



Research & Vehicle Technology
“Infotainment Systems Product Development”

Feature – Wireless Interface Router Client1
v2 - AV

Infotainment Subsystem Part Specific
Specification (SPSS)

Version 1.0
UNCONTROLLED COPY IF PRINTED

Version Date: February 20, 2020

FORD CONFIDENTIAL



Revision History

Date	Ver	Notes	
February 20, 2020	1.0	Initial Release	



Table of Contents

REVISION HISTORY	2
1 OVERVIEW	5
1.1 Terminology and Abbreviations	5
2 ARCHITECTURAL DESIGN	7
2.1 WIR-CLD-REQ-276162/E-Wireless Interface Router Client1	7
2.2 WIR-CLD-REQ-289670/E-Wireless Interface Router Client2	7
2.3 WIR-CLD-REQ-350862/C-Wireless Interface Router Client3	7
2.4 WIR-CLD-REQ-370364/B-Wireless Interface Router Client4	7
2.5 WIR-CLD-REQ-370370/B-Wireless Interface Router Client5	8
2.6 WIR-CLD-REQ-378489/A-Wireless Interface Router Client6	8
2.7 WIR-CLD-REQ-276161/D-Wireless Interface Router Server	8
2.8 Physical Mapping of Classes	8
2.9 WIRClient1 Interface	9
2.9.1 WIR-IIR-REQ-378576/A-WIRClient1_Tx	9
2.9.2 WIR-IIR-REQ-378577/A-WIRClient1_Rx	62
3 FUNCTIONAL DEFINITION	125
3.1 WIR-FUN-REQ-378578/A-Wifi Handler	125
3.1.1 Requirements	125
3.1.2 Use Cases	129
3.1.3 White Box View	129
3.2 WIR-FUN-REQ-378637/A-Local Controller Client	130
3.2.1 Requirements	130
3.2.2 Use Cases	133
3.2.3 White Box View	133
3.3 WIR-FUN-REQ-378652/A-Intents	134
3.3.1 Requirements	134
3.3.2 Use Cases	137
3.3.3 White Box View	137
3.4 WIR-FUN-REQ-378664/A-Policy Manager	138
3.4.1 Requirements	138
3.4.2 Use Cases	139
3.4.3 White Box View	140
3.5 WIR-FUN-REQ-378679/A-Tunnel Manager	141
3.5.1 Requirements	141
3.5.2 Use Cases	145
3.5.3 White Box View	145
3.6 WIR-FUN-REQ-378693/A-Diagnostics	146
3.6.1 Requirements	146
3.6.2 Use Cases	148
3.6.3 White Box View	148
3.7 WIR-FUN-REQ-378702/A-Data Usage	149
3.7.1 Requirements	149
3.7.2 Use Cases	150
3.7.3 White Box View	150



3.8	WIR-FUN-REQ-378718/A-WIFI Connect Reminders - HMI	151
3.8.1	Requirements	151
3.8.2	Use Cases	152
3.8.3	White Box View	152
3.9	WIR-FUN-REQ-378728/A-Captive Portal Check	153
3.9.1	Requirements	153
3.9.2	Use Cases	154
3.9.3	White Box View	163
3.10	WIR-FUN-REQ-379786/A-AV Specific Requirements	164
3.10.1	Requirements	164
3.10.2	Use Cases	164
3.10.3	White Box View	164
3.11	WIR-FUN-REQ-378759/A-Performance	165
3.11.1	Requirements	165
3.11.2	Use Cases	165
3.11.3	White Box View	165
3.12	WIR-FUN-REQ-378765/A-WIRClient1 Configuration	166
3.12.1	Requirements	166
3.12.2	Use Cases	166
3.12.3	White Box View	166
4	APPENDIX: REFERENCE DOCUMENTS.....	167



1 Overview

The Wireless Interface Router feature has an opportunity to reduce greatly Ford's ongoing connectivity costs by leveraging non-metered client mode Wi-Fi and driving cellular usage at cheaper, off peak times. Further, the WIR gives vehicle applications an opportunity to utilize any vehicle edge interface directly, without any intermediate protocol, like CAN.

WIR enables vehicle software applications to reach the cloud via any edge network interface on the vehicle. Further, WIR can schedule notifications to applications when more cost favorable networks are available. The architecture of WIR is expandable to new edge network interfaces and is able to support new ECUs as they integrate with ECG via Ethernet

Routing data through SYNC, TCU and TCU-B Wi-Fi connections will drive down the overall costs of cellular data consumption. Once implemented this service can be used for any connected service within SYNC, Cluster, ADAS, SDS, SDS 10G PHY, ADSIM, DDSM, etc. that requires off board communication to a 1st, 2nd, or 3rd party server.

With the introduction of Ethernet as an internal vehicle network the ability to allow various subsystems to establish an IP connection to an off board server (IP Based Pass through) is a key enabler. These subsystems include any Ethernet connected node in the vehicle: ECG, SYNC, TCU, TCU-B, ADAS, SDS, SDS 10G PHY, DDSM and ADSIM etc. The off board destinations may be in Ford environment or third party systems not necessarily in Ford enterprise IT Systems. Enabling various services offered by these off board destinations will result in customer satisfaction and delight.

IP Based Pass through works with Wireless Interface Router (aka Connectivity Manager) in the ECG, TCU, TCU-B and SYNC, an In-Vehicle Policy Management application and with an authorization/policy management application in the NG SDN. These will be the key components to provide session authorization, management, and security authentication for this service. Once implemented, this service can be used by any Ethernet connected service within SYNC, Cluster, ADAS, SDS, SDS 10G PHY, DDSM and ADSIM etc. that requires an off board connection to a 1st, 2nd, or third party destination address.

Some WiFi network service provider has captive portal page, aka landing page. Once WLAN client is connected to WiFi Hotspot, all IP traffic is ignored except HTTP. Also, all HTTP request is redirected to captive portal page where the customer should accept terms and conditions before getting internet connection through WiFi Hotspot. WIR sends HTTP request to predetermined URL to see response is normal (2XX OK) or redirection (3XX redirection). WIR sends URL and local IP address to WEB engine to display landing page on SYNC display, so the customer can interact with the page. WIR sends internet connection ready notification to WIR application when it receives normal response.

1.1 Terminology and Abbreviations

The following table lists terminologies that are used in this document along with a brief description.

Term	Description
ADSIM	Automated Driving System Interface Module
AV	Autonomous Vehicle
AVDT	Autonomous Vehicle Data Transfer
AVS	Autonomous Vehicle System
CAN	Controller Area Network
CCS	Customer Connectivity Settings
CHR	Connected HMI Radio
DDSM	Digital Data Storage Module
DTM	Data Transfer Manager
ECG	Enhanced Central Gateway
ECU	Electronic Control Unit
FNV	Fully Networked Vehicle
FTCP	Ford Telematics Control Protocol
GRE	Generic Routing and Encapsulation
HMI	Human Machine Interface
IPC	Inter Process Communication



Term	Description
SDN	Service Delivery Network
SDS	Self-Driving System
SOA	Service Oriented Architecture
TCU	Telematics Control Unit
TCU-B	Telematics Control Unit-B
WIR	Wireless Interface Router
WLAN	Wireless Local Area Network
10G PHY	10 Gigabit Ethernet/Physical layer interface (transceivers)



2 Architectural Design

2.1 WIR-CLD-REQ-276162/E-Wireless Interface Router Client1

The Wireless Interface Router Client1 (WIRClient1) is responsible for the tasks listed below:

- Handles connectivity request from applications
- Responsible for calculating data usage
- Responsible for policy validation
- Responsible for setting up Tunnels
 - Routing table instead of Tunnels for AV
- Responsible for handling WIFI connection
- Interfaces with central connection controller
- Providing connections to applications when central controller is not active
- Responsible for landing page display by WEB engine and responds according to customer input

Please review the implementation guide/ block diagram to locate the WIRClient1 class.

2.2 WIR-CLD-REQ-289670/E-Wireless Interface Router Client2

The Wireless Interface Router Client2 (WIRClient2) is responsible for the tasks listed below:

- Handles connectivity request from applications
- Responsible for calculating data usage
- Responsible for policy validation
- Responsible for setting up Tunnels
 - Routing table instead of Tunnels for AV
- Responsible for setting up Cellular APN1 and APN2.
 - Also APN3 and APN4 for China (non-AV)
- Responsible for handling WIFI connection
- Interfaces with central connection controller
- Providing connections to applications when central controller is not active

Please review the implementation guide/ block diagram to locate the WIRClient2 class.

2.3 WIR-CLD-REQ-350862/C-Wireless Interface Router Client3

The Wireless Interface Router Client3 (WIRClient3) is responsible for the tasks listed below:

- Handles connectivity request from applications
- Responsible for calculating data usage
- Responsible for policy validation
- Responsible for setting up Tunnels
 - Routing table instead of Tunnels for AV
- Responsible for handling WIFI connection
- Interfaces with central connection controller
- Providing connections to applications when central controller is not active
- Managing availability of 10G PHY interface for AV
- Responsible for setting up application connections to 10G PHY interface for AV

Please review the implementation guide/ block diagram to locate the WIRClient3 class.

2.4 WIR-CLD-REQ-370364/B-Wireless Interface Router Client4

The Wireless Interface Router Client4 (WIRClient4) is responsible for the tasks listed below:

- Handles connectivity request from applications
- Responsible for calculating data usage
- Responsible for policy validation
- Responsible for setting up Tunnels
 - Routing table instead of Tunnels for AV



- Responsible for handling WIFI connection
- Interfaces with central connection controller
- Providing connections to applications when central controller is not active

Please review the implementation guide/ block diagram to locate the WIRClient4 class.

2.5 WIR-CLD-REQ-370370/B-Wireless Interface Router Client5

The Wireless Interface Router Client5 (WIRClient5) is responsible for the tasks listed below:

- Handles connectivity request from applications
- Responsible for calculating data usage
- Responsible for policy validation
- Responsible for setting up Tunnels
 - Routing table instead of Tunnels for AV
- Responsible for setting up Cellular APN1 and APN2
- Responsible for handling WIFI connection
- Interfaces with central connection controller
- Providing connections to applications when central controller is not active

Please review the implementation guide/ block diagram to locate the WIRClient5 class.

2.6 WIR-CLD-REQ-378489/A-Wireless Interface Router Client6

The Wireless Interface Router Client6 (WIRClient6) is responsible for the tasks listed below:

- Handles connectivity request from applications
- Responsible for calculating data usage
- Responsible for policy validation
- Responsible for setting up routing table
- Responsible for handling WIFI connection
- Interfaces with central connection controller
- Providing connections to applications when central controller is not active

Please review the implementation guide/ block diagram to locate the WIRClient6 class.

2.7 WIR-CLD-REQ-276161/D-Wireless Interface Router Server

The Wireless Interface Router Server (WIRServer) is responsible for the tasks listed below:

- Interfaces with Wireless interface router Clients
- Responsible for providing connection tokens
 - Connection Interface instead of tokens for AV
- Responsible for bandwidth shaping
- Responsible for diagnostics
- Responsible for calculating data usage
- Responsible for policy validation
- Responsible for cloud interface
- Responsible for captive portal check
- Responsible for asking landing page display to WEB engine

Please review the implementation guide/ block diagram to locate the WIRServer class.

2.8 Physical Mapping of Classes

The table below shows an example of how the logical classes that make up the WIR feature may be mapped into physical modules. This mapping is specific to the V713 full production architecture and does not necessarily carryover to other carlines or vehicle architectures.



Logical Class	Physical Module (ECU)
WIRServer	ECG
WIRClient1	APIM
WIRClient2	TCU
WIRClient3	SDS
WIRClient4	ADSIM
WIRClient5	TCU-B
WIRClient6	DDSM

2.9 WIRClient1 Interface

2.9.1 WIR-IIR-REQ-378576/A-WIRClient1_Tx

2.9.1.1 MD-REQ-380258/A-VlanAdd

This API is used internally by WIRClient and WIRServer to request a new VLAN configuration. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
R	origEcu	String	-	Char Value:0-255 No String length limit	Origination ECU enum in string format
R	ifaceEcu	String	-	Char Value:0-255 No String length limit	Interface ECU enum in string format
R	mtu	String	-	Char Value:0-255	Interface MTU in string format



				No String length limit	
R	svcLevel	String	-	Char Value:0-255 No String length limit	Service Level of the VLAN in string format
R	ifaceType	Enum	-	-	Network Interface Type over which the VLAN would be setup
			None	0x0	
			TcuCell	0x1	
			TcuWifi	0x2	
			SyncWifi	0x3	
			SyncApplink	0x4	
			TcubCell	0x5	
			TcubWifi	0x6	
			VdsLan	0x7	
			Error	0x8	
R	vlanEndpointType	Enum	-	-	VLAN Endpoint type
			Host	0x0	Host type endpoint
			Gateway	0x1	Gateway type endpoint
Response					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	
			VLAN_SUCCESS	0x1	
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
R	ipV4Address	String	-	Char Value:0-255 No String length limit	VLAN IP Address

2.9.1.2 MD-REQ-380259/A-VlanRemove

This API is used internally by WIRClient and WIRServer to request removal of old VLAN configuration. WIR also uses this API for its response.

Method Type	One-Shot (A-Synch)
--------------------	--------------------



QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
Response					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	
			VLAN_SUCCESS	0x1	
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id

2.9.1.3 MD-REQ-380260/A-VlanLink

This API is used internally by WIRClient and WIRServer to request link VLAN. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM



R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
R	ifaceName	String	-	Char Value:0-255 No String length limit	Network Interface as enumerated in the OS
R	serviceLevel	String	-	Char Value:0-255 No String length limit	Service Level of the VLAN in string format

Response

R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	
			VLAN_SUCCESS	0x1	
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
R	ifaceName	String	-	Char Value:0-255 No String length limit	Network Interface as enumerated in the OS



R	serviceLevel	String	-	Char Value:0-255 No String length limit	Service Level of the VLAN in string format
R	linkId	String	-	Char Value:0-255 No String length limit	Unique link ID generated by Networking

2.9.1.4 MD-REQ-380261/A-VlanUnlink

This API is used internally by WIRClient and WIRServer to request unlink VLAN. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
R	linkId	String	-	Char Value:0-255 No String length limit	Unique link ID generated by Networking
Response					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	
			VLAN_SUCCESS	0x1	



R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
R	linkId	String	-	Char Value:0-255 No String length limit	Unique link ID generated by Networking

2.9.1.5 MD-REQ-380262/A-VnmReset

This API is used internally by WIRClient and WIRServer to request a VLAN configuration reset. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	EcuType	Enum	-	-	Path Id used internally by CM
			Unknown	0x0	Error
			ECG	0x1	
			TCU	0x2	
			SYNC	0x3	
			TCU_B	0x4	
			VDS	0x5	
			AVPIM	0x6	
			DDSM	0x7	
Response					
R	EcuType	Enum	-	-	Path Id used internally by CM
			Unknown	0x0	Error
			ECG	0x1	
			TCU	0x2	
			SYNC	0x3	
			TCU_B	0x4	
			VDS	0x5	
			AVPIM	0x6	
			DDSM	0x7	
R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	
			VLAN_SUCCES S	0x1	

2.9.1.6 MD-REQ-380263/A-VnmLinkNetwork

This API is used internally by WIRClient and WIRServer to request to link a network to VLAN. WIR also uses this API for its response.

Method Type	One-Shot (A-Synch)
--------------------	--------------------



QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	origEcu	String	-	Char Value:0-255 No String length limit	Origination ECU enum in string format
R	ifaceEcu	String	-	Char Value:0-255 No String length limit	Interface ECU enum in string format
R	interfaceZero	String	-	Char Value:0-255 No String length limit	User's provided interface name to be linked
R	interfaceOne	String	-	Char Value:0-255 No String length limit	Network interface to be linked
R	ipAddr	String	-	Char Value:0-255 No String length limit	Gateway ip address associated with network interface
Response					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	origEcu	String	-	Char Value:0-255 No String length limit	Origination ECU enum in string format
R	linkId	String	-	Char Value:0-255 No String length limit	Unique link ID generated by Networking
R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	
			VLAN_SUCCESS	0x1	

**2.9.1.7 MD-REQ-380264/A-VnmUnlinkNetwork**

This API is used internally by WIRClient and WIRServer to request to unlink a network to VLAN. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	origEcu	String	-	Char Value:0-255 No String length limit	Origination ECU enum in string format
R	ifaceEcu	String	-	Char Value:0-255 No String length limit	Interface ECU enum in string format
R	linkId	String	-	Char Value:0-255 No String length limit	Unique link ID generated by Networking
Response					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	origEcu	String	-	Char Value:0-255 No String length limit	Origination ECU enum in string format
R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	
			VLAN_SUCCESS	0x1	

2.9.1.8 MD-REQ-380265/A-NetworkInterfaceAllocation

This API is used internally by WIRClient and WIRServer to request a new network interface. WIR also uses this API for its response. The response will return success or failure, actual interface is returned through NetworkInterfaceAllocationStatusInd.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					



R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	allocId	Int32	-	0-4294967295	Allocation ID will be assigned by WIR
R	IntentType	Enum	-	-	Intent Type
			Foreground	0x0	
			Background_BestEffort	0x1	
			Background_Guaranteed	0x2	
			Special	0x3	
			OffPeak	0x4	
			LAN_Only	0x5	
R	InterfaceType	Enum	-	-	Interface Type
			None	0x0	
			TcuCell	0x1	
			TcuWifi	0x2	
			SyncWifi	0x3	
			SyncApplink	0x4	
			TcubCell	0x5	
			TcubWifi	0x6	
			VdsLan	0x7	
			Error	0x5	
R	CellApnType	Enum	-	-	Cellular APN Type
			Internet	0x0	
			Ford	0x1	
			Tethering	0x2	
			FOTA	0x3	
			HTTP	0x4	
R	PriorityLevel	Enum	-	-	Priority Level
			Priority 0	0x0	
			Priority 1	0x1	
			Priority 2	0x2	
			Priority 3	0x3	
			Priority 4	0x4	
R	expiry	Int32	-	0-4294967295	Expiration timer
R	OffPeakFlag	Enum	-	-	Offpeak flag
			No	0x0	
			Yes	0x1	
R	WifiPreferredFlag	Enum	-	-	Wifi preferred flag
			No	0x0	
			Yes	0x1	
R	LanPreferredFlag	Enum	-	-	LAN preferred flag
			No	0x0	
			Yes	0x1	
R	ssid	String	-	Char Value:0-255	SSID of network



				No String length limit	
R	bssid	String		Char Value:0-255 No String length limit	Optional - BSSID of AP
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	WlanSecurity	Enum	-	-	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanWepSecurity				WEP Settings
		String	Key	Char Value:0-255 No String length limit	
		Int32	defaultKeyIndex	0-4294967295	
R	WlanWpsSecurity				WPS Settings
		WlanWpsType			Type
		String		Char Value:0-255 No String length limit	Pin
R	WlanWpaPersonalSecurity	String		Char Value:0-255 No String length limit	WPA/WPA2-Personal Settings - password
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None
			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES



			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanWpaEnterprise	TBD	-	-	TBD
R	WlanIpv4AddrType	Enum	-	-	
			WLAN_IPV4_ADDR_MIN	0x0	
			WLAN_IPV4_ADDR_NONE	0x1	No IPv4 Addressing is used
			WLAN_IPV4_ADDR_STATIC	0x2	Static IPv4 Address
			WLAN_IPV4_ADDR_DHCP_CLIENT	0x3	DHCP Client IPv4 Address
			WLAN_IPV4_ADDR_DHCP_SERVER	0x4	DHCP Server IPv4 Address
			WLAN_IPV4_ADDR_MAX	0x5	
R	WlanIpv4Addr	-	-	-	
		String	Ip	Char Value:0-255 No String length limit	IP address of current connection
		String	Netmask	Char Value:0-255 No String length limit	Netmask of current connection
		String	Gateway	Char Value:0-255 No String length limit	default gateway of current connection
		String	dnsPref	Char Value:0-255 No String length limit	Preferred DNS server
		String	dnsAlt	Char Value:0-255 No String length limit	Secondary DNS server



R	WlanIpv6AddrType	Enum	-	-	
			WLAN_IPV6_A DDR_MIN	0x0	
			WLAN_IPV6_A DDR_NONE	0x1	No IPv6 Addressing is used
			WLAN_IPV6_A DDR_STATIC	0x2	Static IPv6 Address
			WLAN_IPV6_A DDR_MAX	0x3	
R	WlanIpv6Addr	TBD	-	-	TBD
R	ExclusiveUse	Boolean	-	0/1	Exclusive to WIR Client
R	WlanProfileSource	Enum	-	-	Origin of the profile
			PROFSRC_DE FAULT	0x0	Default internal
			PROFSRC_HMI	0x1	HMI/User configured
			PROFSRC_CL OUD	0x2	Cloud pushed
			PROFSRC_SP ECIAL	0x3	Special Intent
			PROFSRC_PR OV	0x4	EOL provisioned
R	timestamp	Int64	-	0- 18446744 07370955 1615	Last known time stamp
R	hidden	Boolean	-	0/1	Hidden AP
R	IfaceNametoLink	String	-	Char Value:0- 255 No String length limit	AP SSID
R	Bw_downlink	Int32	-	0- 42949672 95	Application required downlink bandwidth
R	Bw_uplink	Int32	-	0- 42949672 95	Application required uplink bandwidth
Response					
R	appld	String	-	Char Value:0- 255 No String length limit	WIR Client Appld
R	allocId	Int32	-	0- 42949672 95	Allocation ID will be assigned by WIR
R	WirRet	Enum	-	-	Return Command
			WIR_ERROR	0x0	Error/Failure
			WIR_SUCCESS	0x1	Success

2.9.1.9 MD-REQ-380266/A-NetworkInterfaceRelease

This API is used internally by WIRClient and WIRServer to request a network interface release. WIR also uses this API for its response.



Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	allocId	Int32	-	0-4294967295	Allocation ID will be assigned by WIR
Response					
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	allocId	Int32	-	0-4294967295	Allocation ID will be assigned by WIR
R	WirRet	Enum	-	-	Return Command
			WIR_ERROR	0x0	Error/Failure
			WIR_SUCCESS	0x1	Success

2.9.1.10 MD-REQ-380270/A-WIRClientRegisterInd

This API is used internally by WIRClient and WIRServer to register a WIRClient.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	N/A
Response					
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld

2.9.1.11 MD-REQ-380271/A-WIRClientDeregisterInd

This API is used internally by WIRClient and WIRServer to deregister a WIRClient.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	N/A
Response					



R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
---	-------	--------	---	--	------------------

2.9.1.12 MD-REQ-380272/A-WebViewDisplay

This API is used internally by WIRClient and WIRServer to request WEB View display. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	ipAddr	String	-	Char Value:0-255 No String length limit	IP address to be used for connection
R	uri	String	-	Char Value:0-255 No String length limit	WEB page URI
Response					
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	WirRet	Enum	-	-	Return Command
			WIR_ERROR	0x0	Error/Failure
			WIR_SUCCESS	0x1	Success

2.9.1.13 MD-REQ-380273/A-HeartBeat

This API is used internally by WIRClient and WIRServer to report a heartbeat message.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	N/A
Response					
R	hbMsg	String	-		Payload with a message string
R	HeartBeatType	Enum	-	-	HeartBeat event type



			HB_INITIAL	0x0	Initial HeartBeat Type
			HB_REGUL AR	0x1	Regular Heartbeat Type
R	hbIndex	Int32	-	0- 4294967295	HeartBeat event index
R	hbTimer	Int32	-	0- 4294967295	HeartBeat event timer duration provided by ECG

2.9.1.14 MD-REQ-380274/A-Policy

This API is used internally by WIRClient and WIRServer to request a network policy for an application. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0- 255 No String length limit	WIR Client Appld
Response					
R	CtrlRet	Enum	-	-	
			CTRL_ERROR	0x0	Error/Failure
			CTRL_SUCCES S	0x1	Success
R	appld	String	-	Char Value:0- 255 No String length limit	WIR Client Appld
R	Policy	Int32	-	0- 42949672 95	Network policy, bit flag
			TCU_CELLULA R_APN1	0x001	
			TCU_WIFI	0x002	
			TCU_CELLULA R_APN2	0x004	
			SYNC_WIFI	0x008	
			APPLINK_WIFI	0x010	
			OFF_PEAK	0x020	
			TCU- B_CELLULAR_ APN1	0x040	
			TCU-B_WIFI	0x100	
			TCU- B_CELLULAR_ APN2	0x200	
			VDS_LAN	0x400	

2.9.1.15 MD-REQ-380279/A-Enable

This API is used internally by WIRClient and WIRServer to request enabling WLAN. WIR also uses this API for its response.



Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

2.9.1.16 MD-REQ-380280/A-Disable

This API is used internally by WIRClient and WIRServer to request disabling WLAN. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

2.9.1.17 MD-REQ-380281/A-ScanApimAPs

This API is used internally by WIRClient and WIRServer to request a scan of available APs. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					



R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success
R	ifId	Int32	-	0-4294967295	Interface ID
R	scanId	Int64	-	0-18446744073709551615	ID value used to track scans
R	passive	Boolean	-	0/1	true = passive scan (no pkts tx'd), false = active scan
R	WlanChannel	Enum	-	-	Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	ssid	String	-	Char Value:0-255 No String length limit	list of SSIDs to specifically scan for. Used to scan for hidden networks. "passive" will be ignored as an active scan is required
R	Count	Int32	-	0-4294967295	Number of APs found
R	ssid	String	-	Char Value:0-255 No String length limit	SSID – scan result
R	bssid	String	-	Char Value:0-255 No String length limit	MAC address of AP – scan result
R	WlanChannel	Enum	-	-	Channel of AP – scan result
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	



			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	Wlan Bandwidth	Enum	-	-	
			WLAN_BW_MIN	0x0	
			WLAN_BW_MHZ20	0x1	20MHz channel
			WLAN_BW_MHZ40	0x2	40MHz channel; must use 11n/11ac. Few devices support on 2.4Ghz
			WLAN_BW_MHZ80	0x3	80MHz channel; must use 11ac
			WLAN_BW_MHZ8080	0x4	80-80MHz, 2 non-contiguous 80MHz; must use 11ac
			WLAN_BW_MHZ160	0x5	160MHz channel; must 11ac
			WLAN_BW_MAX	0x6	
R	rssi	Int32	-	0-4294967295	RSSI of beacon/probe response
R	WlanSecurity	Enum	-	-	Security settings of scan results
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	isWpsSupported	Boolean	-	0/1	does AP support WPS
R	isEssSupported	Boolean	-	0/1	is AP part of an Extended Service Set
R	ipV4Addr	String	-	Char Value:0-255 No String length limit	IPv4 address
R	netmask	String	-	Char Value:0-255 No String length limit	Net mask
R	connected	Boolean	-	0/1	connected to AP
R	prevConnected	Boolean	-	0/1	Previously connected AP
R	isHidden	Boolean	-	0/1	Hidden AP
R	WifiConnectionType	Enum	-	-	
			CONNECTION_TYPE_PIN	0x0	PIN



			CONNECTION_ TYPE_WPS_P I N	0x1	WPS (Wi-Fi Protected Setup) PIN
			CONNECTION_ TYPE_WPS_P B C	0x2	WPS Push Button Control
			CONNECTION_ TYPE_OPEN	0x3	Open WiFi
			CONNECTION_ TYPE_HIDDEN _SECURE	0x4	Hidden + security, SSID is not broadcasted
			CONNECTION_ TYPE_HIDDEN _OPEN	0x5	Hidden + open, SSID is not broadcasted
R	EWifiConnectionStatus	Enum	-	-	
			WIFI_STATUS_ SUCCESS	0	Success
			WIFI_AP_SCAN _SUCCESS	1	Scan Successful
			WIFI_NO_APS_ AVAILABLE	2	No Access Points available
			WIFI_CONNEC TED	3	Connected to Access Point
			WIFI_NOT_CO NNECTED	4	Wifi is not connected to Access Point
			WIFI_OTH_FAI L	5	Authentication Failure
			WIFI_CONNEC TING	6	Connecting to Access Point
			WIFI_SECURIT Y_MISSMATCH	7	There is security type mismatch between what requested and what is in actual
			WIFI_AUTHENT ICATING	8	In authenticating
			WIFI_ABORTE D	9	Connection aborted
			WIFI_WRONG_ PASSWORD	10	Failed with wrong password
			WIFI_WRONG_ WPS_PIN	11	Failed with wrong WPS PIN
			WIFI_CONNEC TION_PROGRE SS	12	In connecting
			WIFI_RESTRIC TED_NETWORK	13	Connection failure due to trying to connect to a restricted network eg : TCU hotspot
			WIFI_NOT_IMP LEMENTED	101	Not Implemented
			WIFI_STATUS_ FAIL	102	Failure
			WIFI_STATUS_ FAIL_OTHER	203	Other Failure or Connecting to AP Failed
			WIFI_CANCEL _CONNECTION	104	Cancelled
R	complete	Boolea n	-	0/1	TRUE - complete FALSE- more to come

**2.9.1.18 MD-REQ-380282/A-CancelScanApimAPs**

This API is used internally by WIRClient and WIRServer to request to cancel scanning for available APs. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

2.9.1.19 MD-REQ-380283/A-ConnectToApimAP

This API is used internally by WIRClient and WIRServer to request to connect to an available AP. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	ssid	String	-	Char Value:0-255 No String length limit	SSID of network
R	bssid	String	-	Char Value:0-255 No String length limit	Optional - BSSID of AP
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	



			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	WlanSecurity	Enum	-	-	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanWepSecurity	-	-	-	WEP Settings
		String	Key	Char Value:0-255 No String length limit	
		Int32	defaultKeyIndex	0-4294967295	
R	WlanWpsSecurity	-	-	-	WPS Settings
		WlanWpsType			Type
		String	-	Char Value:0-255 No String length limit	Pin
R	WlanWpaPersonalSecurity	String	-	Char Value:0-255 No String length limit	WPA/WPA2-Personal Settings - password
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None
			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES
			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanWpaEnterprise	TBD	-	-	TBD
R	WlanIpv4AddrType	Enum	-	-	



			WLAN_IPV4_A DDR_MIN	0x0	
			WLAN_IPV4_A DDR_NONE	0x1	No IPv4 Addressing is used
			WLAN_IPV4_A DDR_STATIC	0x2	Static IPv4 Address
			WLAN_IPV4_A DDR_DHCP_CL IENT	0x3	DHCP Client IPv4 Address
			WLAN_IPV4_A DDR_DHCP_S ERVER	0x4	DHCP Server IPv4 Address
			WLAN_IPV4_A DDR_MAX	0x5	
R	WlanIpv4Addr	-	-	-	
		String	Ip	Char Value:0- 255 No String length limit	IP address of current connection
		String	Netmask	Char Value:0- 255 No String length limit	Netmask of currenct connection
		String	Gateway	Char Value:0- 255 No String length limit	default gateway of current connection
		String	dnsPref	Char Value:0- 255 No String length limit	Prefered DNS server
		String	dnsAlt	Char Value:0- 255 No String length limit	Secondary DNS server
R	WlanIpv6AddrType	Enum	-	-	
			WLAN_IPV6_A DDR_MIN	0x0	
			WLAN_IPV6_A DDR_NONE	0x1	No IPv6 Addressing is used
			WLAN_IPV6_A DDR_STATIC	0x2	Static IPv6 Address
			WLAN_IPV6_A DDR_MAX	0x3	
R	WlanIpv6Addr	TBD	-	-	
R	ExclusiveUse	Boolea n	-	0/1	Exclusive to WIR Client
R	WlanProfileSource	Enum	-	-	Origin of the profile



			PROFSRC_DE FAULT	0x0	Default internal
			PROFSRC_HMI	0x1	HMI/User configured
			PROFSRC_CL OUD	0x2	Cloud pushed
			PROFSRC_SP ECIAL	0x3	Special Intent
			PROFSRC_PR OV	0x4	EOL provisioned
R	timestamp	Int64	-	0- 18446744 07370955 1615	Last known time stamp
R	hidden	Boolea n	-	0/1	Hidden AP
Response					
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success
R	ssid	String	-	Char Value:0- 255 No String length limit	SSID – scan result
R	bssid	String	-	Char Value:0- 255 No String length limit	MAC address of AP – scan result
R	WlanChannel	Enum	-	-	Channel of AP – scan result
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	Wlan Bandwidth	Enum	-	-	
			WLAN_BW_MI N	0x0	
			WLAN_BW_MH Z20	0x1	20MHz channel
			WLAN_BW_MH Z40	0x2	40MHz channel; must use 11n/11ac. Few devices support on 2.4Ghz
			WLAN_BW_MH Z80	0x3	80MHz channel; must use 11ac
			WLAN_BW_MH Z8080	0x4	80-80MHz, 2 non-contiguous 80MHz; must use 11ac
			WLAN_BW_MH Z160	0x5	160MHz channel; must 11ac
			WLAN_BW_MA X	0x6	
R	rss	Int32	-	0- 42949672 95	RSSI of beacon/probe response



R	WlanSecurity	Enum	-	-	Security settings of scan results
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	isWpsSupported	Boolean	-	0/1	does AP support WPS
R	isEssSupported	Boolean	-	0/1	is AP part of an Extended Service Set
R	ipV4Addr	String	-	Char Value:0-255 No String length limit	IPv4 address
R	netmask	String	-	Char Value:0-255 No String length limit	Net mask
R	connected	Boolean	-	0/1	connected to AP
R	prevConnected	Boolean	-	0/1	Previously connected AP
R	isHidden	Boolean	-	0/1	Hidden AP
R	WifiConnectionType	Enum	-	-	
			CONNECTION_TYPE_PIN	0x0	PIN
			CONNECTION_TYPE_WPS_PIN	0x1	WPS (Wi-Fi Protected Setup) PIN
			CONNECTION_TYPE_WPS_PBC	0x2	WPS Push Button Control
			CONNECTION_TYPE_OPEN	0x3	Open WiFi
			CONNECTION_TYPE_HIDDEN_SECURE	0x4	Hidden + security, SSID is not broadcasted
			CONNECTION_TYPE_HIDDEN_OPEN	0x5	Hidden + open, SSID is not broadcasted
R	EWifiConnectionStatus	Enum	-	-	
			WIFI_STATUS_SUCCESS	0	Success
			WIFI_AP_SCAN_SUCCESS	1	Scan Successful
			WIFI_NO_APS_AVAILABLE	2	No Access Points available



			WIFI_CONNECTED	3	Connected to Access Point
			WIFI_NOT_CONNECTED	4	Wifi is not connected to Access Point
			WIFI_OTH_FAIL	5	Authentication Failure
			WIFI_CONNECTING	6	Connecting to Access Point
			WIFI_SECURITY_MISMATCH	7	There is security type mismatch between what requested and what is in actual
			WIFI_AUTHENTICATING	8	In authenticating
			WIFI_ABORTED	9	Connection aborted
			WIFI_WRONG_PASSWORD	10	Failed with wrong password
			WIFI_WRONG_WPS_PIN	11	Failed with wrong WPS PIN
			WIFI_CONNECTION_PROGRESS	12	In connecting
			WIFI_RESTRICTED_NETWORK	13	Connection failure due to trying to connect to a restricted network eg : TCU hotspot
			WIFI_NOT_IMPLEMENTED	101	Not Implemented
			WIFI_STATUS_FAIL	102	Failure
			WIFI_STATUS_FAIL_OTHER	203	Other Failure or Connecting to AP Failed
			WIFI_CANCEL_CONNECTION	104	Cancelled

2.9.1.20 MD-REQ-380284/A-CancelConnectToApimAP

This API is used internally by WIRClient and WIRServer to request to cancel connecting to an available AP. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	



			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

2.9.1.21 MD-REQ-380285/A-DisconnectFromApimAP

This API is used internally by WIRClient and WIRServer to request to disconnect from an available AP. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

2.9.1.22 MD-REQ-380286/A-ForgetApimAP

This API is used internally by WIRClient and WIRServer to request to forget an available AP. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	ssid	String	-		SSID of AP
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

**2.9.1.23 MD-REQ-380287/A-GetWpsPin**

This API is used internally by WIRClient and WIRServer to request to get a WPS PIN. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success
R	WlanWpsSecurity	Enum	-	-	
			WLAN_WPS_MIN	0x0	
			WLAN_WPS_KEYPAD	0x1	AP supplies the PIN
			WLAN_WPS_PIN	0x2	STA supplies the PIN
			WLAN_WPS_PUSHBUTTON	0x3	Pushbutton
			WLAN_WPS_MAX	0x4	
	pin	String	-	Char Value:0-255 No String length limit	The Pin when using keypad type

2.9.1.24 MD-REQ-380288/A-GetNetworkDetails

This API is used internally by WIRClient and WIRServer to request to get network details. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255	Requesting app ID



				No String length limit	
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success
R	ssid	String	-	Char Value:0-255 No String length limit	SSID of network
R	bssid	String	-	Char Value:0-255 No String length limit	Optional - BSSID of AP
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	WlanSecurity	Enum	-	-	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanWepSecurity	-	-	-	WEP Settings
		String	Key	Char Value:0-255 No String length limit	
		Int32	defaultKeyIndex	0-4294967295	
R	WlanWpsSecurity	-	-	-	WPS Settings
		WlanWpsType	-		Type



		String	-	Char Value:0-255 No String length limit	Pin
R	WlanWpaPersonalSecurity	String	-	Char Value:0-255 No String length limit	WPA/WPA2-Personal Settings - password
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None
			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES
			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanWpaEnterprise	TBD	-	-	TBD
R	WlanIpv4AddrType	Enum	-	-	
			WLAN_IPV4_ADDR_MIN	0x0	
			WLAN_IPV4_ADDR_NONE	0x1	No IPv4 Addressing is used
			WLAN_IPV4_ADDR_STATIC	0x2	Static IPv4 Address
			WLAN_IPV4_ADDR_DHCP_CLIENT	0x3	DHCP Client IPv4 Address
			WLAN_IPV4_ADDR_DHCP_SERVER	0x4	DHCP Server IPv4 Address
			WLAN_IPV4_ADDR_MAX	0x5	
R	WlanIpv4Addr	-	-	-	
		String	Ip	Char Value:0-255 No String length limit	IP address of current connection
		String	Netmask	Char Value:0-255 No String length limit	Netmask of current connection



		String	Gateway	Char Value:0-255 No String length limit	default gateway of current connection
		String	dnsPref	Char Value:0-255 No String length limit	Preferred DNS server
		String	dnsAlt	Char Value:0-255 No String length limit	Secondary DNS server
R	WlanIpv6AddrType	Enum	-	-	
			WLAN_IPV6_ADDR_MIN	0x0	
			WLAN_IPV6_ADDR_NONE	0x1	No IPv6 Addressing is used
			WLAN_IPV6_ADDR_STATIC	0x2	Static IPv6 Address
			WLAN_IPV6_ADDR_MAX	0x3	
R	WlanIpv6Addr	TBD	-	-	
R	ExclusiveUse	Boolean	-	0/1	Exclusive to WIR Client
R	WlanProfileSource	Enum	-	-	Origin of the profile
			PROFSRC_DEFAULT	0x0	Default internal
			PROFSRC_HMI	0x1	HMI/User configured
			PROFSRC_CLOUD	0x2	Cloud pushed
			PROFSRC_SPECIAL	0x3	Special Intent
			PROFSRC_PROVIDED	0x4	EOL provisioned
R	timestamp	Int64	-	0-18446744073709551615	Last known time stamp
R	hidden	Boolean	-	0/1	Hidden AP

2.9.1.25 MD-REQ-380289/A-GetEnableStatus

This API is used internally by WIRClient and WIRServer to request to get enable status. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					



R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success
R	status	Boolean	-	0/1	True - enabled, false-disabled

2.9.1.26 MD-REQ-380290/A-SetNotification

This API is used internally by WIRClient and WIRServer to request to set notification on/off. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	on	Boolean	-	0/1	On or off
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

2.9.1.27 MD-REQ-380291/A-GetNotification

This API is used internally by WIRClient and WIRServer to request to get notification on/off status. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					



R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success
R	on	Boolean	-	0/1	On or off

2.9.1.28 MD-REQ-380298/A-WlanNQM

This API is used internally by WIRClient and WIRServer to request a WLAN network quality measurement. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	EcuType	Enum	-	-	ECU type
			ECU_UNK	0x0	Error
			ECU_ECG	0x1	ECG
			ECU_TCU	0x2	TCU
			ECU_SYNC	0x3	SYNC
			ECU_TCU_B	0x4	TCU-B
			ECU_VDS	0x5	VDS
			ECU_AVPIIM	0x6	AVPIIM
			ECU_DDSDM	0x7	DDSDM
R	NetworkInterfaceType	Enum	-	-	
			IFACE_NONE	0x0	None selected/specified
			IFACE_TCU_CELL	0x1	Cellular interface on TCU
			IFACE_TCU_WIFI	0x2	WLAN interface on TCU
			IFACE_SYNC_WIFI	0x3	WLAN interface on SYNC
			IFACE_SYNC_CAPPL	0x4	AppLink interface on SYNC (currently out of scope for CM)
			IFACE_TCU_BCELL	0x5	Cellular interface on TCU-B
			IFACE_TCU_BWIFI	0x6	WLAN interface on TCU-B
			IFACE_VDS_LAN	0x7	VDS LAN
			IFACE_ERR	0x8	Error condition
Response					
R	NqmRet	Enum	-	-	



			NQM_ERRO R	0x0	Error/Failure
			NQM_SUCC ESS	0x1	Success
R	EcuType	Enum	-	-	ECU type
			ECU_UNK	0x0	Error
			ECU_ECG	0x1	ECG
			ECU_TCU	0x2	TCU
			ECU_SYNC	0x3	SYNC
			ECU_TCU_B	0x4	TCU-B
			ECU_VDS	0x5	VDS
			ECU_AVPIIM	0x6	AVPIIM
			ECU_DDSDM	0x7	DDSDM
R	NetworkInterfaceType	Enum	-	-	
			IFACE_NON E	0x0	None selected/specified
			IFACE_TCU CELL	0x1	Cellular interface on TCU
			IFACE_TCU WIFI	0x2	WLAN interface on TCU
			IFACE_SYN CWIFI	0x3	WLAN interface on SYNC
			IFACE_SYN CAPPL	0x4	AppLink interface on SYNC (currently out of scope for CM)
			IFACE_TCU BCELL	0x5	Cellular interface on TCU-B
			IFACE_TCU BWIFI	0x6	WLAN interface on TCU-B
			IFACE_VDS LAN	0x7	VDS LAN
			IFACE_ERR	0x8	Error condition
R	WlanStaSmState	Enum	-	-	
			WLAN_STA_ SM_STATE_ OFF	0x0	WLAN off
			WLAN_STA_ SM_STATE_ ENABLING	0x1	Enabling
			WLAN_STA_ SM_STATE_ ON	0x2	On
			WLAN_STA_ SM_STATE_ CONNECTIN G	0x3	Connecting
			WLAN_STA_ SM_STATE_ CONNECTE D	0x4	Connected
			WLAN_STA_ SM_STATE_ DISCONNEC TING	0x5	Disconnecting
			WLAN_STA_ SM_STATE_ DISABLING	0x6	Disabling



			WLAN_STA_ SM_STATE_ ERROR	0x7	Error
			WLAN_STA_ SM_STATE_ AUTHERRO R	0x8	Authentication error
			WLAN_STA_ SM_STATE_ NWNOTFOU ND	0x9	Network not found
R	macAddr	String	-	Char Value:0-255 No String length limit	MAC Address
R	WlanStaConnState	Enum	-	-	
			WLAN_STA_ CON_STATE_ DISCONNE CTED	0x0	Disconnected
			WLAN_STA_ CON_STATE_ WPS_ACTI VE	0x1	WPS pending
			WLAN_STA_ CON_STATE_ ASSOCIATI NG	0x2	Associating
			WLAN_STA_ CON_STATE_ IP_ADDRE SSING	0x3	Getting IP address
			WLAN_STA_ CON_STATE_ CONNECT ED	0x4	Connected
			WLAN_STA_ CON_STATE_ DISCONNE CTING	0x5	Disconnecting
			WLAN_STA_ CON_STATE_ AUTHERR OR	0x6	Authentication error
			WLAN_STA_ CON_STATE_ NWNOTFO UND	0x7	Network not found
R	connSsid	String	-	Char Value:0-255	Connected AP's SSID
R	connBssid	String	-	No String length limit	Connected AP's BSSID
R	conn80211Tech	String	-	Char Value:0-255	Connected AP's 802.11 technology, 802.11a, ...
R	connChannel	Int32	-	0-4294967295	Connected AP's channel
R	bandwidth	Int32	-	0-4294967295	Connected AP's bandwidth



R	dataRate	Int32	-	0-4294967295	Connected AP's data rate (bps)
R	signalStrength	Int32	-	0-4294967295	RSSI (dBm)
R	signalNoise	Int32	-	0-4294967295	Noise level (dBm)
R	signalSNR	Int32	-	0-4294967295	SNR (dB)
R	ipAddr	String	-	Char Value:0-255 No String length limit	IP address
R	ipSubnet	String	-	Char Value:0-255 No String length limit	Subnet mask
R	ipGateway	String	-	Char Value:0-255 No String length limit	GW IP address
R	ipDnsPref	String	-	Char Value:0-255 No String length limit	IP address of primary DNS
R	ipDnsAlt	String	-	Char Value:0-255 No String length limit	IP address of secondary DNS

2.9.1.29 MD-REQ-380300/A-WlanNQMInd

This API is used internally by WIRClient and WIRServer to broadcast an update of WLAN network quality measurement.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	-
Response					
R	EcuType	Enum	-	-	ECU type
			ECU_UNK	0x0	Error
			ECU_ECG	0x1	ECG
			ECU_TCU	0x2	TCU
			ECU_SYNC	0x3	SYNC
			ECU_TCU_B	0x4	TCU-B
			ECU_VDS	0x5	VDS
			ECU_AVPIIM	0x6	AVPIIM
			ECU_DDSDM	0x7	DDSDM
R	NetworkInterfaceType	Enum	-	-	
			IFACE_NON E	0x0	None selected/specified
			IFACE_TCU CELL	0x1	Cellular interface on TCU
			IFACE_TCU WIFI	0x2	WLAN interface on TCU
			IFACE_SYN CWIFI	0x3	WLAN interface on SYNC



			IFACE_SYN CAPPL	0x4	AppLink interface on SYNC (currently out of scope for CM)
			IFACE_TCU BCELL	0x5	Cellular interface on TCU-B
			IFACE_TCU BWIFI	0x6	WLAN interface on TCU-B
			IFACE_VDS LAN	0x7	VDS LAN
			IFACE_ERR	0x8	Error condition
R	WlanStaSmState	Enum	-	-	
			WLAN_STA_ SM_STATE_ OFF	0x0	WLAN off
			WLAN_STA_ SM_STATE_ ENABLING	0x1	Enabling
			WLAN_STA_ SM_STATE_ ON	0x2	On
			WLAN_STA_ SM_STATE_ CONNECTIN G	0x3	Connecting
			WLAN_STA_ SM_STATE_ CONNECTE D	0x4	Connected
			WLAN_STA_ SM_STATE_ DISCONNEC TING	0x5	Disconnecting
			WLAN_STA_ SM_STATE_ DISABLING	0x6	Disabling
			WLAN_STA_ SM_STATE_ ERROR	0x7	Error
			WLAN_STA_ SM_STATE_ AUTHERROR	0x8	Authentication error
			WLAN_STA_ SM_STATE_ NWNOTFOU ND	0x9	Network not found
R	macAddr	String	-	Char Value:0-255 No String length limit	MAC Address
R	WlanStaConnState	Enum	-	-	
			WLAN_STA_ CON_STATE_ DISCONN CTED	0x0	Disconnected
			WLAN_STA_ CON_STATE_ WPS_ACTI VE	0x1	WPS pending



			WLAN_STA_CON_STATE_ASSOCIATING	0x2	Associating
			WLAN_STA_CON_STATE_IP_ADDRESSING	0x3	Getting IP address
			WLAN_STA_CON_STATE_CONNECTED	0x4	Connected
			WLAN_STA_CON_STATE_DISCONNECTING	0x5	Disconnecting
			WLAN_STA_CON_STATE_AUTHERROR	0x6	Authentication error
			WLAN_STA_CON_STATE_NWNOTFOUND	0x7	Network not found
R	connSsid	String	-	Char Value:0-255 No String length limit	Connected AP's SSID
R	connBssid	String	-	Char Value:0-255 No String length limit	Connected AP's BSSID
R	conn80211Tech	String	-	Char Value:0-255 No String length limit	Connected AP's 802.11 technology, 802.11a, ...
R	connChannel	Int32	-	0- 4294967295	Connected AP's channel
R	bandwidth	Int32	-	0- 4294967295	Connected AP's bandwidth
R	dataRate	Int32	-	0- 4294967295	Connected AP's data rate (bps)
R	signalStrength	Int32	-	0- 4294967295	RSSI (dBm)
R	signalNoise	Int32	-	0- 4294967295	Noise level (dBm)
R	signalSNR	Int32	-	0- 4294967295	SNR (dB)
R	ipAddr	String	-	Char Value:0-255 No String length limit	IP address
R	ipSubnet	String	-	Char Value:0-255 No String length limit	Subnet mask
R	ipGateway	String	-	Char Value:0-255	GW IP address



				No String length limit	
R	ipDnsPref	String	-	Char Value:0-255 No String length limit	IP address of primary DNS
R	ipDnsAlt	String	-	Char Value:0-255 No String length limit	IP address of secondary DNS

2.9.1.30 MD-REQ-380304/A-ScanWiFiNetworks

This API is used internally by WIRClient and WIRServer to request a WiFi network scan. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	ifld	Int32	-	0-4294967295	Interface ID
R	passive	Boolean	-	0/1	true = passive scan (no pkts tx'd), false = active scan
R	WlanChannel	Enum	-	-	List of channels of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	ssid	String	-	Char Value:0-255 No String length limit	List of SSIDs to specifically scan for. Used to scan for hidden networks. "passive" will be ignored as an active scan is required
R	scanId	Int64	-	0-18446744073709551615	ID value used to track scans
Response					
R	WlanRet	Enum	-	-	
			WLAN_ERROR	0x0	Error/Failure
			WLAN_SUCCESS	0x1	Success
R	ifld	Int32	-	0-4294967295	Interface ID
R	scanId	Int64	-	0-18446744073709551615	ID value used to track scans
R	WlanChannel	Enum	-	-	List of channels of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	



			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	ssid	String	-	Char Value:0-255 No String length limit	List of SSIDs to specifically scan for. Used to scan for hidden networks. "passive" will be ignored as an active scan is required
R	count	Int32	-	0-4294967295	number of APs found, repeats ap field
R	ssid	String	-	Char Value:0-255 No String length limit	SSID – scan result
R	bssid	String	-	Char Value:0-255 No String length limit	MAC address of AP – scan result
R	WlanChannel	Enum	-	-	Channel of AP – scan result
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	Wlan Bandwidth	Enum	-	-	
			WLAN_BW_MIN	0x0	
			WLAN_BW_MHZ20	0x1	20MHz channel
			WLAN_BW_MHZ40	0x2	40MHz channel; must use 11n/11ac. Few devices support on 2.4Ghz
			WLAN_BW_MHZ80	0x3	80MHz channel; must use 11ac
			WLAN_BW_MHZ8080	0x4	80-80MHz, 2 non-contiguous 80MHz; must use 11ac
			WLAN_BW_MHZ160	0x5	160MHz channel; must 11ac
			WLAN_BW_MAX	0x6	
R	rsi	Int32	-	0-4294967295	RSSI of beacon/probe response
R	WlanSecurity	Enum	-	-	Security settings of scan results
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)



			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	isWpsSupported	Boolean	-	0/1	does AP support WPS
R	isEssSupported	Boolean	-	0/1	is AP part of an Extended Service Set
R	scanResultsComplete	Boolean	-	0/1	All the scan results have been reported or not

2.9.1.31 MD-REQ-380305/A-ConnectWiFiAP

This API is used internally by WIRClient and WIRServer to request to connect to an AP. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	ifld	Int32	-	0-4294967295	Interface ID
R	ssid	String	-	Char Value:0-255 No String length limit	SSID of network
R	bssid	String	-	Char Value:0-255 No String length limit	Optional - BSSID of AP
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	WlanSecurity	Enum	-	-	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanWepSecurity	-	-	-	WEP Settings
		String	Key	Char Value:0-255	



				No String length limit	
		Int32	defaultKeyIndex	0-4294967295	
R	WlanWpsSecurity	-	-	-	WPS Settings
		WlanWpsType	-		Type
		String	-	Char Value:0-255 No String length limit	Pin
R	WlanWpaPersonalSecurity	String	-	Char Value:0-255 No String length limit	WPA/WPA2-Personal Settings - password
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None
			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES
			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanWpaEnterprise	TBD	-		TBD
R	WlanIpv4AddrType	Enum	-		
			WLAN_IPV4_ADDR_MIN	0x0	
			WLAN_IPV4_ADDR_NONE	0x1	No IPv4 Addressing is used
			WLAN_IPV4_ADDR_STATIC	0x2	Static IPv4 Address
			WLAN_IPV4_ADDR_DHCP_CLIENT	0x3	DHCP Client IPv4 Address
			WLAN_IPV4_ADDR_DHCP_SERVER	0x4	DHCP Server IPv4 Address
			WLAN_IPV4_ADDR_MAX	0x5	
R	WlanIpv4Addr	-	-	-	
		String	Ip	Char Value:0-255	IP address of current connection



				No String length limit	
		String	Netmask	Char Value:0-255 No String length limit	Netmask of current connection
		String	Gateway	Char Value:0-255 No String length limit	default gateway of current connection
		String	dnsPref	Char Value:0-255 No String length limit	Preferred DNS server
		String	dnsAlt	Char Value:0-255 No String length limit	Secondary DNS server
R	WlanIpv6AddrType	Enum	-	-	
			WLAN_IPV6_ADDR_MIN	0x0	
			WLAN_IPV6_ADDR_NONE	0x1	No IPv6 Addressing is used
			WLAN_IPV6_ADDR_STATIC	0x2	Static IPv6 Address
			WLAN_IPV6_ADDR_MAX	0x3	
R	WlanIpv6Addr	TBD	-	-	
R	ExclusiveUse	Boolean	-	0/1	Exclusive to WIR Client
R	WlanProfileSource	Enum	-	-	Origin of the profile
			PROFSRC_DEFAULT	0x0	Default internal
			PROFSRC_HMI	0x1	HMI/User configured
			PROFSRC_CLOUD	0x2	Cloud pushed
			PROFSRC_SPECIAL	0x3	Special Intent
			PROFSRC_PROVIDED	0x4	EOL provisioned
R	timestamp	Int64	-	0-18446744073709551615	Last known time stamp
R	hidden	Boolean	-	0/1	Hidden AP
Response					
R	WlanRet	Enum	-	-	
			WLAN_ERROR	0x0	Error/Failure



			WLAN_SUCCE SS	0x1	Success
R	ifld	Int32	-	0- 42949672 95	Interface ID
R	ifaceName	String	-	Char Value:0- 255 No String length limit	Interface Name

2.9.1.32 MD-REQ-380306/A-DisconnectWiFiAP

This API is used internally by WIRClient and WIRServer to request to disconnect from an AP. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	ifld	Int32	-	0- 42949672 95	Interface ID
Response					
R	WlanRet	Enum	-	-	
			WLAN_ERROR	0x0	Error/Failure
			WLAN_SUCCE SS	0x1	Success
R	ifld	Int32	-	0- 42949672 95	Interface ID

2.9.1.33 MD-REQ-380307/A-ProfileUpdate

This API is used internally by WIRClient and WIRServer to request to update a profile. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	txld	String	-	Char Value:0- 255 No String length limit	Transaction ID
Response					
R	WlanRet	Enum	-	-	
			WLAN_ERROR	0x0	Error/Failure
			WLAN_SUCCE SS	0x1	Success
R	txld	String	-	Char Value:0- 255	Transaction ID



				No String length limit	
R	updateComplete	Boolean	-	0/1	Is it last segment?
R	profile_cnt	String	-	Char Value:0-255 No String length limit	Number of entries of following 2 fields
R	ssid	String	-	Char Value:0-255 No String length limit	SSID
R	bssid	String	-	Char Value:0-255 No String length limit	Optional - BSSID of AP
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	WlanSecurity	Enum	-	-	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanWepSecurity	-	-	-	WEP Settings
		String	Key	Char Value:0-255 No String length limit	
		Int32	defaultKeyIndex	0-4294967295	
R	WlanWpsSecurity	-	-	-	WPS Settings
		WlanWpsType	-	-	Type
		String	-	Char Value:0-255	Pin



				No String length limit	
R	WlanWpaPersonalSecurity	String	-	Char Value:0-255 No String length limit	WPA/WPA2-Personal Settings - password
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None
			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES
			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanWpaEnterprise	TBD	-		TBD
R	WlanIpv4AddrType	Enum	-		
			WLAN_IPV4_ADDR_MIN	0x0	
			WLAN_IPV4_ADDR_NONE	0x1	No IPv4 Addressing is used
			WLAN_IPV4_ADDR_STATIC	0x2	Static IPv4 Address
			WLAN_IPV4_ADDR_DHCP_CLIENT	0x3	DHCP Client IPv4 Address
			WLAN_IPV4_ADDR_DHCP_SERVER	0x4	DHCP Server IPv4 Address
			WLAN_IPV4_ADDR_MAX	0x5	
R	WlanIpv4Addr	-	-	-	
		String	Ip	Char Value:0-255 No String length limit	IP address of current connection
		String	Netmask	Char Value:0-255 No String length limit	Netmask of current connection
		String	Gateway	Char Value:0-255	default gateway of current connection



				No String length limit	
		String	dnsPref	Char Value:0-255 No String length limit	Preferred DNS server
		String	dnsAlt	Char Value:0-255 No String length limit	Secondary DNS server
R	WlanIpv6AddrType	Enum	-	-	
			WLAN_IPV6_A DDR_MIN	0x0	
			WLAN_IPV6_A DDR_NONE	0x1	No IPv6 Addressing is used
			WLAN_IPV6_A DDR_STATIC	0x2	Static IPv6 Address
			WLAN_IPV6_A DDR_MAX	0x3	
R	WlanIpv6Addr	TBD	-	-	
R	ExclusiveUse	Boolean	-	0/1	Exclusive to WIR Client
R	WlanProfileSource	Enum	-	-	Origin of the profile
			PROFSRC_DE FAULT	0x0	Default internal
			PROFSRC_HMI	0x1	HMI/User configured
			PROFSRC_CLOUD	0x2	Cloud pushed
			PROFSRC_SPECIAL	0x3	Special Intent
			PROFSRC_PROVID	0x4	EOL provisioned
R	timestamp	Int64	-	0-18446744073709551615	Last known time stamp
R	hidden	Boolean	-	0/1	Hidden AP

2.9.1.34 MD-REQ-380308/A-ConnectionStatus

This API is used internally by WIRClient and WIRServer to request connection status. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	ssid	String	-	Char Value:0-255	SSID



				No String length limit	
R	bssid	String	-	Char Value:0-255 No String length limit	BSSID, MAC address
R	ifId	Int32	-	0-4294967295	Interface ID

Response

R	WlanRet	Enum	-	-	
			WLAN_ERROR	0x0	Error/Failure
			WLAN_SUCCESS	0x1	Success
R	ssid	String	-	Char Value:0-255 No String length limit	SSID
R	bssid	String	-	Char Value:0-255 No String length limit	BSSID, MAC address
R	WlanStaConnState	Enum	-	-	
			WLAN_STA_CONNECTION_DISCONNECTED	0x0	Disconnected
			WLAN_STA_CONNECTION_WPS_ACTIVE	0x1	WPS pending
			WLAN_STA_CONNECTION_ASSOCIATING	0x2	Associating
			WLAN_STA_CONNECTION_IP_ADDRESSING	0x3	Getting IP address
			WLAN_STA_CONNECTION_CONNECTED	0x4	Connected
			WLAN_STA_CONNECTION_DISCONNECTING	0x5	Disconnecting
			WLAN_STA_CONNECTION_AUTH_ERROR	0x6	Authentication error
			WLAN_STA_CONNECTION_NOT_FOUND	0x7	Network not found
R	WlanStaSmState	Enum	-	-	
			WLAN_STA_STATE_OFF	0x0	WLAN off



			WLAN_STA_S M_STATE_ENA BLING	0x1	Enabling
			WLAN_STA_S M_STATE_ON	0x2	On
			WLAN_STA_S M_STATE_CON NECTING	0x3	Connecting
			WLAN_STA_S M_STATE_CON NECTED	0x4	Connected
			WLAN_STA_S M_STATE_DIS CONNECTING	0x5	Disconnecting
			WLAN_STA_S M_STATE_DIS ABLING	0x6	Disabling
			WLAN_STA_S M_STATE_ERR OR	0x7	Error
			WLAN_STA_S M_STATE_AUT HERROR	0x8	Authentication error
			WLAN_STA_S M_STATE_NW NOTFOUND	0x9	Network not found
R	ifId	Int32	-	0- 42949672 95	Interface ID

2.9.1.35 MD-REQ-380310/A-StationModeStats

This API is used internally by WIRClient and WIRServer to request WLAN station mode statistics. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	ifId	Int32	-	0- 42949672 95	Interface ID
R	ifaceName	String	-	Char Value:0- 255 No String length limit	Interface Name
R	StatsCmdType	Enum	-	-	
			DISABLE	0x0	Disable
			ENABLE	0x1	Enable
			QUERY	0x2	Query current state
R	reportingWindow	Int32	-	0- 42949672 95	Seconds
Response					



R	ifId	Int32	-	0-4294967295	Interface ID
R	ifaceName	String	-	Char Value:0-255 No String length limit	Interface Name
R	StatsCmdType	Enum	-	-	
			DISABLE	0x0	Disable
			ENABLE	0x1	Enable
			QUERY	0x2	Query current state
R	WlanRet	Enum	-	-	
			WLAN_ERROR	0x0	Error/Failure
			WLAN_SUCCESS	0x1	Success
R	pktsTxV4	Int64	-	0-18446744073709551615	Number of packets transmitted
R	pktsRxV4	Int64	-	0-18446744073709551615	Number of packets received
R	bytesTxV4	Int64	-	0-18446744073709551615	Number of bytes transmitted
R	bytesRxV4	Int64	-	0-18446744073709551615	Number of bytes received
R	PktsDroppedTx	Int64	-	0-18446744073709551615	Number of dropped packets transmitted
R	PktsDroppedRx	Int64	-	0-18446744073709551615	Number of dropped packets received

2.9.1.36 MD-REQ-380312/A-ProfileUpdateInd

This API is used internally by WIRClient and WIRServer to broadcast a profile update indication.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	-
Response					
R	WlanProfileAction	Enum	-	-	
			ACT_UPDATE	0x0	Add/update
			ACT_REMOVE	0x1	Remove



R	ssid	String	-	Char Value:0-255 No String length limit	SSID of network
R	bssid	String	-	Char Value:0-255 No String length limit	Optional - BSSID of AP
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	WlanSecurity	Enum	-	-	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanWepSecurity	-	-	-	WEP Settings
		String	Key	Char Value:0-255 No String length limit	
		Int32	defaultKeyIndex	0-4294967295	
R	WlanWpsSecurity	-	-	-	WPS Settings
		WlanWpsType	-		Type
		String	-	Char Value:0-255 No String length limit	Pin
R	WlanWpaPersonalSecurity	String	-	Char Value:0-255 No String length limit	WPA/WPA2-Personal Settings - password
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None



			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES
			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanWpaEnterprise	TBD	-		TBD
R	WlanIpv4AddrType	Enum	-		
			WLAN_IPV4_ADDR_MIN	0x0	
			WLAN_IPV4_ADDR_NONE	0x1	No IPv4 Addressing is used
			WLAN_IPV4_ADDR_STATIC	0x2	Static IPv4 Address
			WLAN_IPV4_ADDR_DHCP_CLIENT	0x3	DHCP Client IPv4 Address
			WLAN_IPV4_ADDR_DHCP_SERVER	0x4	DHCP Server IPv4 Address
			WLAN_IPV4_ADDR_MAX	0x5	
R	WlanIpv4Addr	-	-	-	
		String	Ip	Char Value:0-255 No String length limit	IP address of current connection
		String	Netmask	Char Value:0-255 No String length limit	Netmask of current connection
		String	Gateway	Char Value:0-255 No String length limit	default gateway of current connection
		String	dnsPref	Char Value:0-255 No String length limit	Preferred DNS server
		String	dnsAlt	Char Value:0-255	Secondary DNS server



				No String length limit	
R	WlanIpv6AddrType	Enum	-	-	
			WLAN_IPV6_ADDR_MIN	0x0	
			WLAN_IPV6_ADDR_NONE	0x1	No IPv6 Addressing is used
			WLAN_IPV6_ADDR_STATIC	0x2	Static IPv6 Address
			WLAN_IPV6_ADDR_MAX	0x3	
R	WlanIpv6Addr	TBD	-	-	
R	ExclusiveUse	Boolean	-	0/1	Exclusive to WIR Client
R	WlanProfileSource	Enum	-	-	Origin of the profile
			PROFSRC_DEFAULT	0x0	Default internal
			PROFSRC_HMI	0x1	HMI/User configured
			PROFSRC_CLOUD	0x2	Cloud pushed
			PROFSRC_SPECIAL	0x3	Special Intent
			PROFSRC_PROVIDED	0x4	EOL provisioned
R	timestamp	Int64	-	0-18446744073709551615	Last known time stamp
R	hidden	Boolean	-	0/1	Hidden AP

2.9.1.37 MD-REQ-380311/A-StationModeStatsInd

This API is used internally by WIRClient and WIRServer to broadcast an update of WLAN station mode statistics.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	-
Response					
R	ifId	Int32	-	0-4294967295	Interface ID
R	ifaceName	String	-	Char Value:0-255 No String length limit	Interface Name
R	StatsCmdType	Enum	-	-	
			DISABLE	0x0	Disable
			ENABLE	0x1	Enable
			QUERY	0x2	Query current state
R	WlanRet	Enum	-	-	
			WLAN_ERROR	0x0	Error/Failure



			WLAN_SUCCESS	0x1	Success
R	pktsTxV4	Int64	-	0-18446744 07370955 1615	Number of packets transmitted
R	pktsRxV4	Int64	-	0-18446744 07370955 1615	Number of packets received
R	bytesTxV4	Int64	-	0-18446744 07370955 1615	Number of bytes transmitted
R	bytesRxV4	Int64	-	0-18446744 07370955 1615	Number of bytes received
R	PktsDroppedTx	Int64	-	0-18446744 07370955 1615	Number of dropped packets transmitted
R	PktsDroppedRx	Int64	-	0-18446744 07370955 1615	Number of dropped packets received

2.9.1.38 MD-REQ-380313/A-WlanStateInd

This API is used internally by WIRClient and WIRServer to broadcast a WLAN station mode state update.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	-
Response					
R	wlanState	Boolean	-	0/1	WLAN On/Off
R	wlanNotifState	Int32	-	-	Notification On/Off

2.9.1.39 MD-REQ-380315/A-SignalUpdateInd

This API is used internally by WIRClient and WIRServer to broadcast a WLAN signal update.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	-
Response					
R	ifId	Int32	-	0-42949672 95	Interface ID



R	rssi	Int32	-	0-4294967295	RSSI of the currently connected AP
---	------	-------	---	--------------	------------------------------------

2.9.2 WIR-IIR-REQ-378577/A-WIRClient1_Rx

2.9.2.1 MD-REQ-027149/A-IgnitionStatus_St (TcSE ROIN-225464-1)

Message Type: Status

Signal used to indicate ignition state.

Name	Literals	Value	Description
Type	-	-	Indicates ignition state
	Unknown	0x0	
	Off	0x1	
	Accessory	0x2	
	Run	0x4	
	Start	0x8	
	Invalid	0xF	

2.9.2.2 MD-REQ-199634/A-BSBattSOC

Message Type: Status

Status used to indicate the vehicle battery's state of charge.

Name	Literals	Value	Description
Type	-	-	Battery's state of charge in percent
	0-127	0x0 - 0x7F	

2.9.2.3 MD-REQ-201601/A-Delay_Accy

Message Type: Status

This signal is used indicate whether Delayed Accessory is active or not.

Name	Literals	Value	Description
Type	-	-	Status of delayed accessory
	Off	0x00	
	On	0x01	

2.9.2.4 MD-REQ-028253/A-GearLeverPosition_St (TcSE ROIN-282103-1)

Message Type: Status

Status used to indicate the current gear selected.

Name	Literals	Value	Description
Type	-	-	Used to indicate current gear selected.
	Park	0x0	



	Reverse	0x1	
	Neutral	0x2	
	Drive	0x3	
	Sport_DriveSport	0x4	
	Low	0x5	
	First	0x6	
	Second	0x7	
	Third	0x8	
	Fourth	0x9	
	Fifth	0xA	
	Sixth	0xB	
	Undefined_Treat_as_Fault	0xC	
	Undefined_Treat_as_Fault1	0xD	
	Unknown_Position	0xE	
	Fault	0xF	

2.9.2.5 MD-REQ-014025/A-VehicleSpeed_St (TcSE ROIN-223023-1)

Message Type: Status

Status used to indicate vehicle speed.

Name	Literals	Value	Description
Type	-	-	Indicates vehicle speed. Unit: kph Resolution:0.01 Offset:0
	kph	0x0 to 0xFFFF	

2.9.2.6 MD-REQ-304019/A-AutosarNM

Message Type: Status

Autosar signal used to wake up the CAN bus.

Name	Literals	Value	Description
Control	-	-	-
		0x00-0xFF	
NodeID	-	-	-
		0x00-0xFF	
NMReserved1	-	-	-
		0x00-0xFF	
NMReserved2	-	-	-
		0x00-0xFF	
NMReserved3	-	-	-
		0x00-0xFF	
NMReserved4	-	-	-
		0x00-0xFF	
GWNMProxy	-	-	-
		0x00-0xFF	
GWOnBoardTester	-	-	-
		0x00-0xFF	

**2.9.2.7 MD-REQ-304038/A-VehicleMode**

Message Type: Status

Signal to manage the key off load of the vehicle when the Ignition is OFF.

Name	Literals	Value	Description
Type	-	-	-
	NormalPower	0x0	
	FactoryPower	0x1	
	TransportPower	0x2	
	Hibernate	0x3	
	CriticalBattery	0x4	
	NotUsed	0x5 - 0xF	

2.9.2.8 MD-REQ-380258/A-VlanAdd

This API is used internally by WIRClient and WIRServer to request a new VLAN configuration. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
R	origEcu	String	-	Char Value:0-255 No String length limit	Origination ECU enum in string format
R	ifaceEcu	String	-	Char Value:0-255 No String length limit	Interface ECU enum in string format



R	mtu	String	-	Char Value:0-255 No String length limit	Interface MTU in string format
R	svcLevel	String	-	Char Value:0-255 No String length limit	Service Level of the VLAN in string format
R	ifaceType	Enum	-	-	Network Interface Type over which the VLAN would be setup
			None	0x0	
			TcuCell	0x1	
			TcuWifi	0x2	
			SyncWifi	0x3	
			SyncApplink	0x4	
			TcubCell	0x5	
			TcubWifi	0x6	
			VdsLan	0x7	
			Error	0x8	
R	vlanEndpointType	Enum	-	-	VLAN Endpoint type
			Host	0x0	Host type endpoint
			Gateway	0x1	Gateway type endpoint
Response					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	
			VLAN_SUCCESS	0x1	
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
R	ipV4Address	String	-	Char Value:0-255 No String length limit	VLAN IP Address

**2.9.2.9 MD-REQ-380259/A-VlanRemove**

This API is used internally by WIRClient and WIRServer to request removal of old VLAN configuration. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
Response					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	
			VLAN_SUCCESS	0x1	
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id

2.9.2.10 MD-REQ-380260/A-VlanLink

This API is used internally by WIRClient and WIRServer to request link VLAN. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			



R/O	Name	Type	Literals	Value	Description
Request					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
R	ifaceName	String	-	Char Value:0-255 No String length limit	Network Interface as enumerated in the OS
R	serviceLevel	String	-	Char Value:0-255 No String length limit	Service Level of the VLAN in string format
Response					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	
			VLAN_SUCCESS	0x1	
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
R	ifaceName	String	-	Char Value:0-255	Network Interface as enumerated in the OS



				No String length limit	
R	serviceLevel	String	-	Char Value:0-255 No String length limit	Service Level of the VLAN in string format
R	linkId	String	-	Char Value:0-255 No String length limit	Unique link ID generated by Networking

2.9.2.11 MD-REQ-380261/A-VlanUnlink

This API is used internally by WIRClient and WIRServer to request unlink VLAN. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
R	linkId	String	-	Char Value:0-255 No String length limit	Unique link ID generated by Networking
Response					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	vlanEndpointId	Int32	-	0-4294967295	Endpoint Id used internally by CM
R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	



			VLAN_SUCCESS	0x1	
R	vlanName	String	-	Char Value:0-255 No String length limit	Unique Vlan name
R	vlanId	String	-	Char Value:0-255 No String length limit	Unique Vlan Id
R	linkId	String	-	Char Value:0-255 No String length limit	Unique link ID generated by Networking

2.9.2.12 MD-REQ-380262/A-VnmReset

This API is used internally by WIRClient and WIRServer to request a VLAN configuration reset. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	EcuType	Enum	-	-	Path Id used internally by CM
			Unknown	0x0	Error
			ECG	0x1	
			TCU	0x2	
			SYNC	0x3	
			TCU_B	0x4	
			VDS	0x5	
			AVPIM	0x6	
			DDSM	0x7	
Response					
R	EcuType	Enum	-	-	Path Id used internally by CM
			Unknown	0x0	Error
			ECG	0x1	
			TCU	0x2	
			SYNC	0x3	
			TCU_B	0x4	
			VDS	0x5	
			AVPIM	0x6	
			DDSM	0x7	
R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	
			VLAN_SUCCESS	0x1	

**2.9.2.13 MD-REQ-380263/A-VnmLinkNetwork**

This API is used internally by WIRClient and WIRServer to request to link a network to VLAN. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	origEcu	String	-	Char Value:0-255 No String length limit	Origination ECU enum in string format
R	ifaceEcu	String	-	Char Value:0-255 No String length limit	Interface ECU enum in string format
R	interfaceZero	String	-	Char Value:0-255 No String length limit	User's provided interface name to be linked
R	interfaceOne	String	-	Char Value:0-255 No String length limit	Network interface to be linked
R	ipAddr	String	-	Char Value:0-255 No String length limit	Gateway ip address associated with network interface
Response					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	origEcu	String	-	Char Value:0-255 No String length limit	Origination ECU enum in string format
R	linkId	String	-	Char Value:0-255 No String length limit	Unique link ID generated by Networking



R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	
			VLAN_SUCCESS	0x1	

2.9.2.14 MD-REQ-380264/A-VnmUnlinkNetwork

This API is used internally by WIRClient and WIRServer to request to unlink a network to VLAN. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	origEcu	String	-	Char Value:0-255 No String length limit	Origination ECU enum in string format
R	ifaceEcu	String	-	Char Value:0-255 No String length limit	Interface ECU enum in string format
R	linkId	String	-	Char Value:0-255 No String length limit	Unique link ID generated by Networking
Response					
R	pathId	Int32	-	0-4294967295	Path Id used internally by CM
R	origEcu	String	-	Char Value:0-255 No String length limit	Origination ECU enum in string format
R	VlanRet	Enum	-	-	Return code for command
			VLAN_ERROR	0x0	
			VLAN_SUCCESS	0x1	

2.9.2.15 MD-REQ-380265/A-NetworkInterfaceAllocation

This API is used internally by WIRClient and WIRServer to request a new network interface. WIR also uses this API for its response. The response will return success or failure, actual interface is returned through NetworkInterfaceAllocationStatusInd.

Method Type	One-Shot (A-Synch)
QoS Level	Default



Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	allocId	Int32	-	0-4294967295	Allocation ID will be assigned by WIR
R	IntentType	Enum	-	-	Intent Type
			Foreground	0x0	
			Background_BestEffort	0x1	
			Background_Guaranteed	0x2	
			Special	0x3	
			OffPeak	0x4	
			LAN_Only	0x5	
R	InterfaceType	Enum	-	-	Interface Type
			None	0x0	
			TcuCell	0x1	
			TcuWifi	0x2	
			SyncWifi	0x3	
			SyncApplink	0x4	
			TcubCell	0x5	
			TcubWifi	0x6	
			VdsLan	0x7	
			Error	0x5	
R	CellApnType	Enum	-	-	Cellular APN Type
			Internet	0x0	
			Ford	0x1	
			Tethering	0x2	
			FOTA	0x3	
			HTTP	0x4	
R	PriorityLevel	Enum	-	-	Priority Level
			Priority 0	0x0	
			Priority 1	0x1	
			Priority 2	0x2	
			Priority 3	0x3	
			Priority 4	0x4	
R	expiry	Int32	-	0-4294967295	Expiration timer
R	OffPeakFlag	Enum	-	-	Offpeak flag
			No	0x0	
			Yes	0x1	
R	WifiPreferredFlag	Enum	-	-	Wifi preferred flag
			No	0x0	
			Yes	0x1	
R	LanPreferredFlag	Enum	-	-	LAN preferred flag
			No	0x0	
			Yes	0x1	



R	ssid	String	-	Char Value:0-255 No String length limit	SSID of network
R	bssid	String		Char Value:0-255 No String length limit	Optional - BSSID of AP
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	WlanSecurity	Enum	-	-	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanWepSecurity				WEP Settings
		String	Key	Char Value:0-255 No String length limit	
		Int32	defaultKeyIndex	0-4294967295	
R	WlanWpsSecurity				WPS Settings
		WlanWpsType			Type
		String		Char Value:0-255 No String length limit	Pin
R	WlanWpaPersonalSecurity	String		Char Value:0-255 No String length limit	WPA/WPA2-Personal Settings - password
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None



			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES
			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanWpaEnterprise	TBD	-	-	TBD
R	WlanIpv4AddrType	Enum	-	-	
			WLAN_IPV4_ADDR_MIN	0x0	
			WLAN_IPV4_ADDR_NONE	0x1	No IPv4 Addressing is used
			WLAN_IPV4_ADDR_STATIC	0x2	Static IPv4 Address
			WLAN_IPV4_ADDR_DHCP_CLIENT	0x3	DHCP Client IPv4 Address
			WLAN_IPV4_ADDR_DHCP_SERVER	0x4	DHCP Server IPv4 Address
			WLAN_IPV4_ADDR_MAX	0x5	
R	WlanIpv4Addr	-	-	-	
		String	Ip	Char Value:0-255 No String length limit	IP address of current connection
		String	Netmask	Char Value:0-255 No String length limit	Netmask of current connection
		String	Gateway	Char Value:0-255 No String length limit	default gateway of current connection
		String	dnsPref	Char Value:0-255 No String length limit	Preferred DNS server
		String	dnsAlt	Char Value:0-255	Secondary DNS server



				No String length limit	
R	WlanIpv6AddrType	Enum	-	-	
			WLAN_IPV6_A DDR_MIN	0x0	
			WLAN_IPV6_A DDR_NONE	0x1	No IPv6 Addressing is used
			WLAN_IPV6_A DDR_STATIC	0x2	Static IPv6 Address
			WLAN_IPV6_A DDR_MAX	0x3	
R	WlanIpv6Addr	TBD	-	-	TBD
R	ExclusiveUse	Boolean	-	0/1	Exclusive to WIR Client
R	WlanProfileSource	Enum	-	-	Origin of the profile
			PROFSRC_DE FAULT	0x0	Default internal
			PROFSRC_HMI	0x1	HMI/User configured
			PROFSRC_CL OUD	0x2	Cloud pushed
			PROFSRC_SP ECIAL	0x3	Special Intent
			PROFSRC_PR OV	0x4	EOL provisioned
R	timestamp	Int64	-	0- 18446744 07370955 1615	Last known time stamp
R	hidden	Boolean	-	0/1	Hidden AP
R	IfaceNametoLink	String	-	Char Value:0- 255 No String length limit	AP SSID
R	Bw_downlink	Int32	-	0- 42949672 95	Application required downlink bandwidth
R	Bw_uplink	Int32	-	0- 42949672 95	Application required uplink bandwidth
Response					
R	appld	String	-	Char Value:0- 255 No String length limit	WIR Client Appld
R	allocId	Int32	-	0- 42949672 95	Allocation ID will be assigned by WIR
R	WirRet	Enum	-	-	Return Command
			WIR_ERROR	0x0	Error/Failure
			WIR_SUCCESS	0x1	Success

**2.9.2.16 MD-REQ-380266/A-NetworkInterfaceRelease**

This API is used internally by WIRClient and WIRServer to request a network interface release. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	allocId	Int32	-	0-4294967295	Allocation ID will be assigned by WIR
Response					
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	allocId	Int32	-	0-4294967295	Allocation ID will be assigned by WIR
R	WirRet	Enum	-	-	Return Command
			WIR_ERROR	0x0	Error/Failure
			WIR_SUCCESS	0x1	Success

2.9.2.17 MD-REQ-380267/A-NetworkInterfaceAllocationStatusInd

This API is used internally by WIRClient and WIRServer to broadcast network interface allocation status.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	N/A
Response					
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	allocId	Int32	-	0-4294967295	Allocation ID will be assigned by WIR
R	NetworkInterfaceAllocationStatus	Enum	-	-	Network Allocation Status
			NET_IFACE_ALLOC_FAILURE	0x0	Error/Failure



			NET_IFACE_AL LOC_SUCCES S	0x1	Success
			NET_IFACE_AL LOC_INQUEUE	0x2	In queue/In progress
			NET_IFACE_AL LOC_ILLEGAL	0x3	Not supported by policy
R	NetworkInterfaceType	Enum	-	-	Interface Type
			IFACE_NONE	0x0	None selected/specified
			IFACE_TCUCE LL	0x1	Cellular interface on TCU
			IFACE_TCUWIF I	0x2	WLAN interface on TCU
			IFACE_SYNCW IFI	0x3	WLAN interface on SYNC
			IFACE_SYNC A PPL	0x4	AppLink interface on SYNC (currently out of scope for CM)
			IFACE_TCUBC ELL	0x5	Cellular interface on TCU-B
			IFACE_TCUBW IFI	0x6	WLAN interface on TCU-B
			IFACE_VDSL A N	0x7	VDS LAN
R	ipAddr	String	-	Char Value:0- 255 No String length limit	IP Address assigned to new network interface

2.9.2.18 MD-REQ-380268/A-NetworkInterfaceDownInd

This API is used internally by WIRClient and WIRServer to broadcast network interface down status.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	N/A
Response					
R	appld	String	-	Char Value:0- 255 No String length limit	WIR Client Appld
R	allocId	Int32	-	0- 42949672 95	Allocation ID will be assigned by WIR
R	NetworkInterfaceAllocat ionStatus	Enum	-	-	Network Allocation Status
			NET_IFACE_AL LOC_FAILURE	0x0	Error/Failure
			NET_IFACE_AL LOC_SUCCES S	0x1	Success



			NET_IFACE_AL LOC_INQUEUE	0x2	In queue/In progress
			NET_IFACE_AL LOC_ILLEGAL	0x3	Not supported by policy
R	NetworkInterfaceType	Enum	-	-	Interface Type
			IFACE_NONE	0x0	None selected/specified
			IFACE_TCUCE LL	0x1	Cellular interface on TCU
			IFACE_TCUWIF I	0x2	WLAN interface on TCU
			IFACE_SYNCW IFI	0x3	WLAN interface on SYNC
			IFACE_SYNCA PPL	0x4	AppLink interface on SYNC (currently out of scope for CM)
			IFACE_TCUBC ELL	0x5	Cellular interface on TCU-B
			IFACE_TCUBW IFI	0x6	WLAN interface on TCU-B
			IFACE_VDSL LAN	0x7	VDS LAN

2.9.2.19 MD-REQ-380269/A-NetworkInterfaceUpInd

This API is used internally by WIRClient and WIRServer to broadcast network interface up status.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	N/A
Response					
R	appld	String	-	Char Value:0- 255 No String length limit	WIR Client Appld
R	allocId	Int32	-	0- 42949672 95	Allocation ID will be assigned by WIR
R	NetworkInterfaceAllocat ionStatus	Enum	-	-	Network Allocation Status
			NET_IFACE_AL LOC_FAILURE	0x0	Error/Failure
			NET_IFACE_AL LOC_SUCCE S	0x1	Success
			NET_IFACE_AL LOC_INQUEUE	0x2	In queue/In progress
			NET_IFACE_AL LOC_ILLEGAL	0x3	Not supported by policy
R	NetworkInterfaceType	Enum	-	-	Interface Type
			IFACE_NONE	0x0	None selected/specified
			IFACE_TCUCE LL	0x1	Cellular interface on TCU
			IFACE_TCUWIF I	0x2	WLAN interface on TCU



			IFACE_SYNCWIFI	0x3	WLAN interface on SYNC
			IFACE_SYNCA PPL	0x4	AppLink interface on SYNC (currently out of scope for CM)
			IFACE_TCUBCELL	0x5	Cellular interface on TCU-B
			IFACE_TCUBWIFI	0x6	WLAN interface on TCU-B
			IFACE_VDSLALAN	0x7	VDS LAN
R	ipAddr	String	-	Char Value:0-255 No String length limit	IP Address assigned to new network interface

2.9.2.20 MD-REQ-380272/A-WebViewDisplay

This API is used internally by WIRClient and WIRServer to request WEB View display. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	ipAddr	String	-	Char Value:0-255 No String length limit	IP address to be used for connection
R	uri	String	-	Char Value:0-255 No String length limit	WEB page URI
Response					
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	WirRet	Enum	-	-	Return Command
			WIR_ERROR	0x0	Error/Failure
			WIR_SUCCESS	0x1	Success

**2.9.2.21 MD-REQ-380274/A-Policy**

This API is used internally by WIRClient and WIRServer to request a network policy for an application. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
Response					
R	CtrlRet	Enum	-	-	
			CTRL_ERROR	0x0	Error/Failure
			CTRL_SUCCESS	0x1	Success
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	Policy	Int32	-	0-4294967295	Network policy, bit flag
			TCU_CELLULAR_APN1	0x001	
			TCU_WIFI	0x002	
			TCU_CELLULAR_APN2	0x004	
			SYNC_WIFI	0x008	
			APPLINK_WIFI	0x010	
			OFF_PEAK	0x020	
			TCU-B_CELLULAR_APN1	0x040	
			TCU-B_WIFI	0x100	
			TCU-B_CELLULAR_APN2	0x200	
			VDS_LAN	0x400	

2.9.2.22 MD-REQ-380275/A-PolicyInd

This API is used internally by WIRClient and WIRServer to broadcast a notification of a network policy change for an application.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	N/A



Response					
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	Policy	Int32	-	0-4294967295	Network policy, bit flag
			TCU_CELLULAR_APN 1	0x001	
			TCU_WIFI	0x002	
			TCU_CELLULAR_APN 2	0x004	
			SYNC_WIFI	0x008	
			APPLINK_WIFI	0x010	
			OFF_PEAK	0x020	
			TCU-B_CELLULAR_APN1	0x040	
			TCU-B_WIFI	0x100	
			TCU-B_CELLULAR_APN2	0x200	
			VDS_LAN	0x400	

2.9.2.23 MD-REQ-380276/A-PolicyTableInd

This API is used internally by WIRClient and WIRServer to broadcast a notification of a network policy table update.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	N/A
Response					
R	Num_entry	Int32		0-4294967295	Number of entries for following two fields
R	appld	String	-	Char Value:0-255 No String length limit	WIR Client Appld
R	Policy	Int32	-	0-4294967295	Network policy, bit flag
			TCU_CELLULAR_APN 1	0x001	
			TCU_WIFI	0x002	
			TCU_CELLULAR_APN 2	0x004	
			SYNC_WIFI	0x008	
			APPLINK_WIFI	0x010	



			OFF_PEAK	0x020	
			TCU- B_CELLUL AR_APN1	0x040	
			TCU- B_WIFI	0x100	
			TCU- B_CELLUL AR_APN2	0x200	
			VDS_LAN	0x400	
R	policyTableVersionMajor	Int32	-	0- 4294967295	Policy Table Version Major
R	policyTableVersionMinor	Int32	-	0- 4294967295	Policy Table Version Minor

2.9.2.24 MD-REQ-380277/A-PolicyTableStatusInd

This API is used internally by WIRClient and WIRServer to broadcast a response to policy table indication.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	N/A
Response					
R	CtrlRet	Enum	-	-	
			CTRL_ERR OR	0x0	Error/Failure
			CTRL_SUC CESS	0x1	Success

2.9.2.25 MD-REQ-380278/A-InterfaceTableInd

This API is used internally by WIRClient and WIRServer to broadcast available network interfaces table update indication.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	N/A
Response					
R	Num_entry	Int32	-	0- 4294967295	Number of entries for following two fields
R	ifaceTable	Int32	-	0- 4294967295	Bit flag of available interfaces
			TCU_CELL ULAR_APN 1	0x001	
			TCU_WIFI	0x002	
			TCU_CELL ULAR_APN 2	0x004	
			SYNC_WIFI	0x008	
			APPLINK_ WIFI	0x010	



			OFF_PEAK	0x020	
			TCU-B_CELLULAR_APN1	0x040	
			TCU-B_WIFI	0x100	
			TCU-B_CELLULAR_APN2	0x200	
			VDS_LAN	0x400	
R	policyTableVersionMajor	Int32	-	0-4294967295	Policy Table Version Major
R	policyTableVersionMinor	Int32	-	0-4294967295	Policy Table Version Minor

2.9.2.26 MD-REQ-380279/A-Enable

This API is used internally by WIRClient and WIRServer to request enabling WLAN. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

2.9.2.27 MD-REQ-380280/A-Disable

This API is used internally by WIRClient and WIRServer to request disabling WLAN. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					



R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

2.9.2.28 MD-REQ-380281/A-ScanApimAPs

This API is used internally by WIRClient and WIRServer to request a scan of available APs. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success
R	ifId	Int32	-	0-4294967295	Interface ID
R	scanId	Int64	-	0-18446744073709551615	ID value used to track scans
R	passive	Boolean	-	0/1	true = passive scan (no pkts tx'd), false = active scan
R	WlanChannel	Enum	-	-	Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	ssid	String	-	Char Value:0-255	list of SSIDs to specifically scan for. Used to scan for hidden networks. "passive" will be ignored as an active scan is required



				No String length limit	
R	Count	Int32	-	0-4294967295	Number of APs found
R	ssid	String	-	Char Value:0-255 No String length limit	SSID – scan result
R	bssid	String	-	Char Value:0-255 No String length limit	MAC address of AP – scan result
R	WlanChannel	Enum	-	-	Channel of AP – scan result
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	Wlan Bandwidth	Enum	-	-	
			WLAN_BW_MIN	0x0	
			WLAN_BW_MHZ20	0x1	20MHz channel
			WLAN_BW_MHZ40	0x2	40MHz channel; must use 11n/11ac. Few devices support on 2.4Ghz
			WLAN_BW_MHZ80	0x3	80MHz channel; must use 11ac
			WLAN_BW_MHZ8080	0x4	80-80MHz, 2 non-contiguous 80MHz; must use 11ac
			WLAN_BW_MHZ160	0x5	160MHz channel; must 11ac
			WLAN_BW_MAX	0x6	
R	rssi	Int32	-	0-4294967295	RSSI of beacon/probe response
R	WlanSecurity	Enum	-	-	Security settings of scan results
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	isWpsSupported	Boolean	-	0/1	does AP support WPS



R	isEssSupported	Boolean	-	0/1	is AP part of an Extended Service Set
R	ipV4Addr	String	-	Char Value:0-255 No String length limit	IPv4 address
R	netmask	String	-	Char Value:0-255 No String length limit	Net mask
R	connected	Boolean	-	0/1	connected to AP
R	prevConnected	Boolean	-	0/1	Previously connected AP
R	isHidden	Boolean	-	0/1	Hidden AP
R	WifiConnectionType	Enum	-	-	
			CONNECTION_TYPE_PIN	0x0	PIN
			CONNECTION_TYPE_WPS_PIN	0x1	WPS (Wi-Fi Protected Setup) PIN
			CONNECTION_TYPE_WPS_PBC	0x2	WPS Push Button Control
			CONNECTION_TYPE_OPEN	0x3	Open WiFi
			CONNECTION_TYPE_HIDDEN_SECURE	0x4	Hidden + security, SSID is not broadcasted
			CONNECTION_TYPE_HIDDEN_OPEN	0x5	Hidden + open, SSID is not broadcasted
R	EWifiConnectionStatus	Enum	-	-	
			WIFI_STATUS_SUCCESS	0	Success
			WIFI_AP_SCAN_SUCCESS	1	Scan Successful
			WIFI_NO_APS_AVAILABLE	2	No Access Points available
			WIFI_CONNECTED	3	Connected to Access Point
			WIFI_NOT_CONNECTED	4	Wifi is not connected to Access Point
			WIFI_OTH_FAIL	5	Authentication Failure
			WIFI_CONNECTING	6	Connecting to Access Point
			WIFI_SECURITY_MISMATCH	7	There is security type mismatch between what requested and what is in actual
			WIFI_AUTHENTICATING	8	In authenticating



			WIFI_ABORTED	9	Connection aborted
			WIFI_WRONG_PASSWORD	10	Failed with wrong password
			WIFI_WRONG_WPS_PIN	11	Failed with wrong WPS PIN
			WIFI_CONNECTION_PROGRESS	12	In connecting
			WIFI_RESTRICTED_NETWORK	13	Connection failure due to trying to connect to a restricted network eg : TCU hotspot
			WIFI_NOT_IMPLEMENTED	101	Not Implemented
			WIFI_STATUS_FAIL	102	Failure
			WIFI_STATUS_FAIL_OTHER	203	Other Failure or Connecting to AP Failed
			WIFI_CANCEL_CONNECTION	104	Cancelled
R	complete	Boolean	-	0/1	TRUE - complete FALSE- more to come

2.9.2.29 MD-REQ-380282/A-CancelScanApimAPs

This API is used internally by WIRClient and WIRServer to request to cancel scanning for available APs. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

2.9.2.30 MD-REQ-380283/A-ConnectToApimAP

This API is used internally by WIRClient and WIRServer to request to connect to an available AP. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description



Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	ssid	String	-	Char Value:0-255 No String length limit	SSID of network
R	bssid	String	-	Char Value:0-255 No String length limit	Optional - BSSID of AP
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	WlanSecurity	Enum	-	-	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanWepSecurity	-	-	-	WEP Settings
		String	Key	Char Value:0-255 No String length limit	
		Int32	defaultKeyIndex	0-4294967295	
R	WlanWpsSecurity	-	-	-	WPS Settings
		WlanWpsType			Type
		String	-	Char Value:0-255 No String length limit	Pin



R	WlanWpaPersonalSecurity	String	-	Char Value:0-255 No String length limit	WPA/WPA2-Personal Settings - password
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None
			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES
			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanWpaEnterprise	TBD	-	-	TBD
R	WlanIpv4AddrType	Enum	-	-	
			WLAN_IPV4_ADDR_MIN	0x0	
			WLAN_IPV4_ADDR_NONE	0x1	No IPv4 Addressing is used
			WLAN_IPV4_ADDR_STATIC	0x2	Static IPv4 Address
			WLAN_IPV4_ADDR_DHCP_CLIENT	0x3	DHCP Client IPv4 Address
			WLAN_IPV4_ADDR_DHCP_SERVER	0x4	DHCP Server IPv4 Address
			WLAN_IPV4_ADDR_MAX	0x5	
R	WlanIpv4Addr	-	-	-	
		String	Ip	Char Value:0-255 No String length limit	IP address of current connection
		String	Netmask	Char Value:0-255 No String length limit	Netmask of current connection
		String	Gateway	Char Value:0-255 No String length limit	default gateway of current connection



		String	dnsPref	Char Value:0-255 No String length limit	Preferred DNS server
		String	dnsAlt	Char Value:0-255 No String length limit	Secondary DNS server
R	WlanIpv6AddrType	Enum	-	-	
			WLAN_IPV6_ADDR_MIN	0x0	
			WLAN_IPV6_ADDR_NONE	0x1	No IPv6 Addressing is used
			WLAN_IPV6_ADDR_STATIC	0x2	Static IPv6 Address
			WLAN_IPV6_ADDR_MAX	0x3	
R	WlanIpv6Addr	TBD	-	-	
R	ExclusiveUse	Boolean	-	0/1	Exclusive to WIR Client
R	WlanProfileSource	Enum	-	-	Origin of the profile
			PROFSRC_DEFAULT	0x0	Default internal
			PROFSRC_HMI	0x1	HMI/User configured
			PROFSRC_CLOUD	0x2	Cloud pushed
			PROFSRC_SPECIAL	0x3	Special Intent
			PROFSRC_PROVIDED	0x4	EOL provisioned
R	timestamp	Int64	-	0-18446744073709551615	Last known time stamp
R	hidden	Boolean	-	0/1	Hidden AP
Response					
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success
R	ssid	String	-	Char Value:0-255 No String length limit	SSID – scan result
R	bssid	String	-	Char Value:0-255 No String length limit	MAC address of AP – scan result
R	WlanChannel	Enum	-	-	Channel of AP – scan result



			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	Wlan Bandwidth	Enum	-	-	
			WLAN_BW_MIN	0x0	
			WLAN_BW_MHZ20	0x1	20MHz channel
			WLAN_BW_MHZ40	0x2	40MHz channel; must use 11n/11ac. Few devices support on 2.4Ghz
			WLAN_BW_MHZ80	0x3	80MHz channel; must use 11ac
			WLAN_BW_MHZ8080	0x4	80-80MHz, 2 non-contiguous 80MHz; must use 11ac
			WLAN_BW_MHZ160	0x5	160MHz channel; must 11ac
			WLAN_BW_MAX	0x6	
R	rssI	Int32	-	0-4294967295	RSSI of beacon/probe response
R	WlanSecurity	Enum	-	-	Security settings of scan results
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	isWpsSupported	Boolean	-	0/1	does AP support WPS
R	isEssSupported	Boolean	-	0/1	is AP part of an Extended Service Set
R	ipV4Addr	String	-	Char Value:0-255 No String length limit	IPv4 address
R	netmask	String	-	Char Value:0-255 No String length limit	Net mask
R	connected	Boolean	-	0/1	connected to AP
R	prevConnected	Boolean	-	0/1	Previously connected AP
R	isHidden	Boolean	-	0/1	Hidden AP



R	WifiConnectionType	Enum	-	-	
			CONNECTION_ TYPE_PIN	0x0	PIN
			CONNECTION_ TYPE_WPS_PI N	0x1	WPS (Wi-Fi Protected Setup) PIN
			CONNECTION_ TYPE_WPS_PB C	0x2	WPS Push Button Control
			CONNECTION_ TYPE_OPEN	0x3	Open WiFi
			CONNECTION_ TYPE_HIDDEN _SECURE	0x4	Hidden + security, SSID is not broadcasted
			CONNECTION_ TYPE_HIDDEN _OPEN	0x5	Hidden + open, SSID is not broadcasted
R	EWifiConnectionStatus	Enum	-	-	
			WIFI_STATUS_ SUCCESS	0	Success
			WIFI_AP_SCAN _SUCCESS	1	Scan Successful
			WIFI_NO_APS_ AVAILABLE	2	No Access Points available
			WIFI_CONNEC TED	3	Connected to Access Point
			WIFI_NOT_CO NNECTED	4	Wifi is not connected to Access Point
			WIFI_OTH_FAI L	5	Authentication Failure
			WIFI_CONNEC TING	6	Connecting to Access Point
			WIFI_SECURIT Y_MISSMATCH	7	There is security type mismatch between what requested and what is in actual
			WIFI_AUTHENT ICATING	8	In authenticating
			WIFI_ABORTE D	9	Connection aborted
			WIFI_WRONG_ PASSWORD	10	Failed with wrong password
			WIFI_WRONG_ WPS_PIN	11	Failed with wrong WPS PIN
			WIFI_CONNEC TION_PROGRE SS	12	In connecting
			WIFI_RESTRIC TED_NETWORK	13	Connection failure due to trying to connect to a restricted network eg : TCU hotspot
			WIFI_NOT_IMP LEMENTED	101	Not Implemented
			WIFI_STATUS_ FAIL	102	Failure
			WIFI_STATUS_ FAIL_OTHER	203	Other Failure or Connecting to AP Failed
			WIFI_CANCEL_ CONNECTION	104	Cancelled

**2.9.2.31 MD-REQ-380284/A-CancelConnectToApimAP**

This API is used internally by WIRClient and WIRServer to request to cancel connecting to an available AP. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

2.9.2.32 MD-REQ-380285/A-DisconnectFromApimAP

This API is used internally by WIRClient and WIRServer to request to disconnect from an available AP. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

2.9.2.33 MD-REQ-380286/A-ForgetApimAP

This API is used internally by WIRClient and WIRServer to request to forget an available AP. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
--------------------	--	--------------------	--	--	--



QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	ssid	String	-		SSID of AP
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

2.9.2.34 MD-REQ-380287/A-GetWpsPin

This API is used internally by WIRClient and WIRServer to request to get a WPS PIN. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success
R	WlanWpsSecurity	Enum	-	-	
			WLAN_WPS_MIN	0x0	
			WLAN_WPS_KEYPAD	0x1	AP supplies the PIN
			WLAN_WPS_PIN	0x2	STA supplies the PIN
			WLAN_WPS_PUSHBUTTON	0x3	Pushbutton



			WLAN_WPS_M AX	0x4	
	pin	String	-	Char Value:0- 255 No String length limit	The Pin when using keypad type

2.9.2.35 MD-REQ-380288/A-GetNetworkDetails

This API is used internally by WIRClient and WIRServer to request to get network details. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0- 255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0- 255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success
R	ssid	String	-	Char Value:0- 255 No String length limit	SSID of network
R	bssid	String	-	Char Value:0- 255 No String length limit	Optional - BSSID of AP
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	WlanSecurity	Enum	-	-	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP



			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanWepSecurity	-	-	-	WEP Settings
		String	Key	Char Value:0-255 No String length limit	
		Int32	defaultKeyIndex	0-4294967295	
R	WlanWpsSecurity	-	-	-	WPS Settings
		WlanWpsType	-		Type
		String	-	Char Value:0-255 No String length limit	Pin
R	WlanWpaPersonalSecurity	String	-	Char Value:0-255 No String length limit	WPA/WPA2-Personal Settings - password
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None
			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES
			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanWpaEnterprise	TBD	-	-	TBD
R	WlanIpv4AddrType	Enum	-	-	
			WLAN_IPV4_ADDR_MIN	0x0	
			WLAN_IPV4_ADDR_NONE	0x1	No IPv4 Addressing is used
			WLAN_IPV4_ADDR_STATIC	0x2	Static IPv4 Address



			WLAN_IPV4_A DDR_DHCP_CL IENT	0x3	DHCP Client IPv4 Address
			WLAN_IPV4_A DDR_DHCP_S ERVER	0x4	DHCP Server IPv4 Address
			WLAN_IPV4_A DDR_MAX	0x5	
R	WlanIpv4Addr	-	-	-	
		String	Ip	Char Value:0- 255 No String length limit	IP address of current connection
		String	Netmask	Char Value:0- 255 No String length limit	Netmask of currenct connection
		String	Gateway	Char Value:0- 255 No String length limit	default gateway of current connection
		String	dnsPref	Char Value:0- 255 No String length limit	Prefered DNS server
		String	dnsAlt	Char Value:0- 255 No String length limit	Secondary DNS server
R	WlanIpv6AddrType	Enum	-	-	
			WLAN_IPV6_A DDR_MIN	0x0	
			WLAN_IPV6_A DDR_NONE	0x1	No IPv6 Addressing is used
			WLAN_IPV6_A DDR_STATIC	0x2	Static IPv6 Address
			WLAN_IPV6_A DDR_MAX	0x3	
R	WlanIpv6Addr	TBD	-	-	
R	ExclusiveUse	Boolea n	-	0/1	Exclusive to WIR Client
R	WlanProfileSource	Enum	-	-	Origin of the profile
			PROFSRC_DE FAULT	0x0	Default internal
			PROFSRC_HMI	0x1	HMI/User configured
			PROFSRC_CL OUD	0x2	Cloud pushed



			PROFSRC_SP ECIAL	0x3	Special Intent
			PROFSRC_PR OV	0x4	EOL provisioned
R	timestamp	Int64	-	0- 18446744 07370955 1615	Last known time stamp
R	hidden	Boolea n	-	0/1	Hidden AP

2.9.2.36 MD-REQ-380289/A-GetEnableStatus

This API is used internally by WIRClient and WIRServer to request to get enable status. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0- 255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0- 255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success
R	status	Boolea n	-	0/1	True - enabled, false-disabled

2.9.2.37 MD-REQ-380290/A-SetNotification

This API is used internally by WIRClient and WIRServer to request to set notification on/off. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0- 255 No String length limit	Requesting app ID
R	on	Boolea n	-	0/1	On or off
Response					



R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success

2.9.2.38 MD-REQ-380291/A-GetNotification

This API is used internally by WIRClient and WIRServer to request to get notification on/off status. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	HmiRet	Enum	-	-	
			HMI_ERROR	0x0	Error/Failure
			HMI_SUCCESS	0x1	Success
R	on	Boolean	-	0/1	On or off

2.9.2.39 MD-REQ-380292/A-WifiAvailableInd

This API is used internally by WIRClient and WIRServer to broadcast a notification to notify that a WiFi AP is available.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	-
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID

**2.9.2.40 MD-REQ-380293/A-WifiUnavailableInd**

This API is used internally by WIRClient and WIRServer to broadcast a notification to notify that a WiFi AP is unavailable.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	-
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID

2.9.2.41 MD-REQ-380294/A-WifiConnectedInd

This API is used internally by WIRClient and WIRServer to broadcast a notification to notify that a WiFi client is connected.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	-
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	ssid	String	-	Char Value:0-255 No String length limit	SSID of network
R	bssid	String	-	Char Value:0-255 No String length limit	Optional - BSSID of AP
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	WlanSecurity	Enum	-	-	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP



			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanWepSecurity	-	-	-	WEP Settings
		String	Key	Char Value:0-255 No String length limit	
		Int32	defaultKeyIndex	0-4294967295	
R	WlanWpsSecurity	-	-	-	WPS Settings
		WlanWpsType	-		Type
		String	-	Char Value:0-255 No String length limit	Pin
R	WlanWpaPersonalSecurity	String	-	Char Value:0-255 No String length limit	WPA/WPA2-Personal Settings - password
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None
			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES
			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanWpaEnterprise	TBD	-		TBD
R	WlanIpv4AddrType	Enum	-		
			WLAN_IPV4_ADDR_MIN	0x0	
			WLAN_IPV4_ADDR_NONE	0x1	No IPv4 Addressing is used
			WLAN_IPV4_ADDR_STATIC	0x2	Static IPv4 Address



			WLAN_IPV4_A DDR_DHCP_CL IENT	0x3	DHCP Client IPv4 Address
			WLAN_IPV4_A DDR_DHCP_S ERVER	0x4	DHCP Server IPv4 Address
			WLAN_IPV4_A DDR_MAX	0x5	
R	WlanIpv4Addr	-	-	-	
		String	Ip	Char Value:0- 255 No String length limit	IP address of current connection
		String	Netmask	Char Value:0- 255 No String length limit	Netmask of currenct connection
		String	Gateway	Char Value:0- 255 No String length limit	default gateway of current connection
		String	dnsPref	Char Value:0- 255 No String length limit	Prefered DNS server
		String	dnsAlt	Char Value:0- 255 No String length limit	Secondary DNS server
R	WlanIpv6AddrType	Enum	-	-	
			WLAN_IPV6_A DDR_MIN	0x0	
			WLAN_IPV6_A DDR_NONE	0x1	No IPv6 Addressing is used
			WLAN_IPV6_A DDR_STATIC	0x2	Static IPv6 Address
			WLAN_IPV6_A DDR_MAX	0x3	
R	WlanIpv6Addr	TBD	-	-	
R	ExclusiveUse	Boolea n	-	0/1	Exclusive to WIR Client
R	WlanProfileSource	Enum	-	-	Origin of the profile
			PROFSRC_DE FAULT	0x0	Default internal
			PROFSRC_HMI	0x1	HMI/User configured
			PROFSRC_CL OUD	0x2	Cloud pushed



			PROFSRC_SPECIAL	0x3	Special Intent
			PROFSRC_PROV	0x4	EOL provisioned
R	timestamp	Int64	-	0-18446744073709551615	Last known time stamp
R	hidden	Boolean	-	0/1	Hidden AP

2.9.2.42 MD-REQ-380295/A-WifiDisconnectedInd

This API is used internally by WIRClient and WIRServer to broadcast a notification to notify that a WiFi client is disconnected.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	-
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID

2.9.2.43 MD-REQ-380296/A-WifiSignalStrengthInd

This API is used internally by WIRClient and WIRServer to broadcast a notification to notify that AP's RSSI was received from a WLAN client.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	-
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	rsssi	Int32	-	0-4294967295	Signal Strength

2.9.2.44 MD-REQ-380297/A-WifiMacAddressInd

This API is used internally by WIRClient and WIRServer to broadcast a notification to notify a client MAC address.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description



Request					
-	-	-	-	-	-
Response					
R	appld	String	-	Char Value:0-255 No String length limit	Requesting app ID
R	macAddr	String	-	Char Value:0-255 No String length limit	MAC Address

2.9.2.45 MD-REQ-380298/A-WlanNQM

This API is used internally by WIRClient and WIRServer to request a WLAN network quality measurement. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	EcuType	Enum	-	-	ECU type
			ECU_UNK	0x0	Error
			ECU_ECG	0x1	ECG
			ECU_TCU	0x2	TCU
			ECU_SYNC	0x3	SYNC
			ECU_TCU_B	0x4	TCU-B
			ECU_VDS	0x5	VDS
			ECU_AVPIIM	0x6	AVPIIM
			ECU_DDSDM	0x7	DDSDM
R	NetworkInterfaceType	Enum	-	-	
			IFACE_NON E	0x0	None selected/specified
			IFACE_TCU CELL	0x1	Cellular interface on TCU
			IFACE_TCU WIFI	0x2	WLAN interface on TCU
			IFACE_SYNC CWIFI	0x3	WLAN interface on SYNC
			IFACE_SYNC CAPPL	0x4	AppLink interface on SYNC (currently out of scope for CM)
			IFACE_TCU BCELL	0x5	Cellular interface on TCU-B
			IFACE_TCU BWIFI	0x6	WLAN interface on TCU-B
			IFACE_VDS LAN	0x7	VDS LAN
			IFACE_ERR	0x8	Error condition
Response					
R	NqmRet	Enum	-	-	
			NQM_ERRO R	0x0	Error/Failure



			NQM_SUCC ESS	0x1	Success
R	EcuType	Enum	-	-	ECU type
			ECU_UNK	0x0	Error
			ECU_ECG	0x1	ECG
			ECU_TCU	0x2	TCU
			ECU_SYNC	0x3	SYNC
			ECU_TCU_B	0x4	TCU-B
			ECU_VDS	0x5	VDS
			ECU_AVPIIM	0x6	AVPIIM
			ECU_DDSDM	0x7	DDSDM
R	NetworkInterfaceType	Enum	-	-	
			IFACE_NON E	0x0	None selected/specified
			IFACE_TCU CELL	0x1	Cellular interface on TCU
			IFACE_TCU WIFI	0x2	WLAN interface on TCU
			IFACE_SYN CWIFI	0x3	WLAN interface on SYNC
			IFACE_SYN CAPPL	0x4	AppLink interface on SYNC (currently out of scope for CM)
			IFACE_TCU BCELL	0x5	Cellular interface on TCU-B
			IFACE_TCU BWIFI	0x6	WLAN interface on TCU-B
			IFACE_VDS LAN	0x7	VDS LAN
			IFACE_ERR	0x8	Error condition
R	WlanStaSmState	Enum	-	-	
			WLAN_STA_ SM_STATE_ OFF	0x0	WLAN off
			WLAN_STA_ SM_STATE_ ENABLING	0x1	Enabling
			WLAN_STA_ SM_STATE_ ON	0x2	On
			WLAN_STA_ SM_STATE_ CONNECTIN G	0x3	Connecting
			WLAN_STA_ SM_STATE_ CONNECTE D	0x4	Connected
			WLAN_STA_ SM_STATE_ DISCONNEC TING	0x5	Disconnecting
			WLAN_STA_ SM_STATE_ DISABLING	0x6	Disabling
			WLAN_STA_ SM_STATE_ ERROR	0x7	Error



			WLAN_STA_ SM_STATE_ AUTHERRO R	0x8	Authentication error
			WLAN_STA_ SM_STATE_ NWNOTFOU ND	0x9	Network not found
R	macAddr	String	-	Char Value:0-255 No String length limit	MAC Address
R	WlanStaConnState	Enum	-	-	
			WLAN_STA_ CON_STATE_ DISCONNE CTED	0x0	Disconnected
			WLAN_STA_ CON_STATE_ WPS_ACTI VE	0x1	WPS pending
			WLAN_STA_ CON_STATE_ ASSOCIATI NG	0x2	Associating
			WLAN_STA_ CON_STATE_ IP_ADDRE SSING	0x3	Getting IP address
			WLAN_STA_ CON_STATE_ CONNECT ED	0x4	Connected
			WLAN_STA_ CON_STATE_ DISCONNE CTING	0x5	Disconnecting
			WLAN_STA_ CON_STATE_ AUTHERR OR	0x6	Authentication error
			WLAN_STA_ CON_STATE_ NWNOTFO UND	0x7	Network not found
R	connSsid	String	-	Char Value:0-255	Connected AP's SSID
R	connBssid	String	-	No String length limit	Connected AP's BSSID
R	conn80211Tech	String	-	Char Value:0-255	Connected AP's 802.11 technology, 802.11a, ...
R	connChannel	Int32	-	0-4294967295	Connected AP's channel
R	bandwidth	Int32	-	0-4294967295	Connected AP's bandwidth
R	dataRate	Int32	-	0-4294967295	Connected AP's data rate (bps)



R	signalStrength	Int32	-	0-4294967295	RSSI (dBm)
R	signalNoise	Int32	-	0-4294967295	Noise level (dBm)
R	signalSNR	Int32	-	0-4294967295	SNR (dB)
R	ipAddr	String	-	Char Value:0-255 No String length limit	IP address
R	ipSubnet	String	-	Char Value:0-255 No String length limit	Subnet mask
R	ipGateway	String	-	Char Value:0-255 No String length limit	GW IP address
R	ipDnsPref	String	-	Char Value:0-255 No String length limit	IP address of primary DNS
R	ipDnsAlt	String	-	Char Value:0-255 No String length limit	IP address of secondary DNS

2.9.2.46 MD-REQ-380304/A-ScanWiFiNetworks

This API is used internally by WIRClient and WIRServer to request a WiFi network scan. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	ifId	Int32	-	0-4294967295	Interface ID
R	passive	Boolean	-	0/1	true = passive scan (no pkts tx'd), false = active scan
R	WlanChannel	Enum	-	-	List of channels of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	ssid	String	-	Char Value:0-255 No String length limit	List of SSIDs to specifically scan for. Used to scan for hidden networks. "passive" will be ignored as an active scan is required
R	scanId	Int64	-	0-18446744	ID value used to track scans



				07370955 1615	
Response					
R	WlanRet	Enum	-	-	
			WLAN_ERROR	0x0	Error/Failure
			WLAN_SUCCESS	0x1	Success
R	ifId	Int32	-	0- 42949672 95	Interface ID
R	scanId	Int64		0- 18446744 07370955 1615	ID value used to track scans
R	WlanChannel	Enum	-	-	List of channels of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	ssid	String	-	Char Value:0- 255 No String length limit	List of SSIDs to specifically scan for. Used to scan for hidden networks. "passive" will be ignored as an active scan is required
R	count	Int32	-	0- 42949672 95	number of APs found, repeats ap field
R	ssid	String	-	Char Value:0- 255 No String length limit	SSID – scan result
R	bssid	String	-	Char Value:0- 255 No String length limit	MAC address of AP – scan result
R	WlanChannel	Enum	-	-	Channel of AP – scan result
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	Wlan Bandwidth	Enum	-	-	
			WLAN_BW_MIN	0x0	
			WLAN_BW_MHZ20	0x1	20MHz channel
			WLAN_BW_MHZ40	0x2	40MHz channel; must use 11n/11ac. Few devices support on 2.4Ghz



			WLAN_BW_MH_Z80	0x3	80MHz channel; must use 11ac
			WLAN_BW_MH_Z8080	0x4	80-80MHz, 2 non-contiguous 80MHz; must use 11ac
			WLAN_BW_MH_Z160	0x5	160MHz channel; must use 11ac
			WLAN_BW_MAX	0x6	
R	rssi	Int32	-	0-4294967295	RSSI of beacon/probe response
R	WlanSecurity	Enum	-	-	Security settings of scan results
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	isWpsSupported	Boolean	-	0/1	does AP support WPS
R	isEssSupported	Boolean	-	0/1	is AP part of an Extended Service Set
R	scanResultsComplete	Boolean	-	0/1	All the scan results have been reported or not

2.9.2.47 MD-REQ-380305/A-ConnectWiFiAP

This API is used internally by WIRClient and WIRServer to request to connect to an AP. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	ifId	Int32	-	0-4294967295	Interface ID
R	ssid	String	-	Char Value:0-255 No String length limit	SSID of network
R	bssid	String	-	Char Value:0-255 No String length limit	Optional - BSSID of AP
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	



			
R	WlanSecurity	Enum	WLAN_CH_165	0xA5	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanWepSecurity	-	-	-	WEP Settings
		String	Key	Char Value:0-255 No String length limit	
		Int32	defaultKeyIndex	0-4294967295	
R	WlanWpsSecurity	-	-	-	WPS Settings
		WlanWpsType	-		Type
		String	-	Char Value:0-255 No String length limit	Pin
R	WlanWpaPersonalSecurity	String	-	Char Value:0-255 No String length limit	WPA/WPA2-Personal Settings - password
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None
			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES
			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanWpaEnterprise	TBD	-		TBD
R	WlanIpv4AddrType	Enum	-		
			WLAN_IPV4_ADDR_MIN	0x0	



			WLAN_IPV4_A DDR_NONE	0x1	No IPv4 Addressing is used
			WLAN_IPV4_A DDR_STATIC	0x2	Static IPv4 Address
			WLAN_IPV4_A DDR_DHCP_CL IENT	0x3	DHCP Client IPv4 Address
			WLAN_IPV4_A DDR_DHCP_S ERVER	0x4	DHCP Server IPv4 Address
			WLAN_IPV4_A DDR_MAX	0x5	
R	WlanIpv4Addr	-	-	-	
		String	Ip	Char Value:0- 255 No String length limit	IP address of current connection
		String	Netmask	Char Value:0- 255 No String length limit	Netmask of currenct connection
		String	Gateway	Char Value:0- 255 No String length limit	default gateway of current connection
		String	dnsPref	Char Value:0- 255 No String length limit	Prefered DNS server
		String	dnsAlt	Char Value:0- 255 No String length limit	Secondary DNS server
R	WlanIpv6AddrType	Enum	-	-	
			WLAN_IPV6_A DDR_MIN	0x0	
			WLAN_IPV6_A DDR_NONE	0x1	No IPv6 Addressing is used
			WLAN_IPV6_A DDR_STATIC	0x2	Static IPv6 Address
			WLAN_IPV6_A DDR_MAX	0x3	
R	WlanIpv6Addr	TBD	-	-	
R	ExclusiveUse	Boolea n	-	0/1	Exclusive to WIR Client
R	WlanProfileSource	Enum	-	-	Origin of the profile
			PROFSRC_DE FAULT	0x0	Default internal



			PROFSRC_HMI	0x1	HMI/User configured
			PROFSRC_CLOUD	0x2	Cloud pushed
			PROFSRC_SPECIAL	0x3	Special Intent
			PROFSRC_PROV	0x4	EOL provisioned
R	timestamp	Int64	-	0-18446744073709551615	Last known time stamp
R	hidden	Boolean	-	0/1	Hidden AP
Response					
R	WlanRet	Enum	-	-	
			WLAN_ERROR	0x0	Error/Failure
			WLAN_SUCCESS	0x1	Success
R	ifId	Int32	-	0-4294967295	Interface ID
R	ifaceName	String	-	Char Value:0-255 No String length limit	Interface Name

2.9.2.48 MD-REQ-380306/A-DisconnectWiFiAP

This API is used internally by WIRClient and WIRServer to request to disconnect from an AP. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	ifId	Int32	-	0-4294967295	Interface ID
Response					
R	WlanRet	Enum	-	-	
			WLAN_ERROR	0x0	Error/Failure
			WLAN_SUCCESS	0x1	Success
R	ifId	Int32	-	0-4294967295	Interface ID

2.9.2.49 MD-REQ-380307/A-ProfileUpdate

This API is used internally by WIRClient and WIRServer to request to update a profile. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			



R/O	Name	Type	Literals	Value	Description
Request					
R	txId	String	-	Char Value:0-255 No String length limit	Transaction ID
Response					
R	WlanRet	Enum	-	-	
			WLAN_ERROR	0x0	Error/Failure
			WLAN_SUCCESS	0x1	Success
R	txId	String	-	Char Value:0-255 No String length limit	Transaction ID
R	updateComplete	Boolean	-	0/1	Is it last segment?
R	profile_cnt	String	-	Char Value:0-255 No String length limit	Number of entries of following 2 fields
R	ssid	String	-	Char Value:0-255 No String length limit	SSID
R	bssid	String	-	Char Value:0-255 No String length limit	Optional - BSSID of AP
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	WlanSecurity	Enum	-	-	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanWepSecurity	-	-	-	WEP Settings



		String	Key	Char Value:0-255 No String length limit	
		Int32	defaultKeyIndex	0-4294967295	
R	WlanWpsSecurity	-	-	-	WPS Settings
		WlanWpsType	-	-	Type
		String	-	Char Value:0-255 No String length limit	Pin
R	WlanWpaPersonalSecurity	String	-	Char Value:0-255 No String length limit	WPA/WPA2-Personal Settings - password
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None
			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES
			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanWpaEnterprise	TBD	-		TBD
R	WlanIpv4AddrType	Enum	-		
			WLAN_IPV4_ADDR_MIN	0x0	
			WLAN_IPV4_ADDR_NONE	0x1	No IPv4 Addressing is used
			WLAN_IPV4_ADDR_STATIC	0x2	Static IPv4 Address
			WLAN_IPV4_ADDR_DHCP_CLIENT	0x3	DHCP Client IPv4 Address
			WLAN_IPV4_ADDR_DHCP_SERVER	0x4	DHCP Server IPv4 Address
			WLAN_IPV4_ADDR_MAX	0x5	
R	WlanIpv4Addr	-	-	-	



		String	Ip	Char Value:0-255 No String length limit	IP address of current connection
		String	Netmask	Char Value:0-255 No String length limit	Netmask of current connection
		String	Gateway	Char Value:0-255 No String length limit	default gateway of current connection
		String	dnsPref	Char Value:0-255 No String length limit	Preferred DNS server
		String	dnsAlt	Char Value:0-255 No String length limit	Secondary DNS server
R	WlanIpv6AddrType	Enum	-	-	
			WLAN_IPV6_ADDR_MIN	0x0	
			WLAN_IPV6_ADDR_NONE	0x1	No IPv6 Addressing is used
			WLAN_IPV6_ADDR_STATIC	0x2	Static IPv6 Address
			WLAN_IPV6_ADDR_MAX	0x3	
R	WlanIpv6Addr	TBD	-	-	
R	ExclusiveUse	Boolean	-	0/1	Exclusive to WIR Client
R	WlanProfileSource	Enum	-	-	Origin of the profile
			PROFSRC_DEFAULT	0x0	Default internal
			PROFSRC_HMI	0x1	HMI/User configured
			PROFSRC_CLOUD	0x2	Cloud pushed
			PROFSRC_SPECIAL	0x3	Special Intent
			PROFSRC_PROVIDED	0x4	EOL provisioned
R	timestamp	Int64	-	0-18446744073709551615	Last known time stamp
R	hidden	Boolean	-	0/1	Hidden AP

**2.9.2.50 MD-REQ-380308/A-ConnectionStatus**

This API is used internally by WIRClient and WIRServer to request connection status. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	ssid	String	-	Char Value:0-255 No String length limit	SSID
R	bssid	String	-	Char Value:0-255 No String length limit	BSSID, MAC address
R	ifId	Int32	-	0-4294967295	Interface ID
Response					
R	WlanRet	Enum	-	-	
			WLAN_ERROR	0x0	Error/Failure
			WLAN_SUCCESS	0x1	Success
R	ssid	String	-	Char Value:0-255 No String length limit	SSID
R	bssid	String	-	Char Value:0-255 No String length limit	BSSID, MAC address
R	WlanStaConnState	Enum	-	-	
			WLAN_STA_CONNECTION_DISCONNECTED	0x0	Disconnected
			WLAN_STA_CONNECTION_WPS_ACTIVE	0x1	WPS pending
			WLAN_STA_CONNECTION_ASSOCIATING	0x2	Associating
			WLAN_STA_CONNECTION_IP_ADDRESSING	0x3	Getting IP address
			WLAN_STA_CONNECTION_CONNECTED	0x4	Connected



			WLAN_STA_C ON_STATE_DI SCONNECTING	0x5	Disconnecting
			WLAN_STA_C ON_STATE_AU TERROR	0x6	Authentication error
			WLAN_STA_C ON_STATE_N WNOTFOUND	0x7	Network not found
R	WlanStaSmState	Enum	-	-	
			WLAN_STA_S M_STATE_OFF	0x0	WLAN off
			WLAN_STA_S M_STATE_ENA BLING	0x1	Enabling
			WLAN_STA_S M_STATE_ON	0x2	On
			WLAN_STA_S M_STATE_CON NECTING	0x3	Connecting
			WLAN_STA_S M_STATE_CON NECTED	0x4	Connected
			WLAN_STA_S M_STATE_DIS CONNECTING	0x5	Disconnecting
			WLAN_STA_S M_STATE_DIS ABLING	0x6	Disabling
			WLAN_STA_S M_STATE_ERR OR	0x7	Error
			WLAN_STA_S M_STATE_AUT HERROR	0x8	Authentication error
			WLAN_STA_S M_STATE_NW NOTFOUND	0x9	Network not found
R	ifId	Int32	-	0- 42949672 95	Interface ID

2.9.2.51 MD-REQ-380309/A-StateUpdateInd

This API is used internally by WIRClient and WIRServer to broadcast an update of ECU's station mode connection state.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	-
Response					
R	WlanEcu	Enum	-	-	
			TCU_WLAN	0	TCU WLAN
			SYNC_WLAN	1	SYNC WLAN
			TCU_B_WLAN	2	TCU-B WLAN
R	WlanStaConnState	Enum	-	-	



			WLAN_STA_C ON_STATE_DI SCONNECTED	0x0	Disconnected
			WLAN_STA_C ON_STATE_W PS_ACTIVE	0x1	WPS pending
			WLAN_STA_C ON_STATE_AS SOCIATING	0x2	Associating
			WLAN_STA_C ON_STATE_IP_ ADDRESSING	0x3	Getting IP address
			WLAN_STA_C ON_STATE_CO NNECTED	0x4	Connected
			WLAN_STA_C ON_STATE_DI SCONNECTING	0x5	Disconnecting
			WLAN_STA_C ON_STATE_AU THERROR	0x6	Authentication error
			WLAN_STA_C ON_STATE_N WNOTFOUND	0x7	Network not found
R	WlanStaSmState	Enum	-	-	
			WLAN_STA_S M_STATE_OFF	0x0	WLAN off
			WLAN_STA_S M_STATE_ENA BLING	0x1	Enabling
			WLAN_STA_S M_STATE_ON	0x2	On
			WLAN_STA_S M_STATE_CON NECTING	0x3	Connecting
			WLAN_STA_S M_STATE_CON NECTED	0x4	Connected
			WLAN_STA_S M_STATE_DIS CONNECTING	0x5	Disconnecting
			WLAN_STA_S M_STATE_DIS ABLING	0x6	Disabling
			WLAN_STA_S M_STATE_ERR OR	0x7	Error
			WLAN_STA_S M_STATE_AUT HERROR	0x8	Authentication error
			WLAN_STA_S M_STATE_NW NOTFOUND	0x9	Network not found
R	macAddr	String	-	Char Value:0- 255	MAC Address



				No String length limit	
R	ssid	String	-	Char Value:0-255 No String length limit	SSID
R	bssid	String	-	Char Value:0-255 No String length limit	BSSID
R	WlanSecurity	Enum	-	-	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None
			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES
			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	bandwidth	Int32	-	0-4294967295	
R	WlanTech	Enum	-	-	
			Unk	0x0	Unknown
			B	0x1	11b; 2.4Ghz, Max data rate 11Mbps
			A	0x2	11a, 5.0Ghz, Max data rate 54Mbps



			G	0x3	11g, 2.4Ghz, Max data rate 54Mbps
			N	0x4	11n, 2.4/5.0Ghz, Max data rate - 1ss (20/40): 72.2/150Mbps
			AC	0x5	11ac, 2.4/5.0Ghz, Max data rate - 1ss (20/40/80/160) : 78/200/433/866Mbps
R	dataRate	Int32	-	0-4294967295	bps
R	signalStrength	Int32	-	0-4294967295	Signal strength, dBm
R	signalNoise	Int32	-	0-4294967295	Noise level, dBm
R	signalSNR	Int32	-	0-4294967295	SNR, dB
R	ipAddr	String	-	Char Value:0-255 No String length limit	IP address
R	ipSubnet	String	-	Char Value:0-255 No String length limit	Subnet mask
R	ipGateway	String	-	Char Value:0-255 No String length limit	GW IP address
R	ipDnsPref	String	-	Char Value:0-255 No String length limit	DNS primary
R	ipDnsAlt	String	-	Char Value:0-255 No String length limit	DNS secondary
R	ifaceName	String	-	Char Value:0-255 No String length limit	Interface name
R	ifId	Int32	-	0-4294967295	Interface ID

**2.9.2.52 MD-REQ-380310/A-StationModeStats**

This API is used internally by WIRClient and WIRServer to request WLAN station mode statistics. WIR also uses this API for its response.

Method Type		One-Shot (A-Synch)			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
R	ifId	Int32	-	0-4294967295	Interface ID
R	ifaceName	String	-	Char Value:0-255 No String length limit	Interface Name
R	StatsCmdType	Enum	-	-	
			DISABLE	0x0	Disable
			ENABLE	0x1	Enable
			QUERY	0x2	Query current state
R	reportingWindow	Int32	-	0-4294967295	Seconds
Response					
R	ifId	Int32	-	0-4294967295	Interface ID
R	ifaceName	String	-	Char Value:0-255 No String length limit	Interface Name
R	StatsCmdType	Enum	-	-	
			DISABLE	0x0	Disable
			ENABLE	0x1	Enable
			QUERY	0x2	Query current state
R	WlanRet	Enum	-	-	
			WLAN_ERROR	0x0	Error/Failure
			WLAN_SUCCESS	0x1	Success
R	pktsTxV4	Int64	-	0-18446744073709551615	Number of packets transmitted
R	pktsRxV4	Int64	-	0-18446744073709551615	Number of packets received
R	bytesTxV4	Int64	-	0-18446744073709551615	Number of bytes transmitted
R	bytesRxV4	Int64	-	0-18446744073709551615	Number of bytes received



				07370955 1615	
R	PktsDroppedTx	Int64	-	0- 18446744 07370955 1615	Number of dropped packets transmitted
R	PktsDroppedRx	Int64	-	0- 18446744 07370955 1615	Number of dropped packets received

2.9.2.53 MD-REQ-380312/A-ProfileUpdateInd

This API is used internally by WIRClient and WIRServer to broadcast a profile update indication.

Method Type		OnChange			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request					
-	-	-	-	-	-
Response					
R	WlanProfileAction	Enum	-	-	
			ACT_UPDATE	0x0	Add/update
			ACT_REMOVE	0x1	Remove
R	ssid	String	-	Char Value:0- 255 No String length limit	SSID of network
R	bssid	String	-	Char Value:0- 255 No String length limit	Optional - BSSID of AP
R	WlanChannel	Enum	-	-	Optional - Channel of AP
			Reserved	0x00	
			WLAN_CH_1	0x01	
			WLAN_CH_2	0x02	
			WLAN_CH_3	0x03	
			
			WLAN_CH_165	0xA5	
R	WlanSecurity	Enum	-	-	Security settings to use
			Reserved	0x00	
			Open	0x01	Open or no security
			WEP	0x02	WEP
			WPS	0x03	WPS (WiFi Protected Setup)
			WPA_Personal	0x04	WPA/WPA2 Personal (passkey)
			WPA_Enterprise	0x05	WPA/WPA2 Enterprise (EAP- PEAP/EAP-TLS/etc) (not supported)
			MAX	0x06	
R	WlanWepSecurity	-	-	-	WEP Settings
		String	Key	Char Value:0- 255	



				No String length limit	
		Int32	defaultKeyIndex	0-4294967295	
R	WlanWpsSecurity	-	-	-	WPS Settings
		WlanWpsType	-		Type
		String	-	Char Value:0-255 No String length limit	Pin
R	WlanWpaPersonalSecurity	String	-	Char Value:0-255 No String length limit	WPA/WPA2-Personal Settings - password
R	WlanPairwiseCipher	Enum	-	-	Pair cipher
			PAIR_MIN	0x0	
			PAIR_NONE	0x1	None
			PAIR_TKIP	0x2	TKIP
			PAIR_CCMP	0x3	CCMP/AES
			PAIR_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			PAIR_MAX	0x5	
R	WlanGroupCipher	Enum	-	-	Group cipher
			GROUP_MIN	0x0	
			GROUP_NONE	0x1	None
			GROUP_TKIP	0x2	TKIP
			GROUP_CCMP	0x3	CCMP/AES
			GROUP_TKIPCCMP	0x4	Mixmode – tkip or ccmp
			GROUP_MAX	0x5	
R	WlanWpaEnterprise	TBD	-		TBD
R	WlanIpv4AddrType	Enum	-		
			WLAN_IPV4_ADDR_MIN	0x0	
			WLAN_IPV4_ADDR_NONE	0x1	No IPv4 Addressing is used
			WLAN_IPV4_ADDR_STATIC	0x2	Static IPv4 Address
			WLAN_IPV4_ADDR_DHCP_CLIENT	0x3	DHCP Client IPv4 Address
			WLAN_IPV4_ADDR_DHCP_SERVER	0x4	DHCP Server IPv4 Address
			WLAN_IPV4_ADDR_MAX	0x5	
R	WlanIpv4Addr	-	-	-	
		String	Ip	Char Value:0-255	IP address of current connection



				No String length limit	
		String	Netmask	Char Value:0-255 No String length limit	Netmask of current connection
		String	Gateway	Char Value:0-255 No String length limit	default gateway of current connection
		String	dnsPref	Char Value:0-255 No String length limit	Preferred DNS server
		String	dnsAlt	Char Value:0-255 No String length limit	Secondary DNS server
R	WlanIpv6AddrType	Enum	-	-	
			WLAN_IPV6_ADDR_MIN	0x0	
			WLAN_IPV6_ADDR_NONE	0x1	No IPv6 Addressing is used
			WLAN_IPV6_ADDR_STATIC	0x2	Static IPv6 Address
			WLAN_IPV6_ADDR_MAX	0x3	
R	WlanIpv6Addr	TBD	-	-	
R	ExclusiveUse	Boolean	-	0/1	Exclusive to WIR Client
R	WlanProfileSource	Enum	-	-	Origin of the profile
			PROFSRC_DEFAULT	0x0	Default internal
			PROFSRC_HMI	0x1	HMI/User configured
			PROFSRC_CLOUD	0x2	Cloud pushed
			PROFSRC_SPECIAL	0x3	Special Intent
			PROFSRC_PROVIDED	0x4	EOL provisioned
R	timestamp	Int64	-	0-18446744073709551615	Last known time stamp
R	hidden	Boolean	-	0/1	Hidden AP



3 Functional Definition

3.1 WIR-FUN-REQ-378578/A-Wifi Handler

3.1.1 Requirements

3.1.1.1 WIR-REQ-378579/A-WI-FI ON/OFF settings

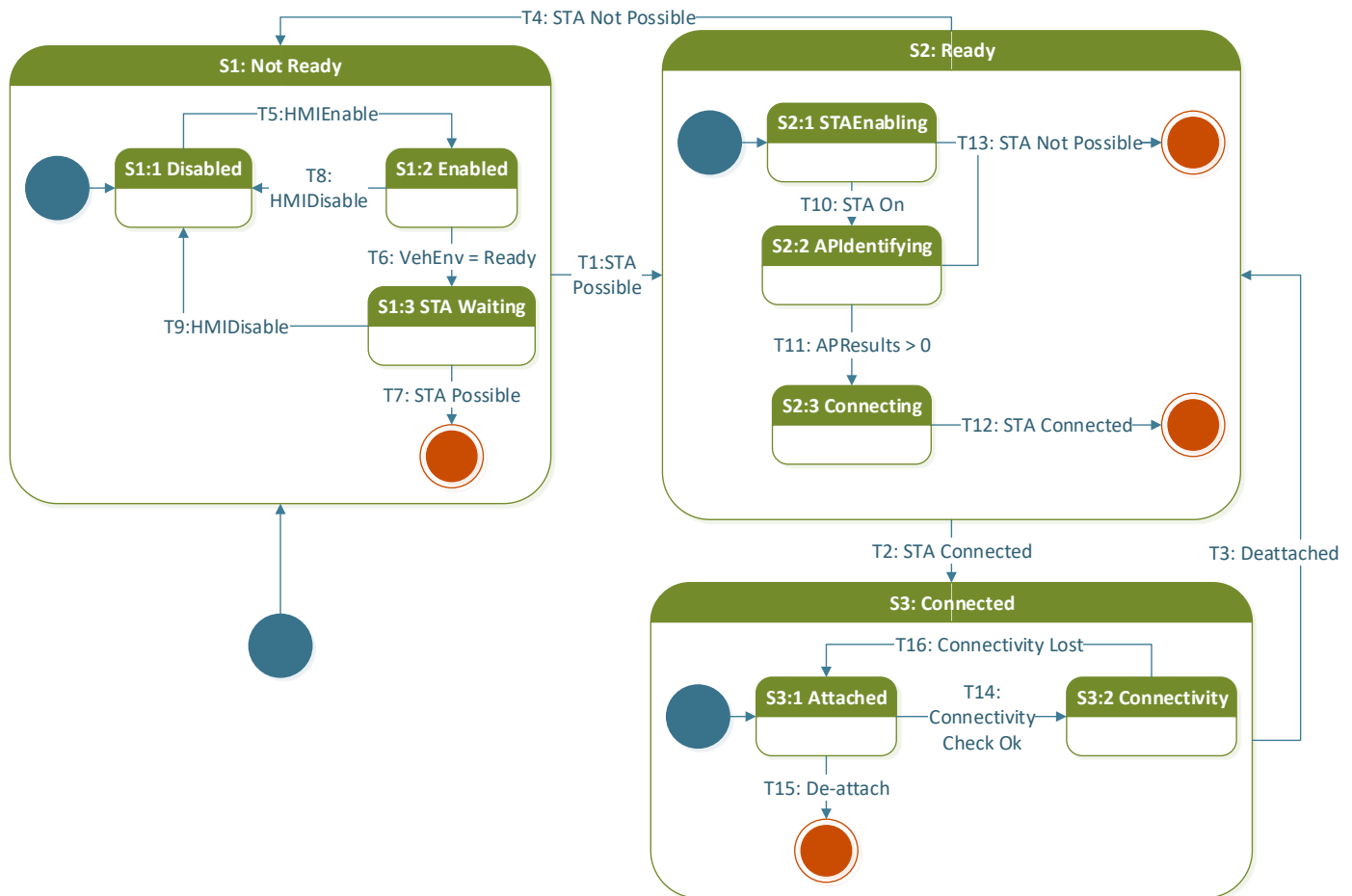
WIRClient1 shall not request WLAN services for any WIFI operation if WI-FI is OFF.

3.1.1.2 WIR-REQ-378580/A-WI-FI Station mode ON/OFF settings

WIRClient1 shall not request WLAN services for any WIFI operation if WI-FI Station mode is OFF.

3.1.1.3 WIR-REQ-378581/A-State machine

SC: WIR STA Connections



3.1.1.4 WIR-REQ-378582/A-Station Mode Enabling

The WIRClient1, upon request from WIRServer or on its own during stand-alone mode, will request STA mode functionality and WLAN services is expected to resolve if station mode is possible in a standalone or shared use mode for a particular WLAN chipset.



3.1.1.5 WIR-REQ-378583/A-Station Mode Enabling Criteria

The WIRClient1 will request STA mode functionality if any or all of the following criteria is met

1. Driver restriction mode is off (vehicle speed is less than driver restriction threshold) and ignition is ON

3.1.1.6 WIR-REQ-378584/A-Station Mode Disabling Criteria

The station mode functionality, when enabled, shall be disabled by WLAN services in the following ways:

1. If user manually disconnects via HMI then WIR shall request WLAN services to disconnect
2. For ECU Low power mode and out of range WLAN services shall handle disconnect process and no involvement of WIR module. WIR shall get notification from WLAN that it successfully disconnected

3.1.1.7 WIR-REQ-378585/A-WIFI Availability after WIR request

If the WIFI becomes available WLAN Services shall notify WIR module regarding WIFI availability and WIR shall determine if request for station mode is needed and preconditions are met and if met WIR shall request WLAN services for WIFI Station mode.

3.1.1.8 WIR-REQ-378586/A-Start Scan

The WIRClient1, upon request from WIRServer or on its own in standalone mode, shall request the WLAN services to perform an active Wi-Fi scan and return the results.

The WLAN services shall start scanning immediately.

3.1.1.9 WIR-REQ-378587/A-Scan Results

Once scanning is completed, WIR shall receive the results from WLAN services.

3.1.1.10 WIR-REQ-378588/A-Periodic Scan

When WIFI chipset is enabled and preconditions as stated in requirement WIR-REQ-276054 as satisfied and WIFI Chipset is not connected to an access point then WIRClient1 shall request WLAN services to scan periodically. The time period between scans is defined in the configurable parameter WIR_WIFI_PERIODIC_SCAN_INTERVAL.

3.1.1.11 WIR-REQ-378589/A-Wifi Access Point Information

WIRClient1 shall be aware of the Wi-Fi access points known by the vehicle, as entered by the driver or supplied by Ford (Ford partnered SSID / Password) in case Ford chooses a WIFI provider.

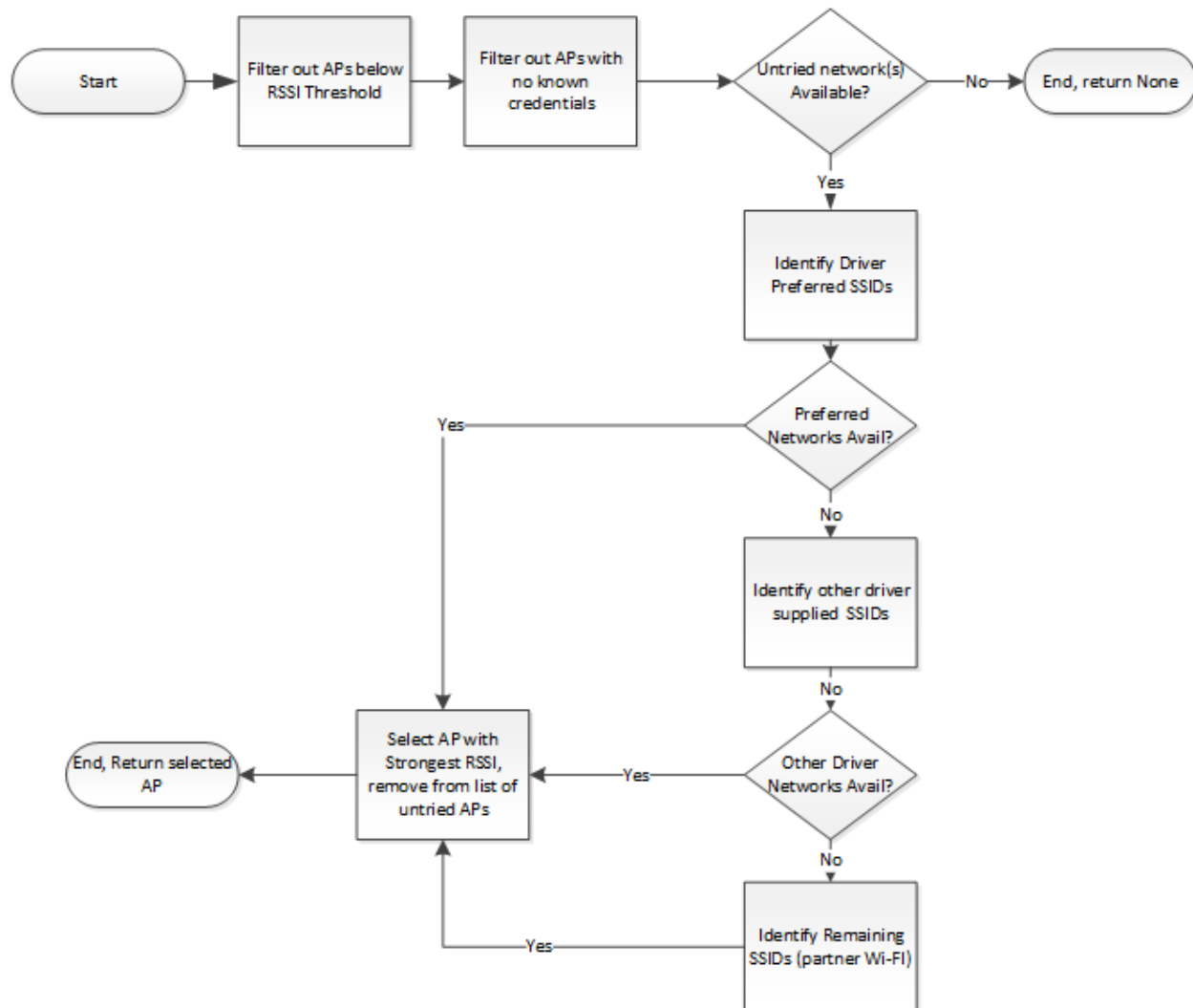
3.1.1.12 WIR-REQ-378590/A-WIFI Access Point Information Storage

WIRClient1 shall be aware of the Wi-Fi access points known by the vehicle, as entered by the driver or supplied by Ford in case Ford chooses a WIFI provider and shall store the same in non-volatile memory for quick retrieval.

WIRClient1 shall be able to store up to 100 WIFI access points and passwords using FIFO order.

3.1.1.13 WIR-REQ-378591/A-Selecting An Access Point

The WIRServer and WIRClient1 / WIRClient2 / WIRClient5, in case of WIRServer not active, follows the below UML Activity diagram when selecting an AP to connect to, from the list provided by the WLAN services scan.

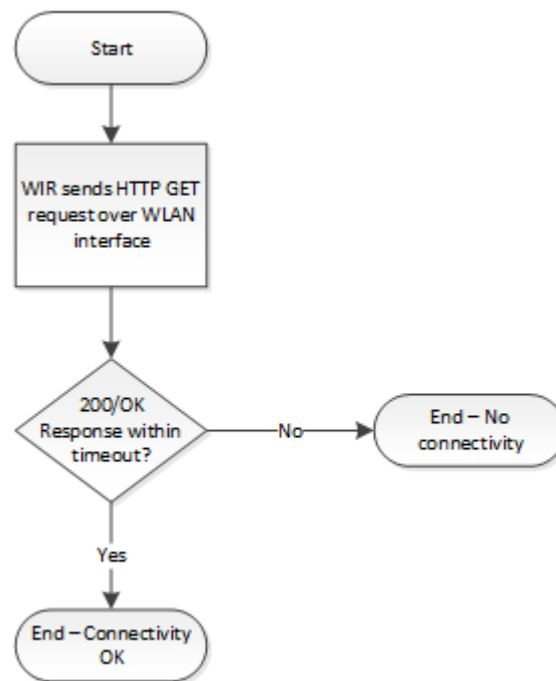


3.1.1.14 WIR-REQ-378592/A-Performing A Connectivity Check

Once a network is selected, WIRClient1 shall perform a connectivity check to determine if the network is good enough to connect

3.1.1.15 WIR-REQ-378593/A-Connectivity Check Procedure

If WLAN services is able to connect, negotiate an IP, and set up an interface for an access point, WIRClient1 should run an HTTP connectivity check over this interface. The endpoint of this HTTP check is TBD. If a 200/OK response can be received within a timeout, the WIRClient1 will consider connectivity over this interface possible, and update the NQM to list the interface as active. The below UML activity diagram describes this connectivity check



3.1.1.16 WIR-REQ-378594/A-Connectivity Check Endpoint

Connectivity check endpoint URL is TBD at this point. If no Connectivity check endpoint URL is available then WIRClient1 shall connect to access point and upon successful connection shall provide interfaces to requesting application without performing connectivity check. The availability of connectivity check endpoint URL is determined by CONNECTIVITY_CHECK_URL_AVAILABILITY.

3.1.1.17 WIR-REQ-378595/A-Wifi Latency Check

Once connected, WIRClient1 must ensure the connected interface continues to be good. The WIRClient1 shall achieve this by performing a latency check. WIR latency check procedure is TBD at this point. This shall be performed only if WIR_LATENCY_CHECK_DID is enabled.

3.1.1.18 WIR-REQ-378596/A-Wifi Low Cost Partner

Ford may identify certain Wi-Fi providers as a low cost Wi-Fi provider. The connection manager should be able to identify these partners' Wi-Fi access point via SSID or Hotspot 2.0 protocol.

3.1.1.19 WIR-REQ-378597/A-Wi-Fi Low Cost Partner Availability

The WIR module shall be able to have a DID that specifies availability of WIFI low cost partner availability.

3.1.1.20 WIR-REQ-378598/A-Wifi Low Cost Partner Credentials

The credentials and relative metered cost of these partners should be stored by the WIRClient1, and re-configurable via the SDN and End of Line.

WIRClient1 shall convey this information to WIRClient2, WIRClient5, and WIRServer.

3.1.1.21 WIR-REQ-378599/A-Wifi SSID / Password From Vehicle User

Customer connected Wi-Fi access points are captured through infotainment interfaces, such as the WIRClient1 vehicle Wi-Fi sub menu. The WIRClient1 securely stores all needed credentials for a entered/known Wi-Fi access point and should be available for use on any STA mode Wi-Fi interface available to the WIR

3.1.1.22 WIR-REQ-378600/A-WIFI Access Point Information removal

WIRClient1 shall also remove access points and passwords from nonvolatile memory when deleted by user and shall NOT use user deleted passwords and SSID for WIFI connection.



3.1.1.23 WIR-REQ-378601/A-WIFI Access Point disconnect

WIRClient2 shall request WLAN services to disconnect from access points when user disconnects the same from HMI. If the user does not select Forget, the network access point's credentials shall still be stored in Non-volatile memory for future automatic reconnects.

If the user selects Forget, the network access point's credentials shall be removed from Non-volatile memory and shall not be used for future automatic reconnects unless user manually connects by reentering the credentials.

3.1.1.24 WIR-REQ-378602/A-WIR HMI

WIFI HMI is not maintained by WIRClient2 and is out of scope for WIR. WIRClient2 shall however receive WIFI Credentials (SSID and password), WIFI disconnect, credentials delete information from WIFI HMI and shall act accordingly as stated in requirements in this section.

3.1.1.25 WIR-REQ-378603/A-WIR Request To Connect To Access Point

The WIRClient1 can request the WLAN services to connect to a given Access point by specifying an SSID or MAC ID and providing access credentials to the WLAN services.

3.1.1.26 WIR-REQ-378604/A-WLAN Services To Support Connection

The WLAN services will connect and establish a local interface, and be responsible for obtaining an IP address from the access point. The WLAN services will resolve an IP disputes, and report the success or failure of the connection to the WIRClient1.

3.1.1.27 WIR-REQ-378605/A-WIRClient1 Publishes WI-FI Status

The WIRClient1 shall be capable of publishing WI-FI status like:

Scan results

Connection status

Signal strength of Connected access point

This information published by WIRClient1 shall be utilized by applications (example OTA) and WIRServer.

3.1.1.28 WIR-REQ-378606/A-WLAN Services To Support Disconnection

The WLAN services shall maintain a strategy to disconnect from an access points due to signal or connection issues with the access point. The success or failure of a disconnect shall be reported to the WIRClient1.

3.1.1.29 WIR-REQ-378607/A-WIRServer Requests WIRClient1 WIFI Disconnect

The WIRClient1 shall disconnect from access point when WIR server requests WIRClient1. The WIRServer shall request WIRClient1 to disconnect in events where WIRServer wants to use WIRClient2 WI-FI or WIRClient5 WI-FI and no application is requesting WIRClient1 to connect.

3.1.2 Use Cases

3.1.3 White Box View



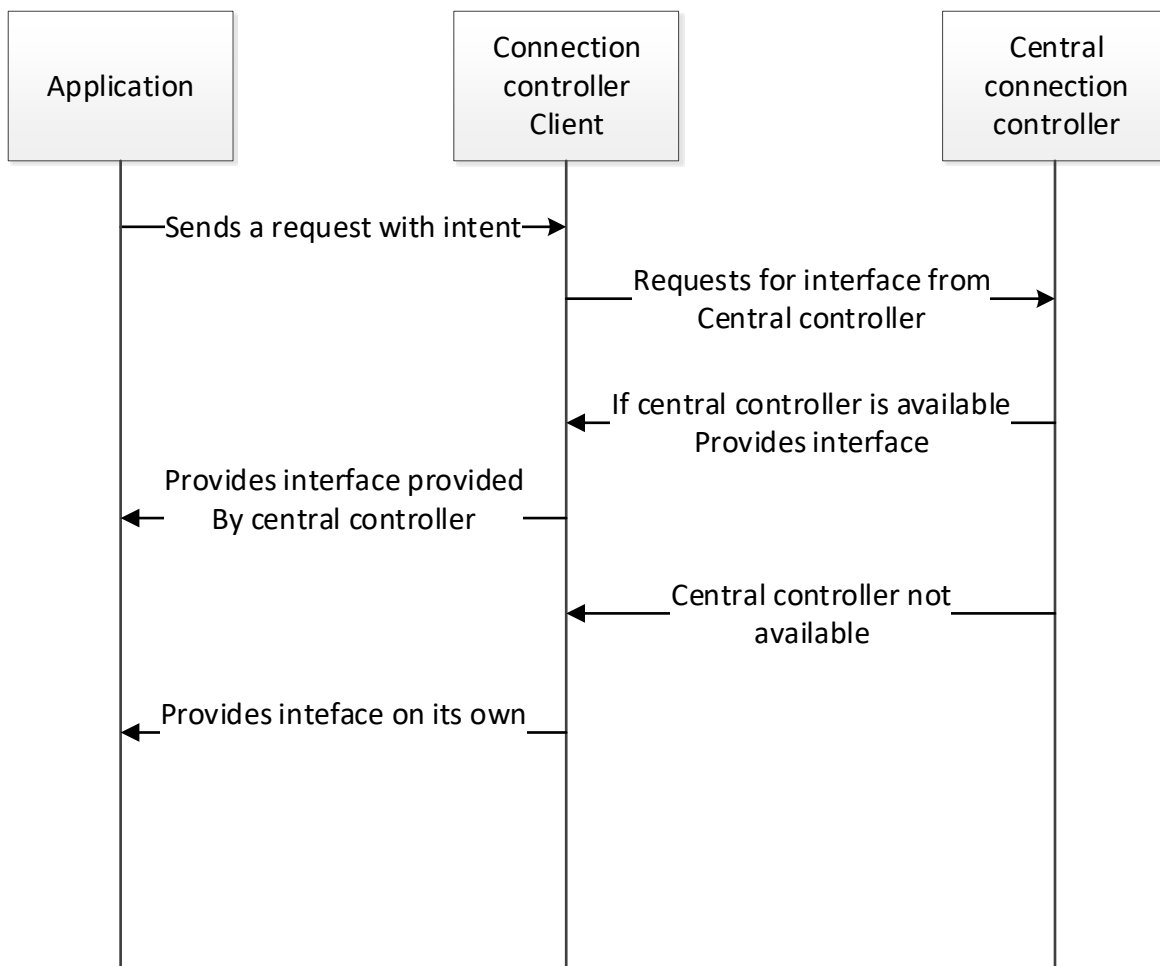
3.2 WIR-FUN-REQ-378637/A-Local Controller Client

3.2.1 Requirements

3.2.1.1 WIR-REQ-378638/A-Local Controller Purpose

The Local controller resides in WIRClient1, WIRClient2, WIRClient3, WIRClient4, WIRClient5, WIRClient6 and WIRServer and can be extended seamlessly to other potential Ethernet interfacing ECU's in the future (Example instrument cluster, ADAS etc.). The central controller resides in WIRServer. Typically, the Local controller interfaces with applications, queues / schedules the requests from applications and interfaces with central controller to provide the appropriate network interface to the applications. The Local controller also provides network interface on its own when Central controller is not available.

3.2.1.2 WIR-REQ-378639/A-Local Controller Interface With Central Controller



Whenever there is a request from application for an interface to the local controller the local controller requests for an interface from the central controller. If the central controller is available and provides an interface, the local controller provides the same interface to requesting application. If the central controller is not available, the local controller provides interface on its own to the requesting application.

The strategy outlined above shall be followed for every request.

3.2.1.3 WIR-REQ-378640/A-Receive Intent Requests

The Local controller shall interface with connectivity requesting apps and receives the intent requests from these applications.

3.2.1.4 WIR-REQ-378641/A-Process Intent Requests

The Local controller shall process the intent request by communicating with central controller from WIRServer. If central controller is not available, Local controller shall act on its own and process the intent requests.

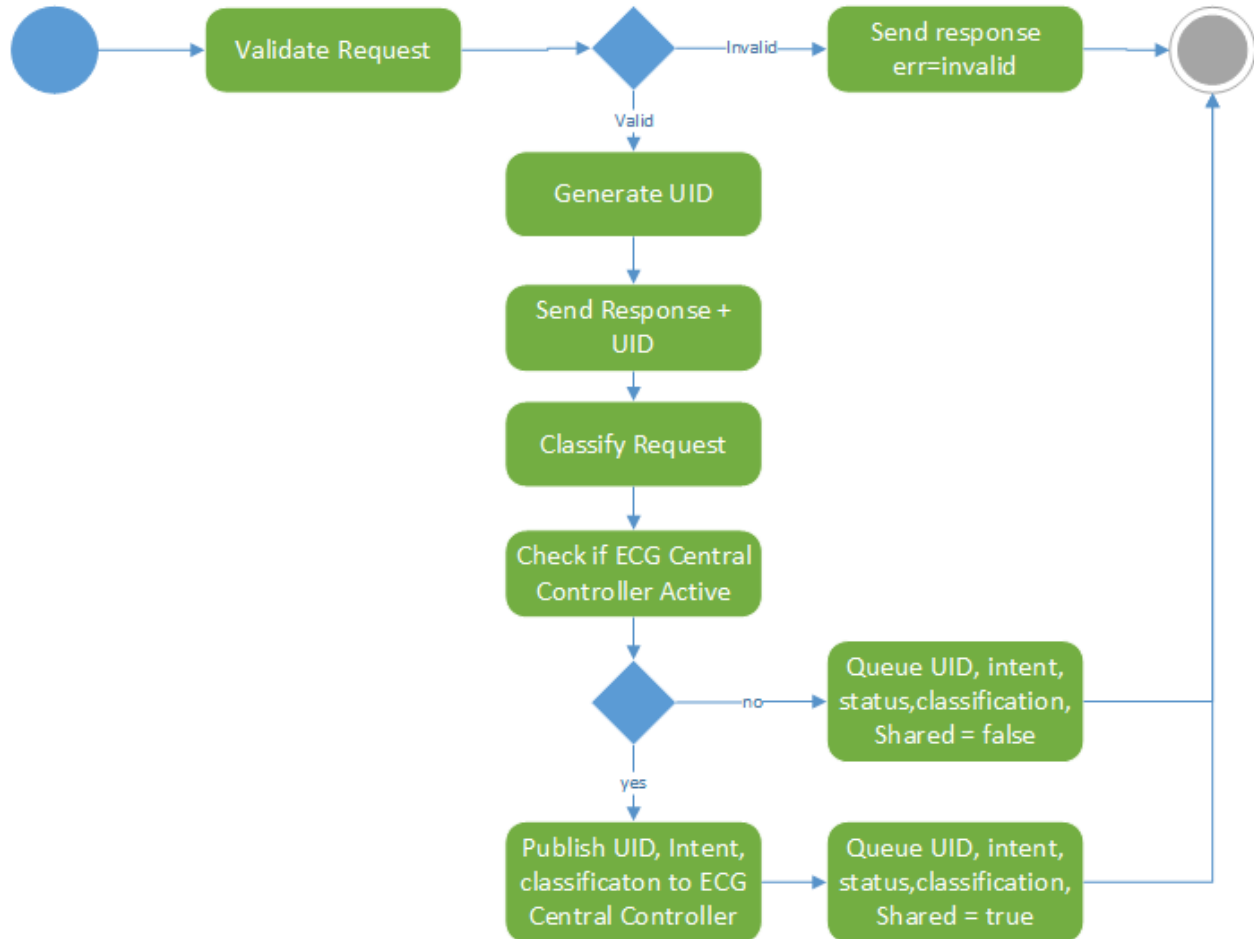
3.2.1.5 WIR-REQ-378642/A-Return Unique Id

The Local controller shall process the intent request and provide a unique identifier as a response to the connectivity requesting application.

The Local controller and requesting application shall use the unique identifier as a reference for all future interactions.

3.2.1.6 WIR-REQ-378643/A-Flow

The below UML diagram describes the flow of connection controller module



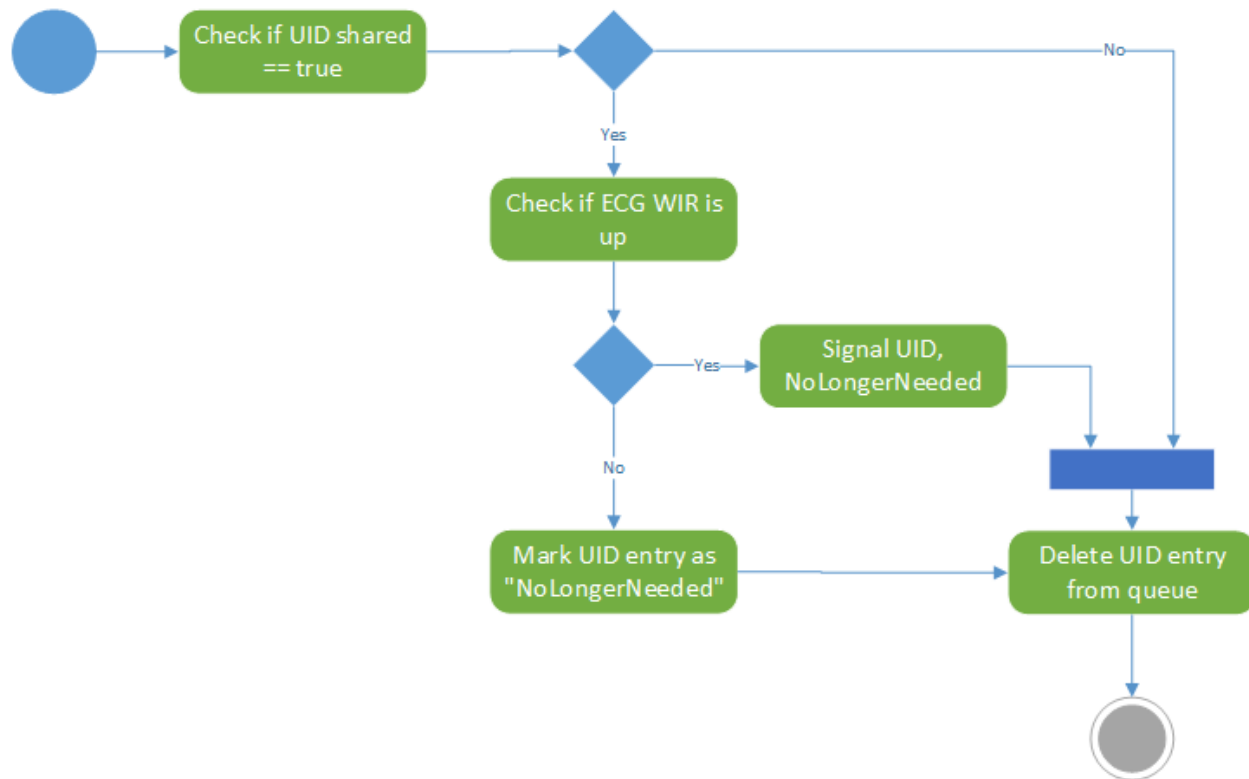
3.2.1.7 WIR-REQ-378644/A-Maintain UID's

The Local connection controller shall maintain the active UID's and their status whether

1. Interface provided
2. Interface is used
3. Network interface available
4. Network interface not available

3.2.1.8 WIR-REQ-378645/A-Remove UID's From Queue Once Application Notifies It Is No Longer Needed

The following UML activity depicts the behavior of the local connection controller when an application signals a UID is complete and no longer needed.



3.2.1.9 WIR-REQ-378646/A-Responding To Queries From Applications Requesting Connectivity

The Local controller will also be responsible for responding to queries such as interface status such as Interface availability, interface health etc. from connectivity demanding applications. The Local controller will use the unique identifier to identify the specific query referenced.

The Local controller shall use the unique ID as a reference to respond to queries.

3.2.1.10 WIR-REQ-378647/A-Query Response Details

The Local controller shall provide one of the following statuses back to the requesting application

1. Success – If interface has been provided successfully
2. Not supported – if the requested interface for the application is not supported
3. In progress – If the requested interface is not yet available to be provided
4. Interface – The IP address of the interface provided
5. Type of Interface – Whether the interface is WIFI (SYNC / TCU), Cellular, LAN etc.

3.2.1.11 WIR-REQ-378648/A-Scheduling

The local controller is also responsible for scheduling the different connectivity interface requests received from applications if central controller is not available.

3.2.1.12 WIR-REQ-378649/A-Removal From The Scheduling Queue

Once connection has been provided to application the request shall be removed from scheduling queue by local controller.

3.2.1.13 WIR-REQ-379781/A-Event **callback**

Central controller shall pass following events to WIR application through callbacks.

- Allocation – When initial allocation is done.
- Network down – When previously allocated network interface is not valid anymore or both WIRClient2 and WIRClient5 lose connection
- Network up – When new network interface becomes available
- Policy update - When policy table for the application has been changed



- Pause – Requesting WIR app temporary stop using the interface
- Resume – Requesting resume previously paused network interface
- Stop – Stop using the network interface until release
- RequestActivityOverInterface – WIR is asking if there is any pending data activity over the interface

3.2.2 Use Cases

3.2.2.1 WIR-UC-REQ-378650/A-Local controller provides connection interface to requesting application

Actors	WIRClient1, WIRServer, WIRClient2, WIRClient3, WIRClient4, WIRClient5, WIRClient6
Pre-conditions	1. Application requests WIR local controller for connection interface
Scenario Description	Local controller works with central controller to provide interfaces to requesting application
Post-conditions	Local controller successfully sends the request to the central controller Central controller processes the request and provides interface back to the local controller
Interfaces	
Notes	Exception use cases Central controller not available

3.2.2.2 WIR-UC-REQ-378651/A-Central controller not available

Actors	WIRClient1, WIRServer, WIRClient2, WIRClient3, WIRClient4, WIRClient5, WIRClient6
Pre-conditions	1. Application requests WIR local controller for connection interface
Scenario Description	Local controller requests central controller to provide interfaces to requesting application. But central controller is not available
Post-conditions	Local controller provides interface to the requesting application on its own.
Interfaces	
Notes	

3.2.3 White Box View



3.3 WIR-FUN-REQ-378652/A-Intents

3.3.1 Requirements

3.3.1.1 WIR-REQ-378653/A-Intent Purpose

When applications request network interfaces to create sockets via the controller API defined above, intent is included as a parameter of the request. The intent parameter provides the WIR context information about when, and with what kind of connection, to service the request.

3.3.1.2 WIR-REQ-378654/A-Types Of Intents

The WIRClient1 applications can leverage the following types of intents

1. Fore ground
2. Back ground
3. Back ground guaranteed
4. Special policy
5. LAN Only

The WIRClient2 applications can leverage the following types of intents

1. Fore ground
2. Back ground
3. Back ground guaranteed
4. Special policy

The WIRClient3 applications can leverage the following types of intents

1. Fore ground
2. Back ground
3. Back ground guaranteed
4. Special policy
5. LAN Only

The WIRClient4 applications can leverage the following types of intents

1. Fore ground
2. Back ground
3. Back ground guaranteed
4. Special policy
5. LAN Only

The WIRClient5 applications can leverage the following types of intents

1. Fore ground
2. Back ground
3. Back ground guaranteed
4. Special policy

The WIRClient6 applications can leverage the following types of intents

1. Fore ground
2. Back ground
3. Back ground guaranteed
4. Special policy
5. LAN Only

The WIRServer applications can leverage the following types of intents

1. Fore ground
2. Back ground
3. Back ground guaranteed
4. Special policy
5. LAN Only

**3.3.1.3 WIR-REQ-378655/A-Intent Priority**

The WIRClient1 applications shall specify the priority of the request along with the intent.

If priority is selected then that application will be treated with higher priority compared to an application which does not select the priority flag.

3.3.1.4 WIR-REQ-378658/A-Intent Structure

The Logical intent structure is as defined below:

Intent	Priority Level	Expiry timer	WiFi Preferred	LAN Preferred	Timer expiration	Supported Interface	Allowed app	Data Size	Band width Uplink	Band width Downlink
Foreground	1: Mission Critical 2: Customer Facing 3: Background/ Low (default) 4: Emergency	NA	Y : WiFi and fallback to cellular N : Cellular and fallback to WiFi	Y: 10G PHY N: Check WiFi preferred	NA	WIRClient2 Cellular APN1, WIRClient2 Cellular APN2, WIRClient5 Cellular APN1, WIRClient5 Cellular APN2, WiFi, 10G PHY	All	NA	1-10 Mbps 0 means unknown and treated as 0~1 Mbps	1-10 Mbps 0 means unknown and treated as 0~1 Mbps
Background	NA	NA	NA	NA	Return failure	WiFi	All except FCI app	NA	NA	NA
Background guaranteed	NA	1~N	NA	NA	Fallback to cellular connection	WiFi and fallback to cellular. After fallback to cellular and WiFi becomes available, WIR shall provide WiFi again	All except FCI app	NA	NA	NA
Special	NA	NA	NA	NA	NA	WIRClient2 Cellular APN1 (FCI), WIRClient5 Cellular APN1 (FCI), WIRClient2 Cellular APN2 (WiFi HotSpot), WiFi with SSID and password (FCI and provisioning)	FCI app Provisioning app WiFi HotSpot app	NA	NA	NA



LAN Only	1-12 (See Priority Defined in WIRServer SDS 10G PHY AVDT Prioritization)	1~N	Y : Fallback to WiFi after expiry timer N : LAN only, wait until available	NA	If WiFi Preferred flag set to Y, fallback to WiFi upon timer expiration	10G PHY WiFi	Data Transfer apps	0-2500 MB	NA	NA
----------	---	-----	---	----	---	--------------	--------------------	-----------	----	----

3.3.1.5 WIR-REQ-378659/A-Intent processing

A. Foreground intent

Foreground intent is for cellular or WiFi interface. If "WiFi Preferred" enabled, WIR shall try WiFi until time out and fallback to cellular interface. If "WiFi Preferred" not enabled, WIR shall try cellular until time out and fallback to WiFi interface.

If timer value is 0, WIR shall not try fallback but return failure. If "LAN Preferred" enabled, WIR shall try LAN interface and fallback to checking the "WiFi Preferred" setting. If "LAN Preferred" not enabled, WIR shall check if "WiFi preferred" is enabled.

B. Background intent

WIR shall try to allocate WiFi interface until time out when timer filed is not 0. If the timer field is set to 0, WIR shall try to allocate WiFi interface until ignition state change. After ignition state change, WIR shall cancel any pending background intent request.

C. Background guaranteed intent

WIR shall try to allocate WiFi interface until time out. Timer value 0 is not allowed and WIR shall return error if timer value is 0.

If off-peak is not set, upon timer expiration, WIR shall convert it to foreground intent and fallback to cellular allocation.

If off-peak is set, upon timer expiration, WIR shall convert it to off-peak intent.

After ignition off, WIR shall cancel any pending background guaranteed intent request without off-peak flag.

If off-peak flag is set, WIR shall check off-peak is allowed DID. If off-peak is allowed, WIR shall convert it to off-peak intent regardless timer expiration state. If off-peak is not allowed, WIR shall return failure.

D. Special intent

Special Intent is for cellular APN1 (FCI), cellular APN2 (WiFi Hotspot application) or WiFi interface (Amazon application). All other applications are not allowed for special intent and WIR shall reject. Since Special intent for WiFi interface provides SSID and password, WIR shall not try to get those values from WiFi HMI but use supplied values. WIR shall check APN ID is correct for corresponding application.

E. LAN Only intent

LAN Only intent is for WiFi or LAN interface. If "WiFi Preferred" enabled, WIR shall try LAN until timer value expires, then fallback to WiFi. If "WiFi Preferred" not enabled, WIR shall wait until LAN is available.

Timer Value: 1~N is a uint32 hex value (range: 0-0xFFFFFFFF) that corresponds to seconds which is specified in intent request.

3.3.1.6 WIR-REQ-378661/A-Intents And Possible Interfaces

The below table describes at a high level the different intents and their possible interfaces:

Intent state	Possible interfaces	Traffic classifications
Foreground	WIRClient2 APN1, WIRClient5 APN1, WIRClient2 APN2, WIRClient5 APN2, WIRClient1 WLAN services, WIRClient2 WLAN services, WIRClient5 WLAN services, WIRClient3 LAN	Traffic classification – 1 Traffic classification – 2 Traffic classification – 3



Background, Background guaranteed but timer not expired	WIRClient1 WLAN services, WIRClient2 WLAN services, WIRClient5 WLAN services	Traffic classification – 4 Traffic classification – 5
Background guaranteed timer expired	WIRClient2 APN1, WIRClient5 APN1, WIRClient2 APN2, WIRClient5 APN2	Traffic classification – 4
Special	WIRClient2 APN1, WIRClient5 APN1, WIRClient2 APN2, WIRClient5 APN2, WIRClient1 WLAN services, WIRClient2 WLAN services, WIRClient5 WLAN services, Applink – WIFI* Applink – cellular*	Traffic classification – 1 Traffic classification – 2 Traffic classification – 3 Traffic classification – 4 Traffic classification – 5
LAN Only, WiFi preferred not selected OR WiFi preferred selected, timer not expired	WIRClient3 LAN	Traffic Classification – 1 Traffic Classification – 4
LAN Only, WiFi preferred selected, timer expired	WIRClient1 WLAN services, WIRClient2 WLAN services, WIRClient5 WLAN services	Traffic Classification – 1 Traffic Classification – 3 Traffic Classification – 4

* Applink is currently out of scope

3.3.1.7 WIR-REQ-378662/A-Bandwidth Shaping

WIRClient1 / Client 2 / Client3 / Client4 / Client5 / Client6 / Server shall support Bandwidth shaping to achieve Traffic classification priority. For example, an application with priority customer facing shall have a better connection than an application with priority background/low.

3.3.2 Use Cases

3.3.3 White Box View



3.4 WIR-FUN-REQ-378664/A-Policy Manager

3.4.1 Requirements

3.4.1.1 WIR-REQ-378665/A-Purpose

Applications shall request connectivity interface from WIR by sending an intent request which consists of Tokens and URL, WIR shall validate the application intent request against the intent privileges set for every application. The connectivity privileges of every application shall be stored in WIRClient1, WIRClient2, WIRClient3, WIRClient4, WIRClient5, WIRClient6, WIRServer and shall be updatable over the air from backend.

3.4.1.2 WIR-REQ-379782/A-Load Balance Validation

WIRServer shall check that application is privileged to use cellular interfaces on both WIRClient2 and WIRClient5 before calling load balancing function (defined later in this document). If an application is privileged to only use either WIRClient2 or WIRClient5, WIRServer shall only assign an application interface to the applicable WIRClient.

3.4.1.3 WIR-REQ-378666/A-Global Application ID

Every application shall have a global application ID which shall be maintained in the Ford cloud. This global application ID shall be used by WIR to identify applications, validate applications and map intent and URL privileges for the application. The individual applications when requesting network interfaces shall supply the Application ID along with the intent.

3.4.1.4 WIR-REQ-378667/A-Intent Policy Table Privileges

Each application intent privilege shall be specified using a 2 byte value.

3.4.1.5 WIR-REQ-378668/A-Intent Policy Table Privilege Breakdown

Each application intent privilege can be specified using a 2 byte value. The encoding of the 2 byte value is defined below (ex. bit 0 is intended for WIRClient2 cellular APN1, bit 1 is intended for WIFI etc.):

Interfaces	Bit position 1- enabled 0 - disabled
WIRClient2 Cellular APN1	0 (LSB)
WIFI	1
WIRClient2 Cellular APN2	2
Reserved	3
Applink cellular	4
Applink WIFI	5
Offpeak	6
WIRClient2 Cellular APN3*	7
WIRClient2 Cellular APN4*	8
Factory WIFI	9
WIRClient5 Cellular APN1	10
WIRClient5 Cellular APN2	11
10G PHY LAN	12
Reserved Modem Preference**	13
Reserved	14-15 (MSB)

APN1, APN3 and APN4 are mutually exclusive. An application shall enable only one of those 3 APNs. Ford server shall guarantee this rule is satisfied before pushing new policy table to vehicle. APN3 and APN4 are available for China vehicle only and are not available for V713 vehicle.

**Reserved Modem Preference is used for future dual modem load balancing assignment flexibility and bit position will be set to determine the preference as follows: 0 – disabled = TCU, 1 – enabled = TCU-B.



An interface is allowed when corresponding bit is set. Bit 0 is LSB (0x01) and bit 15 is MSB (0x80).

3.4.1.6 WIR-REQ-378669/A-Intent Policy Table Structure

The structure of the intent policy table is as defined below:

Application ID	Application	Connectivity intent permissions

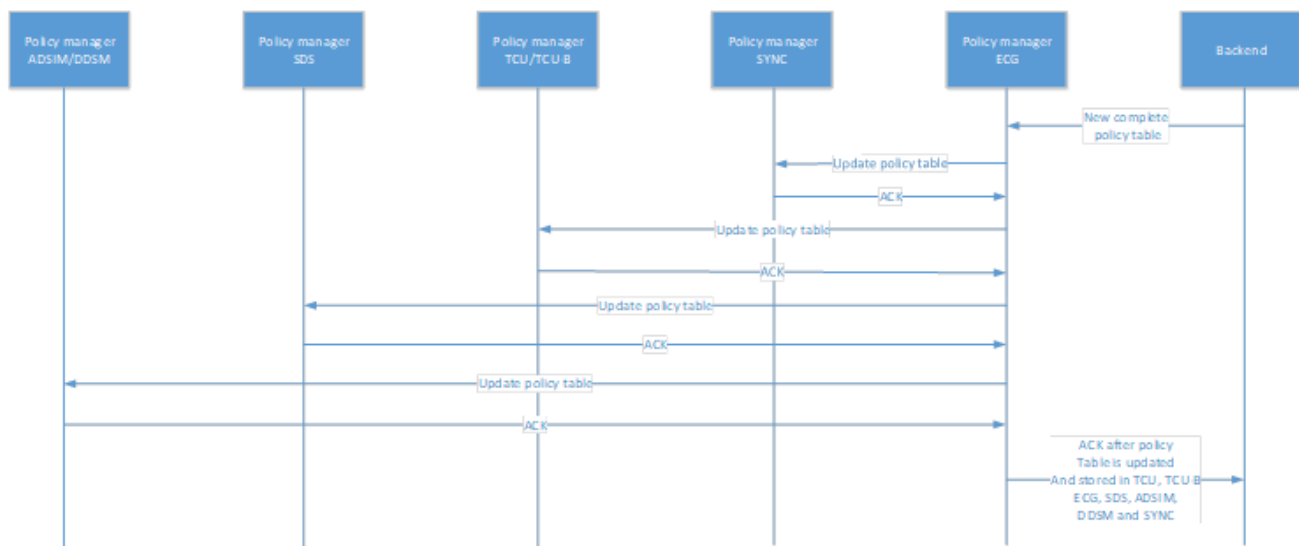
3.4.1.7 WIR-REQ-378670/A-Intent Policy Storage

The WIRClient1 local controller shall store intent policy privilege.

3.4.1.8 WIR-REQ-378671/A-Intent Policy Update

Whenever there is an intent policy table change then Cloud / Backend pushes this information to the WIR policy manager residing in WIRServer. The policy table pushed by cloud shall be complete table, not delta. The WIRServer then validates the change, updates its own policy table and transmits the policy table update to WIRClient6, WIRClient5, WIRClient4, WIRClient3, WIRClient2 and WIRClient1. The WIRClient1, WIRClient2, WIRClient3, WIRClient4, WIRClient5 and WIRClient6 shall in turn update their own policy tables.

The intent policy table update shall be as depicted in the diagram below:



3.4.1.9 WIR-REQ-378675/A-Intent Policy Update Acknowledgement

Once policy table is updated successfully, WIRClient1 shall provide an acknowledgement to WIRServer.

3.4.1.10 WIR-REQ-378676/A-Intent Policy Update Notification To Applications

The WIRClient1 in addition to updating the policy table shall also publish the message via SoA/WIR-APIs so that the individual application shall know there is a change in policy privilege.

3.4.2 Use Cases

3.4.2.1 WIR-UC-REQ-378677/A-Cloud sends updated policy table to WIR

Actors	WIRClient1, WIRServer, WIRClient2, WIRClient3, WIRClient4, WIRClient5, WIRClient6, Ford Cloud
Pre-conditions	1. Ford cloud sends updated policy table to vehicle
Scenario Description	WIRServer receives updated policy table from Ford Cloud and sends the updated policy to all WIRClients



Post-conditions	WIRServer receives the updated policy table from Ford Cloud WIRServer updates the policy and sends policies to all WIRClients WIRClients update the policy table and sends confirmation to WIRServer WIRServer sends confirmation to Ford cloud
Interfaces	
Notes	Exception use case 1. Policy not updated in WIRServer / WIRClients

3.4.2.2 WIR-UC-REQ-378678/A-Policy not updated in WIRServer / WIRClients

Actors	WIRClient1, WIRServer, WIRClient2, WIRClient3, WIRClient4, WIRClient5, WIRClient6, Ford Cloud
Pre-conditions	1. Ford cloud sends updated policy table to vehicle
Scenario Description	WIRServer receives updated policy table from Ford Cloud and but not able to update policy table in vehicle
Post-conditions	WIRServer receives the updated policy table from Ford Cloud WIRServer updates the policy and sends policies to all WIRClients Any combination of WIRClients could not update the policy WIRServer sends failure to Ford cloud
Interfaces	
Notes	

3.4.3 White Box View



3.5 WIR-FUN-REQ-378679/A-Tunnel Manager

3.5.1 Requirements

3.5.1.1 WIR-REQ-378680/A-Purpose

The main purpose of WIR tunnel concept is to enable all networking channels, which includes exposing and sharing network interfaces between the WIRClient1, WIRClient2, WIRClient3, WIRClient4, WIRClient5, WIRClient6 and WIRServer. WIRClient1's Wi-Fi is exposed to WIRServer, WIRClient2 and WIRClient3, WIRClient4, WIRClient5 and WIRClient6. WIRClient2's Cellular and Wi-Fi interfaces are accessible from WIRClient1, WIRClient3, WIRClient4, WIRClient5, WIRClient6 and WIRServer. WIRClient3's 10G PHY LAN interface is accessible from WIRClient1, WIRClient2, WIRClient4, WIRClient5, WIRClient6 and WIRServer. WIRClient5's Cellular and Wi-Fi interfaces are accessible from WIRClient1, WIRClient2, WIRClient3, WIRClient4, WIRClient6 and WIRServer.

3.5.1.2 WIR-REQ-378681/A-Tunnel Support

The WIRClient1 module shall support tunneling protocol. Each tunnel is used for virtualizing a network interface.

3.5.1.3 WIR-REQ-378682/A-WIRClient1 IP Address

WIRClient1 shall support receiving IP address from WIRServer module. The WIRServer module and WIRClient1 module shall make sure that all tunnel IP address are randomly assigned every ignition cycle (Transition from Ignition OFF to Ignition Run).

3.5.1.4 WIR-REQ-378683/A-Ignition Cycle When Tunnel IP Address Is Already In Use

If the tunnel IP address is already in use and Ignition cycle (Transition from Ignition OFF to Ignition Run) happens then WIRServer and WIRClient1 shall treat this as an exception and shall NOT reassign IP address but shall continue to use current IP address.

3.5.1.5 WIR-REQ-378684/A-IP Address Assignment

WIRClient1 shall support assigning the IP address received from WIRServer to the tunnel endpoint.

3.5.1.6 WIR-REQ-378685/A-IP Aliasing

WIRClient1 shall support IP aliasing in WIR, IP aliasing is used to add IP addresses to existing Ethernet device (eth0) to represent tunnel source addresses (ex. 10.11.0.1 and 10.11.0.5 on WIRClient1 side). It is also used to assign IP address (ex. 10.1.0.5) to created tunnel device (tun1).

3.5.1.7 WIR-REQ-378686/A-Allow WIRClient1 Edge Interface Access To Ethernet Connected ECU's

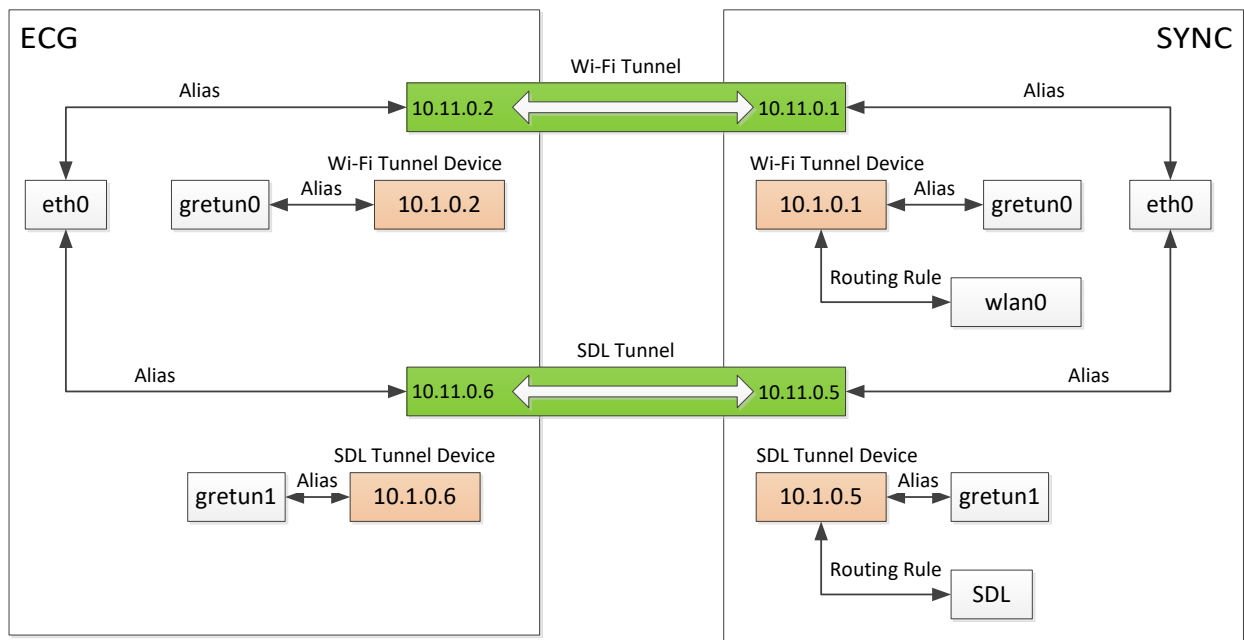
WIRClient1 shall allow usage of edge interface namely WIFI from another Ethernet connected ECU via tunnel concept.

3.5.1.8 WIR-REQ-378687/A-Allow WIRClient1 Applications To Access Edge Interface Of Other Ethernet Connected ECU's

WIRClient1 shall allow applications within WIRClient1 to access WIRClient2 cellular network, WIRClient2 WIFI network, WIRClient3 10G PHY, WIRClient5 cellular network and WIRClient5 WIFI network via Tunneling concept.

3.5.1.9 WIR-REQ-378688/A-WIRClient1 Interfacing With WIRServer Example

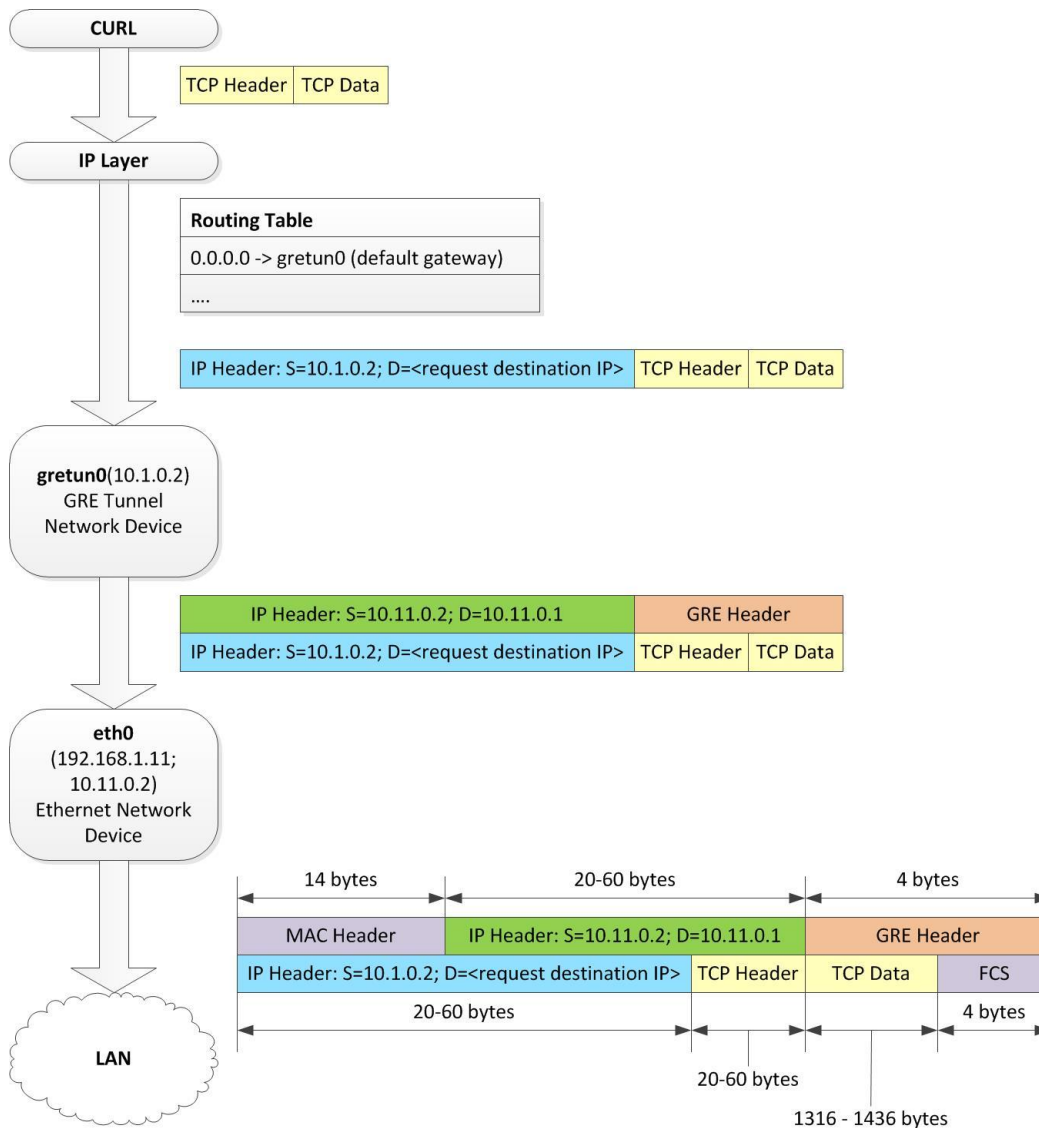
The below diagram outlines (provides an example) of how WIRClient1 will interface with WIRServer and expose its edge interfaces:



GRE tunnel encapsulation flow

3.5.1.10 WIR-REQ-378689/A-Sending A Request From WIRServer

Figures below provide more details on how data exchange works in case of using tunnels. In particular, they show how packages are created and sent from WIRServer to WIRClient1 and Internet and back. These sections assume a client application on the WIRServer would like to use a CURL library to make a cloud request. This request is routed over WIRClient1's Wi-Fi interface.



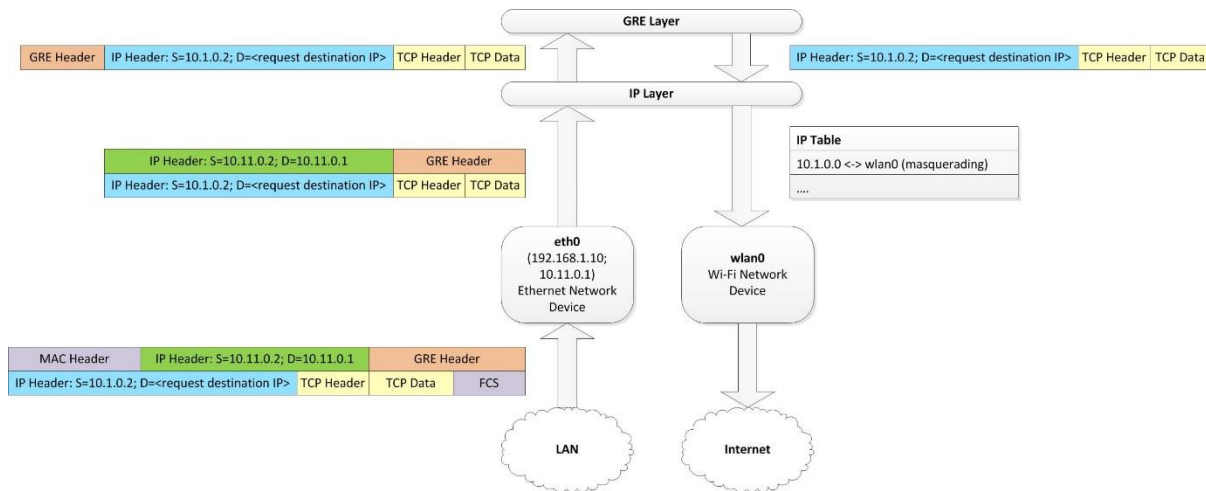
Sending request on the WIRServer

CURL is configured to use gretun0 network device to send its packets. That is why it uses gretun0 IP address as source address. The destination IP address is the final cloud destination. According to the routing table rules that we configure in advance, all packages, coming to Internet, should be sent via tunneling device (gretun1). Tunneling device wraps original IP packet, adding GRE header and new IP header to it. As there is a tunnel established from address 10.11.0.2 to address 10.11.0.1, gretun0 sets in newly added IP header. Notice that the tunneling device IP (10.1.0.2) is not presented in external IP header.

After that packet is wrapped by Ethernet device eth0 with a MAC header and sent to LAN.

This is an example proposal.

3.5.1.11 WIR-REQ-378690/A-Forwarding The Request To WIRClient1 WIFI Example Proposal

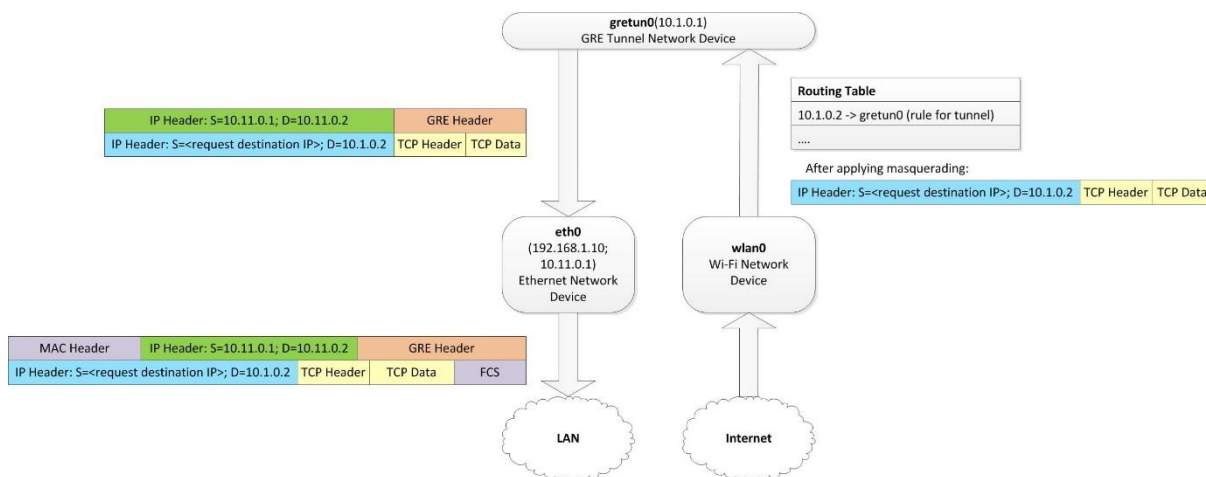


Sending request on WIRClient1

When packet arrives at the WIRClient1 Ethernet interface, it consists of the original IP packet, wrapped in a GRE Header, wrapped in a local IP packet, and finally wrapped in a MAC header. The Ethernet device removes MAC header and FCS and passes the local IP packet to the IP layer. The IP layer makes sure that current system has the IP address (10.11.0.1), which is the destination address in the IP packet header, and removes the IP header and passes the packet to the GRE layer. The GRE interface removes the GRE header and the original IP packet back to the IP routing layer. The IP routing layer sees the packet source address is 10.1.0.2 and matches the packet to an IP routing rule which routes all packets from the 10.1.0.2 IP address to the network device wlan0 (in this particular example we assume that we are sending packets via Wi-Fi). The packet is then transmitted via the WIRClient1 Wi-Fi interface.

3.5.1.12 WIR-REQ-378691/A-Receiving Response On WIRClient1 Module Example Proposal

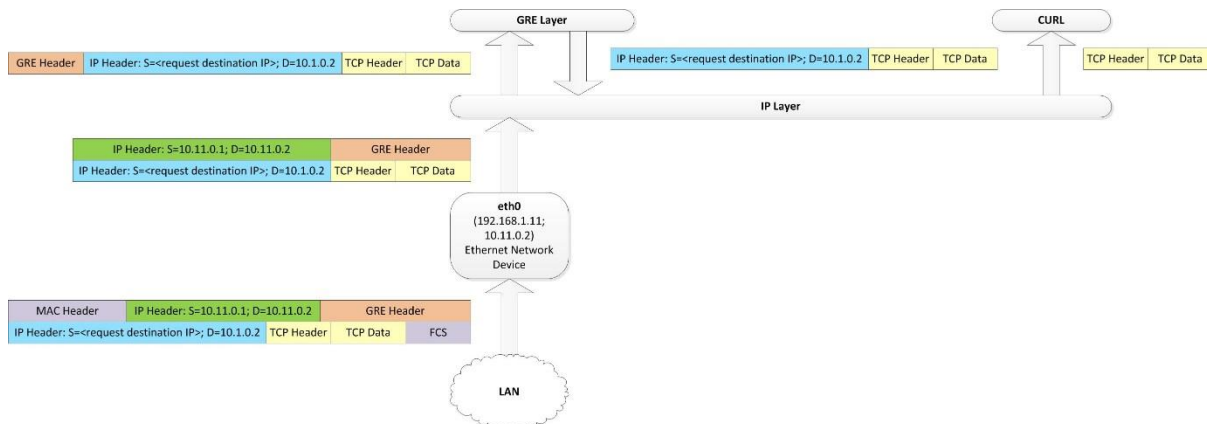
When the destination host in the cloud sends a response IP packet, it arrives at the WIRClient1 Wi-Fi interface. The response IP packet specifies the original source IP as 10.1.0.2, the address of the original GRE tunnel. The WIRClient1 IP layer identifies a rule that routes all packets destined for 10.1.0.2 to gretun0. Similar to earlier, when WIRServer sent a packet through gretun1, the return packet is wrapped in a GRE header by gretun0, and then by the WIRClient1 Ethernet interface.



Receiving response on WIRClient1

3.5.1.13 WIR-REQ-378692/A-Receiving Response On WIRServer Module Example Proposal

Similarly, response packet, when it arrives at the WIRServer interface, the MAC header is stripped by the WIRServer Ethernet interface. The local IP packet is stripped and routed to the GRE layer. The GRE layer strips the GRE header, and sends the encapsulated IP packet back to the IP layer. From there, the response packet is routed down the network stack to the socket being used by the CURL library.



Receiving response on WIRServer

3.5.2 Use Cases

3.5.3 White Box View



3.6 WIR-FUN-REQ-378693/A-Diagnostics

3.6.1 Requirements

3.6.1.1 WIR-REQ-378694/A-Purpose

The purpose of Diagnostics module for WIRClient1 involves diagnosing WIFI connectivity and providing App requests and interface information to WIRServer when WIRClient1 provides interface.

3.6.1.2 WIR-REQ-378695/A-WIFI Diagnostics

Each time the WIFI interface is down, the WIRClient1 shall log the below information.

- SSID (Xfinity / non Xfinity (customer SSID is private and no need to store this information just classification is fine)
- Connect/Disconnect timestamps of the access point (including pause, stop, resume commands)
- Data rate
- Band and channels utilized
- RSSI statistics
- Chipset roles / use
- Statistics about the data transmitted and received if possible
- Reason for disconnect

3.6.1.3 WIR-REQ-378696/A-SDL Diagnostics

Each time the SDL interface is down, the applicable WIRClient shall log the below information.

Each time the SDL interface is down or the edge interface for SDL is down (example cellular / WIFI) the applicable WIRClient shall log the below information:

- SDL channel used (Bluetooth / WIFI direct / USB)
- Bluetooth RSSI
- Edge interface RSSI (cellular / WIFI)
- Connect/Disconnect timestamps of the access point (including pause, stop, resume commands)
- Data rate
- Band and channels utilized if any
- Reason for disconnect

3.6.1.4 WIR-REQ-378697/A-Diagnostics Information Storage

The WIRServer diagnostics module shall store WIFI diagnostics information and try to upload it to Ford cloud through diagnostics framework via only WiFi connection. Cellular connection is not allowed for the upload. If it can't be uploaded, WIRServer shall try to upload it until WIR_WIFI_DIAGNOSTIC_STORAGE_TIME. After WIR_WIFI_DIAGNOSTIC_STORAGE_TIME, WIRServer shall delete it and as a result, it will not be uploaded to the cloud.

3.6.1.5 WIR-REQ-378698/A-On demand diagnostic request

Ford cloud shall be able to request diagnostic data upload. WIRServer shall upload any pending diagnostic data to cloud immediately when it is requested. WIRServer or diagnostic framework shall try WiFi connection first and fallback to cellular for the upload.

3.6.1.6 WIR-REQ-378699/A-Diagnostics Information Sending To Cloud

The applicable WIRClient diagnostics module shall store the diagnostics information and send the diagnostics data to the cloud via diagnostics framework upon request from diagnostic framework.

3.6.1.7 WIR-REQ-378700/A-Sending The App Requests And Interfaces Provided To App To WIRServer Central Controller

If in the event the WIRClient1 provided interfaces and Tokens to requesting applications without WIRServer central controller involvement then WIRClient1 shall store the details and shall send to WIRServer once WIRServer central controller becomes available.



Unique ID and Time stamp	Feature ID	Interface provided and Token information (This is just an example)	Edge Interface	Health of the all edge interfaces available
101 06142017 2.25.11 pm	Online traffic (example 701)	10.100.10(emer, FCI etc.) and Token Key	TCU or TCU-B cellular	Qdiscs information if available
102 06142017 2.25.21 pm	Online traffic (example 701)	10.100.11(multi +applink)	SDL	Qdiscs information if available
103 06142017 2.25.25 pm	Parsed (example 702)	10.101.14(background generic)	TCU or TCU-B WIFI	Qdiscs information if available

If Central controller is available and Central controller provides interface in that event WIRClient1 shall provide UniqueID, Feature ID to the WIRServer Central controller.

3.6.1.8 WIR-REQ-379784/A-Pause, resume and stop event log

WIRServer shall create pause, resume and stop event log entry. When WIRServer calls pause, resume or stop event, it shall create an entry for each event with following items.

- Timestamp
- Application ID
- Allocation ID
- Event ID (pause, resume or stop)

This information shall be stored by WIRServer for a period of 1 month and shall be sent to the diagnostics framework based on Ondemand request from Cloud via FTCP message.

For the Ondemand request message from Cloud please refer FTCP profile.

3.6.1.9 WIR-REQ-379785/A-Cellular event log

WIRServer shall create a cellular event log when following events are happened.

Cellular event to trigger event log

- No service
- 3GPP rejection

Items to log

- Timestamp
- Event Id (No service or rejection)
- TCU ID (TCU or TCU-B)
- APN ID
- MCC
- MNC
- LAC
- Cell ID
- RAT (LTE, HSPA, UMTS, GSM)
- RSSI
- RSRQ
- RSRP
- SINR or Ec/Io

All log items shall be the value when the event occurred or last known value.



3.6.2 Use Cases

3.6.2.1 WIR-UC-REQ-378701/A-WIR provides diagnostics information to cloud based on request from Cloud

Actors	WIRClient1, WIRServer, WIRClient2, WIRClient3, WIRClient4, WIRClient5, WIRClient6, Ford Cloud
Pre-conditions	1. Customer triggers bug report
Scenario Description	Customer has a connectivity issue and triggers bug report to be sent to the cloud via on demand diagnostics
Post-conditions	WIR receives the request for WIR diagnostics logs WIR retrieves the stored logs WIR sends the logs to diagnostics framework Diagnostics framework sends the logs to Ford cloud
Interfaces	
Notes	

3.6.3 White Box View



3.7 WIR-FUN-REQ-378702/A-Data Usage

3.7.1 Requirements

3.7.1.1 WIR-REQ-378703/A-Data usage calculation

The Data usage shall be calculated per application by WIRClient1, WIRClient2, WIRClient3, WIRClient4, WIRClient5, WIRClient6 and WIRServer.

3.7.1.2 WIR-REQ-378704/A-Data usage calculation scope

The Data usage shall be calculated per application in each of the Ethernet connected modules (WIRClient1, WIRClient2, WIRClient3, WIRClient4, WIRClient5, WIRClient6 and WIRServer).

3.7.1.3 WIR-REQ-378705/A-Data usage aggregation per application

WIRClient1 shall provide calculated data usage to WIRServer where data usage is aggregated.

If the same application resides in WIRClient1, WIRClient2, WIRClient3, WIRClient4, WIRClient5, WIRClient6 and WIRServer then data usage shall be calculated collectively and aggregated by WIRServer.

3.7.1.4 WIR-REQ-378706/A-Data usage and edge interface

Data usage for each of the application shall be calculated individually for each of the application and each individual interface and stored in WIRServer module and shall be sent to cloud upon request. The WIRServer shall be responsible for maintaining / aggregating the calculated data usage details. The WIRClients (WIRClient1, WIRClient2, WIRClient3, WIRClient4, WIRClient5, WIRClient6) shall provide the WIRServer details of data usage calculated per App calculated locally.

Below is an example table which WIRServer maintains.

For example:

Data usage for OTA app for period

OTA app using cellular: 10 MB

OTA app using WIFI: 500 MB

	APN1		TCU Wi-Fi		SYNC Wi-Fi		App Link PT		APN1 Off peak	
Application	Sent	Recv	Sent	Recv	Sent	Recv	Sent	Recv	Sent	Recv
ECG OTA			2MB	40MB	10MB	100MB			1MB	30MB
ECG FCI	2.5MB	2.5MB								
SYNC Voice Rec	10MB	10MB								
SYNC VICC	50MB	100MB			10MB	20MB				
On Line Traffic							5MB	10MB		
...										

For SiriusXM, WIRServer shall report 2 separate entries, stream data and user account data. For each data calculation, please refer to WIR-REQ-378717-SiriusXM data usage report.

Data usage shall be accumulated until key off and door opened. At key off and door opened, WIRServer shall report data usage to Ford backend.

3.7.1.5 WIR-REQ-378707/A-Data usage period

Data usage shall be calculated continuously and aggregated until reset is issued (Reset is initially planned from cloud but design shall be flexible to accommodate reset from in vehicle HMI as well).

Data usage period is maintained by Ford cloud.



3.7.1.6 WIR-REQ-378708/A-Data usage storage

At any point in time WIRServer shall aggregate data usage per app per interface and store the same for the period.

3.7.1.7 WIR-REQ-378709/A-Data usage period reset

If WIRServer gets reset command from cloud or in vehicle HMI, it shall reset the period (Erase the data usage per application per edge interface and start calculating again).

3.7.1.8 WIR-REQ-378710/A-Data usage request processing from Ford cloud

The WIRServer shall be capable of receiving request to data usage from cloud via FTCP.

For more details on the type and content please refer FTCP protofile.

3.7.1.9 WIR-REQ-378711/A-Data usage data transmission to Ford cloud

The WIRServer shall be capable of transmitting the Data usage any time to the cloud upon request from the cloud FTCP.

For more details on the type and content please refer FTCP protofile.

3.7.1.10 WIR-REQ-378712/A-Data usage data transmission format

The WIRServer shall be capable of transmitting data usage for individual application and also as a whole.

3.7.1.11 WIR-REQ-378713/A-Data usage request processing from in vehicle HMI

The WIRServer shall be capable of receiving request to data usage from in vehicle HMI.

3.7.1.12 WIR-REQ-378714/A-Data usage data transmission to in vehicle HMI

The WIRServer shall be capable of transmitting the Data usage any time to the cloud upon request from the in vehicle HMI.

3.7.1.13 WIR-REQ-378715/A-Accuracy of data usage calculation

The Data usage calculation per application shall be 99.9999% accuracy rounded to the nearest Kilobyte.

3.7.1.14 WIR-REQ-378716/A-Data usage timestamp for cellular

The WIRServer shall have a log of data usage per session and associated time stamps for the selected period.

3.7.1.15 WIR-REQ-378717/A-SiriusXM data usage report

The WIRServer shall calculate data amount used by SiriusXM application for cellular and WiFi separate. Data amount for uplink and downlink shall be combined. SiriusXM data usage divided between streaming and server configuration. Data usage ratio for those two categories is available through an API provided by SiriusXM application.

- Stream data through cellular (bytes) = ratio of stream data * total cellular usage
- Stream data through WiFi (bytes) = ratio of stream data * total WiFi usage
- User account management data through cellular (bytes) = ratio of server config data * total cellular usage

User account management data through WiFi (bytes) = ratio of server config data * total WiFi usage.

3.7.2 Use Cases

3.7.3 White Box View



3.8 WIR-FUN-REQ-378718/A-WIFI Connect Reminders - HMI

3.8.1 Requirements

3.8.1.1 WIR-REQ-378719/A-WIFI Connect Reminder Popup

SYNC module shall support the WIFI connect reminder pop up.

Text content and pop up shall be finalized and captured in HMI specification.

3.8.1.2 WIR-REQ-378720/A-Trigger

The WIRServer shall count the number of ignition cycles where Vehicle is not connected to WIFI.

Vehicle WIFI refers to either WIRClient1 WIFI or WIRClient2 or WIRClient5 WIFI.

If the vehicle fails to connect to WIFI for WIFI_CONNECT_REMINDER_COUNTER number of ignition cycles WIRServer shall notify WIRClient1 to display WIFI Connect reminder pop up.

3.8.1.3 WIR-REQ-378721/A-Trigger Frequency

The trigger frequency is WIFI_CONNECT_REMINDER_COUNTER ignition cycles. The initial value for WIFI_CONNECT_REMINDER_COUNTER is 100.

3.8.1.4 WIR-REQ-378722/A-WIFI Connect Reminder Popup User Action

The user shall have an option to select “No” or “Remind me later” to the WIFI connect reminder pop up.

3.8.1.5 WIR-REQ-378723/A-User Selects OK

If the user selects “Ok” then vehicle shall not display the pop up once again.

WIRClient1 shall notify WIRServer about user selection. WIRServer shall reset the WIFI_CONNECT_REMINDER_COUNTER counter and wait for trigger conditions as stated in Trigger requirement.

3.8.1.6 WIR-REQ-378724/A-User Selects Remind Me Later

If the user selects “REMIND ME LATER” then vehicle shall not display the pop up once again until WIFI_CONNECT_REMINDER_TEMP ignition cycles have been completed.

WIRClient1 shall notify WIRServer about User selection. WIRServer shall start another counter WIFI_CONNECT_REMINDER_TEMP and WIFI_CONNECT_REMINDER_COUNTER shall still remain at 100. The value of WIFI_CONNECT_REMINDER_TEMP counter shall be configurable and shall be set to 100.

3.8.1.7 WIR-REQ-378725/A-WIFI_CONNECT_REMINDER Popup Configurable

The WIFI_CONNECT_REMINDER pop up shall be a configurable parameter and shall be turned ON / OFF by WIRServer module based on the configuration.

3.8.1.8 WIR-REQ-378726/A-User Selects “No” Check Box

If the user selects “No” again then the pop up is never displayed again until a master reset occurs.

WIRClient1 communicates this information to WIRServer which tracks this status and never triggers the pop up.

3.8.1.9 WIR-REQ-378727/A-Master Reset

If a master reset is performed by user then WIRClient1 notifies WIRServer of Master reset.

WIRServer will reset the pop up trigger conditions and override any previously selected “No”.

Master reset will reset this and pop up will be brought out again.



3.8.2 Use Cases

3.8.3 White Box View



3.9 WIR-FUN-REQ-378728/A-Captive Portal Check

3.9.1 Requirements

3.9.1.1 WIR-REQ-378729/A-Captive portal check support

WIRClient1 shall support captive portal check operation.

3.9.1.2 WIR-REQ-378730/A-WLAN connection request

WIRClient1 CM receives WLAN connection request from WIRServer CM. The request has SSID and password.

3.9.1.3 WIR-REQ-378731/A-WLAN client

WIRClient1 WLAN client receives a request to connect to an external Wi-Fi Hotspot from WIRClient1 CM.

A customer uses Wi-Fi client menu to select WIRClient1 WLAN client, scan available Wi-Fi Hotspot, select and connect to the Wi-Fi Hotspot.

WIRServer detects available Wi-Fi Hotspot in background scanning. If WIRServer detects available Wi-Fi Hotspot and wants to use Wi-Fi Hotspot connection, it sends Wi-Fi connection request to WIRClient1 CM client with SSID and password. The CM client sends the request to WIRClient1 WLAN client and receives connection result.

3.9.1.4 WIR-REQ-378732/A-WEB engine

WEB engine shall support text input, button click, and hyperlink click by a user.

CPC sends a redirected URI which is specified on Location field of 3XX response and local IP address to WEB engine to be displayed on WIRClient1 UI.

Following shows an example 302 HTTP response.

HTTP/1.1 302 Found
Server: Apache/2.4.23 (Unix)
Location: /nbrd/visit/counter.html
Content-Type: text/html
Content-Length: 0

If user's input triggers HTTP connection and data exchange, WEB engine shall bind the supplied local IP address to make a HTTP connection to the WEB server. WEB engine shall process HTTP response as normal.

If landing page screen becomes background screen, WEB engine instance will be killed, and landing page screen will not be available anymore.

If RVC or popup overrides landing page screen, landing page screen will become foreground screen again when RVC or popup finished, and the customer can continue interacting with the landing page.

In China, a customer should enter his smartphone number on landing page. Clicking a button sends the smartphone number to Wi-Fi service provider and the service provider sends a password to the customer's smartphone as a MT-SMS. The customer has to enter the password and clicking a button cause the page sent to Wi-Fi Hotspot service provider to validate the password. If the password is correct, WLAN gets internet connection through Wi-Fi Hotspot, otherwise the service provider sends an error popup or an error through HTML element like password field as red color.

3.9.1.5 WIR-REQ-378733/A-WIRClient1 UI screen switch

WIRClient1 UI screen switch triggers by a customer or an external event. After WEB engine displayed the landing page if the screen switched to other screen, previous WEB engine display killed and no longer available.

If RVC or popup displayed on top of the landing page, the landing page shall be displayed again when RCV or popup finished. The customer shall be able to interact with landing page once it is displayed again.

3.9.1.6 WIR-REQ-378734/A-WiFi password entering by customer

If a password is required for Wi-Fi connection, WIRClient1 UI shall support password enter screen and pass the password to WIRClient1 or WIRClient2 WLAN. Both WIRClient1 or WIRClient2 WLAN shall be able to request password enter screen and receive the password.



3.9.1.7 WIR-REQ-378735/A-WIRClient1 UI restriction

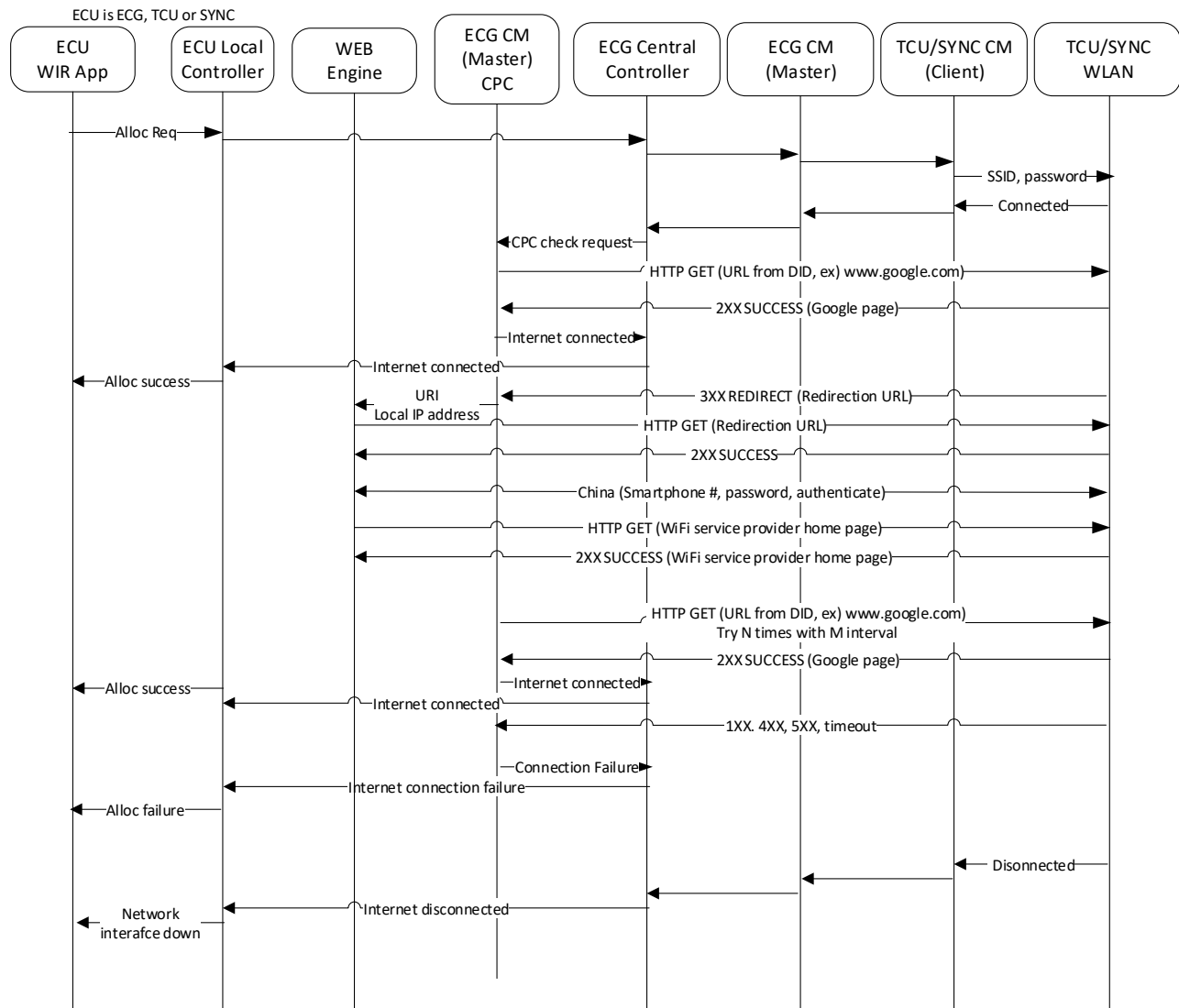
After received a captive portal check request, CPC shall wait CPC_CHECK_INTERVAL seconds. After the wait, CPC shall send HTTP request to CPC_CHECK_URL.

If the HTTP response is 2XX success, CPC determines internet connection through the WLAN is ready and return success. If the HTTP response is any other response or there is no response, CPC shall wait CPC_CHECK_INTERVAL seconds and send next HTTP request until it tried CPC_MAX_CHECK times.

If the HTTP response 3XX redirection, CPC shall send WEB engine display request with URI and local IP address.

3.9.1.8 WIR-REQ-378736/A-CPC check procedure

Following diagram shows CPC check procedure flow.



3.9.2 Use Cases

3.9.2.1 WIR-UC-REQ-378737/A-WIRClient1 Wi-Fi connected to a Wi-Fi Hotspot first time and redirected to landing page

Actors	WIRClient1 Wi-Fi, Wi-Fi Hotspot, customer, WIR
Pre-conditions	WIRClient1 Wi-Fi turned on Customer initiates Wi-Fi scan WIRClient1 Wi-Fi is scanning available Wi-Fi Hotspot signal



Scenario Description	WIRClient1 Wi-Fi finds available Wi-Fi Hotspot and makes a connection to it Wi-Fi Hotspot requires landing page loading and accepts T&C by a customer
Post-conditions	WIRClient1 Wi-Fi finds available SSID Customer selects a Wi-Fi Hotspot to connect from WIRClient1 UI and enters password WIRClient1 Wi-Fi connects to the Wi-Fi Hotspot CPC waits for CPC_CHECK_INTERVAL seconds CPC sends HTTP GET request to connectivity test URL CPC receives HTTP 3XX redirection to landing page response CPC sends redirected URI and local IP address to WEB engine CPC start a wait timer for CPC_CHECK_INTERVAL seconds WEB engine displays the landing page Customer clicks accept button WEB engine binds local IP address, creates HTTP connection and sends HTTP GET request for home page Wi-Fi Hotspot sends home page and WLAN routes it to WEB engine WEB engine displays the home page The wait timer expires CPC sends HTTP GET request to check captive portal stage is done CPC receives HTTP 2XX Success response CPC notifies WIRServer central controller internet connectivity is available WIR saves SSID and password pair for future use
List of Exception Use Cases	
Interfaces	

3.9.2.2 WIR-UC-REQ-378738/A-WIRClient1 Wi-Fi connected to a Wi-Fi Hotspot first time and Wi-Fi Hotspot doesn't have landing page

Actors	WIRClient1 Wi-Fi, Wi-Fi Hotspot, customer, WIR
Pre-conditions	WIRClient1 Wi-Fi turned on WIRClient1 Wi-Fi is scanning available Wi-Fi Hotspot signal
Scenario Description	WIRClient1 Wi-Fi finds available Wi-Fi Hotspot and makes a connection to it. Wi-Fi Hotspot provides internet connection without landing page redirection
Post-conditions	WIRClient1 Wi-Fi finds available SSID Customer selects a Wi-Fi Hotspot to connect from WIRClient1 UI WIRClient1 Wi-Fi connects to the Wi-Fi Hotspot CPC sends HTTP GET request to connectivity test URL CPC receives HTTP 2XX Success response CPC ignores the HTTP response CPC notifies WIRServer central controller internet connectivity is available WIR saves SSID and password pair for future use
List of Exception Use Cases	
Interfaces	

3.9.2.3 WIR-UC-REQ-378739/A-WIRClient1/WIRClient2 Wi-Fi connected to a Wi-Fi Hotspot connected previously

Actors	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, WIR
Pre-conditions	WIRClient1/WIRClient2 Wi-Fi turned on WIRClient1/WIRClient2 Wi-Fi is scanning available Wi-Fi Hotspot signal
Scenario Description	WIRClient1/WIRClient2 Wi-Fi finds available open Wi-Fi Hotspot and makes a connection to it. WIR provides Wi-Fi Hotspot internet connection without user interaction
Post-conditions	WIR sends WLAN connection request WIRClient1/WIRClient2 Wi-Fi finds an available SSID WIRClient1/WIRClient2 Wi-Fi connects to the Wi-Fi Hotspot



	CPC sends HTTP GET request to connectivity test URL CPC receives 2XX OK response CPC notifies WIR internet connection success WIR gets internet connection through Wi-Fi Hotspot
List of Exception Use Cases	
Interfaces	

3.9.2.4 WIR-UC-REQ-378740/A-WIRClient1/WIRClient2 Wi-Fi connected to an open Wi-Fi Hotspot connected previously and redirected to landing page

Actors	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, customer, WIR
Pre-conditions	WIRClient1/WIRClient2 Wi-Fi turned on WIRClient1/WIRClient2 Wi-Fi is scanning available Wi-Fi Hotspot signal
Scenario Description	WIRClient1/WIRClient2 Wi-Fi finds available open Wi-Fi Hotspot and makes a connection to it. Wi-Fi HotSpot redirect to landing page
Post-conditions	WIRClient1/WIRClient2 Wi-Fi finds available SSID Customer selects a Wi-Fi Hotspot to connect from WIRClient1 UI and enters password WIRClient1 Wi-Fi connects to the Wi-Fi Hotspot CPC waits for CPC_CHECK_INTERVAL seconds CPC sends HTTP GET request to connectivity test URL CPC receives HTTP 3XX redirection to landing page response CPC sends redirected URI and local IP address to WEB engine CPC start a wait timer for CPC_CHECK_INTERVAL seconds WEB engine displays the landing page Customer clicks accept button WEB engine binds local IP address, creates HTTP connection and sends HTTP GET request for home page Wi-Fi Hotspot sends home page and WLAN routes it to WEB engine WEB engine displays the home page The wait timer expires CPC sends HTTP GET request to check captive portal stage is done CPC receives HTTP 2XX Success response CPC notifies WIRServer central controller internet connectivity is available WIR saves SSID and password pair for future use
List of Exception Use Cases	
Interfaces	

3.9.2.5 WIR-UC-REQ-378741/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot first time

Actors	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, customer, WIR
Pre-conditions	WIRClient1/WIRClient2 Wi-Fi turned on WIRClient1/WIRClient2 Wi-Fi is scanning available Wi-Fi Hotspot signal
Scenario Description	Customer enters password and WIRClient1/WIRClient2 Wi-Fi gets an internet connection
Post-conditions	WIRClient1/WIRClient2 Wi-Fi finds available password protected Wi-Fi Hotspot Customer selects the Wi-Fi Hotspot to connect from WIRClient1 UI WIRClient1/WIRClient2 Wi-Fi makes a connection to it. Wi-Fi Hotspot password enter screen is shown on WIRClient1 display Customer enters password WIRClient1/WIRClient2 Wi-Fi finishes authentication procedure Customer opens WEB browser WIR saves SSID and password for future use



	CPC sends HTTP GET request to connectivity test URL CPC receives 2XX OK response CPC notifies WIR internet connection success WIR gets internet connection through Wi-Fi Hotspot
List of Exception Use Cases	
Interfaces	

3.9.2.6 WIR-UC-REQ-378742/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot first time and customer enters verification code received

Actors	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, customer, WIR
Pre-conditions	WIRClient1/WIRClient2 Wi-Fi turned on WIRClient1/WIRClient2 Wi-Fi is scanning available Wi-Fi Hotspot signal
Scenario Description	Customer enters password and WIRClient1/WIRClient2 Wi-Fi gets internet connection
Post-conditions	WIRClient1/WIRClient2 Wi-Fi finds available password protected Wi-Fi Hotspot Customer selects the Wi-Fi Hotspot to connect from WIRClient1 UI WIRClient1/WIRClient2 Wi-Fi makes a connection to it. Wi-Fi Hotspot password enter screen is shown on WIRClient1 display Customer enters password WIRClient1/WIRClient2 Wi-Fi finishes authentication procedure Customer opens WEB browser WIR saves SSID and password for future use Customer enters his Smartphone number on the WEB page Wi-Fi Hotspot provider sends verification code to customer's Smartphone through MT-SMS Customer enters verification code on the WEB page Wi-Fi Hotspot provider verifies password and send home page response CPC sends HTTP GET request to connectivity test URL CPC receives 2XX OK response CPC notifies WIR internet connection success WIR has internet connection through Wi-Fi Hotspot
List of Exception Use Cases	
Interfaces	

3.9.2.7 WIR-UC-REQ-378743/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot first time and landing page displayed

Actors	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, customer, WIR
Pre-conditions	WIRClient1/WIRClient2 Wi-Fi turned on WIRClient1/WIRClient2 Wi-Fi is scanning available Wi-Fi Hotspot signal
Scenario Description	Customer enters password and WIRClient1/WIRClient2 Wi-Fi gets internet connection Landing page displayed after connection
Post-conditions	WIRClient1/WIRClient2 Wi-Fi finds available password protected Wi-Fi Hotspot Customer selects the Wi-Fi Hotspot to connect from WIRClient1 UI WIRClient1/WIRClient2 Wi-Fi makes a connection to it. Wi-Fi Hotspot password enter screen is shown on WIRClient1 display Customer enters password WIRClient1/WIRClient2 Wi-Fi finishes authentication procedure Customer opens WEB browser WIR saves SSID and password for future use CPC sends HTTP GET request for test URL CPC receives 3XX redirection response CPC sends redirected URI and local IP address to WEB engine



	Landing pages displayed on WEB engine Customer accepts T&Cs WEB engine make a HTTP connection with supplied local IP address and sends HTTP GET request for Wi-Fi Hotspot home page WEB engine receives home page response and display it on WIRClient1 screen CPC sends HTTP GET request to connectivity test URL CPC receives 2XX OK response CPC notifies WIR internet connection success WIR has internet connection through Wi-Fi Hotspot
List of Exception Use Cases	
Interfaces	

3.9.2.8 WIR-UC-REQ-378744/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot first time and customer enters password with on-screen

Actors	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, customer, WIR
Pre-conditions	WIRClient1/WIRClient2 Wi-Fi turned on WIRClient1/WIRClient2 Wi-Fi is connecting to Wi-Fi Hotspot and customer is asked to enter password
Scenario Description	Customer enters password with on-screen keyboard
Post-conditions	WIRClient1 UI displays password enter screen Customer enters password WIRClient1/WIRClient2 WLAN starts connection procedure to Wi-Fi Hotspot
List of Exception Use Cases	
Interfaces	

3.9.2.9 WIR-UC-REQ-378745/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot again

Actors	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, customer, WIR
Pre-conditions	WIRClient1/WIRClient2 Wi-Fi turned on WIRClient1/WIRClient2 Wi-Fi is scanning available Wi-Fi Hotspot signal
Scenario Description	WIR uses saved SSID/password for connection
Post-conditions	WIRClient1/WIRClient2 Wi-Fi finds available password protected Wi-Fi Hotspot WIRServer central controller find saved SSID/password and provides it to WIRClient1/WIRClient2 Wi-Fi when request to connect WIRClient1/WIRClient2 Wi-Fi finishes connection procedure CPC sends HTTP GET request to connectivity test URL CPC receives 2XX OK response CPC notifies WIR internet connection success WIR has internet connection through Wi-Fi Hotspot
List of Exception Use Cases	
Interfaces	

3.9.2.10 WIR-UC-REQ-378746/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot again and landing page displayed

Actors	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, customer, WIR
Pre-conditions	WIRClient1/WIRClient2 Wi-Fi turned on WIRClient1/WIRClient2 Wi-Fi is scanning available Wi-Fi Hotspot signal
Scenario Description	WIR uses saved SSID/password for connection



Post-conditions	WIRClient1/WIRClient2 Wi-Fi finds available password protected Wi-Fi Hotspot Customer makes a connection to it through WIRClient1 UI WIR found saved SSID/password and provides it to WIRClient1/WIRClient2 Wi-Fi WIRClient1/WIRClient2 Wi-Fi finishes authentication procedure CPC sends HTTP GET request for connectivity test URL CPC receives 3XX redirection response CPC sends redirected URI and local IP address Landing pages displayed on WEB engine Customer accepts T&Cs WEB engine make a HTTP connection with supplied local IP address and sends HTTP GET request for Wi-Fi Hotspot home page WEB engine receives home page response and display it on screen CPC sends HTTP GET request to connectivity test URL CPC receives 2XX OK response CPC notifies WIR internet connection success WIR has internet connection through Wi-Fi Hotspot
List of Exception Use Cases	
Interfaces	

3.9.2.11 WIR-UC-REQ-378747/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot again but authentication fails

Actors	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, customer, WIR
Pre-conditions	WIRClient1/WIRClient2 Wi-Fi turned on WIRClient1/WIRClient2 Wi-Fi is scanning available Wi-Fi Hotspot signal
Scenario Description	Saved password is not valid and customer enters password
Post-conditions	WIRClient1/WIRClient2 Wi-Fi finds an available password protected Wi-Fi Hotspot WIR found saved SSID/password and provides it to WIRClient1/WIRClient2 Wi-Fi WIRClient1/WIRClient2 Wi-Fi fails authentication procedure due to password changed since last connection Wi-Fi Hotspot password enter screen is shown on WIRClient1 display Customer enters password WIRClient1/WIRClient2 Wi-Fi client finishes authentication procedure CPC sends HTTP GET request to connectivity test URL CPC receives 2XX OK response CPC notifies WIR internet connection success WIR has internet connection through Wi-Fi Hotspot
List of Exception Use Cases	
Interfaces	

3.9.2.12 WIR-UC-REQ-378748/A-Wi-Fi connection lost after Wi-Fi Hotspot timeout

Actors	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, WIR
Pre-conditions	WIRClient1/WIRClient2 Wi-Fi is connected to Wi-Fi Hotspot WIR has internet connection through WIRClient1/WIRClient2 Wi-Fi
Scenario Description	WIRClient1/WIRClient2 Wi-Fi connection lost after timeout
Post-conditions	Wi-Fi Hotspot connection timer expired Wi-Fi Hotspot redirect all HTTP request and ignore all other internet traffic WIR application receives a socket error (ex. destination unreachable) for 5 times consecutively WIR application reports the socket error to WIR WIR sends network interface down notification to all WIR app which is using the WLAN



List of Exception Use Cases	
Interfaces	

3.9.2.13 WIR-UC-REQ-378749/A-Wi-Fi connection lost due to Wi-Fi signal becomes too weak

Actors	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, customer, WIR
Pre-conditions	WIRClient1/WIRClient2 Wi-Fi is connected to Wi-Fi Hotspot WIR has internet connection through WIRClient1/WIRClient2 Wi-Fi
Scenario Description	WIRClient1/WIRClient2 Wi-Fi lost connection when signal becomes weak
Post-conditions	Customer drive vehicle Wi-Fi connection lost WIRClient1/WIRClient2 WLAN notifies WIR connection lost event WIR detects Wi-Fi connection lost WIR notifies WIR app network interface down event
List of Exception Use Cases	
Interfaces	

3.9.2.14 WIR-UC-REQ-378750/A-Wi-Fi connection lost during connection setup due to Wi-Fi signal becomes weak

Actors	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, customer, WIR
Pre-conditions	WIRClient1/WIRClient2 Wi-Fi finds available Wi-Fi Hotspot
Scenario Description	WIRClient1/WIRClient2 Wi-Fi lost connection during connection setup when signal becomes weak
Post-conditions	WIR finds Wi-Fi Hotspot Optionally customer enters password if it is not saved and password protected WIRClient1/WIRClient2 initiates connection procedure to Wi-Fi Hotspot During connection procedure Wi-Fi Hotspot signal becomes too weak and connection lost. WIR notifies WIR app allocation failure
List of Exception Use Cases	
Interfaces	

3.9.2.15 WIR-UC-REQ-378751/A-Customer selects other screen while landing page displayed

Actors	WIRClient1 UI, customer
Pre-conditions	WIRClient1 UI displays landing page
Scenario Description	Customer selects other screen to do something
Post-conditions	Customer selects other screen from WIRClient1 UI Customer continues to use WIRClient1 application while landing page display staying on background WEB engine instance is killed and customer can't access landing page again
List of Exception Use Cases	
Interfaces	

3.9.2.16 WIR-UC-REQ-378752/A-Other screen overrides while landing page displayed

Actors	WIRClient1 UI, CPC, customer, WIR
Pre-conditions	WIRClient1 UI displays landing page



Scenario Description	Other screen override landing page screen
Post-conditions	Other screen (rear view camera or popup) displayed on WIRClient1 screen by some event Other screen finished Landing screen displayed again Customer continues accepting T&C CPC sends HTTP GET request to connectivity test URL CPC receives 2XX OK response CPC notifies WIR internet connection success WIR has internet connection through Wi-Fi Hotspot
List of Exception Use Cases	
Interfaces	

3.9.2.17 WIR-UC-REQ-378753/A-WEB engine supports different WIRClient1 screen size

Actors	WEB engine, CPC
Pre-conditions	CPC requests WEB engine to display WEB page
Scenario Description	WEB engine formats WEB page based on WIRClient1 screen size
Post-conditions	CPC receives redirection WEB page CPC requests WEB engine to display the WEB page with URI and local IP address WEB engine properly displays the WEB page based on WIRClient1 screen size WEB engine handles user input correctly
List of Exception Use Cases	
Interfaces	

3.9.2.18 WIR-UC-REQ-378754/A-CPC get internet connection success after multiple try

Actors	CPC, WIR
Pre-conditions	CPC received HTTP 3XX redirection CPC passed redirected URI and local IP address to WEB engine WEB engine displayed the page on WIRClient1 screen Customer is interacting with the page
Scenario Description	CPC checks internet connectivity and gets success after multiple try
Post-conditions	CPC sends HTTP GET request to connectivity test URL CPC receives 3XX redirection response CPC waits CPC_CHECK_INTERVAL seconds CPC sends HTTP GET request to connectivity test URL CPC receives 3XX redirection response CPC waits CPC_CHECK_INTERVAL seconds CPC sends HTTP GET request to connectivity test URL CPC receives 2XX response CPC notifies WIR internet connection success WIR has internet connection through Wi-Fi Hotspot
List of Exception Use Cases	
Interfaces	

3.9.2.19 WIR-UC-REQ-378755/A-CPC get internet connection failure after max try

Actors	CPC, WIR
Pre-conditions	CPC received HTTP 3XX redirection CPC passed redirected URI and local IP address



	WEB engine displayed the page on WIRClient1 screen Customer is interacting with the page
Scenario Description	CPC checks internet connectivity and gets failure after max try
Post-conditions	Customer ignores landing page OR closes WEB engine CPC sends HTTP GET request to connectivity test URL CPC receives 3XX redirection response CPC waits CPC_CHECK_INTERVAL seconds CPC sends HTTP GET request to connectivity test URL CPC receives 3XX redirection response CPC waits CPC_CHECK_INTERVAL seconds CPC sends HTTP GET request to connectivity test URL CPC receives 3XX redirection response CPC tried all CPC_MAX_CHECK times CPC notifies WIR internet connection failure WIR doesn't have internet connection through Wi-Fi Hotspot WIR notifies WIR app allocation failure
List of Exception Use Cases	
Interfaces	

3.9.2.20 WIR-UC-REQ-378756/A-Password enter screen interrupted by other screen

Actors	WIRClient1 UI, customer
Pre-conditions	Password enter screen is displayed
Scenario Description	Password enter screen is displayed but override by other screen Customer switch password screen and continue entering screen
Post-conditions	Customer is entering password WIRClient1 screen is switched to other screen by some event Customer switches to password screen Customer continues entering password
List of Exception Use Cases	
Interfaces	

3.9.2.21 WIR-UC-REQ-378757/A-WEB engine fails to display landing page

Actors	CPC, WEB engine, diagnostic agent, WIR
Pre-conditions	WEB engine received landing page display request from CPC
Scenario Description	WEB engine fails to display landing page and CPC returns internet connection failure to WIR
Post-conditions	WEB engine fails to display the landing page Diagnostic agent detects the failure and creates a log entry CPC tries CPC_MAX_CHECK but receives 3XX redirection response CPC notifies WIR internet connection failure WIR doesn't have internet connection through Wi-Fi Hotspot WIR notifies WIR app allocation failure
List of Exception Use Cases	
Interfaces	

3.9.2.22 WIR-UC-REQ-378758/A-WEB engine opens 2nd WEB page while landing page is displayed---TBD

Actors	WEB engine, customer
Pre-conditions	WEB engine displayed landing page on WIRClient1 screen
Scenario Description	2 nd WEB engine override landing page screen



Post-conditions	WEB engine receives another WEB page display request WEB engine displays 2 nd WEB page Customer switch to landing page screen Customer continue to interact with the landing page
List of Exception Use Cases	
Interfaces	

3.9.3 White Box View



3.10 WIR-FUN-REQ-379786/A-AV Specific Requirements

3.10.1 Requirements

3.10.1.1 REQ-379793/A-WIRClient1 Monitoring

WIRClient1 shall be capable of continuously monitoring its respective connection status, estimating the current throughput of the connection and reporting the status to WIRServer as part of Wi-Fi management.

3.10.1.2 REQ-379794/A-WIRClient1 Monitoring Throughput Estimation

WIRClient1 shall monitor the connection status of its Wi-Fi and available throughput via Wi-Fi analytics.

3.10.1.3 REQ-379795/A-WIRClient1 Monitoring Throughput Storage

WIRClient1 shall check Wi-Fi connection status and calculate throughput values every WIFI_REPORTING_FREQUENCY seconds and send them to WIRServer, where the values will be stored and maintained in the throughput table.

3.10.1.4 REQ-379796/A-WIRClient1 Monitoring Failure

If there is an error in monitoring the connection status or the throughput analytics, the WIRClient1 shall send an error message to WIRServer that the Wi-Fi monitoring has failed.

3.10.1.5 REQ-379797/A-WIRClient1 Reporting

WIRClient1 shall report connection status and throughput analytics values to WIRServer.

3.10.1.6 REQ-379798/A-WIRClient1 Reporting Storage

The WIRServer shall receive Wi-Fi connection status and throughput analytics values from WIRClient1 and store them internally in the throughput table for use when scheduling interfaces. The WIRServer shall maintain this table when it receives updates on throughput change.

3.10.1.7 REQ-379799/A-WIRClient1 Reporting Frequency

The WIRClient1 Wi-Fi connection status and throughput values shall be updated when there is a connection status change, or a 10% or more calculated throughput change for any interface.

3.10.1.8 REQ-379800/A-WIRClient1 Reporting Failure

The WIRServer shall be notified by WIRClient1 that the throughput estimation has failed via error message. If WIRClient1 attempts to send a throughput update after an error notification, WIRServer shall ignore this message.

3.10.2 Use Cases

3.10.3 White Box View



3.11 WIR-FUN-REQ-378759/A-Performance

3.11.1 Requirements

3.11.1.1 WIR-REQ-378760/A-Boot up

WIR module in WIRServer, WIRClient1, WIRClient2, WIRClient3, WIRClient4, WIRClient5, WIRClient6 should be treated as a high priority system service and shall be booted up and ready before connectivity applications are booted up.

3.11.1.2 WIR-REQ-378761/A-Tunnel setup

WIR module shall make sure tunnels are setup before any connectivity needed applications are booted up so WIR local and central controllers can process application requests as soon as application sends requests

3.11.1.3 WIR-REQ-378762/A-WIRServer and WIRClient interfaces

Any inter ECU interfaces between WIRServer, WIRClient1, WIRClient2, WIRClient3, WIRClient4, WIRClient5 and WIRClient6 shall be treated with high priority and processing shall be instantaneous.

3.11.1.4 WIR-REQ-378763/A-Providing interfaces

If an interface is available WIR shall provide the interface instantly to applications without any time delay

3.11.1.5 WIR-REQ-378764/A-Requests from FCI / WIFI Hotspot and high priority applications

Requests from FCI / WIFI Hotspot and applications with high priority intents shall be treated with the highest priority and shall be serviced immediately.

3.11.2 Use Cases

3.11.3 White Box View



3.12 WIR-FUN-REQ-378765/A-WIRClient1 Configuration

3.12.1 Requirements

3.12.1.1 WIR-REQ-378766/A-Configuration Parameters

The list of config parameters for WIR are identified in WIRServer specification. These parameters shall be updatable from cloud as well as from EOL. WIRClient1 shall have an up to date copy of the EOL parameters and shall use them when WIRServer is not available.

3.12.2 Use Cases

3.12.3 White Box View



4 Appendix: Reference Documents

Reference #	Document Title
1	Logical to Physical Signal Mapping
2	
3	
4	
5	
6	
7	
8	
9	