

**Research & Vehicle Technology**

**“Infotainment Systems Product Development”**

**Feature – IP Pass Through Client**

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

Version 1.2

**UNCONTROLLED COPY IF PRINTED**

**Version Date: September 4, 2018**

**FORD CONFIDENTIALF**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Ver** | **Notes** | |
| **April 3, 2018** | **1.0** | **Initial Release** |  |
|  |  |  |  |
| **September 4, 2018** | **1.1** | **Updated Release** |  |
|  | IPPT-SD-REQ-304274/B-Application IPPT Request | | MBORREL4: Updated diagram |
|  | STR-523925/B-Requirements | | MBORREL4: Added REQ-322262, REQ-322263 |
|  | IPPT-REQ-304250/B-IPPT FTCP response | | MBORREL4: Updated content (added Error codes table) |
|  | IPPT-REQ-304251/B-IPPT FTCP response parameters | | MBORREL4: Updated table |
|  | IPPT-REQ-322262/A-Token revocation request | | MBORREL4: New req. |
|  | IPPT-REQ-322263/A-Token revocation response | | MBORREL4: New req. |

**Table of Contents**

[Revision History 2](#_Toc523824689)

[1 Overview 4](#_Toc523824690)

[1.1 Terminology and Abbreviations 4](#_Toc523824691)

[2 Architectural Design 5](#_Toc523824692)

[2.1 IPPT-CLD-REQ-304022/A-IP Pass Through Server 5](#_Toc523824693)

[2.2 IPPT-CLD-REQ-304020/A-IP Pass Through Client 5](#_Toc523824694)

[2.3 IPPT-CLD-REQ-304238/A-IP Pass Through OffBoard Client 5](#_Toc523824695)

[2.4 Physical Mapping of Classes 5](#_Toc523824696)

[2.5 IPPTClient Interface 6](#_Toc523824697)

[2.5.1 IPPT-IIR-REQ-304023/A-IPPTClientInterface\_Tx 6](#_Toc523824698)

[2.5.2 IPPT-IIR-REQ-304024/A-IPPTClientInterface\_Rx 7](#_Toc523824699)

[3 Functional Definition 8](#_Toc523824700)

[3.1 IPPT-FUN-REQ-304265/A-Local Handler 8](#_Toc523824701)

[3.1.1 Requirements 8](#_Toc523824702)

[3.1.2 Use Cases 8](#_Toc523824703)

[3.1.3 White Box View 9](#_Toc523824704)

[3.2 IPPT-FUN-REQ-304260/A-Central Handler 11](#_Toc523824705)

[3.2.1 Requirements 11](#_Toc523824706)

[3.2.2 Use Cases 13](#_Toc523824707)

[3.2.3 White Box View 13](#_Toc523824708)

[3.3 IPPT-FUN-REQ-304267/A-IPPT Performance 14](#_Toc523824709)

[3.3.1 Requirements 14](#_Toc523824710)

[3.3.2 Use Cases 14](#_Toc523824711)

[3.3.3 White Box View 14](#_Toc523824712)

[3.4 IPPT-FUN-REQ-304268/A-IPPT Configuration 15](#_Toc523824713)

[3.4.1 Requirements 15](#_Toc523824714)

[3.4.2 Use Cases 15](#_Toc523824715)

[3.4.3 White Box View 15](#_Toc523824716)

[4 Appendix: Reference Documents 16](#_Toc523824717)

# Overview

IP Based Pass through works with Wireless Interface Router in the ECG, TCU, SYNC and with an authorization/policy management application in the Ford IPPT Cloud. These will be the key components to provide session authorization, management, and authentication for this service. Once implemented, this service can be used by any Ethernet connected service within Cluster, ADAS, VDS etc. that requires an off board connection to a 1st, 2nd, or third party destination address.

## Terminology and Abbreviations

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

| **Term** | **Description** |
| --- | --- |
| CAN | Controller Area Network |
| 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 |
| IPPT | IP Pass Through |
| SDN | Software Delivery Network |
| SoA | Service Oriented Architecture |
| TCU | Telematics Control Unit |
| WIR | Wireless Interface Router |
| WLAN | Wireless Local Area Network |

# Architectural Design

## IPPT-CLD-REQ-304022/A-IP Pass Through Server

The IP Pass Through Server (IPPTServer) is responsible for the tasks listed below:

* Interfaces with IPPTClient to receive application requests
* Interfaces with IPPTOffBoardClient to request and receive tokens
* Performs transaction logging for diagnostics purposes

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

## IPPT-CLD-REQ-304020/A-IP Pass Through Client

The IP Pass Through Client (IPPTClient) is responsible for the tasks listed below:

* Interfaces with applications to receive requests and provide responses
* Interfaces with IPPTServer to receive token response from IPPTOffBoardClient

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

## IPPT-CLD-REQ-304238/A-IP Pass Through OffBoard Client

The IP Pass Through OffBoard Client (IPPTOffBoardClient) is responsible for the tasks listed below:

* Interfaces with IPPTServer to receive token requests
* Performs policy validations
* Interfaces with 3rd party to receive the authentication tokens

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

## Physical Mapping of Classes

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

|  |  |
| --- | --- |
| **Logical Class** | **Physical Module (ECU)** |
| IPPTServer | ECG |
| IPPTClient | ECG, SYNC, TCU |
| IPPTOffBoardClient | Ford Cloud or 3rd Party |

## IPPTClient Interface

### IPPT-IIR-REQ-304023/A-IPPTClientInterface\_Tx

The IPPTClientInterface\_Tx represents all the IPPT feature related signals sent by the IPPTClientInterface object. The below table represents the mapping of the logical signal names (as described in this specification) to the global GSDB signal names.

|  |  |  |
| --- | --- | --- |
| **Logical Signal Name** | **Parameter Name** | **GSDB Signal Name** |
| AutosarNM | Control | TCU\_AutoSarNMControl  APIM\_AutoSarNMControl |
| NodeID | TCU\_AutoSarNMNodeId  APIM\_AutoSarNMNodeId |
| NMReserved1 | TCU\_AutoSarNMReserved1  APIM \_AutoSarNMReserved1 |
| NMReserved2 | TCU\_AutoSarNMReserved2  APIM \_AutoSarNMReserved2 |
| NMReserved3 | TCU\_AutoSarNMReserved3  APIM \_AutoSarNMReserved3 |
| NMReserved4 | TCU\_AutoSarNMReserved4  APIM\_AutoSarNMReserved4 |
| GWNMProxy | TCU\_GWNMProxy  APIM\_GWNMProxy |
| GWOnBoardTester | TCU\_GWOnBoardTester  APIM\_GWOnBoardTester |

Note:  GSDB signal names are reference only.  The Global Signal Database (GSDB) is the master for all signals. If there is a conflict, bring to the module D&R’s attention.

#### 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 |  |

### IPPT-IIR-REQ-304024/A-IPPTClientInterface\_Rx

The IPPTClientInterface\_Rx represents all the IPPT feature related signals received by the IPPTClientInterface object. The below table represents the mapping of the logical signal names (as described in this specification) to the global GSDB signal names.

|  |  |  |
| --- | --- | --- |
| **Logical Signal Name** | **Parameter Name** | **GSDB Signal Name** |
|  |  |  |
|  |  |  |
|  |  |  |

Note:  GSDB signal names are reference only.  The Global Signal Database (GSDB) is the master for all signals. If there is a conflict, bring to the module D&R’s attention.

# Functional Definition

## IPPT-FUN-REQ-304265/A-Local Handler

### Requirements

#### IPPT-REQ-304266/A-Local Handler in IPPTClient

Each IPPTClient shall contain a Local Handler.

#### IPPT-REQ-304239/A-IPPTClient receive request

The IPPTClient Local Handler shall be able to receive requests from application

#### IPPT-REQ-304241/A-IPPTClient passing request to IPPT Server

The IPPTClient Local Handler shall pass Token requests to the IPPTServer Central Handler

#### IPPT-REQ-304242/A-IPPTClient passing response back to application

The IPPTClient Local Handler shall pass Token response received from IPPTServer Central Handler to application

#### IPPT-REQ-304243/A-IPPTClient request validation - Error

The IPPTClient Local Handler shall validate the application request and shall return an error to the application if there is any error in the token request parameter

#### IPPT-REQ-304244/A-IPPTClient request validation – Unique ID

The IPPTClient Local Handler shall validate the application request and shall return an unique ID to the application if there is no error in the token request parameter

The requesting application shall be able to track the request via the unique ID parameter.

### Use Cases

#### IPPT-UC-REQ-304269/A-Token Request From Application

|  |  |
| --- | --- |
| **Actors** | WIR, IPPTServer, IPPTClient, IPPTOffBoardClient |
| **Pre-conditions** | 1. Application sends token request to IPPT Client 2. The IPPTClient generates unique ID and responds to application 3. The IPPTServer receives the request from IPPTClient and sends the request to IPPTOffBoardClient via FTCP 4. The IPPTOffBoardClient sends the request to end point and receives the token details 5. The IPPTOffBoardClient sends the response as a FTCP message 6. IPPTServer receives the FTCP message, process the same 7. IPPTServer with IPPTClient sends the token response information to the requesting application |
| **Scenario Description** | Application requests token |
| **Post-conditions** | Application uses the token provided by IPPT to access end point URL |
| **Interfaces** |  |
| **Notes** | Exception use cases  1 - Token request invalid parameters  2 - No token received from IPPTOffBoardClient |

#### IPPT-UC-REQ-304270/A-Exception 1 – Token Request Invalid Parameters

|  |  |
| --- | --- |
| **Actors** | WIR, IPPTClient |
| **Pre-conditions** | 1. Application sends a request to IPPTClient 2. IPPTClient returns an error back to the requesting application if there is invalid parameter |
| **Scenario Description** | Application sends a token request with invalid parameters |
| **Post-conditions** | The application shall process the error message and resend another request |
| **Interfaces** |  |
| **Notes** |  |

#### IPPT-UC-REQ-304271/A-Exception 2 – No Token Received From IPPTOffBoardClient

|  |  |
| --- | --- |
| **Actors** | WIR, IPPTServer, IPPTClient, IPPTOffBoardClient |
| **Pre-conditions** | 1. Application sends a request to IPPTClient 2. The IPPTClient generates unique ID and responds to application 3. The IPPTServer receives the request from IPPTClient and sends the request to IPPTOffBoardClient via FTCP 4. The IPPTOffBoardClient sends the request to end point and receives no response after all retries 5. The IPPTOffBoardClient notifies this back to the vehicle via an FTCP message 6. The IPPTClient notifies the application of the lack of response |
| **Scenario Description** | Application requests token but end point (IPPTOffBoardClient) fails to respond |
| **Post-conditions** | A defect log is created for the end point URL |
| **Interfaces** |  |
| **Notes** |  |

### White Box View

#### Sequence Diagrams

##### IPPT-SD-REQ-304274/B-Application IPPT Request

Constraints

Pre-Condition

Vehicle is ON

Scenarios

Normal Usage

Application requests token

Post-Condition

Application uses the token provided by IPPT to access end point URL

Sequence Diagram



## IPPT-FUN-REQ-304260/A-Central Handler

### Requirements

#### IPPT-REQ-304314/A-Central Handler in IPPTServer

The IPPTServer shall contain the Central Handler.

#### IPPT-REQ-304245/A-IPPTServer/Client interaction – request

The IPPTServer Central Handler shall be able to interact with all IPPTClient Local Handlers and shall be able to receive and process token requests.

The IPPTServer Central Handler design shall be flexible to accommodate requests from additional IPPTClient Local Handlers in the future as well.

#### IPPT-REQ-304246/A-IPPTServer/Client interaction – response

The IPPTServer Central Handler shall be able interact with all IPPTClients and shall be able to send token response received from IPPTOffBoardClient to the IPPTClient Local Handlers.

#### IPPT-REQ-304247/A-IPPT FTCP request

The IPPTServer Central Handler shall be able to compose an FTCP message and request for token from the IPPTOffBoardClient

For more details please refer the FTCP protofile

#### IPPT-REQ-304248/A-IPPT FTCP request parameters

The IPPTServer Central Handler shall be able to fill FTCP token request based on the information it received from the application

|  |  |
| --- | --- |
| **Request parameters** | **Data type** |
| Feature ID | String |
| Unique ID | String |
| ECU ID | Integer |
| Policy validation | Boolean |
| Config key value pair  URL key value is a string | String array |
| Scope | String array |

For more details please refer the FTCP protofile

#### IPPT-REQ-304249/A-IPPT FTCP request ECU ID

The IPPTServer Central Handler shall be able to fill the appropriate ECU ID based on which ECU the request originated

#### IPPT-REQ-304250/B-IPPT FTCP response

The IPPTServer Central Handler shall be able to decode a token response FTCP message and send the token response to the requesting application. In case of failure, IPPTServer Central Handler shall send an appropriate error code to the application.

|  |  |
| --- | --- |
| **Token Error codes** | **Error message/ Description** |
| PAYLOAD\_ERROR | Mandatory Token fields are missing |
| INVALID\_URL | URL in the Config is not located |
| SUBSCRIPTION\_EXPIRED | Subscription got expired |
| NOT\_ENROLLED | Feature Not enrolled |
| INVALID\_CSS | Invalid CCS |
| EXPIRED\_SUBSCRIPTION\_  AND\_NOT\_ENROLLED | Subscription got expired and Not enrolled |
| EXPIRED\_SUBSCRIPTION\_  AND\_INVALID\_CCS | Subscription expiration and Invalid CCS |
| NOT\_ENROLLED\_AND\_  INVALID\_CCS | Not enrolled and Invalid CCS |
| EXPIRED\_SUBSCRIPTION\_  NOT\_ENROLLED\_AND\_ INVALID\_CCS | Subscription expiration, Not enrolled and Invalid CCS |
| POLICY\_VALIDATION\_SYSTEM\_DOWN | CVFMA/SuMo/CCS down – Unable to check subscription, enrollment and/or CCS |
| PROVIDER\_SYSTEM\_DOWN | Content Provider system down |
| IPPT\_CLOUD\_SYSTEM-DOWN | IPPTOffBoardClient system down |
| PROVIDER\_TOKEN\_REFUSAL | Refusal from content provider for a Token |
| OTHER\_ERRORS | All other Errors - will be defined later |

For more details about FTCP message please refer the FTCP protofile

#### IPPT-REQ-304251/B-IPPT FTCP response parameters

|  |  |
| --- | --- |
| **Response parameters** | **Data type** |
| Access Token | String |
| Expires\_in | Integer |
| Token Type | String |
| Refresh Token | String |
| Unique ID | String |
| Error (optional) | String |
| Error Description (optional) | String |
| Scope (optional) | String array |
| Redirect URL(optional) | String |
| Refresh Token\_expires in (optional) | Integer |

#### IPPT-REQ-304252/A-IPPTOffBoardClient response correlation with application request

Both IPPTServer Central Handler and IPPTClient Local Handler shall be able to correlate an application request and appropriate IPPTOffBoardClient response by means of using the unique ID which acts as a transaction ID.

#### IPPT-REQ-304253/A-IPPTServer Logging

The IPPTServer shall log all requests and response for a period of IPPT\_LOGS\_STORAGE\_TIME

#### IPPT-REQ-304254/A-On demand diagnostics logs

Upon request from IPPTOffBoardClient the IPPTServer shall push the stored diagnostic logs to the IPPTOffBoardClient via diagnostics framework

#### IPPT-REQ-322262/A-Token revocation request

The IPPTServer shall be able to receive the Token revocation request from IPPTOffBoardClient. Upon receiving the token revocation request command the IPPTServer shall send an ack (command response) with status (in progress / failure) back to the IPPTOffBoardClient. Additionally, with help from IPPTClient the IPPTServer shall send the token revocation request to the application based on the Feature ID present in the token revocation request. The token revocation request shall be per application.

For more information please refer the FTCP protofile.

#### IPPT-REQ-322263/A-Token revocation response

Once application successfully sends the revocation response (success / failure) back to the IPPTClient, the IPPTServer shall convey this information back to the IPPTOffBoardClient via Alert.

Based on the response from the application, the token revocation response Alert shall indicate:

0 – Success (Token successfully revoked for the Feature ID)

1 – Failure (No response from the application)

If no response is received from the application, the IPPTServer shall retry 5 times and if there is no response shall return “1 – Failure” to the IPPTOffBoardClient .

For more information please refer to the FTCP Protofile.

#### IPPT-REQ-xxxxxx/A-Configure DNS Server

The current DNS Server on ECG shall be leveraged to configure Policy Zone files.

DNS Resolver shall be leveraged to apply the set of whitelisting rules implemented in the RPZ file



#### IPPT-REQ-xxxxxx/A-RPZ Zone file

The Current DNS server on ECG shall be leveraged to configure RPZ Zone file. RPZ Zone file consists of a policy file with whitelist of URL’s and Blacklist if needed

There can be number of Zone files based on number of applications or just a single file

Note: Current software supports 32 zone files for better performance.

#### IPPT-REQ-xxxxxx/A-Maintain Whitelist Domain

DNS Resolver in ECG shall maintain a list of approved domains that the system, feature, and application may connect to using Response Policy Zones. Only connections to domains on the whitelist shall be resolved. All redirect attempts shall be reconciled against the whitelist. If a requested domain is not on the whitelist, the system shall not permit the connection to be made by returning NXDOMAIN. All connection approvals, request redirects, and connection denials shall be logged and reported in an auditable manner.

All the applications that go through WIR/IPPT shall provide the list of domains that they wanted to connect for content and that list shall be used by DNS RPZ for whitelisting

#### IPPT-REQ-xxxxxx/A-Storing Base domain Whitelist locally

The base whitelist shall be in a write-protected location on the gateway. The base domain whitelist shall be stored locally to always make available to the system

The base whitelist shall include core Ford and SDN domains. These domains shall be the first set of domains in the whitelist file.

#### IPPT-REQ-xxxxxx/A-Configure Caching and Forwarding DNS Server

ECG shall configure Caching and Forwarding DNS Server. ECG shall be the primary DNS server and TCU, SYNC shall be secondary DNS server (when ECG is not available)

#### IPPT-REQ-xxxxxx/A-Maintain Blacklist Domain

For exceptional use case like an application having unlimited number of URL’s

DNS Resolver shall maintain a Blacklist domain against which DNS is resolved

#### IPPT-REQ-xxxxxx/A Forwarding requests to DNS entities

If the domains are not present in the blacklist, DNS Resolver in ECG shall forward requests to approved DNS entities

#### IPPT-REQ-xxxxxx/A-logging and Auditing

All connection Approvals, redirect requests and connection denials shall be logged and reported in an Auditable manner

It shall log security related events in order to troubleshoot defects and trace network

Note: Configuration file in Bind9 shall need to be updated

#### IPPT-REQ-xxxxxx/A-Updating Whitelist

The Whitelist shall be updated by cloud periodically either for new features or for any security related reasons via OTA Updates

The Whitelist update process shall leverage a secure process and all DNS configuration files shall be reviewed by Ford security team via Jira request process

#### IPPT-REQ-xxxxxx/A-Restarting DNS Resolver

DNS Resolver shall be restarted after OTA updates in order to update policy zone file

### Use Cases

#### IPPT-UC-REQ-XXXXX/A-Configuring DNS Server

|  |  |
| --- | --- |
| **Actors** | DNS resolver, RPZ file |
| **Pre-conditions** | DNS resolver shall already be running and need to implement white listing using RPZ files |
| **Scenario Description** | The current DNS server on ECG shall be leveraged to configure policy zone files |
| **Post-conditions** | DNS resolver shall be leveraged to apply the set of white listing rules implemented in the RPZ file |
| **Interfaces** |  |
| **Notes** |  |



#### IPPT-UC-REQ-XXXXX/A-Application requesting for DNS Resolution

|  |  |
| --- | --- |
| **Actors** | Application, DNS Resolver |
| **Pre-conditions** | Application requests for a DNS query by sending its endpoint URL |
| **Scenario Description** | 1. Application requests for end point IP address by sending endpoint URL to the DNS Resolver. 2. DNS Resolver has set of white listing rules setup in such a way that it would only resolve the URL into Destination IP if it is present in the White List 3. DNS resolver contacts authoritative DNS to get IP address 4. DNS resolver returns destination IP address or failure |
| **Post-conditions** | DNS resolver returns destination IP address or failure |
| **Interfaces** |  |
| **Notes** |  |

1. Application requests for DNS Resolution happy path:



1. Application request for DNS Resolution and the URL is not present /updated in Whitelist:



1. Application request for DNS Resolution and it is not having a defined set of whitelist URLs:



#### IPPT-UC-REQ-XXXXX/A-Updating URL’s

|  |  |
| --- | --- |
| **Actors** | Application, DNS Resolver- zone file, OTA Update/ECU Configurations |
| **Pre-conditions** | Applications destination URL shall be changed, or a new URL gets added |
| **Scenario Description** | 1. Whenever the application wants to update the URL/ domain it wants to connect to or it wants to connect to a new domain, it shall be updated in DNS Resolver zone file where the whitelist is managed 2. This update can be done as part of OTA updates 3. DNS resolver shall be restarted to apply updated policy zone file |
| **Post-conditions** | The policy zone file shall be updated with the New URL’s |
| **Interfaces** |  |
| **Notes** |  |



### White Box View

## IPPT-FUN-REQ-304267/A-IPPT Performance

### Requirements

#### IPPT-REQ-304255/A-IPPT application request processing

The IPPTClient and IPPTServer shall not add any latency with respect to token generation.

The round trip time from receipt of a request from application to composing an FTCP message and sending to IPPTOffBoardClient shall happen within IPPT\_REQUEST\_PROCESSING\_TIME which is 1 millisecond

#### IPPT-REQ-304256/A-IPPT application response processing

The IPPTClient and IPPTServer shall not add any latency with respect to token response.

The round trip time from receipt of a token FTCP message from IPPTOffBoardClient to composing and sending the token information to requesting application in vehicle shall happen within IPPT\_RESPONSE\_PROCESSING\_TIME which is 1 millisecond

### Use Cases

### White Box View

## IPPT-FUN-REQ-304268/A-IPPT Configuration

### Requirements

#### IPPT-REQ-304257/A-Configuration parameters

| **Parameter** | **Description** | **Tuned Value FoE** | **Metric** | **Resolution** | **Tuning Range** |
| --- | --- | --- | --- | --- | --- |
| IPPT\_LOGS\_STORAGE\_TIME | Time for which logs will be stored at any point on IPPTServer | 43200 | Minutes | 1 | 0 - 65535 |

### Use Cases

### White Box View

# Appendix: Reference Documents

|  |  |
| --- | --- |
| Reference # | Document Title |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |
| 16 |  |
| 17 |  |