Research & Vehicle Technology
"Infotainment Systems Product Development"

Feature – Time Synchronization Service

Infotainment Subsystem Part Specific Specification (SPSS)

Version 1.1

UNCONTROLLED COPY IF PRINTED

Version Date: August 19, 2021

FORD CONFIDENTIAL

Revision History

Date	Version	Notes	
April 1, 2021	**1.0**	**Initial Release**	
August 19, 2021	**1.1**		
	TSS-STR-877908/B-Overview	rpaquet2 - Added reference section	
	TSS-STR-877911/B-Terminology and Abbreviations	rpaquet2 - Updated table	
	TSS-STR-934010/A-References	rpaquet2 - new	
	TSS-STR-877912/B-TSS Variable Definition	rpaquet2 - update reference numbers	
	TSS-STR-877913/B-Leap Indicator	rpaquet2 - update table number	
	TSS-STR-877914/B-Mode	rpaquet2 - updated	
	TSS-STR-877916/B-Server/Client Mode	rpaquet2 - fixed grammer issue	
	TSS-STR-877920/B-Extension Field Format	rpaquet2 - Updated	
	TSS-STR-877922/B-Kiss Code	rpaquet2 - Updated content	
	TSS-STR-882954/B-Message Digest	rpaquet2 - added exception	
	TSS-STR-877974/B-Physical Mapping of Stratum and Operating Mode	rpaquet2 - updated table number	
	TSS-STR-877903/B-Functional Definition	rpaquet2 - remvoed 411413	
	TSS-FUN-REQ-411359/B-Determination of Stratum and Operating Mode	rpaquet2 - Added section for Use Cases	
	TSS-STR-934095/A-Use Cases	rpaquet2 - New	
	TSS-UC-REQ-431724/A-ADAS-TCU Time Synchronization	rpaquet2 - New	
	TSS-UC-REQ-431725/A-AR-GNSS Source Time Synchronization	rpaquet2 - New	
	TSS-REQ-411360/B-Stratum 1 NTP Primary Server	rpaquet2 - Removed content from this requirement and moved it to 3 new requirements underneath this one	
	TSS-REQ-431686/A-Configuration of Stratum 1 NTP Primary Server	rpaquet2 - New	
	TSS-REQ-431687/A-Tasks for Stratum 1 NTP Primary Server	rpaquet2 - New	
	TSS-REQ-431688/A-Synchronization Tasks for Stratum 1 NTP Primary Server	rpaquet2 - New	
	TSS-REQ-411371/B-Stratum 2 NTP Server/Client	rpaquet2 - Removed content from this requirement and moved it to 5 new requirements underneath this one	
	TSS-REQ-431689/A-Configuration of Stratum 2 NTP Server/Client	rpaquet2 - New	
	TSS-REQ-431690/A-Operation modes of Stratum 2 ECU	rpaquet2 - New	
	TSS-REQ-431691/A-Tasks for Stratum 2 NTP Server/Client	rpaquet2 - New	
	TSS-REQ-431692/A-Synchronization Tasks for Stratum 2 NTP Client	rpaquet2 - New	
	TSS-REQ-431693/A-Synchronization Tasks for Stratum 2 NTP Server	rpaquet2 - New	
	TSS-REQ-411377/B-Stratum 3 NTP Client	rpaquet2 - Removed content from this requirement and moved it to 3 new requirements underneath this one	
	TSS-REQ-431694/A-Configuration of Stratum 3 NTP Client	rpaquet2 - New	
	TSS-REQ-431695/A-Tasks for Stratum 3 NTP Client	rpaquet2 - New	
	TSS-REQ-431696/A-Synchronization Tasks for Stratum 3 NTP Client	rpaquet2 - New	
	TSS-SD-REQ-411431/B-Time synchronization with two ECU's in Stratum 1	rpaquet2 - Updated diagram	
	TSS-FUN-REQ-411355/B-TSS Security	rpaquet2 - Added Sequence Diagram section	
	TSS-STR-877905/B-Requirements	rpaquet2 - Removed 411381, 412762 added new reqs	
	TSS-REQ-411356/B-Overview	rpaquet2 - Updated content	
	TSS-REQ-411379/B-Key Establishment	rpaquet2 - Removed 411380 and update content in this req	

TSS-REQ-436258/A-Stratum 1 ECU Tasks for Key Establishment Phase	rpaquet2 - new
TSS-REQ-436259/A-Stratum 2 ECU Tasks for Key Establishment Phase	rpaquet2 - new
TSS-REQ-436260/A-Stratum 3 ECU Tasks for Key Establishment Phase	rpaquet2 - new
TSS-REQ-411386/B-Secure Time Synchronization	rpaquet2 - Updated content removed 411387
TSS-REQ-411388/B-Server Time Response	rpaquet2 - Updated content
TSS-REQ-411389/B-Client Time Synchronization	rpaquet2 - Updated content
TSS-STR-934106/A-White Box Views	rpaquet2 - New
TSS-STR-934107/A-Sequence Diagrams	rpaquet2 - New
TSS-SD-REQ-431749/A-Key Exchange Symmetric Key Authentication	rpaquet2 - New
TSS-FUN-REQ-411390/B-UTC Offset	rpaquet2 - Added use case section
TSS-STR-934097/A-Use Cases	rpaquet2 - New
TSS-UC-REQ-431726/A-UTC offset for Fleet/UBI	rpaquet2 - New
TSS-REQ-411391/B-UTC Offset	rpaquet2 - updated content
TSS-SD-REQ-411433/B-UTC Offset for ECG APPs and Connected ECUs	rpaquet2 - updated diagram
TSS-FUN-REQ-411394/B-UTC Date/Time over CAN Bus	rpaquet2 - Added Use Case section
TSS-STR-934098/A-Use Cases	rpaquet2 - New
TSS-UC-REQ-431727/A-UTC for evaluating user key - BLEM	rpaquet2 - New
TSS-UC-REQ-431728/A-UTC for evaluating public key certificates - NFAM	rpaquet2 - New
TSS-REQ-411395/B-Introduction to UTC Clock Master	rpaquet2 - Updated content
TSS-REQ-411397/B-CAN Bus in Sleep	rpaquet2 - Updated content
TSS-REQ-411399/B-CAN Bus Awake	rpaquet2 - Updated content
TSS-REQ-411400/B-CAN Bus Transition Awake to Sleep	rpaquet2 - Updated content
TSS-REQ-411401/B-CAN Bus Transition Sleep to Awake	rpaquet2 - Updated content
TSS-FUN-REQ-411403/B-UTC and Vehicle Date/Time for Connected Feature Application	rpaquet2 - Added Use Case section
TSS-STR-934099/A-Use Cases	rpaquet2 - New
TSS-UC-REQ-431729/A-UTC Base Point for Fleet/UBI	rpaquet2 - New
TSS-UC-REQ-431730/A-Vehicle Time API for OTA Scheduling	rpaquet2 - New
TSS-UC-REQ-431731/A-UTC TIME API for all Vehicle - Ford cloud communication	rpaquet2 - New
TSS-REQ-411404/B-UTC and Vehicle Date/Time API's	rpaquet2 - Removed NTS
TSS-REQ-411405/B-UTC Delta	rpaquet2 - Updated content
TSS-REQ-411410/B-Vehicle Time Delta	rpaquet2 - Removed NTS
TSS-SD-REQ-411434/B-UTC Delta and Vehicle Time Delta	rpaquet2 - Updated diagram

Table of Contents

1 Overview

1.1 Time Synchronization Service Overview

Time synchronization service (TSS) is an approach to provide time alignment with high precision and accuracy between fast based ECUs (ECUs with Ethernet connection). To achieve this goal, the well-established time synchronization protocol known as Network Time Protocol (NTP) will be used. The main audiences of this service are as follow:

- Accessory Protocol Interface Module – SYNC / APIM
- Telematics Control Unit – TCU
- Enhanced Central Gateway – ECG
- Augmented Reality – AR
- Advanced Driver Assistance System – ADAS
- All ECUs in the platform with Ethernet connection
- All connected features reside in ECG, TCU, and SYNC

The following diagram conceptually shows the functionality of TSS:

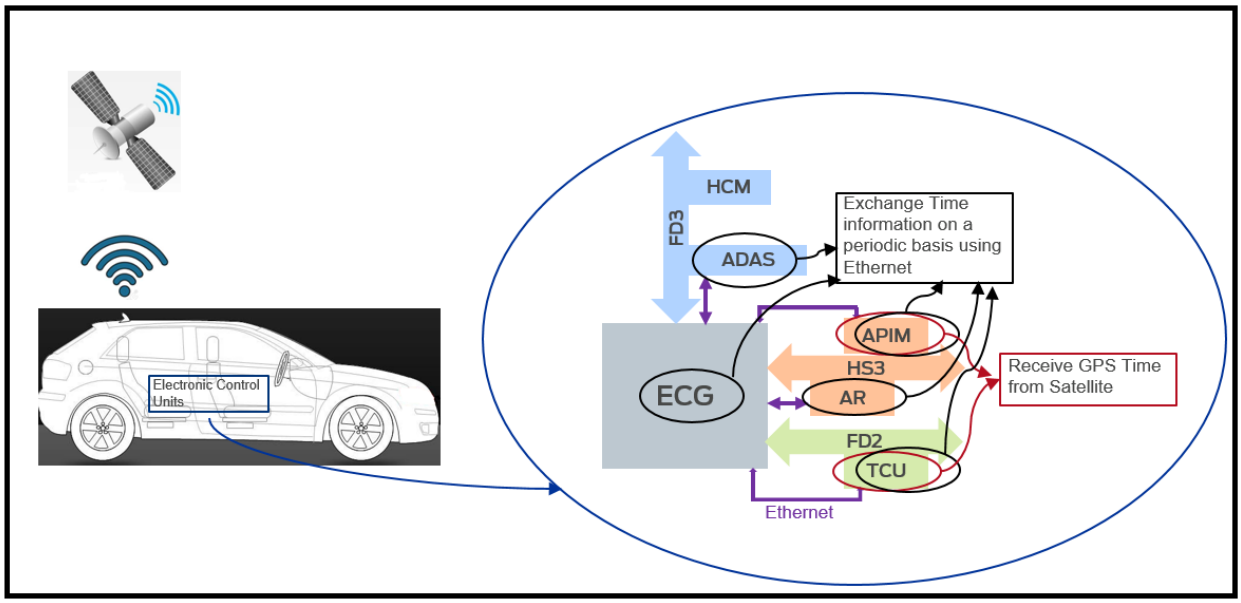


Figure 1. Time Synchronization Architecture

As shown in Figure 1, vehicle receives GNSS information such as UTC from satellite. Then, fast based ECUs exchange time information among each other considering network topology to achieve time alignment.

The main goal of TSS is to accurately, efficiently, and securely synchronize fast based ECUs with high precision UTC.

This service will be deployed in vehicle program (P708 as lead) with FNV3 platform in US, Canada, EU, China, and ROW.

1.2 TSS Assumption

This service will be deployed in the vehicle that has FNV3 architecture, GNSS antenna and built-in GNSS receiver. The availability of Ethernet connection between fast based ECUs like ECG, TCU, SYNC, ADAS and AR is a must.

1.3 Terminology and Abbreviations

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

Term	Description
APIM	Accessory Protocol Interface Module
ECG	Enhanced Central Gateway
UTC	Coordinated Universal Time
FNV3	Fully Network Vehicle 3.0 architecture
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
TCU	Telematics Control Unit
ADAS	Advanced Driver Assistance System
AR	Augmented Reality
MAC	Message Authentication Code
TSS	Time Synchronization Service
NTP	Network Time Protocol – Version 4 is being used for this service
SNTP	Simple Network Time Protocol
CMAC	Cipher-based Message Authentication Code
SHA	Secure Hash Algorithm
AES	Advanced Encryption Standard
CA	Certificate Authority
TLS	Transport Layer Security
EF	Extension Field
BCM	Body Control Module
HMI	Human Machine Interface
Global Clock	Representing BCM (Master Clock) time which is being broadcasted over CAN Bus in periodic basis. Global Clock has local format and user/drive can change its value using HMI screen.
System Time	Representing application processor time
Vehicle Time	Representing local time of vehicle, user/driver can change its value using HMI screen. Can be used interchangeably for Global Clock.
UTC Base Point	Representing UTC time/date at the IGN ON of vehicle
UTC Delta	Representing time/date difference between ECU system time and UTC date/time from GNSS
Vehicle Time Delta	Representing time/date difference between ECU system time and Vehicle Time

Table 1 Terminology and Abbreviations

1.4 References

These requirements build on [IETF RFC 5905], “Network Time Protocol Version 4: Protocol and Algorithms Specification”, and [IETF RFC 8573], “Message Authentication Code for Network Time Protocol “published by Internet Engineering Task Force. Use these documents to implement Time Synchronization Service for fast-based ECUs. Table 2 listed the references that are being used for this document.

Reference	Org	Title	Version/Date	Document Number
IETF RFC 5905	IETF	Network Time Protocol Version 4: Protocol and Algorithms Specification	June 2010	RFC 5905
IETF RFC 8573	IETF	Message Authentication Code for Network Time Protocol	June 2019	RFC 8573

Table 2 References

Please note that sections 1.5 and 1.6 summarize the NTP standard. In this document, the audience shall follow the standard NTPv4 protocol and we will highlight any requirements that is different from original NTPv4 standard.

1.5 TSS Variable Definition

The Network Time Protocol (NTP) utilizes a standard packet known as NTP packet for information exchange between different network nodes. Table 3 shows the NTP packet variables with brief description of each field. In general, the NTP packet consists of three components, the NTP header, one or more optional extension fields, and an optional message authentication code (MAC).

Name	Formula	Description
Leap Indicator	LI	Refer to 1.5.1
Version Number	VN	3-bit integer representing the NTP version number
Mode	mode	Refer to 1.5.2
Stratum	stratum	Refer to 1.5.3
Poll	poll	Refer to 1.5.4
Precision	precision/rho	8-bit signed integer representing the precision of the system clock, in log2 seconds.
Root Delay	Rootdelay/delta_r	Total round-trip delay to the reference Clock, in NTP short format.
Root Dispersion	Rootdisp/epsilon_r	Total dispersion to the reference clock, in NTP short format.
Reference ID	refid	32-bit code identifying the server or reference clock. Refer to 1.5.6.
Reference Timestamp	reftime	The local time at which the local clock was last set or corrected, in 64-bit timestamp format. If the local clock has never been synchronized, the value is zero.
Origin Timestamp	org/T1	The local time at which the request departed the client host for the server host, in 64-bit timestamp format. It will always have a nonzero value.
Receive Timestamp	rec/T2	The local time at which the request from the client host arrived at the server host, in 64-bit timestamp format.
Transmit Timestamp	xmt/T3	The local time at which the response departed the server host for the client host, in 64-bit timestamp format. If no request has ever arrived from the client, the value is zero. The local time when the latest NTP message was transmitted.
Destination Timestamp	sst/T4	Time at the client when the reply arrived from the server, in NTP timestamp format (64 bits). **Not included in NTP header**.
Extension Field N		Refer to 1.5.5
Key Identifier	keyid	32-bit unsigned integer used by the client and server to designate a secret 128-bit long key for AES-128.

Message Digest	dgst	A 128-bit string that is the output of AES-CMAC. Refer to 1.5.7.

Table 3 NTP Packet Variables

Please note that if the NTP has access to the physical layer, then the timestamps are associated with the beginning of the symbol after the start of frame. Otherwise, implementations should attempt to associate the timestamp to the earliest accessible point in the frame.

1.5.1 Leap Indicator

2-bit integer warning of an impending leap second to be inserted or deleted in the last minute of the current month with values defined in Table 4 Leap Indicator.

Value	Meaning
0	No warning
1	Last minute of the day has 61 seconds
2	Last minute of the day has 59 seconds
3	Unknown

Table 4 Leap Indicator

1.5.2 Mode

3-bit integer representing the operating mode of nodes in a network, with values defined in Table 5 Operating Mode.

Meaning	Association Mode Value	Packet Mode Value
Reserved	0	N/A
Symmetric active	1	1 or 2
Symmetric passive	2	1
Client	3	4
Server	4	3
Broadcast	5	5
NTP control message	6	N/A
Reserved for private use	7	N/A

Table 5 Operating Mode

In general, there are three NTP protocol variants: symmetric, client/server, and broadcast. Each is associated with an association mode (a description of the relationship between two NTP speakers) as shown in 5. Server/Client operating modes must be mainly used for time synchronization process. All ECUs in the network must only support Server/Client operating modes.

1.5.2.1 Symmetric Active/Passive Mode

In a symmetric mode, a peer operates as both a server and client using either a symmetric active or symmetric passive association. In a symmetric mode, peers both push and pull synchronization to and from each other.

1.5.2.2 Server/Client Mode

A node in server mode provides synchronization to one or more clients, but do not accept synchronization from them. A server can also be a reference clock driver that obtains time directly from a standard source such as GPS receiver. In the client mode, a client sends packet mode 4 to server, which returns packet mode 3 packets.
A system in client mode pull synchronization from servers or nodes in server mode.

1.5.2.3 Broadcast Server/Client Mode

A node in broadcast server mode periodically sends NTP packet that can be received by multiple clients. Overall, a node in broadcast server mode pushes synchronization to clients and other servers.

1.5.3 Stratum

8-bit integer representing the stratum, with values defined in Table 6.

Value	Meaning
0	Unspecified or invalid
1	Primary server/equipped with a GNSS receiver
2 -15	Secondary server
16	Unsynchronized
17-255	Reserved

Table 6 Stratum

In general, a stratum number defines the level of each server in the hierarchy. Primary servers are assigned stratum one, secondary servers at each lower level are assigned stratum numbers one greater than the preceding level. Overall, as the stratum number increases, its accuracy degrades depending on the particular network path and system clock stability.

1.5.4 Poll

8-bit signed integer representing the maximum interval between successive messages, in $\log_2$ seconds. The minimum poll exponent shall be set to 4 or 16 seconds and the maximum poll exponent shall be set to 17 or 36 hours.

1.5.5 Extension Field Format

One or more extension field can be inserted after the header and before MAC, which is always present when an extension field is present. An extension field contains a request or response message in the format shown in Figure 2. All extension fields are zero-padded to a word boundary.

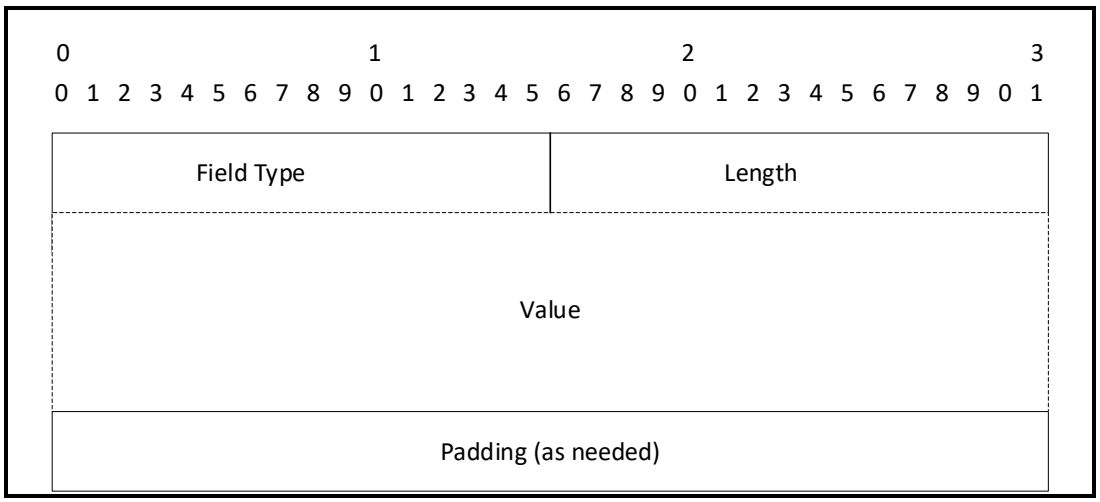


Figure 2. Extension Field Format

The Field type field is specific to the defined function. The Length field is a 16-bit unsigned integer that indicates the length of the entire extension field in octets, including the Padding field. Please note that support for EF is optional based on the design.

1.5.6 Reference ID

32-bit code identifying the particular server or reference clock. The interpretation depends on the value in the stratum field. Table 7 shows the reference ID interpretation based on stratum.

Stratum	Name for Reference ID	Usage
0	Kiss Code	Debugging and monitoring
1	-	Four-octet, left-justified, zero-padded ASCII string assigned to the reference clock
Above 1	-	Detecting timing loops

Table 7 Interpretation of Reference ID

1.5.6.1 Kiss Code

The kiss codes can provide useful information for an intelligent client, either NTPv4 or SNTPv4 (used in network topology containing one NTP server in stratum 1). Kiss codes are encoded in four-character ASCII strings that are left justified and zero filled. The strings are designed for character displays and log files. A list of the currently defined kiss codes is shown in Table 8. Recipients of kiss codes shall inspect them and take actions as stated in the table below. Please note that the support for Kiss Code implementation is optional.

Code		Meaning	Action
ACST		The association belongs to a unicast server.	-
AUTH		Server authentication failed	-
AUTO		Auto key sequence failed.	-
BCST		The association belongs to a broadcast server.	-
CRYP		Cryptographic authentication or identification failed.	-
DENY		Access denied by remote server.	The client MUST demobilize any associations to that server and stop sending packets to that server.
DROP		Lost peer in symmetric mode.	-
RSTR		Access denied due to local policy.	The client MUST demobilize any associations to that server and stop sending packets to that server.
INIT		The association has not yet synchronized for the first time.	-
MCST		The association belongs to a dynamically discovered server	-
NKEY		No key found. Either the key was never installed or is trusted	-
RATE		Rate exceeded. The server has temporarily denied access because the client exceeded the rate threshold.	The client MUST immediately reduce its polling interval to that server and continue to reduce it each time it receives a RATE kiss code.
RMOT		Alteration of association from a remote host running ntpdc.	-
STEP		A step change in system time has occurred, but the association has not yet resynchronized.	-

Table 8 Kiss Codes

Kiss codes beginning with the ASCII character "X" are for unregistered experimentation and development and must be ignored if not recognized.

1.5.7 Message Digest

If NTP authentication is implemented, then AES-CMAC must be computed over all fields (expect KeyID and Message Digest Fields) in the NTP header and any extension fields that are present in the NTP packet. The MAC key for NTP must be an AES-128 key that is 128 bits in length, and the resulting MAC tag must be at least 128 bits in length (not larger than 160 bits). So, ECU participating in the time synchronization must implement AES-CMAC and share the corresponding symmetric key.

1.6 TSS Data Format

All NTP time values must be represented in two's-complement format with bits numbered in big-endian fashion from zero starting at the left, or high order, position. In general, TSS shall support three data formats as shown in Figure 3, a 128-bit date format, a 64-bit timestamp format and a 32-bit short format.

The 128-bit date format shall be used where sufficient storage and word size are available. It includes a 64-bit signed Seconds field and a 64-bit Fraction field. For convenience in mapping between formats, the second field is divided into a 32-bit Era Number field and a 32-bit Era Offset field.

The 64-bit timestamp format shall be used in packet headers and other places with limited word size. It shall include a 32-bit unsigned Seconds field and a 32-bit Fraction field.

The 32-bit short timestamp shall be used in delay and dispersion header fields. It includes a 16-bit unsigned Seconds field and a 16-bit Fraction field.

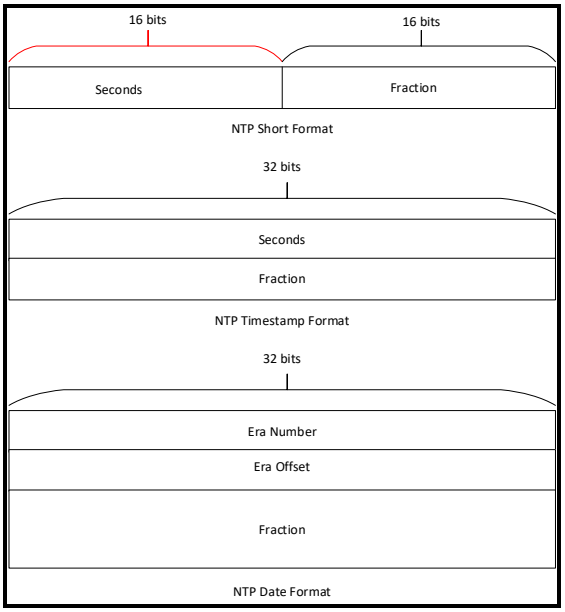


Figure 3. NTP Data Format

In the Date and Timestamp format, the prime epoch or based date shall be set to January 1st, 1900,00:00:00:00 when all bits are zeros. Dates are relative to the prime epoch, values greater than zero shall represent times after that date and values less than zero represent times before it. Note that the Era Offset field of the date format and the Seconds field of the timestamp format have the same interpretation.

2 Architectural Design

2.1 Boundary Diagram

Boundary diagram of Time Synchronization Service.

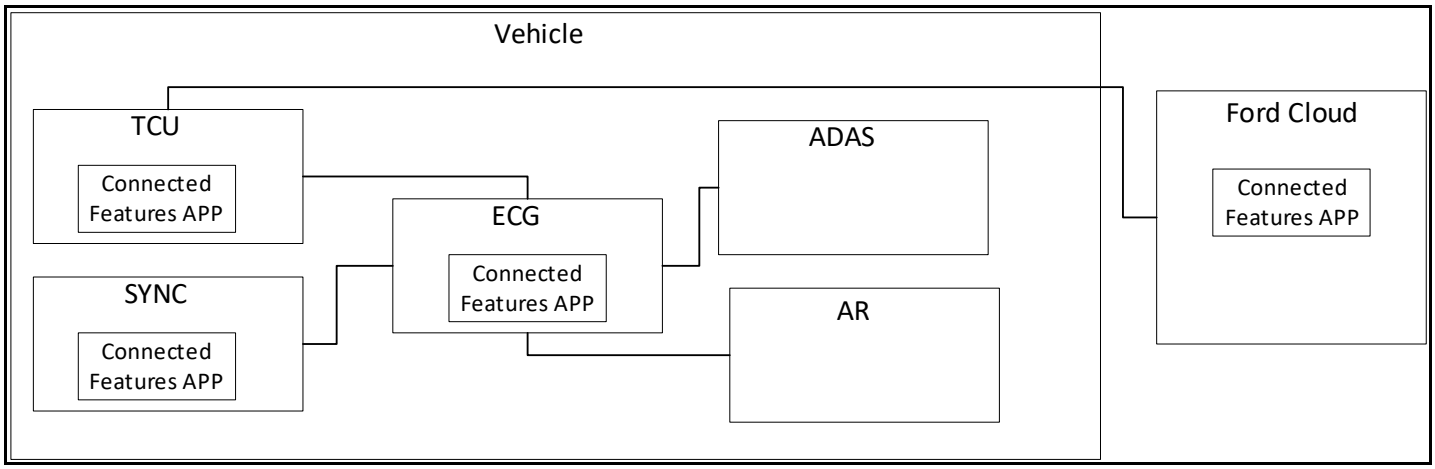


Figure 4. TSS Boundary Diagram

2.2 Physical Mapping of Stratum and Operating Mode

Table 9 shows an example of how the Stratum that make up the Time Synchronization Service can be mapped into physical modules and operating modes. This mapping is an FNV3 example only and does not necessarily carryover to other carlines or vehicle architectures.

Stratum	Physical Module (ECU)	Operating Mode
1	SYNC/TCU	NTP Server
2	ECG	NTP Client & SNTP Server
3	ADAS/AR/SYNC/TCU	NTP Client

Table 9 Mapping Physical Module

3 Functional Definition

3.1 TSS-FUN-REQ-411359/B-Determination of Stratum and Operating Mode

3.1.1 Use Cases

3.1.1.1 TSS-UC-REQ-431724/A-ADAS-TCU Time Synchronization

Actors	Stratum 2 NTP Server & Stratum 3 NTP Client
Pre-conditions	Ignition is on KEY ON position ECG AP is operational Ethernet is fully functional
Scenario Description	The ADAS Module needs to be time synchronized with the TCU. Specifically, the TCU is collecting V2X data from other vehicles via cellular modem, sending the measurements to the ADAS Module over Ethernet. The ADAS module then needs to time align these measurements with measurements observed via radar and cameras.
Post-conditions	The maximum time discrepancy between ADAS and TCU V2X is +/- 5 ms. The features that rely on NTP won't be fully active until 10~12 s after ignition.
List of Exception Use Cases	N/A
Interfaces	V2V sensing and V2X sensing Vehicle System Interface

3.1.1.2 TSS-UC-REQ-431725/A-AR-GNSS Source Time Synchronization

Actors	Stratum 2 NTP Server & Stratum 3 NTP Client
Pre-conditions	Ignition is on KEY ON position ECG AP is operational Ethernet is fully functional
Scenario Description	All Information received by AR need to be time aligned with original source. It is required for AR to receive data from GNSS source that is in time alignment with AR as close as possible.
Post-conditions	N/A
List of Exception Use Cases	N/A
Interfaces	Vehicle System Interface

3.1.2 Requirements

3.1.2.1 TSS-REQ-411360/B-Stratum 1 NTP Primary Server

3.1.2.1.1 TSS-REQ-431686/A-Configuration of Stratum 1 NTP Primary Server

The ECU shall be configured as Stratum 1 NTP primary server if the following conditions are met:

- It has GNSS antenna connection, built-in GNSS receiver, and Ethernet connection with gateway module.
- It can accurately receive UTC time from satellite with at least millisecond resolution.
- ECG DID DE01 and SYNC DID FEB6 shall be used to determine which ECU has built-in GNSS receiver.

The ECU (originally configured as Stratum 1/primary server) must be degraded to secondary server (lowest stratum available in network) if the following conditions are meet:

- Unable to receive GNSS data due to failure in GNSS receiver or GNSS antenna connection.
- Unable to maintain UTC time during power off state. In this case, the ECU must be degraded to secondary server (lowest stratum available in network) until it receives GNSS data.

3.1.2.1.2 TSS-REQ-431687/A-Tasks for Stratum 1 NTP Primary Server

The Stratum 1 NTP primary server is responsible for the tasks listed below:

- Receiving GNSS data from satellites and extract high precision UTC time. Minimum acceptable precision for UTC time is millisecond.
    1. ECU shall perform sanity check on UTC to make sure that the derived UTC time (from GNSS data) is valid/accurate.
        1. If TCU is the GNSS source, then the Fix Type shall not be equal to 0x00 or 0x01 or 0x04 for UTC time and date to be valid. Please refer to “Location Service APIM SPSS” for detailed information on Fix Type.
- Dropping any incoming NTP packets on UDP default port 123 or any port that originate from IP addresses external from the vehicle.
- Adjusting/updating its internal system time (application processor) with high precision UTC time. The difference between internal system time and UTC time (from GNSS data) shall never exceed 2 millisecond.
    1. Adjusting/updating CAN processor time with high precision UTC time is optional. However, we advise to synchronize CAN processor time with high precision UTC time to synchronize in-field logging and debugging timestamp.
    2. If ECU cannot adjust its system time with high precision UTC time, it shall follow the requirements as stated in REQ-411405/B-UTC Delta.
        1. For populating/generating timestamp (to be used in) for NTP packet, ECU must use high precision UTC time with at least millisecond resolution.
- If ECU has the capability to maintain (in power off state) UTC using its RTC, then ECU shall rely on its internal system time after IGN ON and before receiving GNSS data.
    1. From suspended state and cold re-boot state:
        1. The ECU shall use its RTC to generate timestamps for NTP packet until it receives valid UTC time.
        2. Maximum acceptable time drift for RTC is 50 seconds per day.
        3. Default to Unix Epoch time before re-boots completion of other system processors is tolerable/acceptable.
- If ECU does not have the capability to maintain UTC using its RTC, then ECU shall be degraded to secondary server (lowest stratum available in network) until it receives GNSS data.

3.1.2.1.3 TSS-REQ-431688/A-Synchronization Tasks for Stratum 1 NTP Primary Server

Please note that this section provides the high-level description of required steps for Stratum 1 NTP primary server for time synchronization as described in standard NTPv4 protocol (RFC 5905). The ECU configured as stratum 1 operating as NTP server shall conform following steps for time synchronization purpose:

- Must complete the Key Establishment phase as explained in REQ-411379-Key Establishment immediately after ECU re-boot/dependency on the availability of SOA/TLS.
- Shall use UDP for NTP packet exchange between the authorized ECUs from lower stratum (stratum 2).
- Shall continuously listen to the authorized ECUs from lower stratum (stratum 2).
- Shall receive NTP packet from authorized ECUs, record the system time at which the NTP packet arrived and marked the timestamp as received timestamp.
- Shall perform sanity check by comparing the transmit timestamp in the current NTP packet and previous received NTP packet (from the same NTP client), if they match the NTP packet is duplicate.
    - ECU shall discard the packet and ignore the request.
- Shall interchanges the source and destination addresses and ports to be used for the response.
- Shall record the origin timestamp from NTP header packet to be used for the response (copy it and populate it in the origin timestamp field in the NTP respond packet).
- Shall respond to the ECUs by generating key identifier and message digest fields as explained in REQ-411388-Server Time Response and populating received and transmit timestamp in the NTP packet header.

3.1.2.2 TSS-REQ-411371/B-Stratum 2 NTP Server/Client

3.1.2.2.1 TSS-REQ-431689/A-Configuration of Stratum 2 NTP Server/Client

The ECU shall be configured as stratum 2 if the following conditions are met:

- It is a gateway module in the network.
- It has direct Ethernet connection with primary servers (stratum 1 NTP primary server).
- It has direct Ethernet connection with all nodes in the network.

3.1.2.2.2 TSS-REQ-431690/A-Operation modes of Stratum 2 ECU

The ECU in stratum 2 shall serve in two different modes,

1. NTP client, for communication with ECUs in stratum 1 (NTP primary servers). In this mode, ECU must accept time synchronization from authorized NTP primary servers.
2. NTP server, for communication with ECUs in stratum 3. In this mode, ECU must provide time synchronization to one or more clients in lower stratum and must not accept synchronization from them while operating as a secondary server.

3.1.2.2.3 TSS-REQ-431691/A-Tasks for Stratum 2 NTP Server/Client

The ECU configured as stratum 2 is responsible for the tasks listed below:

- Follow Key Establishment process as explained in section REQ-411379-Key Establishment immediately after ECU re-boot/dependency on the availability of SOA/TLS.
- Dropping any incoming NTP packets on UDP default port 123 or any port that originate from IP addresses external from the vehicle.
- The ECU configured as Stratum 2 shall maintain UTC using its external RTC in power off state.
    1. Maximum acceptable time drift for RTC is 20 seconds per day.
- Relying on its internal system time/external RTC immediately after IGN ON and before exchanging NTP packet with primary server.
    1. ECU shall use its internal system time/external RTC to generate timestamps for NTP packet.
- In NTP client mode, periodically (defined by poll field) sending NTP packets to one or more primary server (ECUs in Stratum1 configured as primary server) and processing the returned packets when they are received.
    1. If there are multiple NTP primary servers (both TCU and SYNC has GNSS engine and GNSS antenna) in the network, Stratum 2 NTP Client shall prioritize TCU2 (Modem6) over SYNC High.
- In NTP client mode, each time a client makes a measurement with a sever, it shall calculate following four statistics variables
    1. Offset representing the maximum-likelihood time offset of the server clock relative to the system clock.
    2. Delay representing the round-trip delay between the client and server
    3. Dispersion representing the maximum error inherent in the measurement.
    4. Jitter representing the root-mean-square (RMS) average of the most recent offset differences, and it represents the nominal error in estimating the offset.

- In NTP client mode, synchronizing its internal system time (application processor) and external RTC with high precision UTC time provided by primary servers' time (ECUs in Stratum 1 configured as primary servers).
    1. Adjusting/updating CAN processor time with high precision UTC time is optional. However, we advise to synchronize CAN processor time with high precision UTC time to synchronize in-field logging and debugging timestamp.
    2. The difference between internal system time and UTC time (from primary server) shall never exceed 2 milliseconds.
    3. The difference between external RTC time and UTC time shall never exceed 2 milliseconds.

- In NTP server mode, providing time synchronization to NTP clients in lower stratum (Stratum 3).

3.1.2.2.4 TSS-REQ-431692/A-Synchronization Tasks for Stratum 2 NTP Client

Please note that this section provides the high-level description of required steps for Stratum 2 NTP client for time synchronization as described in standard NTPv4 protocol (RFC 5905). The ECU configured as stratum 2 operating as NTP client shall follow following steps for time synchronization purpose:

- Must complete the Key Establishment phase as explained in REQ-411379-Key Establishment immediately after ECU re-boot/dependency on the availability of SOA/TLS.
- Shall use UDP for NTP packet exchange for communication to the authorized Primacy NTP Servers.
- Shall define its mode as client in communication with primary servers (Mode value in NTP packet as 3)
- Shall periodically send NTP packet to the primary servers. The periodic rate (poll in NTP packet header) shall be set to minimum of 16 seconds and maximum of 36 hours.
- Shall populate reference and original timestamps in NTP packet header upon transmitting NTP packet to the primary servers.
- Shall record destination timestamp/s upon receiving response from NTP primary server/s.
- Must verify the server response by following the procedure explained in section REQ-411389-Client Time Synchronization.
- Shall perform sanity check by comparing the original timestamp in the current packet and the transmit timestamp of previous NTP packet (from the same NTP server), if they do not match then the packet is bogus.
    - ECU shall discard the packet, ignore the respond and avoid processing it.
- Shall calculate delay and offset relative to the primary server/s (Stratum 1 NTP server).
- If there is only one primary server in the network, it shall utilize the offset to update the internal system time.
- If there are more than one primary server then it shall follow below steps:
    1. Shall utilize the **Selection** algorithm to detect “falseticker” (incorrect server) and select “truchimers”(correct server).
    2. Shall utilize the **Cluster** algorithm to find the most accurate set of truechimers.
    3. Shall utilize the **Combine** algorithm to compute the final clock offset by statistically averaging the surviving truechimers.
    4. Shall update its internal system time utilizing the offset calculated in the previous step.

3.1.2.2.5 TSS-REQ-431693/A-Synchronization Tasks for Stratum 2 NTP Server

Please note that this section provides the high-level description of required steps for Stratum 2 NTP server for time synchronization as described in standard NTPv4 protocol (RFC 5905). The ECU configured as stratum 2 operating as NTP server shall follow following steps for providing synchronization:

- Must complete the Key Establishment phase as explained in section REQ-411379-Key Establishment immediately after ECU re-boot/dependency on the availability of SOA/TLS.
- Shall use UDP for NTP packet exchange between the authorized ECUs from lower stratum (stratum 3).
- Shall define its mode as server in communication with ECUs in lower stratum (Mode value in NTP packet as 4).
- Shall continuously listen to the authorized ECUs in client mode from lower stratum (stratum 3)
- Shall receive NTP packet from authorized ECUs, record the system time at which the NTP packet arrived and marked the timestamp as received timestamp.
- Shall record the origin timestamp from NTP header packet to be used for the response (copy it and populate it in the origin timestamp field in the NTP respond packet).
- Shall perform sanity check by comparing the transmit timestamp in the current NTP packet and previous received NTP packet (from the same NTP client), if they match the NTP packet is duplicate.
    - ECU shall discard the packet and ignore the request.
- Shall respond to the authorized ECUs by generating key identifier and message digest fields as explained in REQ-411388-Server Time Response and populating received and transmit timestamp in the NTP packet header.

3.1.2.3 TSS-REQ-411377/B-Stratum 3 NTP Client

3.1.2.3.1 TSS-REQ-431694/A-Configuration of Stratum 3 NTP Client

The ECU shall be configured as Stratum 3 if the following conditions are met:

- It has direct Ethernet connection with gateway module and it is not a Stratum 1 NTP Server.

3.1.2.3.2 TSS-REQ-431695/A-Tasks for Stratum 3 NTP Client

The ECUs configured as stratum 3 are responsible for the tasks listed below:

- Shall operate in client mode for communication with upper stratum.
- Dropping any incoming NTP packets on UDP default port 123 or any port that originate from IP addresses external from the vehicle.
- Synchronizing its internal system time (application processor) with high precision UTC time provided by ECUs in upper stratum.
    1. Adjusting/updating CAN processor time with high precision UTC time is optional. However, we advise to synchronize CAN processor time with high precision UTC time to synchronize in-field logging and debugging timestamp.
- The difference between internal system time and UTC time (from primary server) shall never exceed 2 milliseconds.
- If ECU cannot adjust its system time with high precision UTC time, it shall follow the requirements as stated in REQ-411405/B-UTC Delta.
    1. For populating/generating timestamp (to be used in) for NTP packet, ECU must use high precision UTC time with at least millisecond resolution.
- Relying on its internal system time immediately after IGN ON and before exchanging NTP packet with secondary server in stratum 2.
    1. ECU shall use its internal system time/RTC to generate timestamps for NTP packet.
    2. If ECU system time is not synced with high precision UTC time, it shall use the UTC Delta as stated in REQ-411405/B-UTC Delta to provide UTC time.

3.1.2.3.3 TSS-REQ-431696/A-Synchronization Tasks for Stratum 3 NTP Client

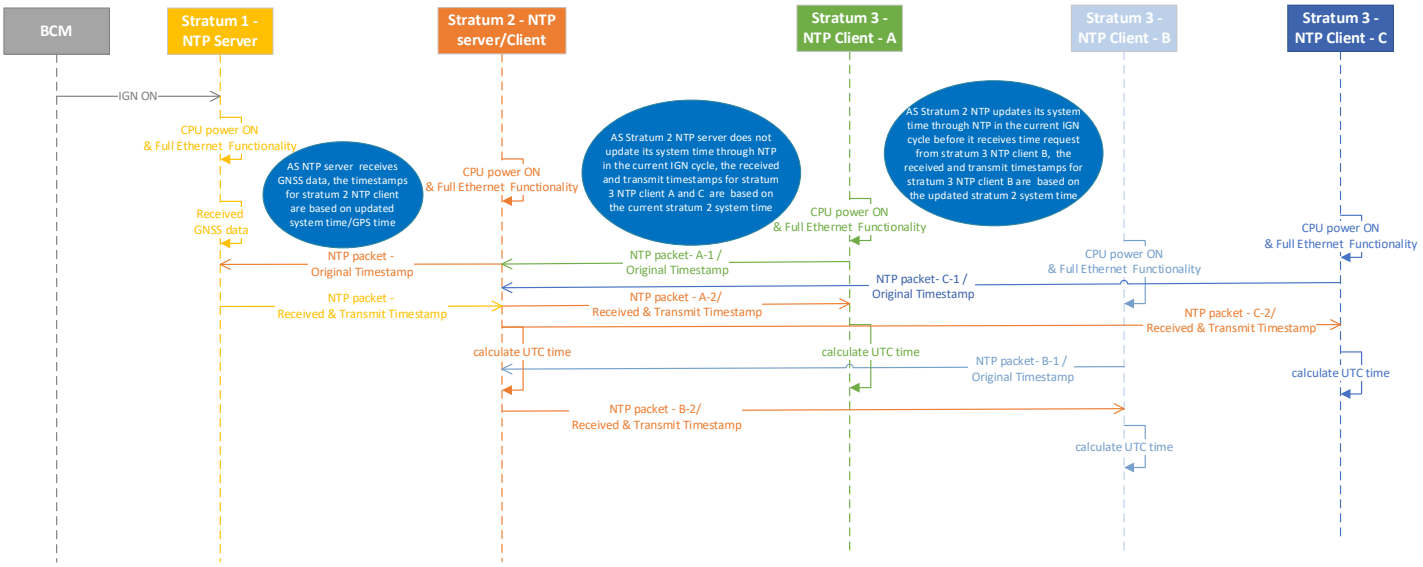
Please note that this section provides the high-level description of required steps for Stratum 3 NTP client for time synchronization as described in standard NTPv4 protocol (RFC 5905). The ECU configured as stratum 3 operating as NTP client shall follow following steps for time synchronization purpose:

- Must complete the Key Establishment phase as explained in REQ-411379-Key Establishment immediately after ECU re-boot/dependency on the availability of SOA/TLS.
- Shall use UDP for NTP packet exchange for communication to the authorized stratum 2 NTP servers.
- Shall define its mode as client in communication with ECUs in stratum 2 (Mode value in NTP packet as 3)
- Shall periodically send NTP packet to the ECUs configured as stratum 2. The periodic rate (poll in NTP packet header) shall be set to minimum of 16 seconds and maximum of 36 hours.
- Shall populate reference and original timestamps in NTP packet header upon transmitting NTP packet to the ECUs in upper stratum.
- Shall record destination timestamp/s upon receiving response from NTP primary server/s.
- Must verify the server response by following the procedure explained in REQ-411389-Client Time Synchronization.
- Shall perform sanity check by comparing the original timestamp in the current packet and the transmit timestamp of previous NTP packet (from the same NTP server), if they do not match then the packet is bogus.
    - ECU shall discard the packet, ignore the respond and avoid processing it.
- Shall calculate delay and offset relative to the ECU/s in stratum 2.
    1. It shall utilize the offset to update the internal system time.

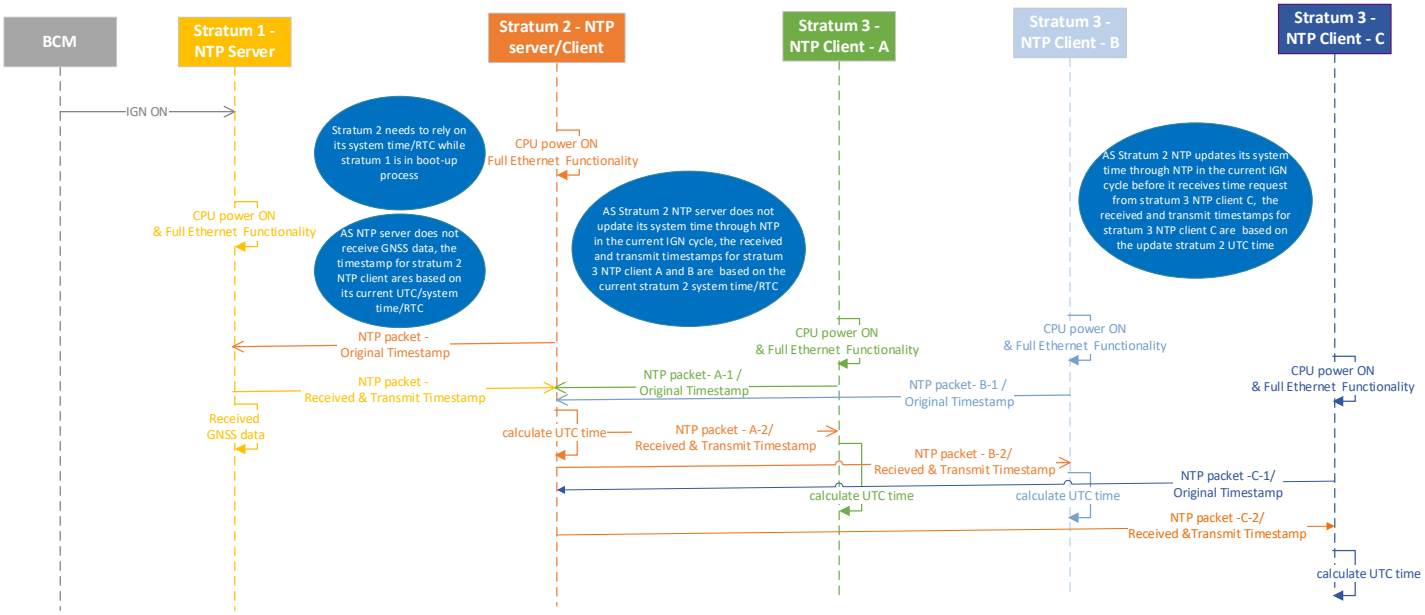
3.1.3 White Box Views

3.1.3.1 Sequence Diagrams

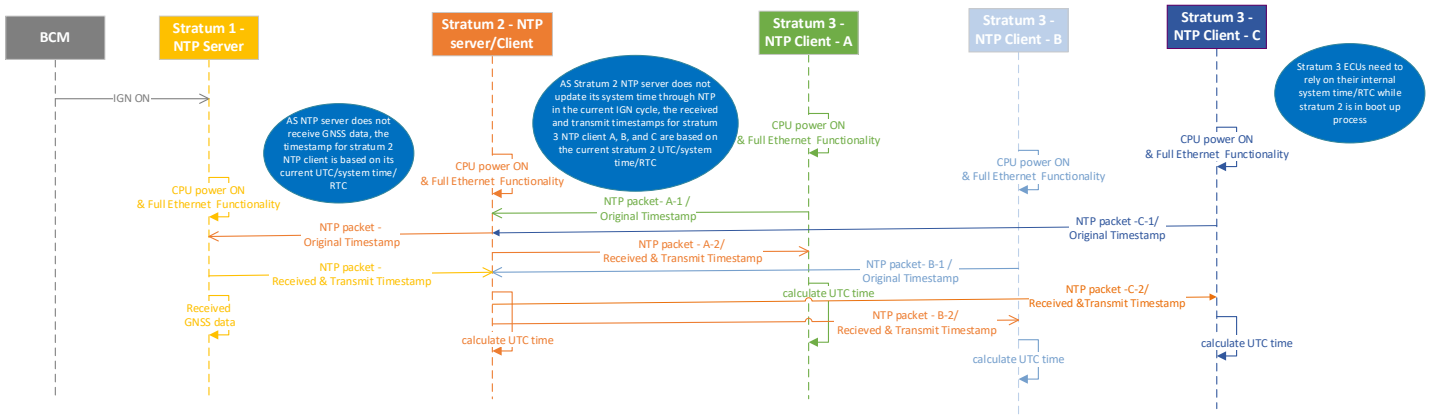
3.1.3.1.1 TSS-SD-REQ-411415/A-Time synchronization and fastest boot up time for Stratum 1



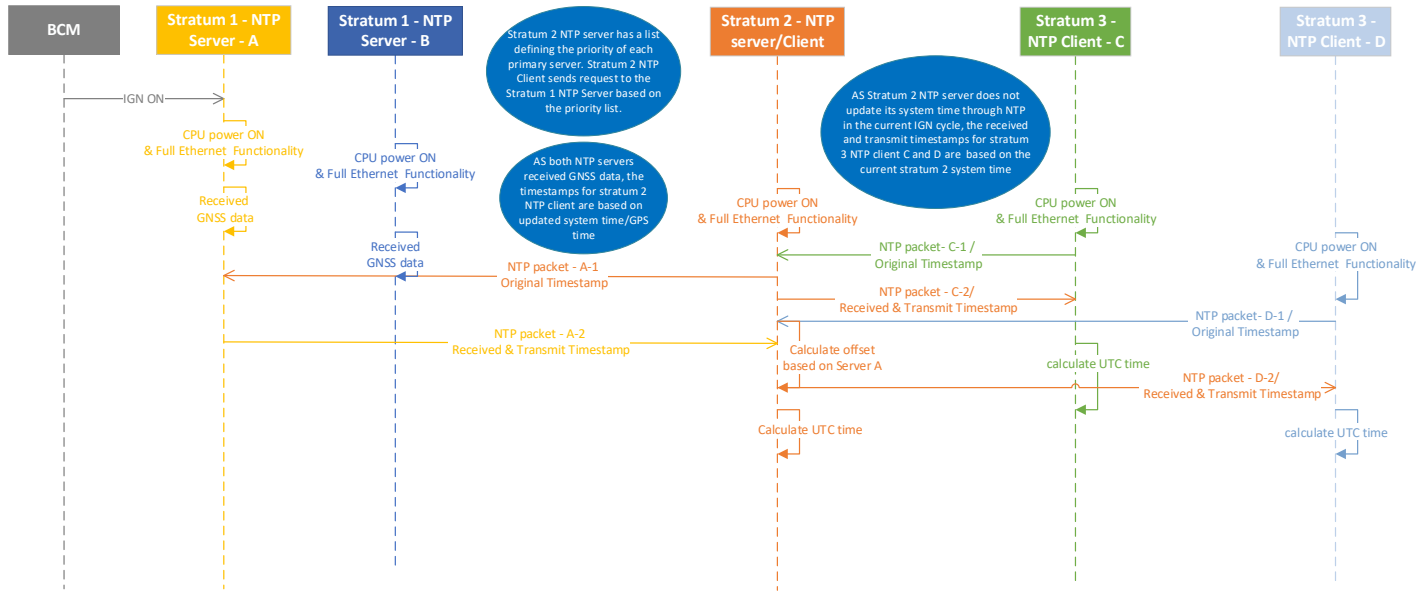
3.1.3.1.2 TSS-SD-REQ-411428/A-Time synchronization and fastest boot up time for Stratum 2



3.1.3.1.3 TSS-SD-REQ-411430/A-Time synchronization and fastest boot up time for Stratum 3



3.1.3.1.4 TSS-SD-REQ-411431/B-Time synchronization with two ECU's in Stratum 1



3.2 TSS-FUN-REQ-411355/B-TSS Security

3.2.1 Requirements

3.2.1.1 TSS-REQ-411356/B-Overview

NTP version 4 offers two authentication methods. 1) Symmetric key approach using pre-shared key, which requires the manual configuration of the client depending on the server. With this approach, simply adding new clients is not possible and changes of the server-side keys result in adjustments to all client. 2) Autokey which showed serious vulnerability. With this approach, attackers can easily break a secured connection and modify the time data in the NTP packets. Due to lack of security in Autokey, it is essential for us to utilize Symmetric Key approach. For this approach, server and client must agree on the Key and Key Identifier to authenticate NTP packet. Keys and related information are specified in a key file which must be distributed and stored using secure means beyond the scope of the NTP protocol itself. This process is called Key Establishment and Key Exchange which will be discussed later in Req-411379-Key Establishment.

Figure 5 shows the overall flow of the NTP with Symmetric Key authentication mechanism. As shown, a single NTP node (centralized key manager /ECG) is responsible for generating a unique key per client (node) per session (ignition cycle). In addition, centralized key manager is responsible to distribute a key (per client) over SOA/TLS with the accessing clients in

each session (session is defined as a power cycle/re-boot of ECU). Once keys being distributed, servers and clients can perform authenticated time synchronization using pre-shared keys and defined encryption algorithm.

Time synchronization proceeds with the indicated NTP server. The client sends the NTP request and the NTP server provides requested information along with Key Identifier and MAC. The NTP client authenticates the NTP response by using key identifier to extract the key from key file, pre-defined encryption algorithm, and MAC. Once, the NTP response passes all cryptographic checks, the NTP client calculates the required statistics and update its system time. If the NTP packet has been modified in any way or replayed by an intruder, it will fail one or more of these checks and be discarded.

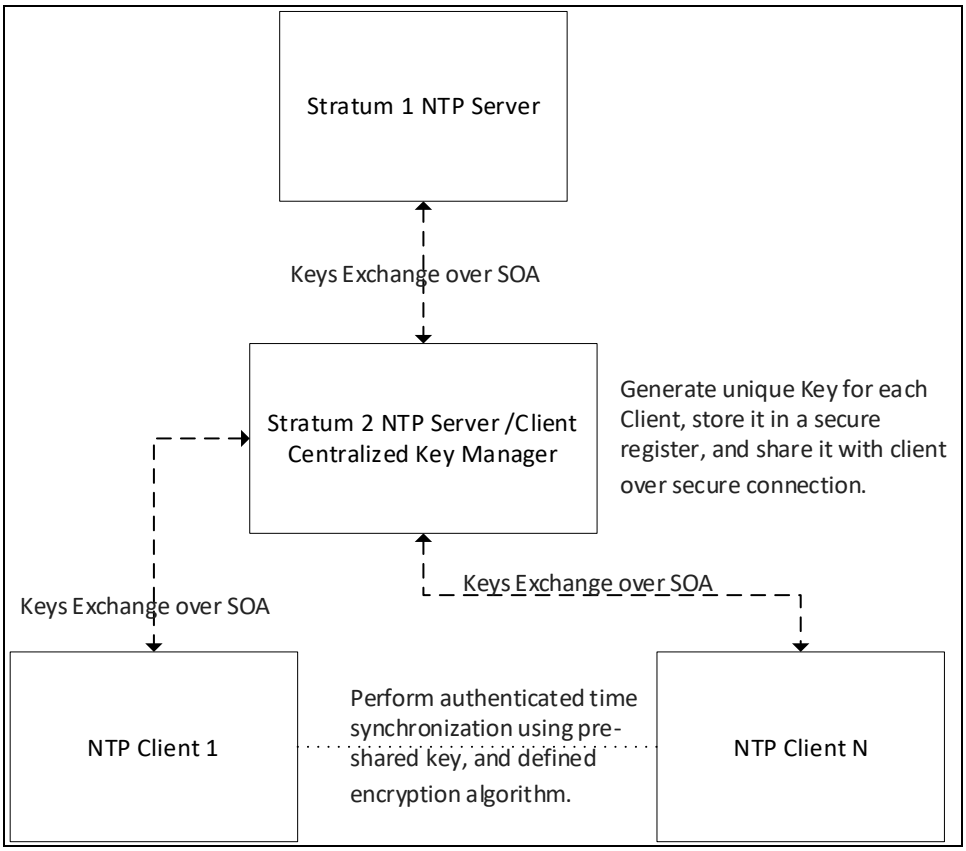


Figure 5 High level interaction of NTP with Symmetric Key

3.2.1.2 TSS-REQ-411379/B-Key Establishment

It is a mechanism for establishing key material for use with the Message Digest filed for NTPv4. The following sections outline the roles and responsibilities of each node in the network.

3.2.1.3 TSS-REQ-436258/A-Stratum 1 ECU Tasks for Key Establishment Phase

Stratum 1 ECU shall follow the steps below for Key Exchange:

1. Immediately after IGN ON/re-boot, ECU shall listen to the defined SOA topic (please refer to Table 10) to receive new Key from Centralized Key Manager.
    a. ECU shall continue to use the old key until new key is provided from Stratum 2 ECU. With this approach, there will be no interruptions for time synchronization.
        i. Once new key is received, ECU shall use the new key for NTP packet exchange.
2. If NTP symmetric Key file got corrupted, unreadable, missing, or TSS configuration is being reset (i.e. sudden momentary power loss or power cut or end of line), then it shall send the SOA request to Stratum 2 obtaining new Key.
    a. Please note that the Stratum 1 ECU shall not continue time synchronization with unauthenticated timestamp until new Key established.
    b. Please note that the Stratum 1 ECU shall refer to Table 11 for SOA topic and Key request.
    c. The structure of SOA message payload shall be as follow:

```
enum NtpSymmetricalKeyStatus
        {
                ECG_TIME_REQUEST_KET = 0;
                ECG_TIME_ACKKNOWLEDGE_KEY = 1;
        }
        Message NtpSymmetricalKeyRequest{
                NtpSymmetricalKeyStatus status;
        }
```

3.2.1.4 TSS-REQ-436259/A-Stratum 2 ECU Tasks for Key Establishment Phase

Stratum 2 ECU shall act as Centralized Key Manager and provide a unique Key to each client (Stratum 1 & 3 ECUs) per session. Please note that we refer to ECG/Stratum 2 ECU as *Key Manager client* in this section.

Stratum 2 ECU shall follow the steps below for Key Exchange process:

1. Immediately after IGN ON/re-boot, Key Manager client shall request new Key/s (one Key per client) to be generated from FNV Key Manager.
    a. FNV Key Manager is responsible to define an API to address this request.
    b. FNV Key Manager shall share the Key/s with Key Manager Client/Time Service.
    c. Key Manager Client is responsible to assign a unique Key to each client once they are accessible.
    d. FNV Key Manger shall persist Key/s across IGN cycle.
2. If any one of the clients is present and online, then Key Manager Client shall use the defined SOA topic to share the Key with the accessing client.
    a. Please note that Key Manager Client shall refer to Table 10 to use the defined SOA topic for exchanging key with the accessing client.
    b. The structure of SOA message payload shall be as follow:

```
Message NtpSymmeticKey{
        optional bytes KeyData = 1;
}
```

3. Once Key is being shared with the accessing client, the Time Service shall store the new Key in NTP configuration file.
    a. Please note that it is the Time Service responsibility to know which ECU uses which Key.
    b. Once accessing client communicate with the new key, the Time Service shall remove the old Key that is being used on start-up.

If the NTP symmetric Key file in Stratum 2 ECU got corrupted, then it shall follow step 1 to 3 as explained above.

- Please note that the Stratum 2 ECU can request for Key re-generation and re-assignment at any moment.

If Stratum 2 ECU receives Key request from Client, it shall follow the steps below:

1) Key Manager client shall request Key from FNV Key Manager.
2) Key Manager Client shall use the defined SOA topic (Table 10) to share the Key with the accessing client.
3) Once Key is being shared with the client, the Time Service shall store the new Key in NTP configuration file.
    a. Once client communicate with the new key, the Time Service shall remove the old Key.

3.2.1.5 TSS-REQ-436260/A-Stratum 3 ECU Tasks for Key Establishment Phase

Stratum 3 ECU shall follow the steps below for Key Exchange:

1) Immediately after IGN ON/re-boot, ECU shall listen to the defined SOA topic (please refer to Table 10) to receive new Key from Centralized Key Manager.
    a. ECU shall continue to use the old key until new key is provided from Stratum 2 ECU. With this approach, there will be no interruptions for time synchronization.
        i. Once new key is received, ECU shall use the new key for NTP packet exchange.
2) If NTP symmetric Key file got corrupted, unreadable, missing, or TSS configuration is being reset (i.e. sudden momentary power loss or power cut), then Stratum 1 ECU shall send the SOA request to Stratum 2 obtaining new Key.
    a. Please note that the Stratum 3 ECU shall not continue time synchronization with unauthenticated timestamp until new Key established.

b. Please note that the Stratum 3 ECU shall refer to Table 11 for SOA Topic and Key request.
c. The structure of SOA message payload shall be as follow:

```
enum NtpSymmetricalKeyStatus
{
ECG_TIME_REQUEST_KET = 0;
ECG_TIME_ACKKNOWLEDGE_KEY = 1;
}
Message NtpSymmetricalKeyRequest{
    NtpSymmetricalKeyStatus status;
}
```

Client	Topic	Group_id	Platform-client-descriptor
TCU	SERVICES/DATA/ECG/TIME/SECURITY/TCU/KEY	ECG_TIME_SECURITY_TCU	TBD
ADAS	SERVICES/DATA/ECG/TIME/SECURITY/ADAS/KEY	ECG_TIME_SECURITY_ADAS	TBD
SYNC	SERVICES/DATA/ECG/TIME/SECURITY/SYNC/KEY	ECG_TIME_SECURITY_SYNC	TBD
AR	SERVICES/DATA/ECG/TIME/SECURITY/AR/KEY	ECG_TIME_SECURITY_AR	TBD

Table 10 SOA Topics for Key Exchange

Client	Topic	Group_id	Platform-client-descriptor
TCU	SERVICES/REQUEST/ECG/TIME/SECURITY/TCU/KEYREFRESH	ECG_TIME_SECURITY_TCU	TBD
ADAS	SERVICES/REQUEST/ECG/TIME/SECURITY/ADAS/KEYREFRESH	ECG_TIME_SECURITY_ADAS	TBD
SYNC	SERVICES/REQUEST/ECG/TIME/SECURITY/SYNC/KEYREFRESH	ECG_TIME_SECURITY_SYNC	TBD
AR	SERVICES/REQUEST/ECG/TIME/SECURITY/AR/KEYREFRESH	ECG_TIME_SECURITY_AR	TBD

Table 11 SOA Topics for Key Request

3.2.1.6 TSS-REQ-411386/B-Secure Time Synchronization

Figure 6 shows the secure NTP time synchronization messaging between Client and Server in a Network. As shown below, the process involved two steps, which must be implemented as follow:

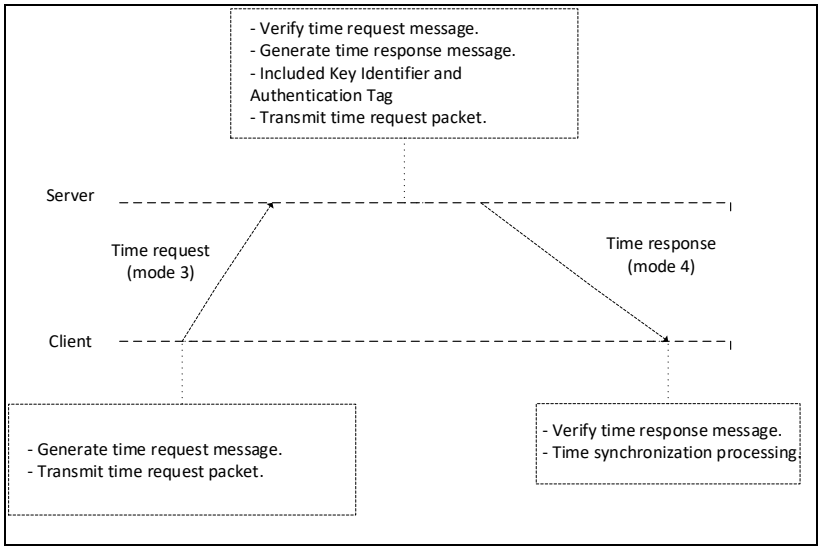


Figure 6 Protected NTP Time Synchronization messages

3.2.1.6.1 TSS-REQ-411388/B-Server Time Response

Upon receiving an NTP request, the server shall respond with the requested information. In addition, the NTP server shall apply AES-CMAC as described in RFC 8573 over all fields in the NTP header that are present in the NTP packet. The MAC key for NTP must be an AES-128 key that is 128 bits in length, and the resulting MAC tag must be at least 128 bits in length. The NTP server shall include Key Identifier and MAC in their response to be used by client for authentication.

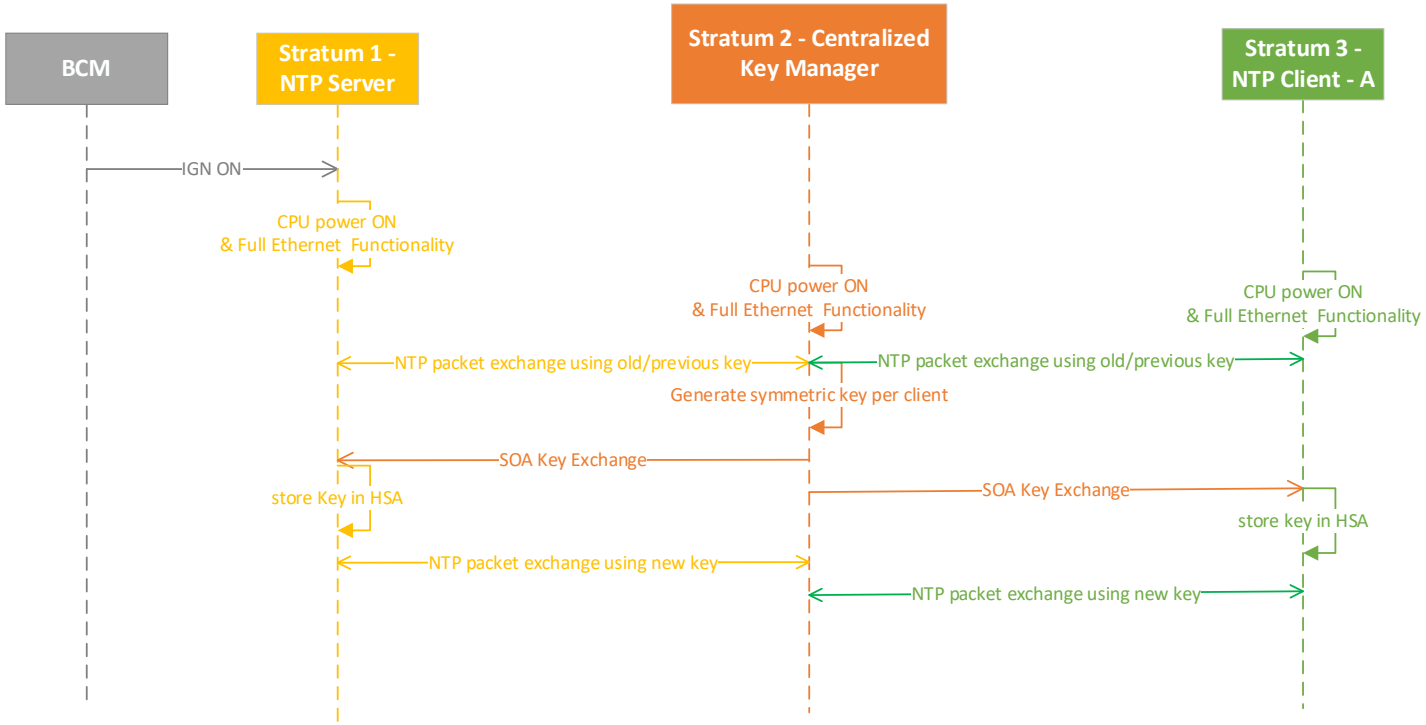
3.2.1.6.2 TSS-REQ-411389/B-Client Time Synchronization

Upon receiving an NTP response, the client shall use the Key Identifier filed to extract the Key from Key File, and then use the Key to authenticate the packet based on the pre-defined encryption algorithm. If the key is valid and authentication succeed, the client shall proceed with time synchronization process and offset calculation. If the client is unable to validate the key or authenticate the request, it shall discard the packet.

3.2.2 White Box Views

3.2.2.1 Sequence Diagrams

3.2.2.1.1 TSS-SD-REQ-431749/A-Key Exchange Symmetric Key Authentication



3.3 TSS-FUN-REQ-411390/B-UTC Offset

3.3.1 Use Cases

3.3.1.1 TSS-UC-REQ-431726/A-UTC offset for Fleet/UBI

Actors	ECG Time Management
Pre-conditions	Ignition is on KEY ON position ECG Time Management is operational
Scenario Description	Insurance companies would like to know the local time of the vehicle in order to provide drivers the best possible rates based off their driving conditions/behaviors and UTC offset can provide us information to calculate accurate local time. Please refer to section REQ-411391-UTC Offset for detailed implementation information.
Post-conditions	N/A
List of Exception Use Cases	Last known UTC offset can be used in the absence of GNSS data.
Interfaces	Vehicle System Interface

3.3.2 Requirements

3.3.2.1 TSS-REQ-411391/B-UTC Offset

UTC offset can provide information for connected features to calculate the vehicle local time. Connect features can populate UTC offset in the FTCP header. To provide UTC offset for connected feature, SYNC shall calculate UTC offset using location data. To accomplish this, the SYNC shall:

- Receive location data (latitude, longitude and UTC time) from location services.
- Send location data to Zone Detect Library that converts location data into time zone ID.
- Send time zone ID to the IANA time zone database that calculates UTC offset using time zone ID.

SYNC shall define new SOA service and publish UTC offset using new topic to the ECG application processor.
SYNC shall publish UTC offset to a retained topic, so that any ECG app can query at any time for the last published value.
SYNC shall use the last known UTC offset and publish it over SOA in case of no GPS fix or GPS signals.

SYNC shall be aware of time transitions and shall update UTC offset upon transitions in time.
SYNC shall persist the UTC offset, and update last know UTC offset upon transitions in time.
SYNC shall persist the UTC offset across ignition cycles and power modes, i.e. full power mode, LPR and Deep sleep.
UTC offset shall always be accurate and available in all vehicle states.

ECG shall divide UTC offset (with second resolution) by 3600, and represent the UTC offset as a 32-bit signed float. ECG shall create an API to store and output UTC offset information.

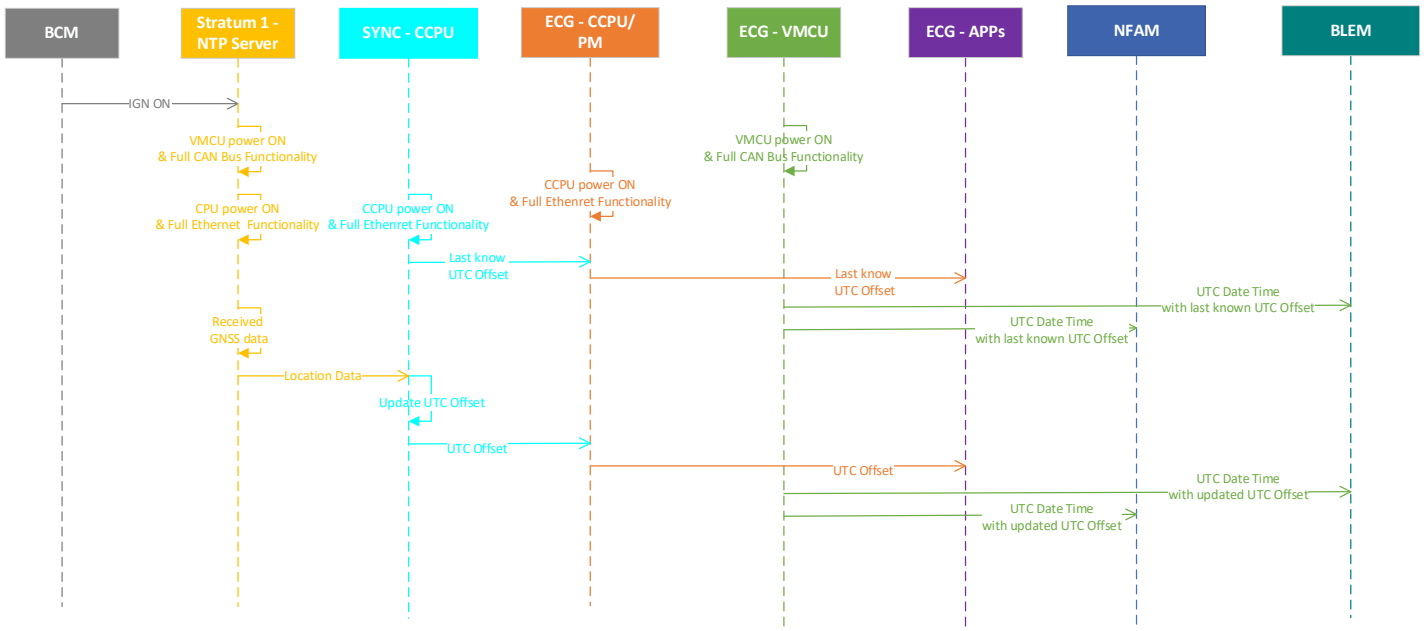
Connected features application can call the defined API to populate the UTC offset in FTCP header using either ModemUTC or internalRTC date/time fields (please refer to Ford Telematics Communication Protocol Common proto for detailed information). Please note that both fields have the same message structure called UTCDateTime in which the UTC offset has been defined as an optional float field.

3.3.3 White Box Views

3.3.3.1 Sequence Diagrams

3.3.3.1.1 TSS-SD-REQ-411433/B-UTC Offset for ECG APPs and Connected ECUs

Please note that the CAN signal will be defined in REQ-411395/B-Introduction to UTC Clock Master.



3.4 TSS-FUN-REQ-411394/B-UTC Date/Time over CAN Bus

3.4.1 Use Cases

3.4.1.1 TSS-UC-REQ-431727/A-UTC for evaluating user key - BLEM

Actors	ECG Time Management
Pre-conditions	Ignition is on KEY ON position ECG Time Management is operational CAN Buses are fully functional
Scenario Description	User requests key from their phone to the Ford backend. Backend communicates to the vehicle in the background and vehicle will send an expiration date/time with the key to the backend for the user to act on once received. The expiration is the time the vehicle gives the user's phone to have access/connection to the vehicle. BLEM needs accurate UTC to check the validity of user Key.
Post-conditions	N/A
List of Exception Use Cases	N/A
Interfaces	Vehicle System Interface

3.4.1.2 TSS-UC-REQ-431728/A-UTC for evaluating public key certificates - NFAM

Actors	ECG Time Management
Pre-conditions	Ignition is on KEY ON position ECG Time Management is operational CAN Buses are fully functional
Scenario Description	Provide drivers with the ability to use an NFC device (cards, wearables, or capable mobile device) to unlock, start and drive a vehicle. Key credential sharing and expiration must be governed via accurate time source, UTC.
Post-conditions	N/A
List of Exception Use Cases	NFAM won't be functional in the following conditions as it does not have accurate UTC: a) Battery disconnects b) Initial power up when vehicle is built/battery connected Once UTC Clock Master provides UTC, NFAM RTC synchronize with it and can properly function.
Interfaces	Vehicle System Interface

3.4.2 Requirements

3.4.2.1 TSS-REQ-411395/B-Introduction to UTC Clock Master

Our in-vehicle platform shall consist of one UTC Clock master providing UTC Date/Time on CAN bus for other ECUs. The CAN based ECUs that receive UTC Date/Time for date/time adjustment will be called **Slave ECU** for the rest of this document. Figure 7 shows the basic time synchronization approach for the CAN based ECUs, but it does not describe the portioning of the system as that can vary between each architecture and carline.

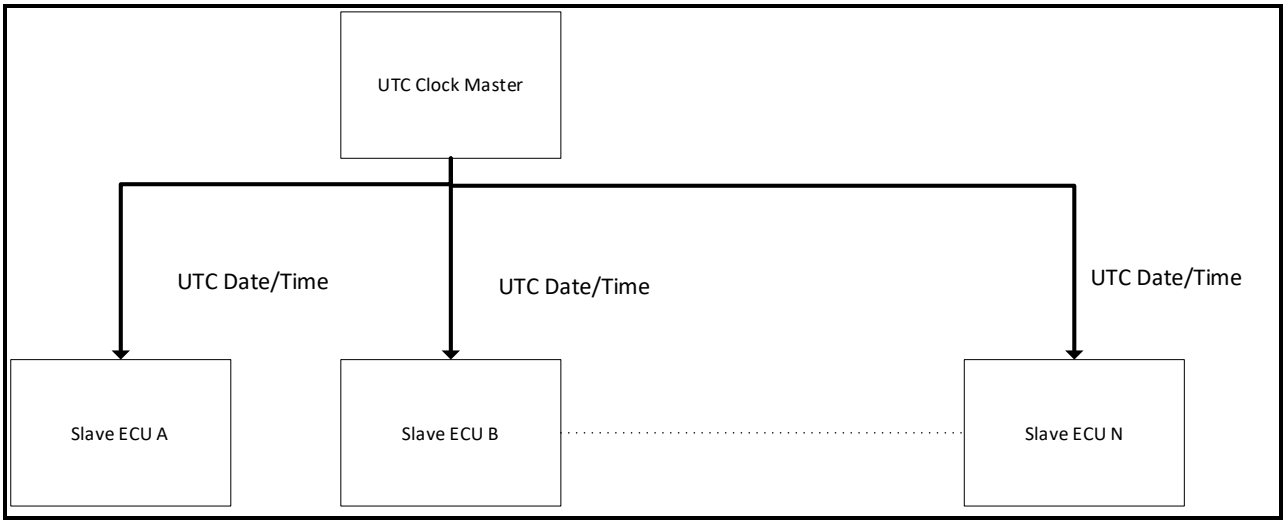


Figure 7. UTC Clock Master Architecture

In FNV3 platform, ECG shall act as UTC Clock Master and provide high precision UTC Date/Time via CAN bus with a fix periodic rate of 1 second. The CAN based ECUs that are interested in receiving UTC time shall listen to the UtcDateTZone_No_Actl signal. The UTC Clock Master shall send the UTC Date/Time in the format shown in Table 12.

Data Field	Length (bits)	Value	Value Range	Description
UTC Year	7	2000 to 2127	0x00= 2000 … 0x7F= 2127	
UTC Month	4	1 to 12	0x1= 1 … 0xC= 12	0 is reserved
UTC Day	5	1 to 31	0x01=1 … 0x1F=31	0 is reserved
UTC Hour	5	0 to 23	0x00= 0 … 0x17= 23	
UTC Minute	6	0 to 59	0x00= 0 … 0x3B= 59	
UTC Second	6	0 to 59	0x00= 0 … 0x3B= 59	
UTC Millisecond	10	0 to 999	0x000= 0 0x3E7= 999	
UTC Offset	1	0 to 1	0= + 1= -	UTC offset time is from -12 to +14

UTC Offset Hour	4	0 to 14	0x0= 0 ... 0xE= 14	
UTC Offset Minute	6	0 to 45	0x00=0 … 0x2D=45	UTC offset is Increments of 15min.
UTC Source	3	0 to 5	0x0= TCU NITZ 0x1= SYNC GNSS 0x2= TCU GNSS 0x3= ECG RTC 0x4= Ford Cloud (NIST) 0x5= Validity	
Reserved	7	0		

Table 12 UTC Date/Time Format

All Slave ECUs shall synchronize their own system time with the UTC Clock Master every time the Master is available for the Slave ECUs.

Table 13 shows the message that is used by the UTC Clock Master to distribute UTC Date/Time to the Slave ECUs. The message and signal names are place holders until reviewed with Netcom at which point actual message and signal names will be updated in this spec.

Message: UTC_Date_Time (Periodicity 1 second)		Size[bits] 64		Event Periodic Message		
Meaning	Size	Start bit	End bit	Offset	Min Hex	Max Hex
UTC_Year	7	0	6	2000	0x00	0x7F
UTC_Month	4	7	10	-	0x1	0xC
UTC_Day	5	11	15	-	0x01	0x1F
UTC_Hour	5	16	20	-	0x00	0x17
UTC_Minute	6	21	26	-	0x00	0x3B
UTC_Second	6	27	32	-	0x00	0x3B
UTC_Millisecond	10	33	42	-	0x000	0x3E7
UTC_Offset	1	43	43	-	0x0	0x1
UTC_Offset_Hour	4	44	47	-	0x0	0xE
UTC_Offset_Minute	6	48	53	-	0x00	0x2D
UTC_Source	3	54	56	-	0x0	0x5
Reserved	7	57	63	-	-	-

Table 13 UTC Date/Time CAN Signal

3.4.2.2 TSS-REQ-411397/B-CAN Bus in Sleep

The UTC Clock Master signal cannot be received by the Slave ECUs if

1) CAN bus is in sleep mode, or
2) Ignition or power Mode is less than or equal to Sleep with RAM Self-Refresh.

In the above two scenarios, the internal Clock of the Slave ECU has to run on its own. Whenever a Slave ECU receives the UTC signal via CAN, the Slave ECU shall set their system time to UTC Date/Time.

3.4.2.3 TSS-REQ-411399/B-CAN Bus Awake

If the CAN BUS is awake, the Slave ECUs must synchronize their internal Clocks with the UTC Date/Time received via CAN from the UTC Clock Master after performing following check.
If the new (received) UTC Date/Time is equal to the previous value (UTC Date/Time), then the Slave ECUs must discard the CAN signal and rely on their internal system time/RTC. Otherwise, they must synchronize their system time with UTC Data/Time received from the UTC Clock Master.

3.4.2.4 TSS-REQ-411400/B-CAN Bus Transition Awake to Sleep

When the CAN Bus transitions to sleep mode the slave internal clock shall be maintained by the Slave ECU if it is required.

3.4.2.5 TSS-REQ-411401/B-CAN Bus Transition Sleep to Awake

When the CAN Bus wakes up, the UTC Clock Master shall begin sending the UTC Date/Time within 20 seconds via CAN to the Slave ECUs. It is critical that, after IGN ON, the UTC Clock Master always sends the latest UTC value in a format shown in Table 12. This is to avoid adjusting with old time values in the Slave ECUs.

3.4.2.6 TSS-REQ-411402/A-Physical Requirements for UTC Clock Master

The UTC Date/Time shall be generated in such a way that it meets the requirement listed in this section.

The Clock-IC used in the UTC Clock Master shall have a minimum accuracy as mentioned in Table 14. Based on the requirement the drift of Real Time Clock (RTC) for ECG shall not exceed +/-20 seconds per day regardless of its power state.

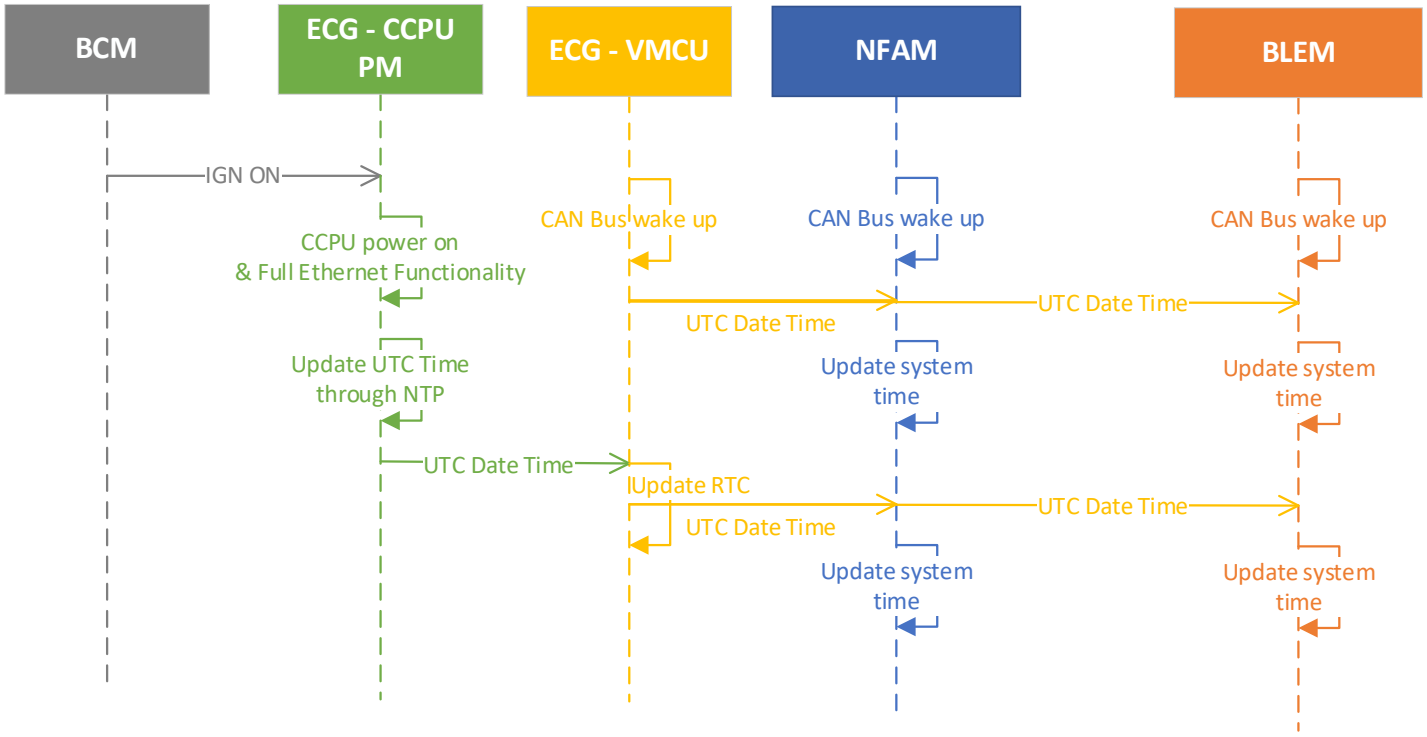
Degrees Celsius min	Degrees Celsius max	Drift
- 40 degrees C	+ 105 degrees C	+/- 20.0 s/day

Table 14 Required Accuracy for UTC Clock Master

3.4.3 White Box Views

3.4.3.1 Sequence Diagrams

3.4.3.1.1 TSS-SD-REQ-411432/A-UTC Date Time Transmission on CAN Bus



3.5 TSS-FUN-REQ-411403/B-UTC and Vehicle Date/Time for Connected Feature Application

3.5.1 Use Cases

3.5.1.1 TSS-UC-REQ-431729/A-UTC Base Point for Fleet/UBI

Actors	ECG Time Management
Pre-conditions	Ignition is on KEY ON position ECG Time Management is operational
Scenario Description	UTC (in vehicle) can potentially jump forward or move backward during a trip. Insurance companies are interested in calculating risk based on driver behavior. Utilizing UTC Base Point with ECG CCPU/AP monotonic timer ensure that there is no jump in timestamp of Fleet/UBI data points. Please refer to section REQ-411409-UTC Base Point
Post-conditions	UTC Base point remains valid for that ignition cycle
List of Exception Use Cases	If UTC date/time is not available at ignition on or it is equal to default value (ECG default system time), then ECG shall wait for UTC time to become available.
Interfaces	Vehicle System Interface

3.5.1.2 TSS-UC-REQ-431730/A-Vehicle Time API for OTA Scheduling

Actors	ECG Time Management & SWUM
Pre-conditions	ECG Time Management and SWUM are operational
Scenario Description	Once user schedule her/his software update time using HMI, the scheduled time will be transmitted to SWUM. SWUM would need to be aware of any vehicle time change in order to update customer's set scheduled accordingly. The following steps need to be taken: 1) Subscribe to system clock change API. 2) Subscribe to Vehicle Time API – Please refer to section REQ-411404-UTC and Vehicle Date/Time API's for detailed implementation information.
Post-conditions	SWUM will update the schedule Timer.
List of Exception Use Cases	N/A
Interfaces	Vehicle System Interface

3.5.1.3 TSS-UC-REQ-431731/A-UTC TIME API for all Vehicle - Ford cloud communication

Please note that use case below only shows the actors from Time perspective. All features and applications for Vehicle to cloud communication will consider actor.

Actors	ECG and TCU Time Management
Pre-conditions	ECG and TCU Time Management is operational
Scenario Description	To populate ModemUTCDateTime in FTCP common header for all connected features. Please refer to section REQ-41144-UTC and Vehicle Date/Time API's for detailed implementation information.
Post-conditions	
List of Exception Use Cases	N/A
Interfaces	Vehicle System Interface

3.5.2 Requirements

3.5.2.1 TSS-REQ-411404/B-UTC and Vehicle Date/Time API's

ECG, SYNC and TCU must define two APIs as follow:

- ***UTC TIME*** API providing UTC date/time for connected features/applications.
- ***VEHICLE TIME*** API providing vehicle date/time in local format for connected features/applications.

Please note that ECG, SYNC, and TCU shall define a configuration strategy to distinguish the supported Time APIs on different architecture.

If ECU's system time and its RTC are synced to UTC, then it must follow requirements that are captured in REQ-411410/B-Vehicle Time Delta to properly implement the above two APIs.
If ECU (SYNC QNX) that is participating in the NTP cannot synchronize its system time or its RTC to UTC time, then it must follow requirements that are captured in REQ-411405/B-UTC Delta to properly implement the above two APIs.

3.5.2.2 TSS-REQ-411405/B-UTC Delta

Fast based ECUs that their RTC and system time are synced with Vehicle Time (Global Clock from BCM – Please refer to Global Clock Strategy Specification for more information) shall define a variable called **UTC Delta** representing time/date difference between ECU system time and UTC date/time provided by GNSS receiver or NTP servers. ECU shall immediately update the UTC Delta:

- Upon change of ECU system time or RTC.

- Upon receiving valid GPS time from GNSS receiver (this is only valid for ECUs in which there is local GNSS receiver and GNSS antenna connection)
- Upon calculating offset through NTP.

ECU shall persist (NVM if needed) UTC Delta across ignition cycles and power modes, i.e. full power mode, LPR and Deep sleep.
ECU shall reset the UTC Delta to the default value of zeros if one of the following conditions is met:

- Battery disconnect which results in resetting the vehicle time to its default time of January 1st, 2000.
- BCM replacement which results in resetting the vehicle time to its default time of January 1st, 2000.
- Current ECU system time is behind previous stored system time.

ECU shall utilize the UTC Delta along with its internal system time/RTC to populate UTC date/time for ***UTC TIME*** API.

3.5.2.3 TSS-REQ-411410/B-Vehicle Time Delta

Fast based ECUs that their RTC and system time are synced with UTC shall define a variable called **Vehicle Time Delta** representing time difference between UTC and Vehicle Time (Global Clock from BCM – Please refer to Global Clock Strategy Specification for more information).
ECU shall calculate the time/date difference between UTC and Global clock, if the time difference exceeds 3 seconds, then ECU shall update the Vehicle Time Delta.
ECU shall immediately update Vehicle Time Delta upon transition in time.
ECU shall persist (NVM if needed) Vehicle Time Delta across ignition cycles and power modes, i.e. full power mode, LPR and Deep sleep.
ECU shall reset the Vehicle Time Delta to the default value of zeros if one of the following conditions is met:

- Battery disconnect which results in resetting the vehicle time to its default time of January 1st, 2000.
- BCM replacement which results in resetting the vehicle time to its default time of January 1st, 2000.
- Current ECU system time is behind previous stored system time.

ECU shall utilize the Vehicle Time Delta along with its internal system time/RTC to populate vehicle time for ***VEHICLE TIME*** API.

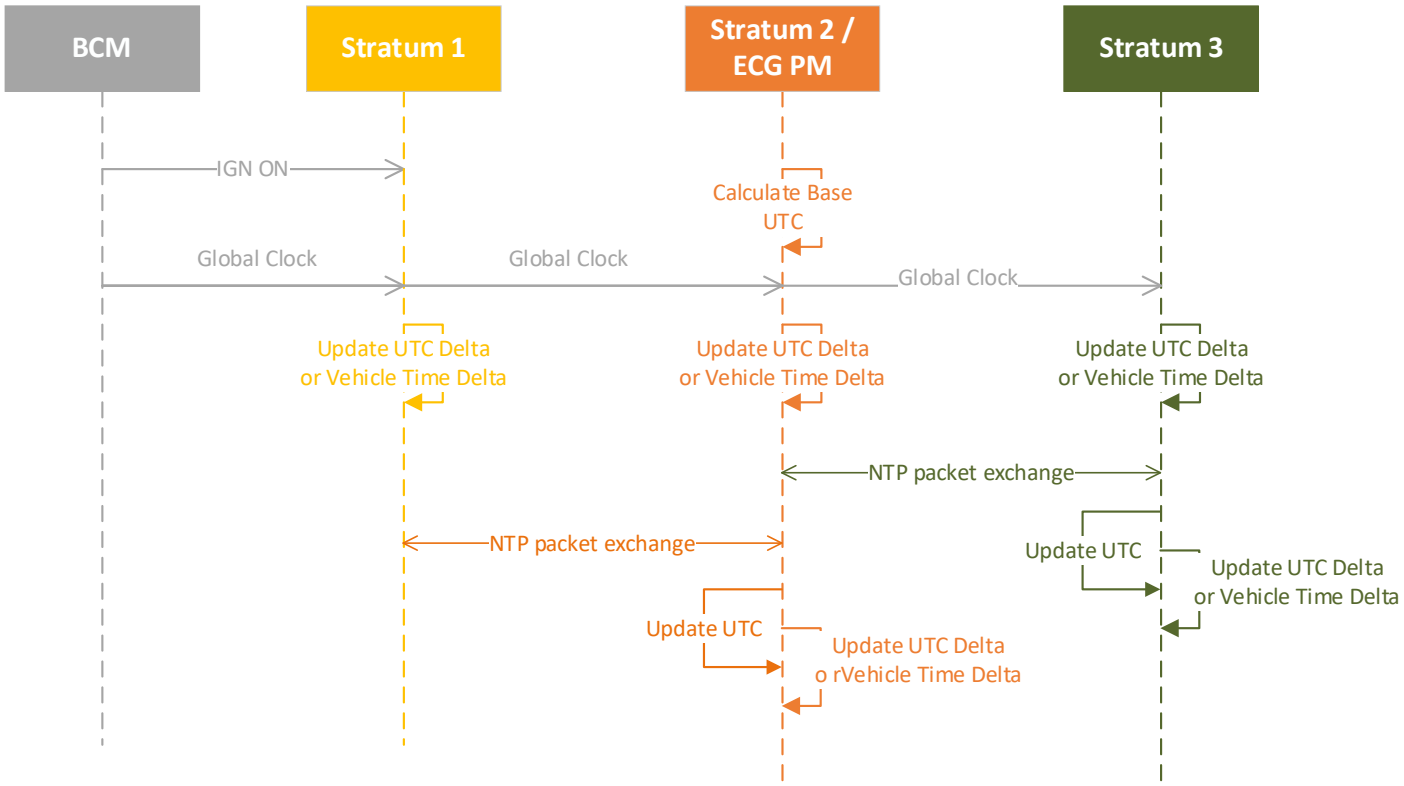
3.5.2.4 TSS-REQ-411409/A-UTC Base Point

ECG shall define an API – ***BASE UTC*** providing UTC date/time at ignition on for connected features/applications.
ECG shall utilize its system time or its RTC at ignition on to populate UTC base point for ***BASE UTC*** API.
ECG shall update the UTC base point only at ignition on or when a default UTC base point is detected.
If UTC date/time is not available at ignition on or it is equal to default value (ECG default system time), then ECG shall wait for UTC time to become available. As soon as UTC time becomes available, ECG shall subtract the UTC time with current VMCU monotonic timer to calculate UTC base point.
If UTC date/time is not available at ignition on or it is equal to default value, ECG shall output zeros for ***BASE UTC*** API.

3.5.3 White Box Views

3.5.3.1 Sequence Diagrams

3.5.3.1.1 TSS-SD-REQ-411434/B-UTC Delta and Vehicle Time Delta



4 Appendix: Reference Documents

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