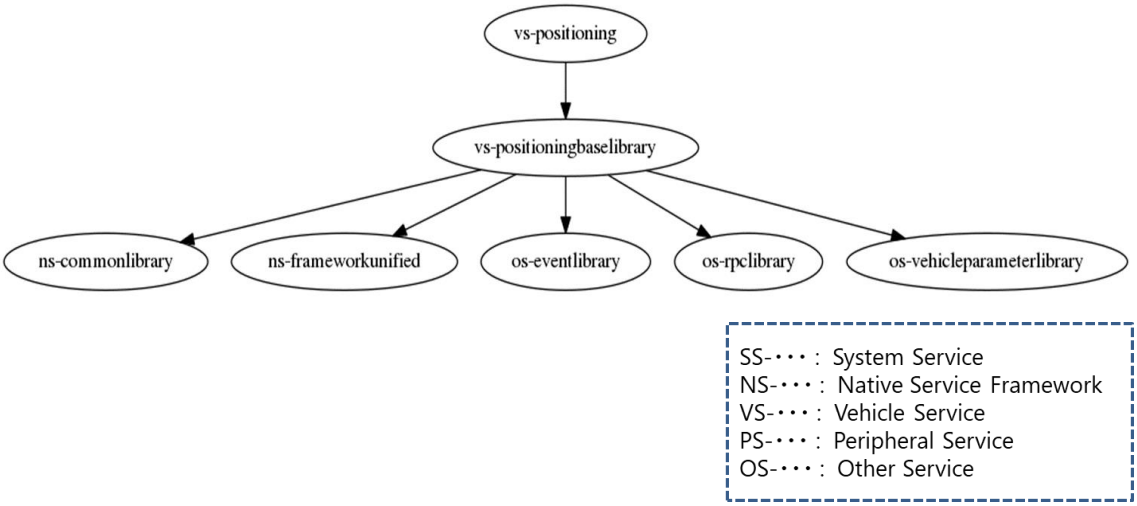


Fig. unit relation diagram of Positioning Base Library

■Use cases

[Message]

・ Create the message queue
・ Send the message based on PNO
・ Send the message based on process name
・ Receive the message based on PNO
・ Get message buffer inside process (not copy API)
・ Send message inside process (not copy API)
・ Create mutex between process
・ Change process name to PNO
・ Change PNO to process name
・ Check resource(message control table)
・ Check resource(process name - PNO table)
・ Release resource(message control table)
・ Release resource(mutex control information)
・ Release resource(process name - PNO table)

- Create semaphore and get semaphore ID
- Get semaphore according to semaphore ID
- Release semaphore according to semaphore ID

[Share Data]

- Create share data
- Link share data

[Timer]

- Start timer
- Stop timer

[Event]

- Create event according to event name, and get event ID
- Wait for the event according to event ID and event value
- Set the event
- Delete event according to event ID
- Get dump information(message)
- Get dump information(mutex)
- Get dump information(timer)
- Get dump information(event)
- Get dump information(memory)
- Get dump information(other)

[Handle]

- Get application handle
- Set application handle

2.5 Other Service

■Overview

Other service is a service package that provides the following functions

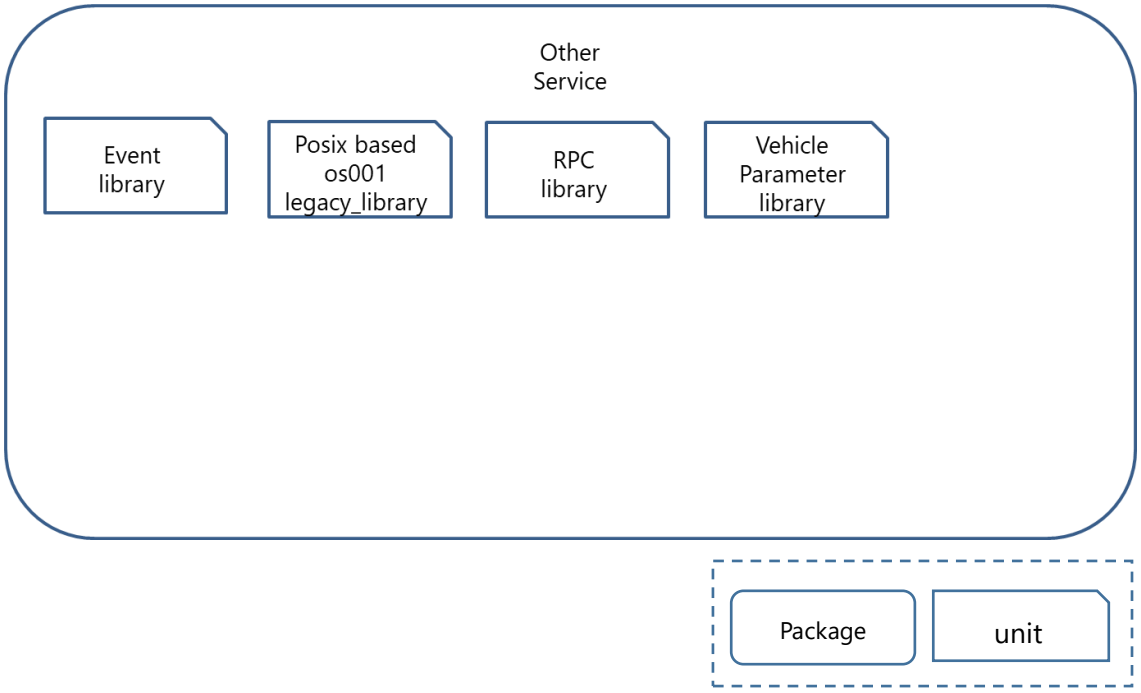


Fig. Other Service Package

2.5.1 Event library

■Feature Overview

This service provides the sending and receiving of message between processes(thread). According to the transferred data, there are two kinds of functions as below:

1) Event's data 's send and receive

This communicates using fixed length bit string (flag).

2) Message's data send and receive

This communicates using a variable-length record byte string (message).

■Unit relation diagram

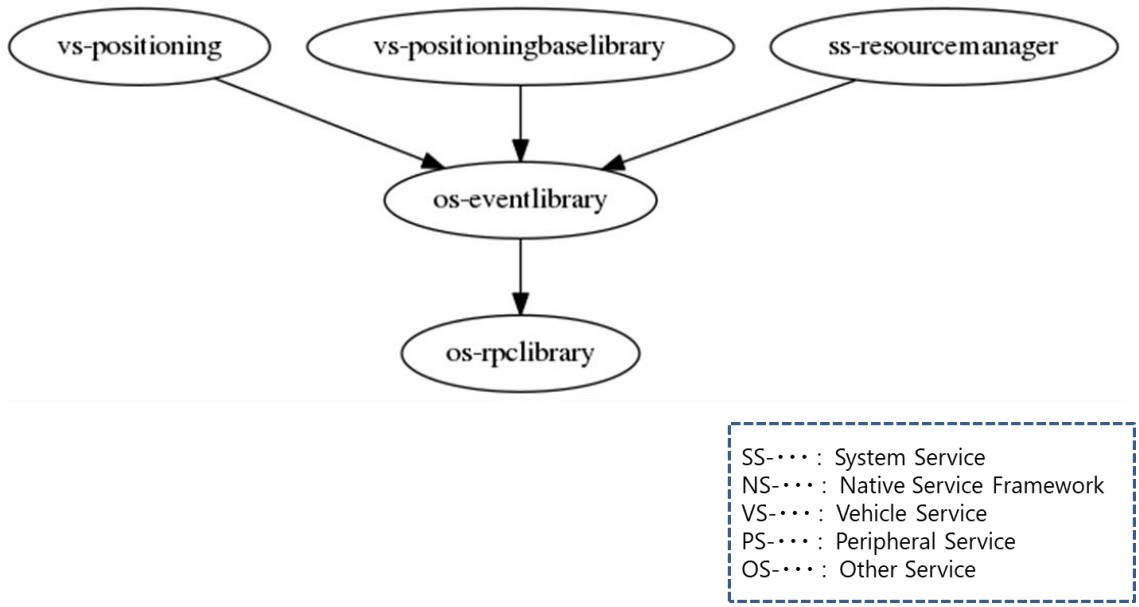


Fig. unit relation diagram of Event Library

■Use cases

[Event's data 's send and receive]

- Change from mID to flagID
- Create flag
- Create flag(mID specified)
- Create a flag(ID is allocated automatically)
- Change from mID to 64bit's flagID
- Create 64bit's flag
- Create 64bit's flag(mID specified)
- Create 64bit's flag(ID is allocated automatically)
- Send flag
- Send flag(mID specified)
- Send 64bit's flag
- Send 64bit's flag(mID specified)
- Get the first reached event(non-block)
- Get the first reached event(non-destructive)
- Get flag event(non-block)
- Get flag event(block)

・ Get flag event(non-destructive)
・ Get FD which is used to poll flag event
・ Get 64bit's flag event(non-block)
・ Get 64bit's flag event(block)
・ Get 64bit's flag event(non-destructive)
・ Delete a flag event
・ Delete flag(mID specified)

※mID: module(unit) ID

2.5.2 Posix based os001 legacy library

■Feature Overview

This service provides support function of system call.

1) Get the number of clock cycles
   This gets the number of clock cycles, time unit (microsecond).
2) Delay thread for the specified time
   This delays thread for the specified time, time unit (millisecond).
3) String handling
   This handles such as itoa, ultoa, strlcpy and strlcat.

■Use cases

[Get the number of clock cycles]

・ Get the number of clock cycles

[Delay thread for the specified time]

・ Delay thread for the specified time

[String handling]

・ Convert integer value to null terminated string
・ Convert unsigned long value to null terminated string
・ Copy the specified size of the source string to the destination string
・ Concatenate the specified size of the source string to the end of the destination string

■Unit relation diagram

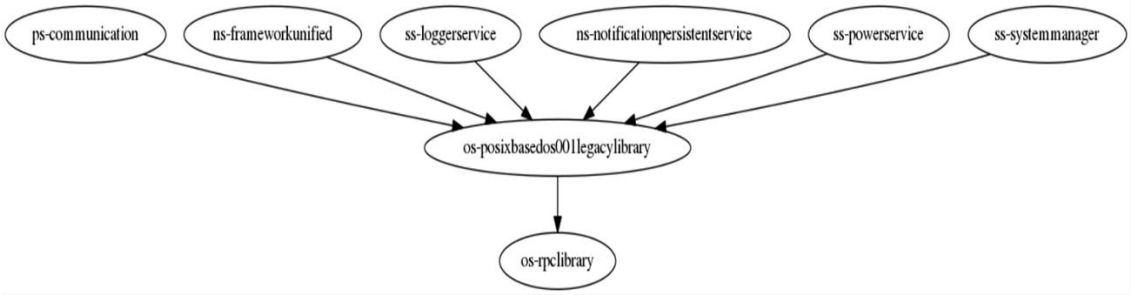


Fig. unit relation diagram of Posix based os001 legacy library

2.5.3 RPC library

■Feature Overview

This service provides the function that allows a subroutine or procedure in another address space to execute from a program.

■Use cases

- Start secure server
- Start server
- Start client
- Set the timeout's time
- Register the credential(UID,GID)
- Get the client' credential(UID,GID)
- Get the RPC server's status
- Marshall arguments
- Free the memory malloced in marshall
- Demarshall arguments
- Call the server's API
- Get FD
- Process the API call
- Record the API call log

- End RPC library
- Force clean up when process is over

■Unit relation diagram

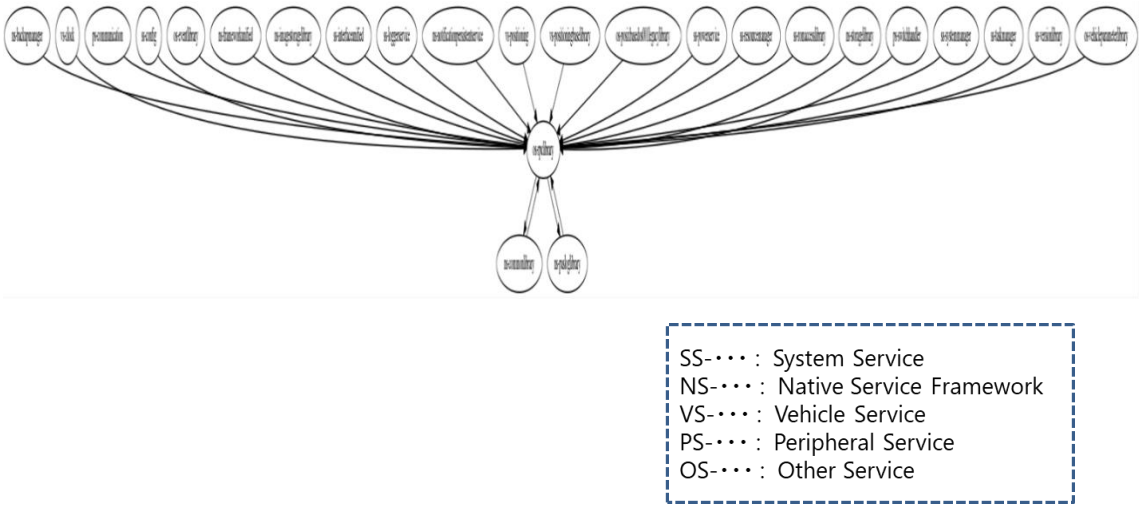


Fig. unit relation diagram of PRC library

2.5.4 Vehicle Parameter library

■Feature Overview

This service provides an API library, which uses to get environment variable.

1) Get vehicle parameter environment variable
2) Get function check result

■Use cases

- Get environment variable
- Get function check result

■Unit relation diagram

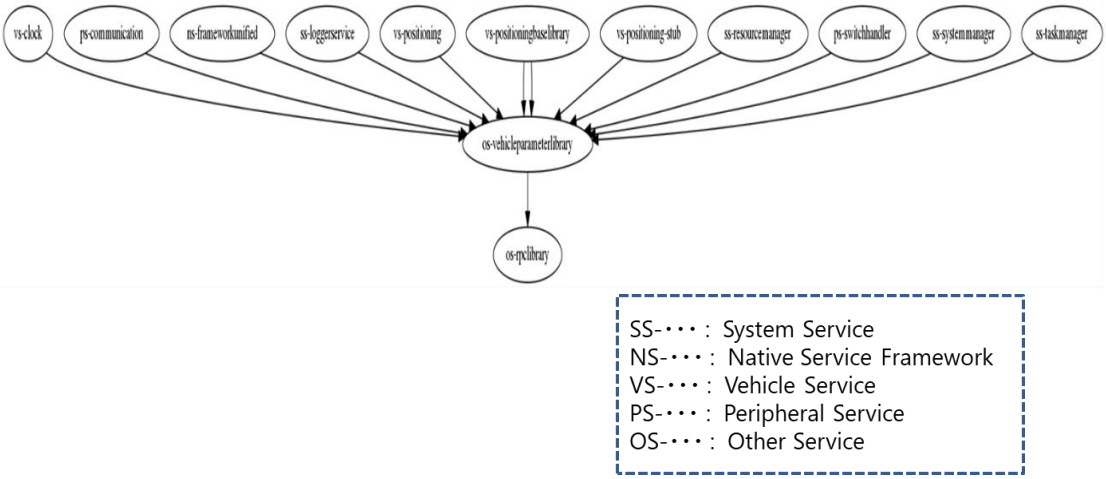


Fig. unit relation diagram of Vehicle Parameter library

3 Typical use-cases and sequence

Each service runs by starting services and communicating between services. Therefore, service activation and communication sequences show.

3.1 Launcher to control of start of the resident service

Resident services launch by System Manager. And resident services state according to the order set in the configuration file.

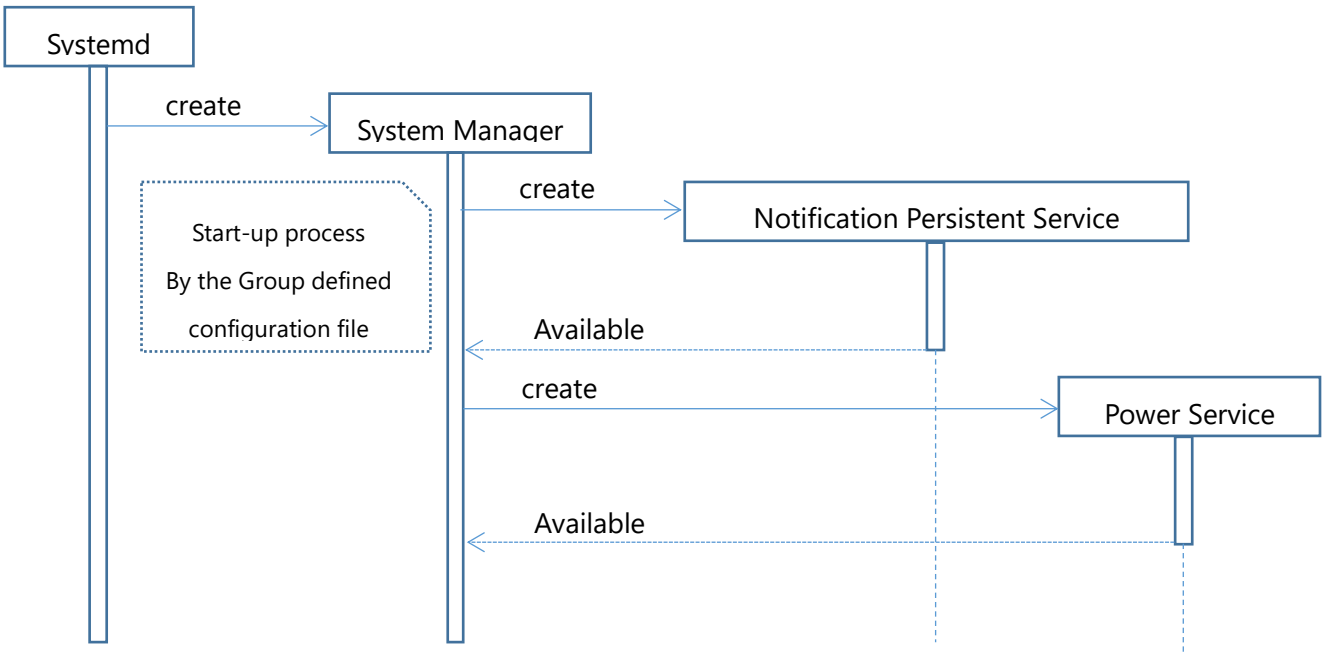


Fig. sequence of Launcher to control of start of the resident service

To use the services, clients need to access them after confirming the service availability because System Manager guarantees the service start order but does not guarantee the service availability.

When Availability of the resident services, which have a session with System Manager is not changed to TRUE within the specified time after the system start, System Manager judges that the system is in abnormal state and reboots the system.

Configuration file defines group elements such as the group name and the waiting time until the next group activation, and launch elements such as service name to activate the number of retries during error processing.

3.2 Launcher to control of start of the non-resident service

Non-resident services launch by Task Manager.

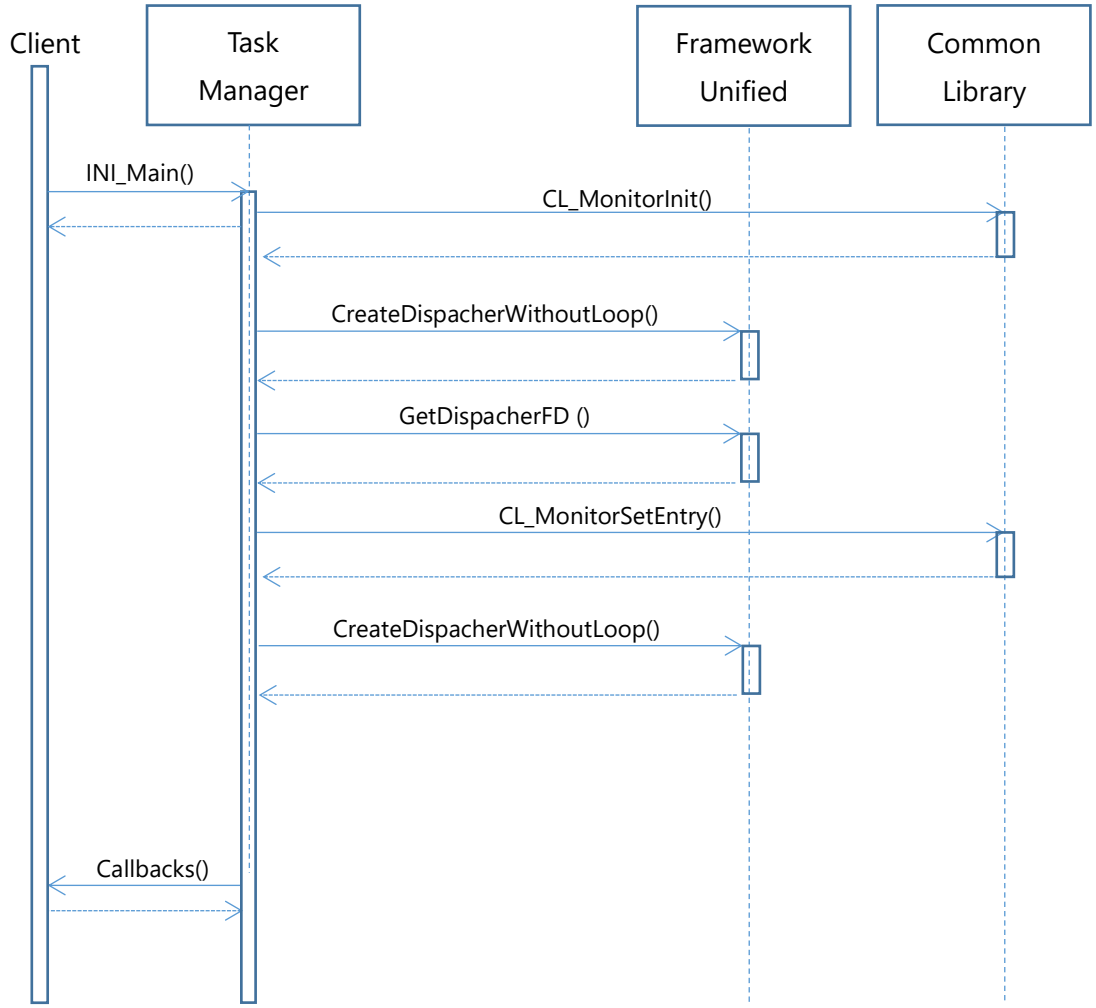


Fig. sequence of launcher to control of start of the non-resident service

3.3 Process communication

Execute one-to-one process communication by service name, command and session.

Command : Command can specify 32bit

Session : Session provides function of process connect.

Session ID is the assigned ID of session and it's unique in Dispatcher instance of server.(ID is automatically generated by CreateSession()).

3.3.1 Session build sequence

Session build sequence is as follows. This sequence is an example when using sync API.

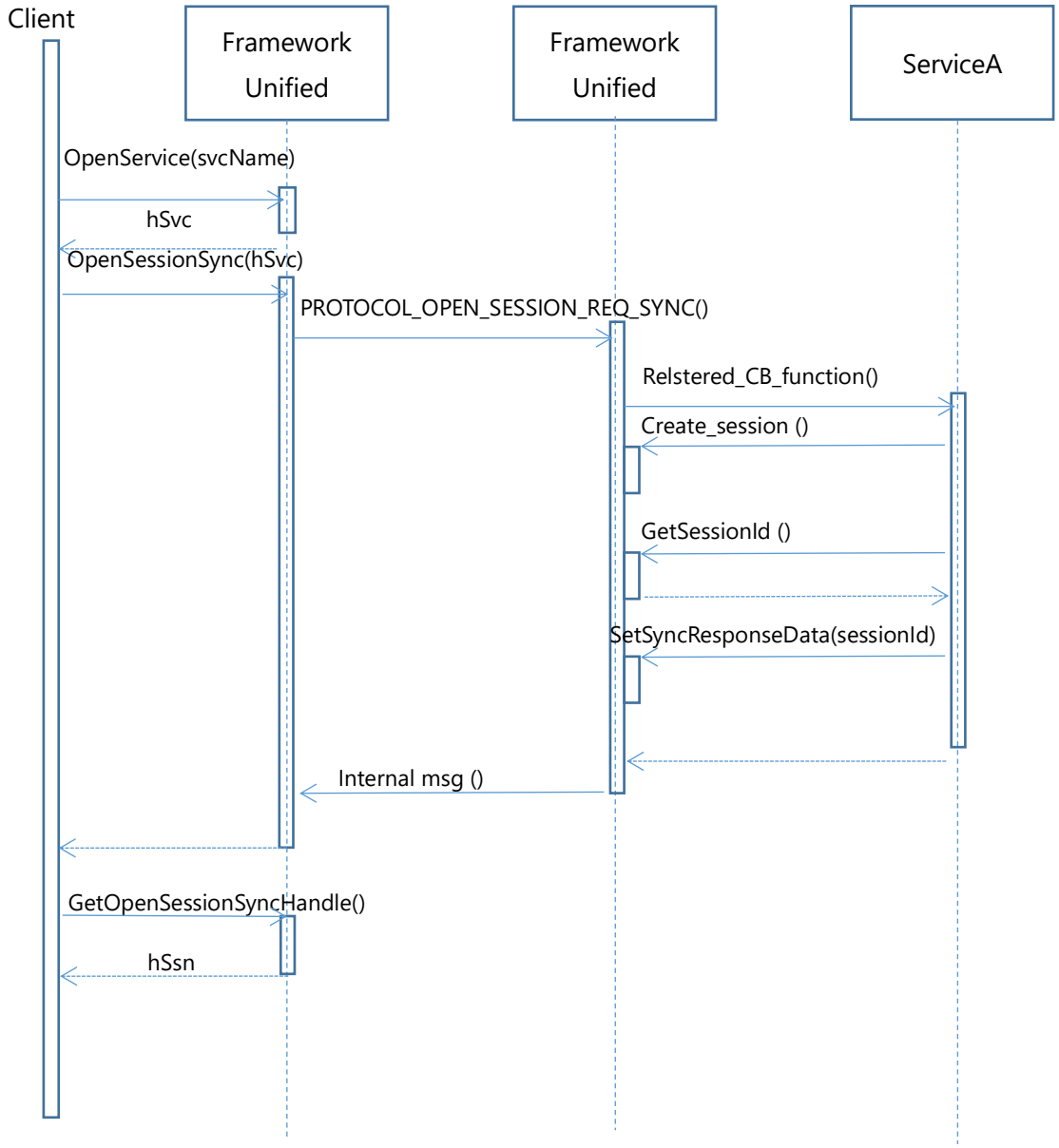


Fig. sequences of session build

3.3.2 Communication type

Sync message and asynchronous message are two method of one-to-one process communication.

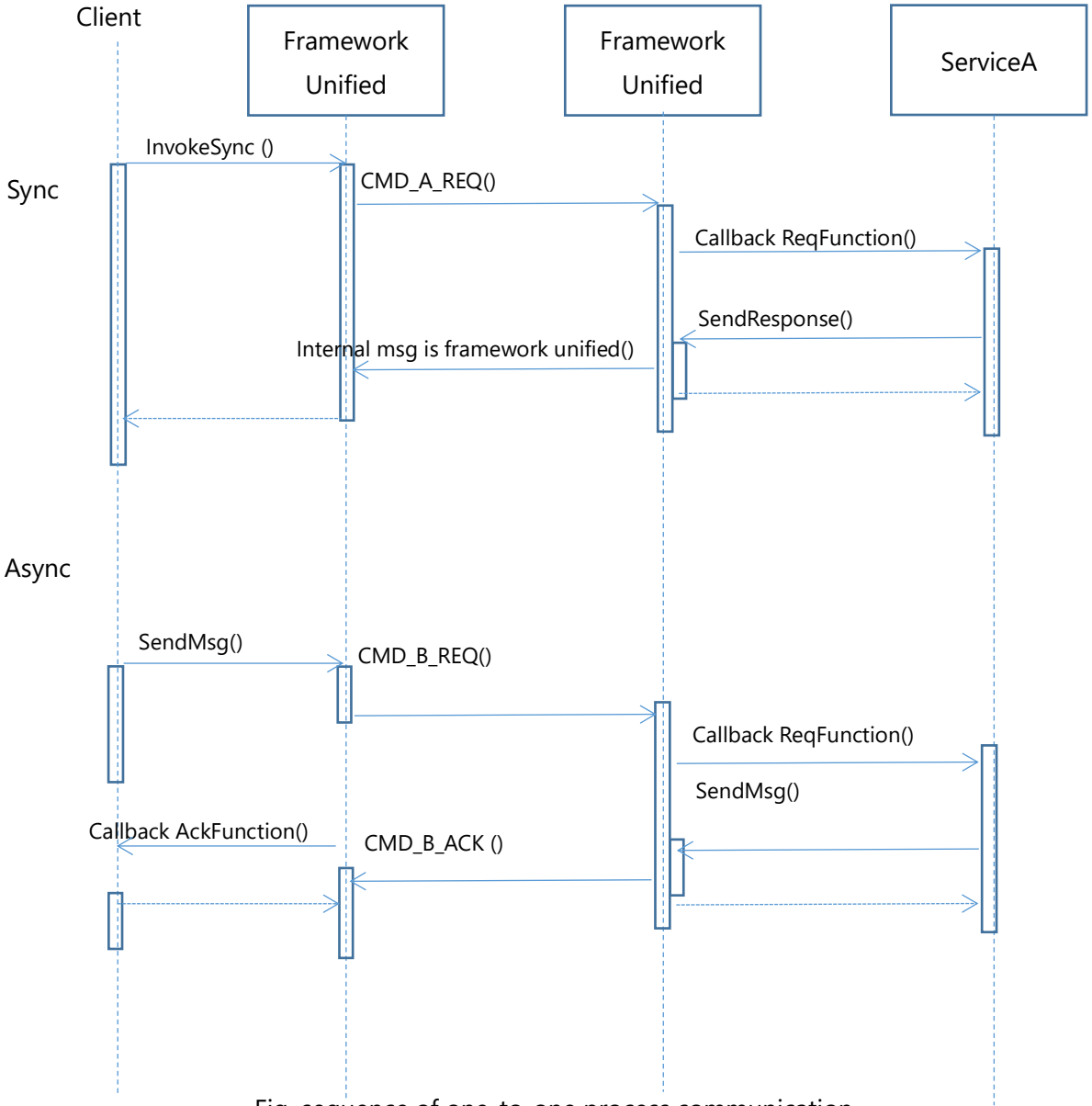


Fig. sequence of one-to-one process communication