

# Hardware Abstraction Layer (HAL)

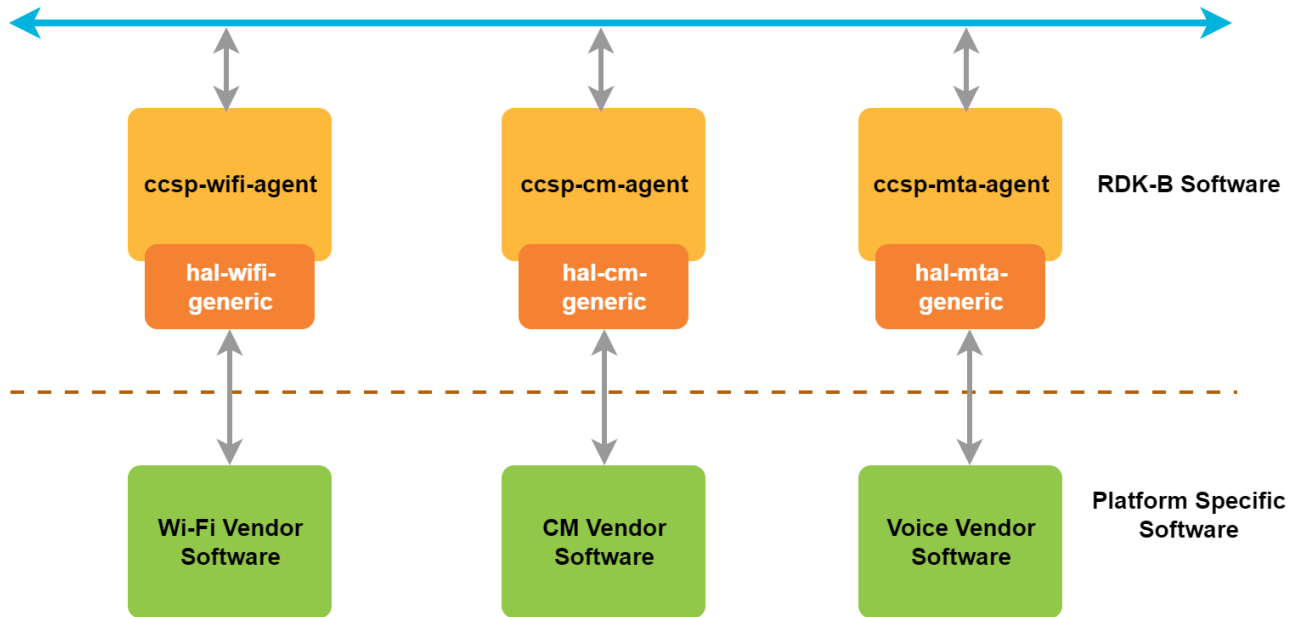
## Introduction

RDK-B components are designed to avoid platform or silicon dependencies. Hardware Abstraction Layer (HAL) defines a standard interface for hardware vendors to implement. The HAL layer abstracts the underlying hardware like MOCA, Wi-Fi, etc. through a standard set of APIs defined as part of RDK-B HAL for the respective components. This HAL layer is implemented per platform and the rest of the components can be compiled to run on the new platform without major modifications.

The [HAL](#) in RDK-B Architecture section gives an overview of CCSP framework's Hardware Abstraction Layer.

HAL can be common-HAL or component-specific-HAL. Components may define a component specific HAL to hardware drivers, that are only used by that component

### CCSP Message Bus



## Component Specific HAL

- HAL APIs will be available in the CMF repo path: `../rdkb/components/opensource/ccsp/hal/source/`
- PandM HAL Integration (back-end) Layer is also known as component specific HAL.
- This layer makes call to underlying Linux system calls/commands, third party modules, open source modules and other CCSP components to execute the requests.
- This layer will be more component specific and will be providing APIs to CCSP so as to manage a particular hardware module of the system.
- Following are some of the component specific HALs available in `../rdkb/components/opensource/ccsp/hal/source/` path.
  - Wifi
  - MoCA
  - MTA Agent
  - CM
  - Ethernet Switch
  - DHCPv4C
  - Virtual LAN
  - Firewall
  - DPoE
  - Bluetooth
  - MSO\_Management
  - Voice
  - WAN
  - [TR69\\_TLV](#)

## Wi-Fi HAL

All HAL functions prototypes and structure definitions are available in `wifi_hal.h` file.

- Wi-fi HAL is used for the RDK-Broadband Wifi radio hardware abstraction layer.
- Latest version of RDKB supports 300+ Wi-Fi HAL API's.
- Based on how Wi-Fi vendor exposes their driver capabilities in user space, the HAL API's can be implemented in `wifi_hal.c`
- Some of the APIs are :

1. wifi\_getRadioChannelStats
2. wifi\_getRadioChannelStats2
3. wifi\_getApAssociatedDeviceRxStatsResult
4. wifi\_getApAssociatedDeviceTxStatsResult
5. wifi\_getApAssociatedDeviceTidStatsResult
6. wifi\_getApAssociatedDeviceStats
7. wifi\_getHalVersion
8. wifi\_factoryReset
9. wifi\_factoryResetRadios
10. wifi\_factoryResetRadio
11. wifi\_setLED
12. wifi\_init
13. wifi\_reset
14. wifi\_down
15. wifi\_createInitialConfigFiles
16. wifi\_getRadioCountryCode
17. wifi\_setRadioCountryCode
18. wifi\_pushCountryCode
19. wifi\_getATMCApable
20. wifi\_setATMCApable

- To see the API specification of Wi-Fi HAL please refer - [Wi-Fi HAL APIs](#)

## MOCA HAL

All HAL functions prototypes and structure definitions are available in moca\_hal.h file.

- MoCA HAL is used for the RDK-Broadband MoCA hardware abstraction layer.
- An abstraction layer, mainly for interacting with MoCA driver.
- The APIs are :
  1. moca\_GetIfConfig
  2. moca\_SetIfConfig
  3. moca\_IfGetDynamicInfo
  4. moca\_IfGetStaticInfo
  5. moca\_IfGetStats
  6. moca\_GetNumAssociatedDevices
  7. moca\_IfGetExtCounter
  8. moca\_IfGetExtAggrCounter
  9. moca\_GetMocaCPEs
  10. moca\_GetAssociatedDevices
  11. moca\_FreqMaskToValue
  12. moca\_HardwareEquipped
  13. moca\_GetFullMeshRates
  14. moca\_GetFlowStatistics
  15. moca\_GetResetCount
  16. moca\_setIfAcaConfig
  17. moca\_getIfAcaConfig
  18. moca\_cancelIfAca
  19. moca\_getIfAcaStatus
  20. moca\_getIfScmod
- To see the API specification of MoCA HAL please refer - [MoCA HAL APIs](#)

## MTA HAL

All HAL functions prototypes and structure definitions are available in mta\_hal.h file. An MTA can deliver Home Phone service in addition to High Speed Internet.

- MTA HAL used for the RDK-Broadband hardware abstraction layer for Cable Modem.
- An abstraction layer, implemented to interact with MTA device.
- mta\_hal.c file provides the function call prototypes and structure definitions used for the MTA hardware abstraction layer.
- Some of the APIs are :
  1. mta\_hal\_InitDB
  2. mta\_hal\_GetDHCPInfo
  3. mta\_hal\_LineTableGetNumberOfEntries
  4. mta\_hal\_LineTableGetEntry
  5. mta\_hal\_TriggerDiagnostics
  6. mta\_hal\_GetServiceFlow
  7. mta\_hal\_DectGetEnable
  8. mta\_hal\_DectSetEnable.
  9. mta\_hal\_DectGetRegistrationMode
  10. mta\_hal\_DectSetRegistrationMode
  11. mta\_hal\_DectDeregisterDectHandset
  12. mta\_hal\_GetCalls
  13. mta\_hal\_GetDect
  14. mta\_hal\_GetDectPIN
  15. mta\_hal\_SetDectPIN
  16. mta\_hal\_GetHandsets
  17. mta\_hal\_GetCALLP
  18. mta\_hal\_GetDSXLogs

- 19. mta\_hal\_GetDSXLogEnable
- 20. mta\_hal\_SetDSXLogEnable
- To see the API specification of MTA HAL please refer - [MTA HAL APIs](#)

## CM HAL

All HAL functions prototypes and structure definitions are available in cm\_hal.h file.

- CM HAL is used for the RDK-Broadband hardware abstraction layer for Cable Modem.
- It provides interface that cable modem software developers can use to interface to RDK-B.
- Some of the APIs are :
  1. cm\_hal\_InitDB
  2. docsis\_InitDS
  3. docsis\_InitUS
  4. docsis\_getCMStatus
  5. docsis\_GetDSChannel
  6. docsis\_GetUsStatus
  7. docsis\_GetUSChannel
  8. docsis\_GetDOCSISInfo
  9. docsis\_GetNumOfActiveTxChannels
  10. docsis\_GetNumOfActiveRxChannels
  11. docsis\_GetErrorCodewords
  12. docsis\_SetMddlpModeOverride
  13. docsis\_GetMddlpModeOverride
  14. docsis\_GetUSChannelId
  15. docsis\_SetUSChannelId
  16. docsis\_GetDownFreq
  17. docsis\_SetStartFreq
  18. docsis\_GetDocsisEventLogItems
  19. cm\_hal\_GetDHCPInfo
  20. cm\_hal\_GetCPEList
- To see the API specification of CM HAL please refer - [CM HAL APIs](#)

## Ethernet Switch HAL

All HAL functions prototypes and structure definitions are available in ccsp\_hal\_ethsw.h file.

- It provides implementation for Ethernet Switch Control.
- Based on how vendor exposes their driver capabilities in user space, the HAL API's can be implemented in hal-ethsw-generic/git/source/ethsw/ccsp\_hal\_ethsw.c
- The APIs are :
  1. CcspHalEthSwInit
  2. CcspHalEthSwGetPortStatus
  3. CcspHalEthSwGetPortCfg
  4. CcspHalEthSwSetPortCfg
  5. CcspHalEthSwGetPortAdminStatus
  6. CcspHalEthSwSetPortAdminStatus
  7. CcspHalEthSwSetAgingSpeed
  8. CcspHalEthSwLocatePortByMacAddress
  9. CcspHalExtSw\_getAssociatedDevice
  10. CcspHalExtSw\_ethAssociatedDevice\_callback\_register
  11. CcspHalExtSw\_getEthWanEnable
  12. CcspHalExtSw\_setEthWanEnable
  13. CcspHalExtSw\_getEthWanPort
  14. CcspHalExtSw\_setEthWanPort
  15. GWP\_RegisterEthWan\_Callback
  16. GWP\_GetEthWanLinkStatus
  17. GWP\_GetEthWanInterfaceName
- To see the API specification of Ethernet Switch HAL please refer - [Ethernet Switch HAL APIs](#)

## DHCPv4C HAL

All HAL functions prototypes and structure definitions are available in dhcpv4c\_api.h file.

- DHCPv4C HAL is used for the RDK-B DHCPv4 Client Status abstraction layer.
- DHCPv4C HAL API's functionality should be implemented by OEMs.
- dhcpv4c\_api.c provides the function call prototypes and structure definitions used for the RDK-Broadband DHCPv4 Client Status abstraction layer.
- Some of the APIs are :
  1. dhcpv4c\_get\_ert\_lease\_time
  2. dhcpv4c\_get\_ert\_remain\_lease\_time
  3. dhcpv4c\_get\_ert\_remain\_renew\_time
  4. dhcpv4c\_get\_ert\_remain\_rebind\_time
  5. dhcpv4c\_get\_ert\_config\_attempts
  6. dhcpv4c\_get\_ert\_ifname
  7. dhcpv4c\_get\_ert\_fsm\_state
  8. dhcpv4c\_get\_ert\_ip\_addr

9. dhcpv4c\_get\_ert\_mask
10. dhcpv4c\_get\_ert\_gw
11. dhcpv4c\_get\_ert\_dns\_svrs
12. dhcpv4c\_get\_ert\_dhcp\_svr
13. dhcpv4c\_get\_ecm\_lease\_time
14. dhcpv4c\_get\_ecm\_remain\_lease\_time
15. dhcpv4c\_get\_ecm\_remain\_renew\_time
16. dhcpv4c\_get\_ecm\_remain\_rebind\_time
17. dhcpv4c\_get\_ecm\_config\_attempts
18. dhcpv4c\_get\_ecm\_ifname
19. dhcpv4c\_get\_ecm\_fsm\_state
20. dhcpv4c\_get\_ecm\_ip\_addr

- To see the API specification of DHCPv4C HAL please refer - [DHCPv4C HAL APIs](#)

## VLAN HAL

All HAL functions prototypes and structure definitions are available in vlan\_hal.h file.

- VLAN HAL is or the RDK-B Broadband VLAN abstraction layer.
- VLAN HAL layer is intended to support VLAN drivers through the System Calls.
- The APIs are :
  1. vlan\_hal\_addGroup
  2. vlan\_hal\_delGroup
  3. vlan\_hal\_addInterface
  4. vlan\_hal\_delInterface
  5. vlan\_hal\_printGroup
  6. vlan\_hal\_printAllGroup
  7. vlan\_hal\_delete\_all\_Interfaces
  8. \_is\_this\_group\_available\_in\_linux\_bridge
  9. \_is\_this\_interface\_available\_in\_linux\_bridge
  10. \_is\_this\_interface\_available\_in\_given\_linux\_bridge
  11. \_get\_shell\_outputbuffer
  12. insert\_VLAN\_ConfigEntry
  13. delete\_VLAN\_ConfigEntry
  14. get\_vlanId\_for\_GroupName
  15. print\_all\_vlanId\_Configuration
- To see the API specification of VLAN HAL please refer - [VLAN HAL APIs](#)

## Firewall HAL

All HAL functions prototypes and structure definitions are available in hal\_firewall.h file.

- This module is responsible for setting firewall rules like port forwarding, port triggering Parental control etc.
- Some of the APIs are :
  1. firewall\_service\_init
  2. firewall\_service\_start
  3. firewall\_service\_restart
  4. firewall\_service\_stop
  5. firewall\_service\_close
  6. GetHttpPortValue
  7. Wan2lan\_log\_deletion\_setup
  8. Wan2lan\_log\_insertion\_setup
  9. GettingWanIP\_remotemgmt\_deletion\_logsetup
  10. GettingWanIP\_remotemgmt\_insertion\_logsetup
  11. DeleteRemoteManagementIptablesRules
  12. AddRemoteManagementIptablesRules
  13. DisablingHttps
  14. EnablingHttps
  15. DisablingHttp
  16. EnablingHttp
  17. SetHttpPort
  18. SetHttpsPort
  19. RemoteManagementiptableRulesetoperation
  20. BasicRouting\_Wan2Lan\_SetupConnection
- To see the API specification of Firewall HAL please refer - [Firewall HAL APIs](#)

## DPOE HAL

All HAL functions prototypes and structure definitions are available in dpoe\_hal.h file.

- DPOE HAL is used for the RDK-Broadband DPOE hardware abstraction layer as per the DPoE-SP-OAMv1.0-I08-140807 specification.
- Some of the APIs are :
  1. dpoe\_getOnuld
  2. dpoe\_getFirmwareInfo
  3. dpoe\_getEponChipInfo
  4. dpoe\_getManufacturerInfo

5. dpoe\_getNumberOfNetworkPorts
  6. dpoe\_getNumberOfS1Interfaces
  7. dpoe\_getOnuPacketBufferCapabilities
  8. dpoe\_getOamFrameRate
  9. dpoe\_getLidForwardingState
  10. dpoe\_getDeviceSysDescrInfo
  11. dpoe\_getMaxLogicalLinks
  12. dpoe\_setResetOnu
  13. dpoe\_getStaticMacTable
  14. dpoe\_getEponMode
  15. dpoe\_getDynamicMacAddressAgeLimit
  16. dpoe\_getDynamicMacLearningTableSize
  17. dpoe\_getDynamicMacTable
  18. dpoe\_getStaticMacTable
  19. dpoe\_getMacLearningAggregateLimit
  20. dpoe\_getOnuLinkStatistics
- To see the API specification of DPOE HAL please refer - [DPOE HAL APIs](#)

## Bluetooth HAL

All HAL functions prototypes and structure definitions are available in bt\_hal.h file.

- The APIs are :
  1. ble\_Enable
  2. ble\_GetStatus
- To see the API specification of Bluetooth HAL please refer - [Bluetooth HAL APIs](#)

## MSO Management HAL

All HAL functions prototypes and structure definitions are available in mso\_mgmt\_hal.h file.

- MSO Management HAL is used for the RDK-Broadband hardware abstraction layer for MSO Management.
- The APIs are :
  1. mso\_pwd\_ret\_status mso\_validatepwd
  2. mso\_set\_pod\_seed
  3. mso\_get\_pod\_seed
- To see the API specification of MSO Management HAL please refer - [MSO Management HAL APIs](#)

## Voice HAL

All HAL functions prototypes and structure definitions are available in voice\_hal.h file.

- Voice HAL is used for the RDK-Broadband hardware abstraction layer for VoIP.
- Some of the APIs are :
  1. voice\_hal\_Init
  2. voice\_hal\_InitDB
  3. voice\_hal\_Deinit
  4. voice\_hal\_DeinitDB
  5. voice\_hal\_setVoiceProcessState
  6. voice\_hal\_getVoiceProcessState
  7. voice\_hal\_getVoiceProcessStatus
  8. voice\_hal\_getConfigSoftwareVersion
  9. voice\_hal\_getCountProfiles
  10. voice\_hal\_getServiceVersion
  11. voice\_hal\_getCountServices
  12. voice\_hal\_getCountLines
  13. voice\_hal\_getCountPhyInterfaces
  14. voice\_hal\_setIpAddressFamily
  15. voice\_hal\_getBoundIfName
  16. voice\_hal\_setBoundIfName
  17. voice\_hal\_setIpAddressFamily
  18. voice\_hal\_getIpAddressFamily
  19. voice\_hal\_setLinkState
  20. voice\_hal\_setIpWanAddress
- To see the API specification of Voice HAL please refer - [Voice HAL APIs](#)

## WAN HAL

All HAL functions prototypes and structure definitions are available in wan\_hal.h file.

- The APIs are :
  1. wan\_hal\_Init
  2. wan\_hal\_SetSelfHealConfig
  3. wan\_hal\_SetWanConnectionEnable
  4. wan\_hal\_SetSelfHealConfig
  5. wan\_hal\_GetWanOEUpstreamCurrRate

6. wan\_hal\_GetWanOEDownstreamCurrRate
  7. wan\_hal\_SetQoSConfiguration
  8. wan\_hal\_RestartWanService
- To see the API specification of WAN HAL please refer - [WAN HAL APIs](#)

## TR69\_TLV HAL

All HAL functions prototypes and structure definitions are available in Tr69\_Tlv.h file.

- Telemetry Key fields and data fields are stored in the database as TLV (Tag, Length, Value)
1. Tag - uniquely identifies the field.
  2. Length - gives the size (in number of bytes) of the data associated with the field.
  3. Value - contains the actual data associated with the field stored in network byte ordering.

## Common HAL

- A common HAL provides the necessary abstraction to all the CCSP components to interface with other common hardware components.
- Eg : Platform HAL

## Platform HAL

- Platform HAL is an abstraction layer, implemented to interact with cable modem device for getting the hardware specific details such as Firmware Name, Boot loader Version, etc.
- This HAL layer is intended to support platform drivers
- platform\_hal.c file provides the function call prototypes and structure definitions used for the platform hardware abstraction layer
- Some of the APIs are :

1. platform\_hal\_GetDeviceConfigStatus
2. platform\_hal\_GetTelnetEnable
3. platform\_hal\_GetSSHEnable
4. platform\_hal\_SetSSHEnable
5. platform\_hal\_GetSNMPEnable
6. platform\_hal\_SetSNMPEnable
7. platform\_hal\_GetSerialNumber
8. platform\_hal\_GetWebUITimeout
9. platform\_hal\_SetWebUITimeout
10. platform\_hal\_GetWebAccessLevel
11. platform\_hal\_SetWebAccessLevel
12. platform\_hal\_PandMDBInit
13. platform\_hal\_DocsisParamsDBInit
14. platform\_hal\_GetModelName
15. platform\_hal\_GetFirmwareName
16. platform\_hal\_GetHardwareVersion
17. platform\_hal\_GetSoftwareVersion
18. platform\_hal\_GetBootloaderVersion
19. platform\_hal\_GetBaseMacAddress
20. platform\_hal\_GetHardware

To see the API specification of Platform HAL please refer - [Platform HAL APIs](#)