

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		1
			No.	1.2

トヨタ自動車株式会社

# 車載器-センタ間通信標準仕様書

## 1.2 版

社内配布先/TMC	Copy
コネクティッド基盤開発部 第 11 電子開発室	1
コネクティッド先行開発部 InfoTech	1
コネクティッド基盤開発部 スマートセンター開発室	1
情報セキュリティ推進室	1
制御電子プラットフォーム開発部 制御ネットワーク・アーキ開発室	1
制御電子プラットフォーム開発部 SDA 車両開発推進室	1
制御電子プラットフォーム開発部 BR 電子開発改革室	1
コネクティッド先行開発部 デジタルコクピットソフト開発室	1
コネクティッド基盤開発部 第 13 電子開発室	1
OTA 推進室	1
バリューチェーン基盤開発部	1
先進プロジェクト推進部	1
ソフトウェアファースト	1
先進電子開発部	1
制御電子システム開発部 車両運動電子システム開発室	1
制御電子システム開発部 電子性能開発室	1
制御電子システム開発部 第 2 試験課	1
自動運転・先進安全開発部 第 2 開発室	1

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		2
			No.	1.2

社外配布先		Copy
	Woven Planet	1
	Woven Core	1
	Woven Alpha	1
	スズキ株式会社	1
	株式会社 S U B A R U	1
	ダイハツ工業株式会社	1
	マツダ株式会社	1

### 作成部署

	コネクティッド基盤開発部 第 1 1 電子開発室	コネクティッド基盤開発部 スマートセンター開発室	コネクティッド先行開発部 InfoTech
担当者	片山	鈴木（悟）	芝崎
承認者	前田	外谷	前田

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		3
			No.	1.2

## 目次

1. 目的・位置づけ .....	10
1.1. 目的 .....	10
1.2. 位置づけ .....	10
2. 用語定義 .....	11
3. 適用範囲 .....	12
3.1. 適用 ECU 一覧 .....	12
3.2. 適用電子 PF .....	14
3.3. 適用サーバー一覧 .....	14
3.4. 適用仕向一覧 .....	14
4. システム構成 .....	15
5. ユースケース .....	16
6. 通信要件 .....	18
6.1. 通信プロトコル .....	18
6.1.1. 通信プロトコルの選定 .....	18
6.1.2. 通信プロトコルとユースケース .....	20
6.1.3. プロトコルスタック .....	21
6.2. セキュリティ .....	22
6.2.1. トランスポートレイヤセキュリティ .....	22
6.2.2. クライアント認証 .....	22
6.2.3. 証明書 .....	22
6.3. セッション管理 .....	22
6.4. 車両識別 .....	22
7. 通信仕様 .....	23
7.1. トランスポートレイヤセキュリティ .....	23
7.1.1. TLS .....	23
7.1.1.1. 基本仕様 .....	23
7.1.1.2. 暗号化アルゴリズム .....	23
7.1.1.2.1. 暗号化スイート .....	23
7.1.1.2.2. 鍵交換方式 .....	23
7.1.1.2.3. 署名方式 .....	23
7.1.1.3. クライアント認証 .....	24
7.1.1.4. セッション管理 .....	25
7.1.1.5. 証明書管理 .....	26

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		4
			No.	1.2

7.1.1.6.	基本シーケンス	29
7.1.1.7.	異常系シナリオ	33
7.2.	アプリケーションプロトコル	37
7.2.1.	アプリケーションデータ	37
7.2.2.	HTTP/1.1	38
7.2.2.1.	基本仕様	38
7.2.2.2.	メッセージフォーマット	38
7.2.2.3.	ヘッダ定義	40
7.2.2.4.	レスポンスステータスコード	40
7.2.2.5.	タイムアウト仕様	42
7.2.2.6.	リトライ仕様	42
7.2.2.7.	セッション管理	42
7.2.2.8.	基本シーケンス	43
7.2.2.8.1.	GET	43
7.2.2.8.2.	POST	44
7.2.2.9.	API 仕様	45
7.2.2.9.1.	データアップロード	45
7.2.2.9.2.	センタからのプッシュ	48
7.2.2.9.3.	車両からセンタへの API 呼出し	48
7.2.2.9.4.	データダウンロード	51
7.2.3.	HTTP/2	53
7.2.3.1.	基本仕様	53
7.2.3.2.	メッセージフォーマット	53
7.2.3.3.	ヘッダ定義	55
7.2.3.4.	レスポンスステータスコード	55
7.2.3.5.	タイムアウト仕様	55
7.2.3.6.	リトライ仕様	55
7.2.3.7.	セッション管理	55
7.2.3.8.	基本シーケンス	56
7.2.3.8.1.	GET	56
7.2.3.8.2.	POST	57
7.2.3.9.	API 仕様	58
7.2.3.9.1.	データアップロード	58
7.2.3.9.2.	センタからのプッシュ	58
7.2.3.9.3.	車両からセンタへの API 呼出し	61
7.2.3.9.4.	データダウンロード	61

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		5
			No.	1.2

7.2.4.	gRPC.....	62
7.2.4.1.	基本仕様 .....	62
7.2.4.2.	メッセージフォーマット.....	62
7.2.4.3.	ヘッダ定義 .....	64
7.2.4.4.	API 仕様.....	64
7.2.4.4.1.	データアップロード .....	65
7.2.4.4.2.	センタからのプッシュ.....	67
7.2.4.4.3.	車両からセンタへの API 呼出し.....	69
7.2.4.4.4.	データダウンロード.....	71
7.2.4.5.	レスポンスステータスコード.....	73
7.2.4.6.	セッション管理 .....	75
7.2.5.	MQTT.....	76
7.2.5.1.	基本仕様 .....	76
7.2.5.2.	メッセージフォーマット.....	76
7.2.5.3.	ヘッダ定義 .....	78
7.2.5.4.	レスポンスステータスコード.....	78
7.2.5.5.	タイムアウト仕様 .....	81
7.2.5.6.	リトライ仕様 .....	81
7.2.5.7.	セッション管理 .....	81
7.2.5.8.	基本シーケンス.....	83
7.2.5.9.	API 仕様.....	86
7.2.5.9.1.	データアップロード .....	86
7.2.5.9.2.	センタからのプッシュ.....	89
7.2.5.9.3.	車両からセンタへの API 呼出し.....	92
7.2.5.9.4.	データダウンロード.....	92
7.3.	車両識別 .....	95
Appendix 1	.....	96
1.	データフォーマット .....	96
2.	ファイル名 .....	97
3.	文字コード.....	98
4.	通信シーケンス.....	99
4.1.	データ通信フロー.....	99
4.1.1.	データアップロード .....	99
4.1.2.	センタからのプッシュ.....	103
4.1.3.	車両からセンタへの API 呼出し .....	106
4.1.4.	データダウンロード .....	109

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		6
			No.	1.2

<b>4.1.5. エラー時シーケンス</b>	113
Appendix 2	117
<b>1. コンフィギュレーション</b>	117
<b>1.1. HTTP/1.1</b>	117
<b>1.1.1. Base URL</b>	117
<b>1.1.2. ポート番号</b>	117
<b>1.2. HTTP/2</b>	117
<b>1.2.1. Base URL</b>	117
<b>1.2.2. ポート番号</b>	117
<b>1.3. gRPC</b>	117
<b>1.3.1. Base URL</b>	117
<b>1.3.2. ポート番号</b>	117
<b>1.4. MQTT</b>	118
<b>1.4.1. Base URL</b>	118
<b>1.4.2. ポート番号</b>	118
<b>2. 認証と接続構成</b>	118
<b>変更履歴</b>	121

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		7
			No.	1.2

## 表一覧

表 2-1 用語定義.....	11
表 3-1 適用電子 PF.....	14
表 5-1 ユースケース.....	17
表 6-1 通信プロトコルの比較.....	19
表 6-2 通信プロトコルとユースケースに対する適性.....	20
表 7-1 暗号化スイート.....	23
表 7-2 TLS 1.3 フルハンドシェイク時のメッセージ.....	30
表 7-3 TLS1.3 セッション再開 (1-RTT)のメッセージ.....	32
表 7-4 サーバ認証における異常系シナリオの対応内容.....	34
表 7-5 クライアント認証における異常系シナリオの対応内容.....	36
表 7-6 HTTP のデータと目的・用途.....	37
表 7-7 MQTT のデータと目的・用途.....	37
表 7-8 HTTP/1.1 レスポンスステータスコード.....	40
表 7-9 データアップロードの API 仕様 (OpenAPI).....	45
表 7-10 車両からセンタへの API 呼出しにおける API 仕様 (OpenAPI).....	48
表 7-11 データダウンロードにおける API 仕様 (OpenAPI).....	51
表 7-12 センタからのプッシュにおける API 仕様 (OpenAPI).....	58
表 7-13 データアップロードの API 仕様 (gRPC Request).....	65
表 7-14 データアップロードの API 仕様 (gRPC Response).....	65
表 7-15 センタからのプッシュの API 仕様 (gRPC Response).....	67
表 7-16 車両からセンタへの API 呼出しの API 仕様 (gRPC Request).....	69
表 7-17 車両からセンタへの API 呼出しの API 仕様 (gRPC Response).....	69
表 7-18 データダウンロードの API 仕様 (gRPC Request).....	71
表 7-19 データダウンロードの API 仕様 (gRPC Response).....	71
表 7-20 レスポンスステータスコードとリトライ可否.....	73
表 7-21 Reason Codeとリトライの可否 (CONNACK).....	79
表 7-22 Reason Codeとリトライの可否 (PUBACK).....	80
表 7-23 Reason Codeとリトライの可否 (SUBACK).....	80
表 7-24 Reason Codeとリトライの可否 (UNSUBACK).....	81
表 7-25 リトライが必要なユースケースおよびリトライ回数、リトライ間隔.....	81
表 7-26 データアップロードの API 仕様 (AsyncAPI).....	86
表 7-27 センタからのプッシュにおける API 仕様 (AsyncAPI).....	89
表 7-28 データダウンロードにおける API 仕様 (AsyncAPI).....	92
表 A2-1 代表的なユースケースと接続形態.....	118

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		8
			No.	1.2

## 図一覧

図 1-1 本書の位置づけ .....	10
図 3-1 適用 ECU と本書の適用範囲 .....	12
図 3-2 適用 ECU と適用範囲（プロトコルスタック） .....	13
図 4-1 システム構成 .....	15
図 6-1 プロトコルスタック .....	21
図 7-1 CRL および OCSP のキャッシュコントロール基本フロー .....	27
図 7-2 CRL/OCSP 失効確認 基本フロー .....	28
図 7-3 TLS1.3 フルハンドシェイク時のシーケンス .....	29
図 7-4 TLS1.3 セッション再開（1-RTT）シーケンス .....	31
図 7-5 サーバ認証における異常系シナリオのシーケンス .....	33
図 7-6 クライアント認証における異常系シナリオのシーケンス .....	35
図 7-7 HTTP/1.1 におけるメッセージフォーマット .....	39
図 7-8 HTTP/1.1 の基本シーケンス（GET） .....	43
図 7-9 HTTP/1.1 の基本シーケンス（POST） .....	44
図 7-10 HTTP/2 におけるメッセージフォーマット .....	54
図 7-11 HTTP/2 の基本シーケンス（GET） .....	56
図 7-12 HTTP/2 の基本シーケンス（POST） .....	57
図 7-13 gRPC におけるメッセージフォーマット .....	63
図 7-14 MQTT における認証、認可および機密保持 .....	76
図 7-15 MQTT におけるメッセージフォーマット .....	77
図 7-16 MQTT の基本シーケンス（データアップロード） .....	83
図 7-17 MQTT の基本シーケンス（センタからのプッシュ） .....	84
図 7-18 MQTT の基本シーケンス（データダウンロード） .....	85
図 A-1 データアップロードシーケンス（HTTP/1.1） .....	99
図 A-2 データアップロードシーケンス（HTTP/2） .....	100
図 A-3 データアップロードシーケンス（gRPC） .....	101
図 A-4 データアップロードシーケンス（MQTT） .....	102
図 A-5 センタからのプッシュシーケンス（HTTP/2） .....	103
図 A-6 センタからのプッシュシーケンス（gRPC） .....	104
図 A-7 センタからのプッシュシーケンス（MQTT） .....	105
図 A-8 車両からセンタへの API 呼出しシーケンス（HTTP/1.1） .....	106
図 A-9 車両からセンタへの API 呼出しシーケンス（HTTP/2） .....	107
図 A-10 車両からセンタへの API 呼出しシーケンス（gRPC） .....	108
図 A-11 データダウンロードシーケンス（HTTP/1.1） .....	109
図 A-12 データダウンロードシーケンス（HTTP/2） .....	110



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		9
			No.	1.2

図 A-13 データダウンロードシーケンス (gRPC) .....	111
図 A-14 データダウンロードシーケンス (MQTT) .....	112
図 A-15 【エラー時のシーケンス】データアップロード (HTTP/1.1) .....	113
図 A-16 【エラー時のシーケンス】データアップロード (HTTP/2) .....	114
図 A-17 【エラー時のシーケンス】データアップロード (gRPC) .....	115
図 A-18 【エラー時のシーケンス】データアップロード (MQTT) .....	116
図 A2-1 車両とセンタ間の認証の全体構成.....	120
図 A2-2 センタからのプッシュの接続構成.....	120

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	10
		No.	1.2

## 1. 目的・位置づけ

### 1.1. 目的

本書は、次期車両向け 24 年頭出しのセンタとそれに接続する ECU 間でデータ通信を行うアプリケーションが共通で対応すべき仕様を定義したものである。

車載器に実装されるセンタ通信を行うアプリケーションの増加が見込まれる中、アプリケーションが利用可能な汎用的なプロトコル仕様と、車載-センタ間での共通の認証仕様を規定することを目的とする。

なお、本書は次世代の車載通信機の技術仕様を共同で開発する OEM にも配布し、TSC に接続する各 OEM の仕様を共通にすることも目的とする。

### 1.2. 位置づけ

本書の位置づけについて、他の仕様書との関連を図 1-1 本書の位置づけに示す。

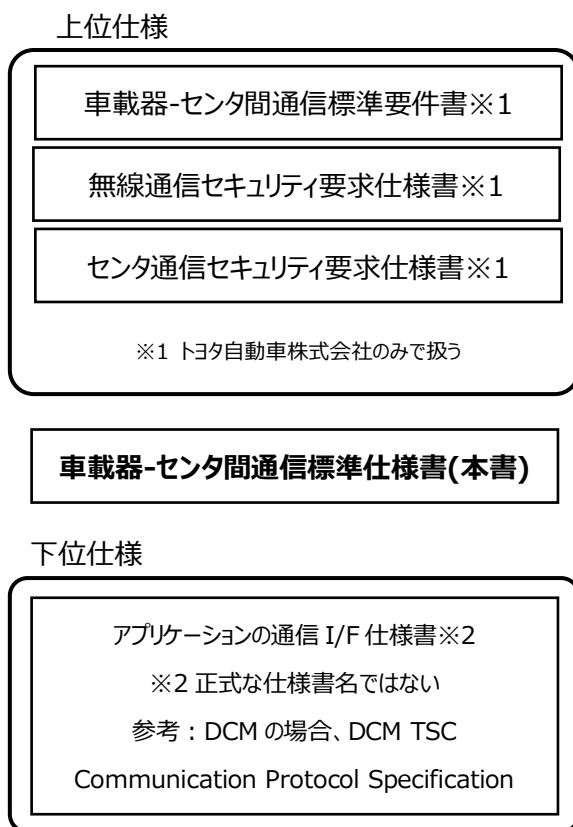


図 1-1 本書の位置づけ

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	11
		No.	1.2

## 2. 用語定義

当仕様書内で使われる用語を 表 2-1 に定義する。

表 2-1 用語定義

用語	説明
ADAS	Advanced Driver-Assistance System
ADF	Air Data Feed
ALB	Application Load Balancer
ALPN	Application Layer Protocol Negotiation
API	Application Programming Interface
CDP	CRL Distribution Point
CRL	Certificate Revocation List
DCM	Data Communication Module
DHE	Diffie-Hellman ephemeral
ECDHE	Elliptic curve Diffie-Hellman ephemeral
ECDSA	Elliptic Curve Digital Signature Algorithm
ECU	Electronic Control Unit
HTTP	Hypertext Transfer Protocol
IDL	Interface Description Language
IMEI	International Mobile Equipment Identity
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
mTLS	Mutual TLS
NEV	New Energy Vehicle
NLB	Network Load Balancer
OAuth	Authorization framework (RFC6749、6750)
OCSP	Online Certificate Status Protocol
QoS	Quality of Service
RPC	Remote Procedure Call
RSASSA PKCS#1	RSASSA Public Key Cryptography Standards
RSASSA-PSS	RSASSA Probabilistic Signature Scheme
SNI	Server Name Indication
TBDC	TOYOTA Big Data Center
TSC	TOYOTA Smart Center
TSP	Telematics Service Providers
UTC	Universal Time Coordinated
V2X	Vehicle to X
VIN	Vehicle Identification Number
電子 PF	電子プラットフォーム (Platform)

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		12
			No.	1.2

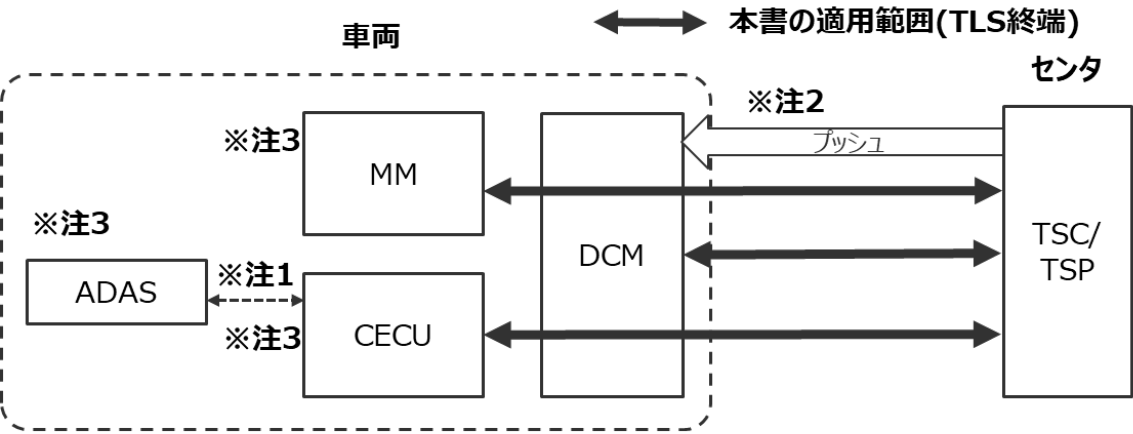
3. 適用範囲

3.1. 適用 ECU 一覧

次期車両向け 24 年頭出しの TSCと通信するアプリケーションを搭載する ECUとする。  
具体的には、DCM、マルチメディア(MM)、Central ECU(CECU)、ADAS ECUとする。

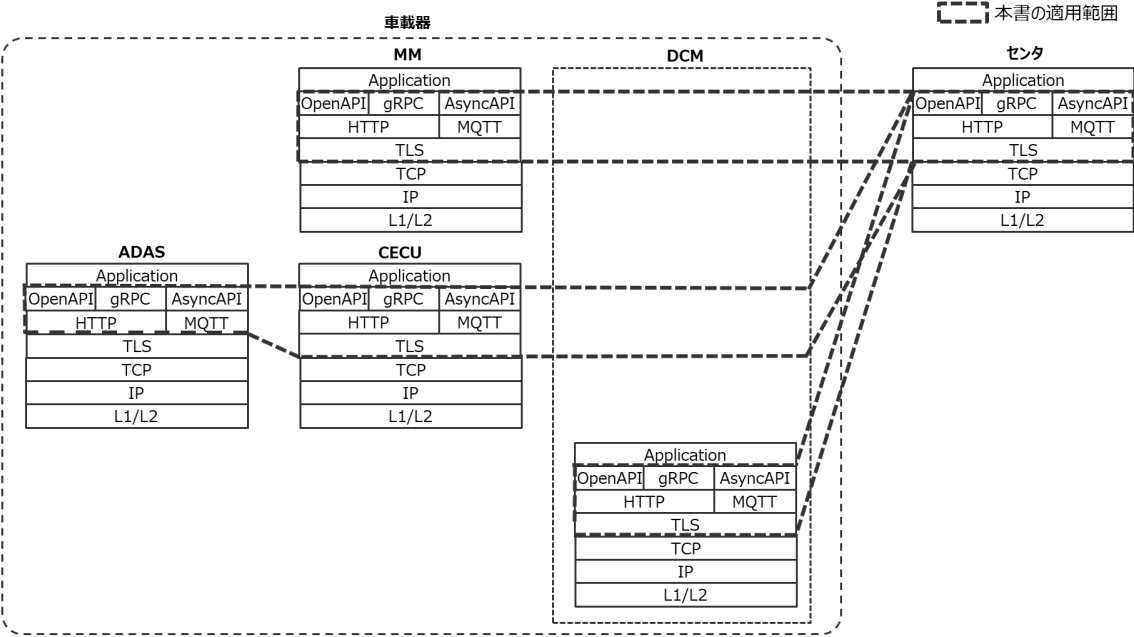
適用 ECUと本書の適用範囲を 図 3-1、図 3-2 に示す。

なお、本書の適用範囲は、TLS より上位のプロトコルとする。( 6.1.3 プロトコルスタック 参照)



- 注 1 ADASは、CECU経由で、本書で規定する通信仕様を満たす前提とする。  
なお、ADASとCECU間の分担・通信仕様については本書では規定しない。
- 注 2 センタから車載器へのプッシュについては、DCMが受付けを行い、DCMから他のECUに通知することを前提とする。  
そのため、本書では、DCMとセンタ間の通信仕様のみを規定する。  
なお、DCMから他のECUへの通信仕様については本書では規定しない。
- 注 3 車両のシステム構成によっては、MM, CECU, ADASが無い車両も存在する。  
該当ECUの有無については、各ECUの仕様書を参照すること。

図 3-1 適用 ECU と本書の適用範囲



※本図は適用範囲を示すものであり、図中のプロトコル(HTTP、gRPC、MQTT等)は各ECUで実際に使用されるものとは異なる場合がある。  
各ECUで実際に使用されるプロトコルについては、各ECUの仕様書を参照のこと。

図 3-2 適用 ECU と適用範囲 (プロトコルスタック)

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		14
			No.	1.2

### 3.2. 適用電子 PF

本章は、トヨタ自動車株式会社の電子 PF にのみ適用される。

表 3-1 に適用電子 PF の一覧を示す。

表 3-1 適用電子 PF

電子 PF	DCM	C-ECU	MM	ADAS
19v1	N/A	N/A	N/A	N/A
19v2	*注 1	*注 1	*注 1	*注 1
P19ePF	24DCM	C-ECU	p21MM	ADAS ECU

\*注 1：19v2 の電子プラットフォームに p19 世代の ECU が搭載される可能性が有る。その場合、本仕様書が適用対象となり得る。

一方で、電子プラットフォーム単位での統一性のため 19v2 世代の（本仕様書に準拠しない）ソフトウェアを用いるなどの対応が行われる可能性が有る。

そのため、本仕様書では、19v2 の電子プラットフォームに p19 世代の ECU が搭載された場合についても適用対象とするが、準拠する範囲（一部または全部準拠するか）については規定しない。

### 3.3. 適用サーバー一覧

次期車両向け 24 年頭出しの、TSC、現地事業体管轄の TSP とする。センタ内の終端となる TSC/TSP の後段基盤は適用外とする。

### 3.4. 適用仕向一覧

特に仕向地を明記していない箇所については、全仕向共通とする。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	15
		No.	1.2

4. システム構成

本書が対象とするシステム構成を図 4-1 に示す。

上位仕様書「無線通信セキュリティ要求仕様書」に従い、1 層目の ECU（DCM、MM、CECU）は TSC、TSP※と直接通信を行うが、2 層目の ADAS は TLS Proxy 経由で通信を行う。

（※ECUと TSC 経由で通信する TSP は本書の対象外とし、ECU と直接通信する TSP のみ本書の対象とする。）

また、TLS Proxy 経由の通信において、不特定の ECU からの接続を許可しないよう、Proxy 側 (CECU)は、認証等により予め許可された ECU からの通信のみを中継すること。

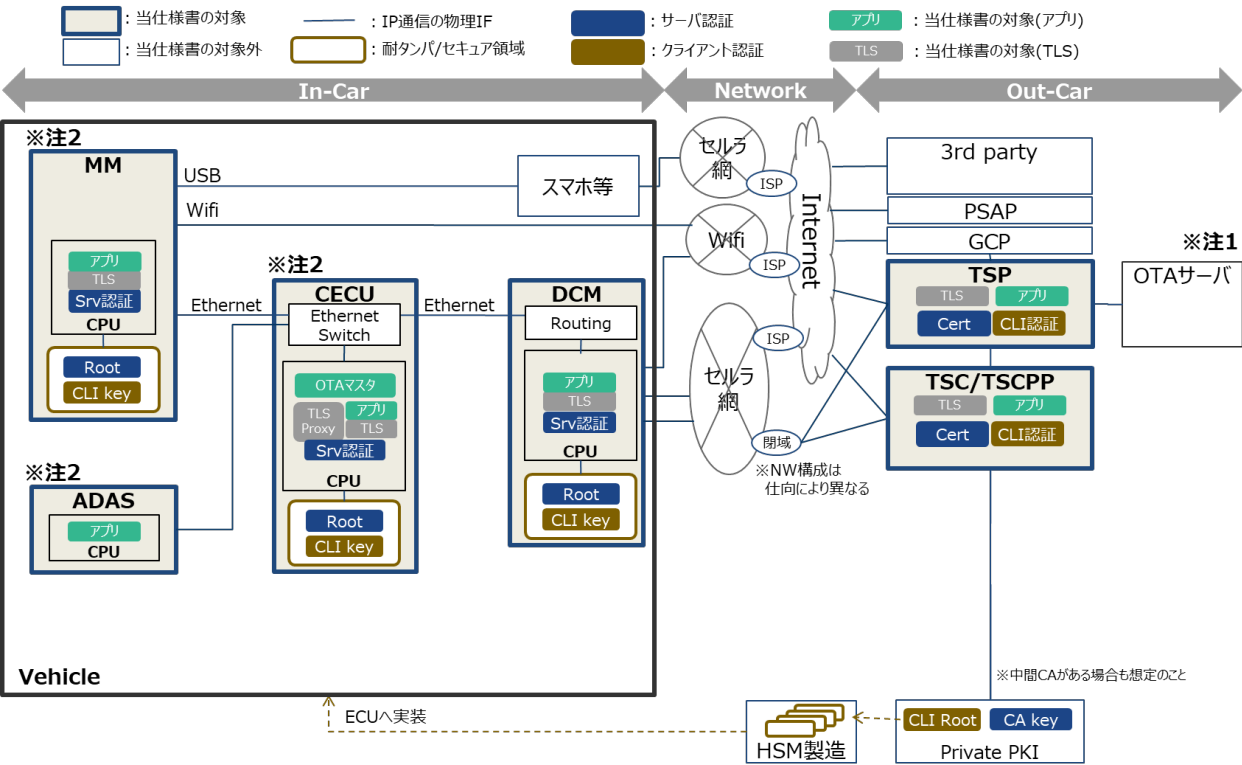


図 4-1 システム構成

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		16
			No.	1.2

## 5. ユースケース

本書が対象とする代表的なユースケースを定義する。

ユースケースは、サービスに基づくシナリオおよび通信特性により表現する。

本書が対象とするユースケースのシナリオ例を以下に示す。

- ・ CAN データのアップロード
- ・ 映像データのアップロード
- ・ ダイアグデータのアップロード
- ・ 車両からセンタへの API アクセス
- ・ センタからのプッシュ（ユニキャスト・マルチキャスト）
- ・ アクセストークン取得（初回認証）
- ・ データダウンロード
- ・ OTA※

※本書の適用範囲は認証のみとし、TLS より上位のアプリケーションプロトコルは対象外とする。

なお、以下のユースケースは本書の対象外とする。

- ・ 音声通話

また、通信特性は、ユースケースにおける通信を、通信データの種別や要求元、周期性等によって分類したものであり、以下のように分類する。

- ・ 通信種別
 

通信の目的を示し、以下に分類する。

  - ・ コマンド
  - ・ データ通信
- ・ 要求元
 

通信契機が ECU から行われるか、センタから行われるかを示す。
- ・ 周期性
 

通信が行われる周期（タイミング）を示し、以下に分類する。

  - ・ 周期（定期的な通信）
  - ・ イベント（不定期に発生する通信）
- ・ 通信頻度
 

通信が発生する頻度を示す。
- ・ 応答性
 

送信側が要求を送信してから最初の応答を受信するまでに求められる時間を示す。
- ・ データサイズ



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car			17
				No.	1.2

イベントまたは 1 周期当たりの通信のデータサイズを示す。

表 5-1 にユースケースを示す。

**表 5-1 ユースケース**

シナリオ	説明	通信特性					
		通信種別	要求元	周期性	通信頻度	応答性	データサイズ
CANデータアップロード	<ul style="list-style-type: none"> <li>・周期的なアップロード(毎分)</li> <li>・サイズは数十 KB 程度</li> <li>・多数の ECU との同時接続</li> </ul>	データ通信	ECU	周期	高 (毎秒~毎分)	中 (数秒)	小~中 (数十 KB~数 MB)
映像データアップロード	<ul style="list-style-type: none"> <li>・非定常(オンデマンド)データ</li> <li>・データサイズは数 MB</li> <li>・連続的なデータ送信</li> </ul>	データ通信	ECU	イベント	中 (数回/時)	低 (十数秒)	大 (数 MB~数十 MB)
ダイアグデータのアップロード	<ul style="list-style-type: none"> <li>・イベント発生時など不定期のアップロード</li> <li>・1 イベントあたりのデータサイズは数十 KB~数十 KB</li> </ul>	データ通信	ECU	イベント	低 (数回/月~数回/日)	中 (数秒)	小~中 (数 KB~数十 KB)
車両からセンタへの API 呼出し	<ul style="list-style-type: none"> <li>・リモートサービスのイベント通知など</li> <li>・高い応答性(数秒)</li> <li>・データサイズは数 KB</li> <li>・到達確認</li> </ul>	コマンド	ECU	イベント	中 (数回/時)	中 (数秒)	小~中 (数 KB~数百 KB)
センタからのプッシュ (ユニキャスト・マルチキャスト)	<ul style="list-style-type: none"> <li>・リモートサービスにおけるリモートコントロールなど、センタから車両に対してのアプリ通信中のプッシュ</li> <li>・データサイズは数 KB</li> </ul>	コマンド	センタ	イベント	中 (数回/時)	中 (数秒)	小(数 KB)
アクセストークン取得 (初回認証)	<ul style="list-style-type: none"> <li>・IG-ON 時の初回アクセスのみ</li> <li>・アクセストークンを発行</li> <li>・データサイズは数 KB</li> <li>・到達確認</li> </ul>	データ通信	ECU	イベント	低 (数回/日)	中 (数秒)	小 (数 KB)
データダウンロード	<ul style="list-style-type: none"> <li>・非定常(オンデマンド)データ</li> <li>・データサイズは数 MB</li> <li>・連続的なデータ受信</li> </ul>	データ通信	ECU	イベント	中 (数回/時)	低 (十数秒)	大 (数 MB~数十 MB)

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		18
			No.	1.2

## 6. 通信要件

### 6.1. 通信プロトコル

#### 6.1.1. 通信プロトコルの選定

5 で示したユースケースにおいて、以下を考慮して適切な通信プロトコルを選定することが必要である。

- ・ 多種多様なデータを大量かつ高頻度にアップロードした場合にコストを抑えた方式とする。
- ・ 定常的なセンタ通信を行う機能はセンタとの常時接続を基本とする。
- ・ アプリケーション毎にセンタとの HTTP セッションを確立すると、センタ側で大量の TCP コネクションを保持する必要がありコストが増大となるため、各車載モジュール単位で 1 つのセッションを使い回すような設計を行う。

上記を鑑み、選定した通信プロトコルとユースケースに対する使い分けの考え方を以下に示す。

- ・ gRPC
  - 多くのユースケースに推奨される。
- ・ HTTP/2
  - 多くのユースケースに推奨される。
- ・ MQTT
  - Publish/Subscribe による通信が有用とされるユースケースに適用する。
- ・ HTTP/1.1
  - 既存の（レガシーの）通信仕様を踏襲する必要がある場合にのみ適用する。（推奨しない）

なお、車載器-TSC 間の API の記述には、以下の理由から OpenAPI、Protocol Buffers、AsyncAPI を用いることを推奨する。

#### 【理由】

API の実装において、クライアントライブラリの生成、サーバスタブの作成、ドキュメント(仕様書)を生成できる等の利点があるため。

また、HTTP/3 は今後の検討とし、本適用電子 PF への適用の対象外とする。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car			19
				No.	1.2

表 6-1 通信プロトコルの比較

分類	項目	通信プロトコル			
		HTTP/1.1	HTTP/2	gRPC	MQTT
通信モデル	コネクション	都度 (再利用不可)	単一 (再利用可)	単一 (再利用可)	単一 (再利用可)
	セッション	ステートレス	ステートフル	ステートフル	ステートフル
	通信方向	クライアント→サーバ	双方向	双方向	双方向
通信品質	到達保証	TCP に依存	TCP に依存	TCP に依存	可 (QoS 設定により可)
	順序保証	不可	TCP に依存	TCP に依存	可 (QoS 設定により可)
	応答性	悪い (コネクション確立による遅延あり)	良い	良い	悪い (非同期通信により遅延大)
	優先度制御	不可	可	可	可
セキュリティ	秘匿	TLS に依存	TLS に依存	TLS に依存	TLS に依存
	認証	Basic 認証 Digest 認証 TLS、OAuth 等他のライブラリに依存	TLS、OAuth 等他のライブラリに依存	TLS、OAuth 等他のライブラリに依存	Basic 認証 TLS に依存

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car			20
				No.	1.2

### 6.1.2. 通信プロトコルとユースケース

各通信プロトコルについて、5 ユースケースに示した代表的なユースケースに対する適性を表 6-2 に示す。

表 6-2 通信プロトコルとユースケースに対する適性

シナリオ	通信特性						プロトコル			
	通信種別	要求元	周期性	通信頻度	応答性	データサイズ	HTTP/1.1	HTTP/2	gRPC	MQTT
CAN データアップロード	データ通信	ECU	周期	高 (毎秒～毎分)	中 (数秒)	小～中 (数十KB～数MB)	×	○	○	○
映像データアップロード	データ通信	ECU	イベント	中 (数回/時)	低 (十数秒)	大 (数MB～数十MB)	○	○※1	△ 大容量データは分割が必要	○
ダイアグデータのアップロード	データ通信	ECU	イベント	低 (数回/月～数回/日)	中 (数秒)	小～中 (数KB～数十KB)	○	○	○	○
車両からセンタへの API 呼出し	コマンド	ECU	イベント	中 (数回/時)	中 (数秒)	小 (数KB)	○※1	○※1	○	×
センタからのプッシュ (ユニキャスト・マルチキャスト)	コマンド	センタ	イベント	中 (数回/時)	中 (数秒)	小 (数KB)	×	○	○	○
アクセストークン取得 (初回認証)	データ通信	ECU	イベント	低 (数回/日)	中 (数秒)	小 (数KB)	○※1	○※1	△ 一回のみの通信には冗長	×
データダウンロード	データ通信	ECU	イベント	中 (数回/時)	低 (十数秒)	大 (数MB～数十MB)	○※1	○※1	△ 大容量データは分割が必要	○

○：適する、△：条件付きで適用可能、×：不適

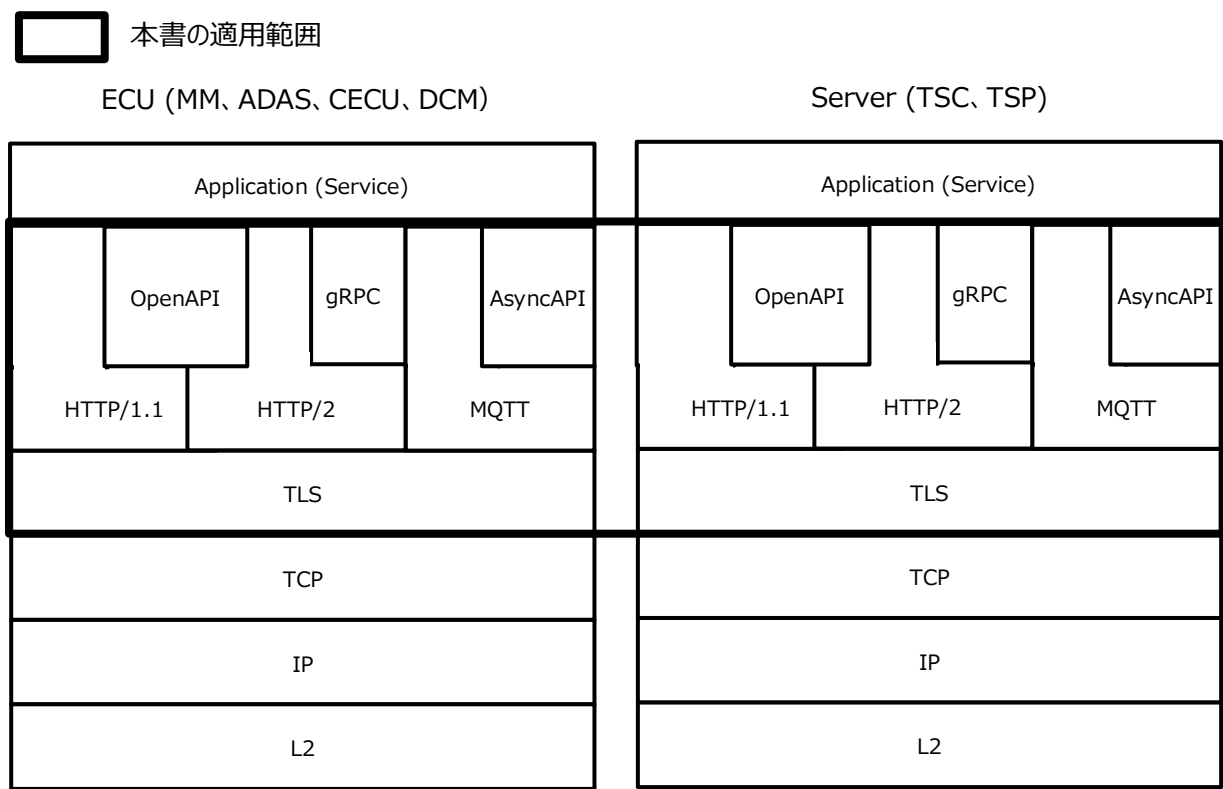
※ セッション維持が不要な場合は HTTP/1.1 が適する。また、HTTP/2 の stream を利用した複数の HTTP/1.1 通信により、効率的な通信が可能である。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	21
		No.	1.2

6.1.3.プロトコルスタック

プロトコルスタックと本書の適用範囲を 図 6-1 に示す。

なお、車載器-TSC 間の API の記述には、OpenAPI、gRPC、AsyncAPI を用いることを推奨する。  
（クライアントライブラリの生成、サーバスタブの作成、ドキュメント(仕様書)を生成できる等の利点があるため）



※各 ECU は DNS によってアドレス解決できることを前提とする。

図 6-1 プロトコルスタック

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	22
		No.	1.2

## 6.2. セキュリティ

### 6.2.1. トランスポートレイヤセキュリティ

車載器とセンタ間では、TLS を使用して通信すること。これにより、暗号化通信とサーバ証明書によるサーバ認証を行う。

### 6.2.2. クライアント認証

TSC および TSP と通信する場合は、トヨタ発行のクライアント証明書を使用し、相互認証を行うこと。

### 6.2.3. 証明書

TSC および TSP と通信する場合、トヨタ自動車発行のルート証明書にチェーンする信頼された証明書のみを用いること。

### 6.2.4. アプリケーションレイヤセキュリティ

DNS サーバは DNSSEC をサポートすることを推奨とし、RFC4033 に準拠すること。

- DNS over HTTPS は、RFC8484 に準拠すること。
- DNS over TLS は、RFC7858 に準拠すること。

## 6.3. セッション管理

車載器と TSC および TSP との通信におけるセッションは、なるべく維持すること。

## 6.4. 車両識別

車載器(ECU)からの通信が、どの車両からか TSC が識別できるようにすること。

認証は、車載器(ECU)に対して行われるのに対して、テレマティクスサービスは車両に対して提供される。そのため、認証だけではテレマティクスサービスの可否を判断できず、正規品のなりすましを防ぐため、車載器(ECU)と車両(車両識別子)との妥当性チェックをセンタ側で実施する。

サーバは、車載器が特定のサービスへのアクセスを許可されていることを確認するために、認証に加えて認可を行うものとする。認証トークンをクライアント証明書にバインドするために、RFC8705 に準拠すること。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	23
		No.	1.2

## 7. 通信仕様

### 7.1. トランスポートレイヤセキュリティ

#### 7.1.1. TLS

##### 7.1.1.1. 基本仕様

- ・トランスポートプロトコルは、下記 TLS1.3 の仕様に準拠すること。
  - RFC8446 The Transport Layer Security (TLS) Protocol Version 1.3
- ・アプリケーションプロトコルは、TLS の使用を必須とする。
- ・SNI に対応すること。
- ・HTTP/2 upgrade を ALPN で行う場合は、RFC7301 に準拠すること。

##### 7.1.1.2. 暗号化アルゴリズム

車載器、センタ間双方の暗号化スイートと鍵交換、署名方式を以下に示す。

##### 7.1.1.2.1. 暗号化スイート

表 7-1 暗号化スイート

優先	暗号化スイート	暗号化(Enc)メッセージ認証コード AEAD 利用(Mac)
1	TLS_AES_256_GCM_SHA384	Enc=AESGCM(256) Mac=AEAD
	TLS_CHACHA20_POLY1305_SHA256	Enc=CHACHA20/POLY1305(256) Mac=AEAD

##### 7.1.1.2.2. 鍵交換方式

- ・ ECDHE 384 ビット以上
- ・ DHE 4096 ビット以上

##### 7.1.1.2.3. 署名方式

将来的に暗号化アルゴリズムの脆弱性が発見された際にも対応可能な様に、以下 2 アルゴリズムを保持

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	24
		No.	1.2

し、利用すること。

- ・ ECDSA256
- ・ RSA3072
  - RSASSA PKCS#1 3072
  - RSASSA-PSS 3072

#### **7.1.1.3. クライアント認証**

車載器とセンタ間の相互認証（mTLS）のため、サーバは TLS1.3（RFC8446）に基づきクライアント認証を行うこと。

サーバは、クライアントの証明書チェーンがトヨタ自動車株式会社から発行されたものであることを検証する。



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	25
		No.	1.2

#### 7.1.1.4. セッション管理

TLS のセッションは、車載器、センタ双方の処理負荷低減のため、新規接続時を除いて、セッション再開方式を適用すること。

TLS セッション再開については、前方秘匿性を確保することを考慮し、stateless resumption PSK+(EC)DHE 1-RTT 方式※で行うこと。

※0-RTT 方式は、リプレイ攻撃の対策をアプリケーションで行う必要があるため適用しないこと。

TLS セッションタイムの有効期間は 12 時間に設定すること。

コネクションの再接続等において、SIGPIPE（一度切断されたソケットを用いた送信におけるプロセスの異常終了）が発生しないよう考慮すること。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	26
		No.	1.2

#### 7.1.1.5. 証明書管理

- ・ 車載器はサーバ証明書チェーンに含まれる全ての証明書、サーバはクライアント証明書チェーンに含まれる全ての証明書の失効確認を行うこと。

##### 【車載器・サーバ共通の仕様】

- ・ 証明書の失効確認について、以下に対応すること。
  - CRL(RFC5280)
  - OCSP(RFC5019, RFC6066, RFC6960, RFC7633, RFC8446)
- ・ TLS 接続時に取得する各証明書に記載された CRL 配布ポイント/OCSP レスポンドから CRL /OCSP 応答を取得すること。
- ・ 証明書チェーンに含まれる証明書が失効した場合、通信は行わないこと。
- ・ 有効な CRL/OCSP 応答 により失効確認できない場合、通信を行わないこと。
- ・ 以下の何れかの場合に最新の CRL/OCSP 応答 を取得すること。
  - (1) CRL/OCSP 応答 の有効期間が満了している
  - (2) CRL/OCSP 応答 の有効期間を判定できない
  - (3) 内部に CRL/OCSP 応答 を保持していない
- ・ CRL 配布ポイント/OCSP レスポンドから取得した CRL/OCSP 応答は、コンテンツとしてキャッシュする。キャッシュコントロールについては、以下に対応すること。基本フローについて図 7-1 に示す。
  - コンテンツキャッシュのコントロール (RFC7234)
  - E-Tag と Last-Modified (RFC7232)
- ・ キャッシュ更新時に CRL/OCSP 応答を取得できなかった場合、キャッシュをクリアせず、キャッシュの有効期間が満了しない限り、保持された CRL/OCSP 応答を用いて失効確認を行うこと。



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	28
	No.	1.2	

- サーバ証明書チェーンのいずれかに must-staple 拡張が存在する場合、かつ OCSP が staple されていない時は、サーバへの接続を拒否すること。(RFC7633)
- 図 7-2 に示すように、トヨタ自動車が発行するサーバ証明書の失効確認に CRL/OCSP をサポートすること。
- トヨタ直接発行のサーバ証明書を利用する場合の失効確認には、CRL を利用すること。  
(トヨタがクロスサインした PKI より発行したサーバ証明書はこの限りではない)

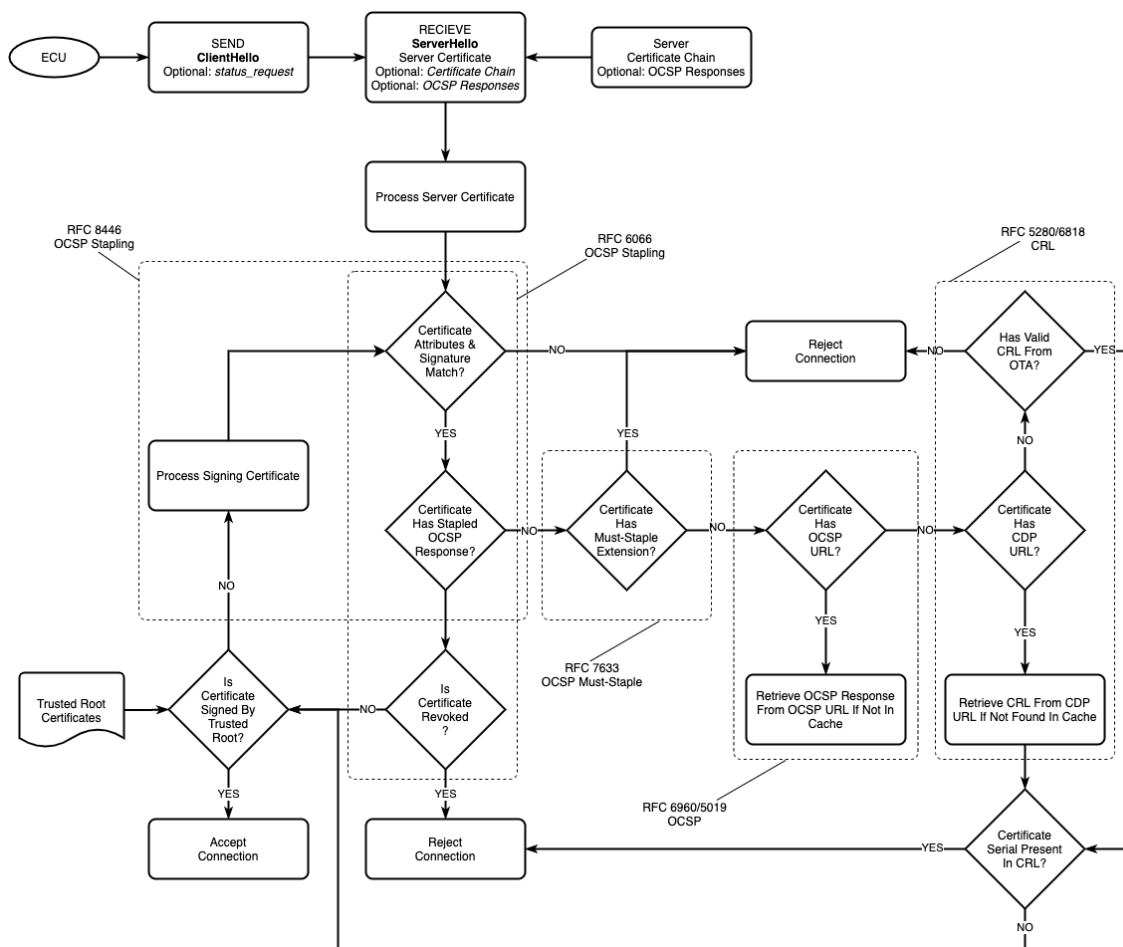


図 7-2 CRL/OCSP 失効確認 基本フロー

#### 【サーバのみの仕様】

- サーバは、クライアント証明書の失効確認に OCSP を用いる場合、OCSP stapling には非対応とする。
- トヨタ発行のクライアント証明書の失効確認には OCSP を用いること。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	29
		No.	1.2

#### 7.1.1.6. 基本シーケンス

TLS1.3 使用時の基本シーケンスとして、フルハンドシェイク時のシーケンスとメッセージを 図 7-3、表 7-2 に、またセッション再開時のシーケンスとメッセージを 図 7-4、表 7-3 に記す。

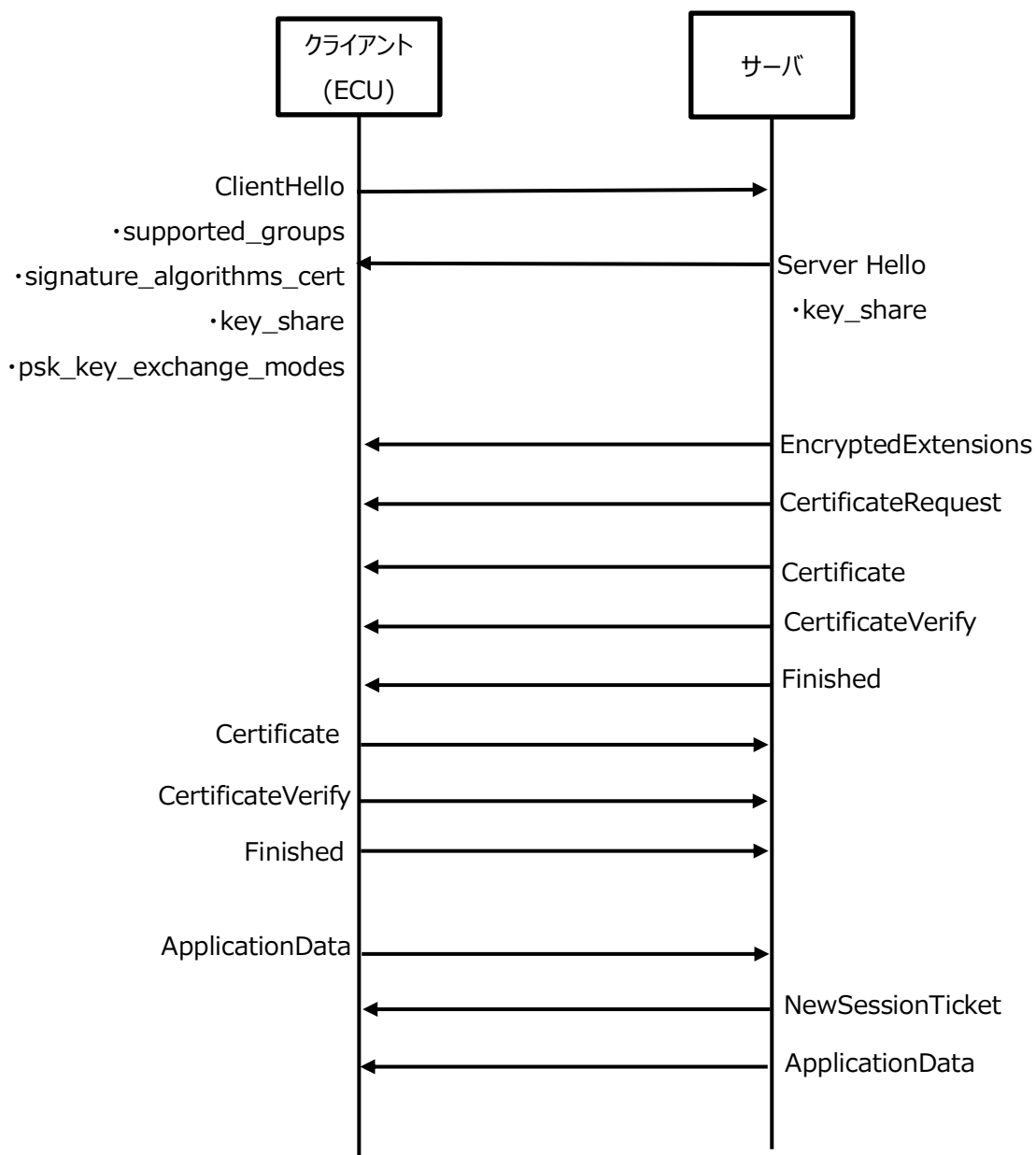


図 7-3 TLS1.3 フルハンドシェイク時のシーケンス

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	30
		No.	1.2

表 7-2 TLS 1.3 フルハンドシェイク時のメッセージ

送信元	メッセージ	内容
クライアント	Client Hello	<ul style="list-style-type: none"> <li>•Version</li> <li>•Random</li> <li>•Session ID</li> <li>•Cipher Suites</li> <li>•Compression Methods</li> <li>•Extensions</li> </ul> supported_versions:TLS1.3 を設定 application_layer_protocol_negotiation:H TTP1.1、または/及び HTTP2 を設定
サーバ	Server Hello	<ul style="list-style-type: none"> <li>•Random</li> <li>•legacy_session_id_echo</li> <li>•Cipher Suite</li> <li>•Compression Method</li> <li>•Extensions</li> </ul> supported_versions:TLS1.3 を設定
	•key_share	鍵交換アルゴリズムに必要な情報 Extension
クライアント	Certificate	サーバに送信するクライアント証明書
	CertificateVerify	送信済の証明書に対する秘密鍵を保持していることを証明するための署名と署名アルゴリズム
	Finished	通信履歴から計算した HMAC 値
	ApplicationData	application_layer_protocol_negotiation で指定したプロトコルによる通信データ
サーバ	NewSessionTicket	セッション再開用の PSK の共有
	ApplicationData	application_layer_protocol_negotiation で指定したプロトコルによる通信データ

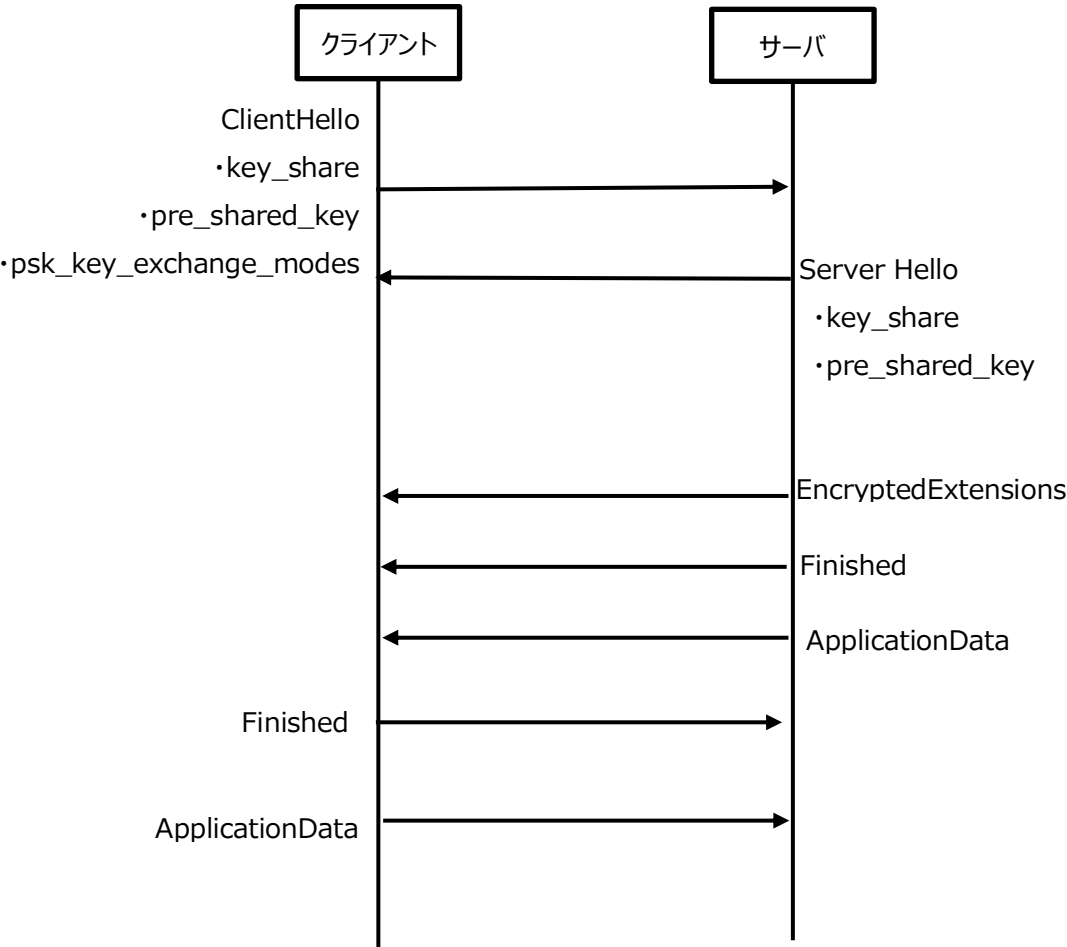


図 7-4 TLS1.3 セッション再開 (1-RTT)シーケンス

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	32
		No.	1.2

**表 7-3 TLS1.3 セッション再開 (1-RTT)のメッセージ**

送信元	メッセージ	内容
クライアント	Client Hello	<ul style="list-style-type: none"> <li>•Version</li> <li>•Random</li> <li>•Session ID</li> <li>•Cipher Suites</li> <li>•Compression Methods</li> <li>•Extensions</li> </ul> supported_versions:TLS1.3を設定 application_layer_protocol_negotiation:HTT P1.1、または/及び HTTP2 を設定
サーバ	Server Hello	<ul style="list-style-type: none"> <li>•Random</li> <li>•legacy_session_id_echo</li> <li>•Cipher Suite</li> <li>•Compression Method</li> <li>•Extensions</li> </ul> supported_versions:TLS1.3を設定
クライアント	ApplicationData	application_layer_protocol_negotiation で指定したプロトコルによる通信データ



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	33
		No.	1.2

7.1.1.7. 異常系シナリオ

TLS の認証フェーズにおける異常系のシナリオと、対応内容について以下に示す。

車載器(ECU)側におけるサーバ認証の異常系のシナリオについて、CRL によるサーバ証明書の失効確認を前提としたシーケンスおよび対応内容を 図 7-5、表 7-4 に示す。

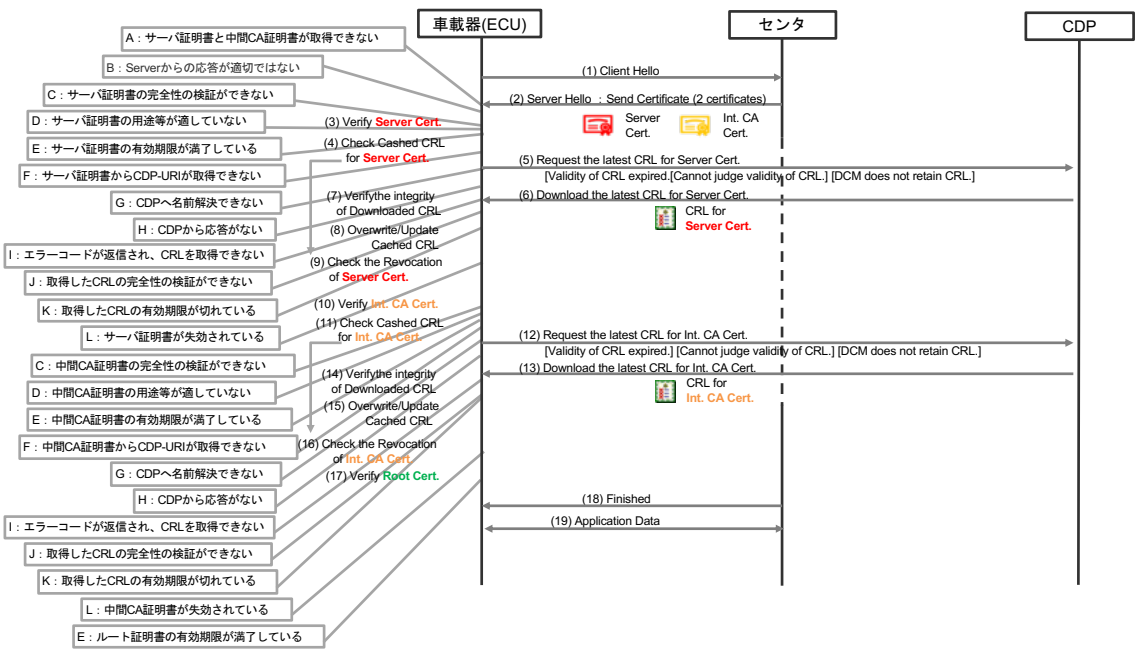


図 7-5 サーバ認証における異常系シナリオのシーケンス

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		34
			No.	1.2

表 7-4 サーバ認証における異常系シナリオの対応内容

異常系シナリオ	期待する動作	
A：証明書が取得できない	リトライ	1sec 間隔で証明書の取得を試みる。
	終了(通信不可)条件	リトライ 1 回失敗。
B：サーバからの応答が適切ではない	リトライ	なし。
	終了(通信不可)条件	応答が無い場合にはタイムアウト時間 1sec。内容が不正な場合には失敗と判断。 取得した証明書は直ちに破棄する。
C：サーバ証明書の完全性の検証ができない	リトライ	1sec 間隔で証明書の取得を試みる。 (証明書の破損が疑われるため)
	終了(通信不可)条件	リトライ後、同じエラーが生じた場合には、失敗と判断。 取得した証明書は直ちに破棄する。
D：サーバ証明書の用途等が適していない	リトライ	なし。
	終了(通信不可)条件	即時、認証処理終了(失敗)。 取得した証明書は直ちに破棄する。
E：サーバ証明書の有効期限が満了している	リトライ	なし。
	終了(通信不可)条件	即時、認証処理終了(失敗)。 取得した証明書は直ちに破棄する。
F：サーバ証明書から CDP/OCSP-URI が取得できない	リトライ	なし。
	終了(通信不可)条件	即時、認証処理終了(失敗)。 取得した証明書は直ちに破棄する。
G：CDP/OCSP レスポンドの名前解決できない	リトライ	タイムアウト時間 1sec で DNS の応答待ちを行う。 DNS 問い合わせ先が複数ある場合は、順次接続を実施。
	終了(通信不可)条件	リトライ 1 回失敗で認証処理終了(失敗)。
H：CDP/OCSP レスポンドから応答がない	リトライ	タイムアウト時間 5sec で接続を試みる。
	終了(通信不可)条件	リトライ 1 回失敗で認証処理終了(失敗)
I：エラーコードが返信され、CRL/OCSP を取得できない	リトライ	1sec 間隔で CRL 取得を試みる。
	終了(通信不可)条件	リトライ X 回失敗で認証処理終了(失敗)。
J：取得した CRL/OCSP の電子署名検証ができない	リトライ	1sec 間隔で証明書の取得を試みる。 (CRL の破損が疑われるため)
	終了(通信不可)条件	リトライで同一エラーが生じた場合、認証処理終了(失敗)。 リトライ 1 回失敗で、認証処理終了(失敗)。 取得した CRL は直ちに破棄する。
K：取得した CRL/OCSP の有効期限が切れている	リトライ	なし。
	終了(通信不可)条件	即時、認証処理終了(失敗)。 取得した CRL は直ちに破棄する。
L：証明書が失効されている	リトライ	なし。
	終了(通信不可)条件	正常判断として、センタ通信を不可とする。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	35
		No.	1.2

サーバ側におけるクライアント認証の異常系のシナリオについて、OCSP によるクライアント証明書の失効確認を前提としたシーケンスおよび対応内容を 図 7-6、表 7-5 に示す。

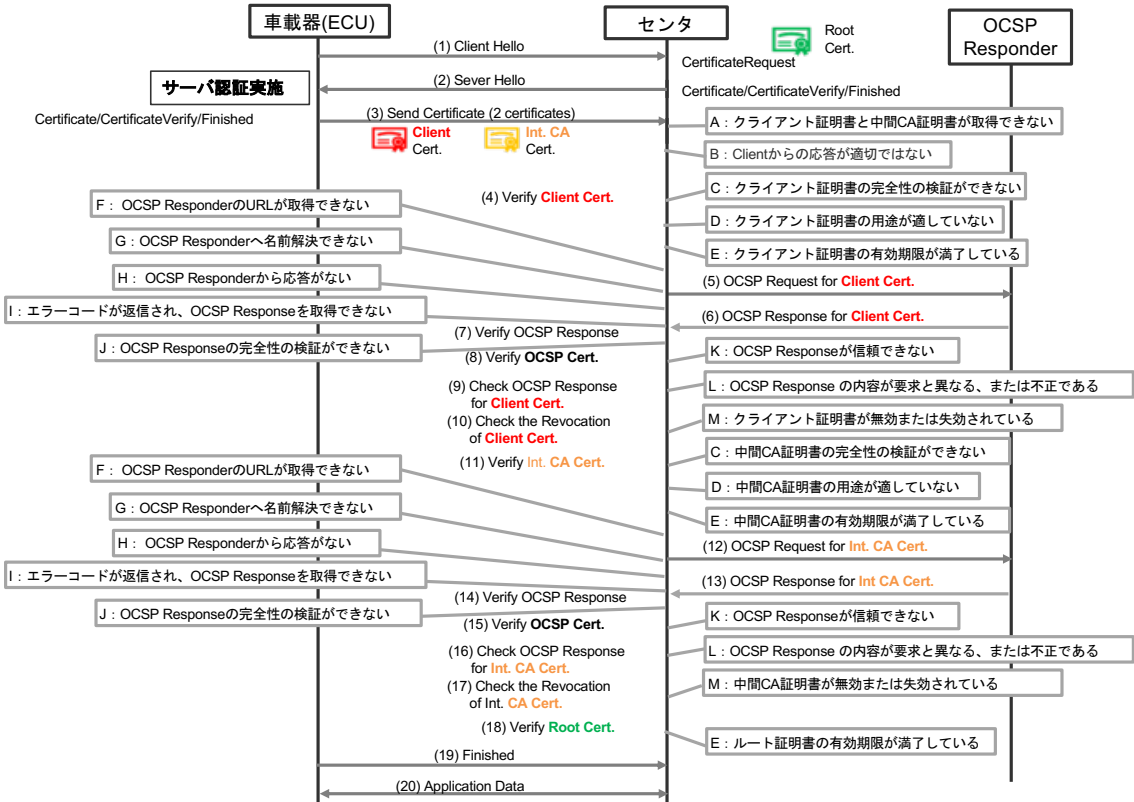


図 7-6 クライアント認証における異常系シナリオのシーケンス

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		36
			No.	1.2

表 7-5 クライアント認証における異常系シナリオの対応内容

異常系シナリオ	期待する動作	
A : クライアント証明書と中間 CA 証明書が取得できない	リトライ	1sec 間隔で証明書の取得を試みる。
	終了(通信不可)条件	リトライ 1 回失敗。
B : クライアントからの応答が適切ではない	リトライ	なし。
	終了(通信不可)条件	応答が無い場合にはタイムアウト時間 1sec。内容が不正な場合には失敗と判断。取得した証明書は直ちに破棄する。
C : クライアント証明書の完全性の検証ができない	リトライ	1sec 間隔で証明書の取得を試みる。(証明書の破損が疑われるため)
	終了(通信不可)条件	リトライ後、同じエラーが生じた場合には、失敗と判断。取得した証明書は直ちに破棄する。
D : クライアント証明書の用途等が適していない	リトライ	なし。
	終了(通信不可)条件	即時、認証処理終了(失敗)。取得した証明書は直ちに破棄する。
E : クライアント証明書の有効期限が満了している	リトライ	なし。
	終了(通信不可)条件	即時、認証処理終了(失敗)。取得した証明書は直ちに破棄する。
F : OCSP Responder の URL が取得できない	リトライ	なし。
	終了(通信不可)条件	即時、認証処理終了(失敗)。取得した証明書は直ちに破棄する。
G : OCSP Responder へ名前解決できない	リトライ	タイムアウト時間 1sec で DNS の応答待ちを行う。DNS 問い合わせ先が複数ある場合は、順次接続を実施。
	終了(通信不可)条件	リトライ 1 回失敗で認証処理終了(失敗)。
H : OCSP Responder から応答がない	リトライ	タイムアウト時間 5sec で接続を試みる。
	終了(通信不可)条件	リトライ 1 回失敗で認証処理終了(失敗)。取得した証明書は直ちに破棄する。
I : エラーコードが返信され、OCSP Response を取得できない	リトライ	1sec 間隔で OCSP Response の取得を試みる。
	終了(通信不可)条件	リトライ 1 回失敗で認証処理終了(失敗)。取得した証明書は直ちに破棄する。
J : OCSP Response の完全性の検証ができない	リトライ	なし
	終了(通信不可)条件	即時、認証処理終了(失敗)。取得した証明書は直ちに破棄する。
K : OCSP Response が信頼できない	リトライ	なし。
	終了(通信不可)条件	即時、認証処理終了(失敗)。取得した証明書は直ちに破棄する。
L : OCSP Response の内容が要求と異なる、または不正である	リトライ	なし。
	終了(通信不可)条件	即時、認証処理終了(失敗)。取得した証明書は直ちに破棄する。
M : クライアント証明書が無効または失効されている	リトライ	なし。
	終了(通信不可)条件	正常判断として、センタ通信を不可とする。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		37
			No.	1.2

## 7.2. アプリケーションプロトコル

### 7.2.1. アプリケーションデータ

アプリケーションデータについて、データの配置と目的・用途を、プロトコル別に 表 7-6、表 7-7 に示す。

**表 7-6 HTTP のデータと目的・用途**

データ		目的・用途
URL	パスパラメータ	センタ側の一意なリソースを表す。
	クエリパラメータ	提供するコンテンツに影響しないアクセス解析、および提供するコンテンツを変えるような動的結果の提供に用いる。 本パラメータは省略可能とする。
メッセージ	ヘッダ	アプリケーションに依存しない情報を表す。 例) ・Content-Type (メディアタイプ) ・User-Agent (製品識別子) ・カスタムヘッダ (車両識別子)
	ボディ	アプリケーション固有のデータのために用いる。

**表 7-7 MQTT のデータと目的・用途**

データ		用途
トピック		センタ側の一意なリソースを表す。 ・API 名 ・車両識別子
メッセージ	ヘッダ	アプリケーションに依存しない情報を表す。 例) ・Content-Type (メディアタイプ)
	ボディ (ペイロード)	アプリケーション固有のデータのために用いる。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	38
		No.	1.2

## 7.2.2. HTTP/1.1

### 7.2.2.1. 基本仕様

HTTP/1.1 のプロトコルをサポート対象とし、以下の仕様／RFC に準拠した通信とすること。

- ・ RFC7230 Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing
- ・ RFC7231 Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content

OpenAPI v3.0 以降の下記仕様に準拠し API を記述すること。

<http://spec.openapis.org/oas/v3.1.0>

### 7.2.2.2. メッセージフォーマット

HTTP/1.1 通信におけるメッセージフォーマット、および本書の適用範囲であるパスパラメータ、クエリパラメータ、リクエスト／レスポンス、ヘッダの共通部（アプリケーション(サービス)が共通に使用する部分）について 図 7-7 に示す。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	39
		No.	1.2

### パスパラメータ・クエリパラメータ

https://xxxx.xxx/<ApplicationId>/<operationId>xxxx

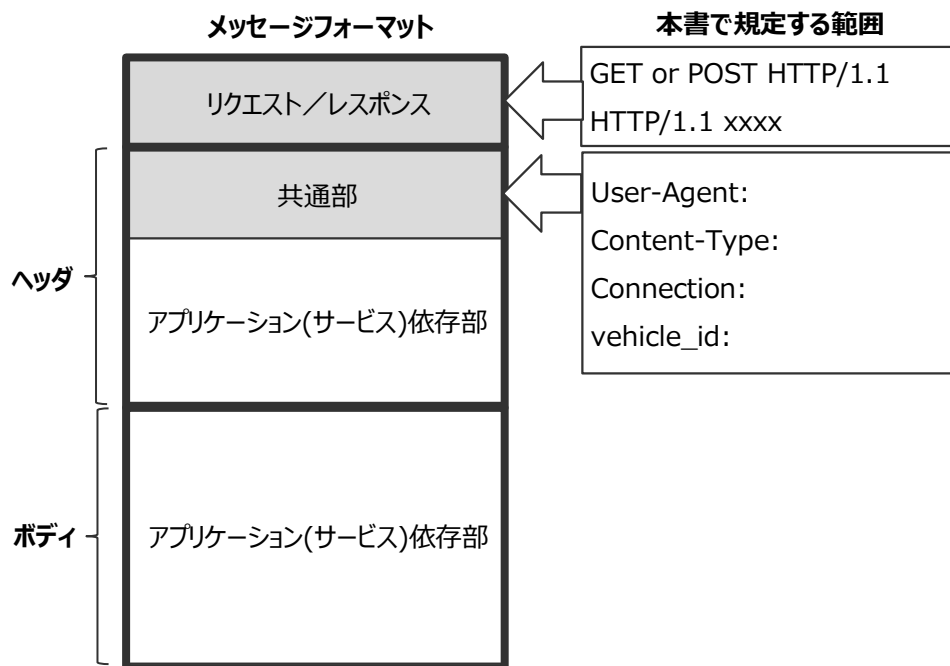


図 7-7 HTTP/1.1 におけるメッセージフォーマット

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		40
			No.	1.2

### 7.2.2.3. ヘッダ定義

以下のメディアタイプをサポートする。メディアタイプは RFC6838 に従い、case-insensitive の文字列として大文字小文字を区別しないこと。HTTP リクエストの header-field の 1 つとして“Content-Type”で指定すること。

- ・ application/octet-stream (バイナリフォーマット)
- ・ application/x-protobuf (Protocol Buffers フォーマット)
- ・ application/json (JSON フォーマット)
- ・ text/plain (テキストフォーマット)

User-Agent は、RFC7231 に従い、製品識別子の付与のために使用し、アプリケーション固有の情報には使用しないこと。

製品識別子は、製品(ECU)の識別情報(情報管理キー)、製品のバージョン、およびサブ製品情報(ソフトウェアの情報やバージョン)などから構成され、情報管理キーを必ず含むこと。

その他の条件を以下に示す。

- ・ User-Agent ヘッダフィールドへの IMEI、GUID、TCON 等の埋め込みには使用しないこと。
- ・ カスタムヘッダとして車両識別子を付与すること。
- ・ 以下のような独自ヘッダフィールドは使用しないこと。
  - X-D-Authenticate、X-Y-Authentication などの Hmac 認証用のフィールド

### 7.2.2.4. レスポンスステータスコード

HTTP レスポンスのステータスコード毎に、リトライの要否を以下に定める。

クライアント側の指定の誤りによるリトライを行わないこと。

サーバ側のステータスに依存するクライアント側エラー(Code:400,404,408,409)については、サーバ側でエラーコード発行時にステータスをロールバック(または破棄)する仕様とし、クライアント側ではリトライを行わないこと。

なお reason-phrase は、status-code をテキストで説明するものであり、クライアントは reason-phrase を結果の判定には使用しないこと。

表 7-8 HTTP/1.1 レスポンスステータスコード

Class	Code	Reason-Phrase	Retry
Informational	100	Continue	No
	101	Switching Protocols	
Successful	200	OK	



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		41
			No.	1.2

	201	Created	
	202	Accepted	
	203	Non-Authoritative Information	
	204	No Content	
	205	Reset Content	
	206	Partial Content	
Redirection	300	Multiple Choices	
	301	Moved Permanently	
	302	Found	
	303	See Other	
	304	Not Modified	
	305	Use Proxy	
	307	Temporary Redirect	
Client Error	401	Unauthorized	
	402	Payment Required	
	403	Forbidden	
	405	Method Not Allowed	
	406	Not Acceptable	
	407	Proxy Authentication Required	
	410	Gone	
	411	Length Required	
	412	Precondition Failed	
	413	Payload Too Large	
	414	URI Too Long	
	415	Unsupported Media Type	
	416	Range Not Satisfiable	
	417	Expectation Failed	
	426	Upgrade Required	
	400	Bad Request	
	404	Not Found	
	408	Request Timeout	
	409	Conflict	
Server Error	500	Internal Server Error	Yes
	501	Not Implemented	
	502	Bad Gateway	

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		42
			No.	1.2

	503	Service Unavailable	
	504	Gateway Timeout	
	505	HTTP Version Not Supported	
	567	次回 IG-ON まで通信しない	
	568	10 分間通信しない	

#### 7.2.2.5. タイムアウト仕様

圏外などにより、リクエスト送出からセンタ応答がない場合に中断するタイムアウト時間を原則 60 秒とする。

ただし、大容量データのアップロードまたはダウンロードを行う場合は対象外とする。

なお、通信タイムアウト時間には回線接続時間を含まないものとする。

センターへの負荷集中を抑止するため、タイムアウト後の再送処理については、再送間隔を延伸する制御をすること。

#### 7.2.2.6. リトライ仕様

リトライが必要な場合のパターンは以下とする。

ただし、NEV 規制などの法規要件は対象外とする。

ユースケース	リアルタイム性	リトライ回数	リトライ間隔
データアップロード (車載器→センタ)	有り	2	3sec
	無し	2	1min
データダウンロード (センタ→車載器)	有り	N/A	N/A

#### 7.2.2.7. セッション管理

##### ・ セッション確立

下記条件の全てが成立している時に、http/1.1 コネクションを確立すること。

- 車載器通信終端が WAN 側 IP アドレスを割り当てられ、データ通信可能な状態であること
- VIN を取得できていること
- ADF アップロードを完了していること

##### ・ セッション維持

HTTP/1.1 利用時において、基本的にセッションの維持は行わない。(Cookie を使用しない)

7.2.2.8. 基本シーケンス

HTTP/1.1 の GET、POST の基本シーケンスを 図 7-8、図 7-9 に示す。

7.2.2.8.1. GET

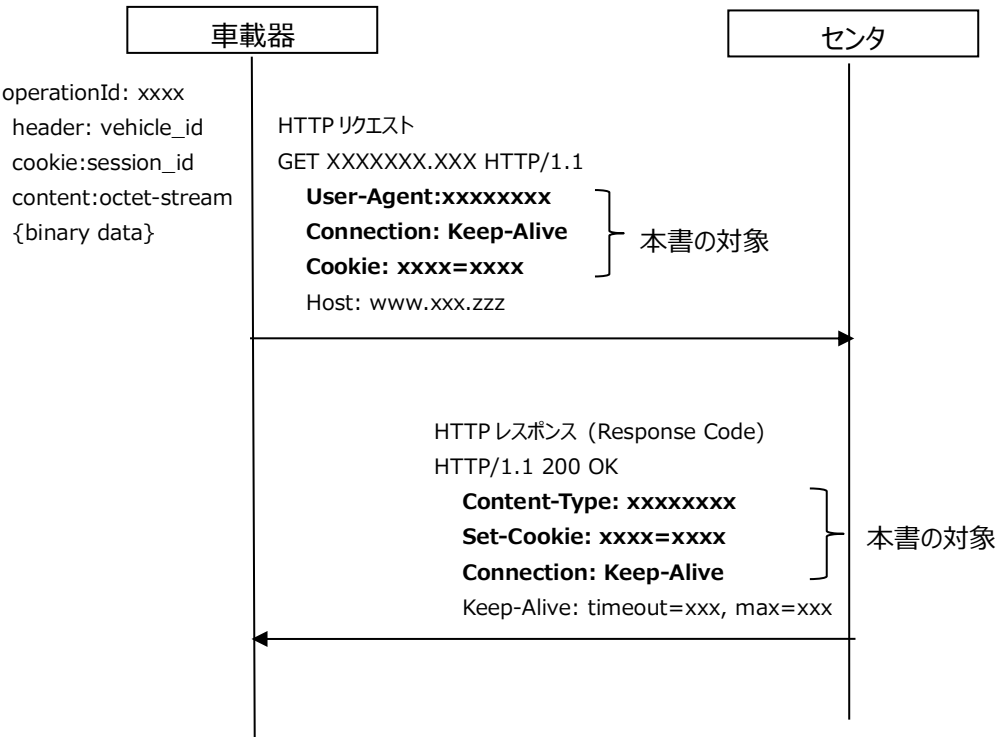


図 7-8 HTTP/1.1 の基本シーケンス (GET)

7.2.2.8.2. POST

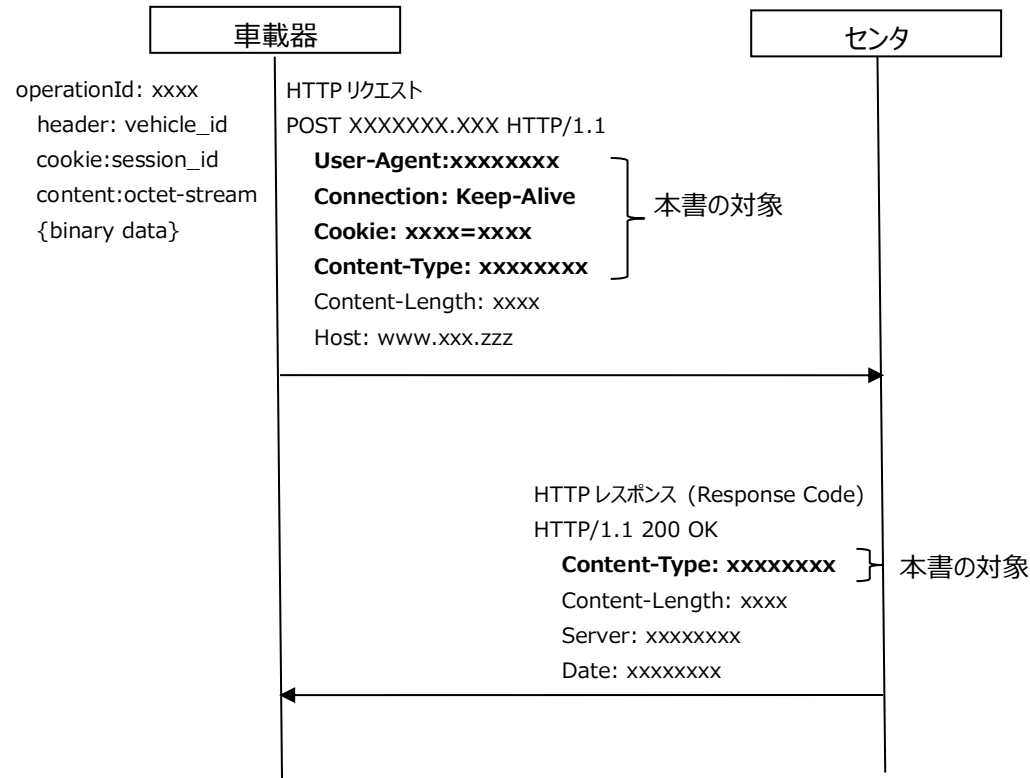


図 7-9 HTTP/1.1 の基本シーケンス (POST)

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	45
		No.	1.2

### 7.2.2.9. API 仕様

API 仕様をユースケース毎に定義する。

なお、別途定義する仕様に基づく yaml ファイルを正しい仕様とし、本書は参考文献として参照すること。

また、本書で言及する yaml ファイルの修正や再実装は行わないこと。

#### 7.2.2.9.1. データアップロード

データアップロードにおける API 仕様を 表 7-9 に示す。

表 7-9 データアップロードの API 仕様 (OpenAPI)

Object/Field			Description
openapi	openapi		OpenAPI バージョン
info	title		API のタイトル
	version		API のバージョン
	description		API の説明
servers	url		サーバの URL
paths	/{ApplicationId}		API のパステンプレート (ApplicationId: アプリケーション名)
	post	operationId	API 名
		parameters	パラメータ
		in: header	name: vehicle_id 車両識別子(7.3 参照)
		in: cookie	name: session_id セッション ID を設定する
		requestBody	データ (Body 部)
		content	メディアタイプ (7.2.2.3 参照)
		schema	スキーマ ※スキーマの内容はアプリケーション(サービス)依存
		responses	
		200	レスポンスコード ※200 以外のエラーコードはアプリケーション(サービス)依存

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car			46
				No.	1.2

Object/Field					Description
				content	メディアタイプ (7.2.2.3 参照)
				schema	スキーマ ※スキーマの内容はアプリケーション(サービス)依存

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		47
			No.	1.2

# <.yaml ファイル : データアップロード>

```

openapi: 3.0.3
info:
  title: api-doc-can-data
  description: API Documentation for CAN data upload
  version: 1.0.2
servers:
  - url: http://example.com/
paths:
  /files/{filename}:
    post:
      operationId: uploadCANDataFile
      parameters:
        - in: header
          name: vehicle_id
          schema:
            type: string
            minLength: 15
            maxLength: 15
        - in: path
          name: filename
          description: Can data file name
          required: true
          schema:
            type: string
        - in: cookie
          name: session_id
          schema:
            type: string
      requestBody:
        content:
          application/octet-stream:
            schema:
              type: string
              format: binary
      responses:
        '200':
          description: Successfully file uploaded 200 response
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Result'
        '400':
          description: Invalid request 400 response
          content: {}
        '500':
          description: Internal error 500 response
          content: {}

components:
  schemas:
    Result:
      type: object
      properties:
        status:
          type: string
          enum:
            - SUCCESS
            - ERROR
      messages:
        type: array
        items:
          type: string
          example: Request success

```

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	48
		No.	1.2

#### 7.2.2.9.2. センタからのプッシュ

HTTP/1.1 に基づく REST は、センタからのプッシュには適さない。(6.1.2 参照)

そのため、センタからのプッシュは REST 適用の対象外とする。

#### 7.2.2.9.3. 車両からセンタへの API 呼出し

車両からセンタへの API 呼出しにおける API 仕様を 表 7-10 に示す。

**表 7-10 車両からセンタへの API 呼出しにおける API 仕様 (OpenAPI)**

Object/Field			Description
openapi	openapi		OpenAPI バージョン
info	title		API のタイトル
	version		API のバージョン
	description		API の説明
servers	url		サーバの URL
paths	/{ApplicationId}/{number}		API のバステンプレート (ApplicationId: アプリケーション名)
	post	operationId	API 名
		parameters	パラメータ
		in: header	name: vehicle_id
		in: cookie	name: session_id
		in: path	name: number
		requestBody	データ (Body 部)
		content	メディアタイプ (7.2.2.3 参照)
		schema	スキーマ ※スキーマの内容はアプリケーション(サービス)依存
		responses	
		200	レスポンスコード ※200 以外のエラーコードはアプリケーション(サービス)依存



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car			49
				No.	1.2

Object/Field					Description
				content	メディアタイプ (7.2.2.3 参照)
				schema	スキーマ ※スキーマの内容はアプリケーション(サービス)依存

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		50
			No.	1.2

<.yaml ファイル : 車両からセンタへの API 呼出し>

```

openapi: 3.0.3
info:
  title: api-doc-call-api
  description: API Documentation for API Call
  version: 1.0.2
servers:
  - url: http://example.com/
paths:
  /api/{number}:
    post:
      operationId: callApi
      parameters:
        - in: header
          name: vehicle_id
          schema:
            type: string
            minLength: 15
            maxLength: 15
        - in: path
          name: number
          description: Number of API
          required: true
          schema:
            type: string
        - in: cookie
          name: session_id
          schema:
            type: string
      requestBody:
        content:
          application/octet-stream:
            schema:
              type: string
              format: binary
      responses:
        '200':
          description: Successfully file uploaded 200 response
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Result'
        '400':
          description: Invalid request 400 response
          content: {}
        '500':
          description: Internal error 500 response
          content: {}

components:
  schemas:
    Result:
      type: object
      properties:
        status:
          type: string
          enum:
            - SUCCESS
            - ERROR
      messages:
        type: array
        items:
          type: string
          example: Request success

```

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	51
		No.	1.2

#### 7.2.2.9.4. データダウンロード

データダウンロードにおける API 仕様を 表 7-11 に示す。

**表 7-11 データダウンロードにおける API 仕様（OpenAPI）**

Object	Field/Type		Description
openapi	openapi		OpenAPI バージョン
info	title		API のタイトル
	version		API のバージョン
	description		API の説明
servers	url		サーバの URL
paths	/{ApplicationId}		API のパステンプレート (ApplicationId: アプリケーション名)
	get	operationId	API 名
		parameters	パラメータ
		in: header	name: vehicle_id 車両識別子(7.3 参照)
		in: cookie	name: session_id セッション ID を設定する
		responses	
		200	レスポンスコード ※200 以外のエラーコードはアプリケーション (サービス)依存
		content	メディアタイプ (7.2.2.3 参照)
		schema	スキーマ ※スキーマの内容はアプリケーション(サービス) 依存

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		52
			No.	1.2

# <.yaml ファイル : データダウンロード>

```

openapi: 3.0.3
info:
  title: api-doc-download
  description: API Documentation for Data Download
  version: 1.0.2
servers:
  - url: http://example.com/
paths:
  /download/{filename}:
    get:
      operationId: download
      parameters:
        - in: header
          name: vehicle_id
          schema:
            type: string
            minLength: 15
            maxLength: 15
        - in: path
          name: filename
          description: Download file name
          required: true
          schema:
            type: string
        - in: cookie
          name: session_id
          schema:
            type: string
      responses:
        '200':
          description: Successfully file uploaded 200 response
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Result'
        '400':
          description: Invalid request 400 response
          content: {}
        '500':
          description: Internal error 500 response
          content: {}

components:
  schemas:
    Result:
      type: object
      properties:
        status:
          type: string
          enum:
            - SUCCESS
            - ERROR
      messages:
        type: array
        items:
          type: string
          example: Request success

```

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	53
		No.	1.2

### 7.2.3. HTTP/2

#### 7.2.3.1. 基本仕様

HTTP/2 のプロトコルをサポート対象とし、以下の仕様／RFC に準拠した通信とすること。

- ・ RFC7540 Hypertext Transfer Protocol Version 2 (HTTP/2)
- ・ RFC7541 HPACK: Header Compression for HTTP/2

OpenAPI v3.0 以降の下記仕様に準拠し API を記述すること。

<http://spec.openapis.org/oas/v3.1.0>

#### 7.2.3.2. メッセージフォーマット

HTTP/2 におけるメッセージフォーマット、および本書の適用範囲であるリクエスト／レスポンス、ヘッダの共通部（アプリケーション(サービス)が共通に使用する部分）について 図 7-10 に示す。

パスパラメータ、クエリパラメータについては以下の考え方にに基づき使用すること。

- ・ パスパラメータ  
センタ側の一意なリソースを表す。
- ・ クエリパラメータ  
提供するコンテンツに影響しないアクセス解析、および提供するコンテンツを変えるような動的結果の提供に用いる。省略可能とする。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	54
		No.	1.2

### パスパラメータ・クエリパラメータ

https://xxxx.xxx/<AppliationId>/<operationId>xxxx



本書で規定する範囲

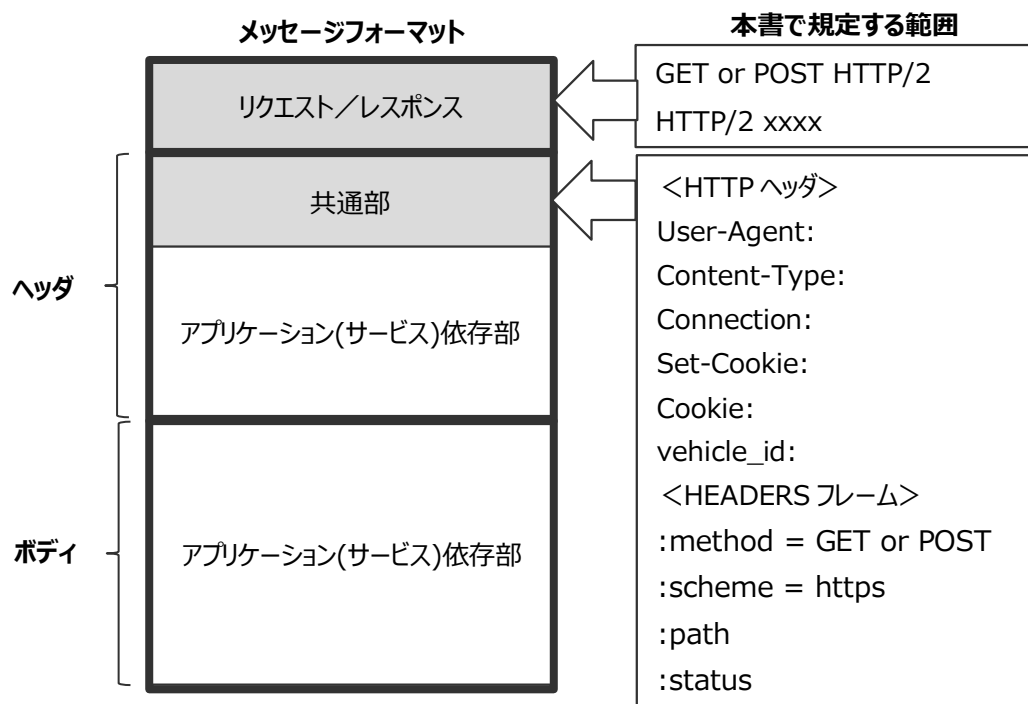


図 7-10 HTTP/2 におけるメッセージフォーマット

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		55
			No.	1.2

### 7.2.3.3. ヘッダ定義

HTTP/2 のリクエストヘッダおよびレスポンスヘッダは、7.2.2.3 に示した仕様（RFC）に準拠すること。

RFC7541 に基づき、ヘッダ圧縮を行うこと。

センタでのセッション管理を前提としたステートフルな接続が必要な場合は、Set-Cookie、Cookie ヘッダを使用すること。

その他のヘッダは、HTTP/1.1 の仕様（7.2.2.3）に準拠すること。

### 7.2.3.4. レスポンスステータスコード

HTTP/1.1 の仕様（7.2.2.4）に準拠すること。

### 7.2.3.5. タイムアウト仕様

HTTP/1.1 の仕様（7.2.2.5）に準拠すること。

### 7.2.3.6. リトライ仕様

HTTP/1.1 の仕様（7.2.2.6）に準拠すること。

### 7.2.3.7. セッション管理

#### ・ セッション確立

下記条件の全てが成立している時に、http/2 コネクションを確立すること。

- 車載器通信終端が WAN 側 IP アドレスを割り当てられ、データ通信可能な状態であること
- VIN を取得できていること
- ADF アップロードを完了していること

#### ・ セッション維持

ALB 利用時において、センタでのセッション管理を前提としたステートフルな接続が必要な場合は、スティッキーセッション（Cookie）を使用すること。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		56
			No.	1.2

7.2.3.8. 基本シーケンス

HTTP/2 の GET、POST の基本シーケンスを 図 7-11、図 7-12 に示す。

7.2.3.8.1. GET

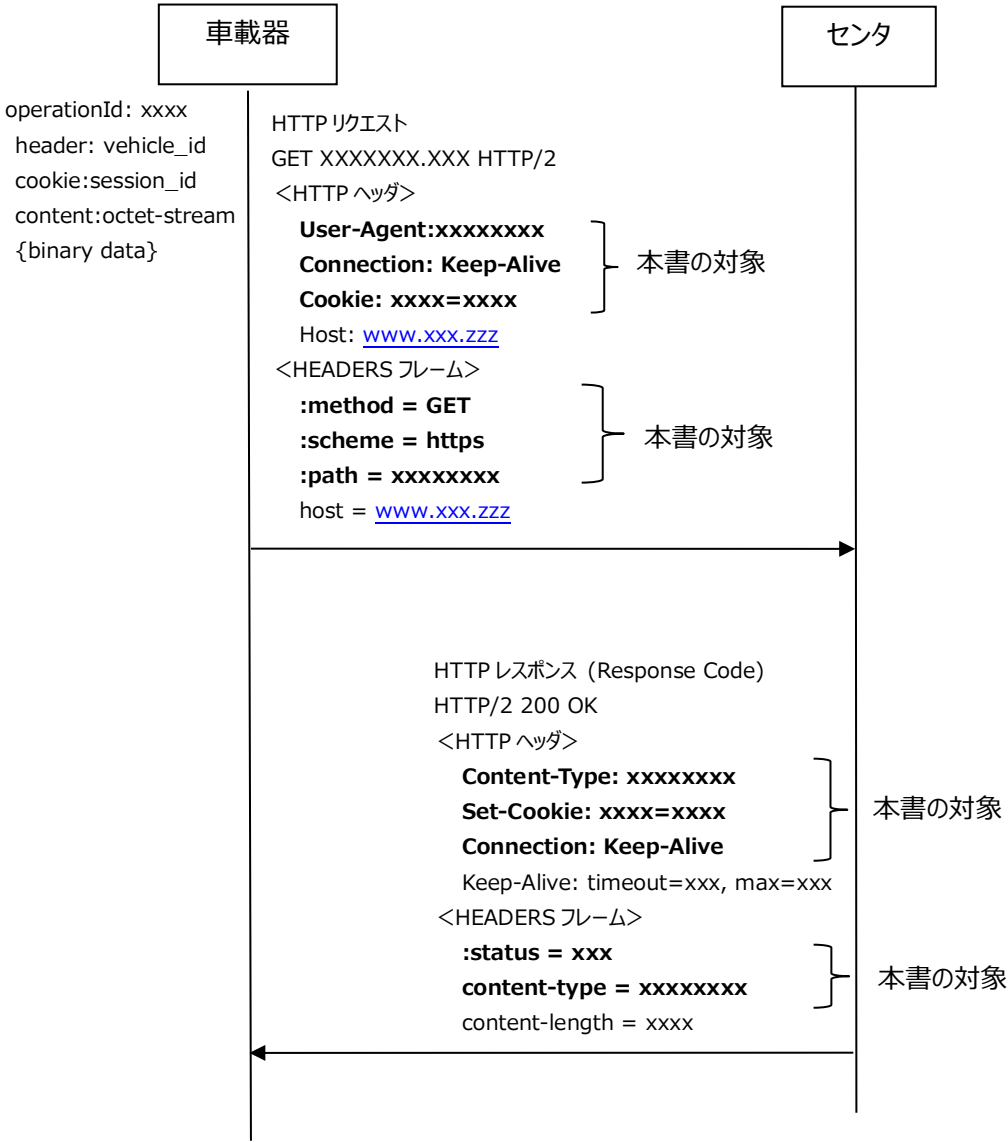


図 7-11 HTTP/2 の基本シーケンス (GET)



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	57
		No.	1.2

7.2.3.8.2. POST

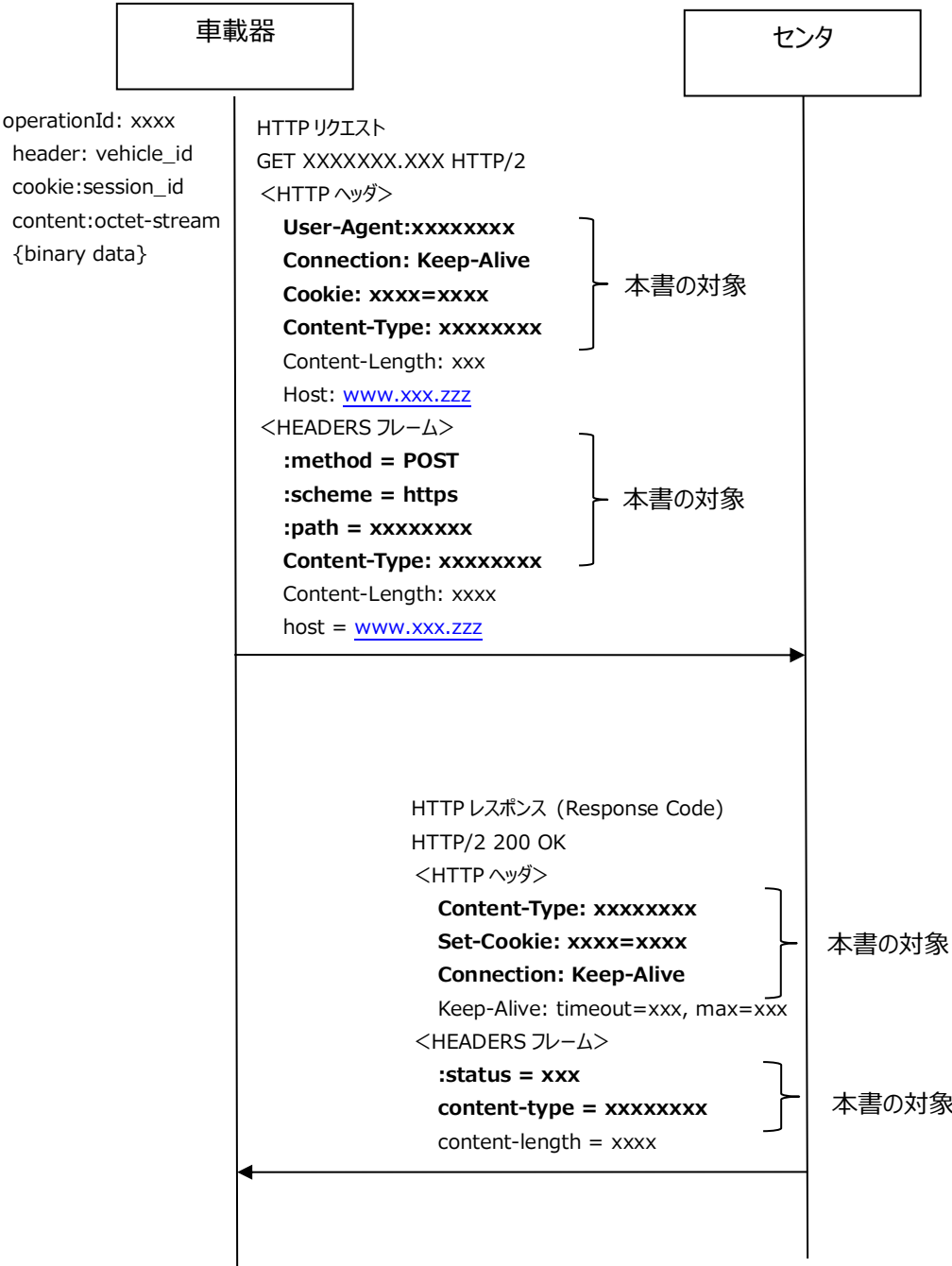


図 7-12 HTTP/2 の基本シーケンス (POST)

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	58
		No.	1.2

### 7.2.3.9. API 仕様

API 仕様をユースケース毎に定義する。

なお、別途定義する仕様に基づく yaml ファイルを正しい仕様とし、本書は参考文献として参照すること。

また、本書で言及する yaml ファイルの修正や再実装は行わないこと。

#### 7.2.3.9.1. データアップロード

HTTP/1.1 の仕様（7.2.2.9.1）に準拠すること。

#### 7.2.3.9.2. センタからのプッシュ

センタからのプッシュにおける API 仕様を 表 7-12 に示す。

表 7-12 センタからのプッシュにおける API 仕様（OpenAPI）

Object/Field			Description
openapi	openapi		OpenAPI バージョン
info	title		API のタイトル
	version		API のバージョン
	description		API の説明
servers	url		サーバの URL
paths	/{ApplicationId}/{app_id}/{sub_id}		API のパステンプレート (ApplicationId: アプリケーション名)
	post	operationId	API 名
		parameters	パラメータ
		in: header	name: vehicle_id 車両識別子(7.3 参照)
		in: cookie	name: session_id セッション ID を設定する
		in: path	name: app_id プッシュ先のアプリケーション ID
		in: path	name: sub_id プッシュ先のアプリケーションサブ ID
		requestBody	データ (Body 部)
		content	メディアタイプ (7.2.2.3 参照)

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		59
			No.	1.2

Object/Field				Description
			schema	スキーマ ※スキーマの内容はアプリケーション(サービス)依存
			responses	
			200	レスポンスコード ※200 以外のエラーコードはアプリケーション(サービス)依存
			content	メディアタイプ (7.2.2.3 参照)
			schema	スキーマ ※スキーマの内容はアプリケーション(サービス)依存

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		60
			No.	1.2

### <.yaml ファイル : センタからのプッシュ>

```

openapi: 3.0.3
info:
  title: api-doc-call-api
  description: API Documentation for API Call
  version: 1.0.2
servers:
- url: http://example.com/
paths:
  /api/{number}:
    post:
      operationId: callApi
      parameters:
        - in: header
          name: vehicle_id
          schema:
            type: string
            minLength: 15
            maxLength: 15
        - in: path
          name: number
          description: Number of API
          required: true
          schema:
            type: string
        - in: cookie
          name: session_id
          schema:
            type: string
      requestBody:
        content:
          application/octet-stream:
            schema:
              type: string
              format: binary
      responses:
        '200':
          description: Successfully file uploaded 200 response
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Result'
        '400':
          description: Invalid request 400 response
          content: {}
        '500':
          description: Internal error 500 response
          content: {}

components:
  schemas:
    Result:
      type: object
      properties:
        status:
          type: string
          enum:
            - SUCCESS
            - ERROR
      messages:
        type: array
        items:
          type: string
          example: Request success

```

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		61
			No.	1.2

#### **7.2.3.9.3.      車両からセンタへの API 呼出し**

HTTP/1.1 の仕様（7.2.2.9.2）に準拠すること。

#### **7.2.3.9.4.      データダウンロード**

HTTP/1.1 の仕様（7.2.2.9.4）に準拠すること。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		62
			No.	1.2

## 7.2.4. gRPC

### 7.2.4.1. 基本仕様

gRPC のバージョンは v1.20 以降をサポート対象とする。

- ・ gRPC over HTTP2 のワイヤープロトコル仕様に従うこと。  
(<https://github.com/grpc/grpc/blob/master/doc/PROTOCOL-HTTP2.md>)
- ・ IDL として Protocol Buffers 3 の使用を前提とする

以下の仕様／RFC に準拠した通信とすること。

- ・ RFC7540 Hypertext Transfer Protocol Version 2 (HTTP/2)
- ・ RFC7541 HPACK: Header Compression for HTTP/2

gRPC で送信するデータサイズについて、映像など、MB を超える構造化されていないバイナリファイルを送る場合は、車載器側で Client Streaming により適切なサイズに分割するか、若しくはサーバ側の設定変更等による対応を行うこと。

なお、前者の場合(車載器側で対応を行う場合)は、分割サイズは 1MB 程度が望ましい。

複数ストリームのデータアップロードに関しては、動画などの容量大のデータ通信を行っている最中に他の通信を行うと帯域が不十分となる可能性があるため、データサイズ等に留意すること。

### 7.2.4.2. メッセージフォーマット

gRPC におけるメッセージフォーマット、および本書の適用範囲であるリクエスト／レスポンス、ヘッダの共通部、およびボディの共通部について 図 7-13 に示す。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	63
		No.	1.2

パスパラメータ・クエリパラメータ  
<https://xxxx.xxx/<ApplicationId>> ← 本書で規定する範囲

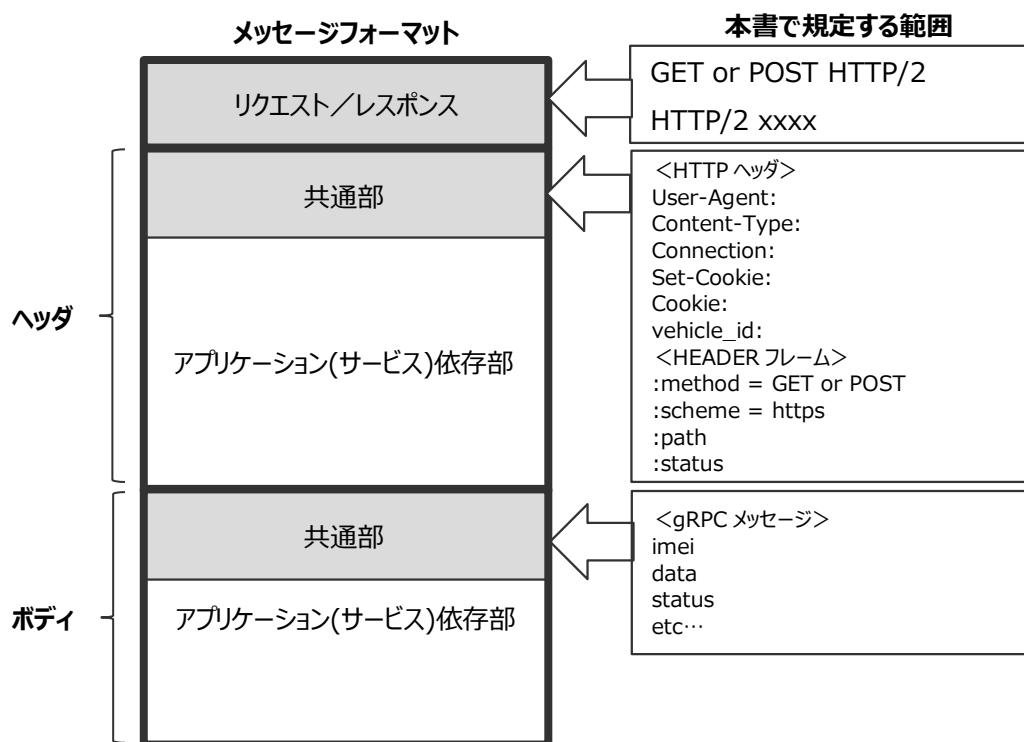


図 7-13 gRPC におけるメッセージフォーマット

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		64
			No.	1.2

#### 7.2.4.3. ヘッダ定義

content-type には、メディアタイプの指定が必要な場合に以下を設定する。

- ・ application/grpc+proto

メディアタイプは RFC6838 に従い、case-insensitive の文字列として大小文字を区別しないこと。

user-agent には、製品識別子および gRPC 推奨のライブラリ名をスペース区切りで設定すること。

例) user-agent: <製品識別子> grpc-go/1.32.0-dev

#### 7.2.4.4. API 仕様

API 仕様をユースケース毎に定義する。

なお、別途定義する仕様に基づく proto ファイルを正しい仕様とし、当ドキュメントは参考文献として参照すること。

また、本書で言及する proto ファイルの修正や再実装は行わないこと。

車両識別子(vehicle\_id)は、proto ファイルではなく、Meta-Data として車載器およびサーバ側のアプリケーションにより設定を行うこと。



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		65
			No.	1.2

#### 7.2.4.4.1. データアップロード

データアップロードにおける API 仕様を 表 7-13、表 7-14 に示す。

**表 7-13 データアップロードの API 仕様 (gRPC Request)**

<Request>

Type	Field Name	Index	Description
required string	vehicle_id	1	車両識別子(7.3 参照)
required bytes	data	2	データ

**表 7-14 データアップロードの API 仕様 (gRPC Response)**

<Response>

Type	Field Name	Index	Description
required message	Status	-	データアップロードに対する、サーバからの応答を示す。
required enum	statuscode	-	OK(0) : 正常 RETRYAFTER(1) : サーバがアプリケーションに一定時間待機させたい場合に使用する。詳細は retry_after_sec の説明に従う。 BUSY(2) : RETRYAFTER 以外でサーバが直ちに応答できない場合に使用する。
required statuscode	status	1	上記 statuscode を示す。
required fixed32	retry_after_sec	2	"status"が"RETRYAFTER"の場合のみ有効となる。(秒) アプリケーションは、本 Response 受信後、直ぐにコネクションを切断し、retry_after_sec が示す秒数待機した後、再びサーバに Request を行う。 サーバが何らかの理由で一時的にサービス提供を停止したい場合に使用する。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		66
			No.	1.2

<.proto ファイル : データアップロード>

```

syntax = "proto3";

option java_multiple_files = true;
option java_package = "grpc.can";
option java_outer_classname = "CanProto";
option objc_class_prefix = "CAN";

service CanService {
  rpc SendCans(stream CanRequest) returns (CanResponse);
}

message CanRequest {
  string vehicle_id = 1;
  bytes data = 2;
}

message CanResponse {
  // message for server status response
  message Status {
    enum statuscodes {
      OK = 0;
      RETRYAFTER = 1;
      BUSY = 2;
    }
    statuscodes status = 1;
    fixed32 retry_after_sec = 2;
  }

  Status status = 1;
}

```

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		67
			No.	1.2

#### 7.2.4.4.2. センタからのプッシュ

センタからのプッシュにおける API 仕様を 表 7-15 に示す。

<Request>

なし。

**表 7-15 センタからのプッシュの API 仕様 (gRPC Response)**

<Response>

Type	Field Name	Index	Description
required message	PushServerStatus	-	サーバの状態を示す。
required enum	statuscode	-	OK(0) : 正常 RETRYAFTER(1) : サーバがアプリケーションに一定時間待機させたい場合に使用する。詳細は retry_after_sec の説明に従う。 BUSY(2) : RETRYAFTER 以外でサーバが直ちに応答できない場合に使用する。
required statuscode	srv_response_code	1	上記 statuscode を示す。
required fixed32	retry_after_sec	2	"status"が"RETRYAFTER"の場合のみ有効となる。(秒) アプリケーションは、本 Response 受信後、直ぐにコネクションを切断し、retry_after_sec が示す秒数待機した後、再びサーバに Request を行う。 サーバが何らかの理由で一時的にサービス提供を停止したい場合に使用する。
required message	Notification	-	プッシュ通知の内容を示すメッセージ。
required fixed32	app_id	1	プッシュ先アプリケーションを示す ID。
required fixed32	sub_id	2	プッシュ先アプリケーションで利用するサブ ID。
required string	expires	3	通知されたプッシュの有効期限。 有効期限の扱いは、アプリケーションに依存する。
required	app_data	4	アプリ毎の任意のデータ。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		68
			No.	1.2

# <Response>

Type	Field Name	Index	Description
bytes			2048byte 未満とすること。

# <proto ファイル>

```

syntax = "proto3";
package service.push;
service PushService {
    rpc GetPushNotifications(Request) returns (stream Response);
}
message Request {}
message Response {
    // message for server status response
    message PushServerStatus {
        enum statuscodes {
            OK = 0;
            RETRYAFTER = 1;
            BUSY = 2;
        }
        statuscodes srv_response_code = 1;
        fixed32 retry_after_sec = 2;
    }
    // message for push notification
    message PushNotification {
        fixed32 app_id = 1;
        fixed32 sub_id = 2;
        string expires = 3;
        bytes app_data = 4;
    }
    PushServerStatus msg_status = 1;
    PushNotification msg_notification = 2;
}

```

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		69
			No.	1.2

#### 7.2.4.4.3. 車両からセンタへの API 呼出し

車両からセンタへの API 呼出しにおける API 仕様を 表 7-16、表 7-17 に示す。

**表 7-16 車両からセンタへの API 呼出しの API 仕様 (gRPC Request)**

<Request>

Type	Field Name	Index	Description
required string	vehicle_id	1	車両識別子(7.3 参照)
required fixed32	api_id	2	API ID
required bytes	data	3	データ (API 呼出し時のデータ)

**表 7-17 車両からセンタへの API 呼出しの API 仕様 (gRPC Response)**

<Response>

Type	Field Name	Index	Description
required message	Status	-	要求に対する、サーバからの応答を示す。
required enum	statuscode	-	OK(0) : 正常 RETRYAFTER(1) : サーバがアプリケーションに一定時間待機させたい場合に使用する。詳細は retry_after_sec の説明に従う。 BUSY(2) : RETRYAFTER 以外でサーバが直ちに応答できない場合に使用する。
required statuscode	status	1	上記 statuscode を示す。
required fixed32	retry_after_sec	2	"status"が"RETRYAFTER"の場合のみ有効となる。(秒) アプリケーションは、本 Response 受信後、直ぐにコネクションを切断し、retry_after_sec が示す秒数待機した後、再びサーバに Request を行う。 サーバが何らかの理由で一時的にサービス提供を停止したい場合に使用する。
required bytes	data	3	API 応答の任意のデータ。 2048byte 未満とすること。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		70
			No.	1.2

<proto ファイル>

```

syntax = "proto3";

option java_multiple_files = true;
option java_package = "grpc.api";
option java_outer_classname = "ApiProto";
option objc_class_prefix = "API";

service ApiService {
  rpc CallApi(ApiRequest) returns (ApiResponse);
}

message ApiRequest {
  string vehicle_id = 1;
  fixed32 api_id = 2;
  bytes data = 3;
}

message ApiResponse {
  // message for server status response
  message Status {
    enum statuscodes {
      OK = 0;
      RETRYAFTER = 1;
      BUSY = 2;
    }
    statuscodes status = 1;
    fixed32 retry_after_sec = 2;
    bytes data = 3;
  }

  Status status = 1;
}

```

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		71
			No.	1.2

#### 7.2.4.4.4. データダウンロード

データダウンロードにおける API 仕様を 表 7-18、表 7-19 に示す。

**表 7-18 データダウンロードの API 仕様 (gRPC Request)**

<Request>

Type	Field Name	Index	Description
required string	vehicle_id	1	車両識別子(7.3 参照)
required bytes	data	2	データ (ダウンロードに必要なデータ)

**表 7-19 データダウンロードの API 仕様 (gRPC Response)**

<Response>

Type	Field Name	Index	Description
required message	Status	-	要求に対する、サーバからの応答を示す。
required enum	statuscode	-	OK(0) : 正常 RETRYAFTER(1) : サーバがアプリケーションに一定時間待機させたい場合に使用する。詳細は retry_after_sec の説明に従う。 BUSY(2) : RETRYAFTER 以外でサーバが直ちに応答できない場合に使用する。
required statuscode	status	1	上記 statuscode を示す。
required fixed32	retry_after_sec	2	"status"が"RETRYAFTER"の場合のみ有効となる。(秒) アプリケーションは、本 Response 受信後、直ぐにコネクションを切断し、retry_after_sec が示す秒数待機した後、再びサーバに Request を行う。 サーバが何らかの理由で一時的にサービス提供を停止したい場合に使用する。
required bytes	data	3	ダウンロードデータ

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		72
			No.	1.2

<proto ファイル>

```

syntax = "proto3";

option java_multiple_files = true;
option java_package = "grpc.dl";
option java_outer_classname = "DIProto";
option objc_class_prefix = "DL";

service DIService {
  rpc Download(DIRequest) returns (stream DIResponse);
}

message DIRequest {
  string vehicle_id = 1;
  bytes data = 2;
}

message DIResponse {
  // message for server status response
  message Status {
    enum statuscodes {
      OK = 0;
      RETRYAFTER = 1;
      BUSY = 2;
    }
    statuscodes status = 1;
    fixed32 retry_after_sec = 2;
    bytes data = 3;
  }

  Status status = 1;
}

```



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		73
			No.	1.2

#### 7.2.4.5. レスポンスステータスコード

gRPC のレスポンスステータスコード、およびリトライの要否を 表 7-20 に示す。

なお、クライアント側の指定が原因のエラーの場合にはリトライを行わない。

**表 7-20 レスポンスステータスコードとリトライ要否**

Code	Mean	HTTP Mapping	Retry
0	OK	200 OK	No
1	CANCELLED	499 Client Closed Request	No
2	UNKNOWN	500 Internal Server Error	Yes
3	INVALID_ARGUMENT	400 Bad Request	No
4	DEADLINE_EXCEEDE	504 Gateway Timeout	Yes
5	NOT_FOUND	404 Not Found	No
6	ALREADY_EXISTS	409 Conflict	No
7	PERMISSION_DENIED	403 Forbidden	No
16	UNAUTHENTICATED	401 Unauthorized	No
8	RESOURCE_EXHAUSTED	429 Too Many Requests	No
9	FAILED_PRECONDITION	400 Bad Request	No
10	ABORTED	409 Conflict	No
11	OUT_OF_RANGE	400 Bad Request	No

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		74
			No.	1.2

Code	Mean	HTTP Mapping	Retry
12	UNIMPLEMENTED	501 Not Implemented	Yes
13	INTERNAL	500 Internal Server Error	Yes
14	UNAVAILABLE	503 Service Unavailable	Yes
15	DATA_LOSS	500 Internal Server Error	Yes

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		75
			No.	1.2

#### 7.2.4.6. セッション管理

- ・ セッション確立

下記条件の全てが成立している時に、gRPC コネクションを確立すること。

- 車載器通信終端が WAN 側 IP アドレスを割り当てられ、データ通信可能な状態であること
- VIN を取得できていること
- ADF アップロードを完了していること

gRPC では同一セッション（チャンネル）内を複数のアプリケーションで利用することが可能であり、あるアプリケーションで発生した障害は、同一チャンネルの他のアプリケーションに影響を及ぼさない。

そのため、効率的な通信を目的とし、1 つのコネクションに複数の通信を行うため、アプリケーションは stream 単位で通信を行うこと。

ただし、アプリケーション間のチャンネルの共有は、通信データ量と通信帯域、遅延リスクを鑑みて行うこと。

- ・ セッション維持 ALB 利用時において、センタでのセッション管理を前提としたステートフルな接続が必要な場合は、スティッキーセッション（Cookie）を使用すること。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	76
		No.	1.2

## 7.2.5. MQTT

### 7.2.5.1. 基本仕様

- MQTT バージョン 5 のプロトコルをサポート対象とする。以下の仕様に準拠した通信とすること。
  - MQTT Version 5.0 OASIS Standard(<https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>)
- 通信量削減のためにトピックエイリアス機能を使用すること。
- MQTT Broker は TLS クライアント認証を用いて、Publisher と Subscriber を認証する。
- 認証に加えて認可を行うものとする。認証トークンをクライアント証明書にバインドするために、RFC8705 に準拠すること。
- MQTT Broker は、認可された Publisher と Subscriber にのみ、トピックの投稿と閲覧を許可する。
- MQTT Publisher と Subscriber は、MQTT で送信される機密性の高いメッセージに対して、エンド・ツー・エンドの機密性保護を実施する。相互の TLS は Broker ないしは Broker 前段の認証 GW で終端し、メッセージは Broker に保存される。

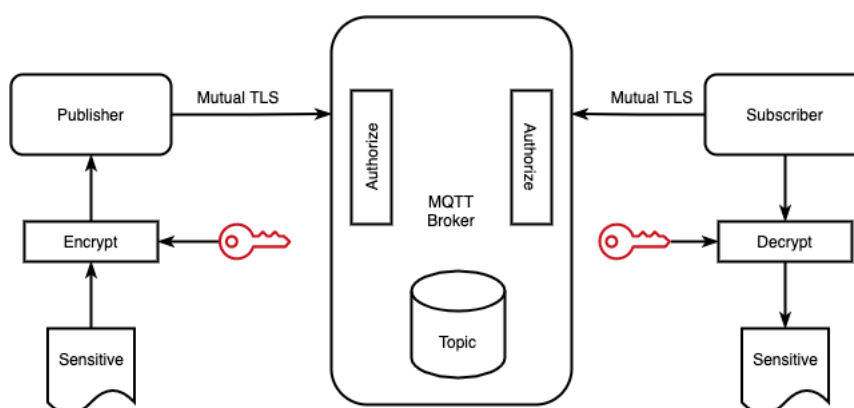


図 7-14 MQTT における認証、認可および機密保持

AsyncAPI v2.0 以降の下記仕様に準拠し API を記述すること。

<https://www.asyncapi.com/docs/specifications/2.0.0>

### 7.2.5.2. メッセージフォーマット

MQTT におけるメッセージフォーマット、および本書の適用範囲である固定ヘッダ、可変ヘッダの共通部（アプリケーション(サービス)が共通に使用する部分）について 図 7-15 に示す。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	77
		No.	1.2

なお、車両識別子、アプリケーション名は、以下の形式でトピック名として付与すること。

- ・ 基本 Topic 名

Topic= <車両識別子>/<アプリケーション名>

- ・ PUBLISH における Topic 名

Topic= <車両識別子>/<アプリケーション名>/<送信方向※>

※送信方向：C2V（センタ→車載器）、V2C（車載器→センタ）

- ・ PUSH 通知を受信したときの Ack 応答に使用する Topic 名

PUSH 通知を受信したアプリケーションは、PUSH 通知の PUBLISH に含まれる Response Topic 宛に、応答の PUBLISH を行うこと。

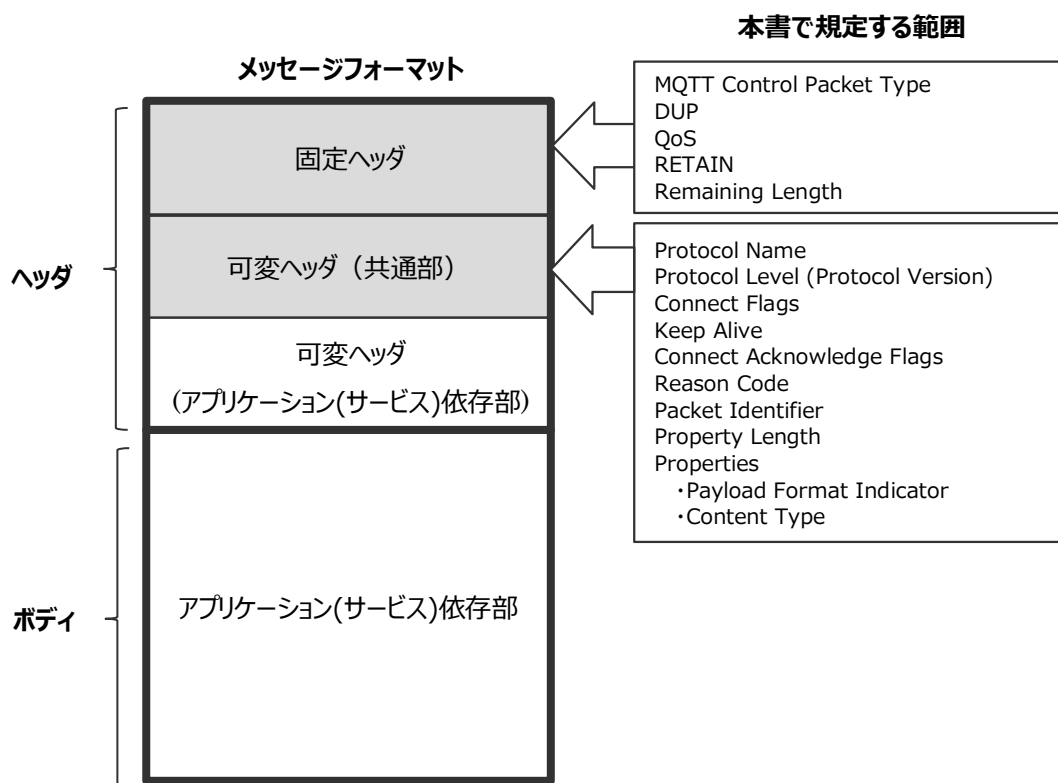


図 7-15 MQTT におけるメッセージフォーマット

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		78
			No.	1.2

### 7.2.5.3. ヘッダ定義

固定ヘッダの QoS には “1 (At least once delivery) ” を設定すること。

CONNECT について、可変ヘッダの keep alive を使用すること。

keep alive の設定値は 7.2.5.7 セッション管理 に示す。

PUBLISH について、以下の可変ヘッダの Property を使用すること。

- ・ Payload Format Indicator
  - 0 (0x00) Byte
  - 1 (0x01) UTF-8 (RFC 3629)
- ・ Content Type
 

RFC6838 に基き、以下の何れかを設定すること。なお、case-insensitive の文字列として大文字小文字を区別しないこと。

  - application/octet-stream (バイナリフォーマット)
  - application/x-protobuf (Protocol Buffers フォーマット)
  - application/json (JSON フォーマット)
  - text/plain (テキストフォーマット)

### 7.2.5.4. レスポンスステータスコード

各コマンドの Reason Code 毎にリトライの要否を 表 7-21、表 7-22、表 7-23、表 7-24 に定める。

クライアント側の指定が原因のエラーの場合にはリトライを行わないこと。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		79
			No.	1.2

- ・ コネクト (CONNACK)

**表 7-21 Reason Code とリトライの要否 (CONNACK)**

Reason Code	Hex	Name	Retry
0	0x00	Success	No
128	0x80	Unspecified error	No
129	0x81	Malformed Packet	No
130	0x82	Protocol Error	Yes
131	0x83	Implementation specific error	Yes
132	0x84	Unsupported Protocol Version	No
133	0x85	Client Identifier not valid	No
134	0x86	Bad User Name or Password	No
135	0x87	Not authorized	No
136	0x88	Server unavailable	Yes
137	0x89	Server busy	Yes
138	0x8A	Banned	Yes
140	0x8C	Bad authentication method	No
144	0x90	Topic Name invalid	No
149	0x95	Packet too large	No
151	0x97	Quota exceeded	Yes
152	0x98	Administrative action	No
153	0x99	Payload format invalid	No
154	0x9A	Retain not supported	No
155	0x9B	QoS not supported	No
156	0x9C	Use another server	No
157	0x9D	Server moved	No
159	0x9F	Connection rate exceeded	No

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		80
			No.	1.2

- ・ パブリッシュ (PUBACK)

**表 7-22 Reason Code とリトライの要否 (PUBACK)**

Reason Code	Hex	Reason Code name	Retry
0	0x00	Success	No
16	0x10	No matching subscribers	No
128	0x80	Unspecified error	No
131	0x83	Implementation specific error	Yes
135	0x87	Not authorized	No
144	0x90	Topic Name invalid	No
145	0x91	Packet identifier in use	No
151	0x97	Quota exceeded	Yes
153	0x99	Payload format invalid	No

- ・ サブスクライブ (SUBACK)

**表 7-23 Reason Code とリトライの要否 (SUBACK)**

Reason Code	Hex	Name	Retry
0	0x00	Granted QoS 0	No
1	0x01	Granted QoS 1	No
2	0x02	Granted QoS 2	No
128	0x80	Unspecified error	No
131	0x83	Implementation specific error	Yes
135	0x87	Not authorized	No
143	0x8F	Topic Filter invalid	No
145	0x91	Packet Identifier in use	No
151	0x97	Quota exceeded	Yes
158	0x9E	Shared Subscription not supported	No
161	0xA1	Subscription Identifiers not supported	No
162	0xA2	Wildcard Subscription not supported	No



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		81
			No.	1.2

- ・ アンサブスクライブ (UNSUBACK)

**表 7-24 Reason Code とリトライの要否 (UNSUBACK)**

Reason Code	Hex	Name	Retry
0	0x00	Success	No
17	0x11	No subscription existed	No
128	0x80	Unspecified error	No
131	0x83	Implementation specific error	Yes
135	0x87	Not authorized	No
143	0x8F	Topic Filter invalid	No
145	0x91	Packet Identifier in use	No

#### 7.2.5.5. タイムアウト仕様

各コマンド送信の ACK 受信の待機時間を以下とする。

- ・ CONNACK の待機時間：30 秒
- ・ PUBACK の待機時間：30 秒
- ・ SUBACK の待機時間：30 秒

ただし、アップロードまたはダウンロードに時間を要する場合は対象外とする。

センターへの負荷集中を抑止するため、タイムアウト後の再送処理については、再送間隔を延伸する制御をすること。

#### 7.2.5.6. リトライ仕様

リトライが必要なユースケースおよびリトライ回数、リトライ間隔を 表 7-25 に示す。

**表 7-25 リトライが必要なユースケースおよびリトライ回数、リトライ間隔**

ユースケース	リアルタイム性	リトライ回数	リトライ間隔
データアップロード (車載器→センタ)	有り	2	3sec
	無し	2	1min
データダウンロード (センタ→車載器)	有り	N/A	N/A

#### 7.2.5.7. セッション管理

- ・ セッション確立

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		82
			No.	1.2

下記条件の全てが成立している間、常に MQTT コネクションを確立すること。

- 車載器通信終端が WAN 側 IP アドレスを割り当てられ、データ通信可能な状態であること
- VIN を取得できていること
- ADF アップロードを完了していること

センタ側での実装 SW 制約により、車載器(ECU)と車両(車両識別子)との妥当性チェックを認証処理にて実現できない場合、MQTT 接続前に Onetime Password 取得機能を使用して、TSC から HTTP GET を使用して Onetime Password (OTP) を取得すること。

MQTT CONNECT の username、password で認証すること。Username には VIN、password には取得した OTP を設定すること。

MQTT CONNECT の Client ID に VIN を設定すること。

- ・ セッション維持

センタからのプッシュの場合は、車載器とセンタ間の keep alive をクライアント側から行うこと。  
(7.2.5.3 参照)

Keep alive の設定値 (間隔) は、消費電力、ネットワーク側からの切断時間などを考慮して設定すること。

- ・ セッション破棄

車載器から自律的に外部通信を切断する場合、MQTT DISCONNECT を完了すること。

7.2.5.8. 基本シーケンス

MQTT の CONNECT、SUBSCRIBE、PUBLISH の一連の流れを示す基本シーケンスを 図 7-16、図 7-17、図 7-18 に示す。

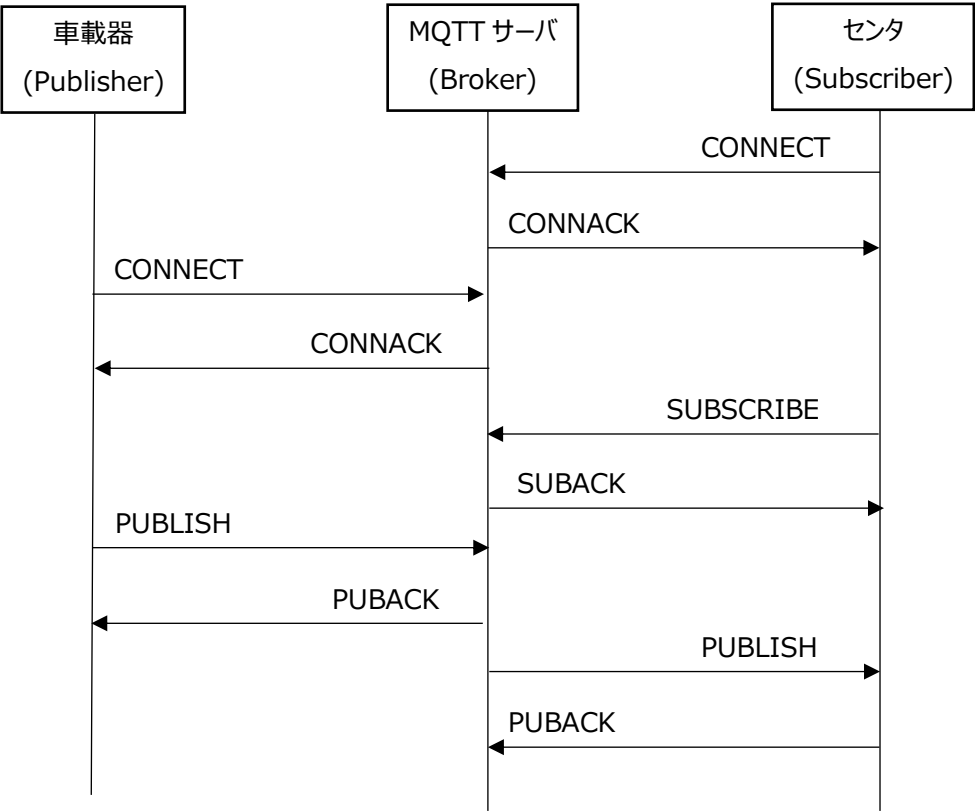


図 7-16 MQTT の基本シーケンス (データアップロード)

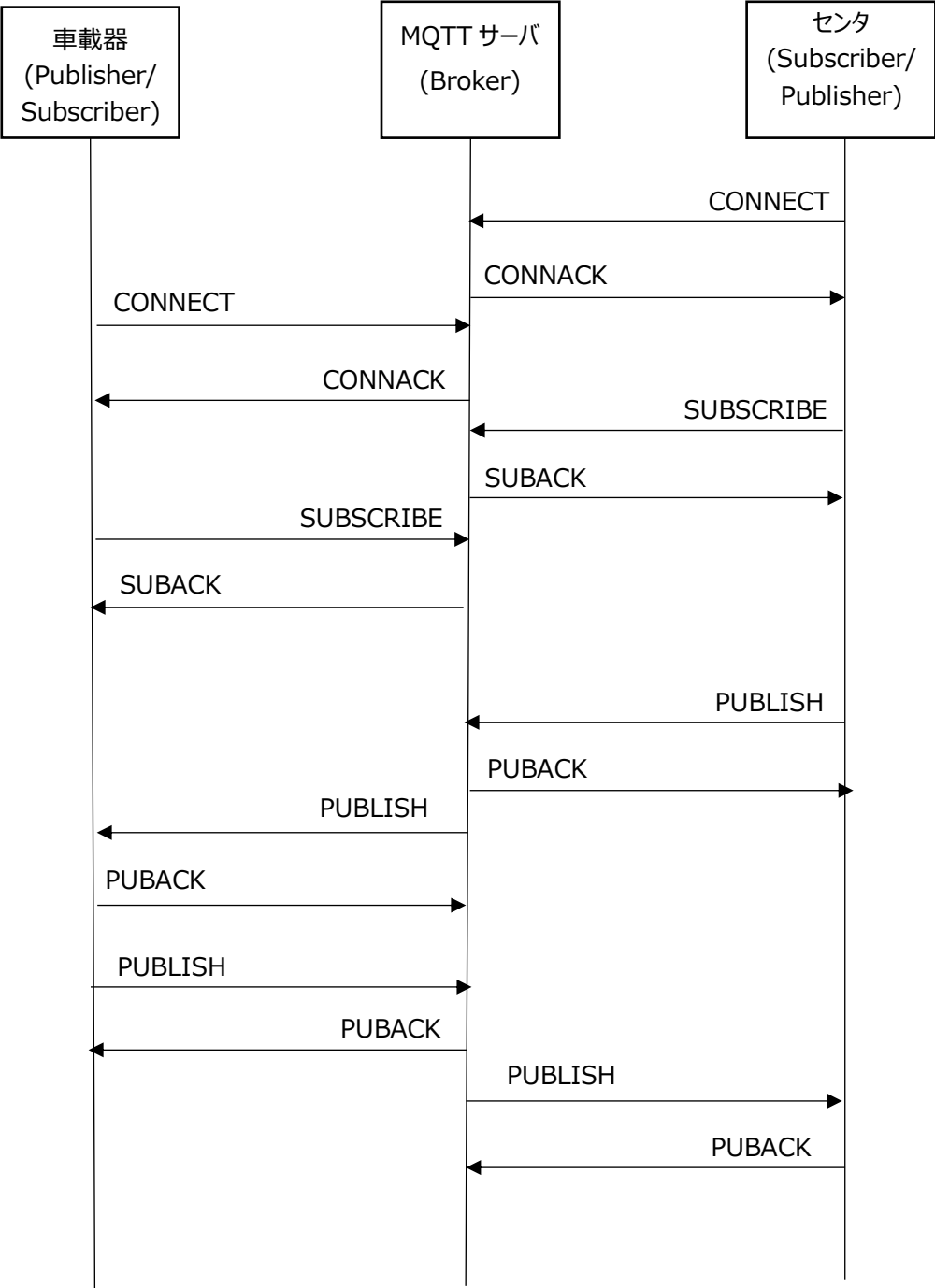


図 7-17 MQTT の基本シーケンス (センタからのプッシュ)

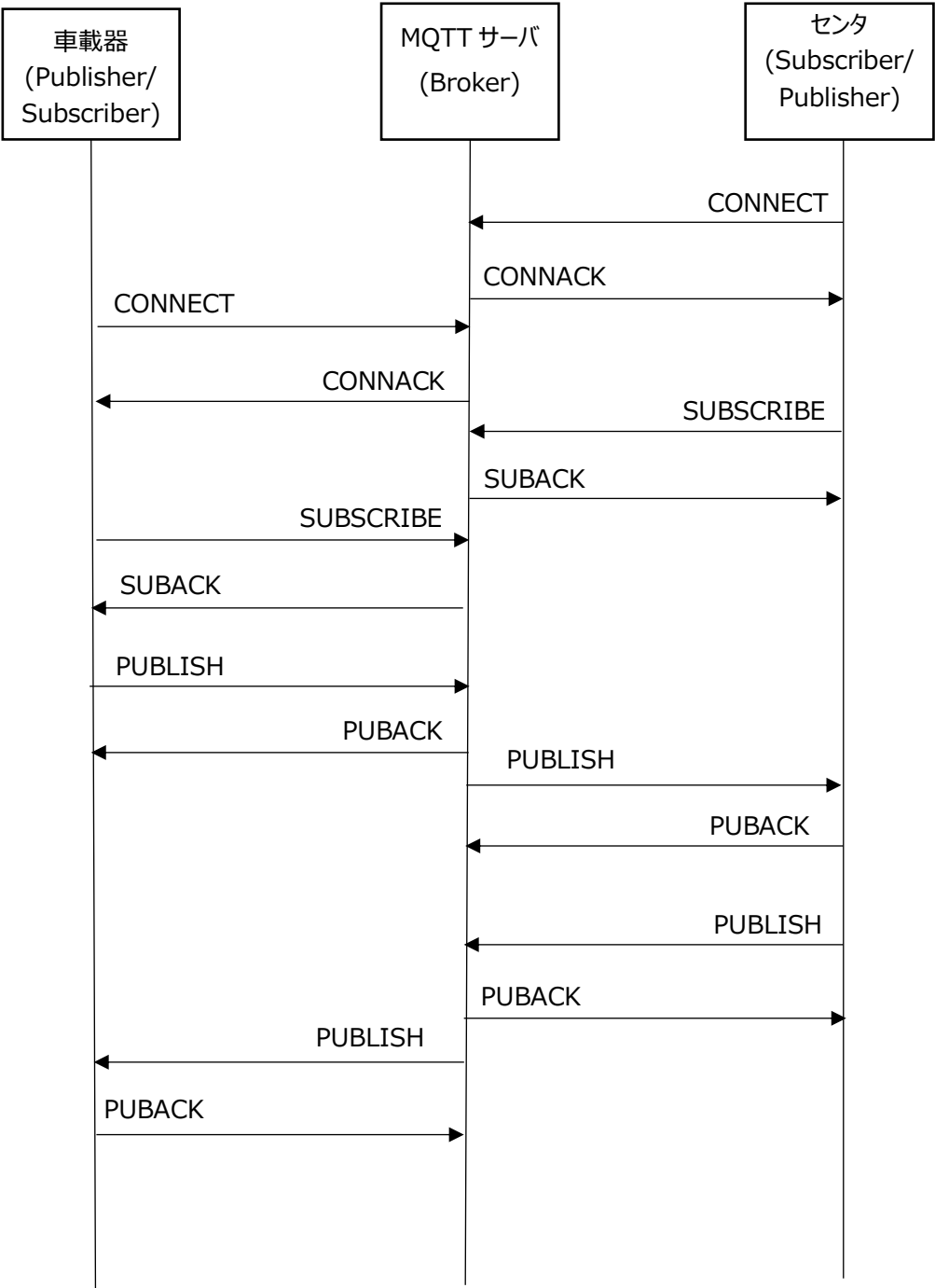


図 7-18 MQTT の基本シーケンス (データダウンロード)

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	86
		No.	1.2

### 7.2.5.9. API 仕様

AsyncAPI は OpenAPI とスキーマ互換である。

そこで、OpenAPI（REST）の API 仕様（7.2.2.9）との互換性を考慮し、以下に各ユースケースの API 仕様を示す。

#### 7.2.5.9.1. データアップロード

データアップロードにおける API 仕様を 表 7-26 に示す。

**表 7-26 データアップロードの API 仕様（AsyncAPI）**

Object/Field			Description	
asyncapi	asyncapi		AsyncAPI バージョン	
info	title		API のタイトル	
	version		API のバージョン	
	description		API の説明	
servers	url		サーバの URL	
	protocol: mqtt		プロトコルとして mqtt を指定	
channels	{vehicle_id}/{ApplicationId}		API のチャネル名 ・vehicle_id: 車両識別子(7.3 参照) ・ApplicationId: アプリケーション名	
	parameters		パラメータ	
	vehicle_id			車両識別子(7.3 参照)
		schema	スキーマ	
		type: string	文字列	
	publish	operationId		API 名
		message		メッセージ ※メッセージの内容はアプリケーション(サービス)依存
		payload	ペイロード	
			type: string	文字列
	subscribe	operationId		API 名

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	87
		No.	1.2

Object/Field			Description
		message	メッセージ ※メッセージの内容はアプリケーション(サービス)依存
		payload	ペイロード
		type: string	文字列

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		88
			No.	1.2

#### <.yaml ファイル : データアップロード>

```

asyncapi: 2.0.0

info:
  title: api-doc-can-data
  version: 1.0.0
  description: API Documentation for CAN data upload

servers:
  production:
    url: example.com
    protocol: mqtt

channels:
  /upload/{vehicle_id}:
    subscribe:
      operationId: uploadCANDataFileReceive
      message:
        payload:
          type: string
    publish:
      operationId: uploadCANDataFileSend
      message:
        payload:
          type: string
    parameters:
      vehicle_id:
        schema:
          type: string

```



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	89
		No.	1.2

#### 7.2.5.9.2. センタからのプッシュ

センタからのプッシュにおける API 仕様を 表 7-27 に示す。

**表 7-27 センタからのプッシュにおける API 仕様 (AsyncAPI)**

Object/Field		Description
asyncapi	asyncapi	AsyncAPI バージョン
info	title	API のタイトル
	version	API のバージョン
	description	API の説明
servers	url	サーバの URL
	protocol: mqtt	プロトコルとして mqtt を指定
channels	{vehicle_id}/{ApplicationId}/{app_id}/{sub_id}	API のチャネル名 ・vehicle_id: 車両識別子(7.3 参照) ・ApplicationId: アプリケーション名 ・app_id: プッシュ先のアプリケーション ID ・sub_id: プッシュ先のアプリケーションサブ ID
	parameters	パラメータ
	vehicle_id	車両識別子(7.3 参照)
	schema	スキーマ
	type: string	文字列
	app_id	プッシュ先のアプリケーション ID
	schema	スキーマ
	type: integer	整数
	format: int32	32 ビット整数
	sub_id	プッシュ先のアプリケーションサブ ID
	schema	スキーマ
	type: integer	整数
	format: int32	32 ビット整数
publish	operationId	API 名

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	90
		No.	1.2

Object/Field			Description
		message	メッセージ ※メッセージの内容はアプリケーション(サービス)依存
		payload	ペイロード
		type: string	文字列
	subscribe	operationId	API 名
		message	メッセージ ※メッセージの内容はアプリケーション(サービス)依存
		payload	ペイロード
		type: string	文字列

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		91
			No.	1.2

# <.yaml ファイル : データアップロード>

```

asyn capi: 2.0.0

info:
  title: api-doc-push
  version: 1.0.0
  description: API Documentation for Server Push

servers:
  production:
    url: example.com
    protocol: mqtt

channels:
  /push/{vehide_id}/{app_id}/{sub_id}:
    subscribe:
      operationId: pushDataReceive
      message:
        payload:
          type: string
    publish:
      operationId: pushDataSend
      message:
        payload:
          type: string
    parameters:
      vehicle_id:
        schema:
          type: string
      app_id:
        schema:
          type: integer
          format: int32
      sub_id:
        schema:
          type: integer
          format: int32

```

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	92
		No.	1.2

### 7.2.5.9.3. 車両からセンタへの API 呼出し

MQTT は、車両からセンタへの API 呼出しには適さない。（6.1.2 参照）

そのため、車両からセンタへの API 呼出しは MQTT 適用の対象外とする。

### 7.2.5.9.4. データダウンロード

データダウンロードにおける API 仕様を 表 7-28 に示す。

**表 7-28 データダウンロードにおける API 仕様（AsyncAPI）**

Object/Field			Description		
asynccapi	asynccapi		AsyncAPI バージョン		
info	title		API のタイトル		
	version		API のバージョン		
	description		API の説明		
servers	url		サーバの URL		
	protocol: mqtt		プロトコルとして mqtt を指定		
channels	{vehicle_id}/{ApplicationId}		API のチャネル名 ・vehicle_id: 車両識別子(7.3 参照) ・ApplicationId: アプリケーション名		
	parameters		パラメータ		
	vehicle_id	schema		スキーマ	
		type: string		文字列	
	publish	operationId		API 名	
		message		メッセージ ※メッセージの内容はアプリケーション(サービス)依存	
		payload			ペイロード
			type: string		文字列
	subscribe	operationId		API 名	

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		93
			No.	1.2

Object/Field			Description
channels		message	メッセージ ※メッセージの内容はアプリケーション(サービス)依存
		payload	ペイロード
		type: string	文字列
	<ApplicationId >/{vehicle_id}/{filename}		API のチャネル名 ・ApplicationId : アプリケーション名 ・vehicle_id: 車両識別子(7.3 参照) ・filename: ファイル名
	parameters		パラメータ
		vehicle_id	車両識別子(7.3 参照)
		schema	スキーマ
		type: string	文字列
		filename	ファイル名
		schema	スキーマ
		type: string	文字列
	publish	operationId	API 名
		message	メッセージ ※メッセージの内容はアプリケーション(サービス)依存
		payload	ペイロード
		type: string	文字列
	subscribe	operationId	API 名
		message	メッセージ ※メッセージの内容はアプリケーション(サービス)依存
		payload	ペイロード
		type: string	文字列

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	94
		No.	1.2

# <.yaml ファイル : データアップロード>

```

asyncapi: 2.0.0

info:
  title: api-doc-download
  version: 1.0.0
  description: API Documentation for Download Data Request and Download

servers:
  production:
    url: example.com
    protocol: mqtt

channels:
  /downloadRequest/{vehicle_id}:
    subscribe:
      operationId: downloadDataRequestReceive
      message:
        payload:
          type: string
    publish:
      operationId: downloadDataRequestSend
      message:
        payload:
          type: string
    parameters:
      vehicle_id:
        schema:
          type: string

  /download/{vehicle_id}/{filename}:
    subscribe:
      operationId: downloadDataReceive
      message:
        payload:
          type: string
    publish:
      operationId: downloadDataSend
      message:
        payload:
          type: string
    parameters:
      vehicle_id:
        schema:
          type: string
      filename:
        schema:
          type: string

```

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	95
		No.	1.2

### 7.3. 車両識別

TSC での車載器(ECU)の識別・管理用途のため、TSC への全ての通信に対して車両を識別できる値 (VIN) を付与すること。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		96
			No.	1.2

## Appendix 1

当 Appendix は、下位仕様を検討するにあたって、有益となる情報を記載する。

### 1. データフォーマット

送受信データのフォーマットは以下をサポートの対象とする。

- ・ バイナリデータ
- ・ テキストデータ
  - JSON
  - プレーンテキスト

HTTP、gRPC 利用時において、データサイズが 2KB※以上（一定サイズ）の場合に gzip で圧縮するものとする。

MQTT は軽量データの取り扱いを前提としていることもあるため、圧縮要否については個別要件にて判断とし、圧縮を行う場合は、各プロトコルで共通とするため、gzip による圧縮を行う。

**※サイズの決定は圧縮負荷と通信サイズのトレードオフで検証により判断要**

gRPC では、メッセージフォーマットとして、Protocol Buffers を使用する。

IDL で Protocol Buffers を使用する場合は、そのシリアル化仕様に従う。

その他定義を以下に示す。

- ・ エンディアン
  - バイナリ形式の場合は各項目に必ずバイトオーダーを記載すること
  - ビッグエンディアンを採用すること
- ・ 有効値
  - 形式を記載すること、数値の場合は自然数か整数かを記載すること
  - 最大桁数、最小値、最大値、値として取りうるパターンを記載すること
- ・ 不定値
  - 不定値には桁数分の 0 を指定すること
  - 0 が有意な意味を持つ項目の場合は桁数分の F を指定すること
- ・ パディング
  - 項目の値が可変で桁数を満たさない場合はパディング形式を記載すること



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		97
			No.	1.2

## 2. ファイル名

YYYYMMDD\_HHMMSS\_<世代>\_<機能>\_<カウンタ値>\_<リクエスト ID>\_<トリガ識別子>.<拡張子>

- ・ 世代は、ファイル名の重複がないようにするため、ECU を示す英数字一文字と、2 桁の 10 進表現とし、どの世代の車載器か分かる様に付与する。（電子 PF の世代に合わせて 2 桁の表現とする）  
例：CXX（CECU の XX 番目）
- ・ 機能は、機能毎に 3 桁の 10 進表現とし、機能毎に付与する。（将来の機能の増加を鑑み 3 桁の表現とする）
- ・ カウンタ値は、車載器で保持するカウンタ値を設定する。  
2 桁の 10 進表現とし、アプリケーション毎に付与する。（日時が秒単位であり、1 秒以内に発生するトリガが多くても数十で収まる前提において、2 桁の表現とする）
- ・ リクエスト ID は、要求されたコマンド内で設定されたリクエスト ID を設定する。  
リクエスト ID がなかった場合、無効値（バイナリの場合"FFF…"、text や json の場合は"000…"）を付与する。
- ・ トリガ識別子は各処理が起動されたトリガを識別するために付与する。  
※同一 URL の場合に何起因かが判別可能な様に識別可能にする。  
アプリケーション固有の識別子は重複しない英数字一文字とすること。  
例）
  - ◇ n: IG ON トリガ
  - ◇ f: IG OFF トリガ
  - ◇ m: センタ SMS 送信契機
  - ◇ s: スケジューラ契機
  - ◇ c: CAN/Ethernet 通信トリガ
- ・ カウンタ値、リクエスト ID、トリガ識別子以外にファイル名に情報付与が必要な機能はトリガ識別子の後に文字列を追加する。
- ・ 拡張子は小文字とし、機能ごとに個別に設定する。
- ・ 日時は UTC（RFC3339）により、以下の通り表示する。

	データ長	設定
YYYY	4 桁	西暦標記とした 4 桁表示とする
MM	2 桁	01～12 の 2 桁表示
DD	2 桁	01～31 の 2 桁表示
HH	2 桁	00～23 の 2 桁表示
MM	2 桁	00～59 の 2 桁表示
SS	2 桁	00～59 の 2 桁表示

※GPS 未受信等で時刻が不明な場合は、時刻を 00000000\_000000 とする。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	98
		No.	1.2

### 3. 文字コード

使用する文字コードは UTF-8（RFC3629）とする。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	99
		No.	1.2

4. 通信シーケンス

4.1. データ通信フロー

4.1.1. データアップロード

4.1.1.1. HTTP/1.1

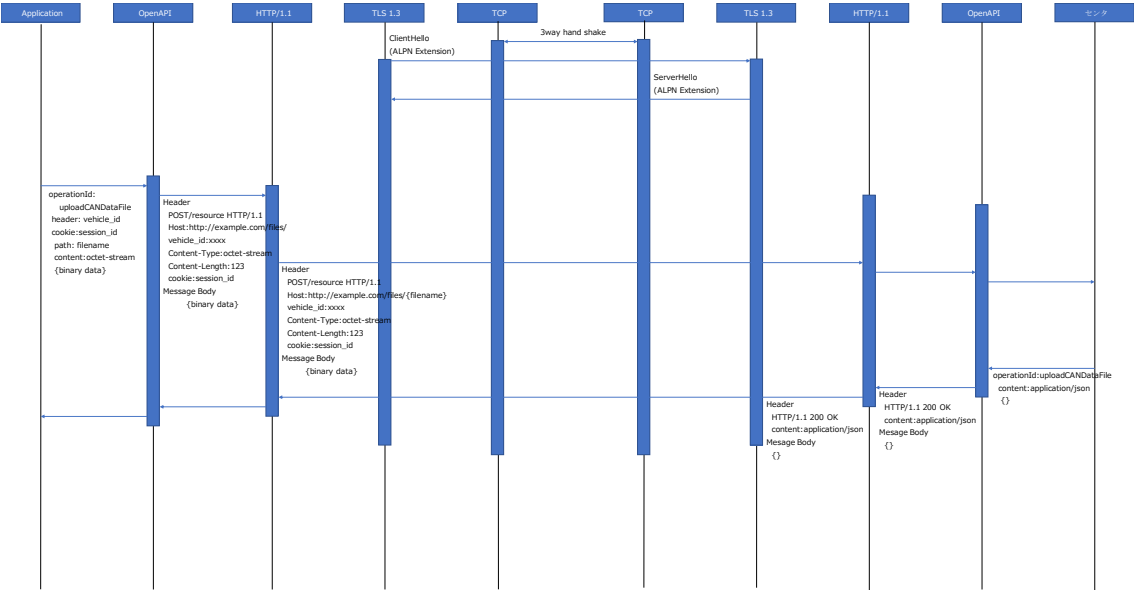


図 A-1 データアップロードシーケンス (HTTP/1.1)

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	100
		No.	1.2

### 4.1.1.2. HTTP/2

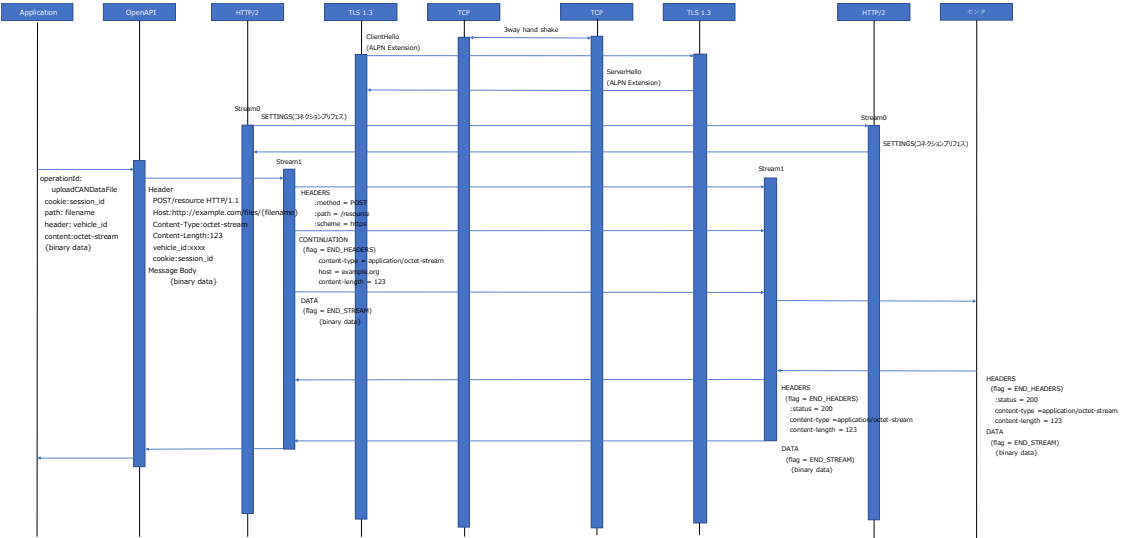


図 A-2 データアップロードシーケンス (HTTP/2)

4.1.1.3. gRPC

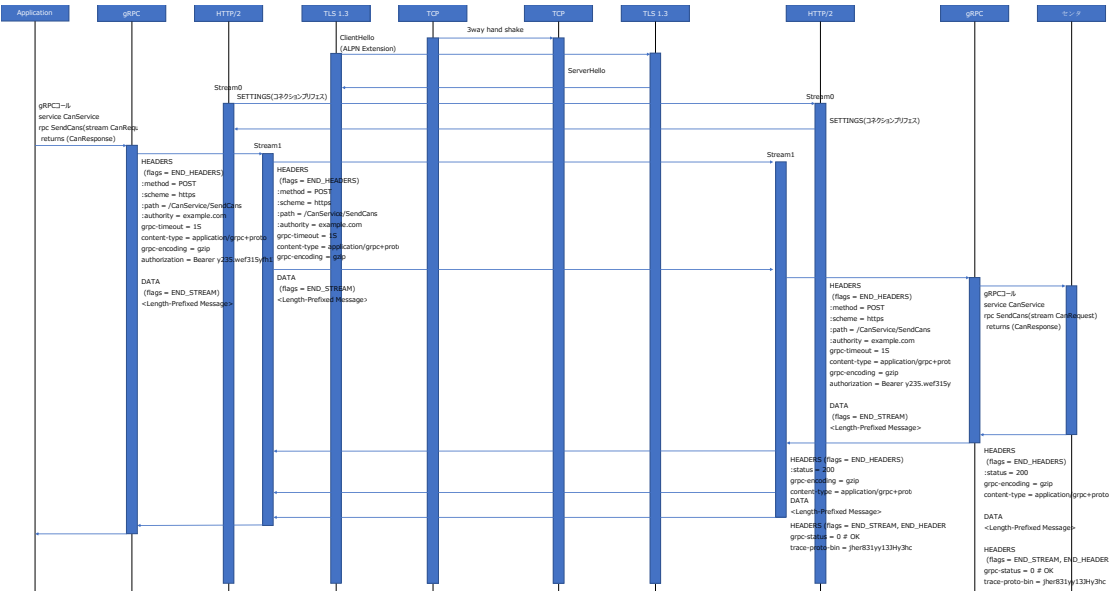


図 A-3 データアップロードシーケンス (gRPC)



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	103
		No.	1.2

4.1.2. センタからのプッシュ

4.1.2.1. HTTP/2

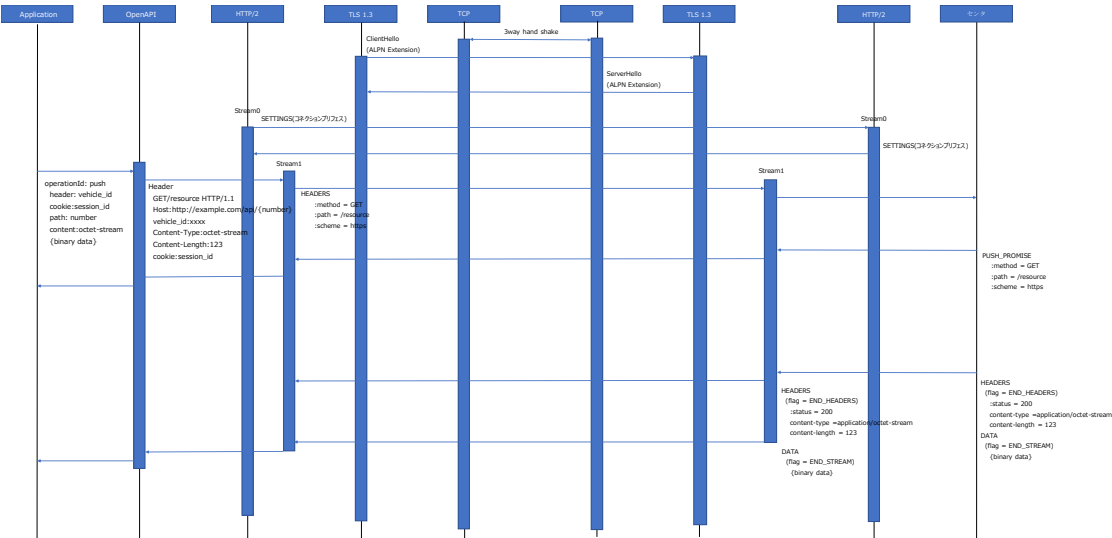


図 A-5 センタからのプッシュシーケンス (HTTP/2)

4.1.2.2. gRPC

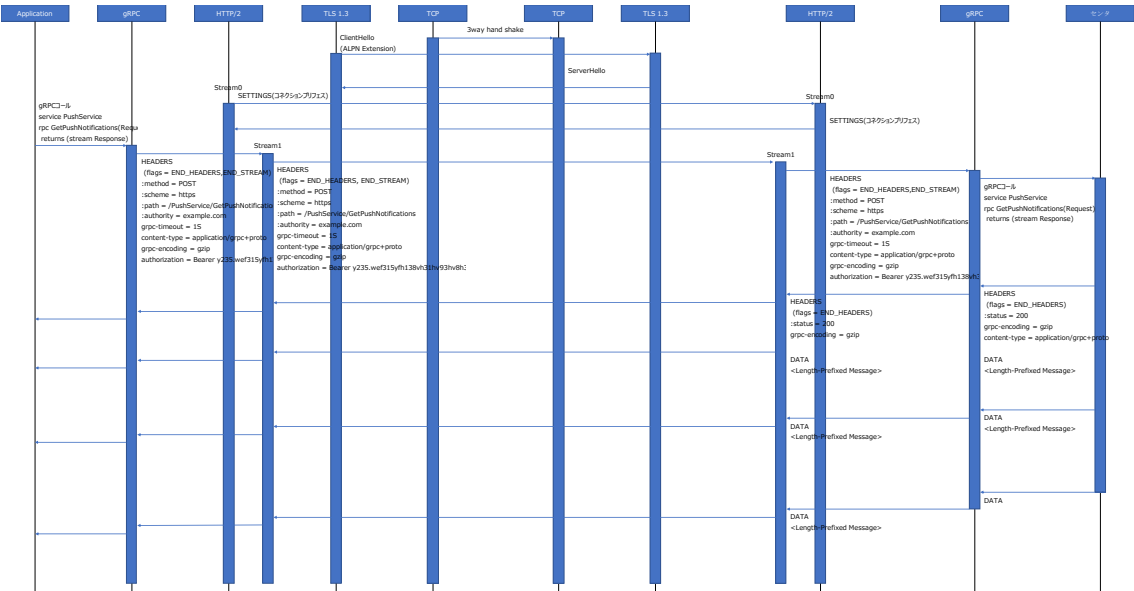


図 A-6 センタからのプッシュシーケンス (gRPC)



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	105
		No.	1.2

4.1.2.3. MQTT

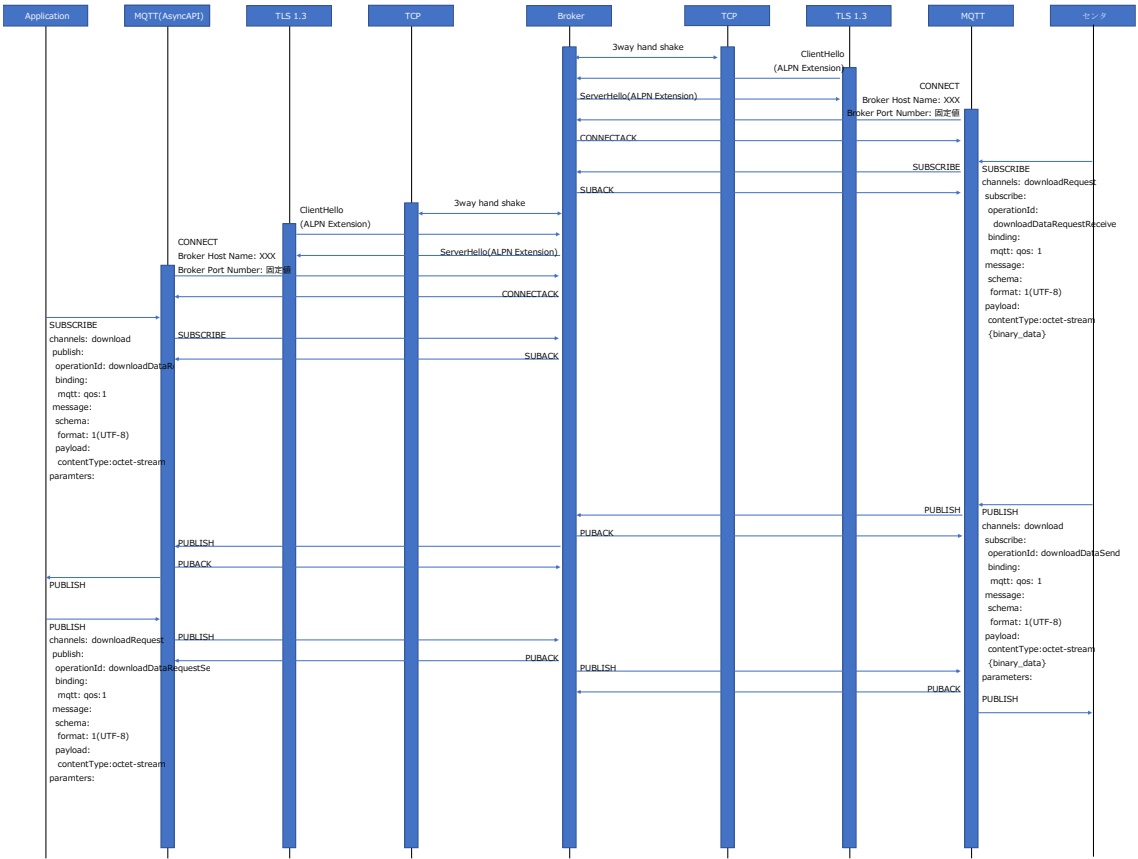


図 A-7 センタからのプッシュシーケンス (MQTT)

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	106
		No.	1.2

### 4.1.3. 車両からセンタへの API 呼出し

#### 4.1.3.1. HTTP/1.1

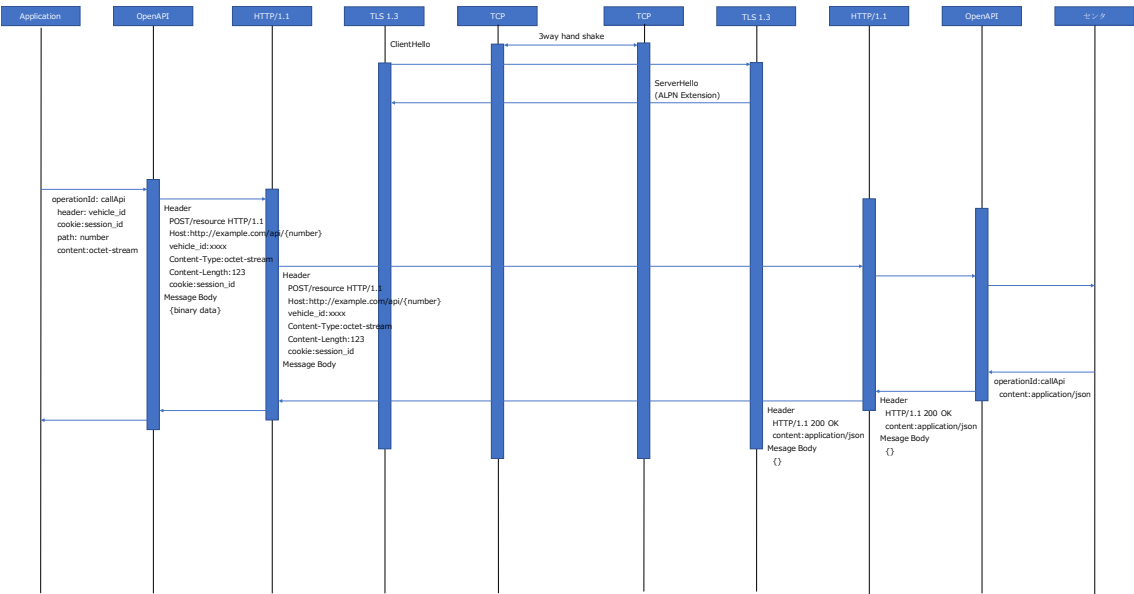


図 A-8 車両からセンタへの API 呼出しシーケンス (HTTP/1.1)



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		108
			No.	1.2

### 4.1.3.3. gRPC

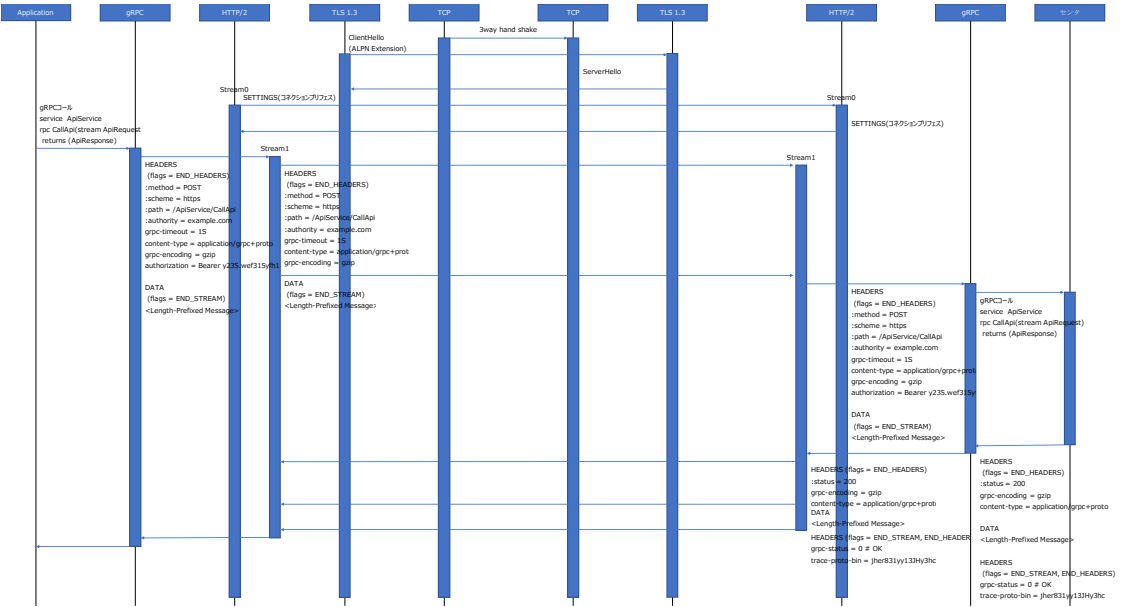


図 A-10 車両からセンタへの API 呼出しシーケンス (gRPC)

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		109
			No.	1.2

### 4.1.4. データダウンロード

#### 4.1.4.1. HTTP/1.1

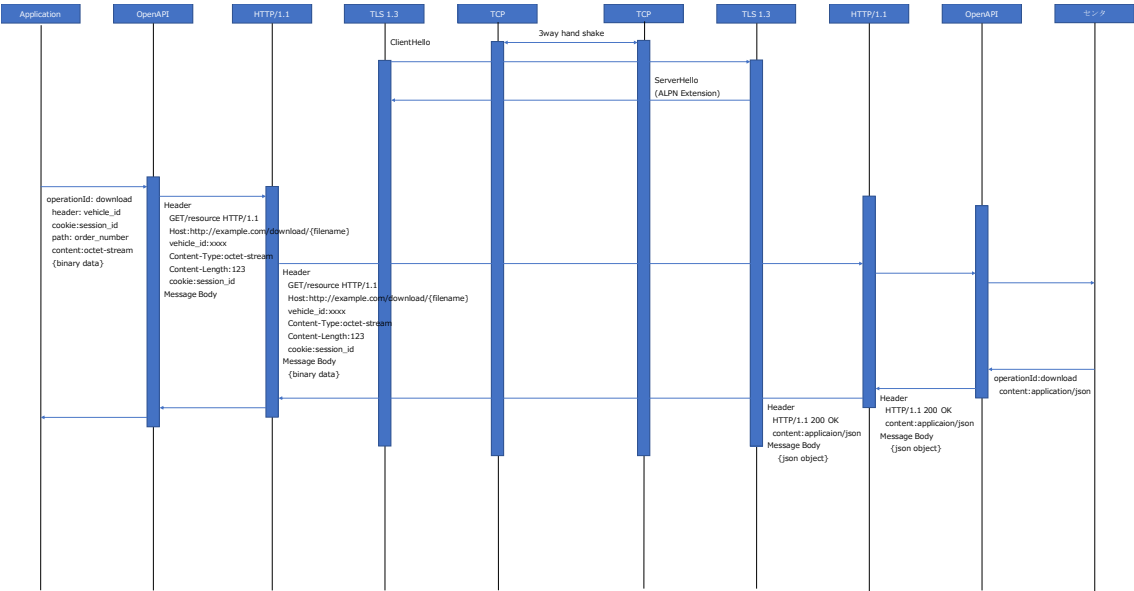


図 A-11 データダウンロードシーケンス (HTTP/1.1)

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	110
		No.	1.2

### 4.1.4.2. HTTP/2

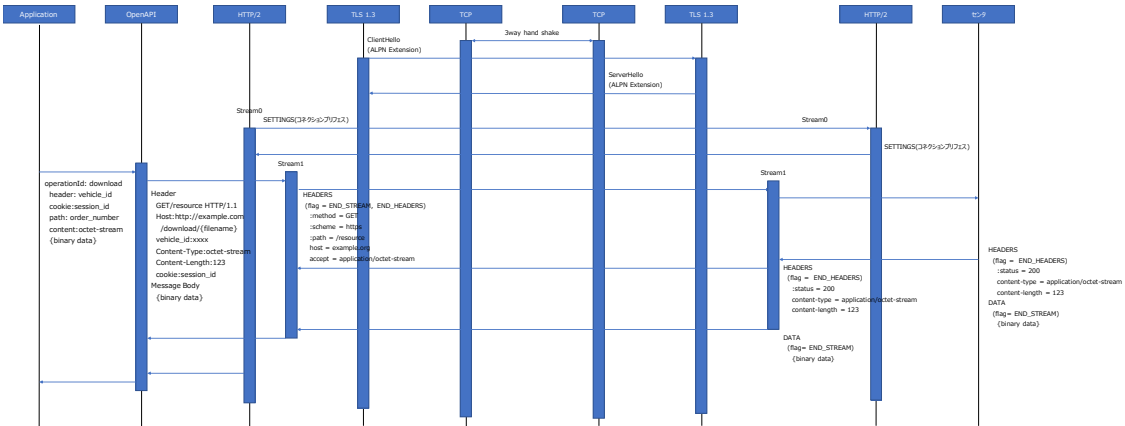


図 A-12 データダウンロードシーケンス (HTTP/2)

4.1.4.3. gRPC

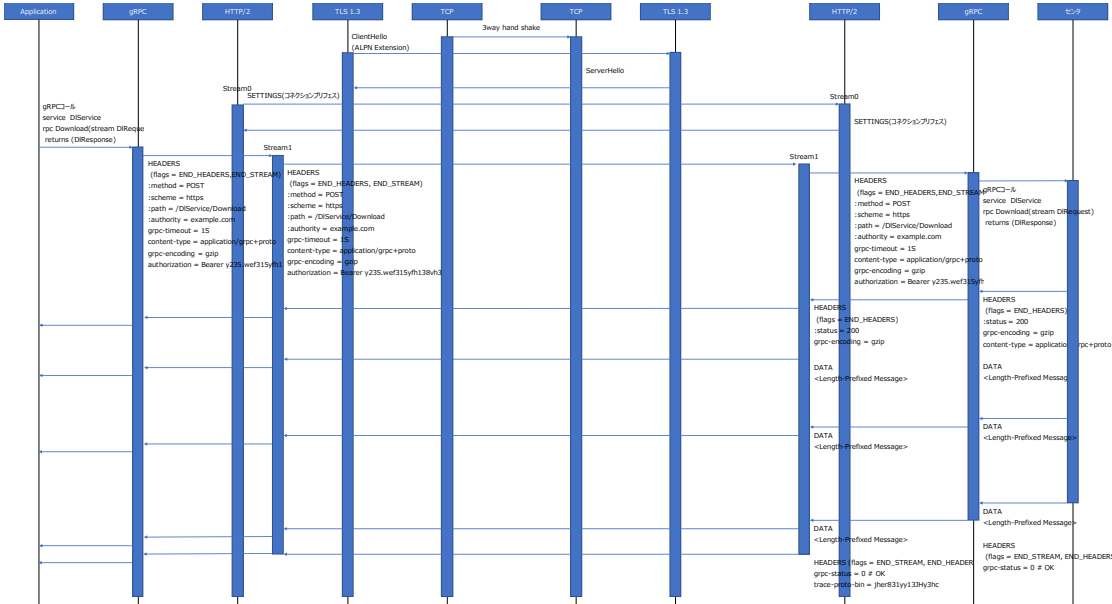


図 A-13 データダウンロードシーケンス (gRPC)





Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		113
			No.	1.2

4.1.5. エラー時シーケンス

4.1.5.1. HTTP/1.1

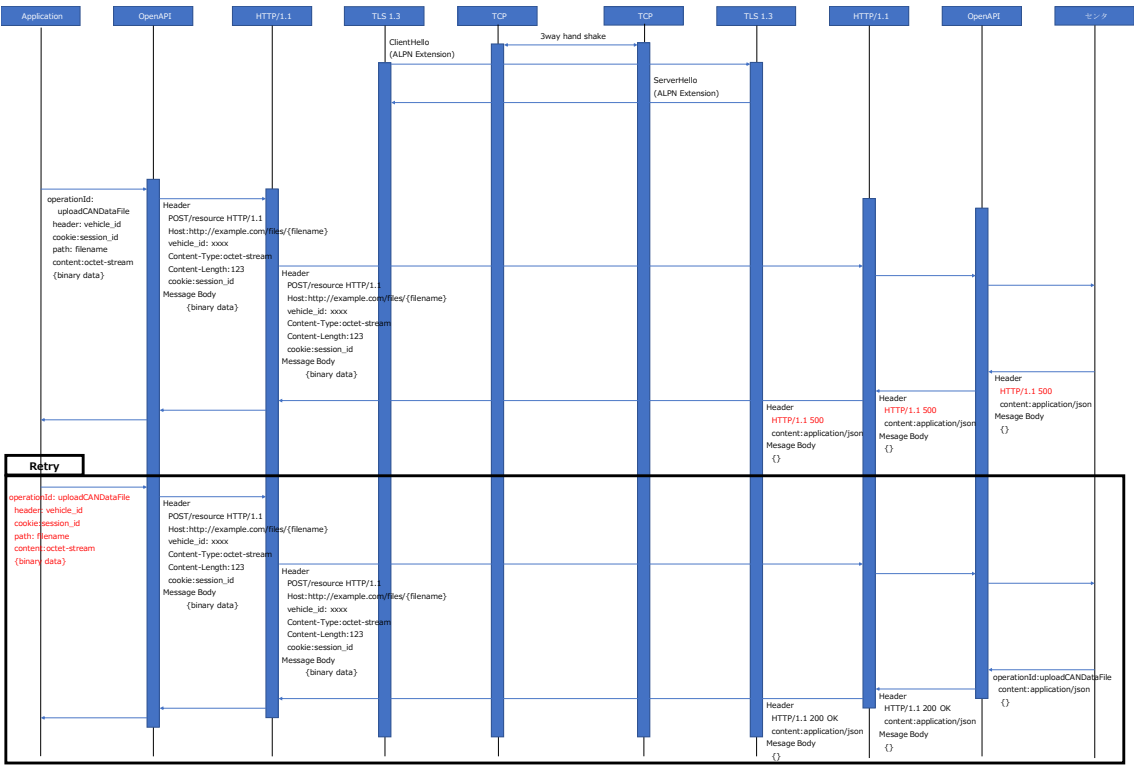


図 A-15 【エラー時のシーケンス】データアップロード（HTTP/1.1）

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	114
		No.	1.2

4.1.5.2. HTTP/2

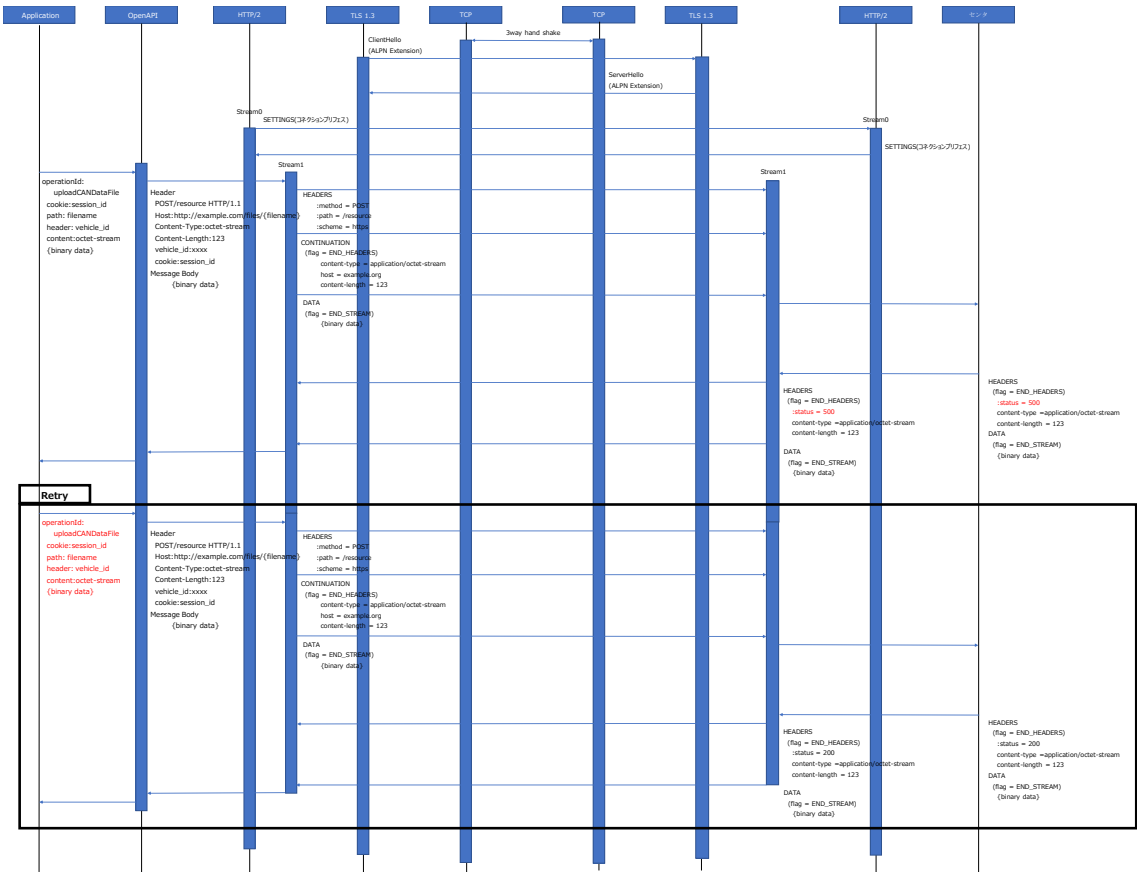


図 A-16 【エラー時のシーケンス】データアップロード（HTTP/2）

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car			115
				No.	1.2

4.1.5.3. gRPC

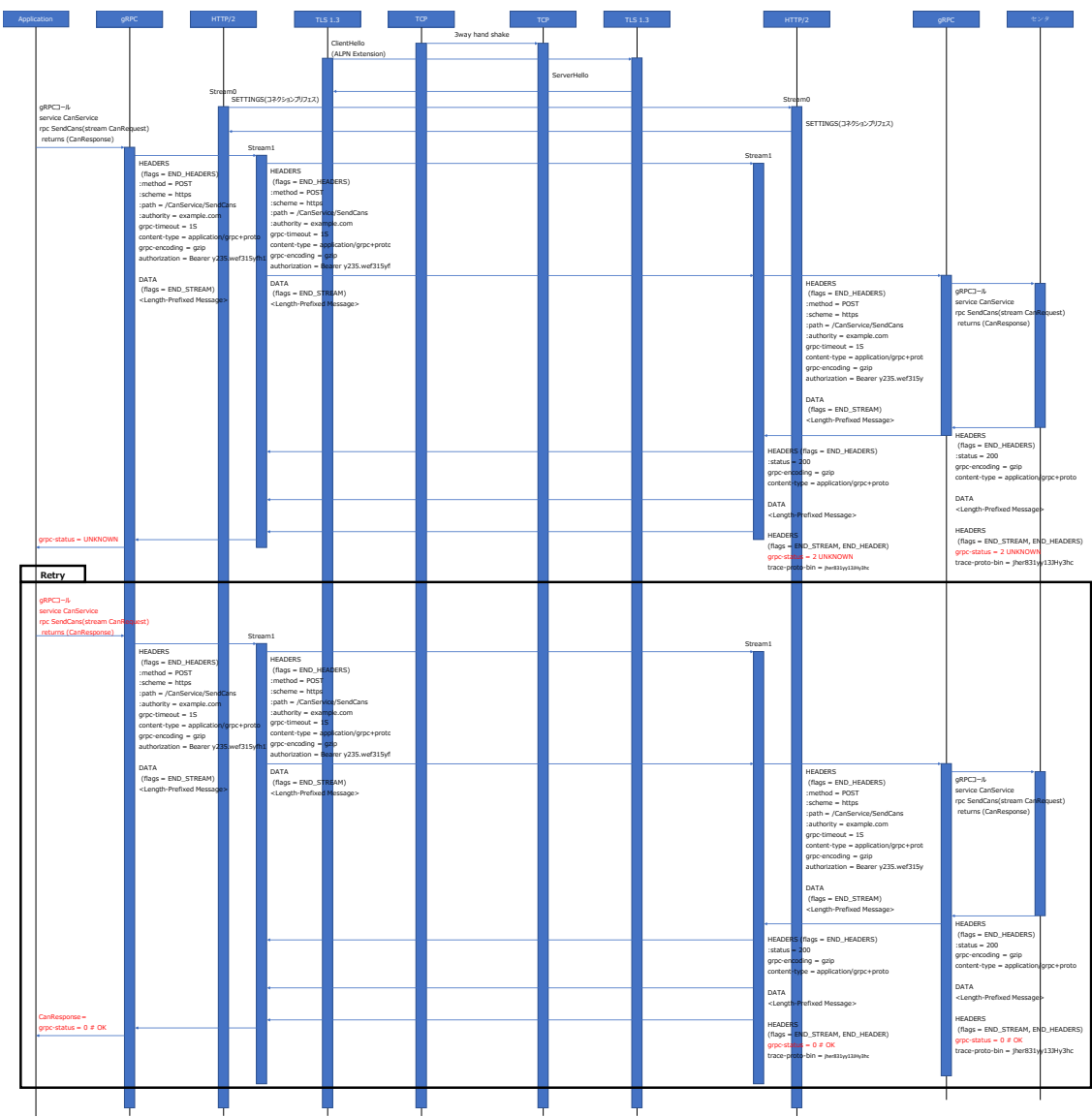


図 A-17 【エラー時のシーケンス】データアップロード (gRPC)

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	116
		No.	1.2

4.1.5.4.
MQTT

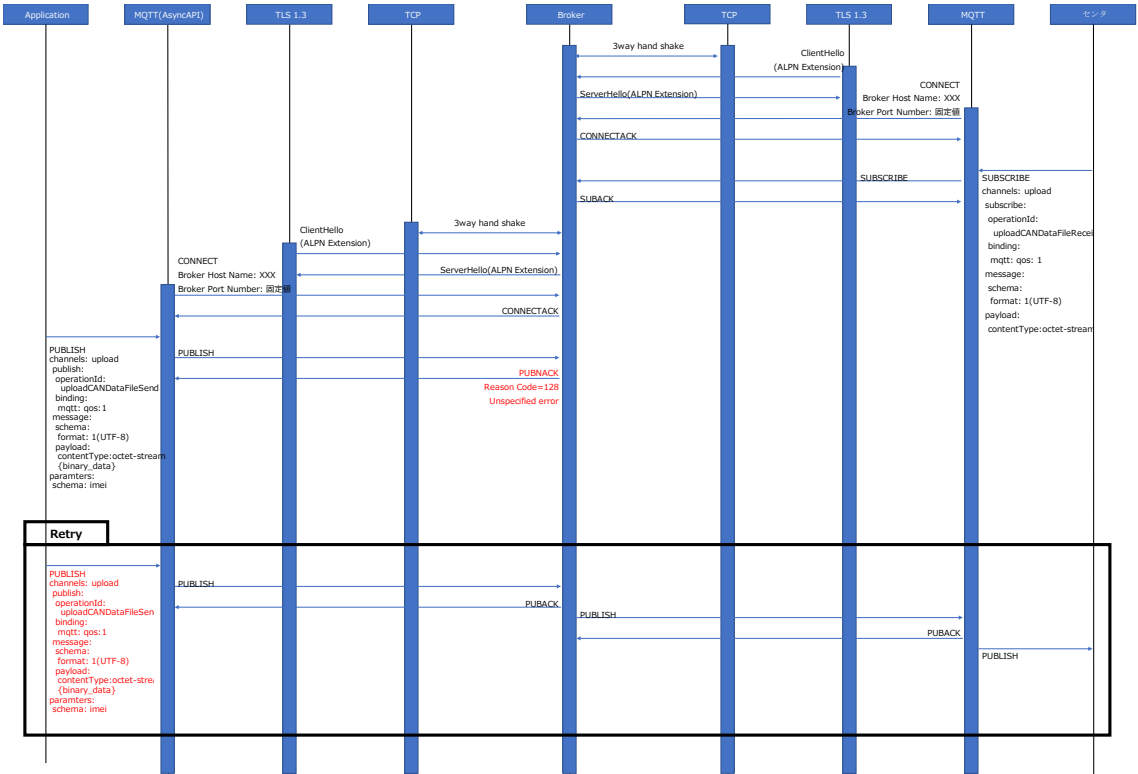


図 A-18 【エラー時のシーケンス】データアップロード（MQTT）

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	117
		No.	1.2

## Appendix 2

当 Appendix は、電子 PF または TSC の構成に今後見直し・改編が発生した場合、それに応じ変更が必要となる内容を記載する。

### 1. コンフィギュレーション

#### 1.1. HTTP/1.1

##### 1.1.1. Base URL

T.B.D（動的に URL を取得する仕組みを検討中）

##### 1.1.2. ポート番号

HTTP リクエストで指定するポート番号は TCP:443(HTTP TLS)とする。

#### 1.2. HTTP/2

##### 1.2.1. Base URL

T.B.D（動的に URL を取得する仕組みを検討中）

##### 1.2.2. ポート番号

T.B.D

#### 1.3. gRPC

##### 1.3.1. Base URL

T.B.D（動的に URL を取得する仕組みを検討中）

##### 1.3.2. ポート番号

HTTP リクエストで指定するポート番号は TCP:9080 とする。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		118
			No.	1.2

## 1.4. MQTT

### 1.4.1. Base URL

T.B.D（動的に URL を取得する仕組みを検討中）

### 1.4.2. ポート番号

MQTT CONNECT で指定するポート番号は TCP:8883（MQTT TLS）とする。

## 2. 認証と接続構成

車載器とセンタ間の認証は、TLS によるクライアント認証を含めた相互認証を行う。

車載器とセンタ間の接続は、セッションの再利用によってコネクション数を低減し、センタにおける負荷低減を図る。

車両とセンタ間で通信コネクションを常時確立する必要がある場合には、センタ機能を複数持たずとも良いように可能な限り車載単位の接続方式としない。

（例：車両—センタ間の常時コネクションが必要なセンタからのプッシュ機能は車載単位ではなく車両単位で基本的に確立する（図 A2-1 参照）。車載—センタ単位で要件を満たす場合には複数のプロトコルを選択しない（図 A2-2 参照）。など）

代表的なユースケースと接続形態を 表 A2-1 に示す。

**表 A2-1 代表的なユースケースと接続形態**

ユースケース	接続形態
CAN データのアップロード	常時接続
映像データのアップロード	都度(イベント毎)接続
ダイアグデータのアップロード	都度(イベント毎)接続
車両からセンタへの API アクセス	都度(イベント毎)接続
センタからのプッシュ	常時接続(DCM-TSC 間)
アクセストークン取得（初回認証）	都度(イベント毎)接続
データダウンロード	都度(イベント毎)接続
OTA	都度(イベント毎)接続

車載器とセンタ間の認証と全体構成を 図 A2-1 に示す。

また、センタからのプッシュに特化した接続構成を 図 A2-2 に示す。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car		119
			No.	1.2

車載器と車両との対応付けにおいて、車載器は、センタとの間でしか知りえない特定の期間や特定の条件のもとで生成される情報をトークンとして用いて、車両識別子や ECU ID の登録、およびテレマティクスサービスのリクエストを行う。

認証トークンをクライアント証明書にバインドするために、RFC8705 に準拠すること。

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	120
		No.	1.2

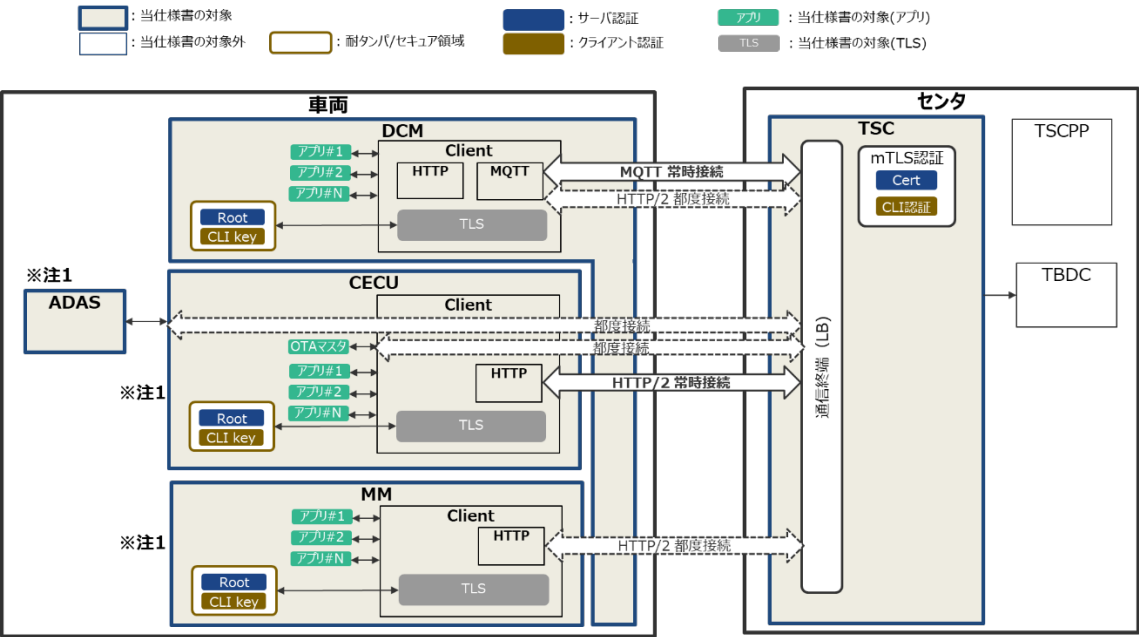


図 A2-1 車両とセンタ間の認証の全体構成

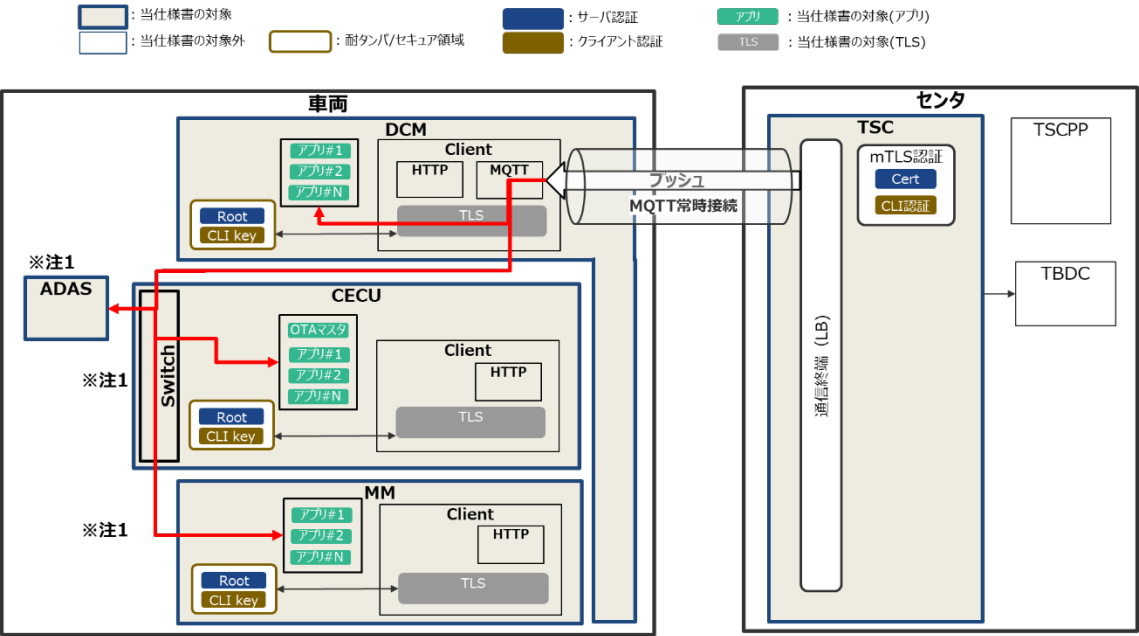


図 A2-2 センタからのプッシュの接続構成



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	121
		No.	1.2

## 変更履歴

Version	Date	Revision	Author
1.2	2021.12.24	<p>OEM 協業の観点で以下修正</p> <ul style="list-style-type: none"> <li>・表紙 社外配布先を明記。社外配布先として OEM を追記。</li> <li>・1.1.目的</li> <li>・1.2.位置づけ</li> <li>・3.1.適用 ECU 一覧</li> <li>・図 3-1 適用 ECU と本書の適用範囲</li> <li>・3.2.適用電子 PF</li> <li>・3.3.適用サーバー一覧</li> <li>・図 4-1 システム構成</li> <li>・Appendix 2</li> <li>・図 A2-1 車両とセンタ間の認証の全体構成</li> <li>・図 A2-2 センタからのプッシュの接続構成</li> </ul> <p>次世代の車載通信機の技術仕様を共同で開発する OEM 向けにトヨタ固有となる表現を修正</p> <p>セキュリティ観点で以下修正</p> <ul style="list-style-type: none"> <li>・6.2.4 アプリケーションレイヤセキュリティ セキュリティ要件から DNSSEC をサポート推奨とし、文言追加</li> <li>・7.2.5.1. 基本仕様 (MQTT) セキュリティ要件の実装方法として拡張認証機能を必要とする可能性があるために削除</li> </ul> <p>MQTT 認証、認可プロセスを追加するために全面的に改訂</p>	
1.1	2021.11.26	<ul style="list-style-type: none"> <li>・6.2.3. 証明書 「チェーンする信頼された証明書のみ」の文言を追加</li> <li>・6.4 車両識別 認証に加えて認可を行う点を追記</li> </ul> <p>認証トークンをクライアント証明書にバインドするために、RFC8705 に準拠すること。</p> <ul style="list-style-type: none"> <li>・7.1.1.1 基本仕様 DNSSEC をサポートする方針とし、文言追加</li> <li>・7.1.1.2.1. 暗号化スイート 強度の観点を踏まえ、優先 1 のみとし優先 2 の記載を削除</li> <li>・7.1.1.2.2. 鍵交換方式 強度の観点を踏まえ、ECDHE を 384 ビット以上と修正</li> <li>・7.1.1.2.3. 署名方式 RSA3072 系統のものをインデントにて区分 (方式の追加</li> </ul>	

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	122
		No.	1.2

Version	Date	Revision	Author
		変更はありません) ・7.1.1.3 クライアント認証 証明書チェーンと正確な文言に修正 ・7.1.1.5. 証明書管理 サーバ認証に OCSP を利用することの追加、CRL/OCSP の キャッシュコントロールの追加を踏まえ、全面的に改訂 図 7-16、図 7-17 を追加 ・7.1.1.7. 異常系シナリオ OCSP 対応につき、表 7-4 の記載事項を追加 ・7.2.5.1. 基本仕様 MQTT 認証、認可プロセスを追加するために全面的に改訂 ・Appendix2 トークンによる認可について一文追加	
1.04	2021.10.01	・3.1 適用 ECU 一覧 ADAS ドメコンを ADASECU 表記に変更 ・図 7-8 HTTP/2 におけるメッセージフォーマット ・図 7-9 HTTP/2 の基本シーケンス (GET) ・図 7-10 HTTP/2 の基本シーケンス (POST) ・図 7-11 gRPC におけるメッセージフォーマット http2 表記に変更 6.1.2 通信プロトコルとユースケース ダイアグデータのアップロードを見直し 7.1.1.2.3 署名方式 将来セキュリティ脅威を踏まえたアルゴ保持を明記 ・7.2.5.5 タイムアウト仕様 タイムアウト時の再送間隔について追記 ・7.2.5.7 セッション管理 セッション確立、維持、破棄の仕様について追記	
1.03	2021.08.26	・1.1. 目的 本書のセンタ側対象範囲について、TSC 以外も含むよう、表現を修正 ・7.1.1.5. 証明書管理 トヨタ発行の証明書の失効確認方法について追記 有効期間満了前でも、+B=OFF 時は保持した CRL を破棄する旨記載を変更 ・7.1.1.7. 異常系シナリオ(TLS) サーバ認証のシーケンスと対応(図 7-3、表 7-4)修正 クライアント認証シーケンスと対応(図 7-4、表 7-5)追加 タイムアウト時間、リトライ間隔、リトライ回数追記(表 7-4、	

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	123
		No.	1.2

Version	Date	Revision	Author
		表 7-5) ・7.2.4.1 基本仕様(gRPC) v.0.9 版で誤って削除した gRPC によるデータ送信時の分割について記載 ・7.2.5.7. セッション管理(MQTT) センタからのプッシュの場合は、keep alive をクライアント側から行い、keep alive の時間間隔は消費電力、ネットワーク側からの切断時間などを考慮して設定する旨追記 ・Appendix1 下位仕様を検討するにあたって、有益となる情報を記載する旨追記 ・Appendix2 p19ePF または TSC の構成に今後見直し・改編が発生した場合、それに応じ変更が必要となる内容を記載する旨追記	
1.02	2021.08.05	・7.1.1.2.1 暗号化スイートの記載を優先度順に変更 ・7.1.1.2.2 DHE の鍵長を 4096 ビット以上とする ・7.1.1.2.3 RSASSA に鍵長を指定 ・7.1.1.5 サーバから OCSP レスpons受信時の車載器側の対応を追記。サーバ側はクライアント証明の失効確認であることを明記 ・誤記修正および校正	
1.00	2021.07.29	・1.2. 位置づけ 図 1 1 本書の位置づけ の参照仕様書を削除 ・7.1.1.7. 異常系シナリオ 図 7-3、表 7-4 は、車載器(ECU)側におけるサーバ認証の異常系のシナリオについて、CRL によるサーバ証明書の失効確認を前提としたシーケンスおよび対応内容である旨追記	
0.92	2021.07.28	・6.1.3. プロトコルスタック 誤記修正 ・7.2.4.3. ヘッダ定義 gRPC の user-agent の説明を修正(製品識別子および gRPC 推奨のライブラリ名をスペース区切りで設定する旨明記) ・図 3-1 適用 ECU と本書の適用範囲 図 4-1 システム構成 図 A2-1 車載器とセンタ間の認証の全体構成 図 A2-2 センタからのプッシュの接続構成 V2X を削除	
0.90	2021.07.21	・ヘッダ、フッタを変更 ・7.1.1.2. 暗号化アルゴリズム	

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	124
		No.	1.2

Version	Date	Revision	Author
		<p>車載器、センタ間双方の暗号化スイートと鍵交換、署名方式であることを記載</p> <p>・7.1.1.4. セッション管理            コネクションの再接続等において、SIGPIPE（一度切断されたソケットを用いた送信におけるプロセスの異常終了）が発生しないよう考慮する旨追記</p> <p>・7.1.1.5. 証明書管理            参照 RFC の間違い修正(8954→6960)</p> <p>・7.2.2.3 ヘッダ定義            Cookie に関する記載を削除            センタでのセッション管理を前提としたステートフルな接続が必要な場合は Set-Cookie、Cookie ヘッダを使用するよう記載内容を変更</p> <p>・7.2.3.3 ヘッダ定義            Cookie に関する記載を追加</p> <p>・7.2.3.7. セッション管理</p> <p>・7.2.4.6. セッション管理            センタでのセッション管理を前提としたステートフルな接続が必要な場合はスティッキーセッション（Cookie）を使用するよう記載内容を変更</p> <p>・7.2.5.5. タイムアウト仕様            アップロードまたはダウンロードに時間を要する場合は対象外とする旨追記</p> <p>・7.2.5.8. 基本シーケンス（MQTT）            Appenx1 4.1.2.3. センタからのプッシュ            図 3-17 MQTT の基本シーケンス（センタからのプッシュ）            図 A 7 センタからのプッシュシーケンス（MQTT）            MQTT のプッシュを双方向のシーケンスに変更</p> <p>・7.3 車両識別            車両を識別できる値として VIN、IMEI 双方への対応が検討である内容に変更</p> <p>・Appendix を Appendix1、Appendix2 に分離</p> <p>・Appendix1            1.1.1. Base URL            1.2.1. Base URL            1.3.1. Base URL            1.3.1. Base URL            T.B.D.であるが、動的に URL を取得する仕組みを検討中である旨追記</p>	

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	125
		No.	1.2

Version	Date	Revision	Author
		<ul style="list-style-type: none"> <li>•Appendix1 2.ファイル名 世代は、電子 PF の世代表現に合わせて 2 桁とする旨追記 機能は、将来の機能の増加を鑑み 3 桁の表現とする旨追記 カウンタ値は、日時が秒単位であり、1 秒以内に発生するトリガが多くても数十で収まる前提において、2 桁の表現とする旨追記</li> <li>•Appendix2 2 認証と接続構成 VIN マッピングについて追記</li> </ul>	
0.55	2021.07.20	<p>4.1. システム構成 ECU と TSC 経由で通信する TSP は本書の対象外とし、ECUと直接通信する TSP のみ本書の対象とする旨追記</p> <ul style="list-style-type: none"> <li>•5. ユースケース 「イベント通知」を「車両からセンタへの API 呼出し」に統合 →説明に、「リモートサービスのイベント通知など」を追記 「リモートコントロール」を「センタからのプッシュ」に統合 →説明に、「リモートサービスにおけるリモートコントロールなど」を追記</li> <li>•7.1.1.2.3. 署名方式 ECDSA256、RSA3072 を追記</li> <li>•7.1.1.4. セッション管理 セッションの有効期間を 12 時間とする旨追記</li> <li>•7.1.1.5. 証明書管理 OCSP、CRL の両方対応とする記載とし、OCSP stapling は非対応であることを記載</li> <li>•7.2.3.3. ヘッダ定義 HTTP/2 のヘッダは RFC7541 に基づき圧縮を行う旨追記</li> <li>•7.2.4.1. 基本仕様 複数ストリームのデータアップロードに関しては、動画などの容量大のデータ通信を行っている最中に他の通信を行うと帯域が不十分となる可能性があるため、データサイズ等に留意する旨追記</li> <li>•7.2.2.7. セッション管理 HTTP/1.1 利用時において、基本的にセッションの維持は行わない。(Cookie を使用しない) 旨記載</li> <li>•7.2.3.7. セッション管理</li> <li>•7.2.4.6. セッション管理 車載器とセンタとの常時接続が必要な場合は、スティッキー</li> </ul>	

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	126
		No.	1.2

Version	Date	Revision	Author
		<p>セッション（Cookie）を使用する内容に変更</p> <p>・7.2.5.2. メッセージフォーマット（MQTT）  PUSH 通知を受信したアプリケーションは、PUSH 通知の PUBLISH に含まれる Response Topic 宛に、応答の PUBLISH を行う旨追記</p> <p>・7.3 車両識別  車両を識別できる値は、VIN または IMEI の何れかを検討中である旨追記</p> <p>・Appendix 2 ファイル名  世代は 2 桁の 10 進表現とする旨追記  ファイル名に機能を追加し、機能毎に 3 桁の 10 進表現とし、機能毎に付与する旨追記  カウンタ値は 2 桁の 10 進表現とし、アプリケーション毎に付与する旨追記  トリガ識別子の具体的を記載</p> <p>・Appendix 章構成  今後の変更の可能性が有るもの、無いものに章立てを分離  変更なし：1～4 章(データフォーマット、ファイル名、文字コード、通信シーケンス)  変更の可能性あり：5、6 章(コンフィギュレーション、認証と接続構成)</p>	
0.54	2021.07.19	<p>・2. 用語定義  NEV、VIN を追加</p> <p>・4.1 システム構成  TLS Proxy 経由の通信において、不特定の ECU からの接続を許可しない旨追記</p> <p>・5. ユースケース  ダイアグデータのユースケース変更  イベント通知、リモートコントロールのユースケース追加</p> <p>・7.1.1.7. 異常系シナリオ  表 3 8 をリトライ、通信不可(終了)条件と、アクションに分けて記載</p> <p>・7.1.1.4. セッション管理</p> <p>・7.2.2.7. セッション管理  ALB 利用時において、セッションの維持が必要な場合、ステイキーセッション（Cookie）を使用する向け説明を変更</p> <p>・7.1.1.5. 証明書管理  サーバ証明書は CRL、クライアント証明書は OCSP により失効確認を行う内容で修正</p>	

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	127
		No.	1.2

Version	Date	Revision	Author
		<ul style="list-style-type: none"> <li>・7.2.2.2. メッセージフォーマット</li> <li>7.2.2.9. API 仕様</li> <li>7.2.3.2. メッセージフォーマット</li> <li>7.2.3.9. API 仕様</li> <li>7.2.4.2. メッセージフォーマット</li> <li>7.2.4.4. API 仕様</li> <li>7.2.5.2. メッセージフォーマット</li> <li>7.2.5.9. API 仕様</li> <li>パス名(HTTP)、トピック名(MQTT)にアプリケーション名を追加</li> <li>・7.2.2.4. レスポンスステータスコード HTTPレスポンスのステータスコード 567を修正（次回 IG-ON まで通信しない）</li> <li>・7.2.2.5. タイムアウト仕様 タイムアウトに対する条件を追記</li> <li>・7.2.2.6 リトライ仕様 法規要件は対象外とする旨追記</li> <li>・7.2.4.1.基本仕様 gRPC のストリーム利用時のサイズへの留意について追記</li> <li>・7.2.5.7. セッション管理 KeepAlive はクライアント側から行う旨追記</li> <li>・7.3 車両識別 車両識別子として VIN または IMEI を使う旨追記</li> <li>・Appendix2 ファイル名 世代管理はファイル名の重複がないようにするためであることを例と共に追記 トリガ識別子はアプリケーション固有の識別子は重複しない英数字一文字とする旨追記 リクエスト ID がなかった場合、バイナリの場合"FFF..."、text や json の場合は"000..."を設定する旨追記 カウンタ値はアプリケーション毎に持つ旨追記 UTC による時刻表記は RFC3339 に基づく旨追記</li> <li>・Appendix 6 図 A 20 センタからのプッシュの接続構成 CECU に対する HTTP/2 によるプッシュを削除</li> </ul>	
0.53	2021.07.13	<ul style="list-style-type: none"> <li>・7.2.4.1 基本仕様(gRPC) 送信するデータサイズが MB を超える場合の分割サイズを追記</li> <li>・7.2.4.4. API 仕様(gRPC) 車両識別子(vehicle_id)は、proto ファイルではなく、</li> </ul>	

Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	128
		No.	1.2

Version	Date	Revision	Author
		Meta-Data としてアプリケーションによる設定を行う旨追記 ・7.2.5.2. メッセージフォーマット(MQTT) トピック名の付与に関するルールを追記 ・Appendix 6 図 A-19 を修正(MQTT は DCM のみ、HTTP は HTTP/2 とする) 図 A-20 センタからのプッシュの接続構成を追加	
0.52	2021.07.08	・2 用語定義 UTC を追加 ・Appendix 2.ファイル名 -GPS 未受信等で時刻が不明な場合の時刻表記を修正 (00000000_000000 とする) -日時は UTC による旨追記	
0.51	2021.07.05	・2 用語定義 mTLS を追加 ・図 3 1 適用 ECU と本書の適用範囲 図から冗長や矢印を削除 ・7.1.1.3. クライアント認証 RFC8446 に基づきクライアント認証を行うよう記載 ・7.1.1.5 証明書管理 車載器とサーバの双方要件である(サーバ証明書とクライアント証明書それぞれ失効確認が必要である)旨追記・修正 ・7.2.1 アプリケーションデータ User-Agent の内容を製品識別子として記載 7.2.2.2. メッセージフォーマット パス・クエリパラメータの冗長な説明を削除 ・7.2.2.3 ヘッダ定義 User-Agent、製品識別子の説明を修正 ・7.2.2.7 セッション管理 ・7.2.3.7 セッション管理 ・7.2.4.6 セッション管理 Cookie はセッション維持が必要な場合のみとする旨追記 ・7.2.2.9.1 データアップロード ・7.2.2.9.3 車両からの API 呼出し ・7.2.2.9.4 データダウンロード ・7.2.3.9.2 センタからのプッシュ yaml ファイルの不備を修正 ・7.2.4.3. ヘッダ定義 情報管理キーを製品識別子に修正	



Communication Specification	System	Common Specification for the Communication Interface between In-Car and Out-Car	129
		No.	1.2

Version	Date	Revision	Author
		・Appendix6 認証と接続構成 車載器とセンタ間の認証と、代表的なユースケース毎の接 続形態を追記	
0.5	2021.06.25	初版	