

Base System

- Base System is software located in a layer close to the basic software (OS: Operating System)
- The Base System provides common functions that can be used by components in the system.

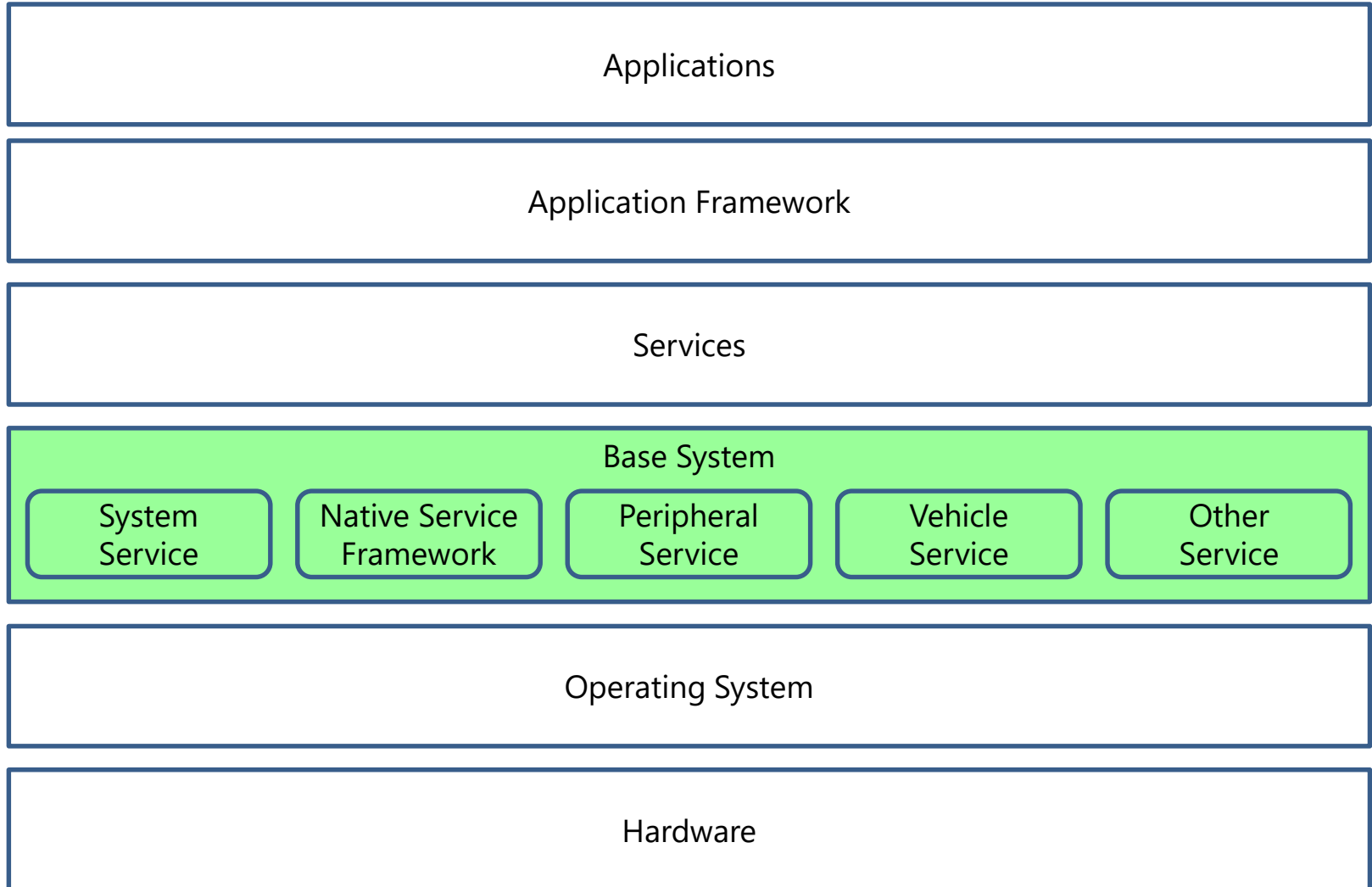
Common functions are classified into the following groups.

Common function group

- System Service :
 - Provide and execute processing related to the entire system.
- Native Service Framework:
 - Provide a means of communication between components and data interchange.
- Peripheral Service :
 - Provide control functions for peripheral devices.
- Vehicle Service :
 - In charge of vehicle network information acquisition and control.
- Other Service :
 - Common functions not provided by other BaseSystem components.

Base System

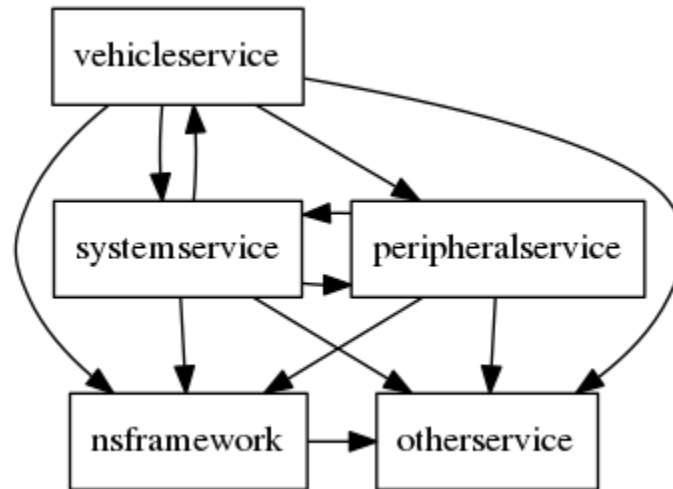
- Software Structure of System



Each Service Group

- System Service
- Native Service Framework
- Peripheral Service
- Vehicle Service
- Other Service

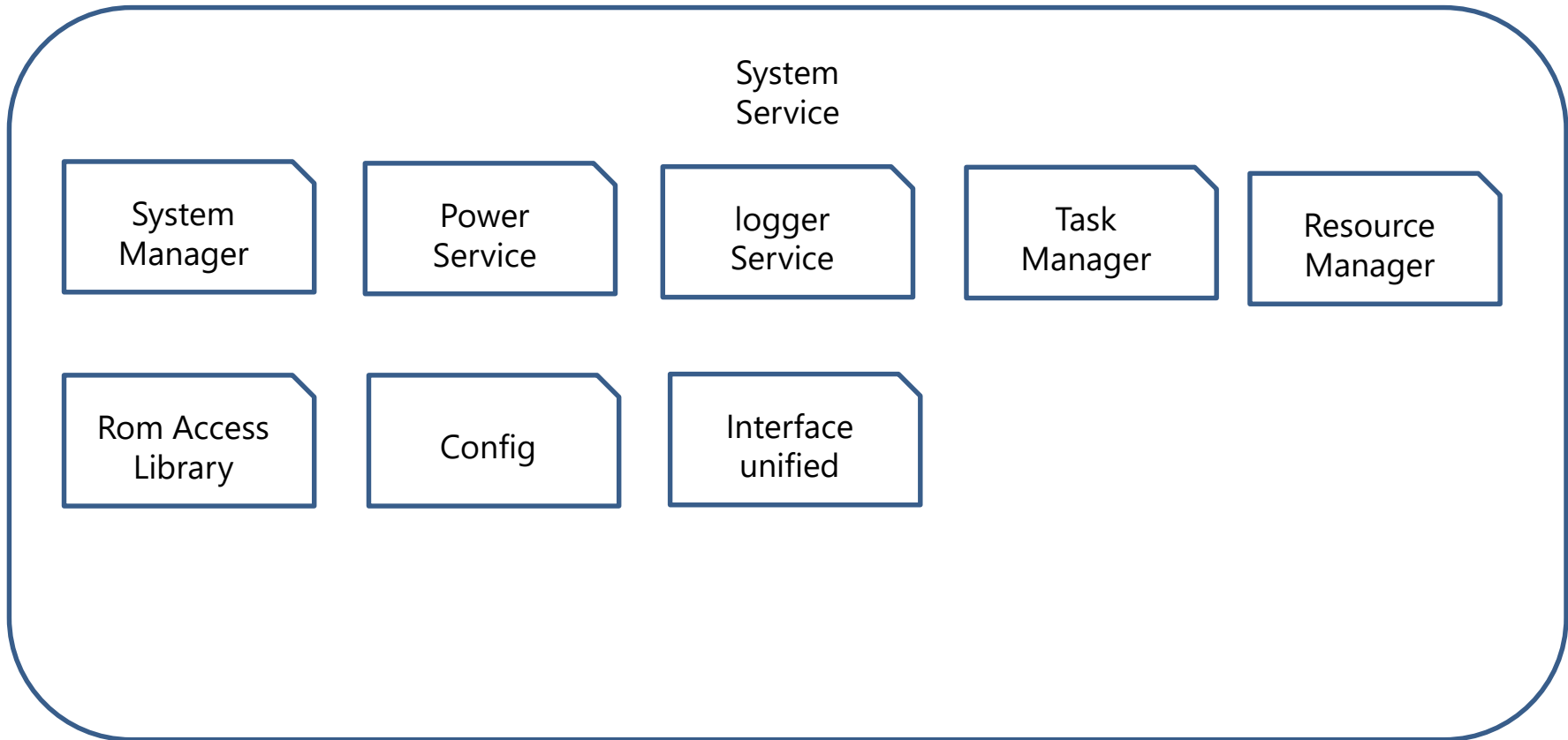
■ Group relation diagram



System Service

■ Overview

System service is service group that provide the following functions



System Manager

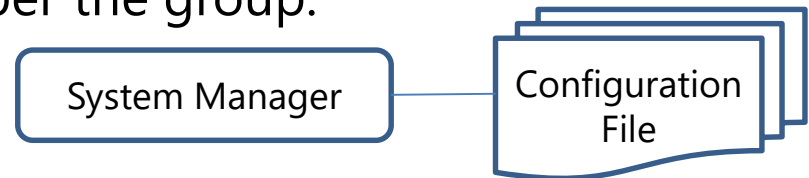
■ Functional Overview

This service provides the function to processes the system start and shutdown.

This provides the following features.

#1 System start-up

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



#2 System shut-down

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

System Manager

#3 Malfunction detections

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

1)Heart Beat

System Manager 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.

2)Services abnormally shutdown

System Manager 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.

System Manager

3) System memory shortage

System Manager 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.

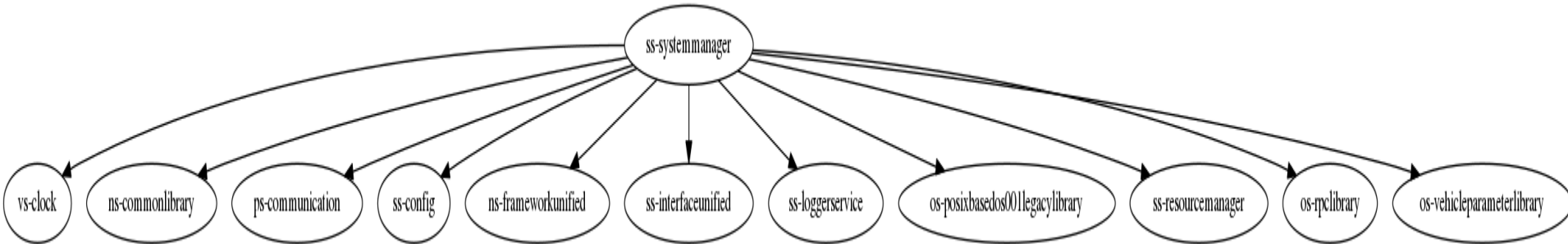
* The Resource Manager is a function within the system service. The Resource Manager will be described later.

#4 Abnormal log output

System Manager provides the function 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.

System Manager

■ Unit relation diagram



systemmanager.png

SS-···· : System Service
NS-···· : Native Service Framework
VS-···· : Vehicle Service
PS-···· : Peripheral Service
OS-···· : Other Service

System Manager

■ Use case

[Main]

- Main (Start sequence)
- Main (Stop sequence)

Power Service

■ Functional Overview

This service provides the function to relay to System Manager unit

Power Service complements the following functions for System Manager.

#1 System start-up

Notify the start-up process completion to PSM unit by the request from System Manager unit.

#2 System shut down

Notify the shutdown process completion to PSM unit by the request from System Manager unit.

※PSM : Power Supply Manager

Power Service

#3 Malfunction detection

Detect the all types of the malfunctions.

1)Heart Beat

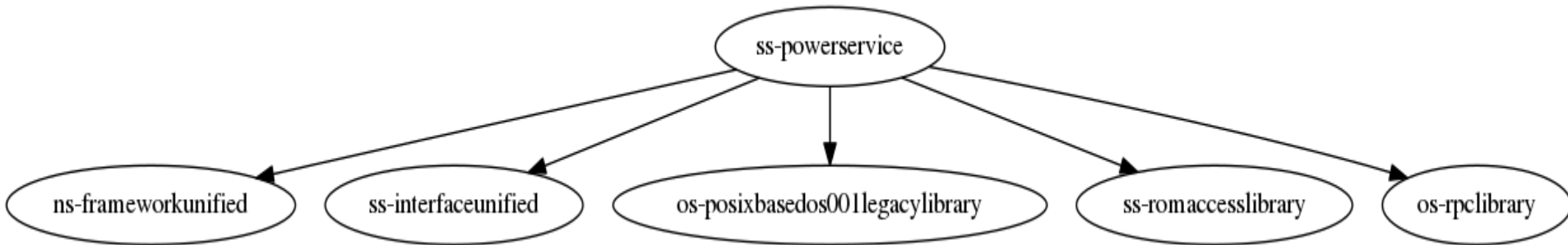
Request System Manager to monitor the malfunctions (Heartbeat) by the request from PSM unit.

2)Service abnormal termination

Request the system reset to PSM unit by the malfunction notification from System Manager unit.

Power Service

■ Unit relation diagram



SS-.... : System Service
NS-.... : Native Service Framework
VS-.... : Vehicle Service
PS-.... : Peripheral Service
OS-.... : Other Service

Power Service

■ Use case

[Main]

- Main (Start sequence)

[Control]

- System start-up :

Notify the start-up process completion to PSM unit by the request from System Manager unit.

- System shutdown :

Notify the shutdown process completion to PSM unit by the request from System Manager unit.

[Notify]

- Malfunction detection (Heart Beat) :Request System Manager to monitor the malfunctions (Heartbeat) by the request from PSM unit.

- Malfunction detection (Service abnormal termination) :Request the system reset to PSM unit by the malfunction notification from System Manager unit

[Service]

- Start the session

- Terminate the session

- Request the system mode information

- Receipt of power level notification :Notify the power value and the power status to NPP Service

- Receipt of shutdown POPUP request :Notify the shutdown POPUP request and POPUP event type to NPP Service

- Receipt of Crank status notification :Receive Crank status notification

Logger Service

■ Functional Overview

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

#1 request the log storing

At the receipt of the storing request read the log generated by other modules and save it.

#2 periodically read

Periodically read the log written by NS Logger.

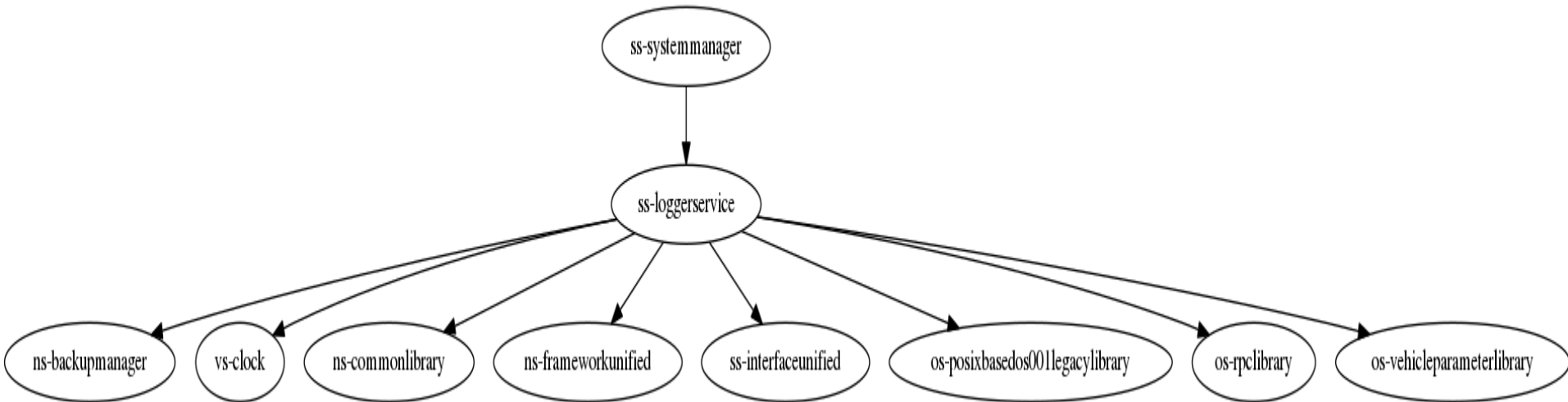
#3 output log to the network

Output the log to the network by UDP.

※NS : Native Service Framework

Logger Service

■ Unit relation diagram



loggerservice.png

SS-... : System Service
NS-... : Native Service Framework
VS-... : Vehicle Service
PS-... : Peripheral Service
OS-... : Other Service

Logger Service

■ Use case

[Main]

- Main (Start sequence)
- Main (Stop sequence)

[Control]

- Session start for the logging
- Session termination for the logging

[Service]

- Callback against the each notifications from IPC
- Event log and emergency error log output request to USB/SD
- Read and log reset request of the event statistical log
- CAN logging start request

[Event Data Send]

- Event data sending to Logger Service (Request notification to clear the event log)
- Event data sending to Logger Service (CAN current date and time sending)
- Event data sending to Logger Service (Vehicle identification number (VIN) sending)
- Event data sending to Logger Service (MOST shutdown completion event sending)
- Event data sending to Logger Service (Event notification to persist the event log)
- Event data sending to Logger Service (Information sending of the screen shot event response)
- Event data sending to Logger Service (Diagnosis ID sending)
- Event data sending to Logger Service (Logging propriety information sending for the selected device)
- Event data sending to Logger Service (Logging propriety information sending for UDP)
- Event data sending to Logger Service (Request notification to upload the event log)
- Event data sending to Logger Service (CAN diagnosis status data sending))

Task Manager

■ Functional Overview

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

Task Manager has the following functions;

#1 Service start and termination

Start and terminate the non-resident service.

#2 Malfunction detections

Detect the abnormal behaviors of the controlled services.

1) Hang-up detections

Task Manager monitors the running status of the started service and forces the corresponding service to terminate at the hang-up detection.

Task Manager

2)Service abnormal termination detections

When the service is abnormally terminated, it is assumed that the application using the applicable service detects this malfunction and executes the failure processing.

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

3)Malfunctions at the service start

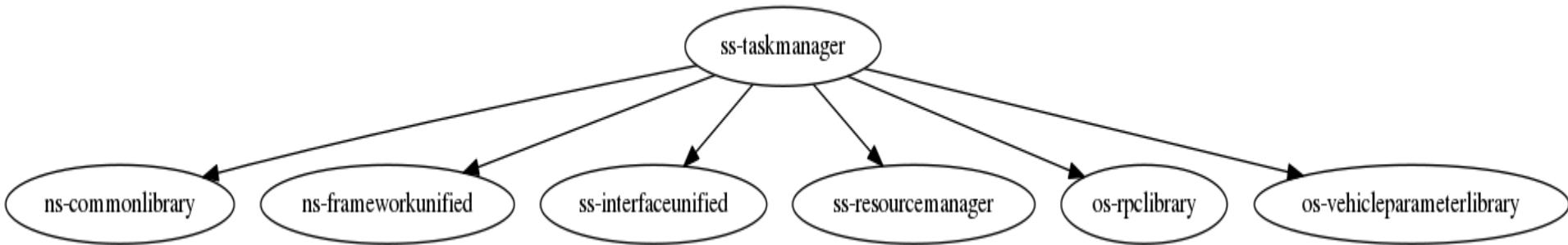
When the start completion is not normally responded within the specified time after the service start, Task Manager recognizes the malfunctions and forces the applicable service to terminate

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.

Task Manager

■ Unit relation diagram



SS-···· : System Service
NS-···· : Native Service Framework
VS-···· : Vehicle Service
PS-···· : Peripheral Service
OS-···· : Other Service

Task Manager

■ Use case

[Main]

- Main (Start sequence)
- Main (Stop sequence)

[Task Manager]

- Data initialization before non-resident service start
- Non-resident service start process by INI_Init()
- Non-resident service termination process by INI_Init()
- Non-resident service start process by INI_Main()
- Non-resident service termination process by INI_Main()
- Service abnormal termination detection
- Start order maintenance
- Non-resident service start process
- Notification of residual system memory decrease
- Resource release at the normal reboot
- Log storage at the malfunction occurrence
- Acquire the specified service information

[primary]

- Start/Termination event completion notification
- Service timeout setting
- Started service information acquisition
- Started service expanded information acquisition
- Acquire the private data

Resource Manager

■ Functional Overview

This service provides the function to provide the abnormal status notifying function in order to execute the failsafe process at the malfunction detections.

Resource Manager provides the following functions;

#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

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

Resource Manager

#3 System memory monitoring

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

#4 Watch Dog Timer (WDT) Update

Monitor 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

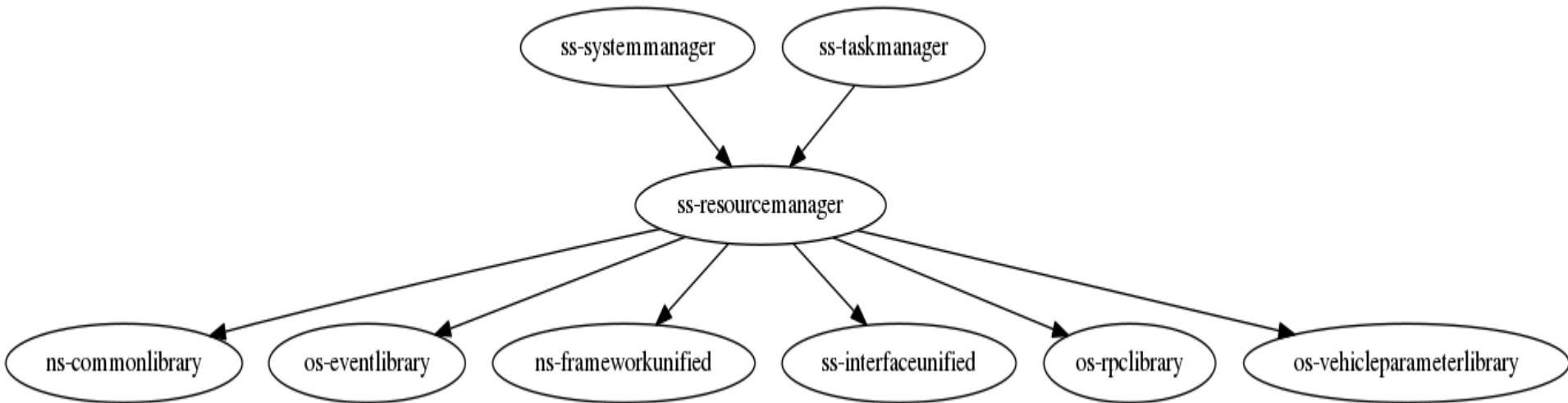
Provide the residual memory information, CPU load information and residual memory information for CMA for the debug display.

Resource Manager

#6 Log output of the minimum residual memory information at ACC-OFF
Maintain the minimum residual memory information after the start and
output the log at ACC-OFF detection.

Resource Manager

■ Unit relation diagram



SS-···· : System Service
NS-···· : Native Service Framework
VS-···· : Vehicle Service
PS-···· : Peripheral Service
OS-···· : Other Service

Resource Manager

■ Use case

[Main]

- Main (Start sequence)

[Control]

- Initial setting sequence
- System memory monitoring sequence
- Event handler start sequence

Rom Access Library

■ Functional Overview

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

#1 libssaccess

libssaccess provides below functions according to specified norflash device pathname and block ID.

- (1)read data from specified main or backup block
- (2)write data to specified main and backup blocks

#2 libSS_RomAccessIf

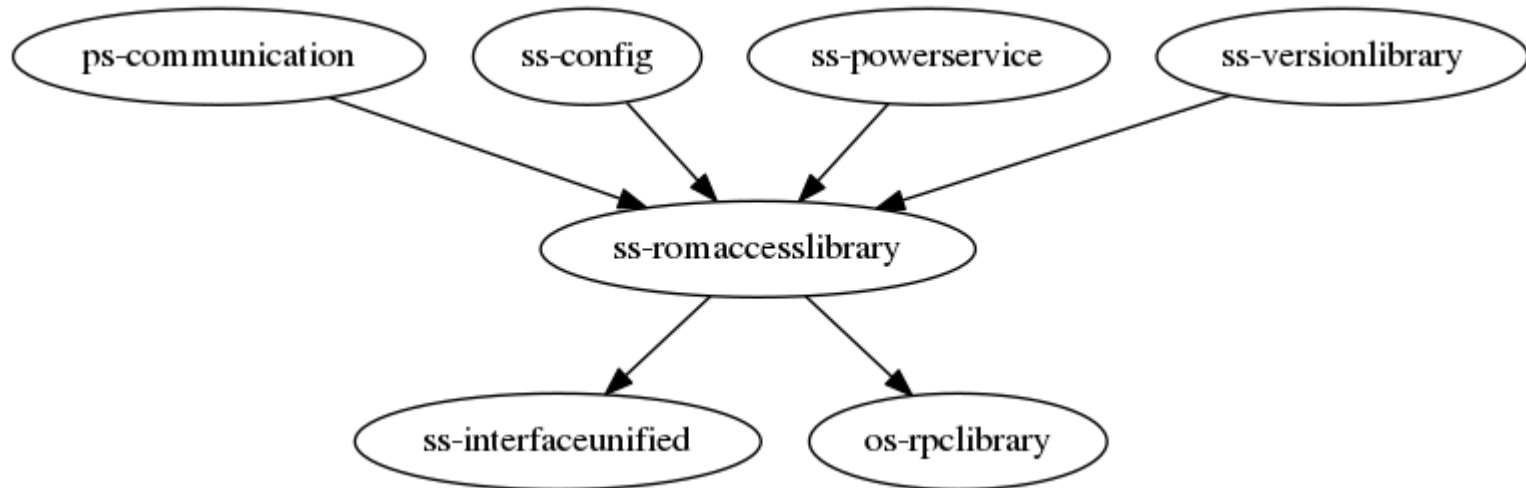
libSS_RomAccessIf provide functions of Read/Write data to non-volatile area according to defined ID. ID belong to category. Category has follow properties.

- Boot area, RAM area, ROM area

※block ID: ID that identifies the data storage location

Rom Access Library

■ Unit relation diagram



SS-.... : System Service
NS-.... : Native Service Framework
VS-.... : Vehicle Service
PS-.... : Peripheral Service
OS-.... : Other Service

Rom Access Library

■ Use case

[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
- init rom access

[libSS_RomAccessIf]

- read data
- write data

Config

■ Functional Overview

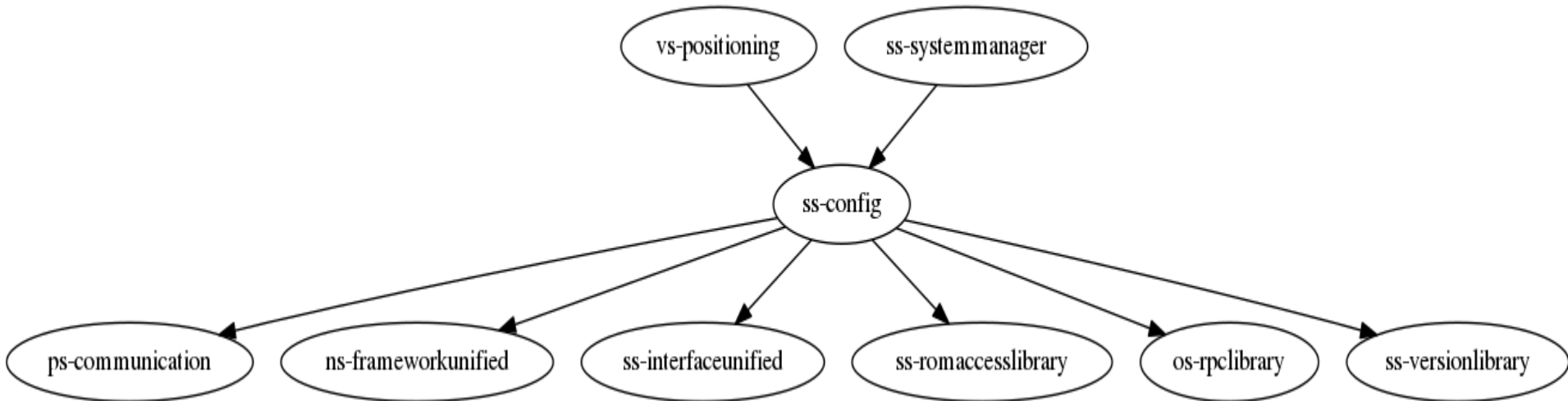
This service provides register services that are not collected by OOM Killer, register Group Re-launch services, and set environment variables

■ Use case

System environment initialize.

Config

■ Unit relation diagram



SS-···· : System Service
NS-···· : Native Service Framework
VS-···· : Vehicle Service
PS-···· : Peripheral Service
OS-···· : Other Service

Interface unified

■ Functional Overview

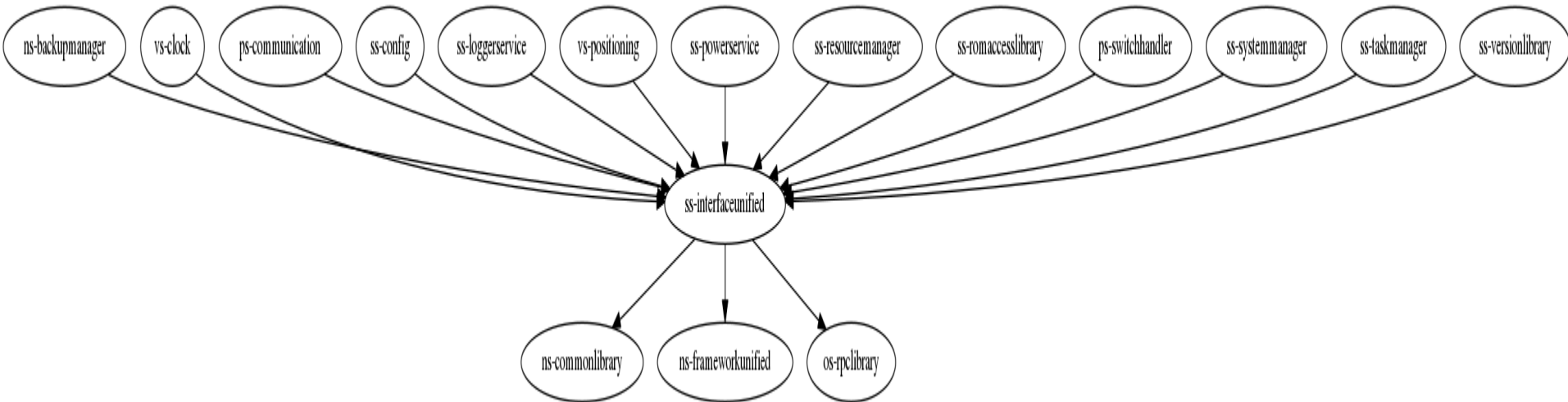
This service provides the function for timer and the library of system manager.

#1 Timer

#2 System startup, system termination, error detection, and error log output functions in System Manager

Interface unified

■ Unit relation diagram



interfaceunified.png

SS-... : System Service
NS-... : Native Service Framework
VS-... : Vehicle Service
PS-... : Peripheral Service
OS-... : Other Service

Interface unified

■ Use case

[Timer]

- One-Shot Timer registration and termination (without Start() parameter)
- One-Shot Timer registration and termination (with Start() parameter)

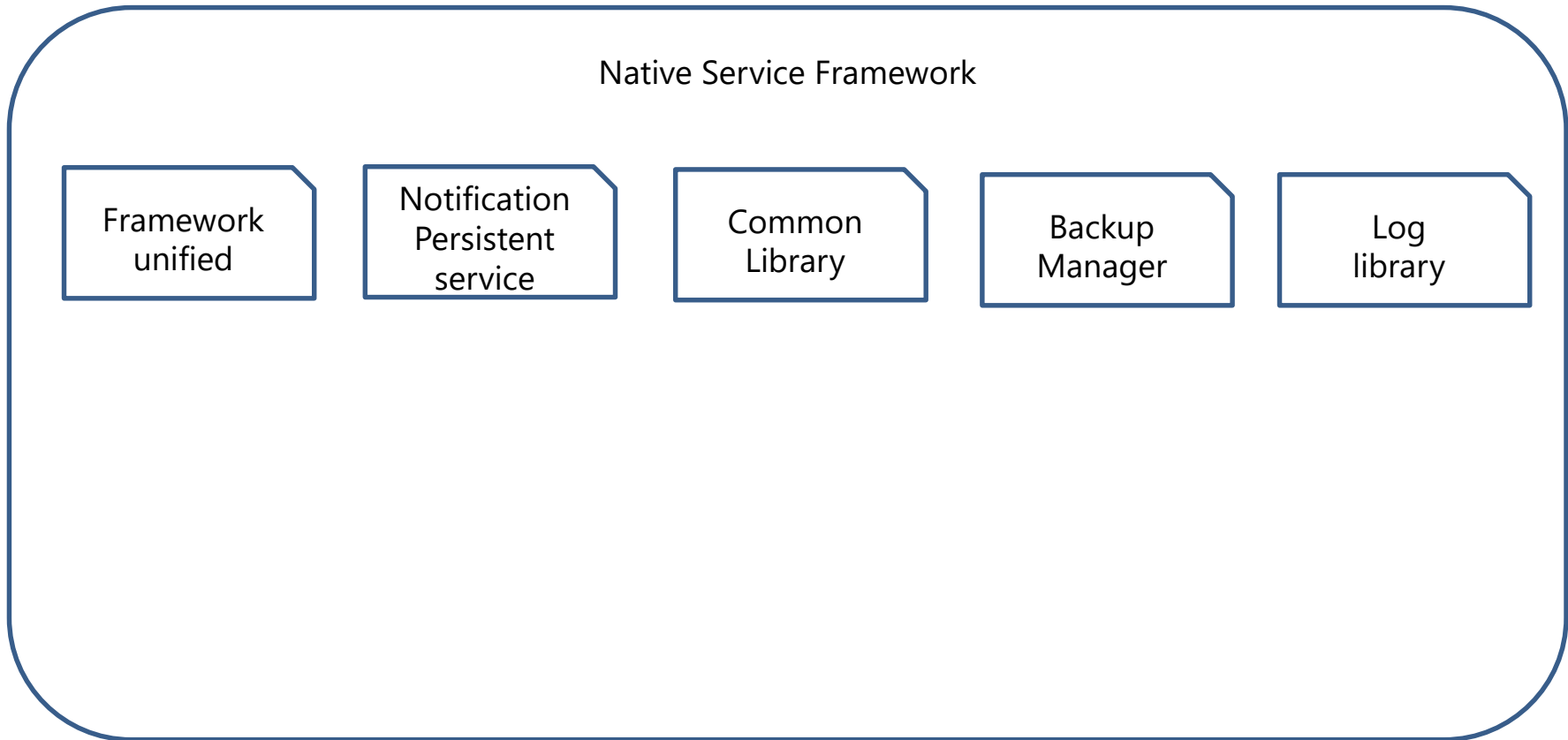
[System IF]

- Service Start
- Service Stop
- Dump output for debug(Abnormal detection is included)
- Log storing in the internal non-volatile memory
- Copying 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
- Recording of the mode at the next boot to the non-volatile memory
- Recording of the data reset mode at the next boot to non-volatile memory
- Whole board reset
- Registration of the callback function invoked in the receiving of System Manager session open ACK
- Stored log clearance (deletion) in the internal non-volatile memory
- Program updated status record in the non-volatile information
- Write request to the specified field of the shared memory with Boot Loader
- Expanded information acquisition of the start and shutdown parameter
- Storing of the start order
- Dump data contents writing in OnDebugDumpResponseReceived sending

Native Service Framework

■ Overview

Native service is service group that provide the following functions



Framework unified

■ Functional Overview

This service provides the following functions.

#1 HANDLE: Provides a type used to hide the internal structure of the instance.

#2 Dispatcher: Provides a mechanism for receiving messages and registering callback functions. If a service wants to receive message from other module, dispatcher is in need

#3 One-to-one inter-process communication: Provides a mechanism to communicate with any process.

#4 One-to-many inter-process communication: Provides a mechanism to communicate with many processes. Execute one-to-one process communication by service name, command and session.

Framework unified

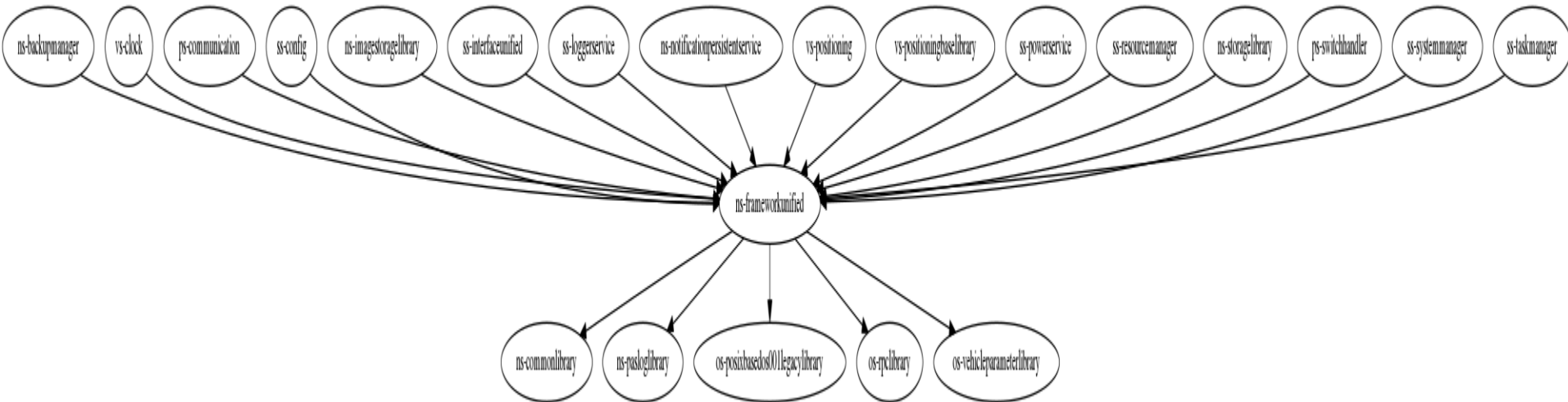
#5 Session: Provides a mechanism to establish a session (communication path) with the service you want to communicate with.

#6 Abnormal Monitor: Provides a function to monitor abnormalities

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

Framework unified

■ Unit relation diagram



frameworkunified.png

SS-... : System Service
NS-... : Native Service Framework
VS-... : Vehicle Service
PS-... : Peripheral Service
OS-... : Other Service

Framework unified

■ Use case

- Service startup
- Event subscription registration
- Unsubscribe event
- Service usage status
- Set / get application data
- Defer Message
- Session open / close
- Session event notification
- Message queue open / close
- Synchronous communication
- Asynchronous communication
- Response data transmission
- Message data reception
- Zero copy communication
- Non-blocking message reception
- Received message discard
- Start sub thread(no loop)
- Start sub thread has loop (sub thread communication)
- Timer(callback)
- Timer(API)
- Timer(NSTimer class)
- Ring Buffer read / write
- Reading / writing Configuration File
- Initialize
- Log output
- Force clear
- Log output control word (setting / getting)
- Log file operation
- Log Level (setting / getting)
- Change to real-time log output
- Send log event
- Mutex
- RWLock
- Version
- HSMEvent
- Session management(data pool)

Notification Persistent Service

■ Functional Overview

This service provides the following functions.

#1 Notification function

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

#2 Persistent function

- Release

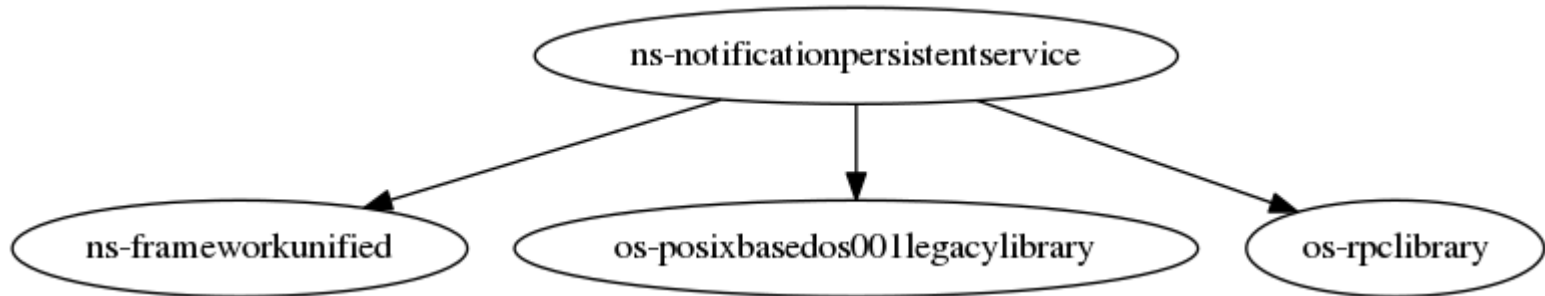
Copies the file on the volatile memory file system specified by the client to the file on the nonvolatile memory file system.

- Load

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

Notification Persistent Service

■ Unit relation diagram



SS-.... : System Service
NS-.... : Native Service Framework
VS-.... : Vehicle Service
PS-.... : Peripheral Service
OS-.... : Other Service

Notification Persistent Service

■ Use case

#1 Notification function

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

#2 Persistent function

- 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
- Sync persistent data
- Clear persisted data
- Release Persistent Data
- Set Persistent Notification Type

#Shutdown

- Shut-Down
- Get NPPStatus

Common Library

■ Functional Overview

This service provides common processing of process management function.

The main usage procedures for the process management process of this function are shown 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.

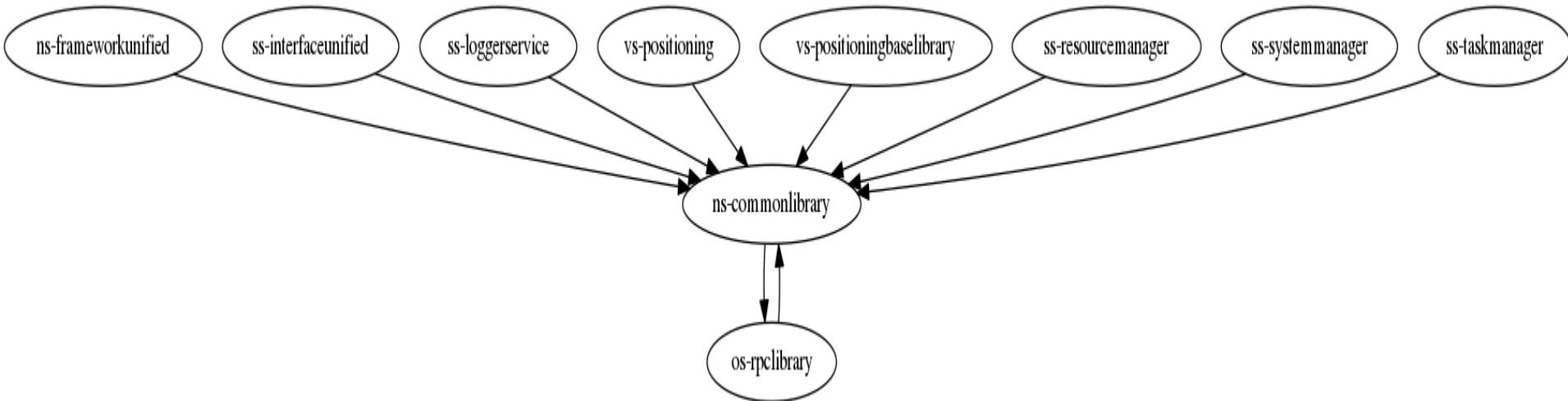
#6 Register the created process in a group.

#7 Use mutex and semaphore for exclusion between processes.

#8 Create a thread.

Common Library

■ Unit relation diagram



commonlibrary.png

SS-···· : System Service
NS-···· : Native Service Framework
VS-···· : Vehicle Service
PS-···· : Peripheral Service
OS-···· : Other Service

Common Library

■ Use case

- Setting process attribute settings
- Creating process
- Grouping process
- Exclusive control by mutex
- Creating thread
- Exclusive control by semaphore
- Maintain the state of each process
- Control the identification information of the created process
- Thread

Backup Manager

■ Functional Overview

This service provides functions of Read/Write nonvolatile data according to defined ID. ID belong to category. Category has follow properties.

#1 Use NAND

#2 Use Cache DRAM

#3 Use Backup DRAM

#4 Real-time write

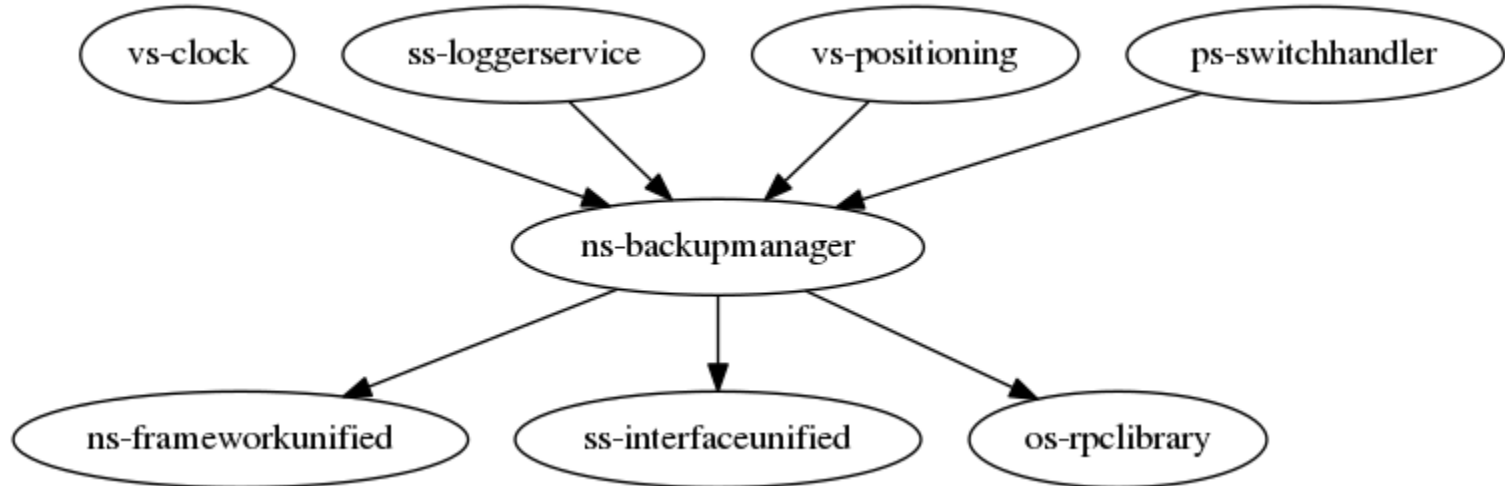
#5 Backup cycle

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.

Backup Manager

■ Unit relation diagram



SS-···· : System Service
NS-···· : Native Service Framework
VS-···· : Vehicle Service
PS-···· : Peripheral Service
OS-···· : Other Service

Backup Manager

■ Use case

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

Log Library

- Functional Overview

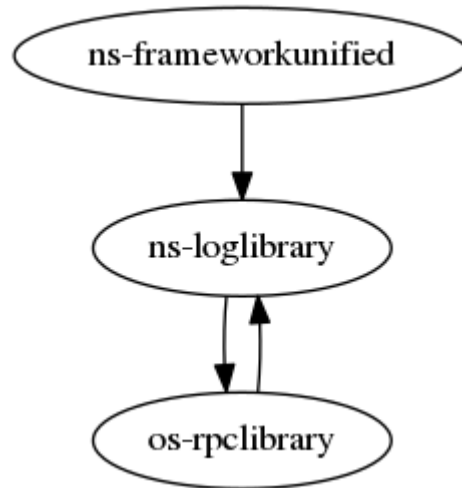
This service provides the log freeze function.

- Use case

Log freeze

Log Library

■ Unit relation diagram

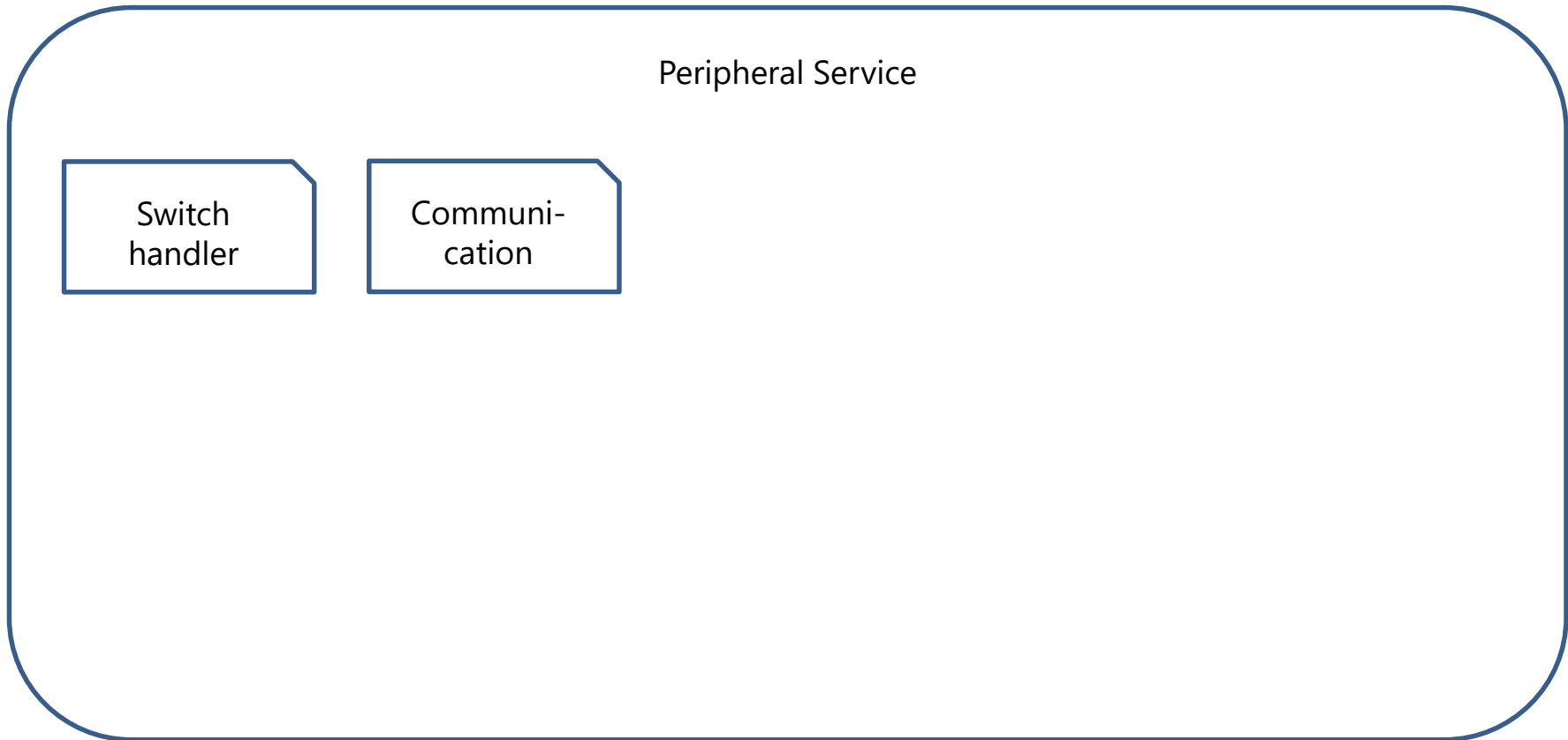


SS-···· : System Service
NS-···· : Native Service Framework
VS-···· : Vehicle Service
PS-···· : Peripheral Service
OS-···· : Other Service

Peripheral Service

■ Overview

Peripheral service is service group that provide the following functions



Switch handler

■ Functional Overview

This service provides the functions of switch handler regarding Touch Panel, Key Listener, Key Switcher.

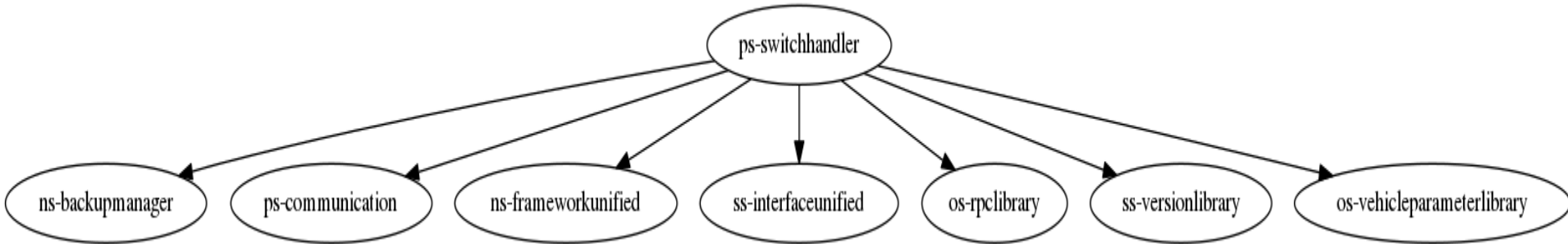
#1 The API for Touch Panel device control

#2 Key Listener API for key delivery register/release, state acquisition

#3 API for Key Switcher

Switch handler

■ Unit relation diagram



switchhandler.png

SS-···· : System Service
NS-···· : Native Service Framework
VS-···· : Vehicle Service
PS-···· : Peripheral Service
OS-···· : Other Service

Switch handler

■ Use case

#1 The API for Touch Panel device control

- Execute electrostatic touch panel Factory inspection mode.
- Acquiring sensitivity level of electrostatic touch panel.
- Setting sensitivity level of electrostatic touch panel.
- Setting radio information being received in electrostatic touch panel.

#2 Key Listener API for key delivery register/release, state acquisition

- Delivery registration of the key.
- Delivery un-registration of the key.
- Get the key status.

#3 API for Key Switcher

- Initialization of the key publisher.
- Status notification request of the key switch.
- Lock control request of the key switch.
- Check whether touch function is valid or not.

Communication

■ Functional Overview

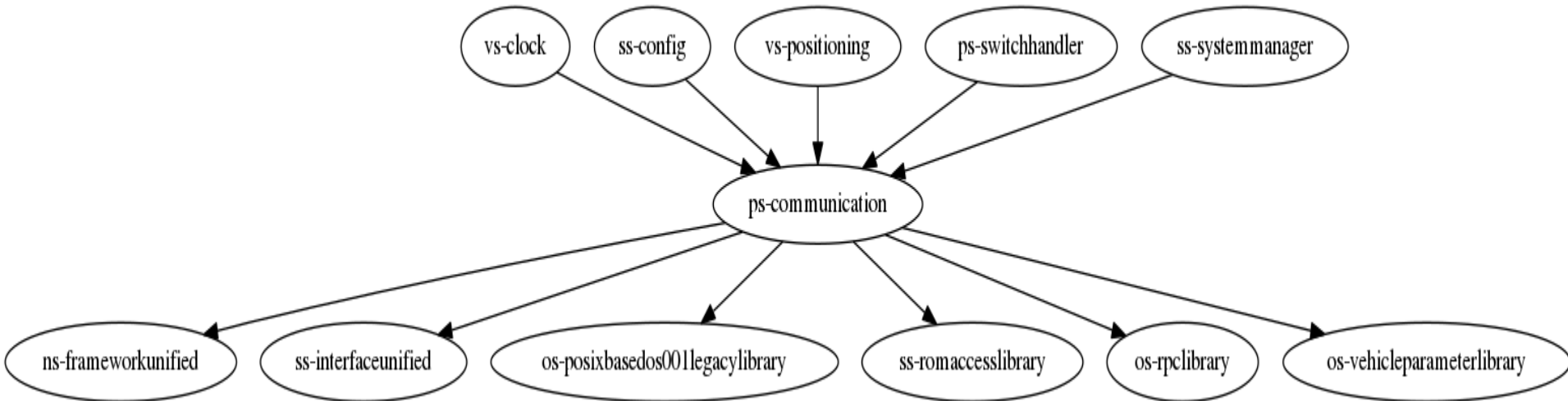
This service provides the following communication functions.

#1 CAN

Support the CAN data communication function by CAN HAL.

Communication

■ Unit relation diagram



communication.png

SS-... : System Service
NS-... : Native Service Framework
VS-... : Vehicle Service
PS-... : Peripheral Service
OS-... : Other Service

Communication

■ Use case

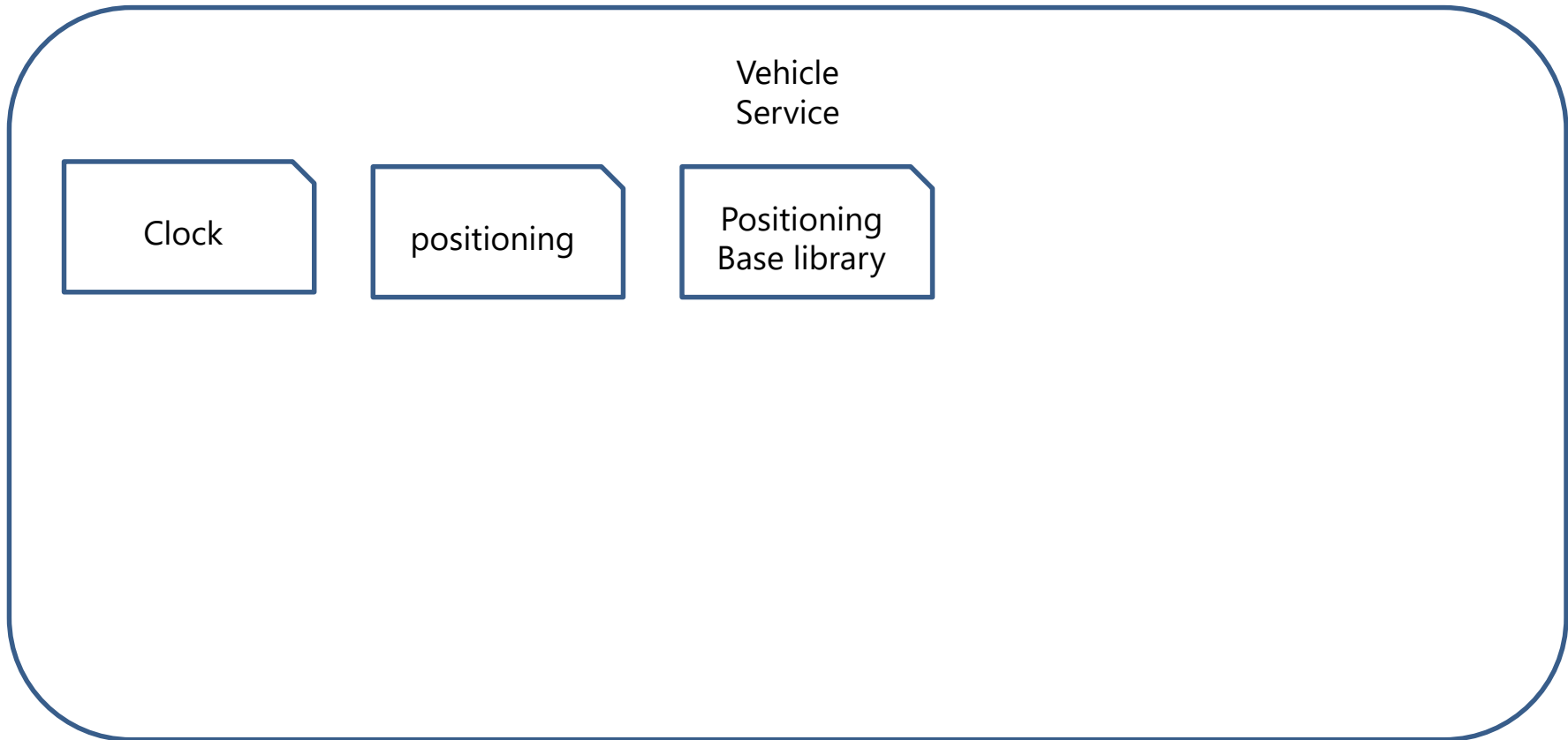
#1 CAN

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

Vehicle Service

■ Overview

Vehicle service is service group that provide the following functions



Clock

■ Functional Overview

This service provides two functions (system time function and display time function).

#1 system time function

The system time function provides methods to get/set time and methods to calculate/convert time for application.

#2 display time function

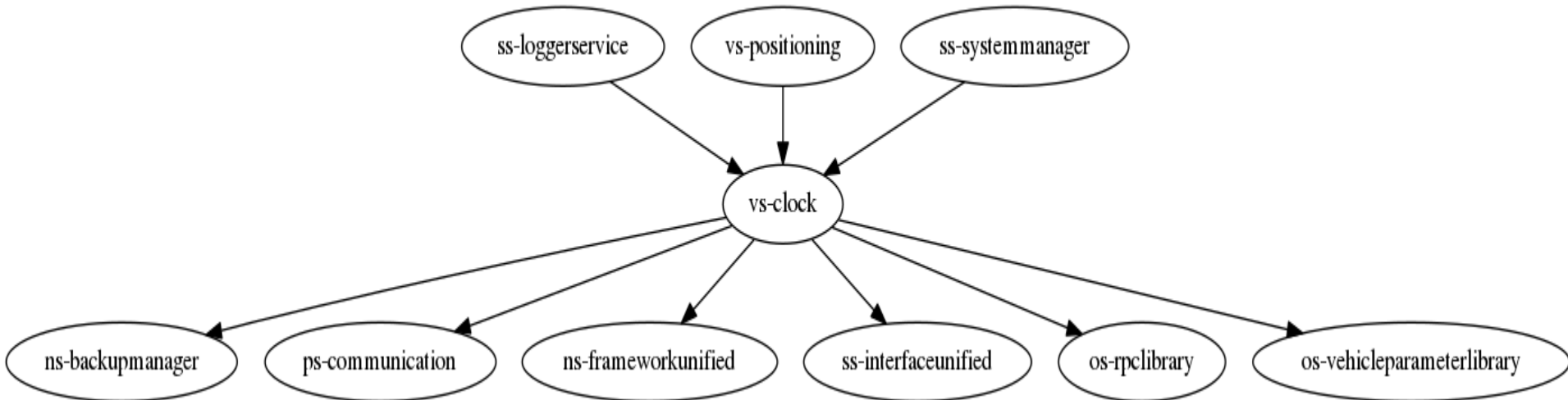
The display time function provides functions described as follows:

Calculate display time and send it to application. #Hide system time for application.

Set UTC time and local time to OS. Third-party software or OSS can get it.

Clock

■ Unit relation diagram



clock.png

SS-... : System Service
NS-... : Native Service Framework
VS-... : Vehicle Service
PS-... : Peripheral Service
OS-... : Other Service

Clock

■ Use case

- Get the system time
- Get the local time
- Convert seconds to date and time
- Convert date and time to seconds
- Set the display time when GPS synchronization setting is off
- Get the display time when GPS synchronization setting is off
- Adjust the display time when GPS synchronization setting is off.
- Get the system time(2038 issue fixed)
- Get the local time(2038 issue fixed)
- Convert seconds to date and time(2038 issue fixed)
- Convert date and time to seconds(2038 issue fixed)
- Load the display setting
- Get the display setting
- Set the GPS time to clock
- Register the delivery of display time notification
- Unregister the delivery of display time notification
- Set the GPS auto adjustment
- Adjust the time(increase 1 hour to time)
- Adjust the time(decrease 1 hour from time)
- Adjust the time(increase 1 minute to time)
- Adjust the time(decrease 1 minute from time)
- Adjust the time(set the minute to 0)
- Set the AM/PM
- Set the display format of 12h/24h
- Set the Time Zone
- Set the DST
- Correct the time information
- Notify SW OFF information

Positioning

■ Functional Overview

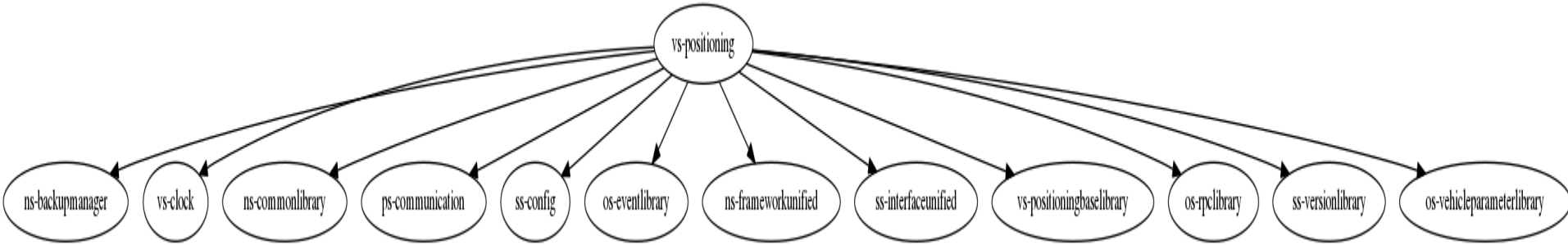
This service provides GPS and Sensor functions.

#1 GPS

#2 Sensor functions

Positioning

■ Unit relation diagram



positioning.png

SS-···· : System Service
NS-···· : Native Service Framework
VS-···· : Vehicle Service
PS-···· : Peripheral Service
OS-···· : Other Service

Positioning

■ Use case

- 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
- 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

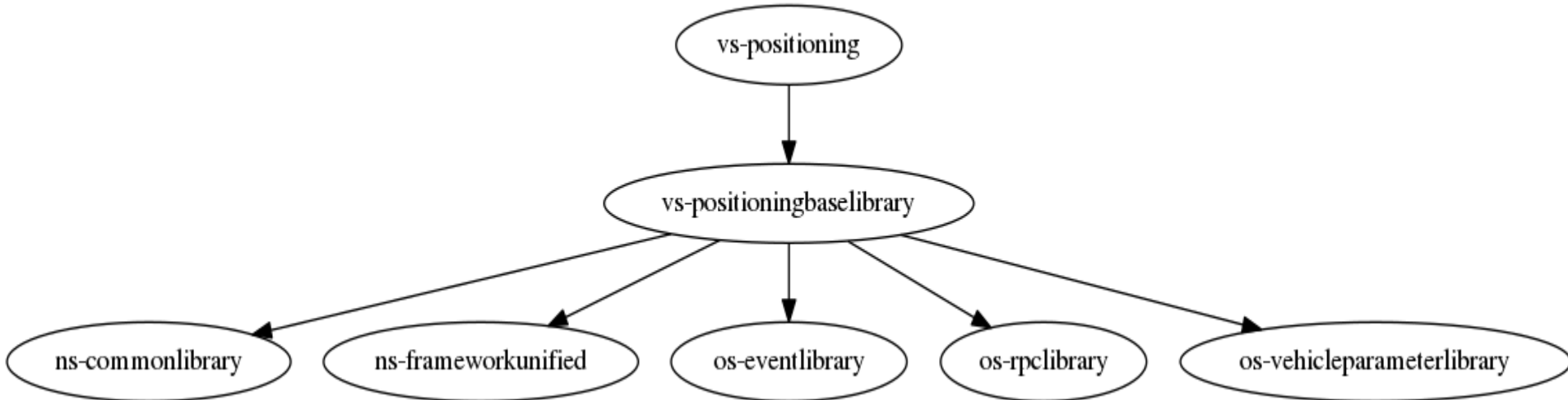
Positioning Base Library

■ Functional Overview

This service provides the base function for positioning.

Positioning Base Library

■ Unit relation diagram



SS-···· : System Service
NS-···· : Native Service Framework
VS-···· : Vehicle Service
PS-···· : Peripheral Service
OS-···· : Other Service

Positioning Base Library

■ Use case

[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)

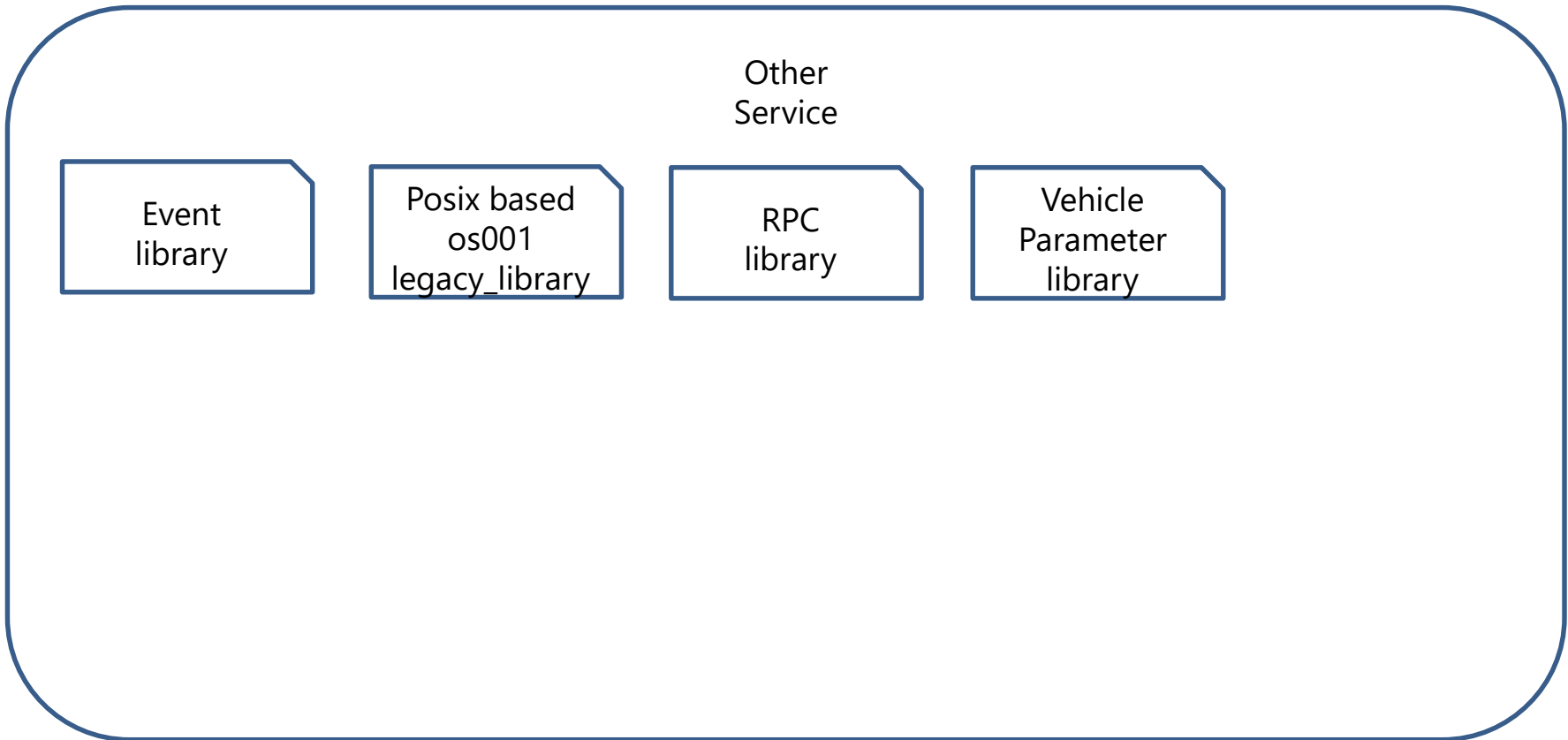
[Handle]

- Get application handle
- Set application handle

Other Service

■ Overview

Other service is service group that provide the following functions



Event library

■ Functional Overview

This service provides the sending and receiving of message between process, thread, according to the transfer data, there are two kind function as below:

#1 Event's data 's send and receive

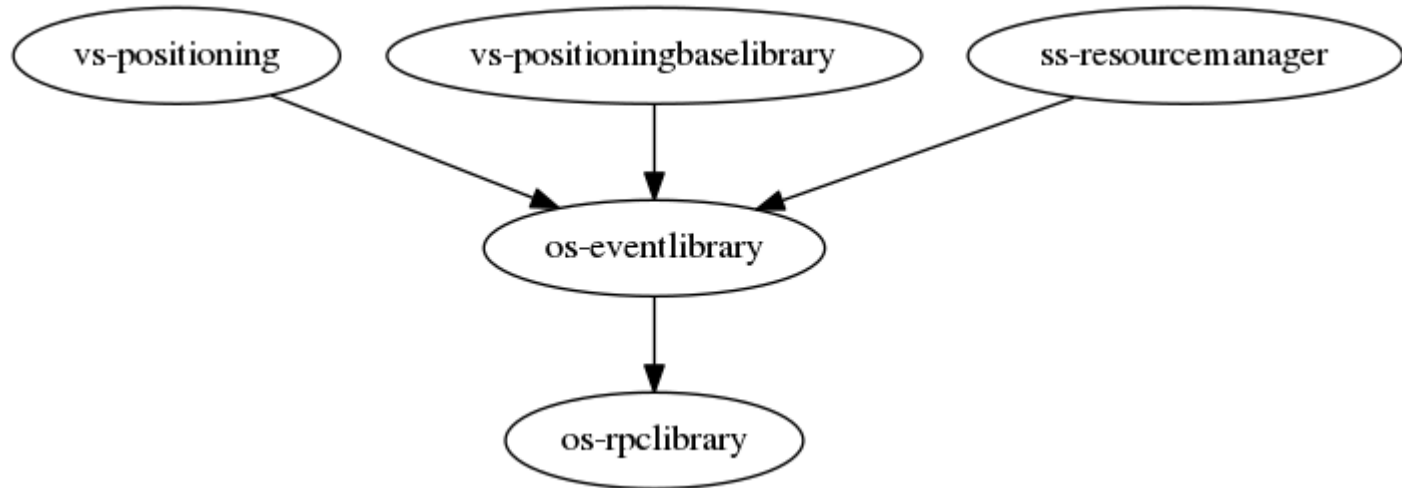
Communicate using fixed length bit string (flag)

#2 Message's data send and receive

Communicate using a variable-length record byte string (message)

Event library

■ Unit relation diagram



SS-···· : System Service
NS-···· : Native Service Framework
VS-···· : Vehicle Service
PS-···· : Peripheral Service
OS-···· : Other Service

Event library

■ Use case

- 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

Posix based os001 legacy library

■ Functional Overview

This service provides support function of system call.

#1 Get the number of clock cycles

Get the number of clock cycles, time unit : microsecond.

#2 Delay thread for the specified time

Delay thread for the specified time, time unit : millisecond.

#3 String handling

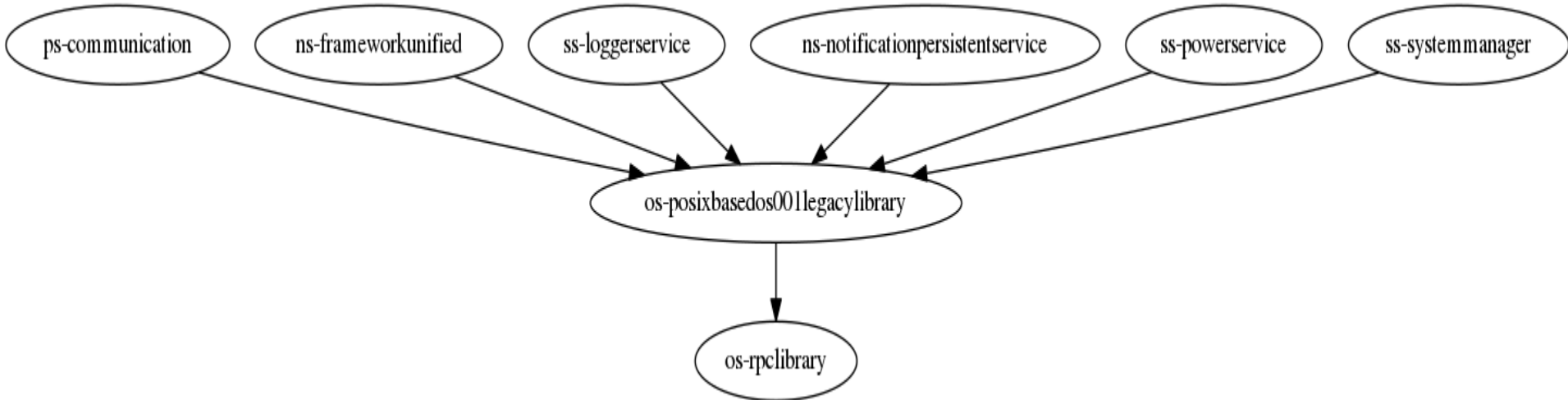
itoa / ultoa / strcpy / strcat

■ Use case

- Get the number of clock cycles
- Delay thread for the specified time
- Convert integer value to null terminated string
- Convert unsigned long value to null terminated string
- String copy: copy the specified size of the source string to the destination string
- Concatenate string : concatenate the specified size of the source string to the end of the destination string

Posix based os001 legacy library

■ Unit relation diagram



posixbasedos001legacylibrary.png

- SS-.... : System Service
- NS-.... : Native Service Framework
- VS-.... : Vehicle Service
- PS-.... : Peripheral Service
- OS-.... : Other Service

RPC library

■ Functional Overview

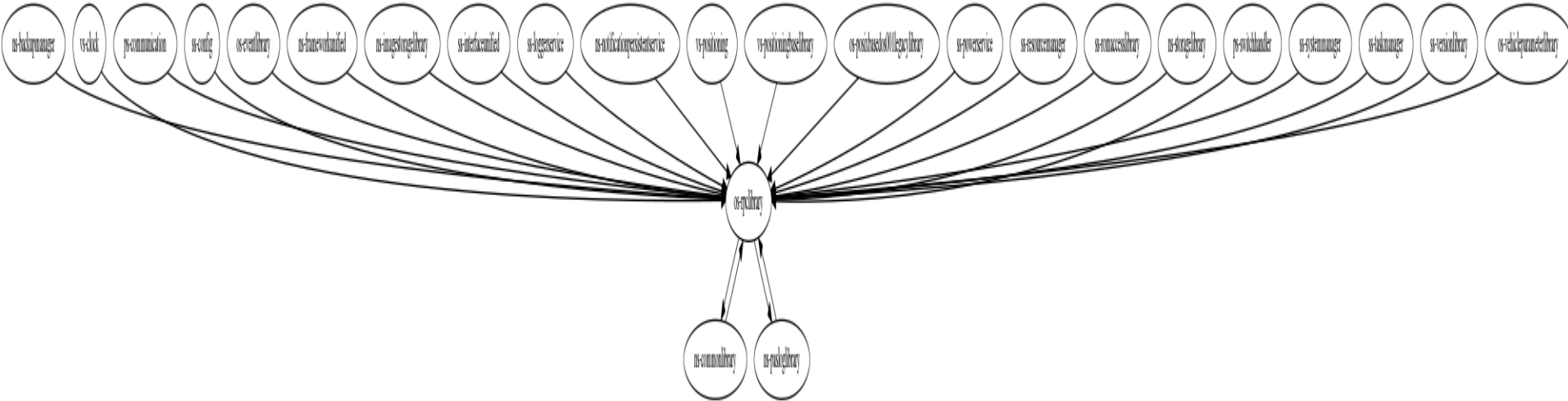
This service provides the function that allows a subroutine or procedure in another address space to be executed from a program.

■ Use case

- 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
- RPC library end
- Force to clean up when process is over

Rpc library

■ Unit relation diagram



rpclibrary.png

SS-.... : System Service
NS-.... : Native Service Framework
VS-.... : Vehicle Service
PS-.... : Peripheral Service
OS-.... : Other Service

Vehicle Parameter library

■ Functional Overview

This service provides an API library which is used to get environment variable.

#1 Get vehicle parameter environment variable

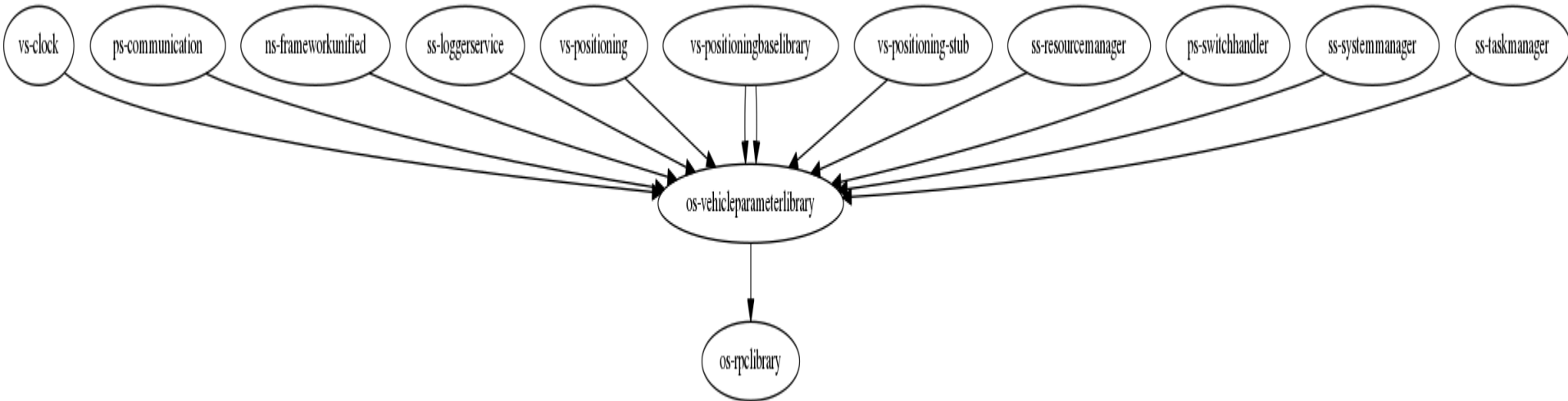
#2 Get function check result

■ Use case

- Get environment variable
- Get function check result

Vehicle Parameter library

■ Unit relation diagram



vehicleparameterlibrary.png

SS-···· : System Service
NS-···· : Native Service Framework
VS-···· : Vehicle Service
PS-···· : Peripheral Service
OS-···· : Other Service