



**Research & Vehicle Technology**  
**“Infotainment Systems Product Development”**

**Feature – Wireless Interface Router Client1**  
**Infotainment Subsystem Part Specific**  
**Specification (SPSS)**

Version 1.3  
**UNCONTROLLED COPY IF PRINTED**

Version Date: November 22, 2019

**FORD CONFIDENTIAL**



## Revision History

Date	Ver	Notes	
December 5, 2017	1.0	Initial Release	50% of "initial" requirements
February 9, 2018	1.1		
	WIR-FRD-REQ-289527/B-Wireless Interface Router Client1 SPSS	MBORREL4: Updated FRD name to Client1	
	STR-356400/B-Overview	MBORREL4: Removed Applink mentions, added TCU/SYNC to IPPT description	
	STR-487072/B-Architectural Design	MBORREL4: Added REQ-276161, removed REQ-289671, removed REQ-289672-76	
	WIR-CLD-REQ-276162/B-Wireless Interface Router Client1	MBORREL4: Updated responsibilities	
	WIR-CLD-REQ-289670/B-Wireless Interface Router Client2	MBORREL4: Updated responsibilities	
	STR-366290/B-Physical Mapping of Classes	MBORREL4: Removed Client3 from table	
	STR-487073/B-Functional Definition	MBORREL4: Added FUN-REQ-295873, FUN-REQ-296062, FUN-REQ-296082	
	STR-356403/B-Requirements	Added REQ-295687-91	
	WIR-REQ-276054/B-Station Mode Enabling Criteria	MBORREL4: Added new criteria	
	WIR-REQ-276057/B-Station Mode Active When Projection Mode Starts	MBORREL4: Added WLAN clarity	
	WIR-REQ-295687/A-Periodic Scan	MBORREL4: New req.	
	WIR-REQ-295688/A-WIFI Access Point Information Storage	MBORREL4: New req.	
	WIR-REQ-276064/B-Connectivity Check Endpoint	MBORREL4: Updated to capture if no connectivity check endpoint URL is available	
	WIR-REQ-276065/B-Wifi Latency Check	MBORREL4: Updated to include WIR_LATENCY_CHECK DID enabled check	
	WIR-REQ-295689/A-Wi-Fi Low Cost Partner Availability	MBORREL4: New req.	
	WIR-REQ-276067/B-Wifi Low Cost Partner Credentials	MBORREL4: Added EOL	
	WIR-REQ-295690/A-WIRClient1 Publishes WI-FI Status	MBORREL4: New req.	
	WIR-REQ-295691/A-WIRServer Requests WIRClient1 WIFI Disconnect	MBORREL4: New req.	
	WIR-FUN-REQ-276018/B-Local Controller Client	MBORREL4: Updated title	
	STR-366263/B-Requirements	MBORREL4: Added REQ-295692	
	WIR-REQ-276077/B-Local Controller Purpose	MBORREL4: Updated title and content	



WIR-REQ-276078/B-Local Controller Interface With Central Controller	MBORREL4: Updated title, content and diagram
WIR-REQ-276079/B-Receive Intent Requests	MBORREL4: Updated content
WIR-REQ-276080/B-Process Intent Requests	MBORREL4: Updated content
WIR-REQ-276081/B-Return Unique Id	MBORREL4: Updated content
WIR-REQ-276085/B-Responding To Queries From Applications Requesting Connectivity	MBORREL4: Updated content
WIR-REQ-276086/B-Query Response Details	MBORREL4: Updated content
WIR-REQ-276087/B-Scheduling	MBORREL4: Updated content
WIR-REQ-295692/A-Removal From The Scheduling Queue	MBORREL4: New req.
STR-366264/B-Use Cases	MBORREL4: Added RQ-296107-108
WIR-UC-REQ-296107/A-Local controller provides connection interface to requesting application	MBORREL4: New Usecase
WIR-UC-REQ-296108/A-Central controller not available	MBORREL4: New Usecase
STR-366266/B-Requirements	MBORREL4: Added REQ-296059
WIR-REQ-276089/B-Intent Purpose	MBORREL4: Updated content
WIR-REQ-276090/B-Types Of Intents	MBORREL4: Updated content
WIR-REQ-276093/B-Off-Peak And Local Controller	MBORREL4: Updated title and content
WIR-REQ-276094/B-Intent Structure	MBORREL4: Updated table
WIR-REQ-276096/B-Intents And Possible Interfaces	MBORREL4: Updated content
WIR-REQ-296059/A-Bandwidth Shaping	MBORREL4: New req.
WIR-REQ-276116/B-Purpose	MBORREL4: Updated content
WIR-REQ-276117/B-Global Feature ID	MBORREL4: Updated title and content
WIR-REQ-276120/B-Intent Policy Table Structure	MBORREL4: Updated content
WIR-REQ-276121/B-Intent Policy Storage	MBORREL4: Updated content
WIR-REQ-276122/B-Intent Policy Update	MBORREL4: Updated content
WIR-REQ-276123/B-Intent Policy Update Acknowledgement	MBORREL4: Updated content
STR-366288/B-Use Cases	MBORREL4: Added REQ-296113-114
WIR-UC-REQ-296113/A-Cloud sends updated policy table to WIR	MBORREL4: New Usecase
WIR-UC-REQ-296114/A-Policy not updated in WIRServer / WIRClients	MBORREL4: New Usecase



STR-366269/B-Requirements	MBORREL4: Added REQ-295870-71
WIR-REQ-276100/B-Purpose	MBORREL4: Updated content
WIR-REQ-276101/B-Tunnel Support	MBORREL4: Updated title and content
WIR-REQ-276102/B-WIRClient1 IP Address	MBORREL4: Updated content
WIR-REQ-295870/A-Ignition Cycle When Tunnel IP Address Is Already In Use	MBORREL4: New req.
WIR-REQ-295871/A-IP Address Assignment	MBORREL4: New req.
WIR-REQ-276103/B-IP Aliasing	MBORREL4: Updated content
WIR-REQ-276104/B-Allow WIRClient1 Edge Interface Access To Ethernet Connected ECU's	MBORREL4: Updated content
WIR-REQ-276105/B-Allow WIRClient1 Applications To Access Edge Interface Of Other Ethernet Connected ECU's	MBORREL4: Updated content
WIR-REQ-276106/B-WIRClient1 Interfacing With WIRServer Example	MBORREL4: Updated title and content
WIR-REQ-276107/B-Sending A Request From WIRServer	MBORREL4: Updated title and content
WIR-REQ-276108/B-Forwarding The Request To WIRClient1 WIFI Example Proposal	MBORREL4: Updated title and content
WIR-REQ-276109/B-Receiving Response On WIRClient1 Module Example Proposal	MBORREL4: Updated title and content
WIR-REQ-276110/B-Receiving Response On WIRServer Module Example Proposal	MBORREL4: Updated title and content
STR-366284/B-Requirements	MBORREL4: Added REQ-295872
WIR-REQ-276112/B-Purpose	MBORREL4: Updated content
WIR-REQ-276113/B-WIFI Diagnostics	MBORREL4: Updated content
WIR-REQ-276114/B-SDL Diagnostics	MBORREL4: Updated content
WIR-REQ-276159/B-Diagnostics Information Storage	MBORREL4: Updated content
WIR-REQ-276160/B-Diagnostics Information Sending To Cloud	MBORREL4: Updated content
WIR-REQ-295872/A-Sending The App Requests And Interfaces Provided To App To WIRServer Central Controller	MBORREL4: New req.



STR-366285/B-Use Cases	MBORREL4: Added REQ-296112
WIR-UC-REQ-296112/A-WIR provides diagnostics information to cloud based on request from Cloud	MBORREL4: New Usecase
WIR-REQ-289655/B-Data usage calculation	MBORREL4: Updated content
WIR-REQ-289656/B-Data usage calculation scope	MBORREL4: Updated content
WIR-REQ-289657/B-Data usage aggregation per application	MBORREL4: Updated content
WIR-REQ-289658/B-Data usage and edge interface	MBORREL4: Updated content
WIR-REQ-289659/B-Data usage period	MBORREL4: Updated content
WIR-REQ-289660/B-Data usage storage	MBORREL4: Updated content
WIR-REQ-289661/B-Data usage period reset	MBORREL4: Updated content
WIR-REQ-289662/B-Data usage request processing from Ford cloud	MBORREL4: Updated content
WIR-REQ-289663/B-Data usage data transmission to Ford cloud	MBORREL4: Updated content
WIR-REQ-289664/B-Data usage data transmission format	MBORREL4: Updated content
WIR-REQ-289665/B-Data usage request processing from in vehicle HMI	MBORREL4: Updated content
WIR-REQ-289666/B-Data usage data transmission to in vehicle HMI	MBORREL4: Updated content
WIR-REQ-289667/B-Accuracy of data usage calculation	MBORREL4: Updated content
WIR-REQ-289668/B-Data usage timestamp for cellular	MBORREL4: Updated content
WIR-REQ-289669/B-Sending Data usage timestamp to cloud	MBORREL4: Updated content
WIR-REQ-295874/A-WIFI Connect Reminder Popup	MBORREL4: New req.
WIR-REQ-295875/A-Trigger	MBORREL4: New req.
WIR-REQ-295876/A-Trigger Frequency	MBORREL4: New req.
WIR-REQ-295877/A-WIFI Connect Reminder Popup User Action	MBORREL4: New req.
WIR-REQ-295878/A-User Selects OK	MBORREL4: New req.
WIR-REQ-295879/A-User Selects Remind Me Later	MBORREL4: New req.
WIR-REQ-295880/A-WIFI_CONNECT_REMINDER Popup Configurable	MBORREL4: New req.
WIR-REQ-295881/A-User Selects "Do Not Show This Pop-up" Check Box	MBORREL4: New req.



WIR-REQ-295882/A-Master Reset	MBORREL4: New req.
WIR-FUN-REQ-296062/A-Performance	MBORREL4: New Function
STR-504077/A-Requirements	MBORREL4: New STR
WIR-REQ-296063/A-Boot up	MBORREL4: New req.
WIR-REQ-296064/A-Tunnel setup	MBORREL4: New req.
WIR-REQ-296065/A-WIRServer and WIRClient1 and WIRClient2 interfaces	MBORREL4: New req.
WIR-REQ-296066/A-Providing interfaces	MBORREL4: New req.
WIR-REQ-296067/A-Requests from FCI / WIFI Hotspot and high priority applications	MBORREL4: New req.
STR-504078/A-Use Cases	MBORREL4: New STR
STR-504079/A-White Box View	MBORREL4: New STR
WIR-FUN-REQ-296082/A-WIRClient1 Configuration	MBORREL4: New Function
STR-504092/A-Requirements	MBORREL4: New STR
WIR-REQ-296091/A-Configuration Parameters	MBORREL4: New req.
STR-504093/A-Use Cases	MBORREL4: New STR
STR-504094/A-White Box View	MBORREL4: New STR

May 22, 2018

1.2

WIR-CLD-REQ-276162/C-Wireless Interface Router Client1	MBORREL4: Removed access token handling
WIR-CLD-REQ-289670/C-Wireless Interface Router Client2	MBORREL4: Removed access token handling
WIR-CLD-REQ-276161/B-Wireless Interface Router Server	MBORREL4: Removed access token handling
WIR-IIR-REQ-289530/B-WIRClient1_Rx	MBORREL4: Added MD's
MD-REQ-304019/A-AutosarNM	MBORREL4: New MD for autosar wake up signal
MD-REQ-304038/A-VehicleMode	MBORREL4: New MD for keyoff load signal
STR-356403/C-Requirements	MBORREL4: Added REQ-311563-568. Removed REQ-276056-057, REQ-276071
WIR-REQ-311563/A-WI-FI ON/OFF settings	MBORREL4: New req. per team review
WIR-REQ-311564/A-WI-FI Station mode ON/OFF settings	MBORREL4: New req. per team review
WIR-REQ-276053/B-Station Mode Enabling	MBORREL4: Updated content per team review
WIR-REQ-276054/C-Station Mode Enabling Criteria	MBORREL4: Updated content per team review
WIR-REQ-276055/B-Station Mode Disabling Criteria	MBORREL4: Updated content per team review
WIR-REQ-311565/A-WIFI Availability after WIR request	MBORREL4: New req. per team review
WIR-REQ-276058/B-Start Scan	MBORREL4: Updated content per team review



WIR-REQ-276059/B-Scan Results	MBORREL4: Updated content per team review
WIR-REQ-276060/B-Wifi Access Point Information	MBORREL4: Updated content per team review
WIR-REQ-295688/B-WIFI Access Point Information Storage	MBORREL4: Updated content per team review
WIR-REQ-276061/B-Selecting An Access Point	MBORREL4: Updated content per team review
WIR-REQ-276064/C-Connectivity Check Endpoint	MBORREL4: Updated content per team review
WIR-REQ-276067/C-Wifi Low Cost Partner Credentials	MBORREL4: Updated content per team review
WIR-REQ-311566/A-WIFI Access Point Information removal	MBORREL4: New req. per team review
WIR-REQ-311567/A-WIFI Access Point disconnect	MBORREL4: New req. per team review
WIR-REQ-311568/A-WIR HMI	MBORREL4: New req. per team review
WIR-REQ-295690/B-WIRClient1 Publishes WI-FI Status	MBORREL4: Updated content per team review
WIR-REQ-276085/C-Responding To Queries From Applications Requesting Connectivity	MBORREL4: Updated content per team review
WIR-REQ-276090/C-Types Of Intents	MBORREL4: Updated content per team review
WIR-REQ-276094/C-Intent Structure	MBORREL4: Updated content per team review
WIR-REQ-276096/C-Intents And Possible Interfaces	MBORREL4: Updated content per team review
WIR-REQ-296059/B-Bandwidth Shaping	MBORREL4: Updated content per team review
WIR-REQ-276119/B-Intent Policy Table Privilege Breakdown	MBORREL4: Updated table per team review
WIR-REQ-276124/B-Intent Policy Update Notification To Applications	MBORREL4: Updated content per team review
WIR-UC-REQ-296114/B-Policy not updated in WIRServer / WIRClients	MBORREL4: Updated content per team review
WIR-REQ-276112/C-Purpose	MBORREL4: Updated content per team review
WIR-REQ-276113/C-WIFI Diagnostics	MBORREL4: Updated content per team review
WIR-REQ-276159/C-Diagnostics Information Storage	MBORREL4: Updated content per team review
WIR-REQ-276160/C-Diagnostics Information Sending To Cloud	MBORREL4: Updated content per team review
WIR-UC-REQ-296112/B-WIR provides diagnostics information to cloud based on request from Cloud	MBORREL4: Updated content per team review
WIR-REQ-289658/C-Data usage and edge interface	MBORREL4: Updated content per team review
WIR-REQ-296067/B-Requests from FCI / WIFI Hotspot and high priority applications	MBORREL4: Editorial changes
WIR-REQ-296091/B-Configuration Parameters	MBORREL4: Updated content per team review

November 22, 2019

1.3

STR-356400/C-Overview	MBORREL4: Updated text
STR-366262/C-Terminology and Abbreviations	MBORREL4: Updated table





STR-487072/C-Architectural Design	MBORREL4: Added REQ-350862, REQ-370364, REQ-370370
WIR-CLD-REQ-276162/D-Wireless Interface Router Client1	MBORREL4: Updated req.
WIR-CLD-REQ-289670/D-Wireless Interface Router Client2	MBORREL4: Updated req.
WIR-CLD-REQ-350862/B-Wireless Interface Router Client3	MBORREL4: New req.
WIR-CLD-REQ-370364/A-Wireless Interface Router Client4	MBORREL4: New req.
WIR-CLD-REQ-370370/A-Wireless Interface Router Client5	MBORREL4: New req.
WIR-CLD-REQ-276161/C-Wireless Interface Router Server	MBORREL4: Updated req.
STR-366290/D-Physical Mapping of Classes	MBORREL4: Updated table
STR-487073/C-Functional Definition	MBORREL4: Added FUN-REQ-370276
WIR-REQ-276054/D-Station Mode Enabling Criteria	MBORREL4: Updated req.
WIR-REQ-276061/C-Selecting An Access Point	MBORREL4: Updated req.
WIR-REQ-276067/D-Wifi Low Cost Partner Credentials	MBORREL4: Updated req.
WIR-REQ-295690/C-WIRClient1 Publishes WI-FI Status	MBORREL4: Updated req.
WIR-REQ-295691/B-WIRServer Requests WIRClient1 WIFI Disconnect	MBORREL4: Updated req.
WIR-REQ-276127/B-App Link Plug-in Setup	MBORREL4: Corrected spelling
WIR-REQ-276142/B-Reliable Transport	MBORREL4: Updated format (removed table)
WIR-REQ-276077/C-Local Controller Purpose	MBORREL4: Updated req.
WIR-UC-REQ-296107/B-Local controller provides connection interface to requesting application+	MBORREL4: Added WIRClient3
WIR-UC-REQ-296107/C-Local controller provides connection interface to requesting application	MBORREL4: Updated req.
WIR-UC-REQ-296108/C-Central controller not available	MBORREL4: Updated req.
STR-366266/C-Requirements	MBORREL4: Added REQ-370273, REQ-370307
WIR-REQ-276090/D-Types Of Intents	MBORREL4: Updated req.
WIR-REQ-276094/D-Intent Structure	MBORREL4: Updated req.
WIR-REQ-370273/A-Intent processing	MBORREL4: New req.
WIR-REQ-276096/D-Intents And Possible Interfaces	MBORREL4: Updated req.
WIR-REQ-296059/C-Bandwidth Shaping	MBORREL4: Updated req.
WIR-REQ-370307/A-APN3 and APN4	MBORREL4: Updated req.
STR-366287/B-Requirements	MBORREL4: Added REQ-370274, REQ-370514, REQ-370516
WIR-REQ-276116/C-Purpose	MBORREL4: Updated req.
WIR-REQ-276117/C-Global Application ID	MBORREL4: Updated title and req.





WIR-REQ-276119/C-Intent Policy Table Privilege Breakdown	MBORREL4: Updated req.
WIR-REQ-276120/C-Intent Policy Table Structure	MBORREL4: Updated req.
WIR-REQ-276122/C-Intent Policy Update	MBORREL4: Updated req.
WIR-REQ-370274/A-Sends Error for Corrupt policy file	MBORREL4: New req.
WIR-REQ-370514/A-WIRClient3 Policies	MBORREL4: New req.
WIR-REQ-370516/A-WIRClient4 Policies	MBORREL4: New req.
WIR-UC-REQ-296113/B-Cloud sends updated policy table to WIR	MBORREL4: Updated req.
WIR-UC-REQ-296114/C-Policy not updated in WIRServer / WIRClients	MBORREL4: Updated req.
WIR-REQ-276100/C-Purpose	MBORREL4: Updated req.
WIR-REQ-276105/C-Allow WIRClient1 Applications To Access Edge Interface Of Other Ethernet Connected ECU's	MBORREL4: Updated req.
STR-366284/C-Requirements	MBORREL4: Added REQ-370308
WIR-REQ-276159/D-Diagnostics Information Storage	MBORREL4: Updated req.
WIR-REQ-370308/A-On demand diagnostic request	MBORREL4: New req.
WIR-UC-REQ-296112/D-WIR provides diagnostics information to cloud based on request from Cloud	MBORREL4: Updated req.
STR-488345/B-Requirements	MBORREL4: Removed REQ-289669. Added REQ-370326
WIR-REQ-289655/C-Data usage calculation	MBORREL4: Updated req.
WIR-REQ-289656/C-Data usage calculation scope	MBORREL4: Updated req.
WIR-REQ-289657/C-Data usage aggregation per application	MBORREL4: Updated req.
WIR-REQ-289658/D-Data usage and edge interface	MBORREL4: Updated req.
WIR-REQ-370326/A-SiriusXM data usage report	MBORREL4: New req.
WIR-REQ-295875/B-Trigger	MBORREL4: Updated req.
WIR-REQ-295877/B-WIFI Connect Reminder Popup User Action	MBORREL4: Updated req.
WIR-REQ-295878/B-User Selects OK	MBORREL4: Updated req.
WIR-REQ-295879/B-User Selects Remind Me Later	MBORREL4: Updated req.
WIR-REQ-295881/B-User Selects "No" Check Box	MBORREL4: Updated req.
WIR-REQ-295882/B-Master Reset	MBORREL4: Updated req.
WIR-FUN-REQ-370276/A-Captive Portal Check	MBORREL4: New function
STR-706472/A-Requirements	MBORREL4: New STR
WIR-REQ-370277/A-Captive portal check support	MBORREL4: New req.
WIR-REQ-370278/A-WLAN connection request	MBORREL4: New req.
WIR-REQ-370279/A-WLAN client	MBORREL4: New req.



WIR-REQ-370280/A-WEB engine	MBORREL4: New req.
WIR-REQ-370281/A-WIRClient1 UI screen switch	MBORREL4: New req.
WIR-REQ-370282/A-WiFi password entering by customer	MBORREL4: New req.
WIR-REQ-370283/A-WIRClient1 UI restriction	MBORREL4: New req.
WIR-REQ-370284/A-CPC check procedure	MBORREL4: New req.
STR-706473/A-Use Cases	MBORREL4: New STR
WIR-UC-REQ-370285/A-WIRClient1 Wi-Fi connected to a Wi-Fi Hotspot first time and redirected to landing page	MBORREL4: New req.
WIR-UC-REQ-370286/A-WIRClient1 Wi-Fi connected to a Wi-Fi Hotspot first time and Wi-Fi Hotspot doesn't have landing page	MBORREL4: New req.
WIR-UC-REQ-370287/A-WIRClient1/WIRClient2 Wi-Fi connected to a Wi-Fi Hotspot connected previously	MBORREL4: New req.
WIR-UC-REQ-370288/A-WIRClient1/WIRClient2 Wi-Fi connected to an open Wi-Fi Hotspot connected previously and redirected to landing page	MBORREL4: New req.
WIR-UC-REQ-370289/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot first time	MBORREL4: New req.
WIR-UC-REQ-370290/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot first time and customer enters verification code recei	MBORREL4: New req.
WIR-UC-REQ-370291/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot first time and lading page displayed	MBORREL4: New req.
WIR-UC-REQ-370292/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot first time and customer enters password with on-screen	MBORREL4: New req.
WIR-UC-REQ-370293/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot again	MBORREL4: New req.
WIR-UC-REQ-370294/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot again and landing page displayed	MBORREL4: New req.
WIR-UC-REQ-370295/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot again but authentication fails	MBORREL4: New req.
WIR-UC-REQ-370296/A-Wi-Fi connection lost after Wi-Fi Hotspot timeout	MBORREL4: New req.



WIR-UC-REQ-370297/A-Wi-Fi connection lost due to Wi-Fi signal becomes too weak	MBORREL4: New req.
WIR-UC-REQ-370298/A-Wi-Fi connection lost during connection setup due to Wi-Fi signal becomes weak	MBORREL4: New req.
WIR-UC-REQ-370299/A-Customer selects other screen while landing page displayed	MBORREL4: New req.
WIR-UC-REQ-370300/A-Other screen overrides while landing page displayed	MBORREL4: New req.
WIR-UC-REQ-370301/A-WEB engine supports different WIRClient1 screen size	MBORREL4: New req.
WIR-UC-REQ-370302/A-CPC get internet connection success after multiple try	MBORREL4: New req.
WIR-UC-REQ-370303/A-CPC get internet connection failure after max try	MBORREL4: New req.
WIR-UC-REQ-370304/A-Password enter screen interrupted by other screen	MBORREL4: New req.
WIR-UC-REQ-370305/A-WEB engine fails to display landing page	MBORREL4: New req.
WIR-UC-REQ-370306/A-WEB engine opens 2nd WEB page while landing page is displayed---TBD	MBORREL4: New req.
STR-706474/A-White Box View	MBORREL4: New STR
WIR-REQ-296063/B-Boot up	MBORREL4: Updated req.
WIR-REQ-296065/B-WIRServer and WIRClient interfaces	MBORREL4: Updated title and req.



# Table of Contents

REVISION HISTORY .....	2
<b>1 OVERVIEW .....</b>	<b>14</b>
1.1 Terminology and Abbreviations .....	14
<b>2 ARCHITECTURAL DESIGN.....</b>	<b>16</b>
2.1 WIR-CLD-REQ-276162/D-Wireless Interface Router Client1 .....	16
2.2 WIR-CLD-REQ-289670/D-Wireless Interface Router Client2 .....	16
2.3 WIR-CLD-REQ-350862/B-Wireless Interface Router Client3 .....	16
2.4 WIR-CLD-REQ-370364/A-Wireless Interface Router Client4 .....	16
2.5 WIR-CLD-REQ-370370/A-Wireless Interface Router Client5 .....	17
2.6 WIR-CLD-REQ-276161/C-Wireless Interface Router Server .....	17
2.7 Physical Mapping of Classes .....	17
2.8 WIRClient1 Interface .....	17
2.8.1 WIR-IIR-REQ-289529/A-WIRClient1_Tx .....	17
2.8.2 WIR-IIR-REQ-289530/B-WIRClient1_Rx .....	17
<b>3 FUNCTIONAL DEFINITION .....</b>	<b>21</b>
3.1 WIR-FUN-REQ-274350/A-Wifi Handler .....	21
3.1.1 Requirements .....	21
3.1.2 Use Cases .....	25
3.1.3 White Box View .....	25
3.2 WIR-FUN-REQ-274352/A-AppLink Handler .....	26
3.2.1 Requirements .....	26
3.2.2 Use Cases .....	32
3.2.3 White Box View .....	32
3.3 WIR-FUN-REQ-276018/B-Local Controller Client .....	33
3.3.1 Requirements .....	33
3.3.2 Use Cases .....	35
3.3.3 White Box View .....	36
3.4 WIR-FUN-REQ-276020/A-Intents .....	37
3.4.1 Requirements .....	37
3.4.2 Use Cases .....	40
3.4.3 White Box View .....	40
3.5 WIR-FUN-REQ-276032/A-Policy Manager .....	41
3.5.1 Requirements .....	41
3.5.2 Use Cases .....	42
3.5.3 White Box View .....	43
3.6 WIR-FUN-REQ-276022/A-Tunnel Manager.....	44
3.6.1 Requirements .....	44
3.6.2 Use Cases .....	48
3.6.3 White Box View .....	48
3.7 WIR-FUN-REQ-276030/A-Diagnostics .....	49
3.7.1 Requirements .....	49
3.7.2 Use Cases .....	50
3.7.3 White Box View .....	50
3.8 WIR-FUN-REQ-289654/A-Data Usage.....	51



3.8.1	Requirements .....	51
3.8.2	Use Cases .....	52
3.8.3	White Box View .....	52
3.9	<i>WIR-FUN-REQ-295873/A-WIFI Connect Reminders - HMI</i> .....	53
3.9.1	Requirements .....	53
3.9.2	Use Cases .....	54
3.9.3	White Box View .....	54
3.10	<i>WIR-FUN-REQ-370276/A-Captive Portal Check</i> .....	55
3.10.1	Requirements .....	55
3.10.2	Use Cases .....	56
3.10.3	White Box View .....	65
3.11	<i>WIR-FUN-REQ-296062/A-Performance</i> .....	66
3.11.1	Requirements .....	66
3.11.2	Use Cases .....	66
3.11.3	White Box View .....	66
3.12	<i>WIR-FUN-REQ-296082/A-WIRClient1 Configuration</i> .....	67
3.12.1	Requirements .....	67
3.12.2	Use Cases .....	67
3.12.3	White Box View .....	67
4	<b>APPENDIX: REFERENCE DOCUMENTS</b> .....	68



# 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, ADSIM, 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, 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, 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
CAN	Controller Area Network
CCS	Customer Connectivity Settings
CHR	Connected HMI Radio
CPC	Captive Portal Check
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
SDN	Software Delivery Network
SDS	Self Driving System
SoA	Service Oriented Architecture
TCU	Telematics Control Unit



Term	Description
ADSIM	Automated Driving System Interface Module
TCU-B	Telematics Control Unit B
WIR	Wireless Interface Router
WLAN	Wireless Local Area Network





## 2 Architectural Design

### 2.1 WIR-CLD-REQ-276162/D-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
- 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/D-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
- Responsible for setting up Cellular APN1 and APN2. Also APN3 and APN4 for China
- 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/B-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
- 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 WIRClient3 class.

### 2.4 WIR-CLD-REQ-370364/A-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
- 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/A-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
- 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-276161/C-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
- 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.

## 2.7 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 an example only (FNV2, Z2) 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

## 2.8 WIRClient1 Interface

### 2.8.1 WIR-IIR-REQ-289529/A-WIRClient1\_Tx

### 2.8.2 WIR-IIR-REQ-289530/B-WIRClient1\_Rx

#### 2.8.2.1 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	

**2.8.2.2 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	

**2.8.2.3 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	

**2.8.2.4 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	

**2.8.2.5 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	

**2.8.2.6 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	

**2.8.2.7 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	



### 3 Functional Definition

#### 3.1 WIR-FUN-REQ-274350/A-Wifi Handler

##### 3.1.1 Requirements

###### 3.1.1.1 WIR-REQ-311563/A-WI-FI ON/OFF settings

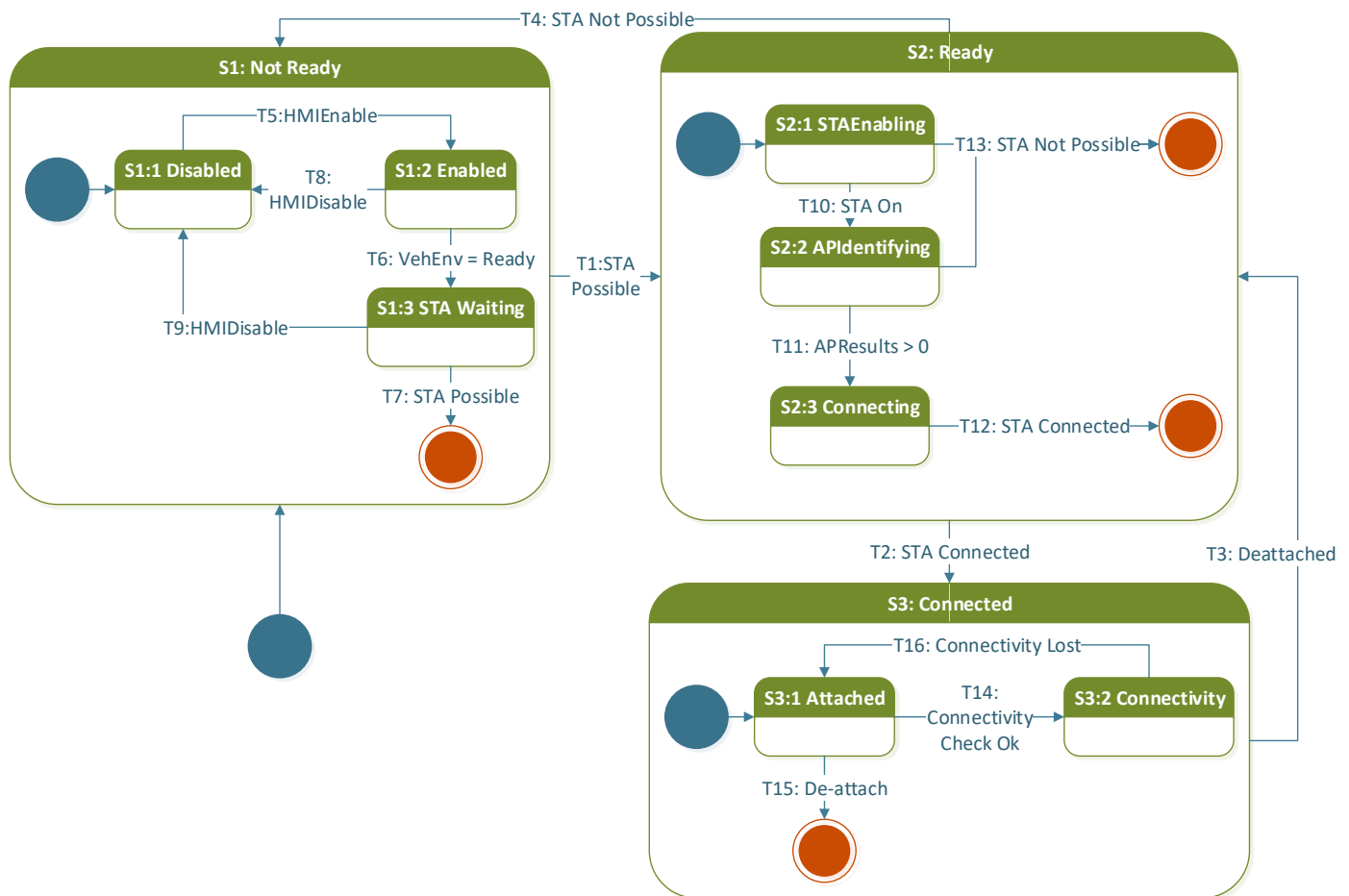
WIRClient1 shall not request WLAN services for any WIFI operation if WI-FI is OFF.

###### 3.1.1.2 WIR-REQ-311564/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-276052/A-State machine

SC: WIR STA Connections



###### 3.1.1.4 WIR-REQ-276053/B-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-276054/D-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-276055/B-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-311565/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-276058/B-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-276059/B-Scan Results

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

### 3.1.1.10 WIR-REQ-295687/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-276060/B-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-295688/B-WIFI Access Point Information Storage

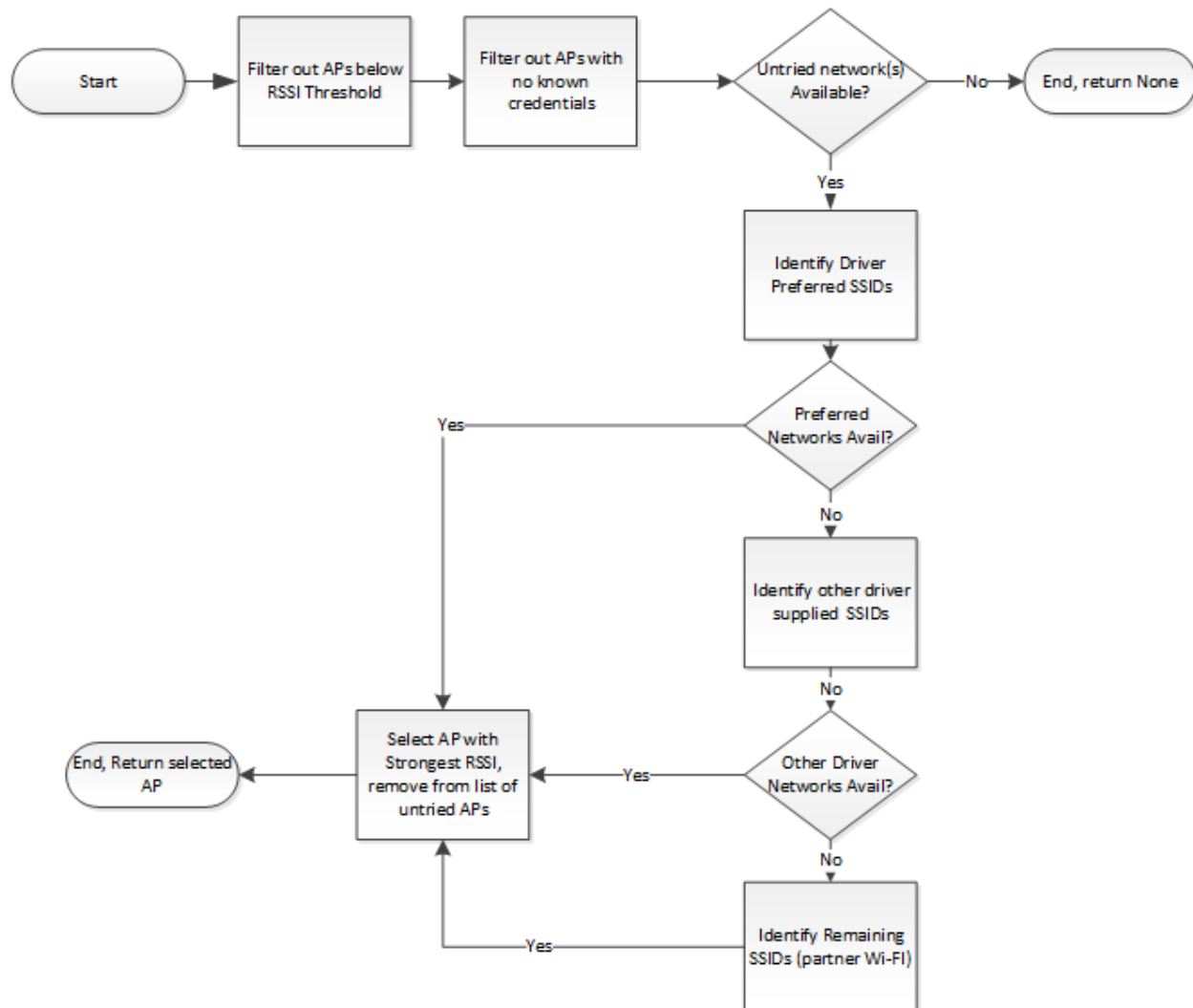
WIRClient1 shall be aware of the Wi-Fi access points known by the vehicle, as entered by the driver or supplied by Ford in case Ford chooses a WIFI provider and shall store the same in non-volatile memory for quick retrieval.

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

### 3.1.1.13 WIR-REQ-276061/C-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.



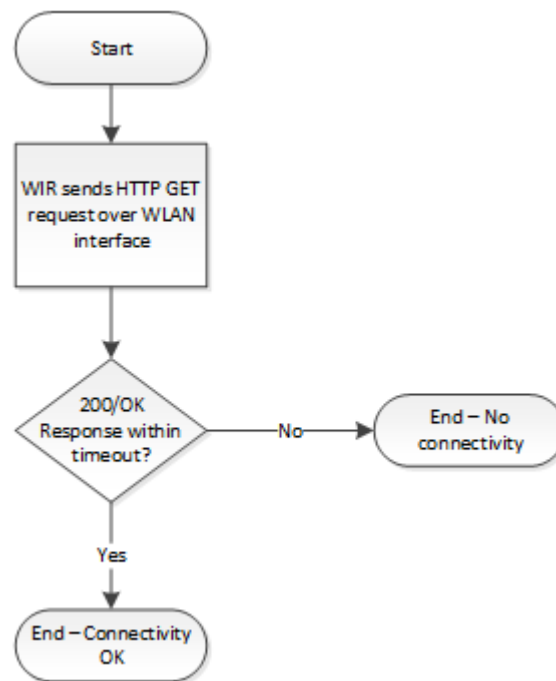


#### 3.1.1.14 WIR-REQ-276062/A-Performing A Connectivity Check

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-276063/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-276064/C-Connectivity Check Endpoint

Connectivity check end point 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-276065/B-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-276066/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-295689/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-276067/D-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-276068/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-311566/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-311567/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-311568/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-276069/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-276070/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-295690/C-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-276072/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-295691/B-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**

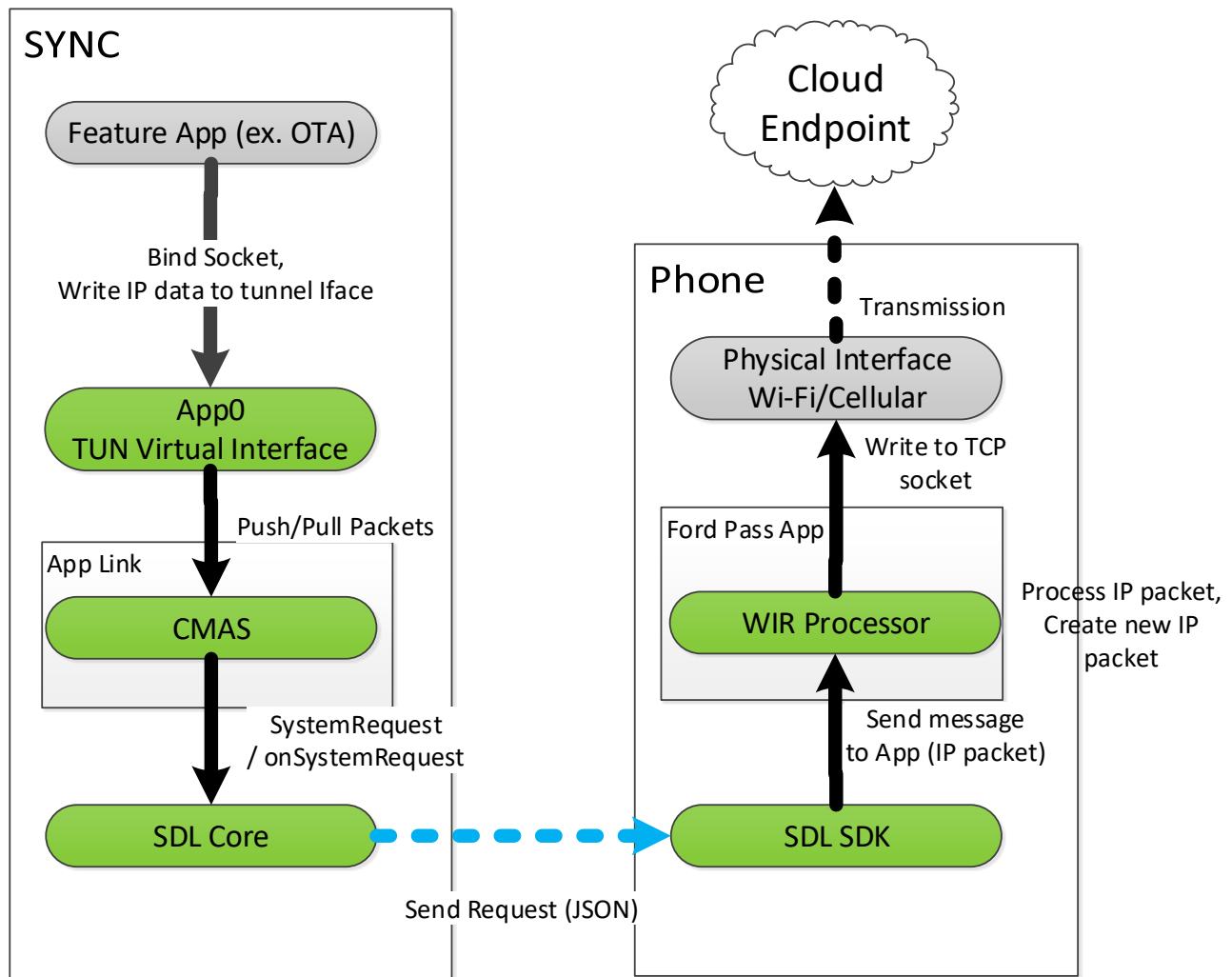
## **3.1.3 White Box View**

## 3.2 WIR-FUN-REQ-274352/A-AppLink Handler

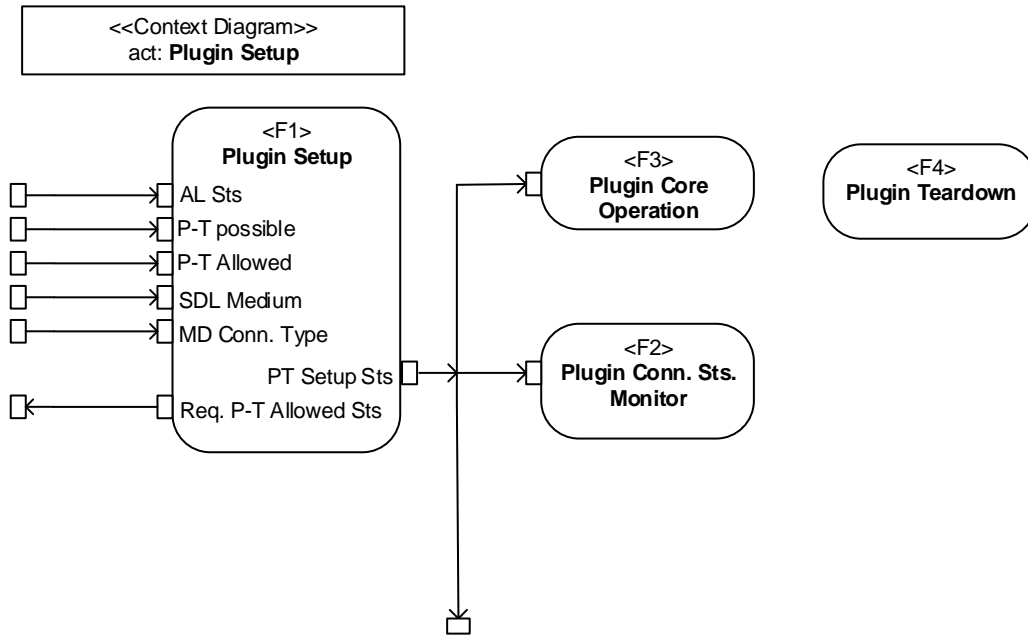
### 3.2.1 Requirements

#### 3.2.1.1 WIR-REQ-276126/A-App Link General Flow

#### App Link Pass-Through Function Group Context



#### 3.2.1.2 WIR-REQ-276127/B-App Link Plug-in Setup

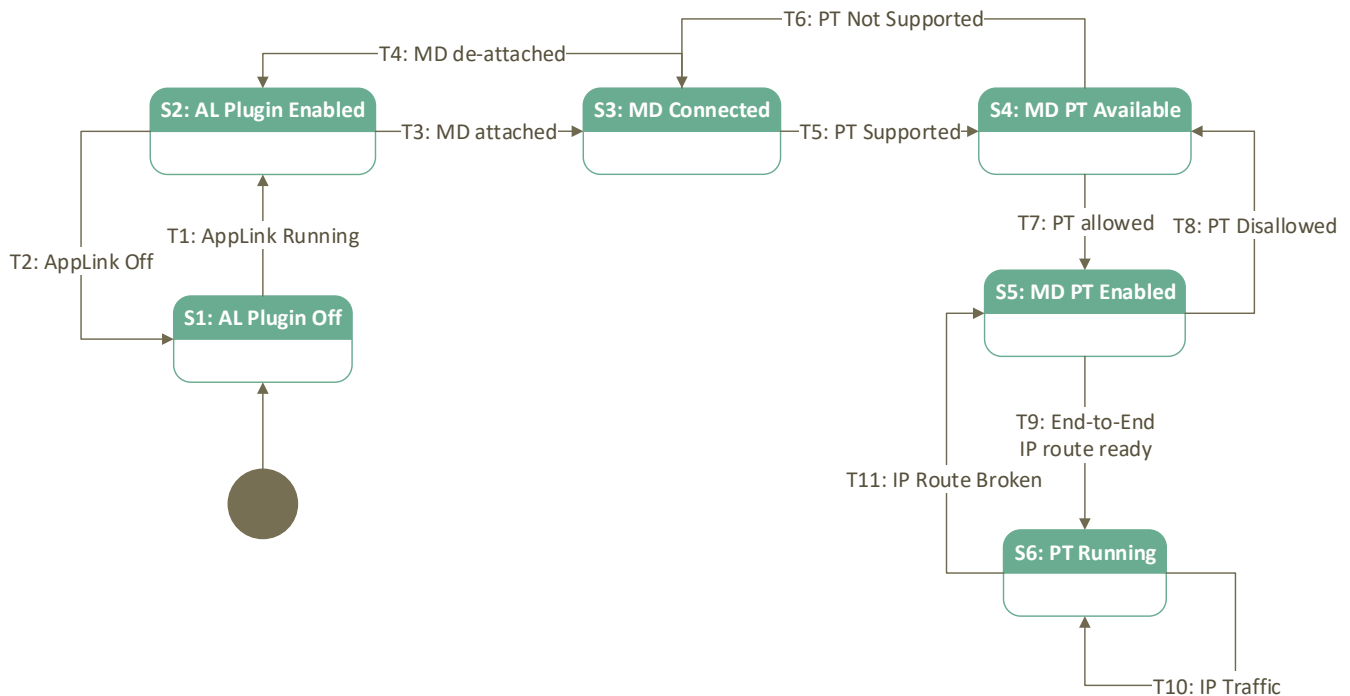


The setup function encompasses:

- 1) Setting up the unix TUN interface
- 2) Enabling the App Link plugin that handles passing IP packets to and from the TUN interface.
- 3) Verifying pass-through support on paired mobile devices.
- 4) Enabling and verifying the end to end connectivity of the pass-through functionality
- 5) Signaling to the WIRClient1 CM that the pass through interface is ready.

### 3.2.1.3 WIR-REQ-276128/A-Function Modeling

SC: WIR SYNC App-Link Pass-through  
Plugin





### 3.2.1.4 WIR-REQ-276129/A-Plugin Initialization

The App Link Pass Through Plugin shall be initialized and responsive to signals any time the WIRClient1 App Link service is active. The plugin shall respond to all app link service messages with no more than 100ms of delay.

### 3.2.1.5 WIR-REQ-276130/A-Request For Pass Through Permission

If the App Link Pass-Through Plugin receives the Pass-Through Possible Signal set to TRUE, the plugin shall request the Pass-Through Allowed status from the mobile device.

### 3.2.1.6 WIR-REQ-276131/A-TUN Interface Initialization

The Pass Through Plugin shall initialize, enable, and assign an IP address to a UNIX TUN interface only when all of the following criteria are met.

1. The Pass-Through Possible signal is TRUE.
2. The Pass-Through Allowed signal is of type "Allowed".
3. The SDL medium, as signaled, is permitted by the "Allowed SDL Mediums" parameter.
4. The paired mobile device's cloud connectivity, as signaled, is allowed by the "Allowed Mobile Connectivity" parameter.

### 3.2.1.7 WIR-REQ-276132/A-TUN Initialization And Time To Ready

The TUN interface shall be configured, the plugin read to receive IP data, and the "Pass-Through Setup Status" signal set to "Ready" in no more than 1 second after the requirements of following the requirements of R\_FNC\_App Link Pass-Through\_00002 are met.

### 3.2.1.8 WIR-REQ-276133/A-IP Handler Server

Within 1 second of establishing the TUN interface describe in "R\_FNC\_App Link Pass-Through\_00006", the plugin shall enable an IP handler server to process IP packets as described by the requirements for the "Plugin Core Operation" function.

### 3.2.1.9 WIR-REQ-276134/A-Plugin Setup Status Control

The Plugin Setup Status signal shall be set to TRUE only when the plugin IP handler server AND the TUN interface is enabled and assigned an IP address.

### 3.2.1.10 WIR-REQ-276135/A-TUN Interface IP Address And Enumeration

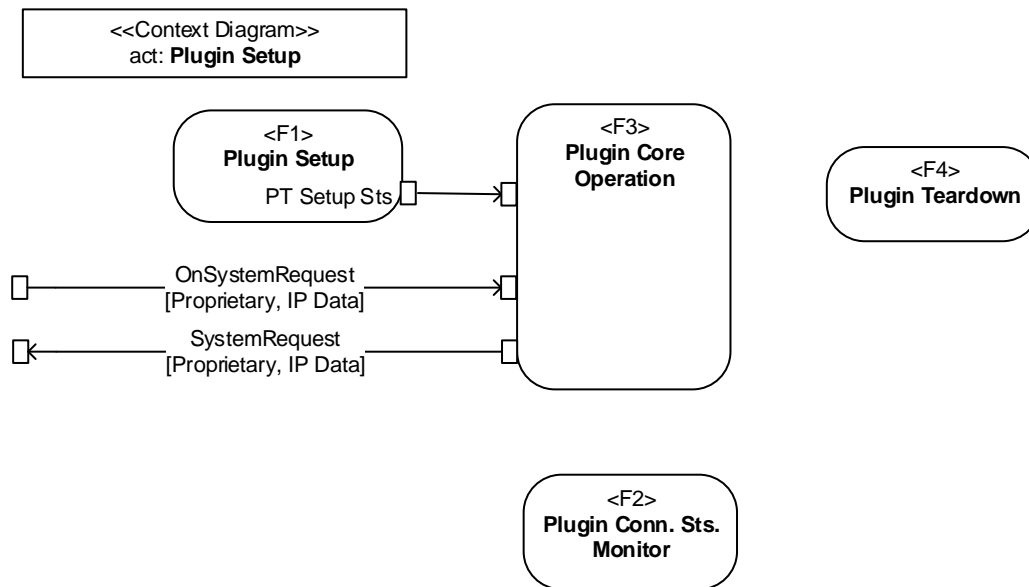
When assigning an IP address to the plugin TUN interface, the Pass-Through Plugin shall assign the IPv4 IP address designated by the "Pass-Through IP address" parameter and assign the enumeration designated by the "Pass-Through Interface Name" parameter.

### **3.2.1.11 Error Handling**

#### 3.2.1.11.1 WIR-REQ-276137/A-Permission Retry Strategy

If the attached mobile device does not respond to the Pass Through Possible Status Request issued by the Plugin, the Plugin shall retry the request with the period defined by the "Plugin Setup Retry Time Interval" for up to the number of retries defined in the "Plugin Setup Retry Limit."

App Link Pass-Through Plugin Core Operation



The core operation function encompasses:

- 1) Sending IP packets sent into the TUN interfaces by applications to the App Link service
- 2) Sending IP packets received from the App Link Service to the TUN interface.

#### 3.2.1.11.2 WIR-REQ-276138/A-Handling Incoming TUN IP Packets

The IP Handler Server shall retrieve IP packets from the TUN interface and publish them to the mobile device via App Link APIs.

The IP Handler server shall publish IP packets to the mobile device using the SystemRequest RPC, with the type set to "Proprietary" and the subtype set to "IP data".

The IP Handler shall retrieve a packet from the TUN Interface and publish it to the mobile device in no more than 10ms.

#### 3.2.1.11.3 WIR-REQ-276139/A-Handling Return IP Packets

The IP Handler Server shall receive OnSystemRequest RPC messages with type "Proprietary" and subtype set to "IP Data" from the App Link Service.

Upon receipt of the IP data OnSystemRequest, the IP handler shall extract the payload, extract the payload of the IP packet, and publish the packet back to the TUN interface.

The IP handler shall receive the packet from the OnSystemRequest RPC and publish the IP payload back to the TUN interface in no more than 10ms.

#### 3.2.1.11.4 WIR-REQ-276140/A-TCP-UDP Support

The IP Handler Server shall support all TCP and UDP IP Packet payloads. The IP Handler Server shall be agnostic to any protocols built on top of TCP or UDP, unless otherwise specified.

#### 3.2.1.11.5 WIR-REQ-276141/A-DNS Support

The IP Handler Server shall support the pass through of DNS queries originated by vehicle applications to the mobile device.

#### 3.2.1.11.6 WIR-REQ-276142/B-Reliable Transport

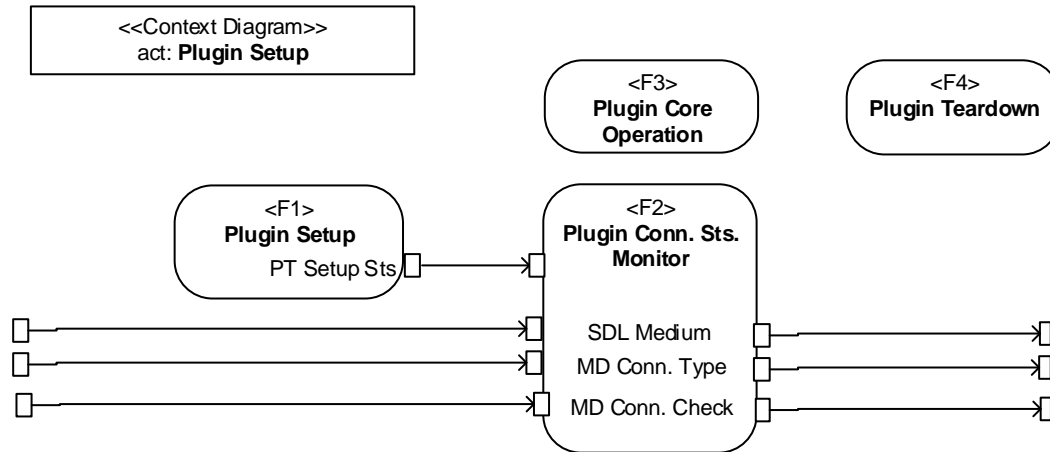
The IP handler server shall maintain, and not violate, the reliable transmission of protocols inside the IP payload sent and received through the past through plugin. In effect, IP handler server shall preserve TCP reliability.





### 3.2.1.12 Pass Through Plugin Connection Status Monitor

#### 3.2.1.12.1 WIR-REQ-276144/A-Function Scope



The Connection Status monitor encompasses:

- 1) Informing the WIR CM Network Quality Manager about the medium, connection, and connectivity being used by the pass through plugin.

#### 3.2.1.12.2 WIR-REQ-276145/A-SDL Medium

An input signal to the App Link Pass-Through Plugin, indicating the medium used to connect the mobile phone to app link.

Data Type	Init Value	Default Value (missing signal)
Enum {unknown:0, BT:1, USB:2, Wi-Fi:3, Reserved:4-7}	0	0

#### 3.2.1.12.3 WIR-REQ-276146/A-Mobile Device Connectivity

Input signal to the App Link Pass-Through Plugin, indicating the connectivity status of the mobile device

Data Type	Init Value	Default Value (missing signal)
Enum { unknown: 0, Wi-Fi: 1, Cellular: 2, Other:3, reserved:4-7}	0	0

#### 3.2.1.12.4 WIR-REQ-276147/A-Mobile Device Connection Check

Input signal to the Pass-Through plugin, generated by the mobile device, indicating if the current active network interface has working end to end connectivity. This generator of this signal relies on the mobile device OS connection manager, i.e. the Android Connection Manager or Apple's Connection Manager

Data Type	Init Value	Default Value (missing signal)
Enum{unknown:0, NotConnected:1, Connected:2}	0	0

#### 3.2.1.12.5 WIR-REQ-276148/A-Plugin Connection Monitor Operation

The Pass-Through Plugin Connection Monitor shall operate whenever the Plugin Setup signal is set to "Ready".



### 3.2.1.12.6 WIR-REQ-276149/A-SDL Medium Monitor

The Plugin Connection Status Monitor shall track and report the underlying connectivity method between SDL and the mobile device to the WIR Connection Manager in real time.

The underlying connectivity medium refers to which method is used to pair the mobile device to WIRClient1: BT, USB, Wi-Fi and any future method.

To track the underlying connectivity medium the plugin connection status monitor shall monitor the “SDL Medium” signal.

### 3.2.1.12.7 WIR-REQ-276150/A-Mobile Device Connectivity Monitor

The Plugin Connection Status Monitor shall track and report the “Mobile Device Connectivity” signal to the WIR Connection Manager in Real time.

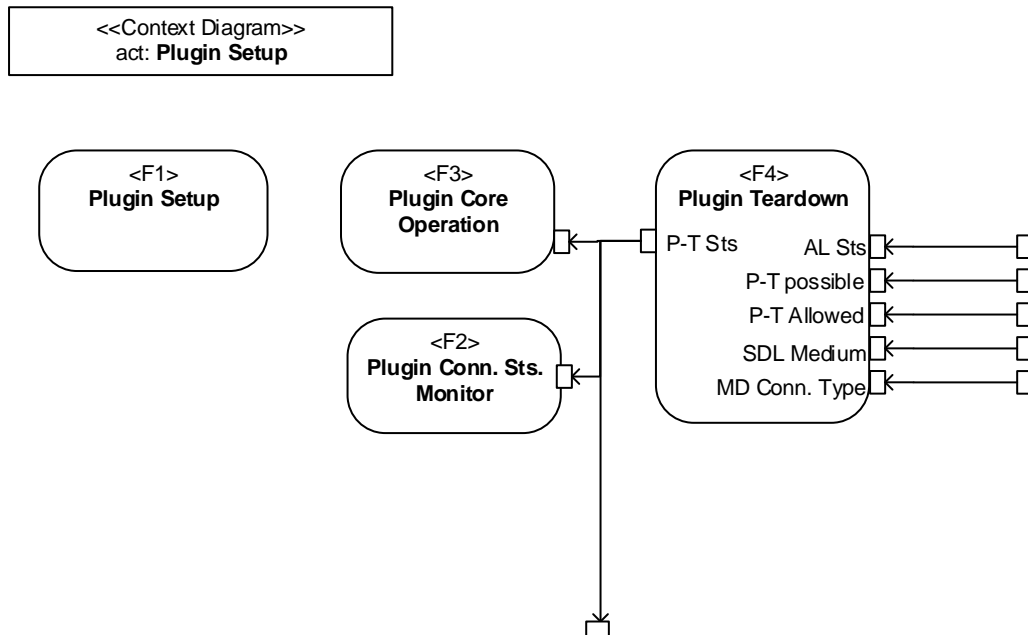
This signal shall report the mobile device’s cloud connection type, typically Wi-Fi or Cellular.

### 3.2.1.12.8 WIR-REQ-276151/A-Mobile Device Connectivity Check

The Plugin Connection Status monitor shall track a mobile device signal, “Mobile Device Connection Check” signal and report this status to the Connection Manager in real time.

## 3.2.1.13 Pass Through Plugin Teardown

### 3.2.1.13.1 WIR-REQ-276153/A-Function Scope



The teardown encompasses:

- 1) Stopping the IP handler Service and connection monitor if the pass through functionality must be turned off.

### 3.2.1.13.2 WIR-REQ-276154/A-Shutdown Of The IP Handler Server And TUN Interface

If the IP Handler Server is running and the pass through plugin status signal transitions from ready to not ready,, the IP Handler Server shall be stopped and the TUN interface torn down in no more than 500ms.

### 3.2.1.13.3 WIR-REQ-276155/A-Shutdown Of The Connection Monitor

If the pass-through connection monitor is running, and the pass-through plugin status signal transitions from “ready” to “not ready”, the connection monitor shall be stopped in no more than 500ms.

### 3.2.1.13.4 WIR-REQ-276156/A-Pass Through Interface Status Not Ready

If one or more of the following conditions are met, the plugin shall set the Pass Through Plugin Status signal to “not ready”.

1. The App Link Service Status signal is “Not Running”.
2. The AL Pass-Through Possible signal is false.



3. The AL Pass-Through Allowed status is false.
4. The SDL Medium is not allowed by "Allowed SDL Mediums" parameter.
5. The Mobile Connectivity is not allowed by the "Allowed Mobile Connectivity" parameter.
6. The App Link Passthrough Enablement Parameter is changed to Disabled.

The core operation function encompasses:

- 1) Sending IP packets sent into the TUN interfaces by applications to the App Link service
- 2) Sending IP packets received from the App Link Service to the TUN interface.

### 3.2.2 Use Cases

### 3.2.3 White Box View



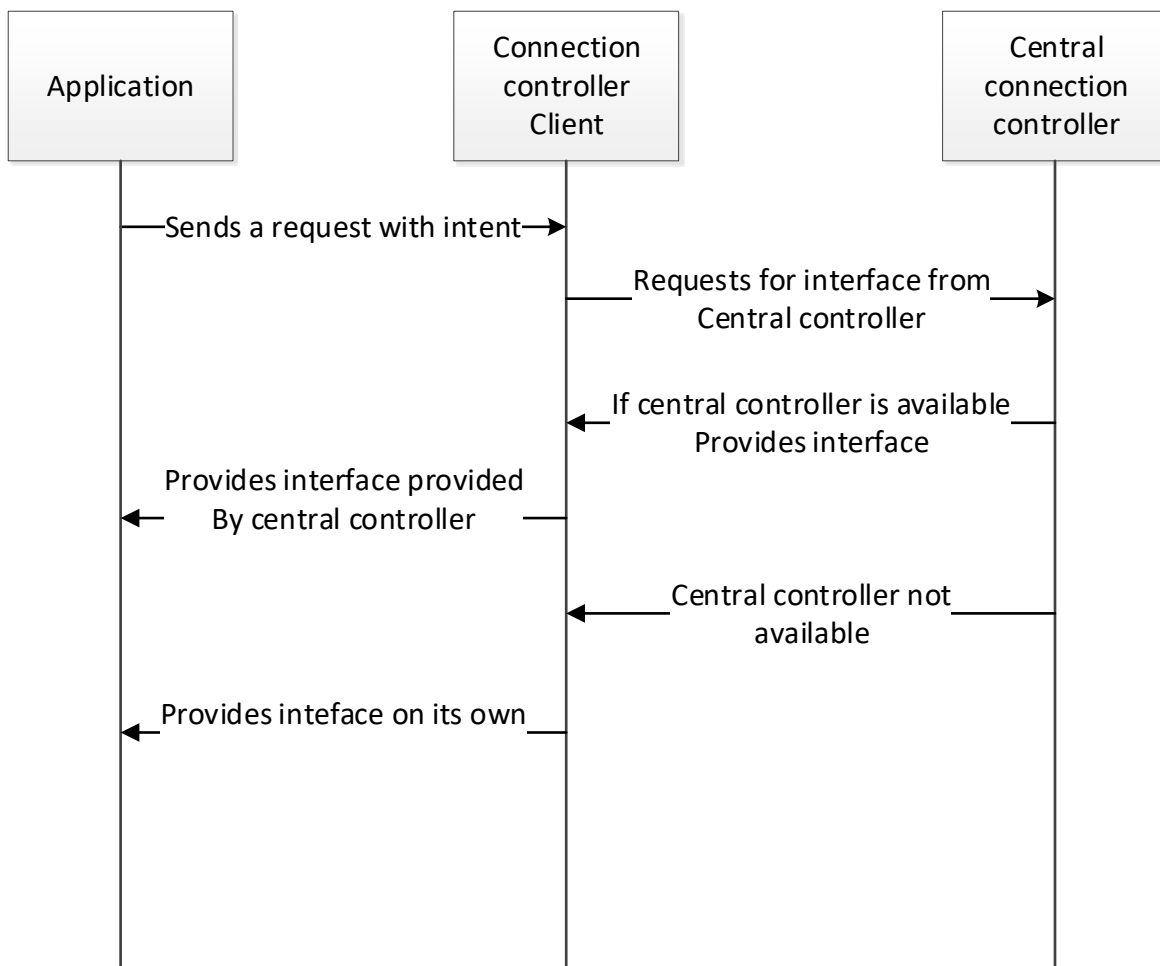
### 3.3 WIR-FUN-REQ-276018/B-Local Controller Client

#### 3.3.1 Requirements

##### 3.3.1.1 WIR-REQ-276077/C-Local Controller Purpose

The Local controller resides in WIRClient1, WIRClient2, WIRClient3, WIRClient4, WIRClient5, 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.

##### 3.3.1.2 WIR-REQ-276078/B-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.

##### 3.3.1.3 WIR-REQ-276079/B-Receive Intent Requests

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

### 3.3.1.4 WIR-REQ-276080/B-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.

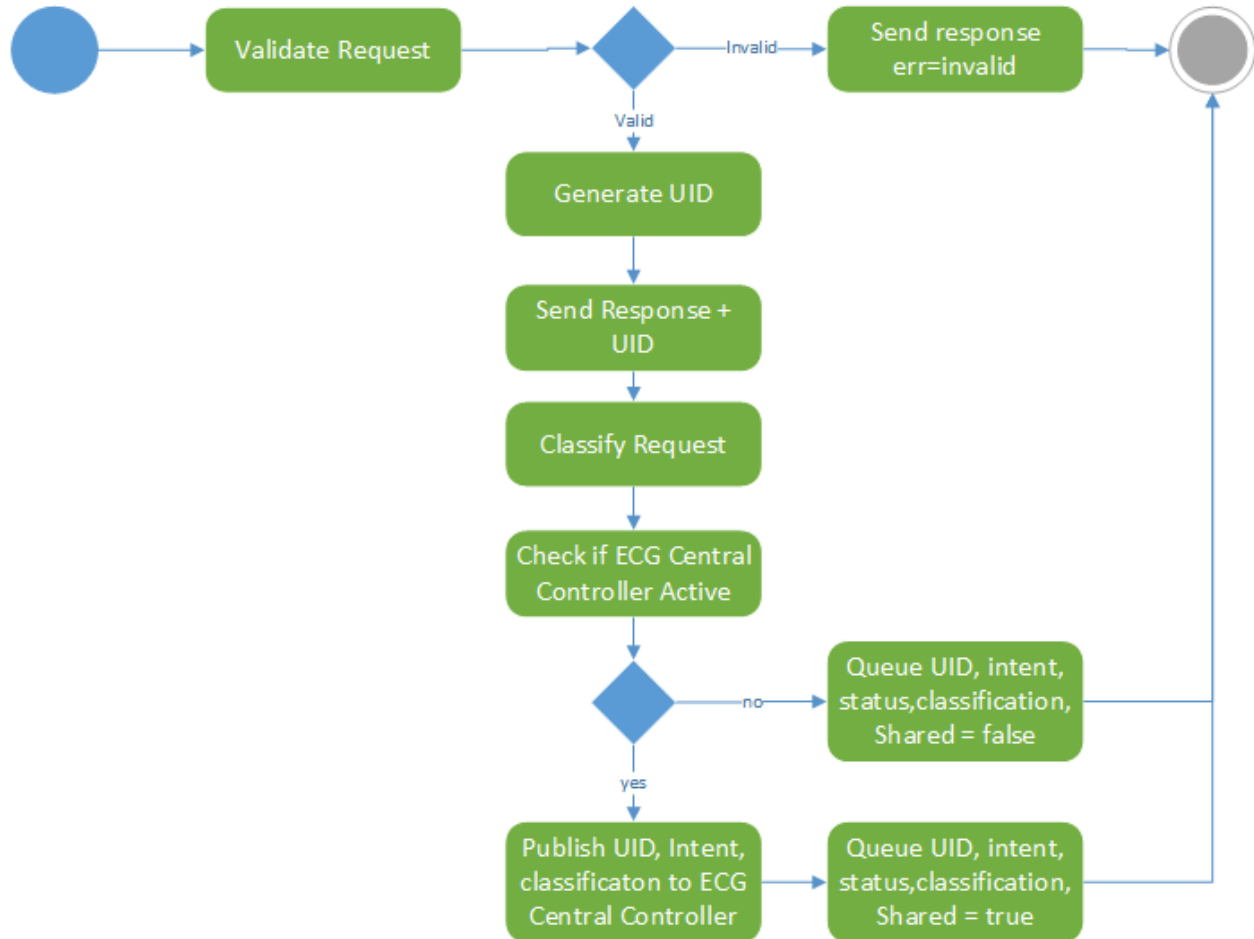
### 3.3.1.5 WIR-REQ-276081/B-Return Unique Id

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.

### 3.3.1.6 WIR-REQ-276082/A-Flow

The below UML diagram describes the flow of connection controller module



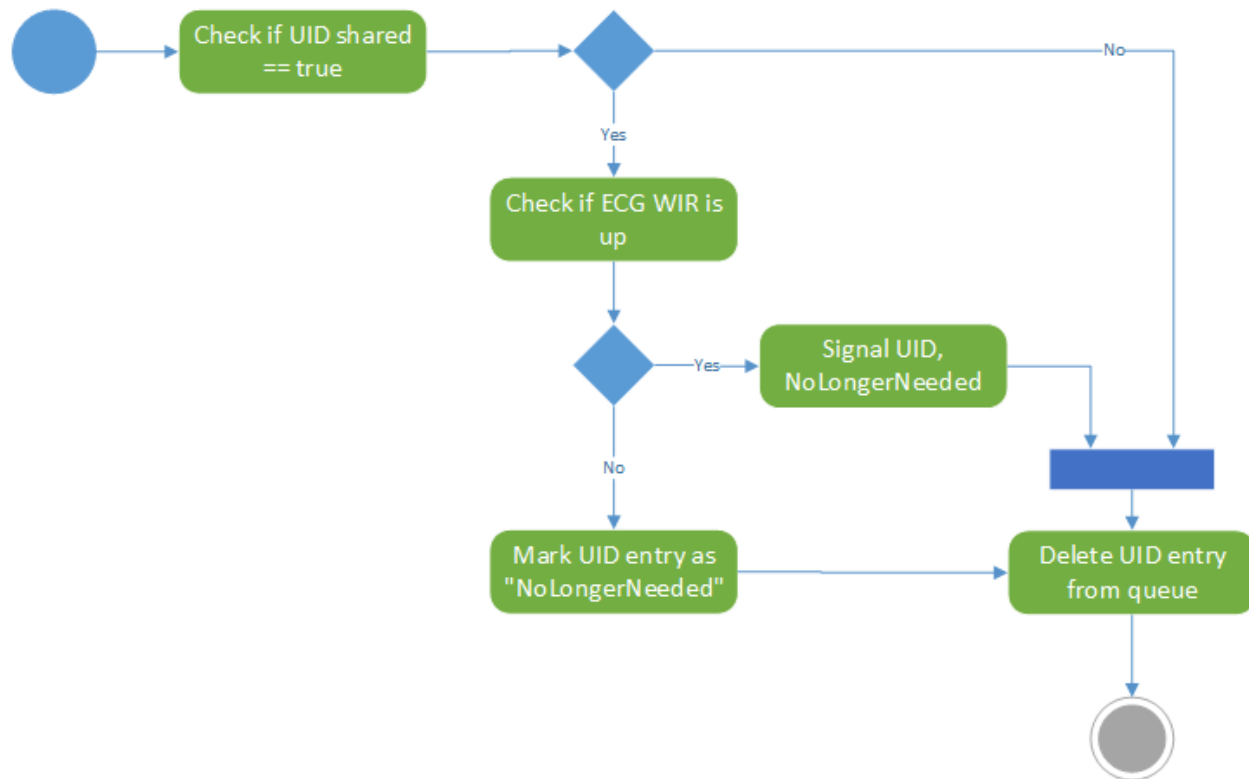
### 3.3.1.7 WIR-REQ-276083/A-Maintain UID's

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

### 3.3.1.8 WIR-REQ-276084/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.



### 3.3.1.9 WIR-REQ-276085/C-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.

### 3.3.1.10 WIR-REQ-276086/B-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 etc.

### 3.3.1.11 WIR-REQ-276087/B-Scheduling

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

### 3.3.1.12 WIR-REQ-295692/A-Removal From The Scheduling Queue

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

## 3.3.2 Use Cases

### 3.3.2.1 WIR-UC-REQ-296107/C-Local controller provides connection interface to requesting application

<b>Actors</b>	WIRClient1, WIRServer, WIRClient2, WIRClient3, WIRClient4, WIRClient5
<b>Pre-conditions</b>	1. Application requests WIR local controller for connection interface



<b>Scenario Description</b>	Local controller works with central controller to provide interfaces to requesting application
<b>Post-conditions</b>	Local controller successfully sends the request to the central controller Central controller processes the request and provides interface back to the local controller
<b>Interfaces</b>	
<b>Notes</b>	Exception use cases Central controller not available

### 3.3.2.2 WIR-UC-REQ-296108/C-Central controller not available

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

### 3.3.3 White Box View





### 3.4 WIR-FUN-REQ-276020/A-Intents

#### 3.4.1 Requirements

##### 3.4.1.1 WIR-REQ-276089/B-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. The Intent parameter also includes the URL which the application shall use.

##### 3.4.1.2 WIR-REQ-276090/D-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

The WIRClient2 applications can leverage the following types of intents

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

The WIRClient3 applications can leverage the following types of intents

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

The WIRClient4 applications can leverage the following types of intents

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

The WIRClient5 applications can leverage the following types of intents

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

The WIRServer applications can leverage the following types of intents

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

##### 3.4.1.3 WIR-REQ-276091/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.

#### 3.4.1.4 WIR-REQ-276092/A-Off-Peak

Due to vehicle battery power constraints, WIRClient1 applications are not allowed to use off-peak flags.

#### 3.4.1.5 WIR-REQ-276093/B-Off-Peak And Local Controller

If the WIRClient1 local controller receives off-peak as intent then it shall respond with an “un-supported” response.

#### 3.4.1.6 WIR-REQ-276094/D-Intent Structure

The Logical intent structure is as defined below:

Intent	Priority Level	Expiry timer	WiFi Preferred	Timer expiration	Supported Interface	Allowed app
Foreground	High Medium Low (default)	NA	Y : WiFi and fallback to cellular N : Cellular and fallback to WiFi	NA	Cellular APN1, Cellular APN2, APN3 (China only), APN4 (China only), WiFi	All
Background	NA	NA	NA	Return failure	WiFi	All except FCI app
Background guaranteed	NA	1~N	NA	Off-peak=Y : Convert to off-peak Off-peak=N : Convert to foreground cellular Once foreground intent is provided, WIR shall not save the request anymore.	WiFi and fallback to cellular. After fallback to cellular and WiFi becomes available, WIR shall provide WiFi again Supports APN3 and APN4 for China	All except FCI app
Special	NA	NA	NA	NA	Cellular APN1 (FCI), Cellular APN2 (WiFi HotSpot)	FCI app WiFi HotSpot app Amazon app
Off-peak	High Medium	NA	NA	NA	Cellular APN1	All

#### 3.4.1.7 WIR-REQ-370273/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.

##### 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.

##### 3.4.1.8 WIR-REQ-276095/A-Intent API

The intent API's shall be defined in subsequent releases of this document.

##### 3.4.1.9 WIR-REQ-276096/D-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, WIRClient2 APN3, WIRClient2 APN4, WIRClient5 APN1, WIRClient5 APN3, WIRClient5 APN4, WIRClient1 WLAN services, WIRClient2 WLAN services, WIRClient5 WLAN services	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 and off-peak selected	WIRClient2 APN1 off-peak, WIRClient5 APN1 off-peak	Traffic classification – 4
Background guaranteed timer expired and off-peak not selected	WIRClient2 APN1, WIRClient5 APN1	Traffic classification – 4
Off- peak	WIRClient2 APN 1 off-peak, WIRClient5 APN 1 off-peak	Traffic classification – 4 Traffic classification – 5
Special	WIRClient2 APN1, WIRClient5 APN1, 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

\* Applink is currently out of scope

##### 3.4.1.10 WIR-REQ-296059/C-Bandwidth Shaping

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

##### 3.4.1.11 WIR-REQ-370307/A-APN3 and APN4

WIRServer shall verify application requesting APN3 and APN4 has valid permission. APN3 and APN4 are supported for China vehicle only.



### 3.4.2 Use Cases

### 3.4.3 White Box View



### 3.5 WIR-FUN-REQ-276032/A-Policy Manager

#### 3.5.1 Requirements

##### 3.5.1.1 WIR-REQ-276116/C-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, WIRClient5, WIRServer and shall be updatable over the air from backend.

##### 3.5.1.2 WIR-REQ-276117/C-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 with URL.

##### 3.5.1.3 WIR-REQ-276118/A-Intent Policy Table Privileges

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

##### 3.5.1.4 WIR-REQ-276119/C-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
Reserved	12-15 (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.

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

##### 3.5.1.5 WIR-REQ-276120/C-Intent Policy Table Structure

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

Application ID	Application	Connectivity intent permissions

##### 3.5.1.6 WIR-REQ-276121/B-Intent Policy Storage

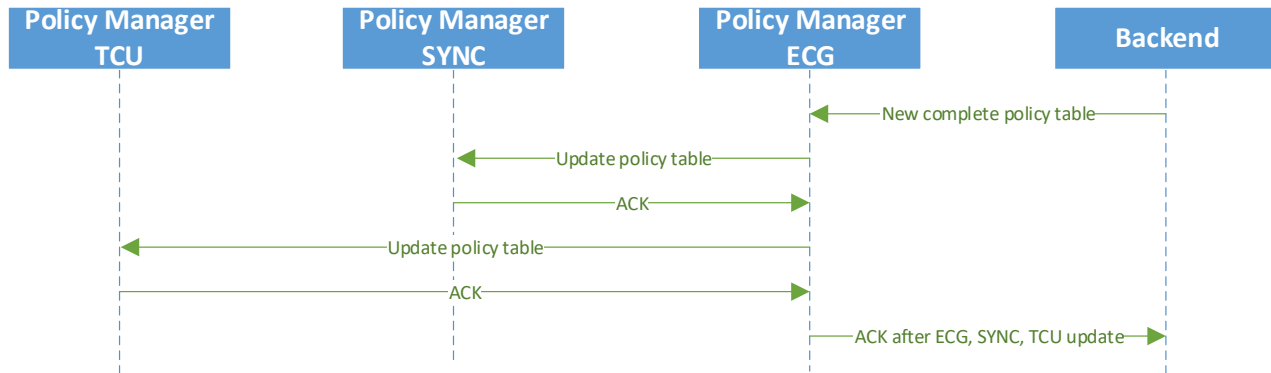
The WIRClient1 local controller shall store intent policy privilege.



### 3.5.1.7 WIR-REQ-276122/C-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 WIRClient5, WIRClient2 and WIRClient1. The WIRClient1, WIRClient2, and WIRClient5 shall in turn update their own policy tables.

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



### 3.5.1.8 WIR-REQ-370274/A-Sends Error for Corrupt policy file

WIRServer shall not update the policy table if a corrupt policy file is received, instead it shall send an error.

### 3.5.1.9 WIR-REQ-370514/A-WIRClient3 Policies

The WIRServer shall maintain the policies for WIRClient3. WIRClient3 shall have an updatable policy table in the future.

### 3.5.1.10 WIR-REQ-370516/A-WIRClient4 Policies

The WIRServer shall maintain the policies for WIRClient4. WIRClient4 shall have an updatable policy table in the future.

### 3.5.1.11 WIR-REQ-276123/B-Intent Policy Update Acknowledgement

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

### 3.5.1.12 WIR-REQ-276124/B-Intent Policy Update Notification To Applications

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

## 3.5.2 Use Cases

### 3.5.2.1 WIR-UC-REQ-296113/B-Cloud sends updated policy table to WIR

<b>Actors</b>	WIRClient1, WIRServer, WIRClient2, WIRClient5, Ford Cloud
<b>Pre-conditions</b>	1. Ford cloud sends updated policy table to vehicle
<b>Scenario Description</b>	WIRServer receives updated policy table from Ford Cloud and sends the updated policy to WIRClient1, WIRClient2, and WIRClient5.
<b>Post-conditions</b>	WIRServer receives the updated policy table from Ford Cloud WIRServer updates the policy and sends policies to WIRClient1, WIRClient2, and WIRClient5 WIRClient1, WIRClient2, and WIRClient5 updates the policy table and sends confirmation to WIRServer WIRServer sends confirmation to Ford cloud
<b>Interfaces</b>	
<b>Notes</b>	Exception use case 1. Policy not updated in WIRServer / WIRClients

**3.5.2.2 WIR-UC-REQ-296114/C-Policy not updated in WIRServer / WIRClients**

<b>Actors</b>	WIRClient1, WIRServer, WIRClient2, WIRClient5, Ford Cloud
<b>Pre-conditions</b>	1. Ford cloud sends updated policy table to vehicle
<b>Scenario Description</b>	WIRServer receives updated policy table from Ford Cloud and but not able to update policy table in vehicle
<b>Post-conditions</b>	WIRServer receives the updated policy table from Ford Cloud WIRServer updates the policy and sends policies to WIRClient1, WIRClient2, and WIRClient5 WIRClient1, WIRClient2, and WIRClient5 could not update the policy WIRServer sends failure to Ford cloud
<b>Interfaces</b>	
<b>Notes</b>	

**3.5.3 White Box View**



## 3.6 WIR-FUN-REQ-276022/A-Tunnel Manager

### 3.6.1 Requirements

#### 3.6.1.1 WIR-REQ-276100/C-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 and WIRServer.

WIRClient1's Wi-Fi is exposed to WIRServer, WIRClient2, WIRClient3, WIRClient4 and WIRClient5.

WIRClient2's Cellular and Wi-Fi interfaces are accessible from WIRClient1, WIRClient3, WIRClient4, WIRClient5 and WIRServer.

WIRClient5's Cellular and Wi-Fi interfaces are accessible from WIRClient1, WIRClient2, WIRClient3, WIRClient4 and WIRServer.

#### 3.6.1.2 WIR-REQ-276101/B-Tunnel Support

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

#### 3.6.1.3 WIR-REQ-276102/B-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).

#### 3.6.1.4 WIR-REQ-295870/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.

#### 3.6.1.5 WIR-REQ-295871/A-IP Address Assignment

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

#### 3.6.1.6 WIR-REQ-276103/B-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).

#### 3.6.1.7 WIR-REQ-276104/B-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.

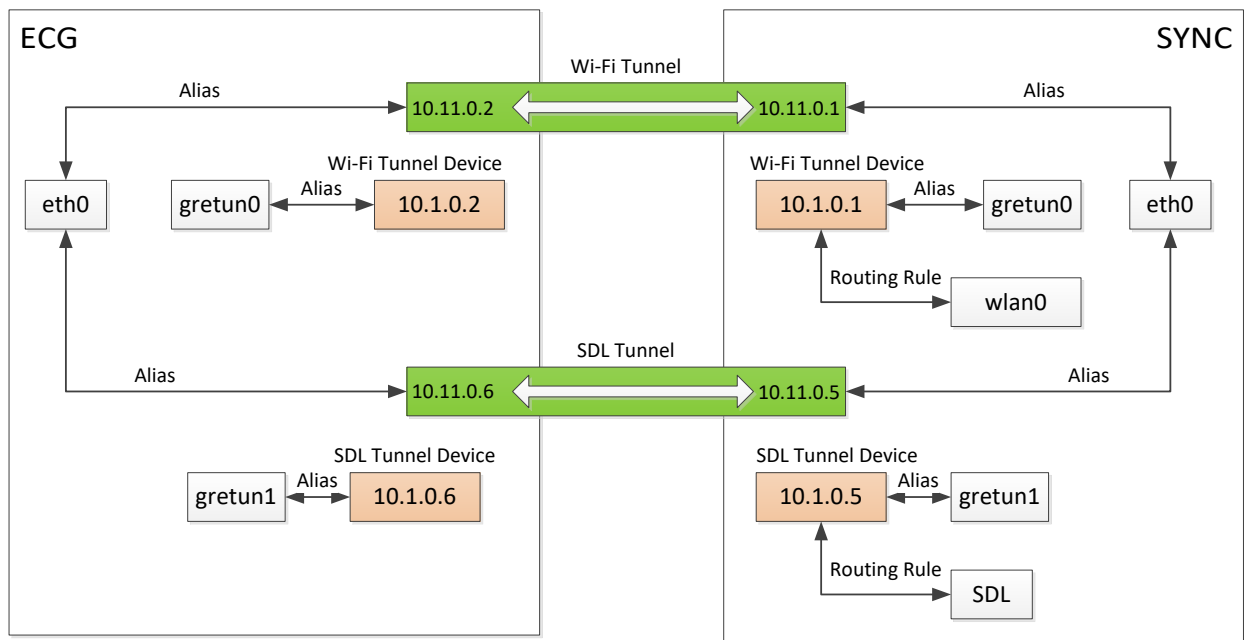
#### 3.6.1.8 WIR-REQ-276105/C-Allow WIRClient1 Applications To Access Edge Interface Of Other Ethernet Connected ECU's

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

#### 3.6.1.9 WIR-REQ-276106/B-WIRClient1 Interfacing With WIRServer Example

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

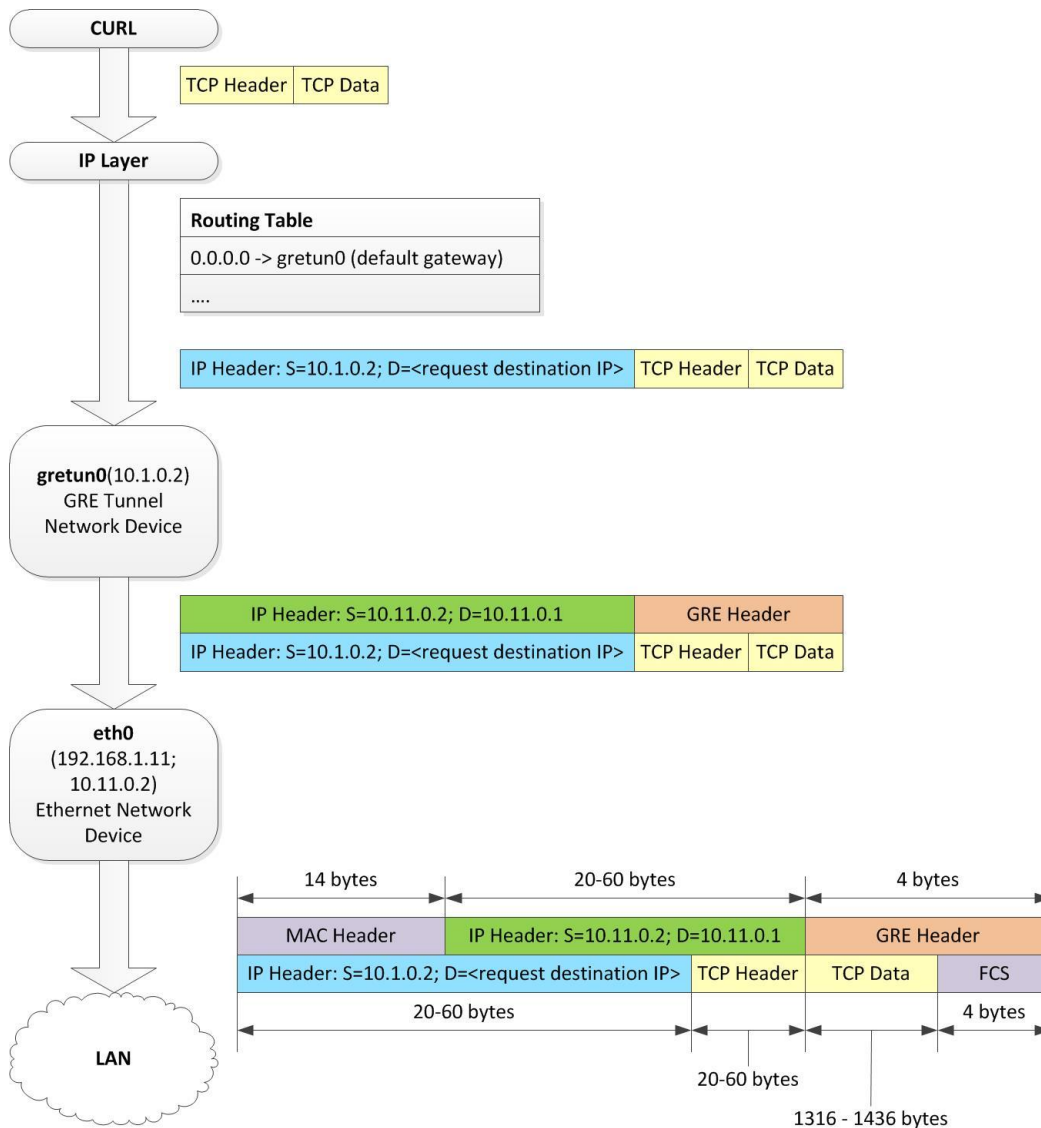




GRE tunnel encapsulation flow

### 3.6.1.10 WIR-REQ-276107/B-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.



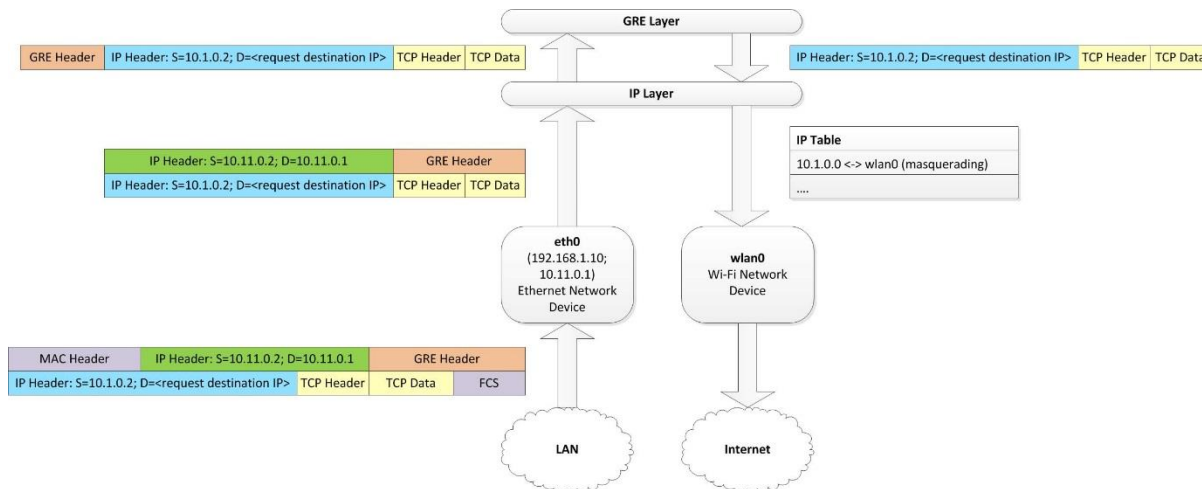
Sending request on the WIRServer

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.

#### 3.6.1.11 WIR-REQ-276108/B-Forwarding The Request To WIRClient1 WIFI Example Proposal

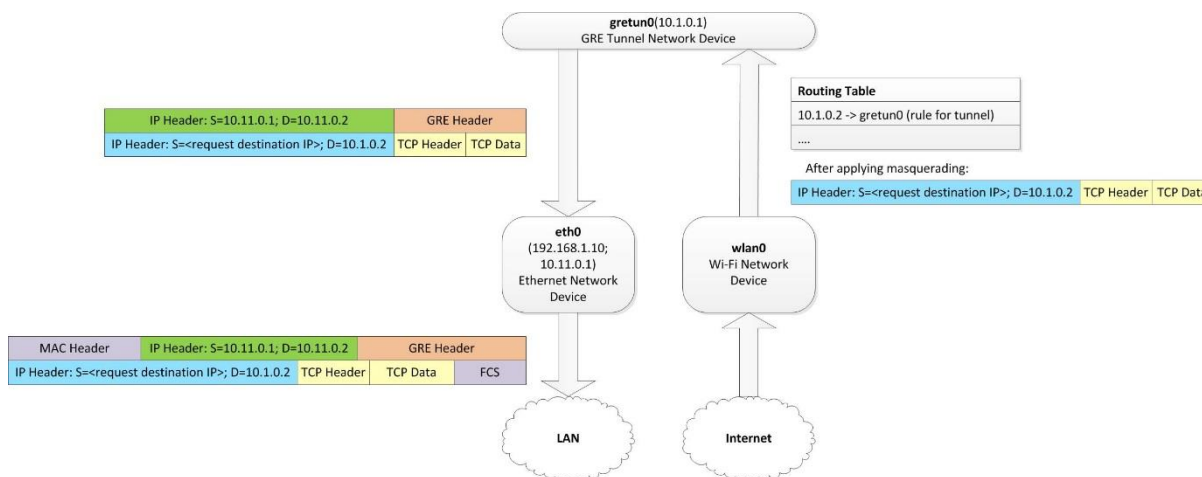


Sending request on WIRClient1

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.

### 3.6.1.12 WIR-REQ-276109/B-Receiving Response On WIRClient1 Module Example Proposal

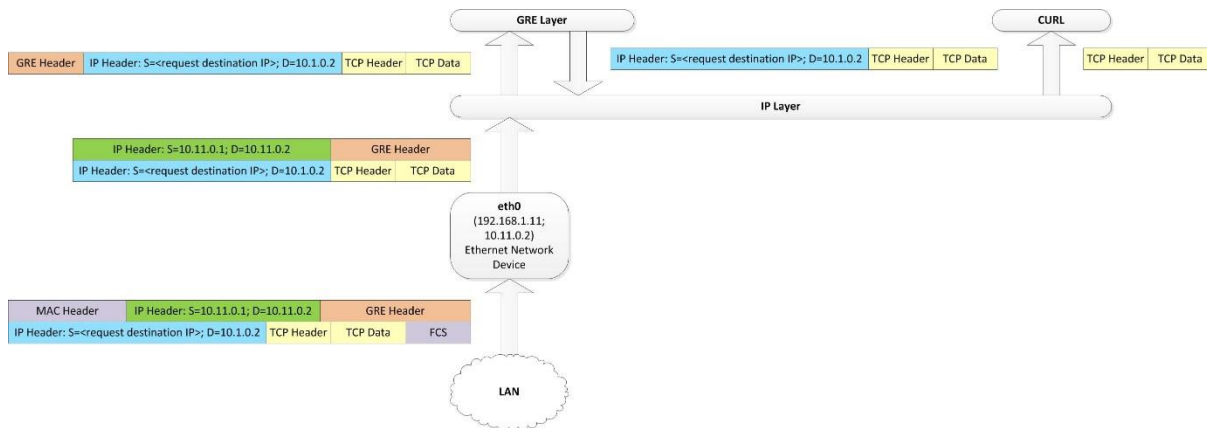
When the destination host in the cloud sends a response IP packet, it arrives at the WIRClient1 Wi-Fi interface. The response IP packet specifies the original source IP as 10.1.0.2, the address of the original GRE tunnel. The WIRClient1 IP layer identifies a rule that routes all packets destined for 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

### 3.6.1.13 WIR-REQ-276110/B-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

### 3.6.2 Use Cases

### 3.6.3 White Box View



### 3.7 WIR-FUN-REQ-276030/A-Diagnostics

#### 3.7.1 Requirements

##### 3.7.1.1 WIR-REQ-276112/C-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.

##### 3.7.1.2 WIR-REQ-276113/C-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
- Data rate
- Band and channels utilized
- RSSI statistics
- Chipset roles / use
- Statistics about the data transmitted and received if possible
- Reason for disconnect

##### 3.7.1.3 WIR-REQ-276114/B-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
- Data rate
- Band and channels utilized if any
- Reason for disconnect

##### 3.7.1.4 WIR-REQ-276159/D-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.

##### 3.7.1.5 WIR-REQ-370308/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.

##### 3.7.1.6 WIR-REQ-276160/C-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.

##### 3.7.1.7 WIR-REQ-295872/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 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 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.

### 3.7.2 Use Cases

#### 3.7.2.1 WIR-UC-REQ-296112/D-WIR provides diagnostics information to cloud based on request from Cloud

<b>Actors</b>	WIRClient1, WIRServer, WIRClient2, WIRClient3, WIRClient4, WIRClient5, Ford Cloud
<b>Pre-conditions</b>	1. Customer triggers bug report
<b>Scenario Description</b>	Customer has a connectivity issue and triggers bug report to be sent to the cloud via on demand diagnostics
<b>Post-conditions</b>	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
<b>Interfaces</b>	
<b>Notes</b>	

### 3.7.3 White Box View



### 3.8 WIR-FUN-REQ-289654/A-Data Usage

#### 3.8.1 Requirements

##### 3.8.1.1 WIR-REQ-289655/C-Data usage calculation

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

##### 3.8.1.2 WIR-REQ-289656/C-Data usage calculation scope

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

##### 3.8.1.3 WIR-REQ-289657/C-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 and WIRServer then data usage shall be calculated collectively and aggregated by WIRServer.

##### 3.8.1.4 WIR-REQ-289658/D-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) 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-370326-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.

##### 3.8.1.5 WIR-REQ-289659/B-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.



### 3.8.1.6 WIR-REQ-289660/B-Data usage storage

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

### 3.8.1.7 WIR-REQ-289661/B-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).

### 3.8.1.8 WIR-REQ-289662/B-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.

### 3.8.1.9 WIR-REQ-289663/B-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.

### 3.8.1.10 WIR-REQ-289664/B-Data usage data transmission format

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

### 3.8.1.11 WIR-REQ-289665/B-Data usage request processing from in vehicle HMI

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

### 3.8.1.12 WIR-REQ-289666/B-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.

### 3.8.1.13 WIR-REQ-289667/B-Accuracy of data usage calculation

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

### 3.8.1.14 WIR-REQ-289668/B-Data usage timestamp for cellular

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

### 3.8.1.15 WIR-REQ-370326/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.

## 3.8.2 Use Cases

## 3.8.3 White Box View





### 3.9 WIR-FUN-REQ-295873/A-WIFI Connect Reminders - HMI

#### 3.9.1 Requirements

##### 3.9.1.1 WIR-REQ-295874/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.

##### 3.9.1.2 WIR-REQ-295875/B-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.

##### 3.9.1.3 WIR-REQ-295876/A-Trigger Frequency

The trigger frequency is WIFI\_CONNECT\_REMINDER\_COUNTER ignition cycles. The initial value for WIFI\_CONNECT\_REMINDER\_COUNTER is 100.

##### 3.9.1.4 WIR-REQ-295877/B-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.

##### 3.9.1.5 WIR-REQ-295878/B-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.

##### 3.9.1.6 WIR-REQ-295879/B-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.

##### 3.9.1.7 WIR-REQ-295880/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.

##### 3.9.1.8 WIR-REQ-295881/B-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.

##### 3.9.1.9 WIR-REQ-295882/B-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.



### 3.9.2 Use Cases

### 3.9.3 White Box View



### 3.10 WIR-FUN-REQ-370276/A-Captive Portal Check

#### 3.10.1 Requirements

##### 3.10.1.1 WIR-REQ-370277/A-Captive portal check support

WIRClient1 shall support captive portal check operation.

##### 3.10.1.2 WIR-REQ-370278/A-WLAN connection request

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

##### 3.10.1.3 WIR-REQ-370279/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.

##### 3.10.1.4 WIR-REQ-370280/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.

##### 3.10.1.5 WIR-REQ-370281/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.

##### 3.10.1.6 WIR-REQ-370282/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.



### 3.10.1.7 WIR-REQ-370283/A-WIRClient1 UI restriction

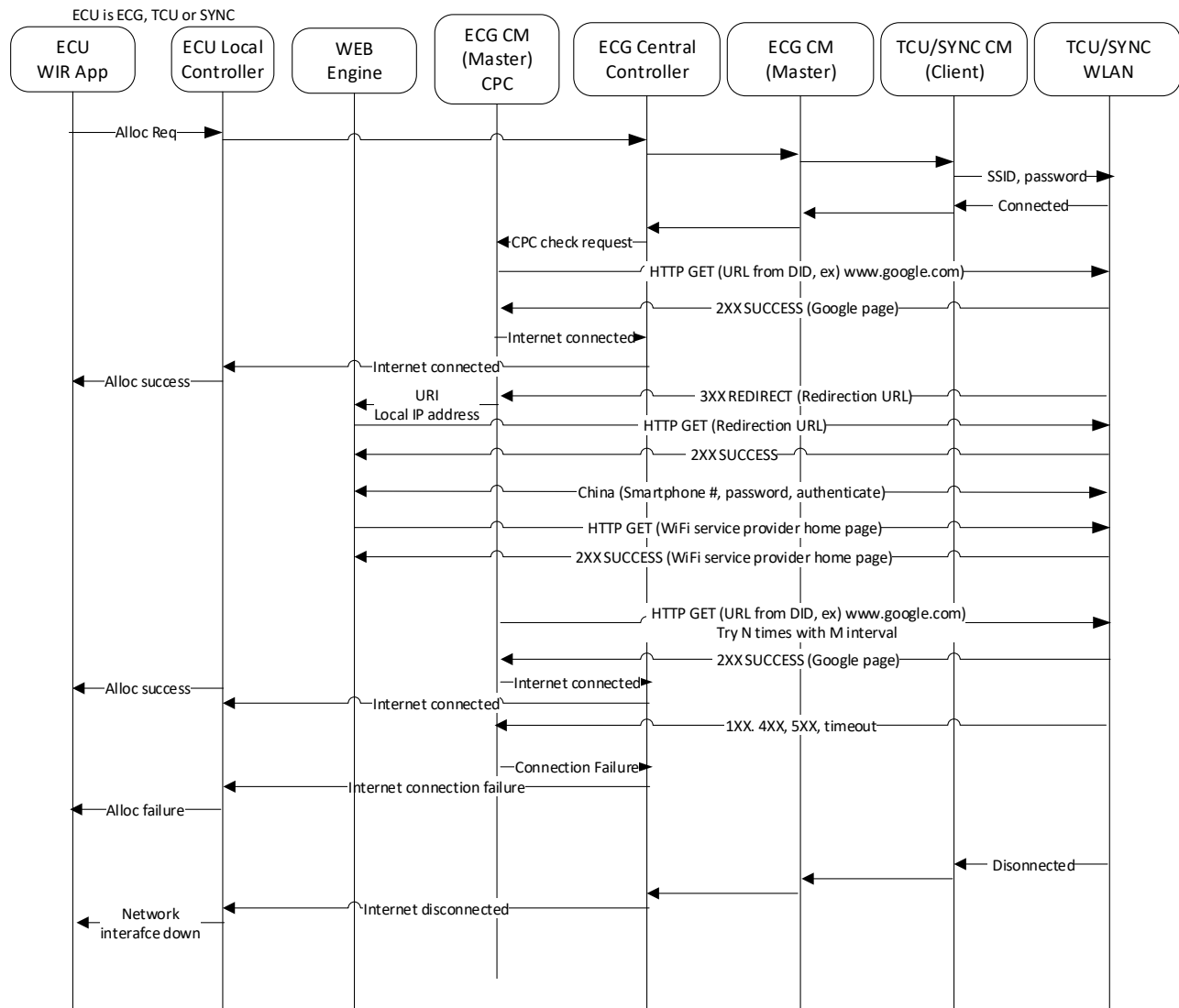
After received a captive portal check request, CPC shall wait CPC\_CHECK\_INTERVAL seconds. After the wait, CPC shall send HTTP request to CPC\_CHECK\_URL.

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.

### 3.10.1.8 WIR-REQ-370284/A-CPC check procedure

Following diagram shows CPC check procedure flow.



## 3.10.2 Use Cases

### 3.10.2.1 WIR-UC-REQ-370285/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



<b>Scenario Description</b>	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
<b>Post-conditions</b>	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
<b>List of Exception Use Cases</b>	
<b>Interfaces</b>	

### 3.10.2.2 WIR-UC-REQ-370286/A-WIRClient1 Wi-Fi connected to a Wi-Fi Hotspot first time and Wi-Fi Hotspot doesn't have landing page

<b>Actors</b>	WIRClient1 Wi-Fi, Wi-Fi Hotspot, customer, WIR
<b>Pre-conditions</b>	WIRClient1 Wi-Fi turned on WIRClient1 Wi-Fi is scanning available Wi-Fi Hotspot signal
<b>Scenario Description</b>	WIRClient1 Wi-Fi finds available Wi-Fi Hotspot and makes a connection to it. Wi-Fi Hotspot provides internet connection without landing page redirection
<b>Post-conditions</b>	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
<b>List of Exception Use Cases</b>	
<b>Interfaces</b>	

### 3.10.2.3 WIR-UC-REQ-370287/A-WIRClient1/WIRClient2 Wi-Fi connected to a Wi-Fi Hotspot connected previously

<b>Actors</b>	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, WIR
<b>Pre-conditions</b>	WIRClient1/WIRClient2 Wi-Fi turned on WIRClient1/WIRClient2 Wi-Fi is scanning available Wi-Fi Hotspot signal
<b>Scenario Description</b>	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
<b>Post-conditions</b>	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	

### 3.10.2.4 WIR-UC-REQ-370288/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	

### 3.10.2.5 WIR-UC-REQ-370289/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	

**3.10.2.6 WIR-UC-REQ-370290/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	

**3.10.2.7 WIR-UC-REQ-370291/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	

### 3.10.2.8 WIR-UC-REQ-370292/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	

### 3.10.2.9 WIR-UC-REQ-370293/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	

### 3.10.2.10 WIR-UC-REQ-370294/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





<b>Post-conditions</b>	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
<b>List of Exception Use Cases</b>	
<b>Interfaces</b>	

### 3.10.2.11 WIR-UC-REQ-370295/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot again but authentication fails

<b>Actors</b>	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, customer, WIR
<b>Pre-conditions</b>	WIRClient1/WIRClient2 Wi-Fi turned on WIRClient1/WIRClient2 Wi-Fi is scanning available Wi-Fi Hotspot signal
<b>Scenario Description</b>	Saved password is not valid and customer enters password
<b>Post-conditions</b>	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
<b>List of Exception Use Cases</b>	
<b>Interfaces</b>	

### 3.10.2.12 WIR-UC-REQ-370296/A-Wi-Fi connection lost after Wi-Fi Hotspot timeout

<b>Actors</b>	WIRClient1/WIRClient2 Wi-Fi, Wi-Fi Hotspot, WIR
<b>Pre-conditions</b>	WIRClient1/WIRClient2 Wi-Fi is connected to Wi-Fi Hotspot WIR has internet connection through WIRClient1/WIRClient2 Wi-Fi
<b>Scenario Description</b>	WIRClient1/WIRClient2 Wi-Fi connection lost after timeout
<b>Post-conditions</b>	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	

**3.10.2.13 WIR-UC-REQ-370297/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	

**3.10.2.14 WIR-UC-REQ-370298/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	

**3.10.2.15 WIR-UC-REQ-370299/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	

**3.10.2.16 WIR-UC-REQ-370300/A-Other screen overrides while landing page displayed**

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



<b>Scenario Description</b>	Other screen override landing page screen
<b>Post-conditions</b>	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
<b>List of Exception Use Cases</b>	
<b>Interfaces</b>	

**3.10.2.17 WIR-UC-REQ-370301/A-WEB engine supports different WIRClient1 screen size**

<b>Actors</b>	WEB engine, CPC
<b>Pre-conditions</b>	CPC requests WEB engine to display WEB page
<b>Scenario Description</b>	WEB engine formats WEB page based on WIRClient1 screen size
<b>Post-conditions</b>	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
<b>List of Exception Use Cases</b>	
<b>Interfaces</b>	

**3.10.2.18 WIR-UC-REQ-370302/A-CPC get internet connection success after multiple try**

<b>Actors</b>	CPC, WIR
<b>Pre-conditions</b>	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
<b>Scenario Description</b>	CPC checks internet connectivity and gets success after multiple try
<b>Post-conditions</b>	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
<b>List of Exception Use Cases</b>	
<b>Interfaces</b>	

**3.10.2.19 WIR-UC-REQ-370303/A-CPC get internet connection failure after max try**

<b>Actors</b>	CPC, WIR
<b>Pre-conditions</b>	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
<b>Scenario Description</b>	CPC checks internet connectivity and gets failure after max try
<b>Post-conditions</b>	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
<b>List of Exception Use Cases</b>	
<b>Interfaces</b>	

**3.10.2.20 WIR-UC-REQ-370304/A-Password enter screen interrupted by other screen**

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

**3.10.2.21 WIR-UC-REQ-370305/A-WEB engine fails to display landing page**

<b>Actors</b>	CPC, WEB engine, diagnostic agent, WIR
<b>Pre-conditions</b>	WEB engine received landing page display request from CPC
<b>Scenario Description</b>	WEB engine fails to display landing page and CPC returns internet connection failure to WIR
<b>Post-conditions</b>	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
<b>List of Exception Use Cases</b>	
<b>Interfaces</b>	

**3.10.2.22 WIR-UC-REQ-370306/A-WEB engine opens 2nd WEB page while landing page is displayed---TBD**

<b>Actors</b>	WEB engine, customer
<b>Pre-conditions</b>	WEB engine displayed landing page on WIRClient1 screen
<b>Scenario Description</b>	2 <sup>nd</sup> WEB engine override landing page screen



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

### 3.10.3 White Box View



### 3.11 WIR-FUN-REQ-296062/A-Performance

#### 3.11.1 Requirements

##### 3.11.1.1 WIR-REQ-296063/B-Boot up

WIR module in WIRServer, WIRClient1, WIRClient2, WIRClient3, WIRClient4, WIRClient5 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-296064/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-296065/B-WIRServer and WIRClient interfaces

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

##### 3.11.1.4 WIR-REQ-296066/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-296067/B-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-296082/A-WIRClient1 Configuration

### 3.12.1 Requirements

#### 3.12.1.1 WIR-REQ-296091/B-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	
2	
3	
4	
5	
6	
7	
8	
9	