



Research & Vehicle Technology
“Infotainment Systems Product Development”

Feature – EV Trip Planner

**APIM Infotainment Subsystem Part Specific
Specification (SPSS)**

Version 1.0

UNCONTROLLED COPY IF PRINTED

Version Date: March 15, 2019

FORD CONFIDENTIAL



Revision History

Date	Version	Notes	
March 15, 2019	1.0	Initial Release	



Table of Contents

REVISION HISTORY	2
1 OVERVIEW	5
1.1 Feature Operation	5
1.2 Feature Assumptions	5
1.3 Logical Block Diagram.....	6
1.4 Terminology and Abbreviations.....	6
2 ARCHITECTURAL DESIGN.....	7
2.1 CLD-REQ-345898/A-EV Trip Planner Interface Client	7
2.2 CLD-REQ-345899/A-EV Trip Planner Vehicle Data Server.....	7
2.3 CLD-REQ-345914/A-EV Trip Planner Status Client	7
2.4 CLD-REQ-346070/A-EV Trip Planner OnBoard Server.....	7
2.5 CLD-REQ-346072/A-EV Trip Planner OffBoard Server1.....	7
2.6 CLD-REQ-346355/A-EV Trip Planner OffBoard Server2.....	7
2.7 CLD-REQ-346071/A-EV Trip Planner OffBoard Client.....	7
2.8 Physical Mapping of Classes	7
2.9 IIR-REQ-345900/A-EV Trip Planner Interface Client _Rx.....	8
2.9.1 MD-REQ-345902/A-VehicleData_St	8
2.9.2 MD-REQ-346075/A-TripData	11
2.9.3 MD-REQ-346073/A-RouteData	11
2.9.4 MD-REQ-346687/A-getTripData_Rsp.....	11
2.9.5 MD-REQ-346688/A-putTripData_Rsp.....	12
2.9.6 MD-REQ-346686/A-getRouteData_Rsp	12
2.10 IIR-REQ-345903/A-EV Trip Planner Interface Client _Tx	12
2.10.1 MD-REQ-345904/A-DistanceToStopover_St.....	12
2.10.2 MD-REQ-345905/A-StopoverType_St	12
2.10.3 MD-REQ-346074/A-getTripData_Rq	13
2.10.4 MD-REQ-346351/A-putTripData_Rq	13
2.10.5 MD-REQ-346075/A-TripData	14
2.10.6 MD-REQ-346076/A-getRouteData_Rq	14
3 GENERAL REQUIREMENTS	15
3.1 EVTP-REQ-346353/A-Utilization of Wireless Interface Router framework.....	15
3.2 EVTP-REQ-345919/A-Utilization of IP Pass-Through Framework	15
3.3 EVTP-REQ-346695/A-Response Codes from the EV Trip Planner OffBoard Servers.....	15
3.4 EVTP-REQ-346982/A-Delete Token and Trip Data Upon a Master Reset	15
3.5 EVTP-REQ-346354/A-EV Trip Planner CCS Requirement	15
4 FUNCTIONAL DEFINITION	16
4.1 EVTP-FUN-REQ-345906/A-Load New Trips from Cloud	16
4.1.1 Requirements	16
4.1.2 White Box View	17
4.2 EVTP-FUN-REQ-345931/A-Update Existing Trip In-Vehicle.....	19
4.2.1 Requirements	19



4.2.2	White Box View	19
4.3	<i>EVTP-FUN-REQ-345923/A-Create New Trip In-Vehicle</i>	20
4.3.1	Requirements	20
4.3.2	White Box View	21
4.4	<i>EVTP-FUN-REQ-345939/A-Update Stopover Status</i>	22
4.4.1	Requirements	22
4.4.2	White Box View	23
5	APPENDIX: REFERENCE DOCUMENTS	25



1 Overview

EV Trip Planner or Online Trip Planner (OTP) is a navigation feature for Electric Vehicles to be able to prepare an optimal (shortest distance or shortest time) yet safest route. Safest route refers to the route where there are enough EV chargers to charge the EV vehicle to last long enough to reach the destination. This is to ensure that the user is not stranded anywhere along the route due a low battery charge and all the data the user needs is available to the user for charging the vehicle.

1.1 Feature Operation

The User is planning a trip in advance using the FordPass app on their mobile device. For a trip that starts in Chicago with a destination of a hotel in New York, multiple stops (or "stopovers") on the way could be planned by the user, such as a stopover at a restaurant in Detroit and a charge station in Cleveland. The User knows that their EV vehicle needs charging and would like the vehicle to make well informed decisions in case the user's knowledge about where to charge is limited. With Chicago being origin, the restaurant in Detroit becomes waypoint1, the charge station in Cleveland becomes waypoint2 and hotel in New York becomes the final destination.

Based upon above user provided data, the NavCloud will return a route to the FordPass app depicting the route from the Chicago to New York. When the User is ready to start the trip, they will start the car and be prompted to accept this trip on the in-vehicle HMI. Upon User confirmation, the NavCloud would call the VPOI to provide the EV charging station data along the route. A route will be generated and sent it to the vehicle. Assuming a Distance to Empty range of approximately 250 miles, the Navigation system determines the distance between Chicago to Detroit, that turns out to be more than 250 miles (280 miles), so the Navigation system will have to add one waypoint as a charging station along the route. Similarly the user could manually add a waypoint as a charging station while planning this trip using Ford Pass, in which case, the Navigation system would not need to add another charge station assuming that user is stopping over on a compatible charge station.

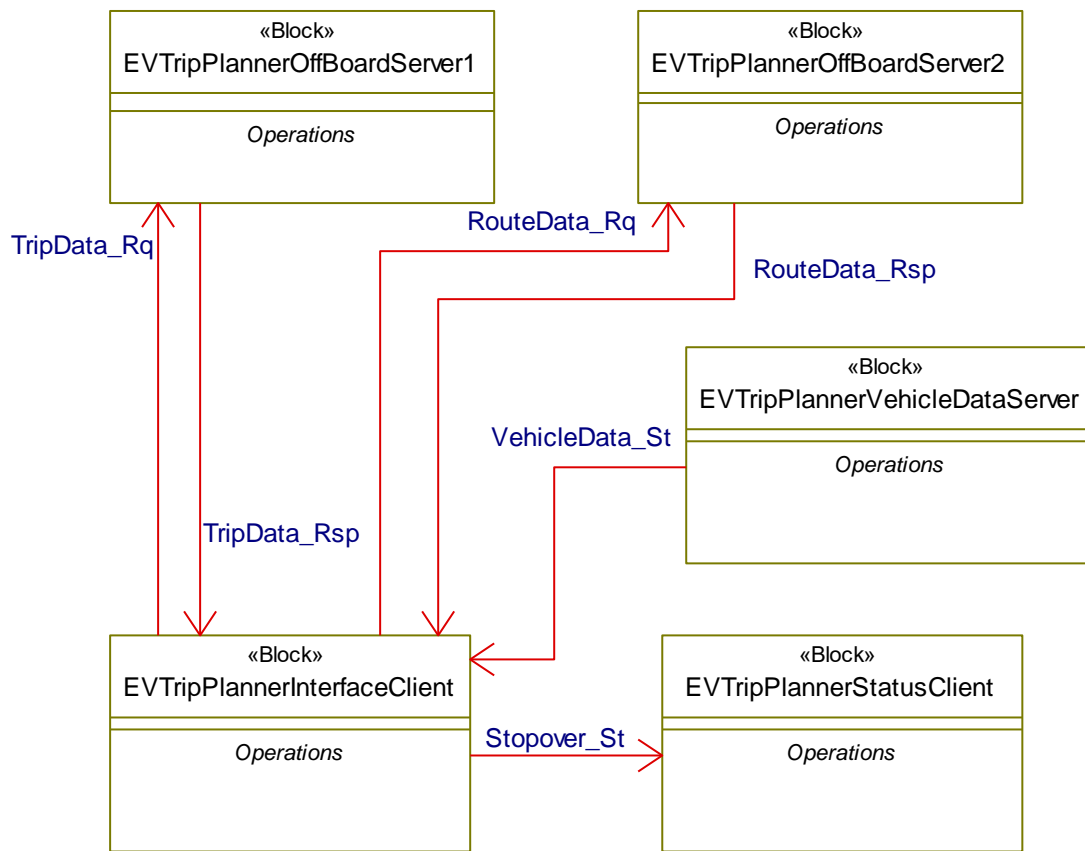
Depending on the user's driving behavior along the route (speed), outside air temperature, the battery level, and other vehicle data, the Navigation system will determine the possible range between various stopovers and add any additional waypoints (charging stations) as needed. In this case the NavCloud will update the route and the Navigation system will update the trip and send the updated traip back to the FordPass Cloud subject to modem connectivity.

1.2 Feature Assumptions

This feature is designed in such a manner that it will utilize existing implementation of features such as "Connection Manager", "IP Pass-Through", and "Online Traffic". This feature also utilizes the FordPass application on a compatible mobile device for the off-board trip planning functions to be supported. There will be in-vehicle HMI for on-board trip planning functions such as creating a trip and updating a trip.



1.3 Logical Block Diagram



1.4 Terminology and Abbreviations

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

Term	Description
IPPT	Internet Protocol Pass-through
LBS	Location Based Services
VPOI	Vehicle Point of Interest
DTE	Distance to Empty
HPCM	Hybrid Powertrain Control Module
BECM	Battery Energy Control Module
ECG	Enhanced Central Gateway
FP	Ford Pass
VSDN	Vehicle Service Delivery Network
WIR	Wireless Interface Router
FCI	Ford Cloud Interface
SDK	Software Development Kit
APIM	Accessory Protocol Interface Module
IPC	Instrument Panel Cluster



2 Architectural Design

2.1 CLD-REQ-345898/A-EV Trip Planner Interface Client

The EV Trip Planner Interface Client is responsible for containing the core logic of the EV Trip Planner feature, such as providing the In-Vehicle HMI for the User to interact with, loading and updating any trips from the Cloud, and creating any new trips in-vehicle. The EV Trip Planner Client is also responsible for receiving vehicle data from the EV Trip Planner Server and providing trip data to the EV Trip Planner Status Client.

2.2 CLD-REQ-345899/A-EV Trip Planner Vehicle Data Server

The EV Trip Planner Vehicle Data Server is responsible for providing vehicle data to the EV Trip Planner Interface Client.

2.3 CLD-REQ-345914/A-EV Trip Planner Status Client

The EV Trip Planner Status Client is responsible for receiving trip data from the EV Trip Planner Interface Client and providing some In-Vehicle HMI to update the user on the status of the planned trip along and how it relates to certain elements of vehicle data.

2.4 CLD-REQ-346070/A-EV Trip Planner OnBoard Server

The EV Trip Planner OnBoard Server shall act as gateway and is responsible for relaying requests to receive/send Trip and Route data, and the data payloads themselves, from the EV Trip Planner Interface Client to the EV Trip Planner OffBoard Servers.

2.5 CLD-REQ-346072/A-EV Trip Planner OffBoard Server1

The EV Trip Planner OffBoard Server1 is responsible for processing requests to send and receive trip data to and from the EV Trip Planner Interface Client.

2.6 CLD-REQ-346355/A-EV Trip Planner OffBoard Server2

The EV Trip Planner OffBoard Server2 is responsible for processing requests to generate and send route data to the EV Trip Planner Interface Client.

2.7 CLD-REQ-346071/A-EV Trip Planner OffBoard Client

The EV Trip Planner OffBoard Client is responsible for allowing the user to plan a new trip remotely, as well as receive updates to any existing trips.

2.8 Physical Mapping of Classes

The table below shows an example of how the logical classes that make up the EV Trip Planner feature may be mapped into physical modules. This mapping example is specific to the FNV2 architecture on EV vehicles and does not necessarily carry over to other carlines or vehicle architectures.

Logical Class	Physical Module (ECU)
EV Trip Planner Interface Client	APIM
EV Trip Planner Vehicle Data Server1	HPCM
EV Trip Planner Vehicle Data Server2	BECM
EV Trip Planner Status Client	IPC
EV Trip Planner On Board Server	ECG
EV Trip Planner Off Board Server1	Ford VPOI Cloud Services
EV Trip Planner Off Board Server2	3 rd Party Nav Cloud Services
EV Trip Planner Off Board Client	Mobile Device



2.9 IIR-REQ-345900/A-EVTripPlannerInterfaceClient _Rx

2.9.1 MD-REQ-345902/A-VehicleData_St

Message Type: Status

The method is used to report the status of multiple vehicle data related signals relevant for EV Trip Planning to the EVTripPlannerInterfaceClient

Name	Literals	Value	Description
DistanceToEmpty_St	-	-	The remaining distance in km before battery is depleted
	<range>	0x0-0xFFD	0.0 – 409.3 km (resolution of tenths of a km)
	Unknown	0xFFE	
	Faulted	0xFFF	
Charging_St	-	-	Signal represents the vehicle charging status and indicates if there are any charge faults.
	NotReady	0x0	
	Fault Inside Car	0x1	
	Fault Outside Car	0x2	
	EVSE Not Compatible	0x3	
	EVSE Not Detected	0x4	
	EVSE Paused	0x5	
	Charge Scheduled	0x6	
	Charging – AC	0x7	
	Charging - DC Fast Charge	0x8	
	Charging – Inductive	0x9	
	Charging – System Maintain	0xA	
	Cabin Preconditioning	0xB	
	Drive Conditioning	0xC	
	Charge Target Reached	0xD	
	NotUsed_1	0xE	
	NotUsed_2	0xF	
BatteryChargeLevel_St	-	-	Current battery charge level as a percentage
	<range>	0x0-0xFD	0.0 – 126.5% (resolution of 0.5%)
	NoDataExists	0xFE	
	Faulty	0xFF	
VehicleEnergyEfficiency_St	-	-	Average energy per unit distance in wattHr/km
	<range>	0x0-0x7D	-100 – 1150 wattHr/km (resolution of 10 wattHr/km, with an offset of -100)



	NoDataExists	0x7E	
	Faulty	0x7F	
RangePerChargeAvg_St	-	-	Vehicle Range in km with full charge
	<range>	0x0-0xFFD	0.0 – 409.3 km (resolution of tenths of a km)
	NoDataExists	0xFFE	
	Faulty	0xFFF	
TracBatteryAvailCapacity_St	-	-	Available high voltage traction battery capacity in watt hours
	<range>	0x0-0xFFD	0 – 204,650 watt hours (resolution of 50 watt hours)
	NoDataExists	0xFFE	
	Faulty	0xFFF	
TracBatteryMaxCapacity_St	-	-	Maximum high voltage traction battery capacity in watt hours
	<range>	0x0-0xFFD	0 – 204,650 watt hours (resolution of 50 watt hours)
	NoDataExists	0xFFE	
	Faulty	0xFFF	
BatteryChargeTargetSoC_St	-	-	Charge target for the battery's State of Charge
	<range>	0x0-0x7F	0 – 127% (resolution of 1%)
DestinationChargeEndDay_St	-	-	Charge time in days to achieve the target SOC to allow the vehicle to reach the destination
	<range>	0x1-0x1F	1 – 31 days (resolution of 1 day)
DestinationChargeEndHr_St	-	-	Charge time in hours to achieve the target SOC to allow the vehicle to reach the destination
	<range>	0x0-0x1F	0 – 31 hours (resolution of 1 hr)
DestinationChargeEndMin_St	-	-	Charge time in min to achieve the target SOC to allow the vehicle to reach the destination
	<range>	0x0-0x3F	0 – 63 min (resolution of 1 min)
ChargeStartTimeYear2_St	-	-	Charge Start Time Year
	<range>	0x0-0x7F	2017 - 2144 (resolution of 1 year, offset of 2017)



ChargeStartTimeYear_St	-	-	Charge Start Time Year
	<range>	0x0-0x1E	2010 - 2040 (resolution of 1 year, offset of 2010)
	Invalid	0x1F	
ChargeStartTimeMonth_St	-	-	Charge Start Time Month
	<range>	0x0-0xE	1 - 15 (resolution of 1 month, offset of 1)
	Invalid	0xF	
ChargeStartTimeDay_St	-	-	Charge Start Time Day
	<range>	0x0-0x1E	1 – 31 days (resolution of 1 day, offset of 1)
	Invalid	0xF	
ChargeStartTimeHour_St	-	-	Charge Start Time Hour
	<range>	0x0-0x1D	0 – 29 hours (resolution of 1 hour)
	Unknown	0xE	
	Invalid	0xF	
ChargeStartTimeMin_St	-	-	Charge Start Time Minute
	<range>	0x0-0x3D	0 – 61 minutes (resolution of 1 min)
	Unknown	0x3E	
	Invalid	0x3F	
ChargeEndTimeYear2_St	-	-	Charge End Time Year
	<range>	0x0-0x7F	2017 - 2144 (resolution of 1 year, offset of 2017)
ChargeEndTimeYear_St	-	-	Charge End Time Year
	<range>	0x0-0x1E	2010 - 2040 (resolution of 1 year, offset of 2010)
	Invalid	0x1F	
ChargeEndTimeMonth_St	-	-	Charge End Time Month
	<range>	0x0-0xE	1 - 15 (resolution of 1 month, offset of 1)
	Invalid	0xF	
ChargeEndTimeDay_St	-	-	Charge End Time Day
	<range>	0x0-0x1E	1 – 31 days (resolution of 1 day, offset of 1)
	Invalid	0xF	
ChargeEndTimeHour_St	-	-	Charge End Time Hour
	<range>	0x0-0x1D	0 – 29 hours (resolution of 1 hour)



	Unknown	0xE	
	Invalid	0xF	
ChargeEndTimeMin_St	-	-	Charge End Time Minute
	<range>	0x0- 0x3D	0 – 61 minutes (resolution of 1 min)
	Unknown	0x3E	
	Invalid	0x3F	

2.9.2 MD-REQ-346075/A-TripData

This logical API is used to send trip data from the off-board cloud services to the vehicle and vice versa.

Method Type		On-Change			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Response (_Rsp)					
R	TripData	bytes	Contains the following objects: <ul style="list-style-type: none">- TripInfo- Waypoints- TripStartCoordinates- TripEndCoordinates Refer to the TripPlanner.yaml file for payload name and definition	-	Relevant Trip Data for a planned trip

2.9.3 MD-REQ-346073/A-RouteData

This logical API is used to send route data from the off-board cloud services to the vehicle..

Method Type		On-Change			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Response (_Rsp)					
R	RouteData	bytes	Refer to Nav Vendor documentation for payload name and definition	-	Route data generated by EV Trip Planner Off-Board Server2

2.9.4 MD-REQ-346687/A-getTripData_Rsp

This logical API is used to request to get trip data which is generated by the off-board cloud services.

Method Type		On-Change			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description

**Response (_Rsp)**

R	getTripData_Rsp	bytes	Refer to the TripPlanner.yaml file for operation name and definition	-	Response to the Get Trip Data request
---	-----------------	-------	--	---	---------------------------------------

2.9.5 MD-REQ-346688/A-putTripData_Rsp

This logical API is used to send trip data to the off-board cloud services.

Method Type		On-Change			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Response (_Rsp)					
R	putTripData_Rsp	bytes	Refer to the TripPlanner.yaml file for operation name and definition	-	Response to the Put Trip Data request

2.9.6 MD-REQ-346686/A-getRouteData_Rsp

This logical API is used to get route data from the off-board cloud services to the vehicle.

Method Type		On-Change			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Response (_Rsp)					
R	getRouteData_Rsp	bytes	Refer to Nav Vendor documentation for operation name and definition	-	Response to the Get Route Data request

2.10 IIR-REQ-345903/A-EVTripPlannerInterfaceClient _Tx**2.10.1 MD-REQ-345904/A-DistanceToStopover_St**

Message Type: Status

This signal is used to indicate the distance to the next stopover location

Name	Literals	Value	Description
DistanceToStopover_St	-	-	A status of the distance to the next stopover location sent to the EVTripPlannerStatusClient
	<Range>	0x0000 – 0xFFFFE	Distance in tenths of a kilometer (0 – 65533)
	Faulty	0xFFFF	

2.10.2 MD-REQ-345905/A-StopoverType_St

Message Type: Status



The signal is used to provide the status of the type of the next stopover location

Name	Literals	Value	Description
StopoverType_St			Indicates the type of stopover location to the EVTripPlannerStatusClient
	Null	0x0	
	NotValid	0x1	
	Final Destination	0x2	
	Charge Station	0x3	
	Home	0x4	
	Work	0x5	
	NotUsed1	0x6	
	NotUsed2	0x7	

2.10.3 MD-REQ-346074/A-getTripData_Rq

This logical API is used to request to get trip data which is generated by the off-board cloud services.

Method Type		On-Change			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request (_Rq)					
R	getTripData_Rq	bytes	Refer to the TripPlanner.yaml file for operation name and definition	-	Request to Get Trip Data generated by EV Trip Planner Off-Board Server1

2.10.4 MD-REQ-346351/A-putTripData_Rq

This logical API is used to send trip data to the off-board cloud services.

Method Type		On-Change			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request (_Rq)					
R	putTripData_Rq	bytes	Refer to the TripPlanner.yaml file for operation name and definition	-	Request to send updated Trip Data to the EV Trip Planner Off-Board Server1

**2.10.5 MD-REQ-346075/A-TripData**

This logical API is used to send trip data from the off-board cloud services to the vehicle and vice versa.

Method Type		On-Change			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Response (_Rsp)					
R	TripData	bytes	Contains the following objects: <ul style="list-style-type: none">- TripInfo- Waypoints- TripStartCoordinates- TripEndCoordinates Refer to the TripPlanner.yaml file for payload name and definition	-	Relevant Trip Data for a planned trip

2.10.6 MD-REQ-346076/A-getRouteData_Rq

This logical API is used to get route data from the off-board cloud services to the vehicle.

Method Type		On-Change			
QoS Level		Default			
Retained		No			
R/O	Name	Type	Literals	Value	Description
Request (_Rq)					
R	getRouteData_Rq	bytes	Refer to Nav Vendor documentation for operation name and definition	-	Request to get route data generated by EV Trip Planner Off-Board Server2



3 General Requirements

3.1 EVTP-REQ-346353/A-Utilization of Wireless Interface Router framework

The EV Trip Planner Interface client shall utilize the Wireless Interface Router framework to facilitate the establishment of connectivity with any off-board services. Please see the Wireless Interface Router SPSS for further details on this framework.

3.2 EVTP-REQ-345919/A-Utilization of IP Pass-Through Framework

The EV Trip Planner Interface Client shall leverage the existing IP Pass-Through framework to issue a token request and facilitate communication with the EV Trip Planner OffBoard Servers. Refer to the latest IP Pass-Through SPSS for further details on this framework.

3.3 EVTP-REQ-346695/A-Response Codes from the EV Trip Planner OffBoard Servers

Responses to requests sent to the EV Trip Planner OffBoard Servers from the EV Trip Planner Interface Client may contain error codes if the request cannot be honored. If the request can be honored, a success response code will be sent. Refer to the TripPlanner.yaml file definition on the response code types.

3.4 EVTP-REQ-346982/A-Delete Token and Trip Data Upon a Master Reset

When a Master Reset has occurred, the EV Trip Planner Interface Client shall delete any stored IPPT tokens and trip/route data.

3.5 EVTP-REQ-346354/A-EV Trip Planner CCS Requirement

The EV Trip Planner feature code is TRIP and has a CCS entity of TripPlanner (ID 57, Type 1). The feature is impacted by CCS as follows in the requirement and table below.

If the bUAllow bit for the meta entity Trip Planner is disabled, then only the connected elements of the EV Trip Planner feature should be disabled and cease communicating, but the embedded or "On-Board" elements of the EV Trip Planner featured should continue to be functional.

If the bSAllow bit for the meta entity Trip Planner is disabled, then the EV Trip Planner feature as a whole should be disabled and cease communicating.

	Vehicle Connectivity disabled	Location disabled	Vehicle Data disabled	Driving Characteristics disabled	Remote Command & Control disabled
EV Trip Planner	On-Board functionality only	On-Board functionality only	On-Board functionality only	On-Board functionality only	N/A
Update Off-Board Trips	Disable Function	Disable Function	Disable Function	Disable Function	N/A
Update On-Board Trips	Enable Function	Enable Function	Enable Function	Enable Function	N/A
Create Off-Board Trips	Disable Function	Disable Function	Disable Function	Disable Function	N/A
Create On-Board Trips	Enable Function	Enable Function	Enable Function	Enable Function	N/A

Please refer to the Customer Connectivity Settings Manager SPSS for further detail.



4 Functional Definition

4.1 EVTP-FUN-REQ-345906/A-Load New Trips from Cloud

4.1.1 Requirements

4.1.1.1 EVTP-REQ-345907/A-Loading Trips from the OffBoard Server

Upon start-up the EV Trip Planner Interface Client shall query the EV Trip Planner OffBoard Server1 for the latest Trip Info using the getTripData logical API. Upon receiving and processing the TripData payload, the EV Trip Planner Interface Client shall refresh the FordPass Trips folder and update the Trips view to prompt the user that a new trip(s) has been downloaded.

4.1.1.2 EVTP-REQ-345908/A-Generating a Route from the OffBoard Server

Upon user confirmation of the trip, EV Trip Planner Interface Client shall proceed to query the EV Trip Planner OffBoard Server2 for a route to be generated using the getRouteData logical API. Upon receiving and processing the RouteData payload, the EV Trip Planner Interface Client shall update the Route view to inform the user that a route has been downloaded.

4.1.1.3 EVTP-REQ-345909/A-Vehicle Data Processing

The EV Trip Planner Interface Client shall process all signals from the VehicleData_St method as needed for the purposes of route generation and to support the Navigation Application HMI.

4.1.1.4 EVTP-REQ-345910/A-Charge Connector Type

The EV Trip Planner Interface Client shall determine the charge connector type according to the DID value defined in the latest Infotainment Diagnostic Specification.

4.1.1.5 EVTP-REQ-345918/A-On-Board Route Generation When Connectivity is Lost

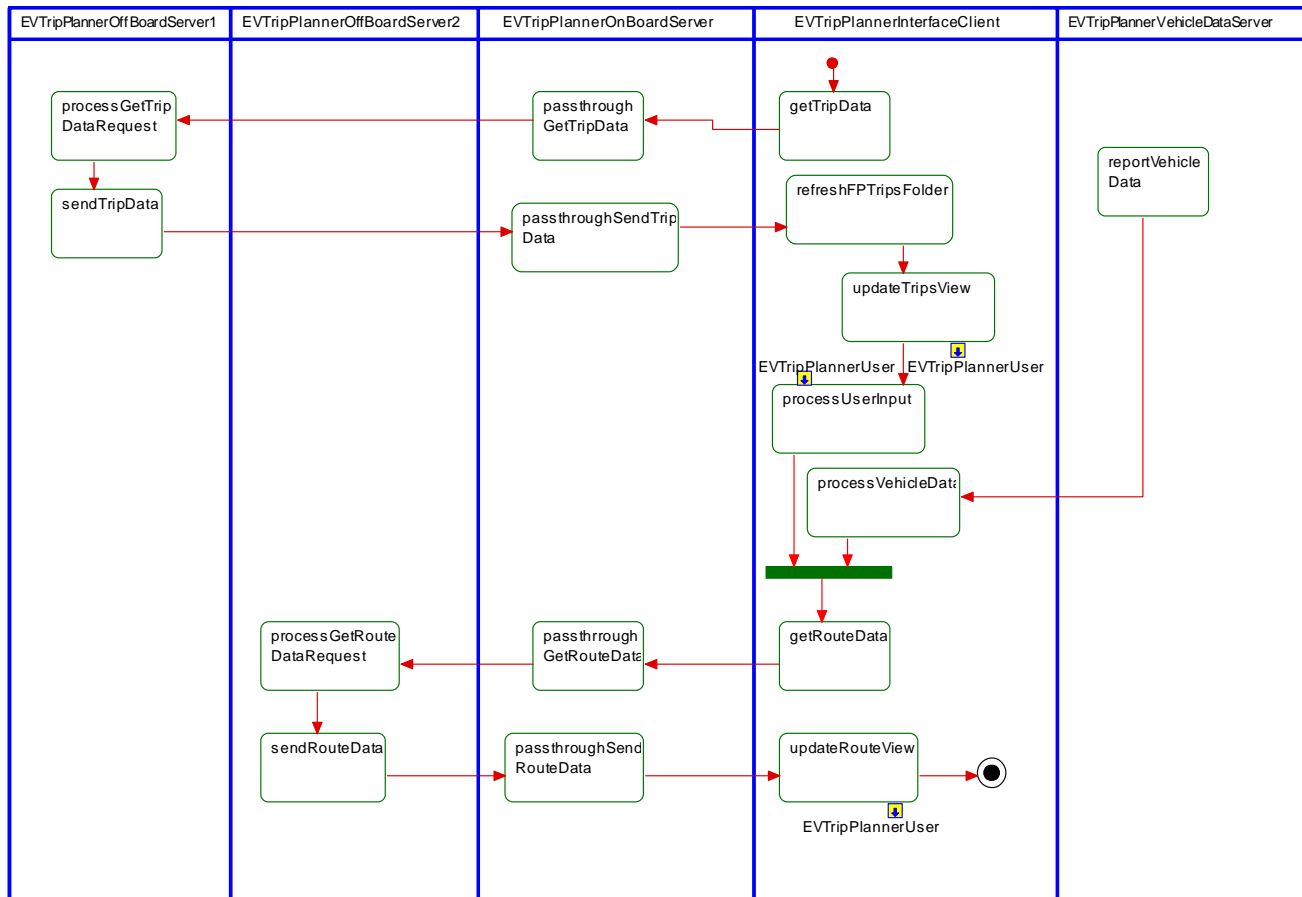
When the vehicle has no connectivity, the EV Trip Planner Interface Client shall generate a route on-board using the embedded navigation application.



4.1.2 White Box View

4.1.2.1 EVTP-ACT-REQ-345912/A-Load New Trips from Cloud

Activity Diagram





4.1.2.2 EVTP-SD-REQ-345913/A-Load New Trips from Cloud

Scenarios

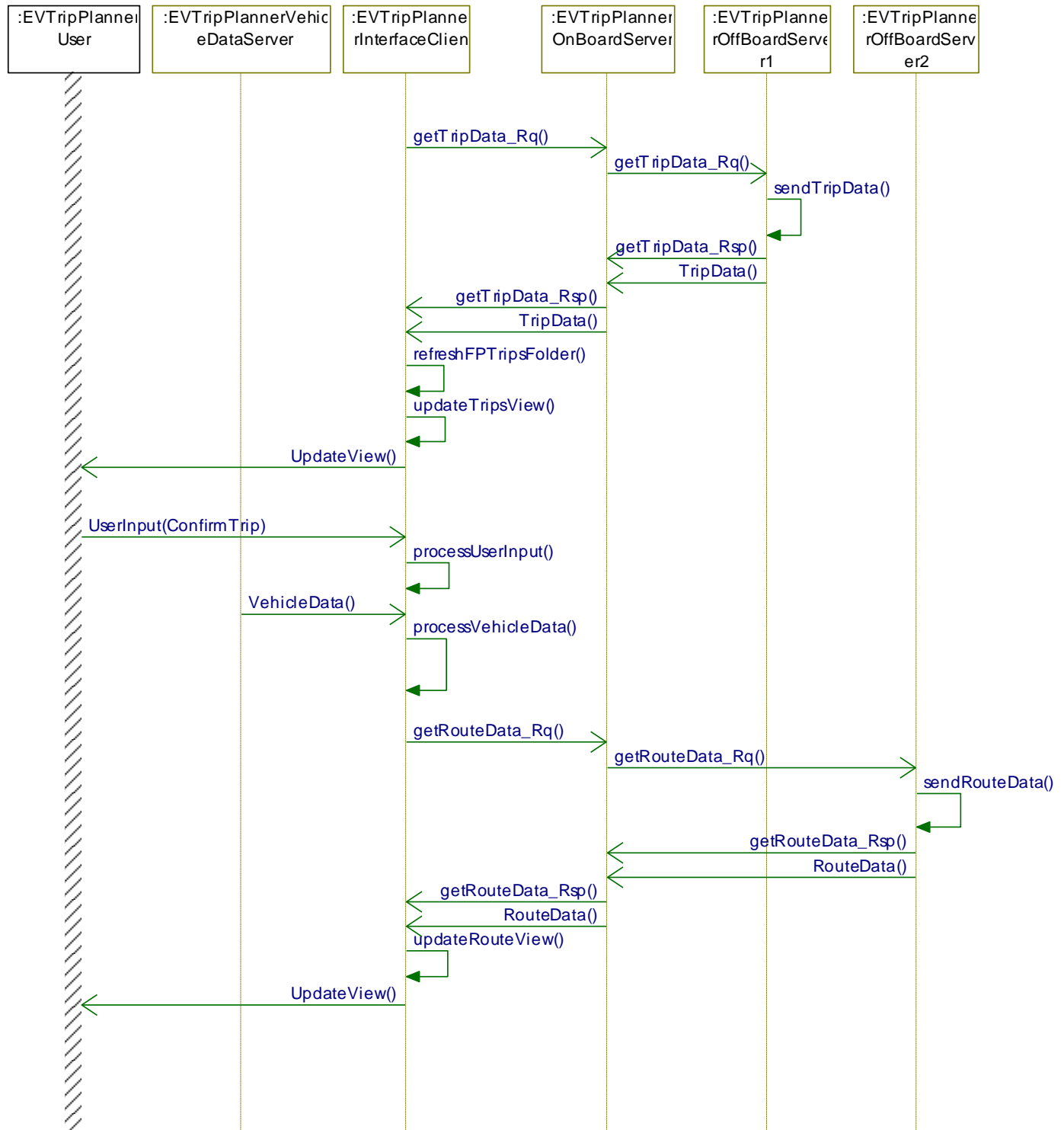
Normal Usage

Constraints

Pre-condition

Post-condition

Sequence Diagram





4.2 EVTP-FUN-REQ-345931/A-Update Existing Trip In-Vehicle

4.2.1 Requirements

4.2.1.1 EVTP-REQ-345932/A-Update Existing Trip Data for a Trip Update Event

The EV Trip Planner Interface Client shall send Trip Data to OffBoard Server1 using the putTripData logical API when a valid Trip Update Event is detected.

4.2.1.2 EVTP-REQ-345933/A-Trip Update Event

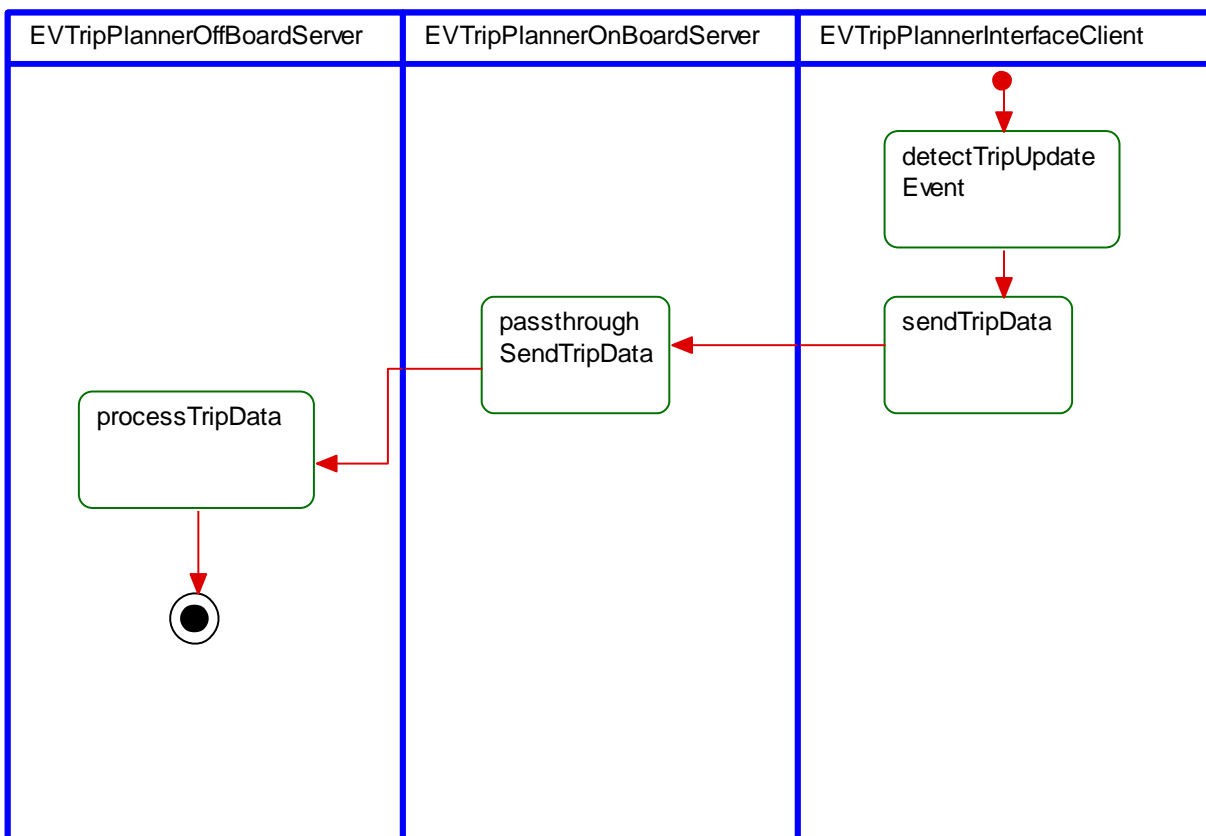
A Trip Update Event is defined as any of the following conditions:

- A new waypoint is added by the user
- A change in the status of an existing waypoint
- A change in the destination
- A change in the status of the destination
- A change in the status of the trip (e.g. completed or cancelled)
- A change in the status of the vehicle ignition state from Run/Accessory to Off

4.2.2 White Box View

4.2.2.1 EVTP-ACT-REQ-345937/A-Update Existing Trip In-Vehicle

Activity Diagram





4.2.2.2 EVTP-SD-REQ-345938/A-Update Existing Trip In-Vehicle

Scenarios

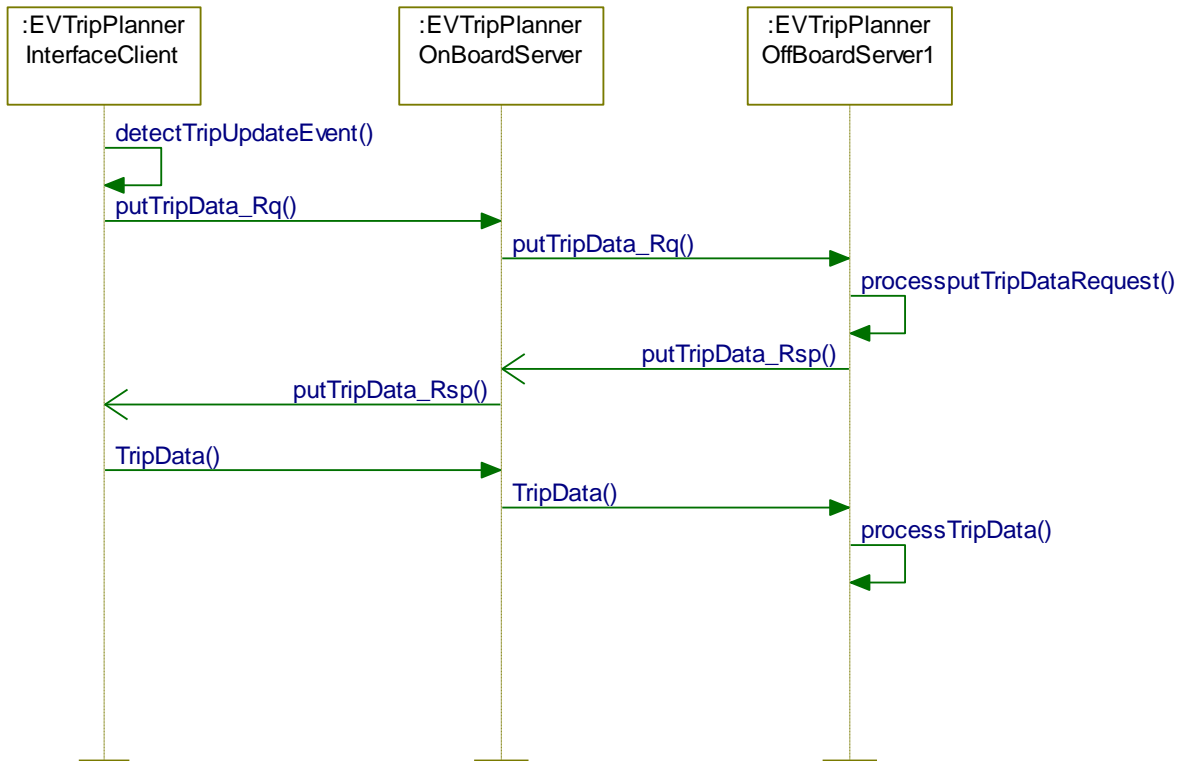
Normal Usage

Constraints

Pre-condition

Post-condition

Sequence Diagram



4.3 EVTP-FUN-REQ-345923/A-Create New Trip In-Vehicle

4.3.1 Requirements

4.3.1.1 REQ-345925/A-Creating New Trips from the In-Vehicle HMI

The EV Trip Planner Client shall contain an HMI setting in the Navigation Application to allow the user to share any created Trips using the in-vehicle HMI to be shared with the EV Trip Planner OffBoard Server1 so that they can be accessible on the EV Trip Planner OffBoard Client.

Once a user has modified the setting to allow sharing of Trip data, any Trips that are created will require a route to be generated, which is requested from the EV Trip Planner OffBoard Server2 via the getRouteData_Rq. Once a route has been downloaded and a Trip has been created successfully, the data from this Trip shall be saved to the FordPassTrips Folder, and then uploaded to the EV Trip Planner OffBoard Server1 via the putTripData_Rq.

4.3.1.2 EVTP-REQ-345926/A-Low-Charge Alert Routing Prompt

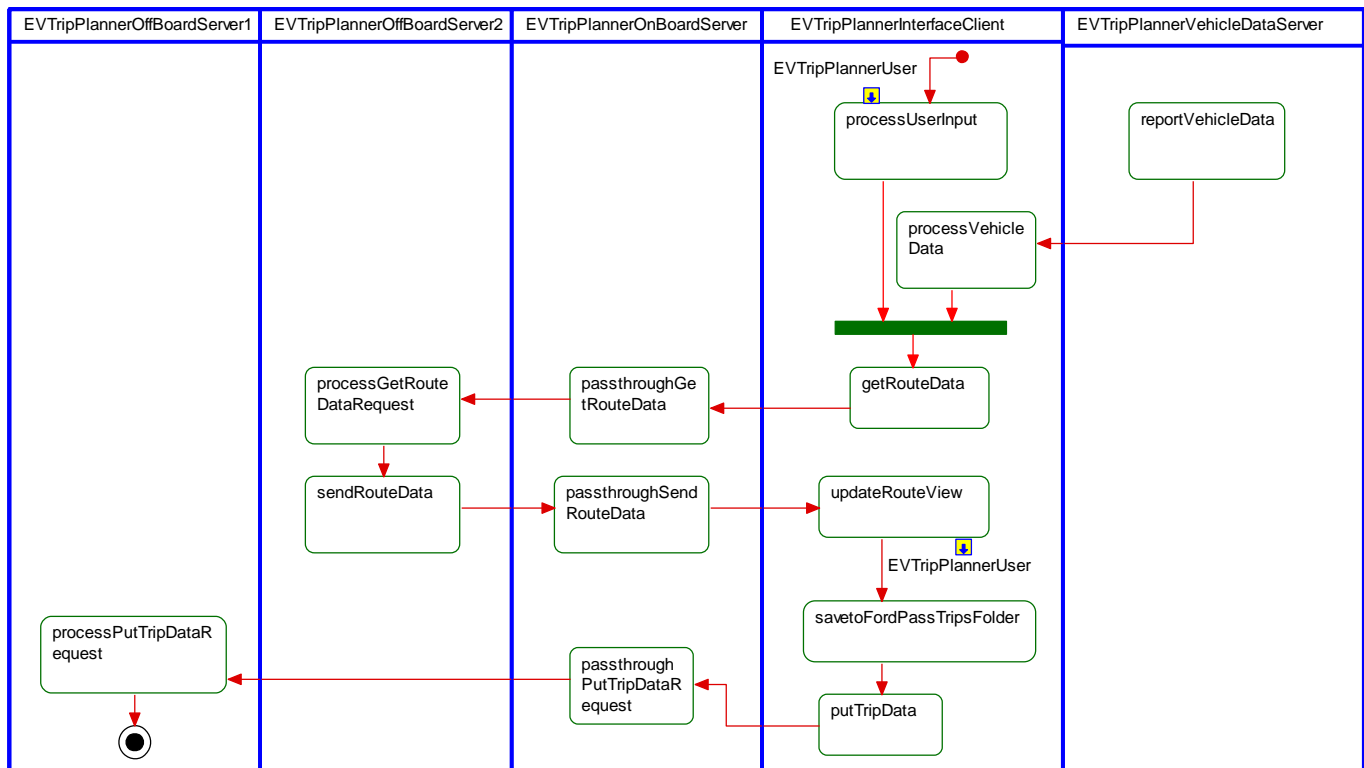
When the EV Trip Planner Interface Client has detected a low-charge alert condition and informed the user accordingly, it shall also prompt the user with the possibility to begin the route creation process for the purposes of generating a route to the nearest charge station.



4.3.2 White Box View

4.3.2.1 EVTP-ACT-REQ-345929/A-Create New Trip In-Vehicle

Activity Diagram



4.3.2.2 EVTP-SD-REQ-345930/A-Create New Trip In-Vehicle

Scenarios

Normal Usage

User creates a new Trip using the Navigation Application HMI

Constraints

Pre-condition

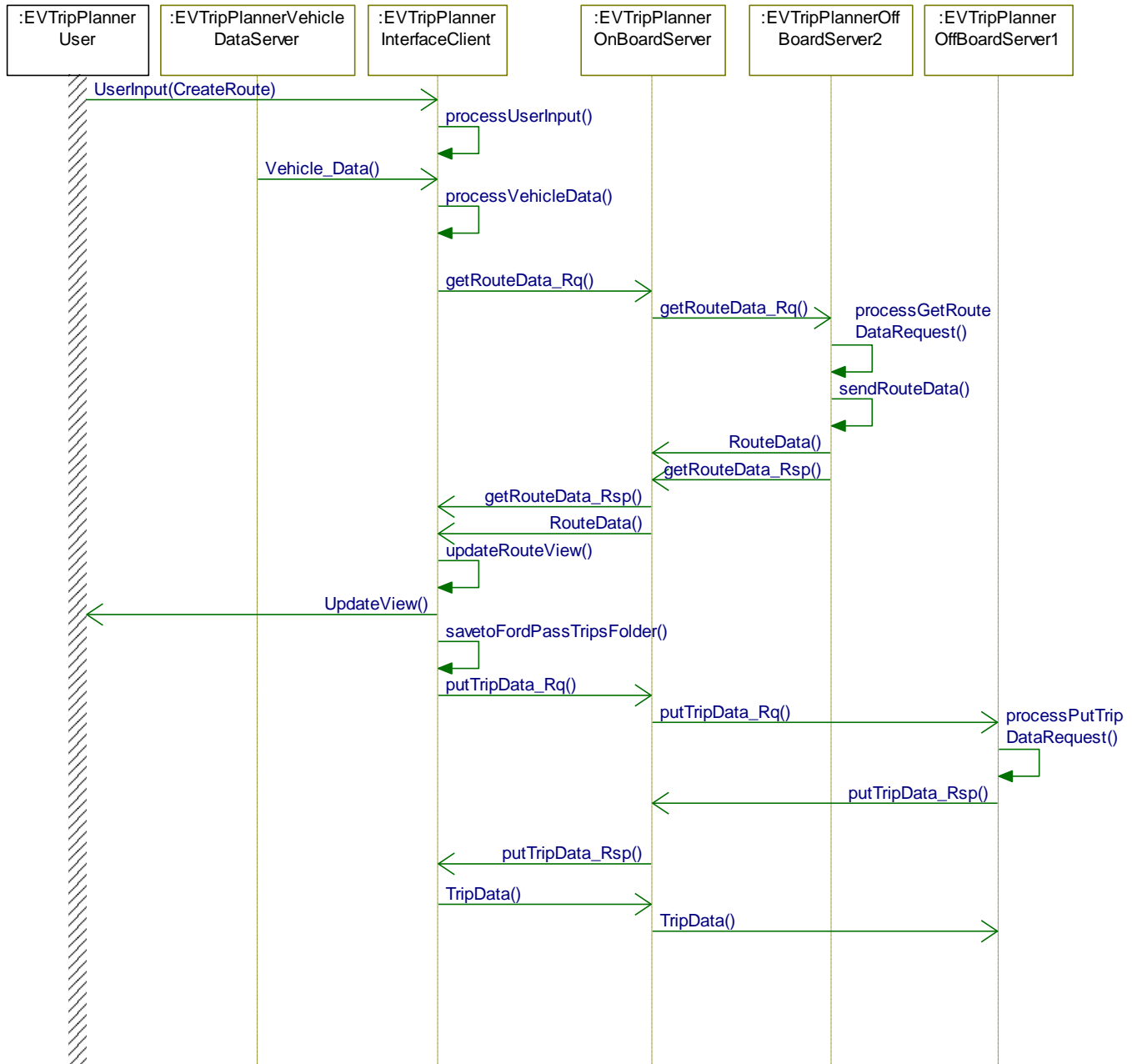
The HMI setting in the Navigation Application to share trips with the EV Trip Planner OffBoard Client is set to ON

Post-condition

The data for the newly created Trip has been uploaded to the EV Trip Planner OffBoard Server1



Sequence Diagram



4.4 EVTP-FUN-REQ-345939/A-Update Stopover Status

4.4.1 Requirements

4.4.1.1 EVTP-REQ-345940/A-Stopover Distance Calculation

The EV Trip Planner Interface Client shall calculate the distance to the next stopover location of the active route and update the status of the DistanceToStopover_St signal periodically.

4.4.1.2 EVTP-REQ-345941/A-Stopover Location Type

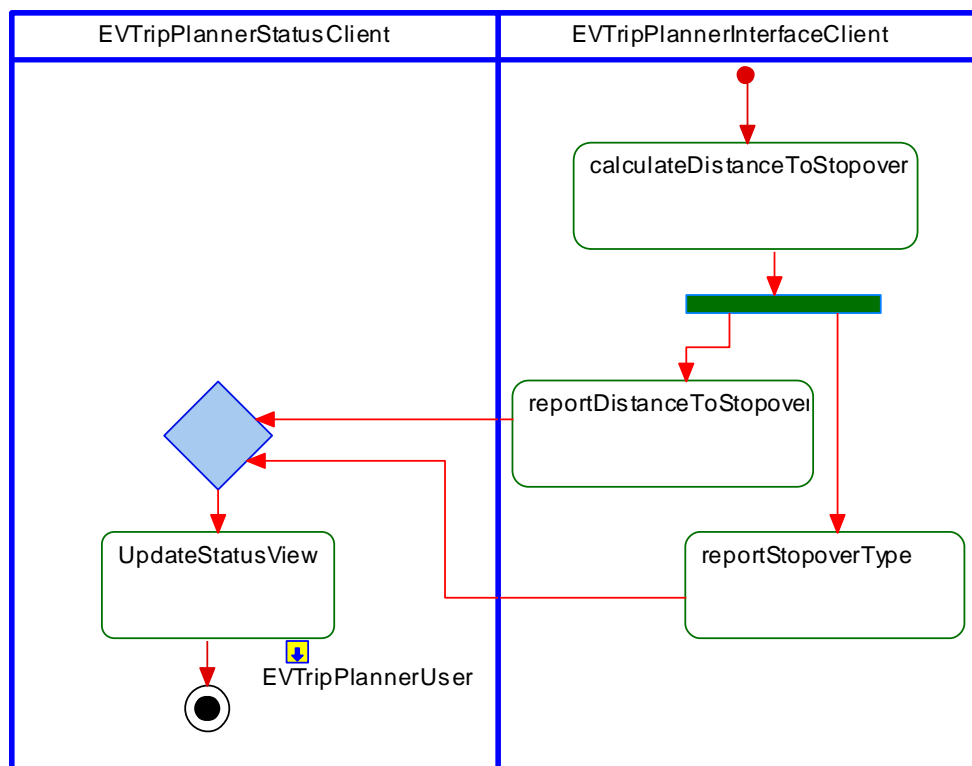
The EV Trip Planner Interface Client shall set the StopoverType_St signal to the type of the next stopover location of the active route.



4.4.2 White Box View

4.4.2.1 EVTP-ACT-REQ-345945/A-Update Stopover Status

Activity Diagram



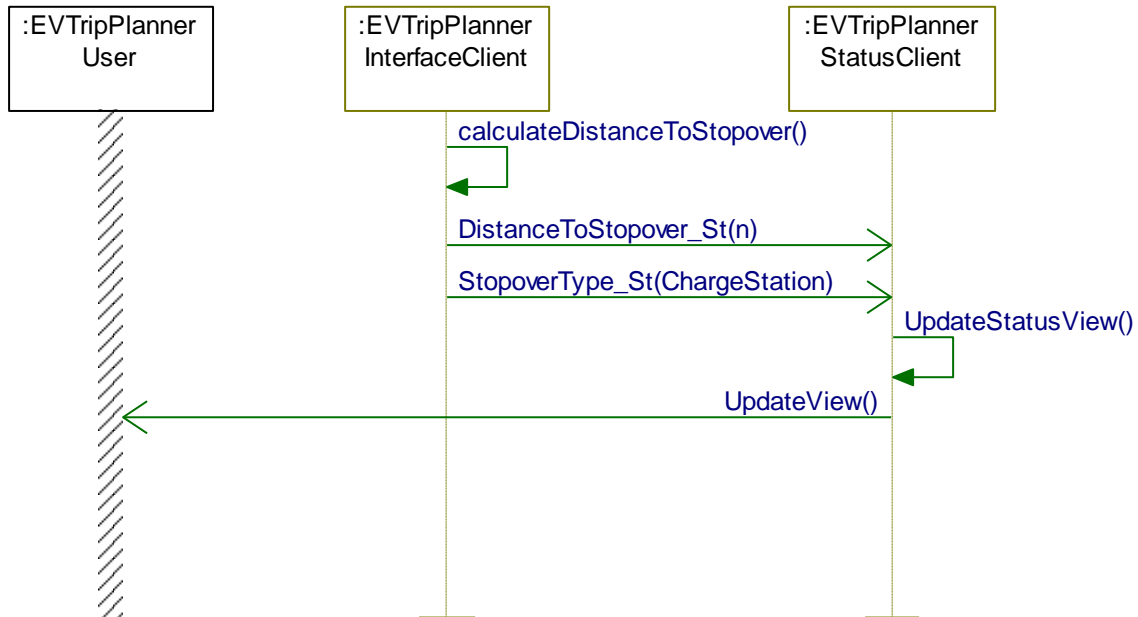
**4.4.2.2 EVTP-SD-REQ-345946/A-Update Stopover Status****Scenarios**

Normal Usage

Constraints

Pre-condition

Post-condition

Sequence Diagram



5 Appendix: Reference Documents

Reference #	Document Title
1	TripPlanner.yaml
2	EV Trip Planner HLD.doc
3	IP Pass Through Client SPSS v1.1 September 4, 2018.doc
4	Infotainment Diagnostic Specification
5	Wireless Interface Router Client1 SPSS v1.2 May 22, 2018.doc
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	