SURFACE VEHICLE RECOMMENDED PRACTICE

J2847™/6 SEP2020

Issued 2015-08
Revised 2020-09

Superseding J2847/6 AUG2015

(R) Communication for Wireless Power Transfer Between Light-Duty Plug-in Electric Vehicles and Wireless EV Charging Stations

RATIONALE

SAE J2847/6 defines abstract messages supporting the wireless transfer of energy between EVs and the wireless charger (WEVSE). This document is based on the use cases in SAE J2836/6 that established the wireless charging requirements.

TABLE OF CONTENTS

1. SCOPE

SAE J2847/6 establishes minimum requirements for communication between an electric vehicle and an inductive battery charging system for wireless power transfer (WPT). Where relevant, this document notes—but does not formally specify—interactions between the vehicle and vehicle operator.

This document leverages the work of the SAE J2954 Alignment and Controls Sub-Team in the Wireless Power Transfer and Alignment Task Force by extending a JSON-based message set (protocol) originally developed to bench test wireless energy transfer interoperability between unmatched Ground Assembly (GA) and Vehicle Assembly (VA) systems (i.e., components manufactured by different companies). SAE J2847/6 furthers that work by adding messages sufficient to indicate that proper coil alignment has been achieved, initialize the sub-systems for wireless charging, ramp-up to full power, perform active wireless power transfer, and terminate the WPT session.

Guidance for an engineering implementation of the JSON protocol is provided as a reference for developing verification and validation of test plans by enabling the GA and VA to transmit a basic set of information using standard wireless IEEE 802.11n (Wi-Fi) communications hardware.

1.1 Purpose

The primary purpose of SAE J2847/6 is to specify minimum communication requirements for control of wireless power transfer (WPT, or inductive charging) using a standard Wi-Fi interfact between a WPT charging station and an electric vehicle, irrespective of manufacturer or variations in the inductive charging technology employed.

2. REFERENCES

2.1 Applicable Documents

The following publications form a part of this specification to the extent specified herein. Unless otherwise indicated, the latest issue of SAE publications shall apply.

2.1.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or +1 724-776-4970 (outside USA), www.sae.org.

2.2 Required SAE Publications (Mandatory)

SAE J2836-6 Use Cases for Wireless Charging Communication for Plug-in Electric Vehicles

SAE J2931-6 Signaling Communication for Wirelessly Charged Electric Vehicles

SAE J2954 Wireless Power Transfer for Light-Duty Plug-in/Electric Vehicles and Alignment Methodology

2.3 Related Publications

The following publications are provided for information purposes only and are not a required part of this SAE Technical Report.

2.3.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or +1 724-776-4970 (outside USA), www.sae.org.

SAE J1715 Hybrid Electric Vehicle (HEV) and Electric Vehicle (EV) Terminology

SAE J2836-1 Use Cases for Communication Between Plug-in Vehicles and the Utility Grid

SAE J2847-1 Communication for Smart Charging of Plug-in Electric Vehicles Using Smart Energy Profile 2.0

SAE J2847-2 Communication Between Plug-in Vehicles and Off-Board DC Chargers

2.3.2 ISO Publications

Copies of these documents are available online at http://webstore.ansi.org/.

ISO 10731:1994 Open Systems Interconnection - Basic Reference Model - Conventions for the Definition of OSI Services

ISO 15118-8:2018 Road Vehicles - Vehicle to Grid Communication Interface - Part 8: Physical Layer and Data Link Layer Requirements for Wireless Communication

2.3.3 IEC Publications

Available from IEC Central Office, 3, rue de Varembe, P.O. Box 131, CH-1211 Geneva 20, Switzerland, Tel: +41 22 919 02 11, www.iec.ch.

IEC 61980-1:2015 Electric Vehicle Wireless Power Transfer (WPT) Systems - Part 1: General Requirements

IEC 61980-2:2019 Electric Vehicle Wireless Power Transfer (WPT) Systems - Part 2: Specific Requirements for Communication Between Electric Road Vehicle (EV) and Infrastructure

IEC 61980-3:2019 Electric Vehicle Wireless Power Transfer (WPT) Systems - Part 3: Specific Requirements for the Magnetic Field Wireless Power Transfer Systemscharging System - Part 1: General Requirements

3. DEFINITIONS

3.1 AP (ACCESS POINT)

A hardware device or computer's software functioning as a communication hub for users of a IEEE 802.11 (Wi-Fi) client devices, allowing the Wi-Fi client to connect to the wireless local area network (WLAN).

3.2 CAN (CONTROLLER AREA NETWORK)

Vehicle bus standard allowing microcontrollers to communicate via a message-based protocol as defined by SAE J1939.

3.3 DHCP (DYNAMIC HOST CONFIGURATION PROTOCOL)

A network managment protocol for automatically and dynamically assigning a unique IP address to a new device connecting to the network.

3.4 EV (ELECTRIC VEHICLE)

An automobile as defined in 49 CFR 523.3. See also 3.13.

3.5 EVCC (ELECTRIC VEHICLE COMMUNICATIONS CONTROLLER)

Generic term for the EV module responsible for communicating with the supply equipment communication controller (SECC) using a high-level digital protocol in order to implement enhanced mobility and vehicle-to-grid services.

3.6 GA (GROUND ASSEMBLY)

Wireless power transfer charging infrastructure consisting of the GA coil, power/frequency conversion unit, wiring from the electric grid, filtering circuits, housing(s), and communication required to safely transfer power to the Vehicle Assembly (VA).

3.7 GA-CP (GROUND ASSEMBLY COMMUNICATION PROXY)

A software application part of a Ground Assembly (GA) wireless charging subsystem capable of translating proprietary inductive charging control information into a common set of protocol-based messages for the Vehicle Assembly (VA) subsystem in order to control wireless power transfer as specified in this standard.

3.8 HTTP (HYPERTEXT TRANSPORT PROTOCOL)

A request-response protocol that forms the foundation for data communication for the World Wide Web, defining how messages are formatted and transmitted and what actions web servers and browsers should take in response to various commands.

3.9 IP (INTERNET PROTOCOL)

A set of rules and technical format for addressing and routing packets of data (called datagrams) between computers on the same network or across a series of interconnected networks. IP addresses are assigned to every computer, printer, switch, router, or other device part of a TCP/IP-based network. IP is a connectionless protocol offering no guarantee of packet delivery, sequencing, or error detection and correction.

3.10 IPv4 (INTERNET PROTOCOL VERSION 4)

IPv4 defines IP addresses using a 32-bit format, most often written in dot-decimal notation consisting of four octets expressed individually in decimal notation separated by periods. Devices in IPv4 networks either need to be configured manually or with a DHCP server.

3.11 IPv6 (INTERNET PROTOCOL VERSION 6)

IPv6 was released in 2012 to provide large scalability with addresses represented as eight groups of four hexadecimal digits separated by colons. IPv6 also supports auto-stateless configuration not requiring use of a DHCP server to obtain IPv6 addresses.

3.12 JSON (JAVASCRIPT OBJECT NOTATION)

An open standard human-readable text-based data interchange format designed for transmitting structured data. JSON is viewed as a more compact alternative to XML because it does not require tags for each element.

3.13 LIGHT-DUTY PHEV (LIGHT-DUTY PLUG-IN ELECTRIC VEHICLE)

A three- or four-wheeled vehicle rated at less than 4545 kg gross vehicle weight (GVW), intended primarily for use on public streets, roads, and highways; propelled solely by an electric motor; and powered by a rechargeble on-vehicle energy storage system, such as a battery.

3.14 OSI MODEL (OPEN SYSTEMS INTERCONNECTION MODEL)

An ISO worldwide standard defining a conceptual and logical layout of network communication used by systems to interconnect and communicate. Broken into seven layers (or subcomponents) each with a discrete set of functional services, control of an application passes from one layer to the next.

3.15 SECC (SUPPLY EQUIPMENT COMMUNICATIONS CONTROLLER)

Generic term for the inductive or conductive charging infrastructure module responsible for communicating with the EV communication controller (EVCC) using a high-level digital protocol in order to implement enhanced mobility and vehicle-to-grid services.

3.16 SSID (SERVICE SET IDENTIFIER)

The network name associated with a IEEE 802.11 (Wi-Fi) wireless local area network access point, used by Wi-Fi client devices to identify and join a particular Wi-Fi private or public network.

3.17 TCP (TRANSMISSION CONTROL PROTOCOL)

TCP is connection-oriented protocol that allows two hosts to establish a virtual connection and exchange streams of data. Because IP is a connectionless protocol, most networks combine IP with transmission control protocol (TCP) in order to guarantee delivery of data and that packets arrive at their intended destination in the same order they were sent.

3.18 UDP (USER DATAGRAM PROTOCOL)

UDP is a simple open systems interconnection (OSI) transport layer protocol used for client-server network applications to send short messages called datagrams. Known as a “stateless” protocol, it does not guarantee datagram delivery, nor does it employ handshaking dialogs for reliability, ordering, and data integrity.

3.19 VA (VEHICLE ASSEMBLY)

The electric vehicle side of a wireless power transfer system consisting of the VA coil, power electronics, control, and communication components necessary to safely transfer power for battery charging from the Ground Assembly (GA).

3.20 VA-CP (VEHICLE ASSEMBLY COMMUNICATION PROXY)

A software application part of the Vechile Assembly (VA) wireless charging subsystem capable of translating proprietary inductive charging control information into a common set of protocol-based messages for the Ground Assembly (GA) subsystem in order to control wireless power transfer as specified in this standard.

3.21 WEVSE (WIRELESS ELECTRIC VEHICLE SUPPLY EQUIPMENT)

See also 3.6.

3.22 WLAN (WIRELESS LOCAL AREA NETWORK)

Two or more computing devices that form a local area network (LAN) using wireless communication such as Wi-Fi within a limited area such as a home, school, office building, campus, or private facility.

3.23 WPA2 (WI-FI PROTECT ACCESS 2)

A method of securing a Wi-Fi WLAN network using pre-shared key (PSK) authentication, designed for home or small facility users without an enterprise authentication server. WPA2-AES has been used on all certified Wi-Fi hardware since 2006 and is the default choice for newer routers.

4. COMMUNICATIONS SYSTEM ARCHITECTURAL CONFIGURATIONS

4.1 Functional Decomposition of a WPT System

This section describes a simple decomposition of the WPT system into communications and power transfer components (PTCs). PTCs consist of all elements of the system involved in power transfer (such as power electronics, rectifiers, coils, etc.). Off-board PTCs are associated with the wireless charging infrastructure and on-board PTCs are the components installed on the electric vehicle.

The GA is the WPT subsystem responsible for transmitting energy wirelessly to a vehicle-mounted VA, which receives the energy and uses it to charge the vehicle battery. According to SAE J2954, the GA and VA must also share a Wi-Fi communication system allowing the GA to know the state of the VA, and for the GA to receive messages from the VA regarding frequency, power, and current requirements among other critical information needed to initialize and efficiently transfer power between the two subsystems.

An example WPT system is illustrated in Figure 1, along with a high-level view of communication control interfaces of the GA subsystem and a VA subsystem installed on an electric vehicle.

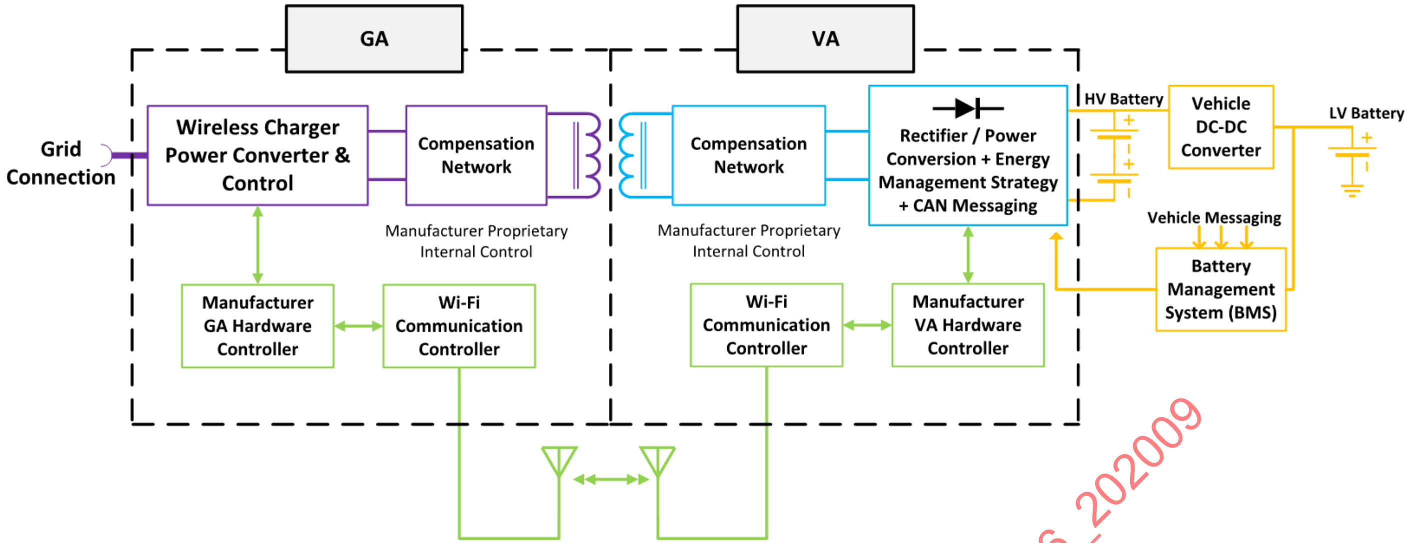


Figure 1 - Functional decomposition of a WPT system

If VA and GA subsystems are manufactured by the same entity (i.e., matched component system), the Wi-Fi WLAN communication interface is likely integrated into the manufacturer's proprietary internal control loop as illustrated in Figure 1. In this instance, the controller application allows VA and GA subsystems to exchange wireless charging control information in a proprietary format over a standard Wi-Fi communication interface.

4.2 Communication Proxy Application

The communication proxy application defined in this standard enables VA and GA subsystems to communicate basic information necessary to control inductive charging using a JSON-based message set (protocol). In this document, the component running the WPT proxy application in the VA is referred to as the VA communication proxy (VA-CP). Correspondingly, the entity running the proxy application in the GA will be referred to as the GA communication proxy (GA-CP).

Implementation of the SAE J2847/6 communication proxy application requires, at a minimum:

- Communication interface with proprietary manufacturer GA and VA WPT hardware controllers.
- Translation of manufacturer proprietary WPT control parameters to SAE J2847/6 JSON message format.
- Interface with an IEEE 802.11n Wi-Fi chipset or Wi-Fi capable device.
- [VA = Client; GA = Access Point].
- For lab bench testing of unmatched VA and GA subsystems, the SAE J2847/6 communication proxy may easily be configured to run as an application on a laptop computer with integrated 802.11n Wi-Fi as shown in Figure 2.

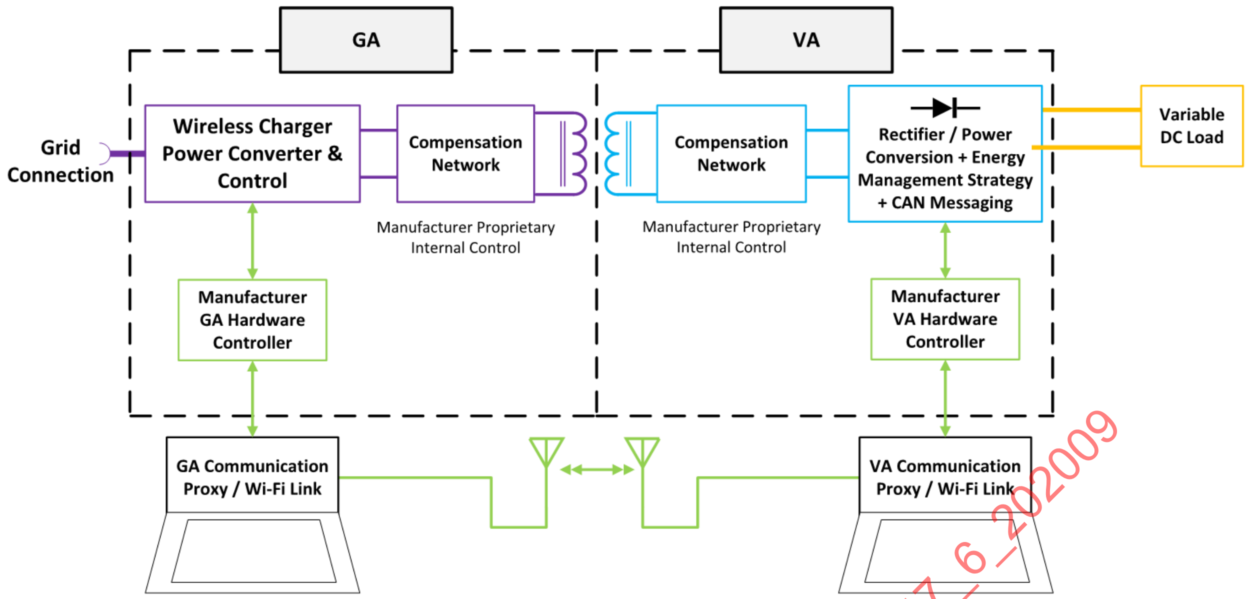


Figure 2 - GA and VA with SAE J2847/6 communication proxy for lab bench testing

The proxy application differs from specialized communication controllers that support globally interoperable enhanced mobility services such as digital authorization and payment, load management, smart charging, and other vehicle-to-grid (V2G) energy services. Integration of these specialized high-level communication modules with existing vehicle and/or supply equipment hardware and ECUs can be complex and costly. Additionally, support for digital encryption and specialized key storage may be required.

The basic function of VA-CP and GA-CP applications is to translate a limited number of parameters needed to initialize, control, and terminate a WPT charging session into standardized JSON-based messages. Figure 3 illustrates the SAE J2847/6 communication proxy as a separate system entity configured for vehicle-level testing. However, due to the small amount of memory and processing required by the application, it is feasible that VA-CP and GA-CP applications could optionally be integrated with an existing manufacturer VA/GA Wi-Fi radio system, OEM Wi-Fi radio module, or third-party aftermarket wireless systems.

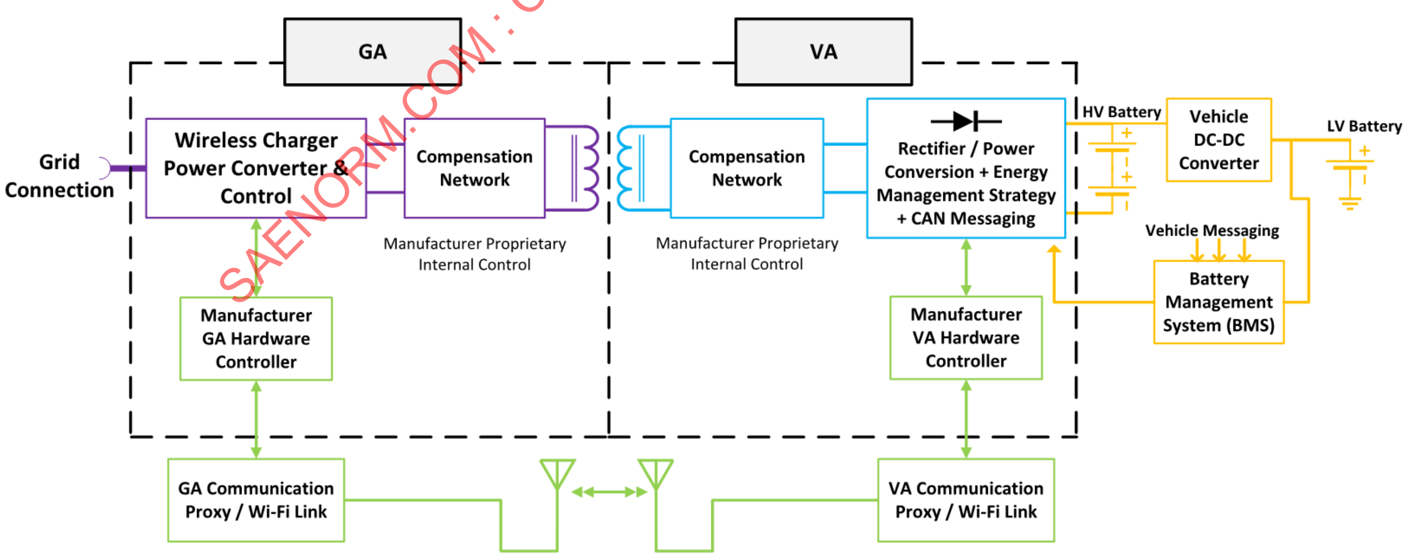


Figure 3 - GA and VA with SAE J2847/6 communication proxy for vehicle-level testing

4.3 Dedicated Configuration

The protocol specified in this document applies to a one-to-one VA-CP and GA-CP Wi-Fi connection. In a dedicated VA-GA configuration, each off-board PTC has its own communication proxy. Such a configuration has the advantage of isolating failure modes to a single charging spot.

5. TECHNICAL REQUIREMENTS

5.1 Overview of OSI Layers

SAE J2647/6 is based on internet protocol (IP). The layers of the network stack may be understood in context of the open systems interconnection (OSI) model. Table 1 summarizes the OSI layers used. Refer to ISO 10731 for more information about the OSI model.

Table 1 - OSI layers

OSI Layer #	OSI Layer Name	Network Protocol(s)	Document Section(s)
7	Application Layer	SAE J2847/6 Application Layer Messages, DHCP	5.8, 7, 8, 10, 11
6	Presentation Layer	JSON	5.7
5	Session Layer	HTTP	5.6
4	Transport Layer	TCP, UDP	5.5
3	Network Layer	IPv4	5.4
2	Data Link Layer	IEEE 802.11	5.3
1	Physical Layer	IEEE 802.11	5.2

5.2 Physical Layer

5.2.1 IEEE 802.11

IEEE 802.11, commonly also known as "Wi-Fi," is a communications standard that specifies both physical and data link layers for J2847/6. The latest version of IEEE 802.11 referred to in this document is IEEE Std 802.11n-2009. Table 2 summarizes required and optional IEEE 802.11 specifications for J2847/6.

Table 2 - IEEE 802.11 specifications

	Required	Optional
IEEE 802.11b compatibility	X	
IEEE 802.11g compatibility		X
IEEE 802.11n compatibility		X
2.4 GHz band operation	X	
5.0 GHz band operation		X

5.3 Data Link Layer

5.3.1 IEEE 802.11

At the data link layer, the IEEE 802.11 Wi-Fi architecture used by SAE J2847/6 consists of an GA-CP as the WLAN access point and a VA-CP as the WLAN station. A VA-CP associates with an GA-CP using the SSID and WPA2 pre-shared key of the network advertised by the GA-CP.

Table 3 - WLAN specifications

WLAN Access Point	GA-CP
WLAN Station	VA-CP
Security	WPA2

5.3.2 Wi-Fi WLAN Security

Per 5.3.1, SAE J2847/6 requires the use of WPA2 to secure the Wi-Fi WLAN. WPA2 implements mandatory elements of the IEEE 802.11i-2004 standard, including use of an enhanced data cryptographic encapsulation mechanism based on the AES standard to address vulnerabilities of wired equivalent privacy (WEP). Certification to the Wi-Fi protected access 2 (WPA2) authentication mechanism is required for all new devices bearing the Wi-Fi trademark of the Wi-Fi Alliance. Specific requirements for implementation of WLAN security mechanisms and frameworks for 802.11 networks is beyond the scope of this standard.

Because security researchers have documented weaknesses in the WPA2 protocol, deployment of SAE J2847/6 in a public environment is not recommended without additional security measures not covered in this document. Generally speaking, SAE J2847/6 is recommended for use only on WLAN networks deployed in private environments, test facilities, or laboratories (where restricted access to WPT hardware and Wi-Fi network equipment is assumed).

If for any reason an elevated level of WLAN security is required, use of the extensible authentication protocol (EAP) framework as defined in RFC 5247 to secure the Wi-Fi WLAN connection is recommended. EAP is a Layer 2 authentication protocol that can be applied to wired and wireless networks. The framework does not mandate the use of a specific authentication method. For this reason, multiple EAP types can be used in a WLAN security deployment. Certain EAP methods provide for certificate-based authentication of clients and servers in a Wi-Fi network.

EAP-transport layer security (EAP-TLS) is supported by all manufacturers of wireless LAN hardware and software, and is considered to be one of the most secure EAP standards currently available, provided users and network administrators understand potential vulnerabilities surrounding potential use of false credentials (e.g., "man-in-the-middle" attacks) by unauthorized entities. EAP-TLS provides mutual authentication but requires both client-side and server-side certificates.

EAP-TTLS (tunneled transport layer security) requires only server-side certificates. However, EAP-TTLS deployments can optionally utilize client-side certificates as well in order to provide mutual client and network authentication through an encrypted channel (or tunnel) and derive dynamic per-session WEP keys.

One major drawback of EAP-TLS and EAP-TTLS is that certificates must be managed on both the server and/or client devices, which can prove to be a cumbersome task. Certificates used for EAP network access must meet X.509 certificate requirements as well as requirements for secure connections using secure sockets layer (SSL) encryption and transport level security (TLS) encryption. In addition, a RADIUS server would be required in the network to validate client credentials as well as client (supplicant) software on the VA-CP capable of securely storing private keys.

5.4 Network Layer

5.4.1 IPv4

The IP network layer is responsible for routing IP packets within and across networks. The network layer for SAE J2847/6 is required to support internet protocol Version 4 (IPv4), described in RFC 791. Support for IPv6 is optional, but support for IPv4 is mandatory. IPv6 is not backwards compatible with IPv4.

Table 4 - Internet protocol

	Required	Optional
IPv4	X	
IPv6		X

5.5 Transport Layer

5.5.1 User Datagram Protocol (UDP)

User datagram protocol (UDP) is a connectionless transport layer protocol that offers low latency compared to transmission control protocol (TCP) but does not guarantee reliable data transfer. The UDP broadcast feature makes it useful for discovery protocols. In SAE J2847/6, UDP is only used by DHCP during IP address assignment. See 5.8.1 for information on DHCP. UDP is specified in RFC 768.

5.5.2 Transmission Control Protocol (TCP)

Transmission control protocol (TCP) is a connection-based transport protocol used for reliable data transfer with built-in features such as flow control and congestion control. It is used as the transport layer for reliable application layer message exchange in SAE J2847/6. The GA-CP operates as a TCP server and the VA-CP operates as a TCP client. TCP is specified in RFC 793.

Table 5 - TCP specifications

TCP Server	GA-CP
TCP Client	VA-CP

5.6 Session Layer

5.6.1 Hypertext Transfer Protocol (HTTP)

Table 6 - HTTP specifications

HTTP Version	1.1
HTTP Server	GA-CP
HTTP Client	VA-CP
Application Request Message Transport	HTTP PUT
Application Response Message Transport	200 "OK"
Port Number	80
Host	www.weccp.com
Request-URI	/messages

A domain name system (DNS) is not be used to resolve the specified host name to an IP address, since the network architecture consists of a one-to-one connection between the VA-CP and GA-CP. Hence, the host name specified in Table 6 will not be used other than to satisfy the HTTP 1.1 specification. Instead, the VA-CP will establish the HTTP connection by directly using the GA-CP's IP address sent to the VA-CP by the GA-CP during the DHCP setup process (5.8.1).

An example HTTP request message containing JSON data in the message body follows:

```
PUT /messages HTTP/1.1
Host: www.weccp.com
Date: Wed, 12 Sept 2018 1:41:18 EDT
Content-Type: application/json
{
    "TerminateCommunicationsRequest": {
        "MessageID": 480,
        "StatusCode": "OK"
    }
}
```

An example HTTP response message containing JSON data in the message body follows:

```
HTTP/1.1 200 "OK"

Content-Type: application/json
{
    "TerminateCommunicationsResponse": {
        "MessageID": 481,
        "ResponseCode": "OK"
    }
}
```

5.7 Presentation Layer

JavaScript object notation (JSON) is a text-based, human-readable data format that organizes data into name-value pairs. JSON is specified in RFC 7159. In SAE J2847/6, JSON provides a format for application layer message contents and supports the development of schema to define those messages. See Section 12 for the SAE J2847/6 application layer message schema.

5.8 Application Layer

5.8.1 Dynamic Host Configuration Protocol (DHCP)

Dynamic host configuration protocol (DHCP) is a protocol by which a DHCP client discovers a DHCP server and receives a dynamically-assigned IP address and other network settings from the DHCP server. In SAE J2847/6, DHCP is used by the VA-CP to acquire an IP address for itself from the GA-CP and to receive the GA-CP's own IP address. The VA-CP then uses the GA-CP's own IP address to establish a direct HTTP connection with the GA-CP HTTP server. DHCP is specified in RFC 2131.

5.8.2 Application Layer Messages Overview

The SAE J2847/6 application layer messages are comprised of a series of request-response message pairs. Utilizing the synchronous HTTP protocol, SAE J2847/6 request and response messages are synchronous by nature; the VA-CP always sends a request mssage and waits for the corresponding response message from the GA-CP before sending the next request message. This enables a well-understood and predictable message sequence and ordering. Using a synchronous protocol also considers future work involving a single GA-CP that is responsible for communications for multiple GAs; multiple VA-CP clients may connect to a single GA-CP server and send data without overwhelming the GA-CP server with many asynchronous requests.

6. SYSTEM STATES

System states form a basis for what behaviors are expected from the GA and VA in any given scenario, at a system level. They intend to transcend specific communication protocols, but do inform what information must be exchanged in the communications link in order for those behaviors to occur.

6.1 VA System States

The system state diagram from the perspective of the VA is shown in Figure 4.

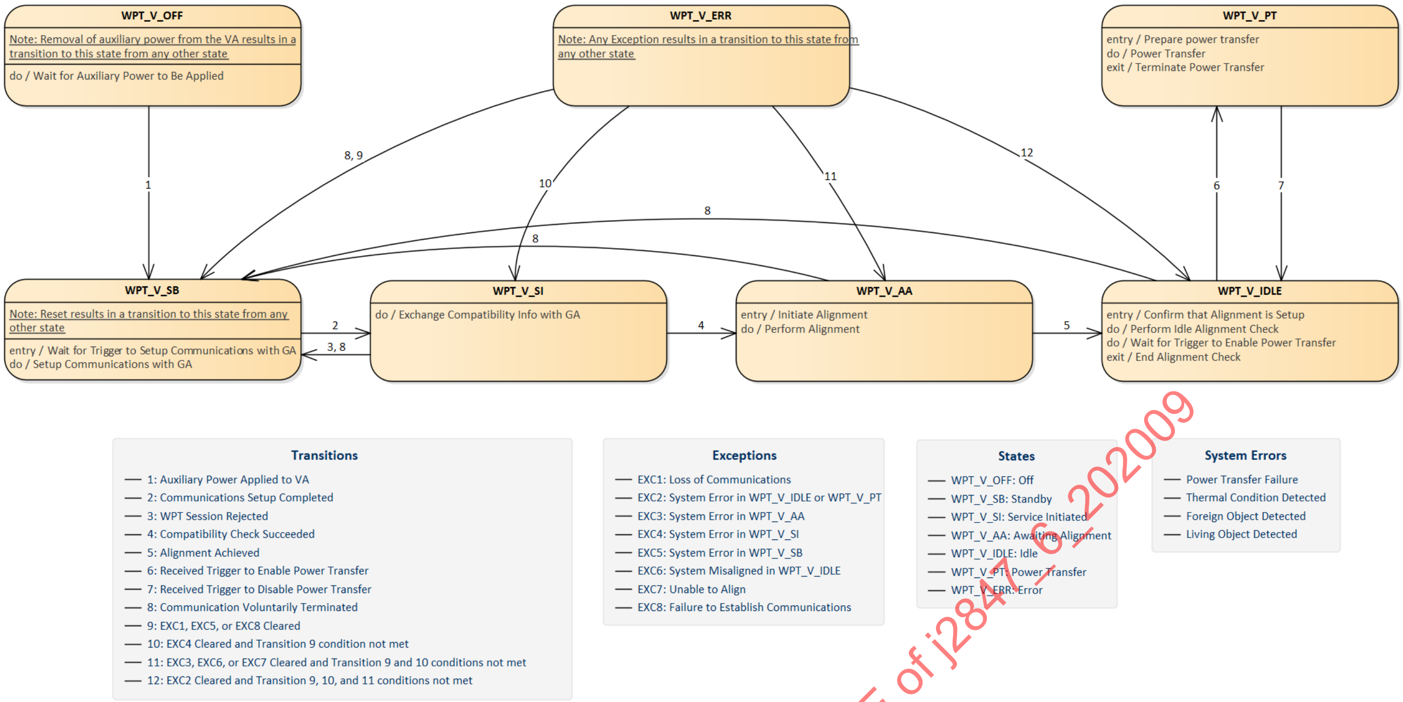


Figure 4 - VA system state diagram

6.2 GA System States

The system state diagram from the perspective of the GA is shown in Figure 5.

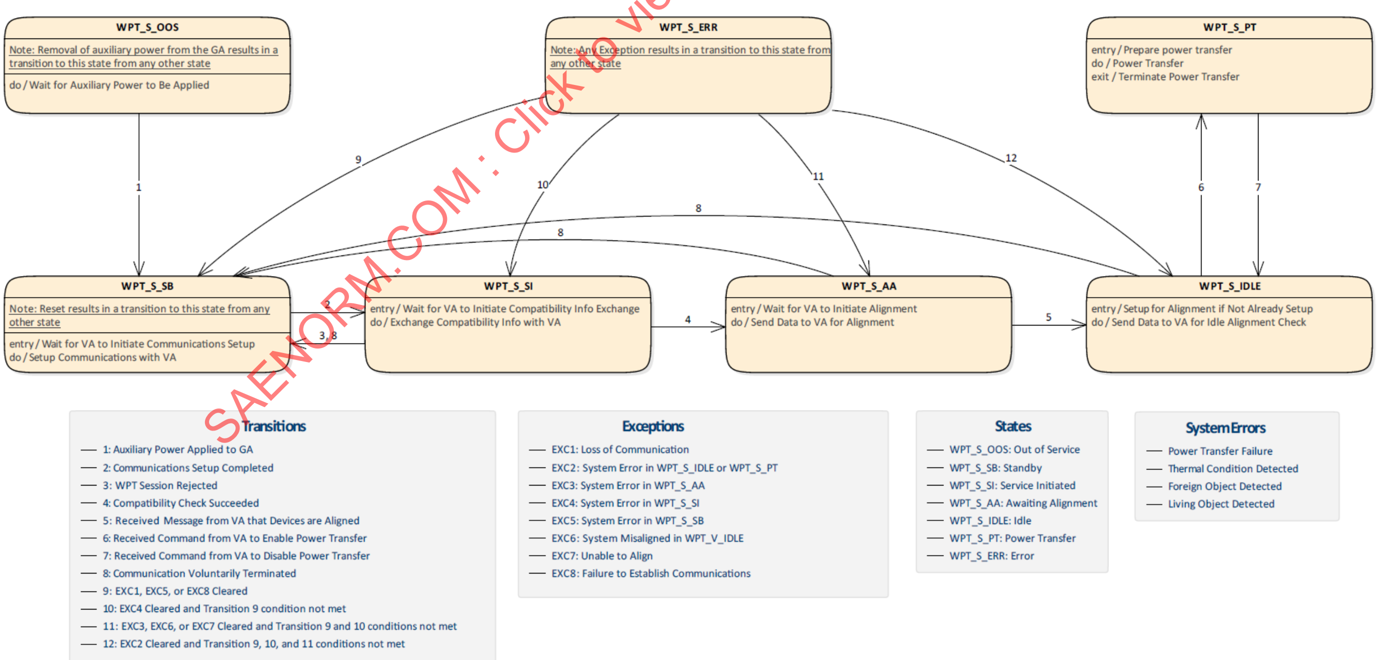


Figure 5 - GA system state diagram

6.3 Transitions Between VA and GA System States

Section 7 specifies the messages to be applied by the communication proxy application in the following VA and GA system states:

SI InitialRequest/InitialResponse Messages

AA FinePositioningRequest/FinePositoningResponse Messages

IDLE FinePositioningRequest/FinePositioningResponse Messages

PT PowerRequest/PowerResponse Messages and
TerminatePowerRequest /TerminatePowerResponse Messages

ERR StatusExchangeRequest/StatusExchangeResponse Messages

Transitions between states occur based on criteria given in Table 7. Also see Section 9 for examples of representative interactions between VA-CP and GA-CP during various message exchange and associated state transitions.

Table 7 - Criteria for state transitions

State(s)	Messages
1 & 2	Physical state of hardware and low-level communication setup. No SAE J2847/6 messages associated with these transitions.
4	Occurs when VA-CP sends GA-CP InitialRequest and the GA-CP sends InitialResponseCode “OK.” Transition only occurs if VA-CP causes the transition if it is also OK (i.e., compatible).
3	Occurs when either the GA-CP or VA-CP determines there is incompatibility (i.e., if not transition 4, then 3).
5	Occurs when FinePositioningRequest has AlignStatusCode = “Aligned” and FinePositioningResponseCode = “OK.”
6	Occurs when GA-CP receives PowerRequest message.
7	Occurs when VA-CP receives TerminatePowerResponse Code “OK.”
9-10-11-12	Occurs when VA-CP StatusExchangeRequest/StatusCode is “OK” and the Exception Code is “None,” and the GA-CP responds the same way.
8	Occurs when TerminateCommunicationRequestStatus/ResponseCodes = “OK.”
Transition to Error State	Occurs when a StatusCode or ResponseCodes = “Fail.”

6.4 Scheduling Power Transfer

Power transfer can be scheduled by completing all prior transitions into the power transfer (PT) state at which time the VA can request zero power from the GA. The VA can wait until a designated time or based on a desired schedule and then can choose when to request that the GA ramp up the power to the level desired by the VA.

7. APPLICATION LAYER MESSAGES

7.1 InitialRequest

7.1.1 Message Overview

The InitialRequest message is sent from VA-CP to GA-CP to initiate the exchange of compatibility information. See 11.1 for message usage. See 13.1 for the JSON schema for this message.

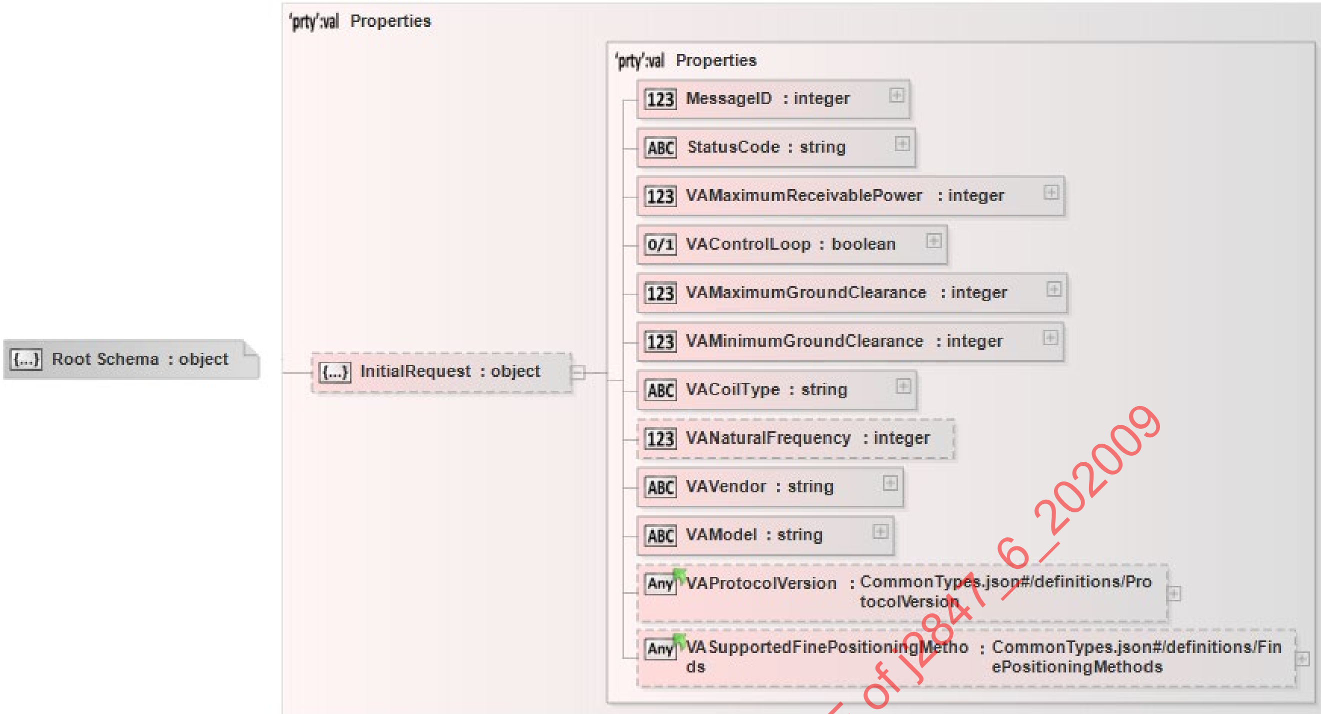


Figure 6 - InitialRequest schema diagram

Table 8 - InitialRequest properties

Property Name	Type	Required	Description
MessageID	Integer	Yes	An integer value where value ≥ 0 and value ≤ 65535. Initialize to 0. Must be incremented from last message received.
StatusCode	String	Yes	Section 7.1.2: StatusCode.
VAMaximumReceivablePower	Integer	Yes	An integer value where value ≥ 0 and value ≤ 22000. Number of watts that the VA is willing to receive, as measured at the output of the VA.
VAControlLoop	Boolean	Yes	A boolean value “true” if VA executes a control loop, “false” otherwise.
VAMaximumGroundClearance	Integer	Yes	An integer value where value ≥ 100 and value ≤ 250. Maximum height of the VA in mm.
VAMinimumGroundClearance	Integer	Yes	An integer value where value ≥ 100 and value ≤ 250. Minimum height of the VA in mm.
VACoilType	String	Yes	Section 7.1.3: VACoilType.
VANaturalFrequency	Integer	Yes	Section 7.1.4 VANaturalFrequency in Hz.
VAVendor	String	Yes	A string value where length ≥ 1 and length ≤ 64. Name of the VA vendor.
VAModel	String	Yes	A string value where length ≥ 1 and length ≤ 64. Model of VA device.
VAProtocolVersion	Object	No	Section 8.2: ProtocolVersion.
VASupportedFinePositioningMethods	Array	No	Section 8.3: FinePositioningMethods.

7.1.2 StatusCode

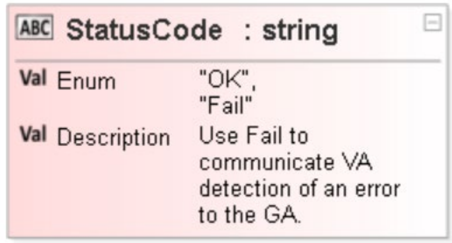


Figure 7 - StatusCode schema diagram

7.1.3 VACoilType

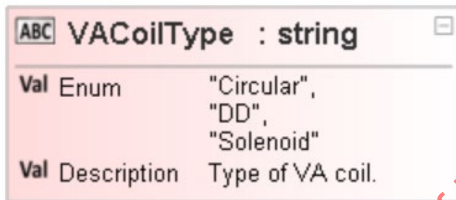


Figure 8 - VACoilType schema diagram

7.1.4 VANaturalFrequency

The self-resonant frequency of the VA under fully loaded condition with no GA present.

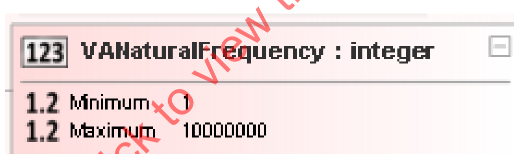


Figure 9 - VANaturalFrequency schema diagram

7.1.5 Example InitialRequest Message


```
{
    "InitialRequest": {
        "MessageID": 0,
        "StatusCode": "OK",
        "VAMaximumGroundClearance": 210,
        "VAMinimumGroundClearance": 140,
        "VACoilType": "Circular",
        "VANaturalFrequency": 85500,
        "VAVendor": "Tech Company",
        "VAModel": "Rocket Charger",
        "VAControlLoop": true,
        "VAMaximumReceiveablePower": 9000,
        "VAProtocolVersion": {
            "Namespace": "WECCP",
            "MajorVersionNumber": 0,
            "MinorVersionNumber": 1
        },
        "VASupportedFinePositioningMethods": [
            "Proprietary",
            "LPE",
            "LF"
        ]
    }
}
```


7.2 InitialResponse

7.2.1 Message Overview

The InitialResponse message is sent from GA-CP to VA-CP in response to an InitialRequest message. See 11.2 for message usage. See 13.2 for the JSON schema for this message.

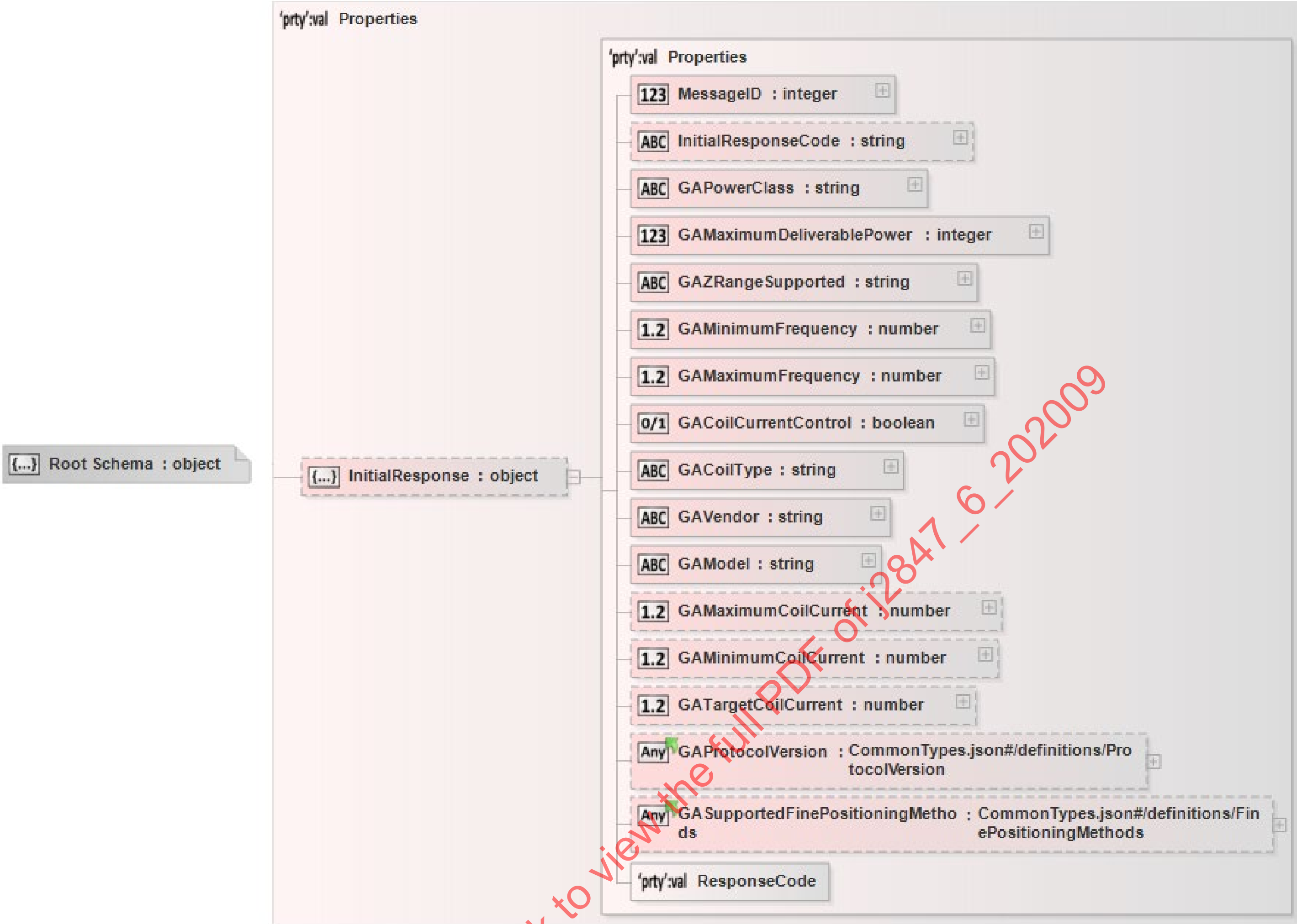


Figure 10 - InitialResponse schema diagram

Table 9 - InitialResponse properties

Property Name	Type	Required	Description
MessageID	integer	Yes	An integer value where value ≥ 0 and value ≤ 65535. Must be incremented from the last message received.
InitialResponseCode	string	Yes	Section 7.2.2: InitialResponseCode.
GAPowerClass	string	Yes	Section 7.2.3: GAPowerClass.
GAMaximumDeliverablePower	integer	Yes	An integer value where value ≥ 0 and value ≤ 22000. Number of watts that the GA is willing to transmit, as measured at the input of the GA.
GAZRangeSupported	String	Yes	Section 7.2.4: GAZRangeSupported.
GAMinimumFrequency	Number	Yes	A number value where value ≥ 79 and value ≤ 90 and value mod 0.001 = 0. Maximum frequency the GA supports in kHz (e.g., 82.5).
GAMaximumFrequency	Number	Yes	A number value where value ≥ 79 and value ≤ 90 and value mod 0.001 = 0. Maximum frequency the GA supports in kHz (e.g., 82.5).
GACoilCurrentControl	Boolean	Yes	A boolean value "true" indicates the GA has explicit control over the current entering the GA coil, "false" indicates no explicit current control.
GACoilType	String	Yes	Section 7.2.5: GACoilType.
GAVendor	String	Yes	A string value where length ≥ 1 and length ≤ 64. Name of the GA vendor.
GAModel	String	Yes	A string value where length ≥ 1 and length ≤ 64. Model of the GA.
GAMaximumCoilCurrent	Number	No	A number value where value ≥ 0 and value ≤ 127 and value mod 0.1 = 0. Maximum current the GA can deliver into the GA coil.
GAMinimumCoilCurrent	Number	No	A number value where value ≥ 0 and value ≤ 127 and value mod 0.1 = 0. Minimum current the GA can deliver into the GA coil.
GATargetCoilCurrent	Number	No	A number value where value ≥ 0 and value ≤ 127 and value mod 0.1 = 0. Target current into the GA coil.
GAProtocolVersion	Object	No	Section 8.2: ProtocolVersion
GASupportedFinePositioningMethods	Array	No	Section 8.3: FinePositioningMethods

7.2.2 InitialResponseCode

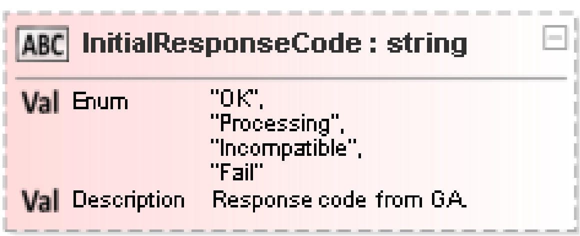


Figure 11 - InitialResponseCode schema diagram

7.2.3 GAPowerClass

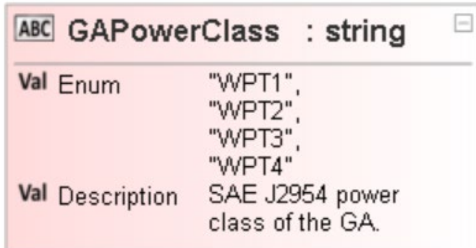


Figure 12 - GAPowerClass schema diagram

7.2.4 GAZRangeSupported

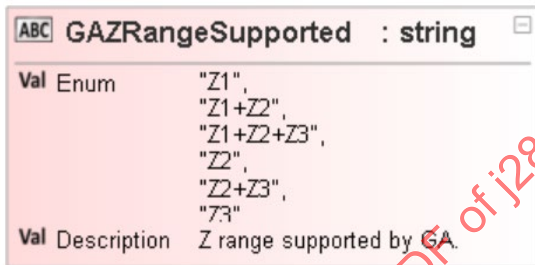


Figure 13 - GAZRangeSupported schema diagram

7.2.5 GACoilType

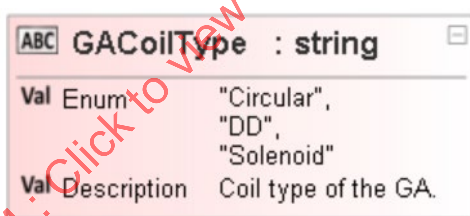


Figure 14 - GACoilType schema diagram

7.2.6 Example InitialResponse Message


```
{
    "InitialResponse": {
        "MessageID": 1,
        "InitialResponseCode": "OK",
        "GAPowerClass": "WPT3",
        "GAMaximumDeliverablePower": 10000,
        "GAZRangeSupported": "Z3",
        "GAMinimumFrequency": 79.0,
        "GAMaximumFrequency": 90.0,
        "GACoilCurrentControl": false,
        "GACoilType": "Circular",
        "GAVendor": "Turbo Chargers",
        "GAModel": "GA11K",
        "GAProtocolVersion": {
            "Namespace": "WECCP",
            "MajorVersionNumber": 0,
            "MinorVersionNumber": 1
        },
        "GASupportedFinePositioningMethods": [
            "Proprietary",
            "LPE",
            "LF"
        ]
    }
}
```


7.3 FinePositioningRequest

7.3.1 Message Overview

The FinePositionin Request message is sent from VA-CP to GA-CP to initiate and perform fine positioning of the GA and VA devices. See 11.1 for message usage. See 13.3 for the JSON schema for this message.

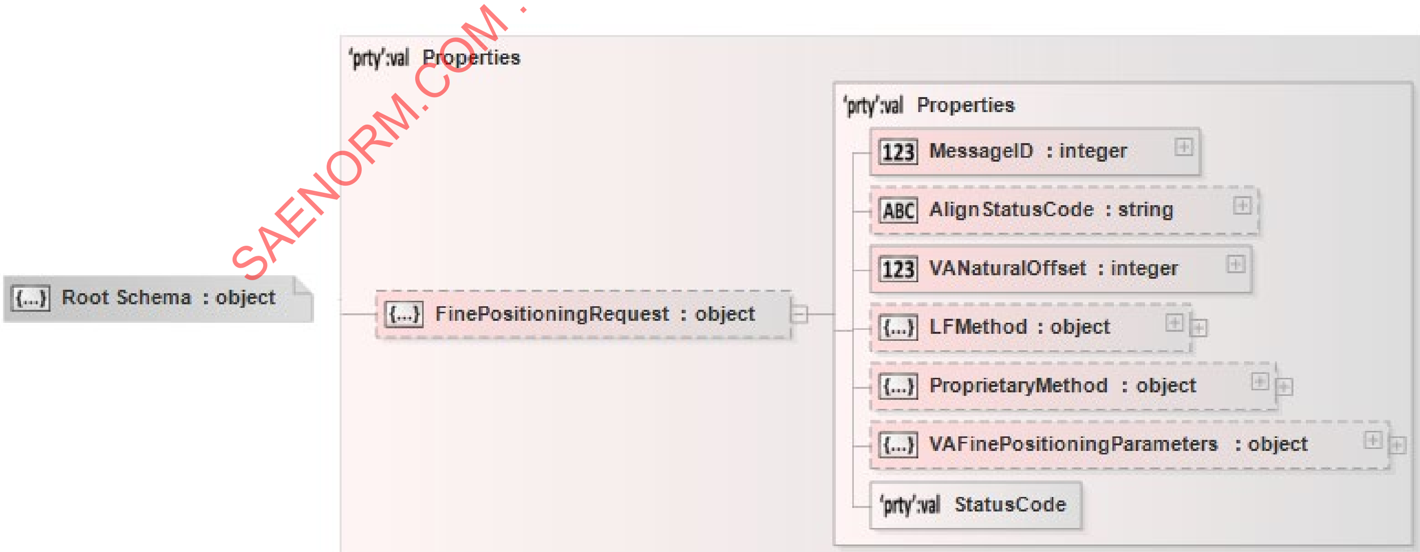


Figure 15 - FinePositioningRequest schema diagram

Table 10 - FinePositioningRequest properties

Property Name	Type	Required	Description
MessageID	Integer	Yes	An integer value where value ≥ 0 and value ≤ 65535. Must be incremented from the last message received.
AlignStatusCode	String	Yes	Section 7.3.2: AlignStatusCode.
VANaturalOffset	Integer	Yes	An integer value where value ≥ -32768 and value ≤ 32767. Indicates offset of the center alignment point in mm (in x-direction as defined by SAE J2954), which results from the coil design of the VA.
LFMethod	Object	No	Section 7.3.3: LF.
ProprietaryMethod	Object	No	Sections 7.3.4 and 7.3.5: ProprietaryMethod and ProprietaryData.
VAFinePositioningParameters	Object	No	Section 7.3.6: VAFinePositioningParameters.

7.3.2 AlignStatusCode

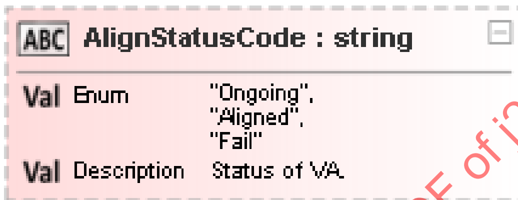


Figure 16 - AlignStatusCode schema diagram

7.3.3 LFMethod

LFMethod is a customizable message container for performing a "low frequency" positioning using one or more transmitters and one or more receivers located on the GA and VA. For additional details on performing LF positioning, refer to SAE J2954.

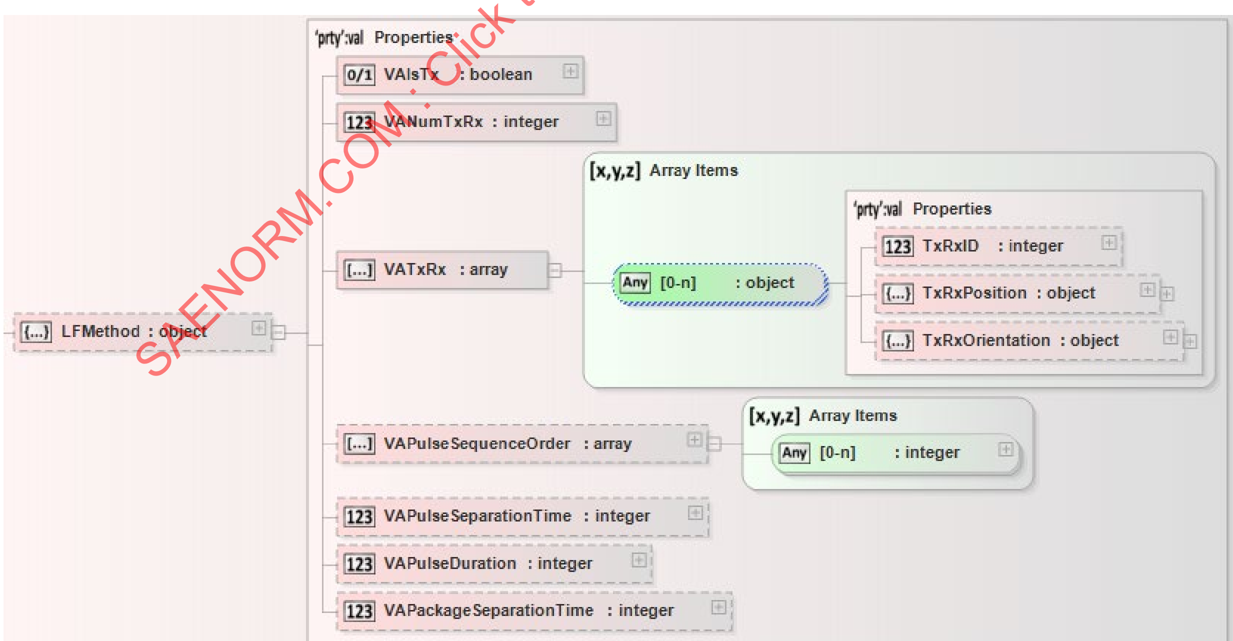


Figure 17 - LFMethod schema diagram

Table 11 - LFMethod properties

Property Name	Type	Required	Description
VAIsTx	Boolean	Yes	“True” if the VA has LF transmitters. “False” if the VA has LF receivers.
VANumTxRx	Integer	Yes	Number of transmitters/receivers on VA. An integer value where value ≥ 0 and value ≤ 255.
VATxRx	Object	Yes	An array describing each of the transmitters/receivers. Must have VANumTxRx number of elements. See 7.3.3.1.
VAPulseSequenceOrder	Integer array	No	An array with 0 to 255 ordered elements containing the transmitter IDs describing the order of signaling. Only valid if VAIsTx is “true.” Each array element is an integer value where value ≥ 0 and value ≤ 255.
VAPulseSeparationTime	Integer	No	Time in ms between individual pulses within the pulse package. An integer value where value ≥ 0 and value ≤ 255.
VAPulseDuration	Integer	No	Time duration in ms of each individual pulse within the pulse package. An integer value where value ≥ 0 and value ≤ 255.
VAPackageSeparationTime	Integer	No	Time in ms between two subsequent pulse packages. An integer value where value ≥ 0 and value ≤ 65535.

7.3.4 VATxRxObject

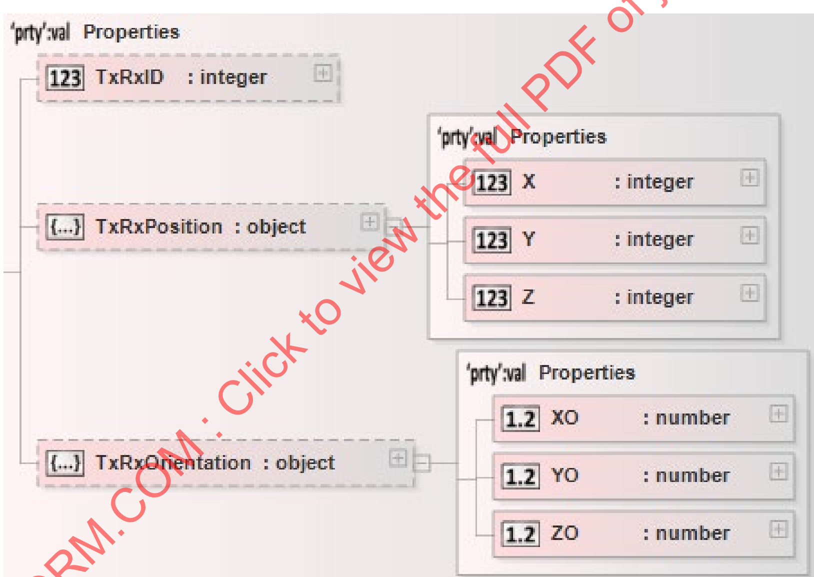


Figure 18 - VATxRx object schema and sub-properties

Table 12 - VATxRx object properties

Property Name	Type	Required	Description
TxRxID	Integer	Yes	ID of the LF transmitter/receiver. An integer value where value ≥ 0 and value ≤ 255.
TxRxPosition	Object of X, Y, Z integers	Yes	X, Y, and Z properties are integer coordinates of the transmitter/receiver relative to the center of the VA coil given in mm. Each is an integer value where value ≥ -32768 and value ≤ 32767.
TxRxOrientation	Object of XO, YO, ZO numbers	Yes	XO, YO, and ZO properties are decimal values which describe a unit vector given the direction of measurement relative to the center of the VA coil. If no direction is applicable, then all three values are set to zero. Each is a decimal where value ≥ -1.0 and value ≤ 1.0 and value mod 0.001 = 0.

7.3.5 ProprietaryMethod

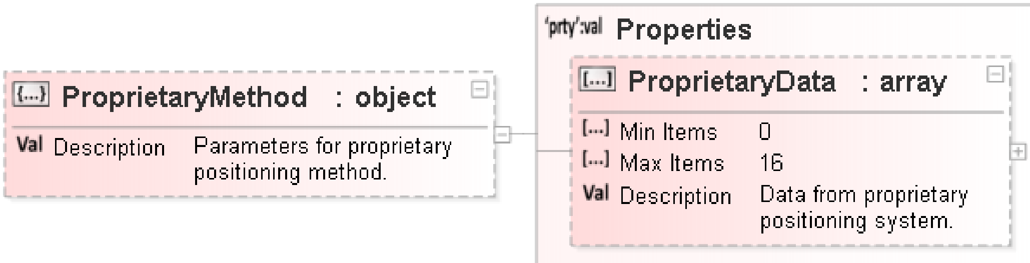


Figure 19 - ProprietaryMethod schema diagram

Table 13 - Proprietary data properties

Property Name	Type	Required	Description
ProprietaryData	Array	No	Section 7.3.5: ProprietaryData.

7.3.6 ProprietaryData

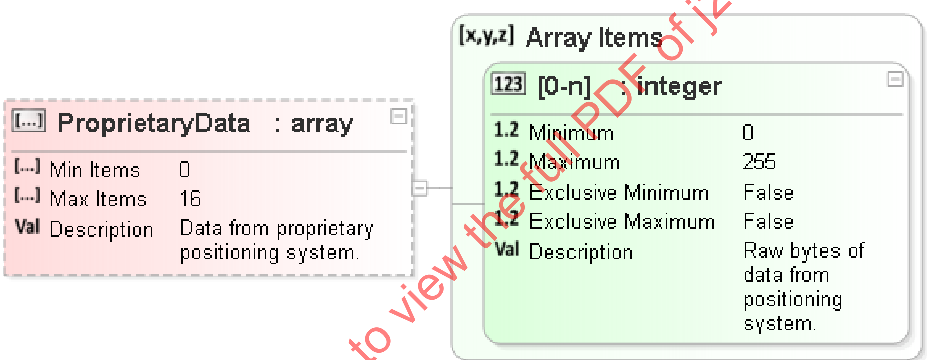


Figure 20 - ProprietaryData schema diagram

7.3.7 VAFinePositioningParameters

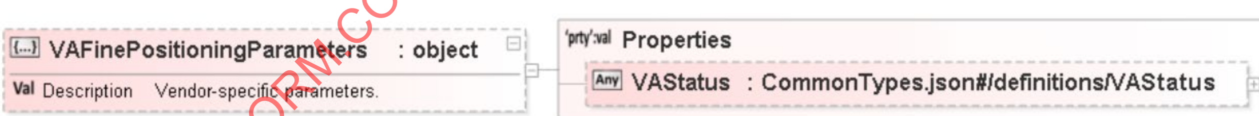


Figure 21 - FinePositioningParameters schema diagram

Table 14 - FinePositioningParameters properties

Property Name	Type	Description
VAStatus	Object	Section 8.4: VAStatus.

7.3.8 Example FinePositioningRequest Message

```
{
    "FinePositioningRequest": {
        "MessageID": 489,
        "AlignStatusCode": "Ongoing",
        "ProprietaryMethod": {
            "ProprietaryData": [
                37,
                55,
                2,
                245,
                128,
                0,
                0,
                37,
                88
            ]
        },
        "VAFinePositioningParameters": {
            "VAStatus": {
                "VAException": "None",
                "VAState": "WPT_V_AA"
            }
        }
    }
```

7.4 FinePositioningResponse

7.4.1 Message Overview

The FinePositionin Response message is sent from GA-CP to VA-CP in response to a FinePositioningRequest message. See 11.2 for message usage. See 13.4 for the JSON schema for this message.

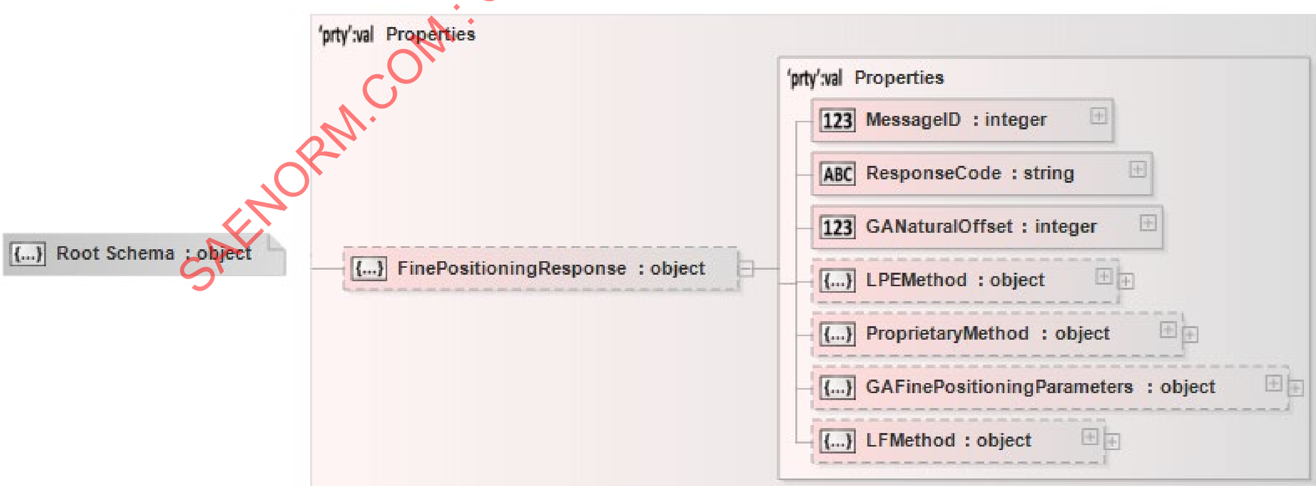


Figure 22 - FinePositioningResponse schema diagram

Table 15 - FinePositioningResponse properties

Property Name	Type	Required	Description
MessageID	Integer	Yes	An integer value where value ≥ 0 and value ≤ 65535. Must be incremented from the last message received.
ResponseCode	String	Yes	Section 7.4.2: ResponseCode.
GANaturalOffset	Integer	Yes	An integer value where value ≥ -32768 and value ≤ 32767. Indicates offset of the center alignment point in mm (in x-direction as defined by SAE J2954), which results from the coil design of the GA.
LPEMethod	Object	No	Section 7.4.3: LPE.
LFMethod	Object	No	Section 7.4.4: LF.
ProprietaryMethod	Object	No	Sections 7.4.5 and 7.4.6 : ProprietaryMethod and ProprietaryData.
GAFinePositioningParameters	Object	No	Section 7.4.7: GAFinePositioningParameters.

7.4.2 ResponseCode

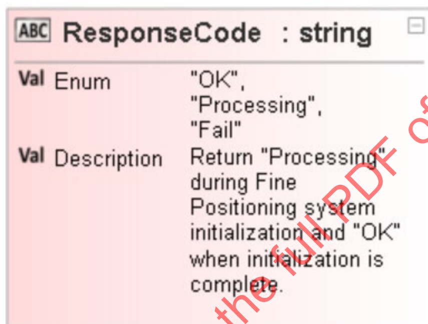


Figure 23 - ResponseCode schema diagram

7.4.3 LPEMethod

LPEMethod is a message container for performing “low power excitation” positioning and alignment check as described in SAE J2954. LPE positioning and alignment check utilizes low current excitation of the GA coil to which is sensed by the VA.

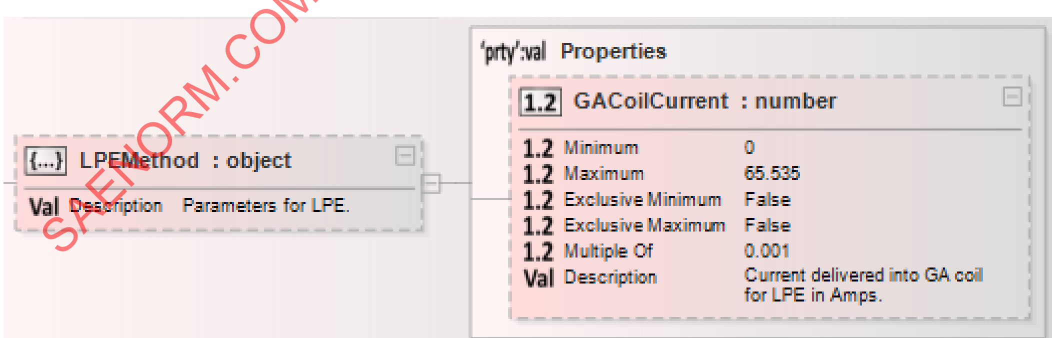


Figure 24 - LPEMethod schema diagram

7.4.4 LFMethod

LFMethod is a customizable message container for performing a “low frequency” positioning using one or more transmitters and one or more receivers located on the GA and VA. For additional details on performing LF positioning, refer to SAE J2954.

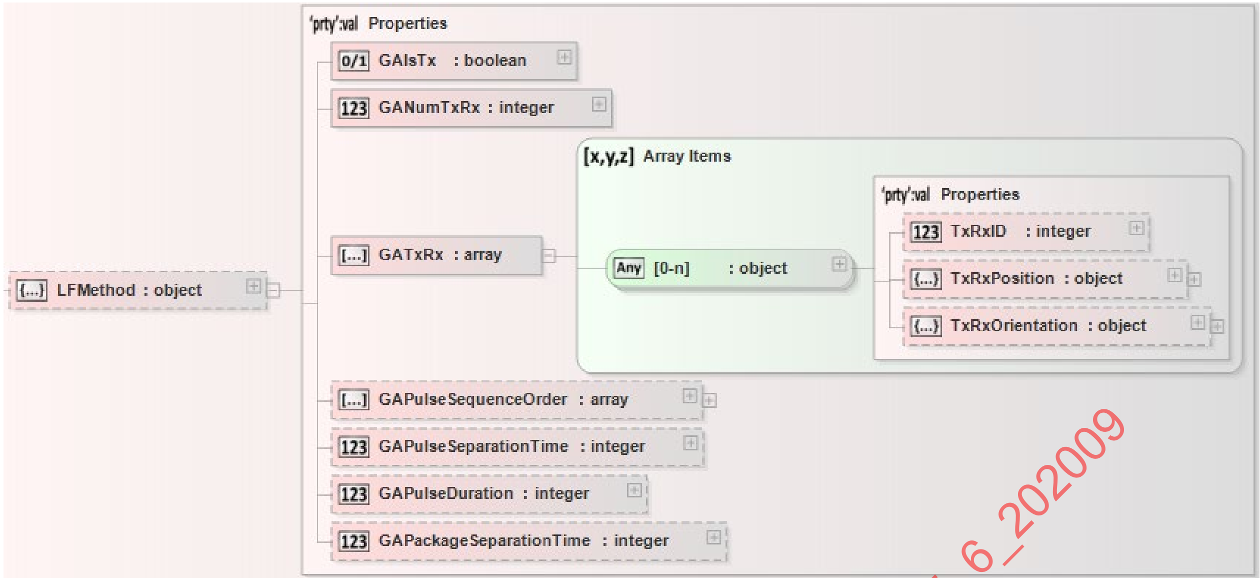


Figure 25 - LFMethod schema diagram

Table 16 - LFMethod properties

Property Name	Type	Required	Description
GAIsTx	Boolean	Yes	"True" if the GA has LF transmitters. "False" if the VA has LF receivers.
GANumTxRx	Integer	Yes	Number of transmitters/receivers on GA. An integer value where value ≥ 0 and value ≤ 255.
GATxRx	Object	Yes	An array describing each of the transmitters/receivers. Must have GANumTxRx number of elements. See 7.4.4.1.
GAPulseSequenceOrder	Integer array	No	An array with 0 to 255 ordered elements containing the transmitter IDs describing the order of signaling. Only valid if GAIsTx is "true." Each array element is an integer value where value ≥ 0 and value ≤ 255.
GAPulseSeparationTime	Integer	No	Time in ms between individual pulses within the pulse package. An integer value where value ≥ 0 and value ≤ 255.
GAPulseDuration	Integer	No	Time duration in ms of each individual pulse within the pulse package. An integer value where value ≥ 0 and value ≤ 255.
GAPackageSeparationTime	Integer	No	Time in ms between two subsequent pulse packages. An integer value where value ≥ 0 and value ≤ 65535.

7.4.5 GATxRxObject

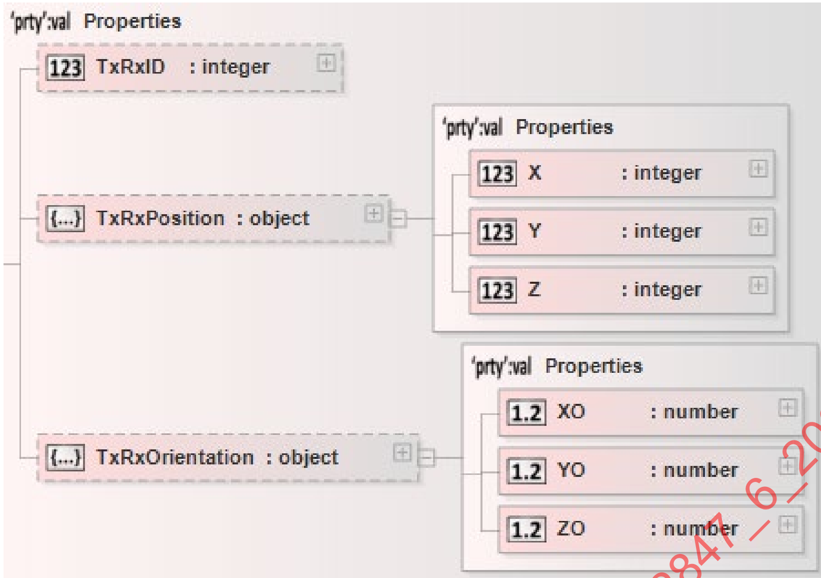


Figure 26 - GATxRxObject schema and sub-properties

Table 17 - GATxRxObject properties

Property Name	Type	Required	Description
TxRxID	Integer	Yes	ID of the LF transmitter/receiver. An integer value where value ≥ 0 and value ≤ 255.
TxRxPosition	Object of X, Y, Z integers	Yes	X, Y, and Z properties are integer coordinates of the transmitter/receiver relative to the center of the GA coil given in mm. Each is an integer value where value ≥ -32768 and value ≤ 32767.
TxRxOrientation	Object of XO, YO, ZO numbers	Yes	XO, YO, and ZO properties are decimal values which describe a unit vector given the direction of measurement relative to the center of the GA coil. If no direction is applicable, then all three values are set to zero. Each is a decimal where value ≥ -1.0 and value ≤ 1.0 and value mod 0.001 = 0.

7.4.6 ProprietaryMethod

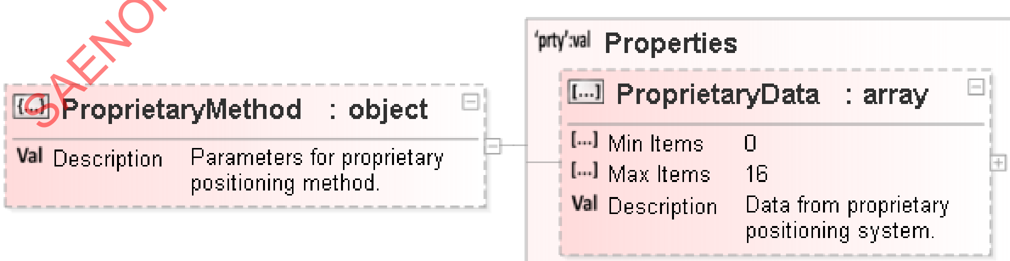


Figure 27 - ProprietaryMethod schema diagram

Table 18 - ProprietaryMethod properties

Property Name	Type	Required	Description
ProprietaryData	Array	No	Section 7.4.6: ProprietaryData.

7.4.7 ProprietaryData

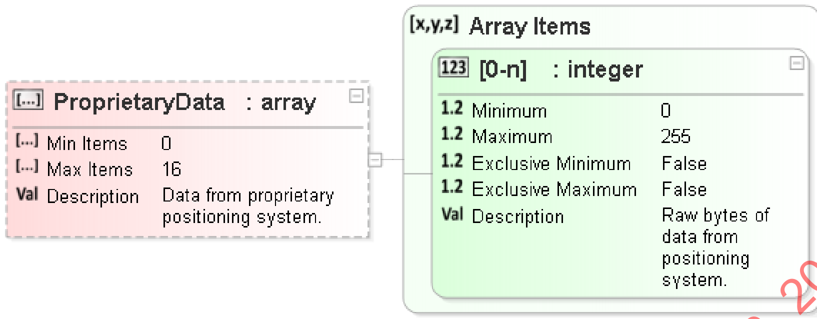


Figure 28 - ProprietaryData schema diagram

7.4.8 GAFinePositioningParameters

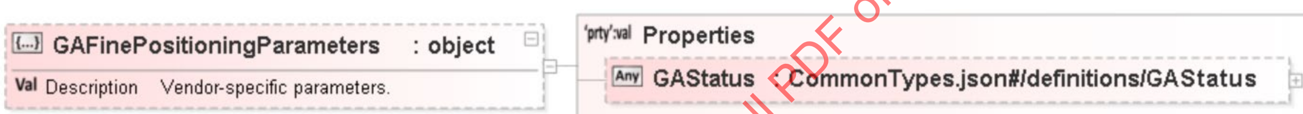


Figure 29 - FinePositioningParameters schema diagram

Table 19 - FinePositioningParameters properties

Property Name	Type	Description
GAStatus	Object	Section 8.5: GAStatus.

7.4.9 Example FinePositioningResponse Message


```
{
    "FinePositioningResponse": {
        "MessageID": 916,
        "ResponseCode": "OK",
        "ProprietaryMethod": {
            "ProprietaryData": [
                1,
                200,
                133,
                45,
                73,
                89,
                3
            ]
        },
        "GAFinePositioningParameters": {
            "GAStatus": {
                "GAException": "None",
                "GAState": "WPT_S_AA"
            }
        }
    }
```

7.5 PowerRequest

7.5.1 Message Overview

The PowerRequest message is sent from VA-CP to GA-CP to initiate and control power transfer. See 11.1 for message usage. See 13.5 for the JSON schema for this message.

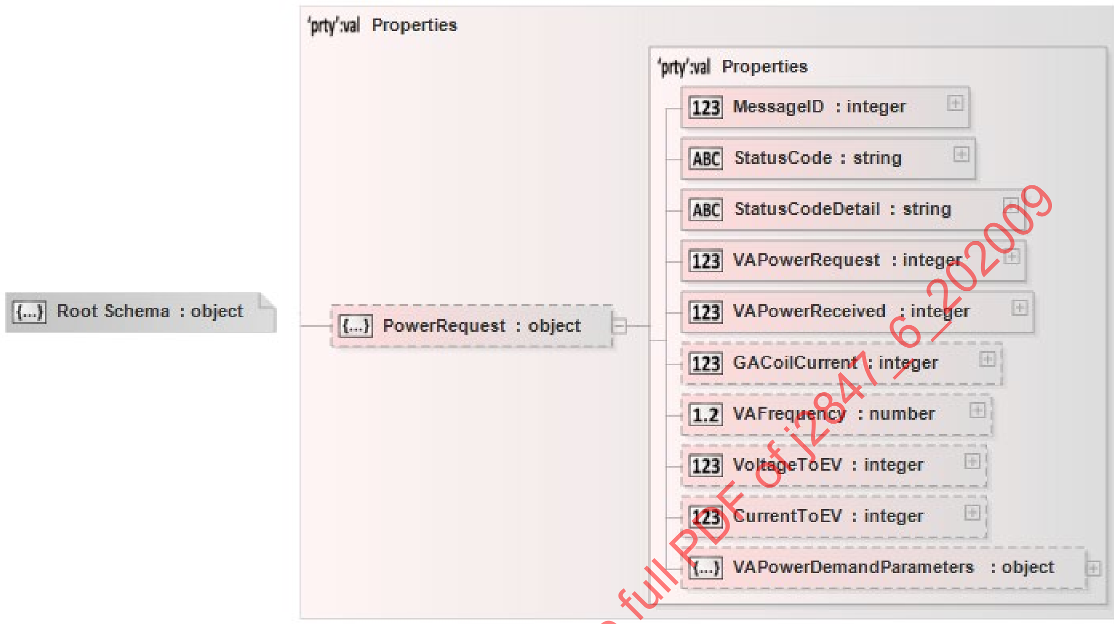


Figure 30 - PowerRequest schema diagram

Table 20 - PowerRequest properties

Property Name	Type	Required	Description
MessageID	Integer	Yes	An integer value where value ≥ 0 and value ≤ 65535. Must be incremented from the last message received.
StatusCode	String	Yes	Section 7.5.2: StatusCode.
StatusCodeDetail	String	Yes	Section 7.5.3: StatusCodeDetail.
VAPowerRequest	Integer	Yes	An integer value where value ≥ 0 and value ≤ 22000. Power the VA is requesting in watts, as measured at the DC output of the rectifier.
VAPowerReceived	Integer	Yes	An integer value where value ≥ 0 and value ≤ 32767. Power received by the VA as measured at the output of the VA electronics in watts.
GACoilCurrent	Integer	No	An integer value where value ≥ 0 and value ≤ 127. Number of amps the GA should be sending into the GA coil.
VAFrequency	Number	No	A number value where value ≥ 79 and value ≤ 90 and value mod 0.001 = 0. Frequency the VA wants the GA to use for power transfer in kHz (e.g., 85.5).
VoltageToEV	Integer	No	An integer value where value ≥ 0 and value ≤ 65535. Voltage measured at the output of the VA electronics in volts.
CurrentToEV	Integer	No	An integer value where value ≥ 0 and value ≤ 32767. Current measured at the output of the VA electronics in amps.
VAPowerDemandParameters	Object	No	Section 7.5.4: VAPowerDemandParameters.

7.5.2 StatusCode

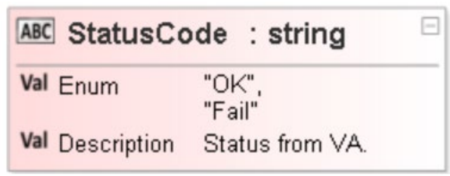


Figure 31 - StatusCode schema diagram

7.5.3 StatusCodeDetail

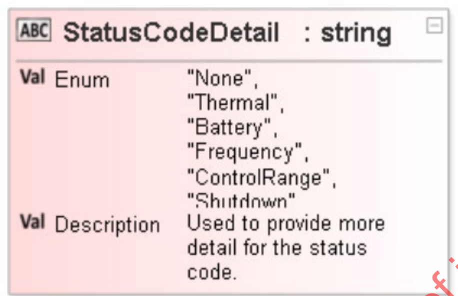


Figure 32 - StatusCodeDetail schema diagram

7.5.4 VAPowerDemandParameters

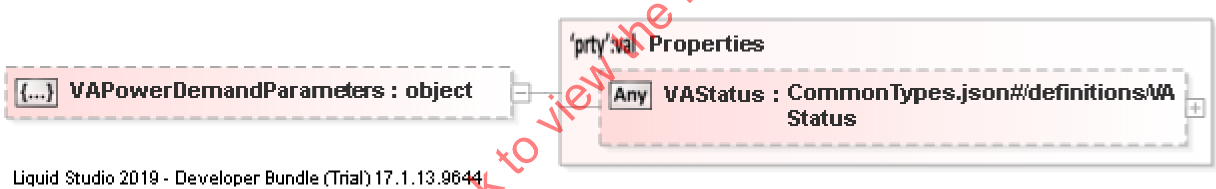


Figure 33 - PowerDemandParameters schema diagram

Table 21 - PowerDemandParameters properties

Property Name	Type	Description
VAStatus	Object	Section 8.4: VAStatus.

7.5.5 Example PowerRequest Message

```
{
    "PowerRequest": {
        "MessageID": 155,
        "StatusCode": "OK",
        "StatusCodeDetail": "None",
        "VAPowerRequest": 10000,
        "VAPowerReceived": 9990,
        "VAPowerDemandParameters": {
            "VAStatus": {
                "VAException": "None",
                "VAState": "WPT_V_PT"
            }
        }
    }
}
```

7.6 PowerResponse

7.6.1 Message Overview

The PowerResponse message is sent from GA-CP to VA-CP in response to a PowerRequest message. See 11.2 for message usage. See 13.6 for the JSON schema for this message.

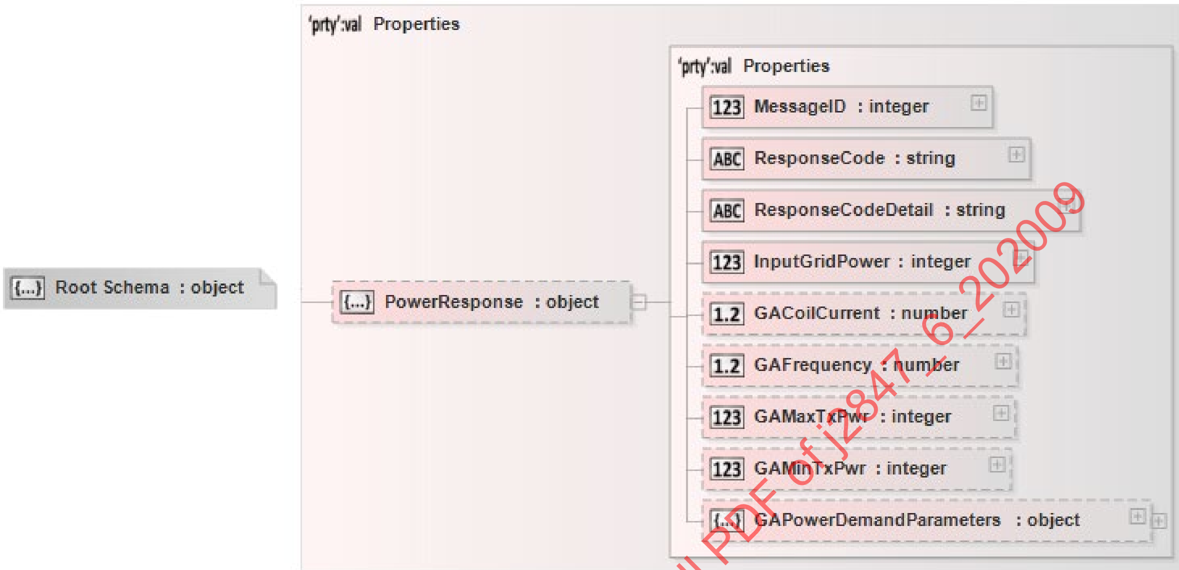


Figure 34 - PowerResponse schema diagram

Table 22 - PowerResponse properties

Property Name	Type	Required	Description
MessageID	Integer	Yes	An integer value where value ≥ 0 and value ≤ 65535. Must be incremented from the last message received.
ResponseCode	String	Yes	Section 7.6.2: ResponseCode.
ResponseCodeDetail	String	Yes	Section 7.6.3: ResponseCodeDetail.
InputGridPower	Integer	Yes	An integer value where value ≥ 0 and value ≤ 32767. Power the GA is consuming from the grid in watts.
VAPowerRequest	Integer	Yes	An integer value where value ≥ 0 and value ≤ 22000. Confirmation of power the VA is requesting in watts.
GACoilCurrent	Number	No	A number value where value ≥ 0 and value ≤ 3276.7 and value mod 0.1 = 0. Number of amps the GA is sending into the GA coil.
GAFrequency	Number	No	A number value where value ≥ 79 and value ≤ 90 and value mod 0.001 = 0. Frequency the GA is using for power transfer in kHz (e.g., 85.5).
GAMaxTxPwr	Integer	No	An integer value where value ≥ 0 and value ≤ 22000. Maximum number of watts that the GA is willing to transmit.
GAMinTxPwr	Integer	No	An integer value where value ≥ 0 and value ≤ 22000. Minimum number of watts that the GA is willing to transmit.
GAPowerDemandParameters	Object	No	Section 7.6.4: GAPowerDemandParameters.

7.6.2 ResponseCode

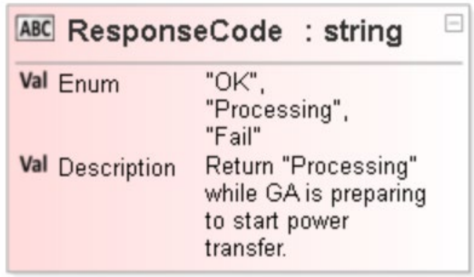


Figure 35 - ResponseCode schema diagram

7.6.3 ResponseCodeDetail

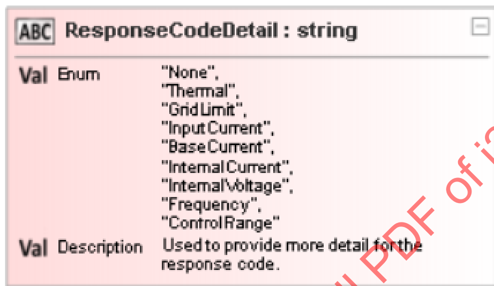


Figure 36 - ResponseCodeDetail schema diagram

7.6.4 GAPowerDemandParameters

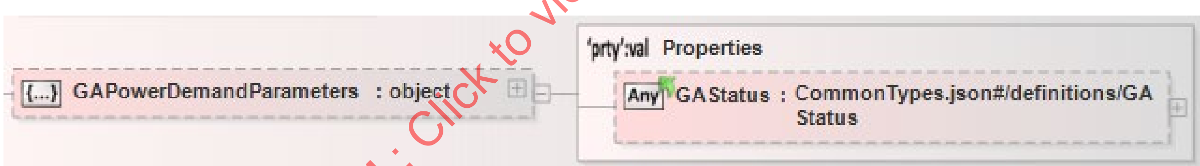


Figure 37 - PowerDemandParameters schema diagram

Table 23 - PowerDemandParameters properties

Property Name	Type	Description
GAStatus	Object	Section 8.5: GAStatus.

7.6.5 Example PowerResponse Message


```
{
    "PowerResponse": {
        "MessageID": 631,
        "ResponseCode": "OK",
        "ResponseCodeDetail": "None",
        "InputGridPower": 8000,
        "GAPowerDemandParameters": {
            "GAStatus": {
                "GAException": "None",
                "GAState": "WPT_S_PT"
            }
        }
    }
}
```


7.7 TerminatePowerRequest

7.7.1 Message Overview

The TerminatePowerRequest message is sent from VA-CP to GA-CP to initiate termination of power transfer. See 11.1 for message usage. See 13.7 for the JSON schema for this message.

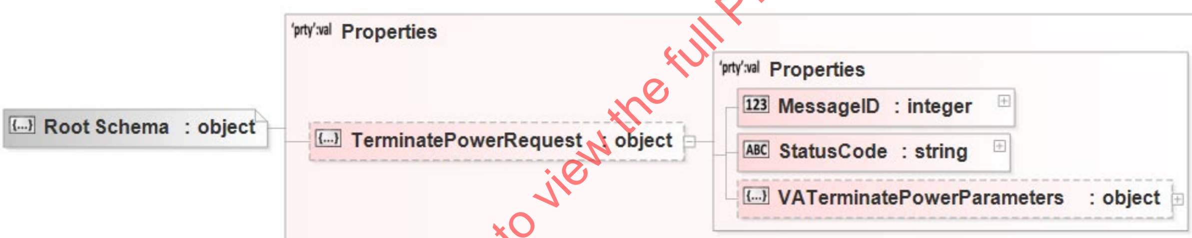


Figure 38 - TerminatePowerRequest schema diagram

Table 24 - TerminatePowerRequest properties

Property Name	Type	Required	Description
MessageID	Integer	Yes	An integer value where value ≥ 0 and value ≤ 65535. Must be incremented from the last message received.
StatusCode	String	Yes	Section 7.7.2: StatusCode.
VATerminatePowerParameters	Object	No	Section 7.7.3: VATerminatePowerParameters.

7.7.2 StatusCode

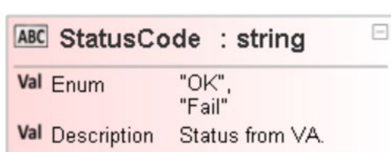


Figure 39 - StatusCode schema diagram

7.7.3 VATerminatePowerParameters

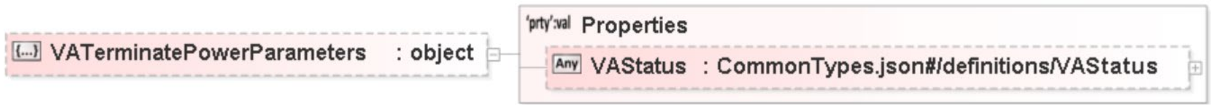


Figure 40 - TerminatePowerParameters schema diagram

Table 25 - TerminatePowerParameters properties

Property Name	Type	Description
VAStatus	Object	Section 8.4: VAStatus.

7.7.4 Example TerminatePowerRequest Message

```
{
    "TerminatePowerRequest": {
        "MessageID": 480,
        "StatusCode": "OK",
        "VATerminatePowerParameters": {
            "VAStatus": {
                "VAException": "None",
                "VAState": "WPT_V_PT"
            }
        }
    }
}
```

7.8 TerminatePowerResponse

7.8.1 Message Overview

The TerminatePowerResponse message is sent from GA-CP to VA-CP in response to a TerminatePowerRequest message. See 11.2 for message usage. See 13.8 for the JSON schema for this message.

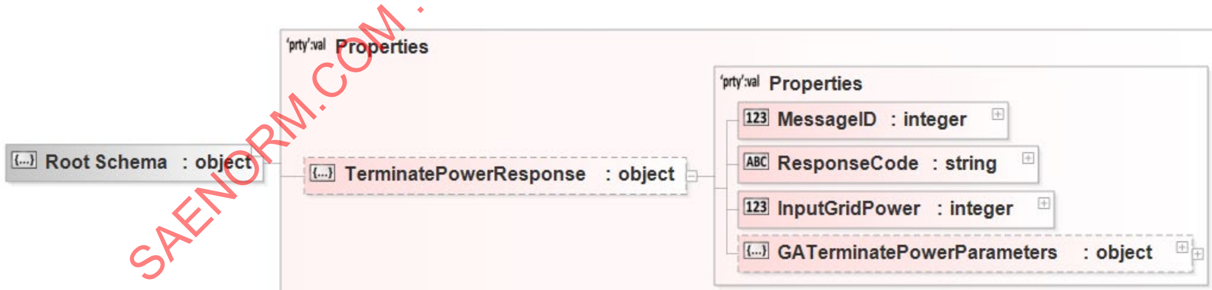


Figure 41 - TerminatePowerResponse schema diagram

Table 26 - TerminatePowerResponse properties

Property Name	Type	Required	Description
MessageID	Integer	Yes	An integer value where value ≥ 0 and value ≤ 65535. Must be incremented from the last message received.
ResponseCode	String	Yes	Section 7.8.2: ResponseCode.
InputGridPower	Integer	Yes	An Integer value where value ≥ 0 and value ≤ 32767. Power the GA is consuming from the grid in watts.
GATerminatePowerParameters	Object	No	Section 7.8.3: GATerminatePowerParameters.

7.8.2 ResponseCode

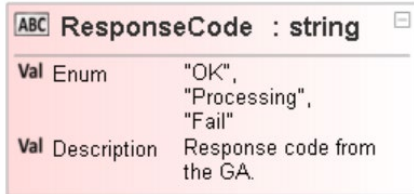


Figure 42 - ResponseCode schema diagram

7.8.3 GATerminatePowerParameters

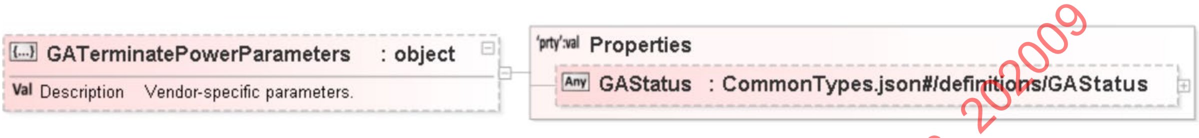


Figure 43 - GATerminatePowerParameters schema diagram

Table 27 - TerminatePowerParameters properties

Property Name	Type	Description
GAStatus	Object	Section 8.5: GAStatus.

7.8.4 Example TerminatePowerResponse Message

```
{
    "TerminatePowerResponse": {
        "MessageID": 971,
        "ResponseCode": "Processing",
  "InputGridPower": 3000,
        "GATerminatePowerParameters": {
            "GAStatus": {
                "GAException": "None",
                "GAState": "WPT_S_PT"
            }
        }
    }
}
```

7.9 StatusExchangeRequest

7.9.1 Message Overview

The StatusExchangeRequest message is sent from VA-CP to GA-CP to initiate the exchange of status when errors are detected by the system. See 11.1 for message usage. See 13.9 for the JSON schema for this message.

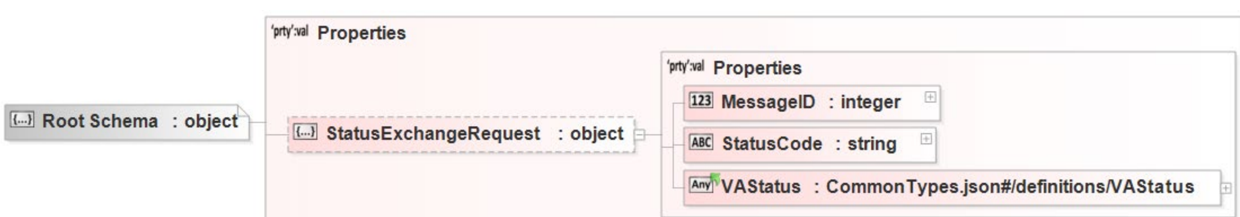


Figure 44 - StatusExchangeRequest

Table 28 - StatusExchangeRequest properties

Property Name	Type	Required	Description
MessageID	Integer	Yes	An integer value where value ≥ 0 and value ≤ 65535. Must be incremented from the last message received.
StatusCode	String	Yes	Section 7.9.2: StatusCode.
VAStatus	Object	Yes	Section 8.4: VAStatus.

7.9.2 StatusCode

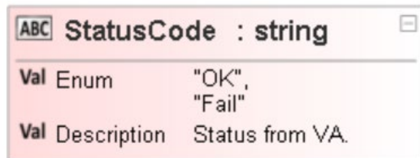


Figure 45 - StatusCode schema diagram

7.9.3 Example StatusExchangeRequest Message

```
{
    "StatusExchangeRequest": {
        "MessageID": 496,
        "StatusCode": "OK",
        "VAStatus": {
            "VAException": "SystemErrorInAA",
            "VAState": "WPT_V_ERR"
        }
    }
}
```

7.10 StatusExchangeResponse

7.10.1 Message Overview

The StatusExchangeResponse message is sent from GA-CP to VA-CP in response to a StatusExchangeRequest message. See 11.2 for message usage. See 13.10 for the JSON schema for this message.

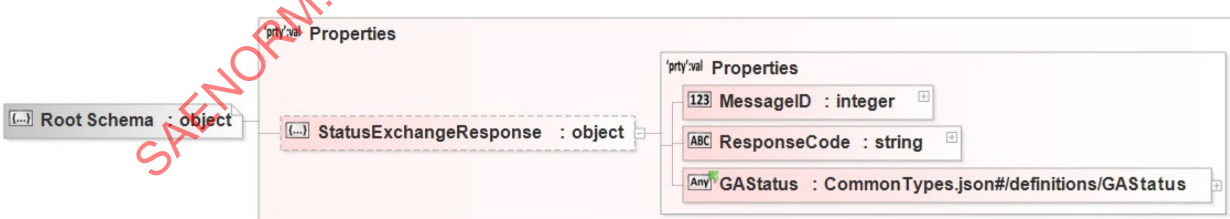


Figure 46 - StatusExchangeResponse schema diagram

Table 29 - StatusExchangeResponse properties

Property Name	Type	Required	Description
MessageID	Integer	Yes	An integer value where value ≥ 0 and value ≤ 65535. Must be incremented from the last message received.
ResponseCode	String	Yes	Section 7.10.2: ResponseCode.
GAStatus	Object	Yes	Section 8.5: GAStatus.

7.10.2 ResponseCode

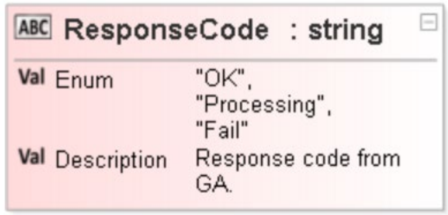


Figure 47 - ResponseCode schema diagram

7.10.3 Example StatusExchangeResponse Message

```
{
    "StatusExchangeResponse": {
        "MessageID": 915,
        "ResponseCode": "OK",
        "GAStatus": {
            "GAException": "SystemErrorInIdleOrPT",
            "GAState": "WPT_S_ERR"
        }
    }
}
```


7.11 TerminateCommunicationsRequest

7.11.1 Message Overview

The TerminateCommunicationsRequest message is sent from VA-CP to GA-CP to initiate termination of wireless communications between the VA-CP and GA-CP. See 11.1 for message usage. See 13.11 for the JSON schema for this message.

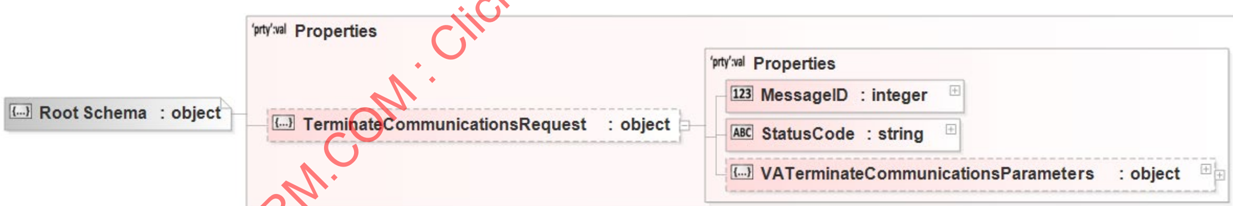


Figure 48 - TerminateCommunicationsRequest schema diagram

Table 30 - TerminateCommunicationsRequest properties

Property Name	Type	Required	Description
MessageID	Integer	Yes	An integer value where value ≥ 0 and value ≤ 65535. Required. Must be incremented from the last message received.
StatusCode	String	Yes	Section 7.11.2: StatusCode.
VATerminateCommunicationsParameters	Object	No	Section 7.11.3: VATerminateCommunicationsParameters.

7.11.2 StatusCode

Figure 49 - StatusCode schema diagram

7.11.3 VATerminateCommunicationsParameters

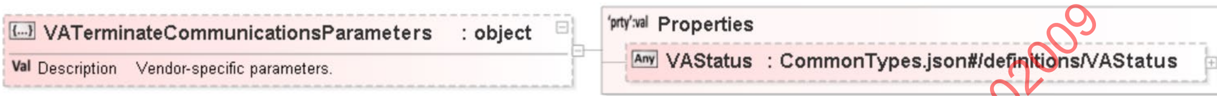


Figure 50 - TerminateCommunicationsParameters schema diagram

Table 31 - TerminateCommunicationsParameters properties

Property Name	Type	Description
VAStatus	Object	Section 8.4: VAStatus.

Example TerminateCommunicationRequestMessage:

```
{
    "TerminateCommunicationsRequest": {
        "MessageID": 560,
        "StatusCode": "OK",
        "VATerminateCommunicationsParameters": {
            "VAStatus": {
                "VAException": "None",
                "VAState": "WPT_V_IDLE"
            }
        }
    }
}
```

7.12 TerminateCommunicationsResponse

7.12.1 Message Overview

The TerminateCommunicationsResponse message is sent from GA-CP to VA-CP in response to a TerminateCommunications Request message. See 11.2 for message usage. See 13.12 for the JSON schema for this message.

Figure 51 - TerminateCommunicationsResponse schema diagram

Table 32 - TerminateCommunicationsResponse properties

Property Name	Type	Required	Description
MessageID	Integer	Yes	An integer value where value ≥ 0 and value ≤ 65535. Must be incremented from the last message received.
ResponseCode	String	Yes	Section 7.12.2: ResponseCode.
GATerminateCommunicationsParameters	Object	No	Section 7.12.3: GATerminateCommunicationsParameters.

7.12.2 ResponseCode

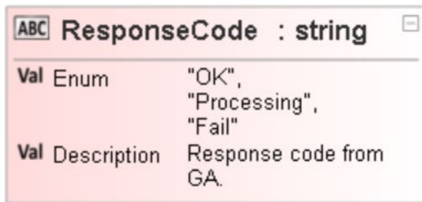


Figure 52 - ResponseCode schema diagram

7.12.3 GATerminateCommunicationsParameters

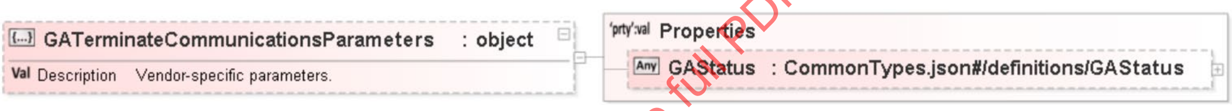


Figure 53 - TerminateCommunicationsParameters schema diagram

Table 33 - TerminateCommunicationsParameters properties

Property Name	Type	Description
GAStatus	Object	Section 8.5: GAStatus.

7.12.4 Example TerminateCommunicationResponse Message

```
{
    "TerminateCommunicationsResponse": {
        "MessageID": 271,
        "ResponseCode": "Processing",
        "GATerminateCommunicationsParameters": {
            "GAStatus": {
                "GAException": "None",
                "GAState": "WPT_S_IDLE"
            }
        }
    }
}
```

8. COMMON TYPES

8.1 Overview of Common Types

Common types include schema fragments that are shared across multiple other schema. Defining them separately promotes schema reuse through the use of JSON references.

8.2 ProtocolVersion

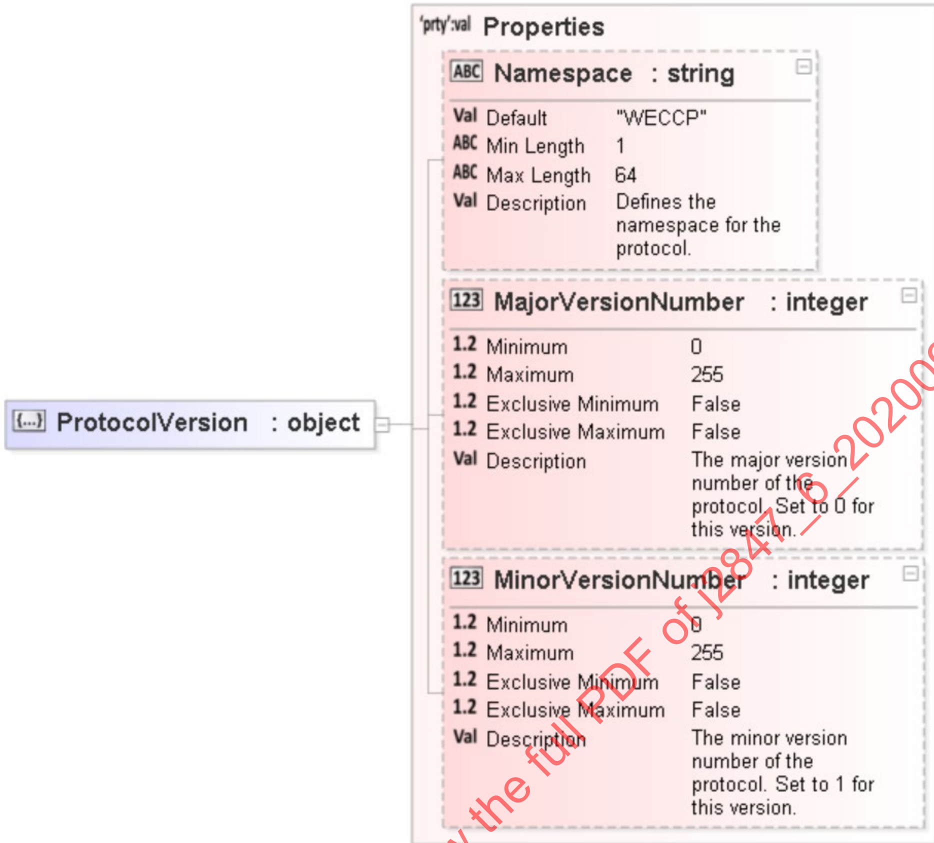


Figure 54 - ProtocolVersion schema diagram

8.3 FinePositioningMethods

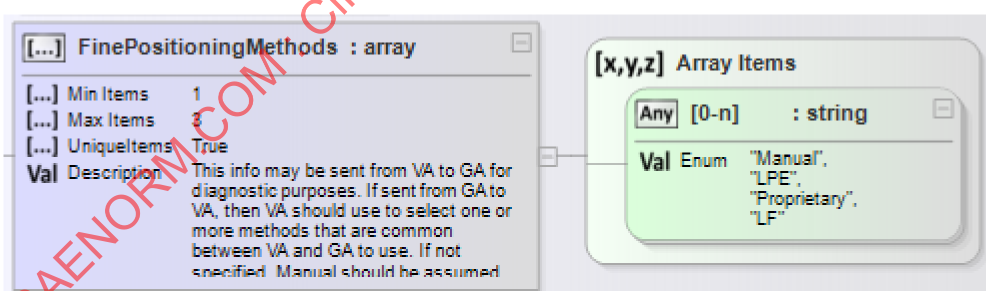


Figure 55 - FinePositioningMethods schema diagram

8.4 VAStatus

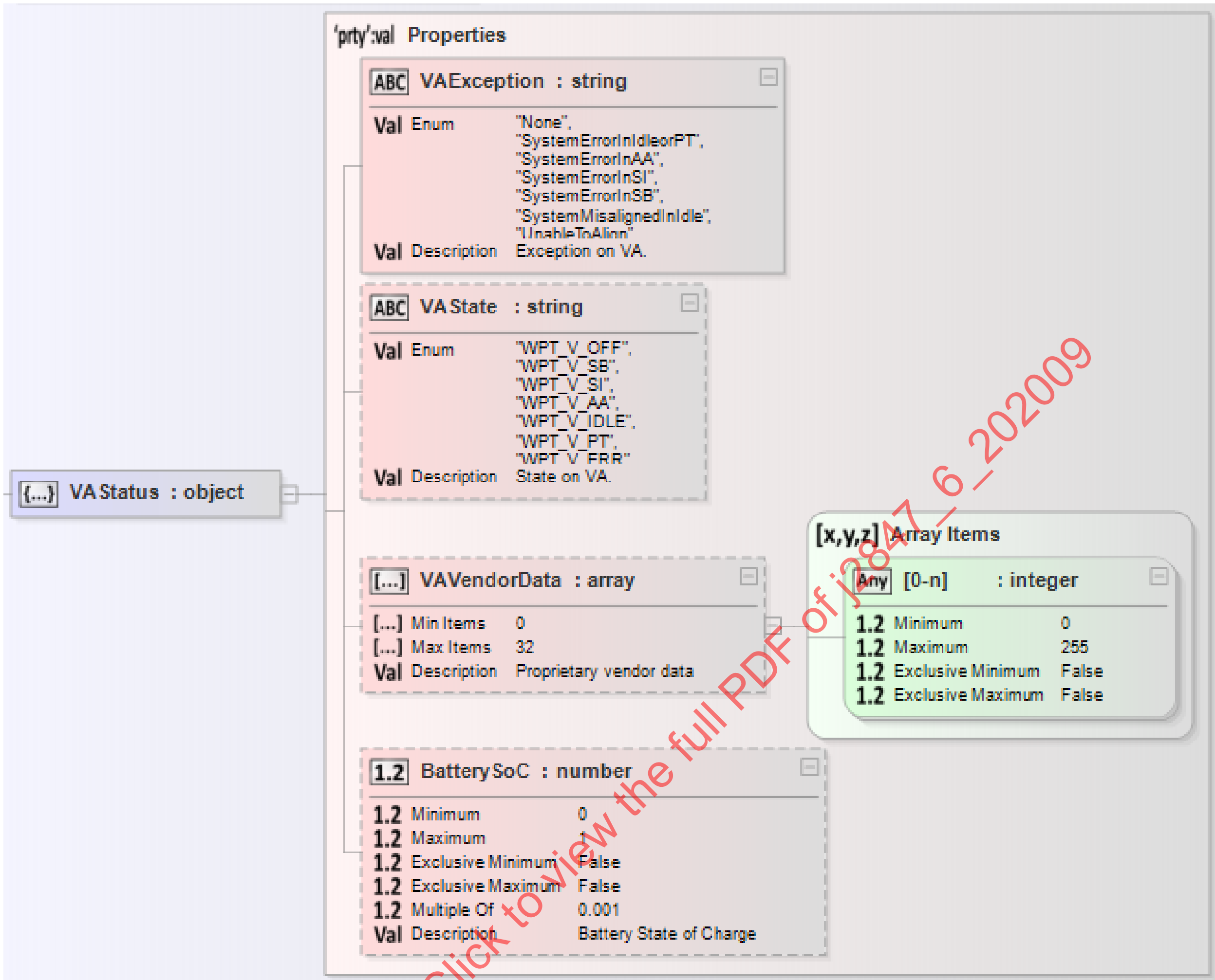


Figure 56 - VAStatus schema diagram

8.5 GAStatus

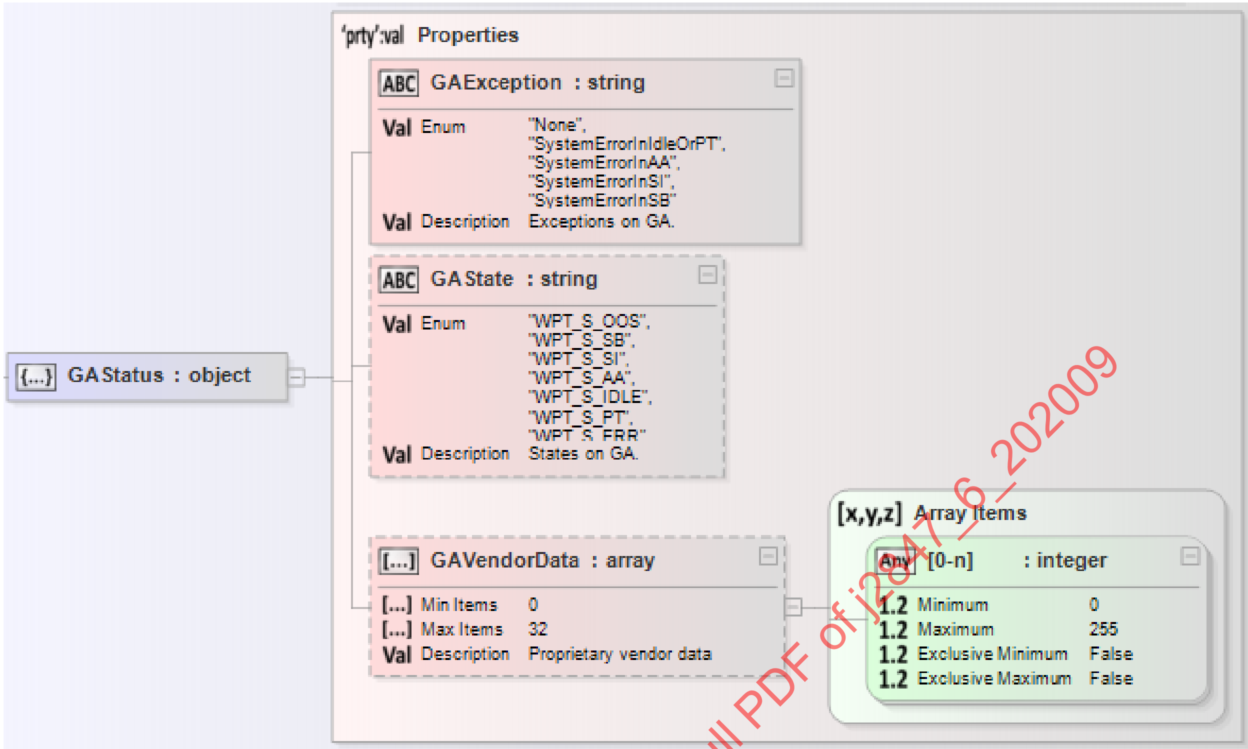
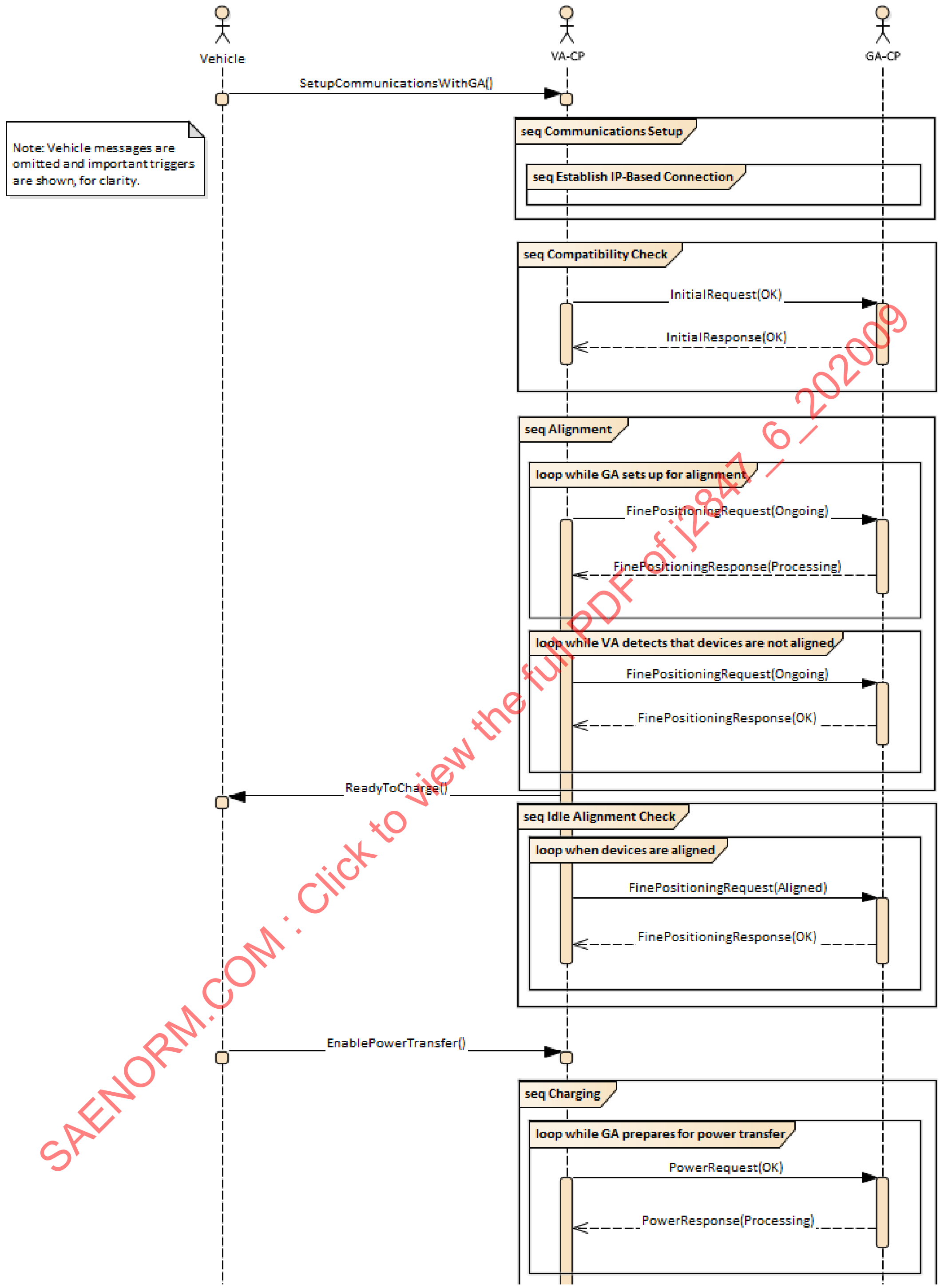


Figure 57 - GAStatus schema diagram

9. EXAMPLE MESSAGE SEQUENCES

The message sequence diagrams illustrate various example scenarios to show representative interactions between VA-CP and GA-CP during the various message cycles. They are not an exhaustive specification of all scenarios, nor necessarily a characterization of worst case scenarios. For greater detail on how each GA-CP and VA-CP should behave across many conditions, see Section 11.

9.1 Alignment and Charging



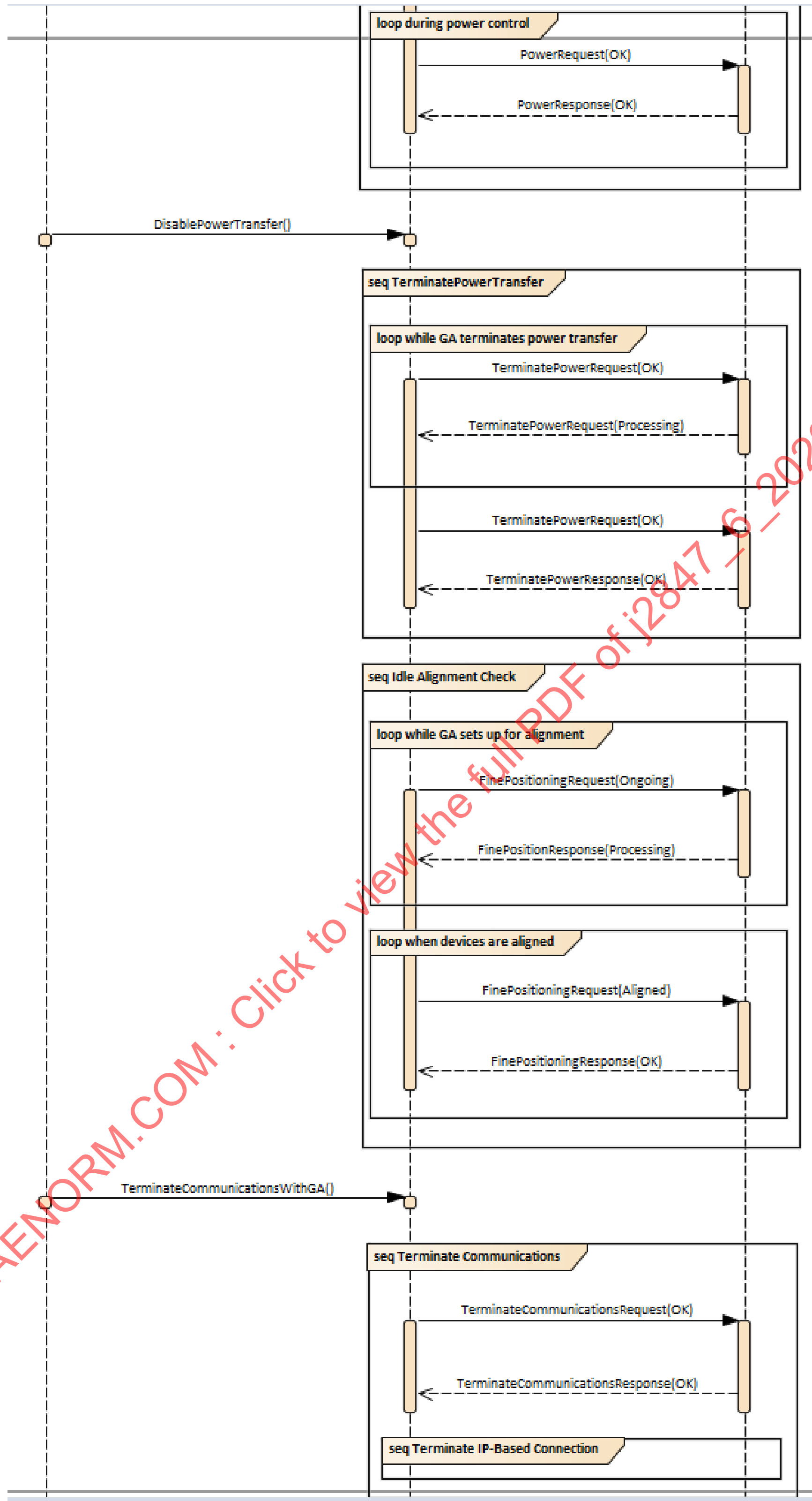
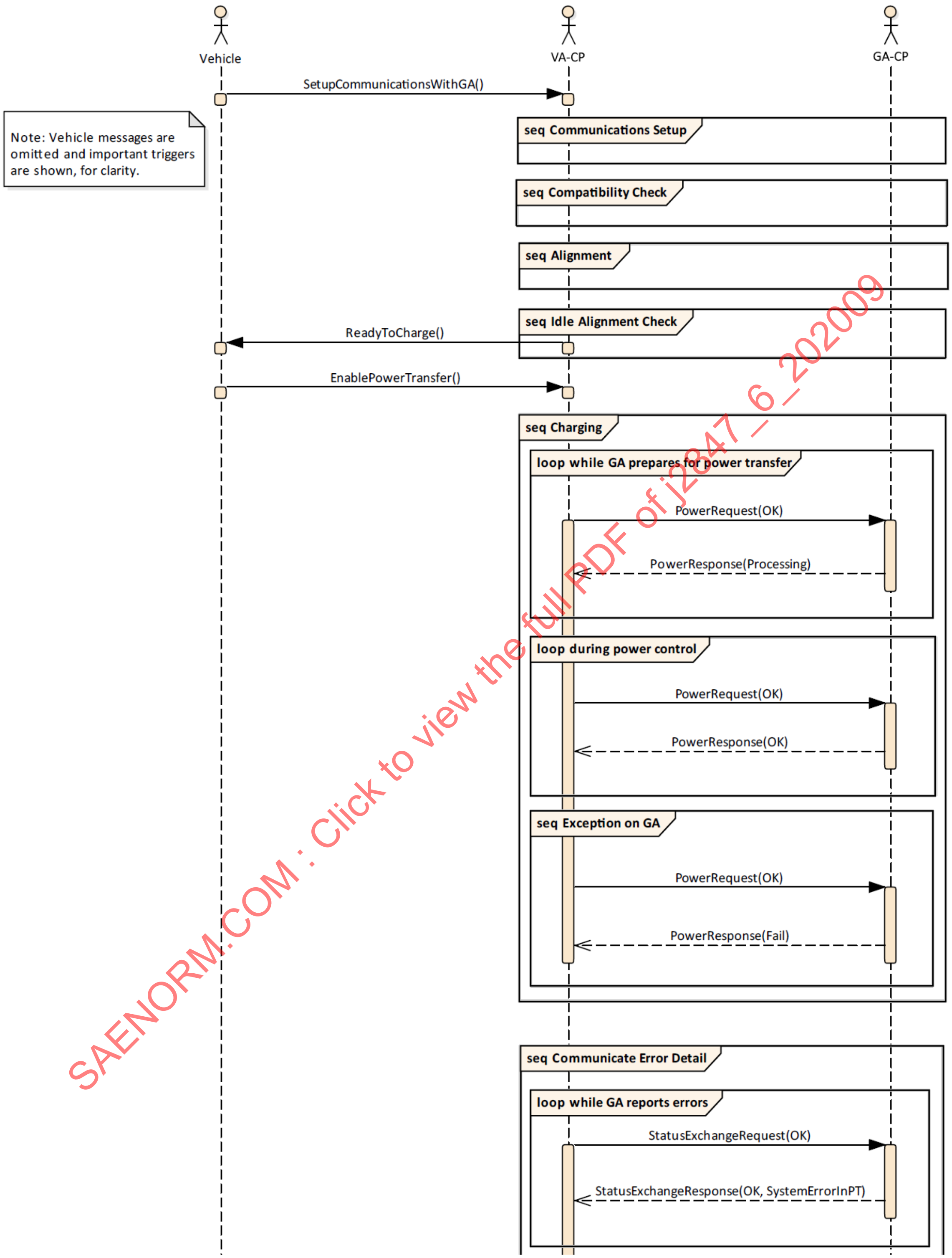


Figure 58 - Alignment and charging sequence diagram

9.2 GA Exception During Power Transfer



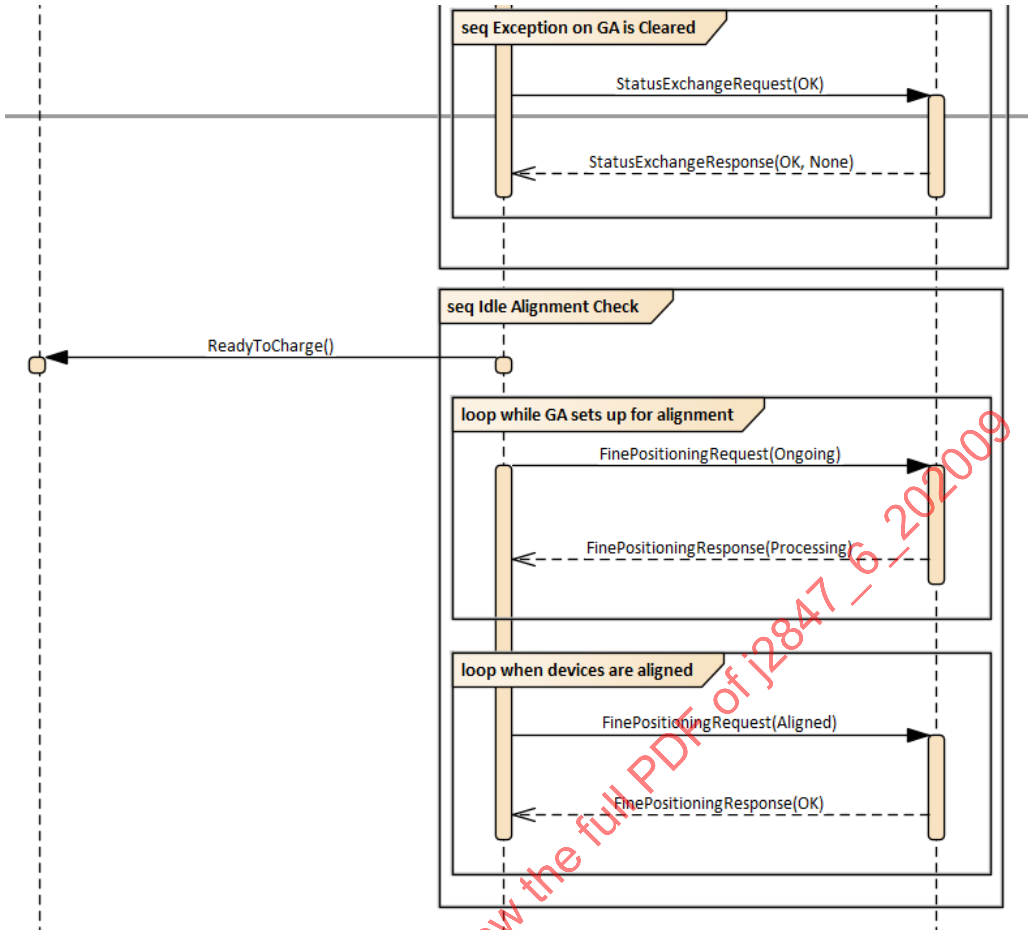
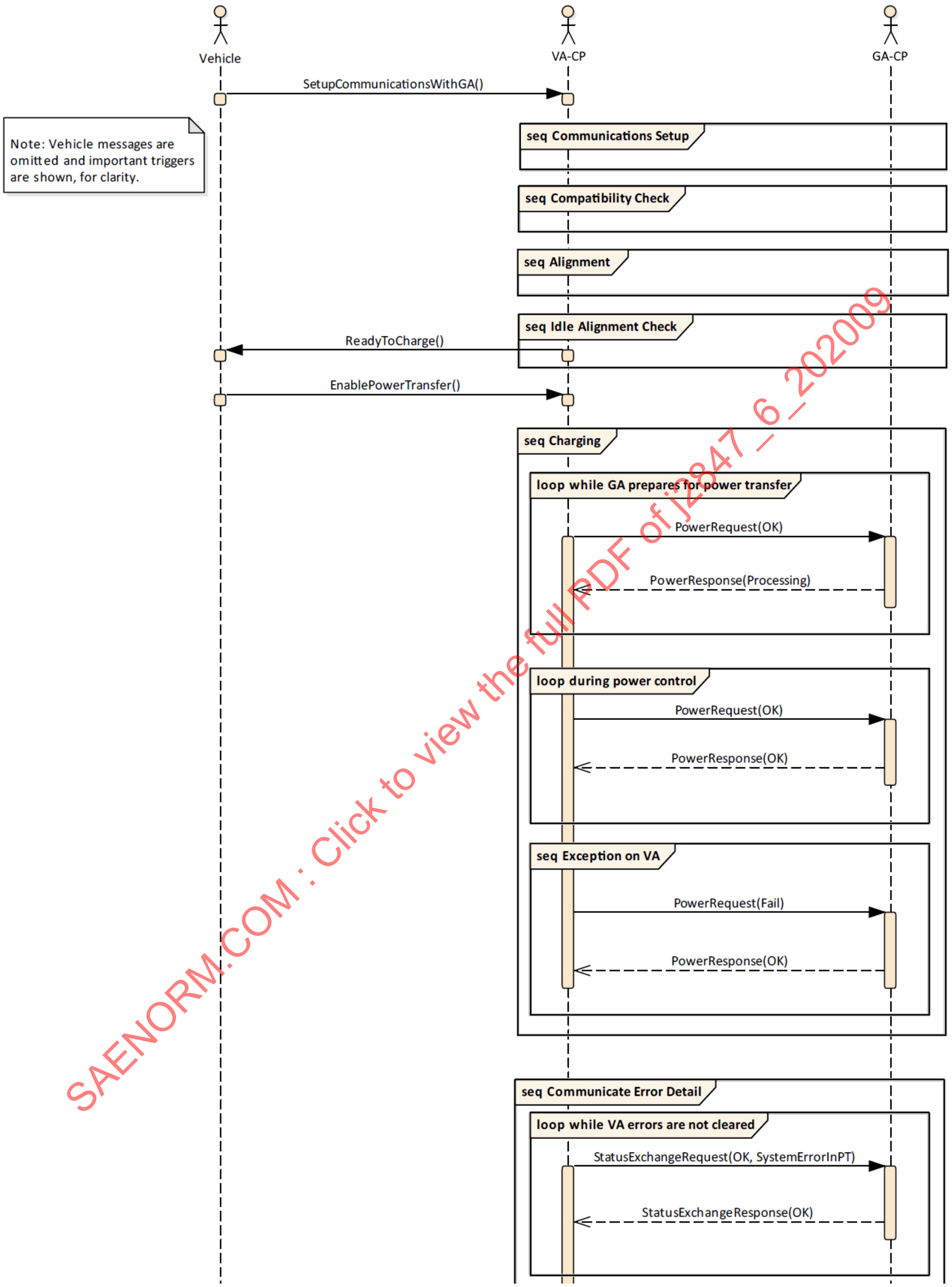


Figure 59 - GA exception during power transfer sequence diagram

9.3 VA Exception During Power Transfer



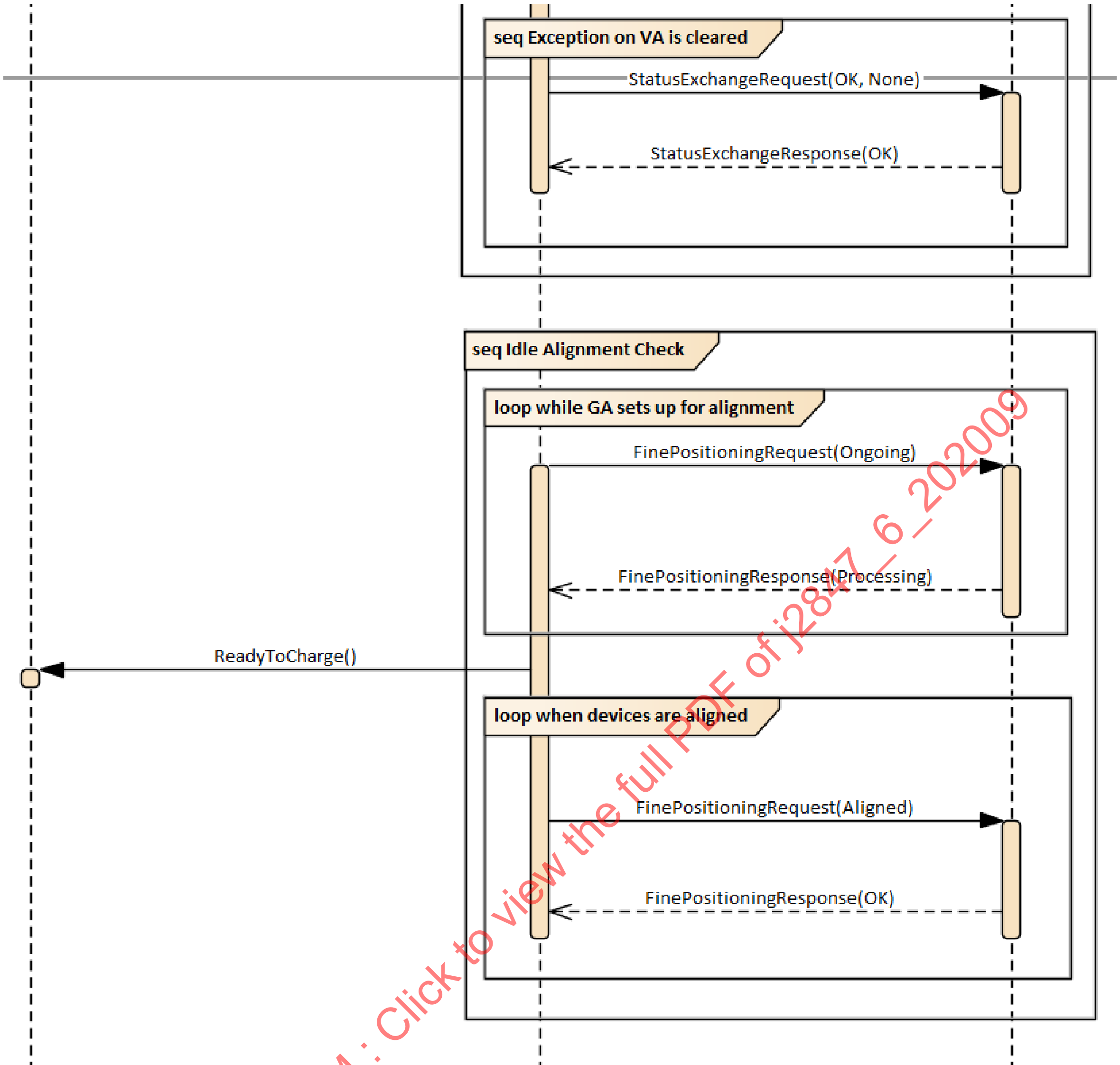
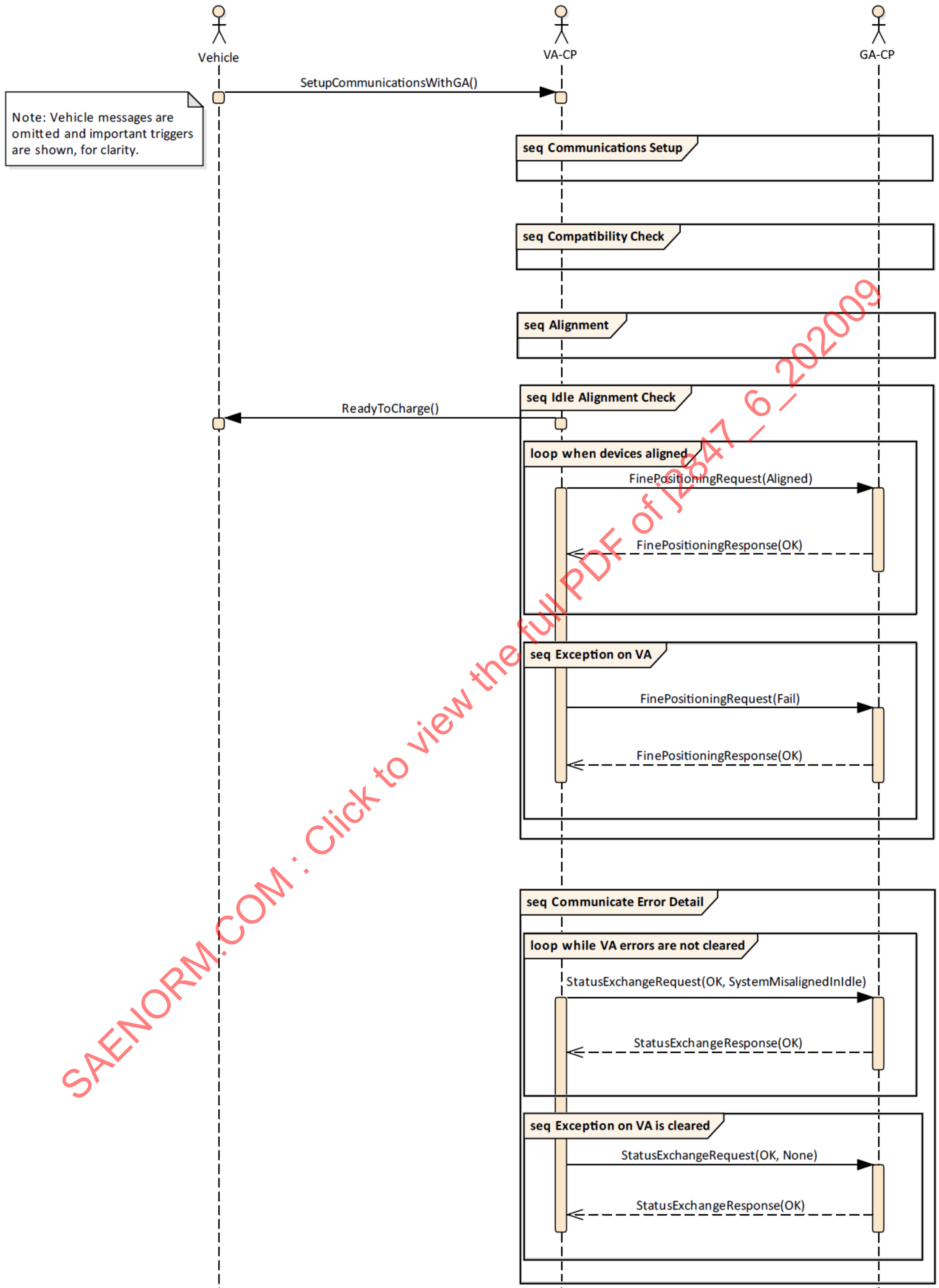


Figure 60 - VA exception during power transfer sequence diagram

9.4 GA and VA Become Misaligned



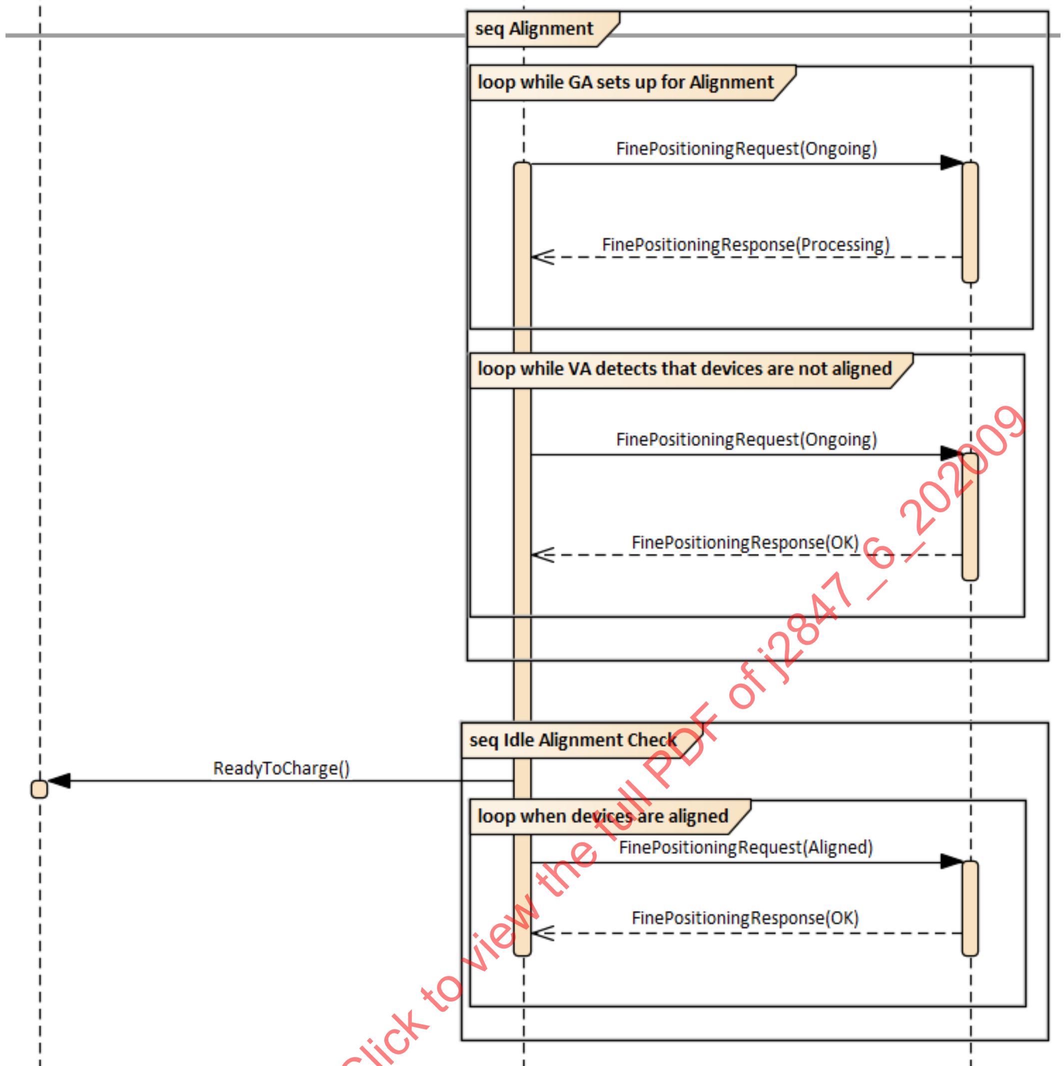


Figure 61 - GA and VA become misaligned sequence diagram

10. MESSAGE TIMING

The timing specifications attempt to define the timers of importance and specify where timeouts, performance times, and execution periods should be implemented. They also provide default values for timeouts, performance times, and execution periods.

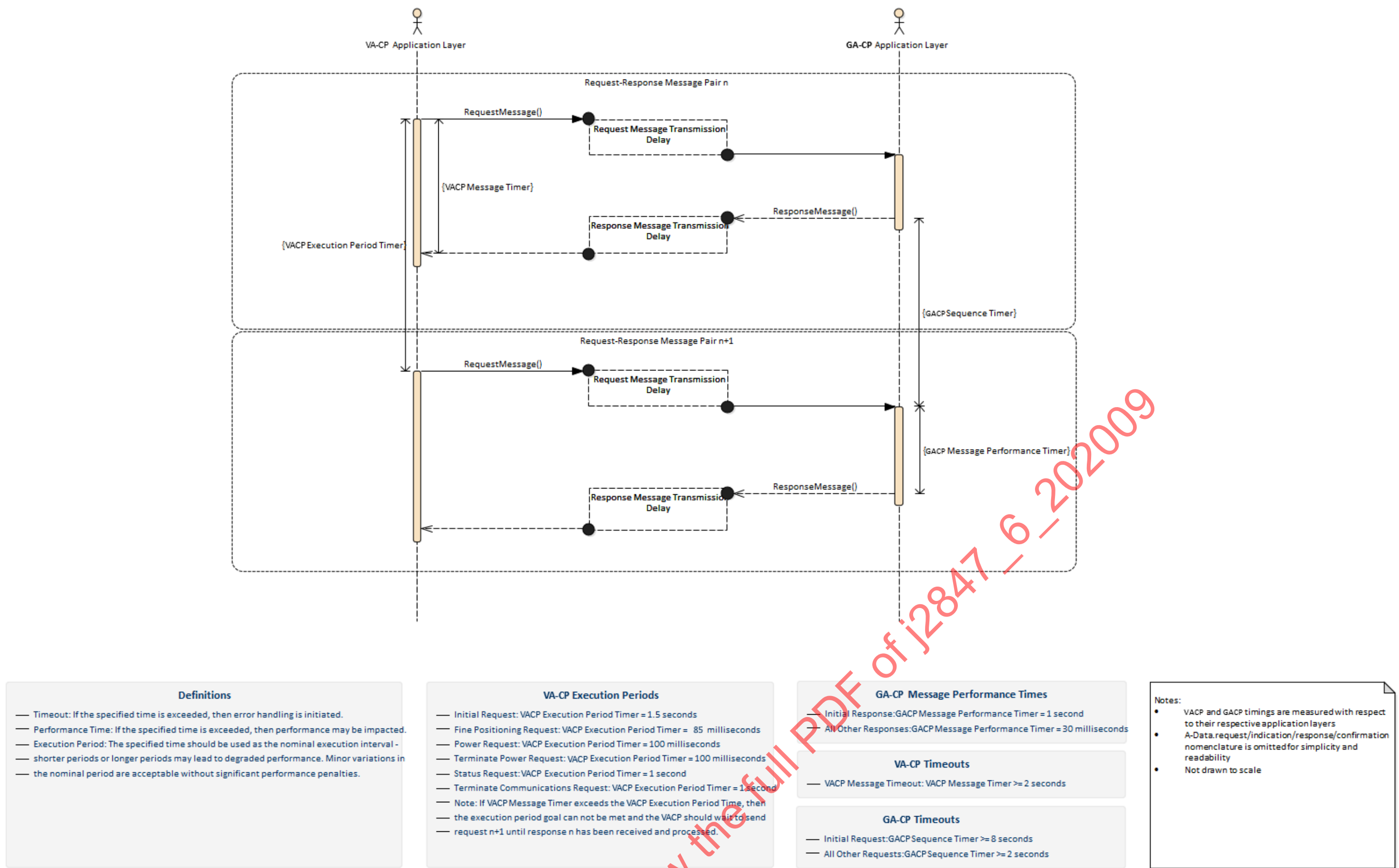
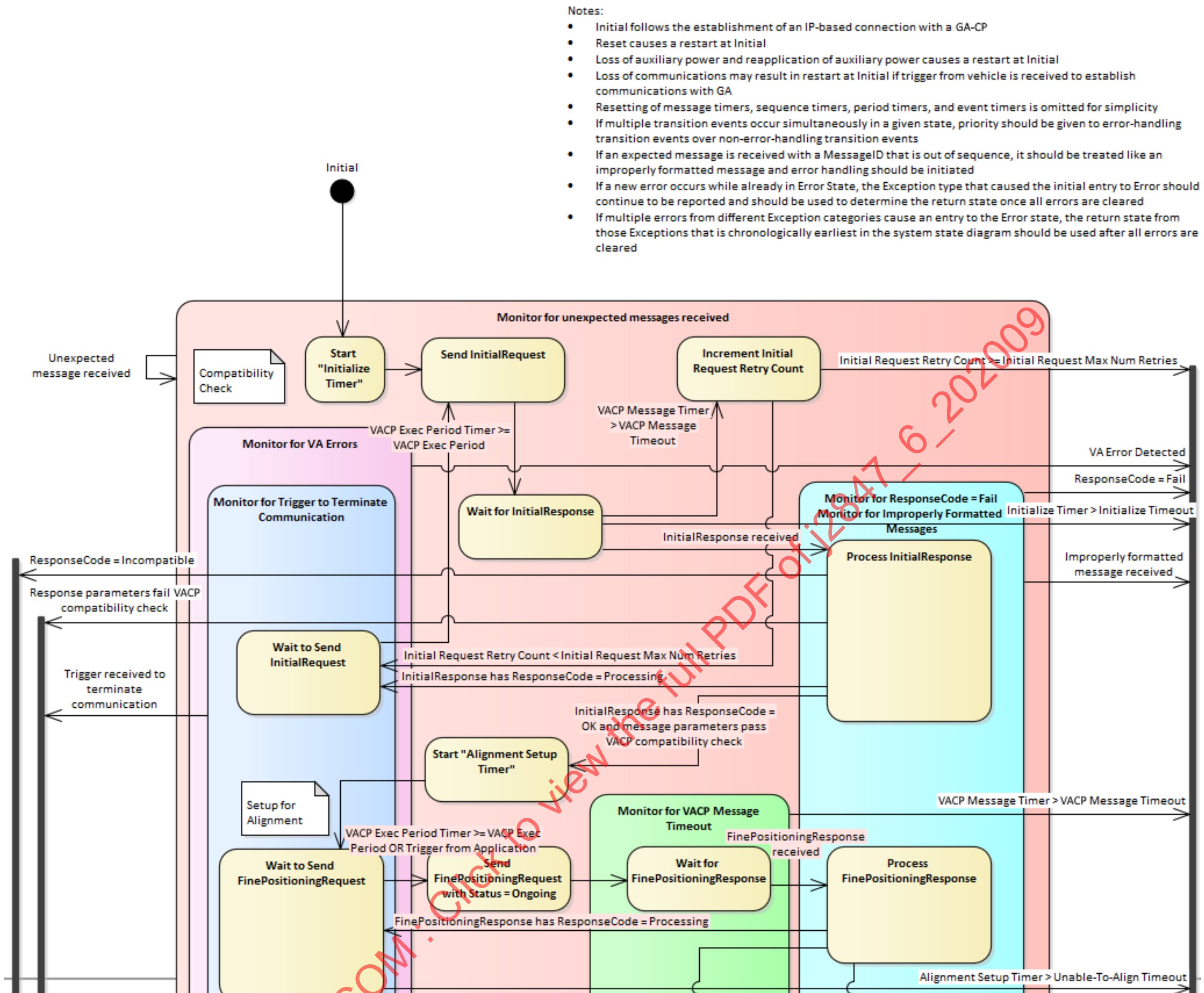


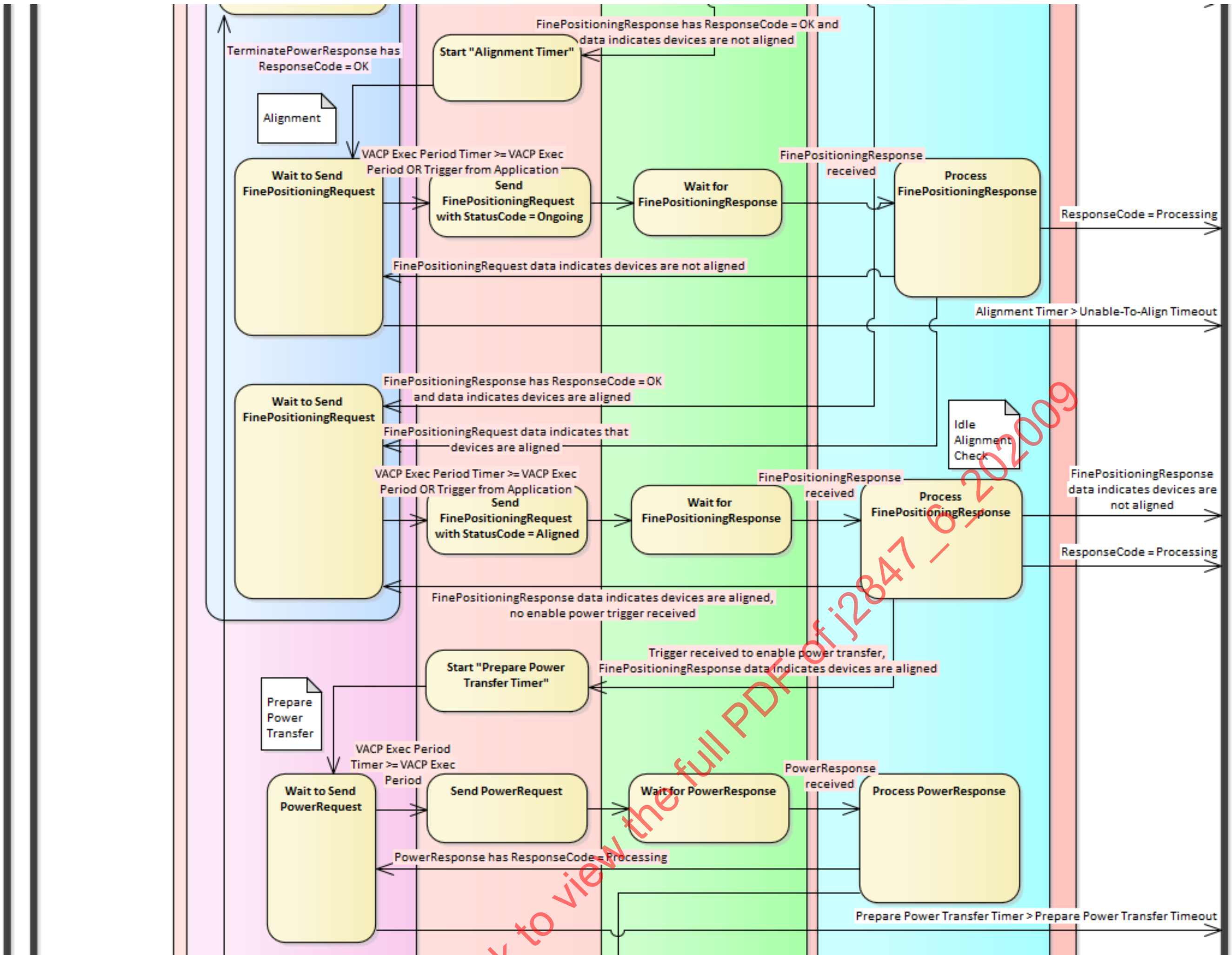
Figure 62 - Timing specifications

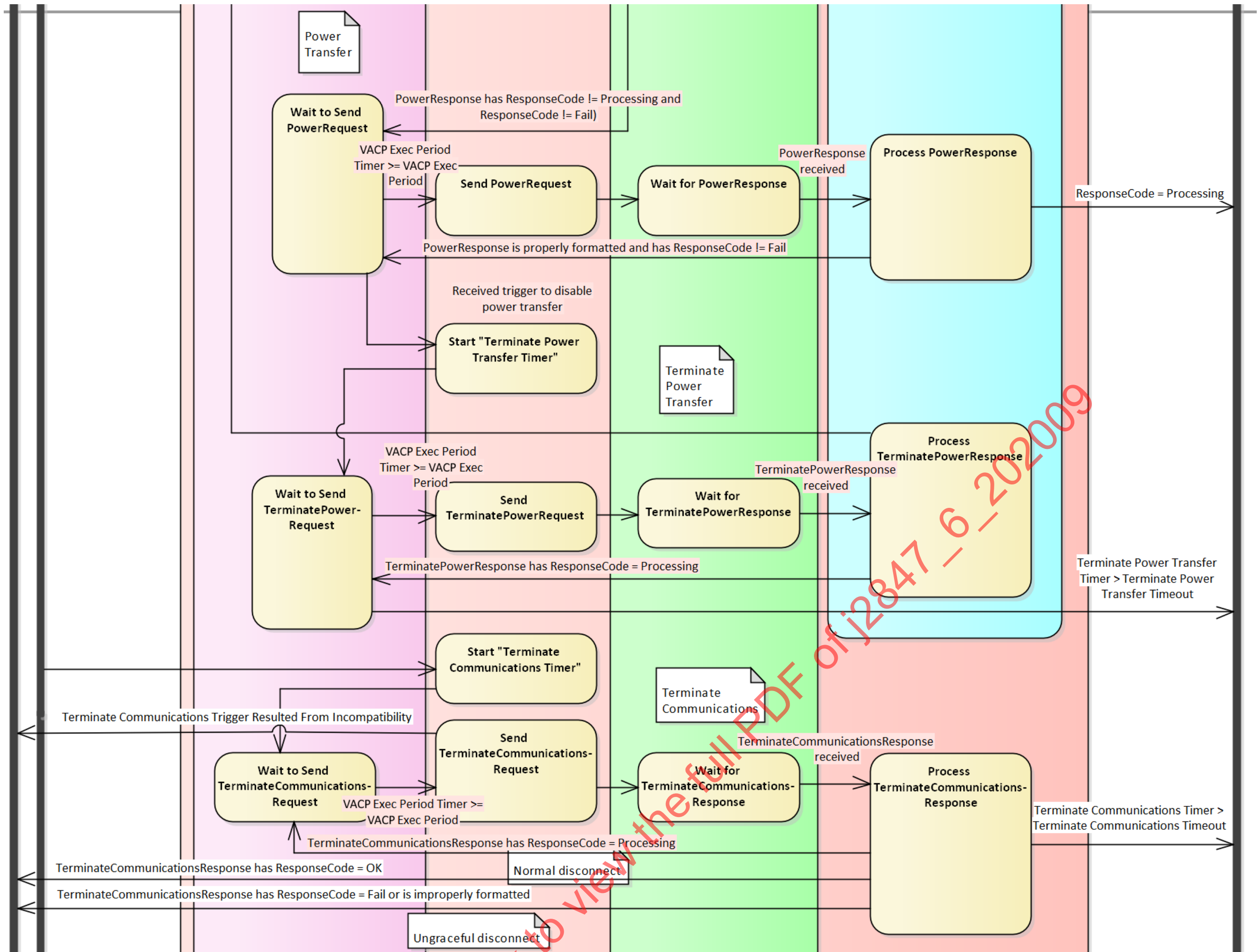
11. COMMUNICATION STATES

The communication state diagrams attempt to provide a detailed specification of the expected behavior of the VA-CP and GA-CP.

11.1 VA-CP Communication States




FinePositioningResponse has ResponseCode = OK and data indicates devices are not aligned
TerminatePowerResponse has ResponseCode = OK
Start "Alignment Timer"
Alignment
VACP Exec Period Timer >= VACP Exec Period OR Trigger from Application
Wait to Send FinePositioningRequest
Send FinePositioningRequest with StatusCode = Ongoing
Wait for FinePositioningResponse
FinePositioningResponse received
Process FinePositioningResponse
ResponseCode = Processing
FinePositioningRequest data indicates devices are not aligned
Alignment Timer > Unable-To-Align Timeout
FinePositioningResponse has ResponseCode = OK and data indicates devices are aligned
Wait to Send FinePositioningRequest
FinePositioningRequest data indicates that devices are aligned
Idle Alignment Check
VACP Exec Period Timer >= VACP Exec Period OR Trigger from Application
Send FinePositioningRequest with StatusCode = Aligned
Wait for FinePositioningResponse
FinePositioningResponse received
Process FinePositioningResponse
FinePositioningResponse data indicates devices are not aligned
ResponseCode = Processing
FinePositioningResponse data indicates devices are aligned, no enable power trigger received
Trigger received to enable power transfer, FinePositioningResponse data indicates devices are aligned
Start "Prepare Power Transfer Timer"
Prepare Power Transfer
VACP Exec Period Timer >= VACP Exec Period
Wait to Send PowerRequest
Send PowerRequest
Wait for PowerResponse
PowerResponse received
Process PowerResponse
PowerResponse has ResponseCode = Processing
Prepare Power Transfer Timer > Prepare Power Transfer Timeout


Power Transfer
Wait to Send PowerRequest
PowerResponse has ResponseCode != Processing and ResponseCode != Fail)
VACP Exec Period Timer >= VACP Exec Period
Send PowerRequest
Wait for PowerResponse
PowerResponse received
Process PowerResponse
ResponseCode = Processing
PowerResponse is properly formatted and has ResponseCode != Fail
Received trigger to disable power transfer
Start "Terminate Power Transfer Timer"
Terminate Power Transfer
Process TerminatePowerResponse
VACP Exec Period Timer >= VACP Exec Period
Wait to Send TerminatePower-Request
Send TerminatePowerRequest
Wait for TerminatePowerResponse
TerminatePowerResponse received
TerminatePowerResponse has ResponseCode = Processing
Terminate Power Transfer Timer > Terminate Power Transfer Timeout
Start "Terminate Communications Timer"
Terminate Communications
Terminate Communications Trigger Resulted From Incompatibility
Send TerminateCommunications-Request
TerminateCommunicationsResponse received
Wait to Send TerminateCommunications-Request
VACP Exec Period Timer >= VACP Exec Period
Wait for TerminateCommunications-Response
Process TerminateCommunications-Response
Terminate Communications Timer > Terminate Communications Timeout
TerminateCommunicationsResponse has ResponseCode = Processing
TerminateCommunicationsResponse has ResponseCode = OK
Normal disconnect
TerminateCommunicationsResponse has ResponseCode = Fail or is improperly formatted
Ungraceful disconnect

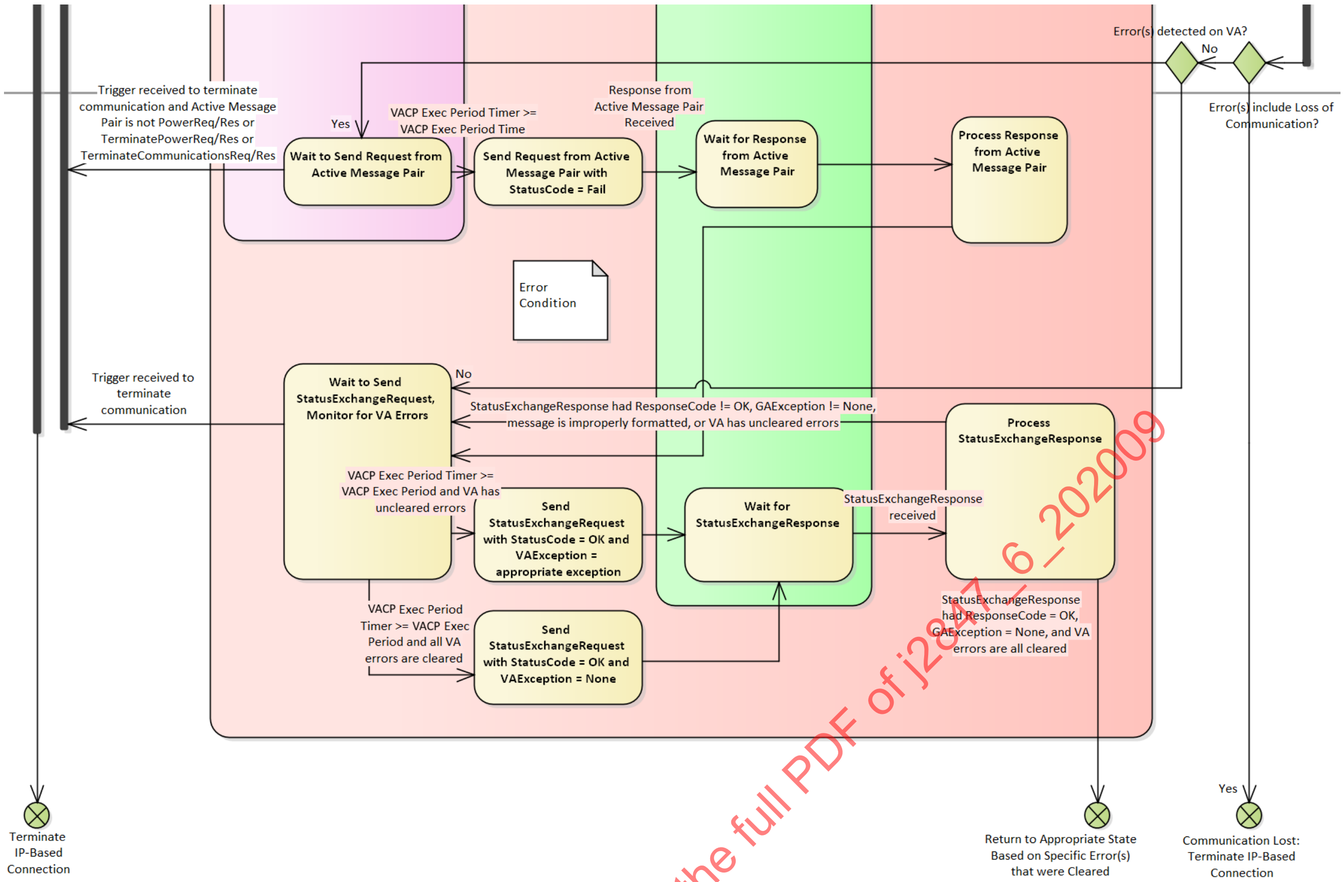
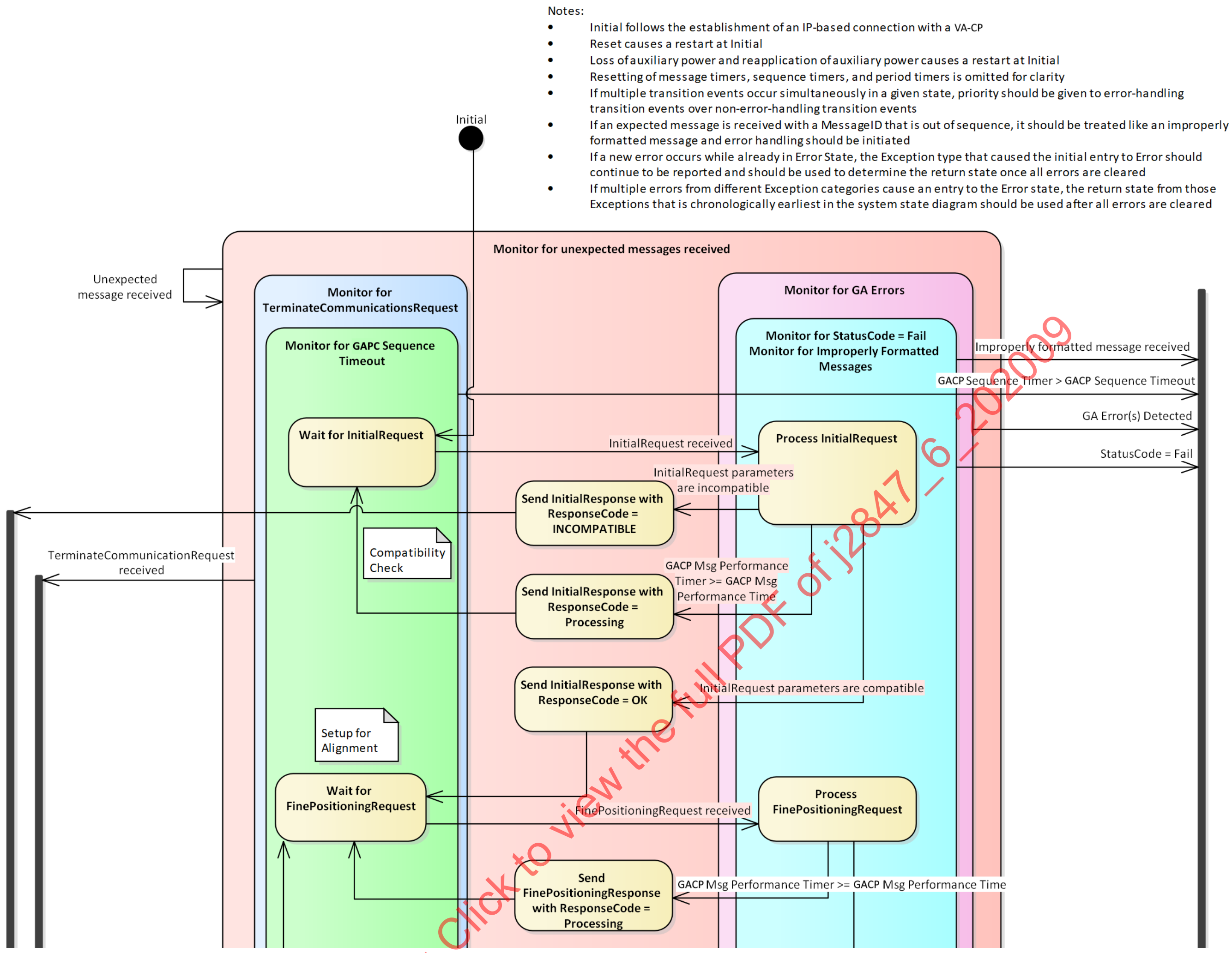
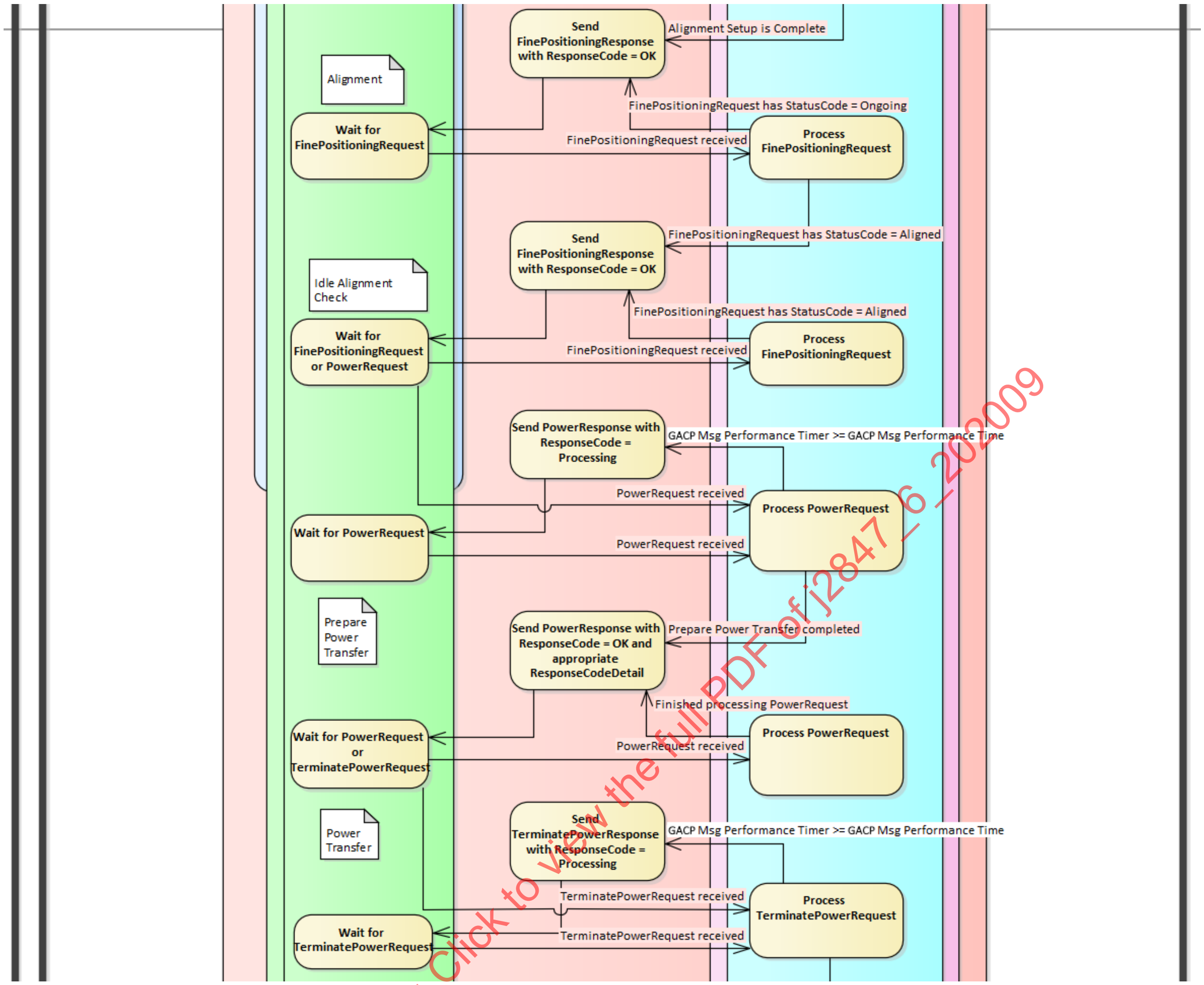


Figure 63 - VA-CP communication states

11.2 GA-CP Communication States




Send FinePositioningResponse with ResponseCode = OK
Alignment Setup is Complete
Alignment
FinePositioningRequest has StatusCode = Ongoing
Wait for FinePositioningRequest
FinePositioningRequest received
Process FinePositioningRequest
Send FinePositioningResponse with ResponseCode = OK
FinePositioningRequest has StatusCode = Aligned
Idle Alignment Check
FinePositioningRequest has StatusCode = Aligned
Wait for FinePositioningRequest or PowerRequest
FinePositioningRequest received
Process FinePositioningRequest
Send PowerResponse with ResponseCode = Processing
GACP Msg Performance Timer >= GACP Msg Performance Time
PowerRequest received
Process PowerRequest
Wait for PowerRequest
PowerRequest received
Prepare Power Transfer
Send PowerResponse with ResponseCode = OK and appropriate ResponseCodeDetail
Prepare Power Transfer completed
Finished processing PowerRequest
Wait for PowerRequest or TerminatePowerRequest
PowerRequest received
Process PowerRequest
Power Transfer
Send TerminatePowerResponse with ResponseCode = Processing
GACP Msg Performance Timer >= GACP Msg Performance Time
TerminatePowerRequest received
Process TerminatePowerRequest
Wait for TerminatePowerRequest
TerminatePowerRequest received

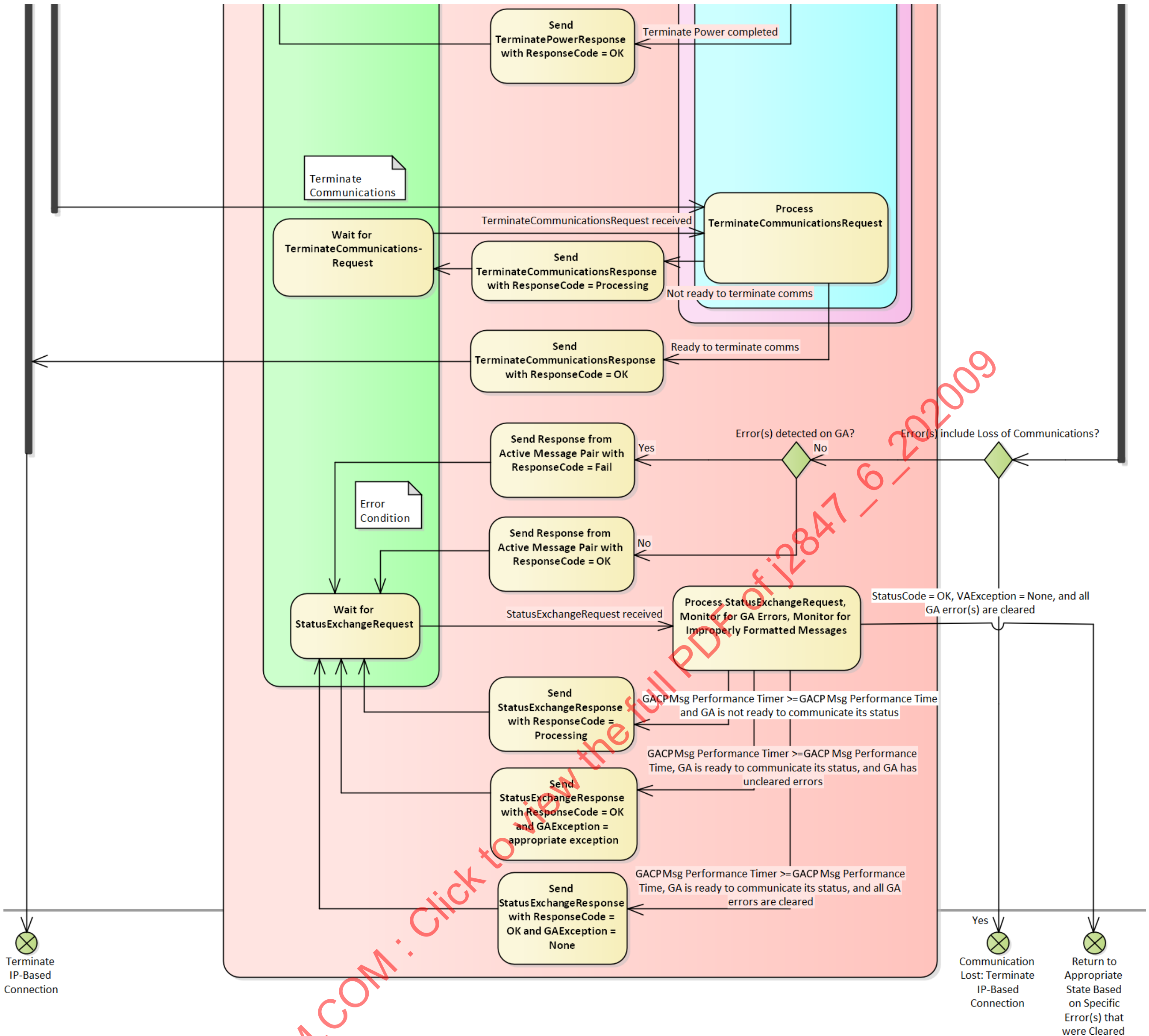


Figure 64 - GA-CP communication states

12. MESSAGE MATRIX

Below is an example enumeration table for the message matrix.

NOTE: In the table, the following notation is used: M = Mandatory; O = Optional.

				Initial		FinePositioning		Power		TerminatePower		Status Exchange		Terminate Communications	
Signal	**#**	**Value**	**More**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**
AlignStatusCode	0	Ongoing				M									
	1	Aligned				M									
	2	Fail				M									
CurrentToEV								O							
GACoilCurrent								O	O						
GACoilCurrentControl					M										
GACoilType	0	Circular			M										
	1	DD			M										
	2	Solenoid			M										
GAException	0	None									O		M		O
	1	SystemErrorInIdleOrPT									O		M		O
	2	SystemErrorInAA									O		M		O
	3	SystemErrorInSI									O		M		O
	4	SystemErrorInSB									O		M		O
GAFinePositioningMethods		GAStatus					O								
GAFrequency									O						
GAMaximumCoilCurrent					O										
GAMaximumDeliverablePower					M										
GAMaximumFrequency					M										
GAMaxTxPwr									O						
GAMinimumCoilCurrent					O										
GAMinimumFrequency					M										
GAMinTxPwr									O						
GAModel					M										
GANaturalOffset							M								

				Initial		FinePositioning		Power		TerminatePower		Status Exchange		Terminate Communications	
Signal	**#**	**Value**	**More**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**
GAPowerClass	0	WPT1			M										
	1	WPT2			M										
	2	WPT3			M										
	3	WPT4			M										
GAPowerDemandParameters		GAStatus							O						
GAProtocolVersion					O										
GAState	0	WPT_S_OOS									O		M		O
	1	WPT_S_SB									O		M		O
	2	WPT_S_SI									O		M		O
	3	WPT_S_AA									O		M		O
	4	WPT_S_IDLE									O		M		O
	5	WPT_S_PT									O		M		O
	6	WPT_S_ERR									O		M		O
GAStatus		GAException	None								O		M		O
			SystemErrorInIdleOrPT								O		M		O
			SystemErrorInAA								O		M		O
			SystemErrorInSI								O		M		O
			SystemErrorInSB								O		M		O
		GAState	WPT_S_OOS								O		M		O
			WPT_S_SB								O		M		O
			WPT_S_SI								O		M		O
			WPT_S_AA								O		M		O
			WPT_S_IDLE								O		M		O
			WPT_S_PT								O		M		O
			WPT_S_ERR								O		M		O
GASupportedFinePositioningMethods					O										
GATargetCoilCurrent					O										
GATerminateCommunicationsParameters															O
GATerminatePowerParameters		GAStatus									O				
GATxRx	0	TxRxID					O								

Signal	#	Value	More	Initial		FinePositioning		Power		TerminatePower		Status Exchange		Terminate Communications	
				Rq	Rs	Rq	Rs	Rq	Rs	Rq	Rs	Rq	Rs	Rq	Rs
	1	TxRxPosition					O								
	2	TxRxOrientation					O								
GAVendor					M										
GAZRangeSupported	0	Z1			M										
	1	Z1+Z2			M										
	2	Z1+Z2+Z3			M										
	3	Z2			M										
	4	Z2+Z3			M										
	5	Z3			M										
InitialResponseCode	0	OK			M										
	1	Processing			M										
	2	Incompatible			M										
	3	Fail			M										
InputGridPower									M		M				
LFMethod		VAIsTx				O									
		GAIsTx					O								
		VANumTxRx				O									
		GANumTxRx					O								
		VATxRx	TxRxID			O									
			TxRxPosition			O									
			TxRxOrientation			O									
		GATxRx	TxRxID				O								
			TxRxPosition				O								
			TxRxOrientation				O								
		VAPulseSequenceOrder				O									
		GAPulseSequenceOrder					O								
		VAPulseSeparationTime				O									
		GAPulseSeparationTime					O								
		VAPulseDuration				O									
		GAPulseDuration					O								

				Initial		FinePositioning		Power		TerminatePower		Status Exchange		Terminate Communications	
Signal	**#**	**Value**	**More**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**
		VAPackageSeparationTime				O									
		GAPackageSeparationTime					O								
MessageID				M	M	M	M	M	M	M	M	M	M	M	M
ProprietaryMethod		ProprietaryData				O	O								
ResponseCode	0	OK					M		M		M		M		M
	1	Processing					M		M		M		M		M
	3	Fail					M		M		M		M		M
ResponseCodeDetail	0	None							M						
	1	Thermal							M						
	2	GridLimit							M						
	3	InputCurrent							M						
	4	BaseCurrent							M						
	5	InternalCurrent							M						
	6	InternalVoltage							M						
	7	Frequency							M						
	8	ControlRange							M						
StatusCode	0	OK		M				M		M		M		M	
	1	Fail		M				M		M		M		M	
StatusCodeDetail	0	None						M							
	1	Thermal						M							
	2	Battery						M							
	3	Frequency						M							
	4	ControlRange						M							
	5	Shutdown						M							
VACoilType	0	Circular		M											
	1	DD		M											
	2	Solenoid		M											
VAControlLoop				M											
VAException	0	None								O		M		O	
	1	SystemErrorInIdleorPT								O		M		O	

				Initial		FinePositioning		Power		TerminatePower		Status Exchange		Terminate Communications	
Signal	**#**	**Value**	**More**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**
	2	SystemErrorInAA								O		M		O	
	3	SystemErrorInSI								O		M		O	
	4	SystemErrorInSB								O		M		O	
	5	SystemMisalignedInIdle								O		M		O	
	6	UnableToAlign								O		M		O	
VAFinePositioningParameters		VAStatus				O									
VAFrequency								O							
VAMaximumGroundClearance				M											
VAMaximumReceivablePower				M											
VAMinimumGroundClearance				M											
VAModel				M											
VANaturalFrequency				M											
VANaturalOffset						M									
VAPowerDemandParameters								O							
VAPowerReceived								M							
VAPowerRequest								M	M						
VAProtocolVersion				O											
VAState	0	WPT_V_OFF								O		M		O	
	1	WPT_V_SB								O		M		O	
	2	WPT_V_SI								O		M		O	
	3	WPT_V_AA								O		M		O	
	4	WPT_V_IDLE								O		M		O	
	5	WPT_V_PT								O		M		O	
	6	WPT_V_ErR								O		M		O	
VAStatus		VAException	None							O		M		O	
			SystemErrorInIdleorPT							O		M		O	
			SystemErrorInAA							O		M		O	
			SystemErrorInSI							O		M		O	
			SystemErrorInSB							O		M		O	
			SystemMisalignedInIdle							O		M		O	

				Initial		FinePositioning		Power		TerminatePower		Status Exchange		Terminate Communications	
Signal	**#**	**Value**	**More**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**	**Rq**	**Rs**
			UnableToAlign							O		M		O	
		VAState	WPT_V_OFF							O		M		O	
			WPT_V_SB							O		M		O	
			WPT_V_SI							O		M		O	
			WPT_V_AA							O		M		O	
			WPT_V_IDLE							O		M		O	
			WPT_V_PT							O		M		O	
			WPT_V_ErR							O		M		O	
VASupportedFinePositioningMethods				O											
VATerminateCommunicationsParameters		VAStatus												O	
VATerminatePowerParameters										O					
VATxRx	0	TxRxID				O									
	1	TxRxPosition				O									
	2	TxRxOrientation				O									
VAVendor				M											
VoltageToEV								O							

13. JSON SCHEMAS

This section should be used as the authority on how to properly form all JSON messages. The schema provides complete information where schema diagrams presented earlier in the document may abbreviate.

13.1 InitialRequest


```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "InitialRequest": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "MessageID": {
          "type": "integer",
          "description": "Initialize to 0. Must be incremented from last message received.",
          "maximum": 65535,
          "minimum": 0,
          "exclusiveMinimum": false,
          "exclusiveMaximum": false
        },
        "StatusCode": {
          "type": "string",
          "description": "Use Fail to communicate VA detection of an error to the GA.",
          "enum": [
            "OK",
            "Fail"
          ]
        },
        "VAMaximumReceivablePower": {
          "type": "integer",
          "description": "Number of watts that the VA is willing to receive, as measured at the output of the VA.",
          "maximum": 22000,
          "minimum": 0,
          "exclusiveMinimum": false,
          "exclusiveMaximum": false
        },
        "VAControlLoop": {
          "type": "boolean",
          "description": "\"true\" if VA executes a control loop, \"false\" otherwise."
        },
        "VAMaximumGroundClearance": {
          "type": "integer",
          "description": "Maximum height of the VA in mm.",
          "maximum": 250,
          "minimum": 100,
```

```
            "exclusiveMinimum": false,
            "exclusiveMaximum": false
        },
        "VAMinimumGroundClearance": {
            "type": "integer",
            "description": "Minimum height of the VA in mm.",
            "maximum": 250,
            "minimum": 100,
            "exclusiveMinimum": false,
            "exclusiveMaximum": false
        },
        "VACoilType": {
            "type": "string",
            "description": "Type of VA coil.",
            "enum": [
                "Circular",
                "DD",
                "Solenoid"
            ]
        },
        "VANaturalFrequency": {
            "type": "integer",
            "description": "The natural resonant frequency of the VA not in the presence of a GA.",
            "maximum": 10000000,
            "minimum": 1
        },
        "VAVendor": {
            "type": "string",
            "description": "Name of the VA vendor.",
            "maxLength": 64,
            "minLength": 1
        },
        "VAModel": {
            "type": "string",
            "description": "Model of VA device.",
            "maxLength": 64,
            "minLength": 1
        },
        "VAProtocolVersion": {
            "$ref": "CommonTypes.json#/definitions/ProtocolVersion"
        },
        "VASupportedFinePositioningMethods": {
            "$ref": "CommonTypes.json#/definitions/FinePositioningMethods"
        }
    },
    "required": [
        "MessageID",
        "StatusCode",
        "VAMaximumReceivablePower",
```

```
            "VAControlLoop",
            "VAMaximumGroundClearance",
            "VAMinimumGroundClearance",
            "VACoilType",
            "VANaturalFrequency",
            "VAVendor",
            "VAModel"
          ]
        }
    }
}
```


13.2 InitialResponse


```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "additionalProperties": false,
    "properties": {
        "InitialResponse": {
            "type": "object",
            "additionalProperties": false,
            "properties": {
                "MessageID": {
                    "type": "integer",
                    "description": "Must be incremented from the last message received.",
                    "maximum": 65535,
                    "minimum": 0,
                    "exclusiveMinimum": false,
                    "exclusiveMaximum": false
                },
                "InitialResponseCode": {
                    "type": "string",
                    "description": "Response code from GA.",
                    "enum": [
                        "OK",
                        "Processing",
                        "Incompatible",
                        "Fail"
                    ]
                },
                "GAPowerClass": {
                    "type": "string",
                    "description": "SAE J2954 power class of the GA.",
                    "enum": [
                        "WPT1",
                        "WPT2",
                        "WPT3",
                        "WPT4"
                    ]
```

```
        },
        "GAMaximumDeliverablePower": {
          "type": "integer",
          "description": "Number of watts that the GA is willing to transmit, as measured at the input of the GA.",
          "maximum": 22000,
          "minimum": 0,
          "exclusiveMinimum": false,
          "exclusiveMaximum": false
        },
        "GAZRangeSupported": {
          "type": "string",
          "description": "Z range supported by GA.",
          "enum": [
            "Z1",
            "Z1+Z2",
            "Z1+Z2+Z3",
            "Z2",
            "Z2+Z3",
            "Z3"
          ]
        },
        "GAMinimumFrequency": {
          "type": "number",
          "description": "Minimum frequency the GA supports in kHz (e.g. 82.5).",
          "multipleOf": 0.001,
          "maximum": 90,
          "minimum": 79,
          "exclusiveMinimum": false,
          "exclusiveMaximum": false
        },
        "GAMaximumFrequency": {
          "type": "number",
          "description": "Maximum frequency the GA supports in kHz (e.g. 82.5).",
          "multipleOf": 0.001,
          "maximum": 90,
          "minimum": 79,
          "exclusiveMinimum": false,
          "exclusiveMaximum": false
        },
        "GACoilCurrentControl": {
          "type": "boolean",
          "description": "\"true\" indicates the GA has explicit control over the current entering the GA coil, \"false\" indicates
no explicit current control."
        },
        "GACoilType": {
          "type": "string",
          "description": "Coil type of the GA.",
          "enum": [
            "Circular",
```

```
        "DD",
        "Solenoid"
      ]
    },
    "GAVendor": {
      "type": "string",
      "description": "Name of the GA vendor.",
      "maxLength": 64,
      "minLength": 1
    },
    "GAModel": {
      "type": "string",
      "description": "Model of the GA.",
      "maxLength": 64,
      "minLength": 1
    },
    "GAMaximumCoilCurrent": {
      "type": "number",
      "description": "Maximum current the GA can deliver into the GA coil.",
      "multipleOf": 0.1,
      "maximum": 127,
      "minimum": 0,
      "exclusiveMinimum": false,
      "exclusiveMaximum": false
    },
    "GAMinimumCoilCurrent": {
      "type": "number",
      "description": "Minimum current the GA can deliver into the GA coil.",
      "multipleOf": 0.1,
      "maximum": 127,
      "minimum": 0,
      "exclusiveMinimum": false,
      "exclusiveMaximum": false
    },
    "GATargetCoilCurrent": {
      "type": "number",
      "description": "Target current into the GA coil.",
      "multipleOf": 0.1,
      "maximum": 127,
      "minimum": 0,
      "exclusiveMinimum": false,
      "exclusiveMaximum": false
    },
    "GAProtocolVersion": {
      "$ref": "CommonTypes.json#/definitions/ProtocolVersion"
    },
    "GASupportedFinePositioningMethods": {
      "$ref": "CommonTypes.json#/definitions/FinePositioningMethods"
    }
```

```
        },
        "required": [
          "MessageID",
          "ResponseCode",
          "GAPowerClass",
          "GAMaximumDeliverablePower",
          "GAZRangeSupported",
          "GAMinimumFrequency",
          "GAMaximumFrequency",
          "GACoilCurrentControl",
          "GACoilType",
          "GAVendor",
          "GAModel"
        ]
      }
    }
}
```


13.3 FinePositioningRequest


```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "FinePositioningRequest": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "MessageID": {
          "type": "integer",
          "description": "Must be incremented from the last message received.",
          "maximum": 65535,
          "minimum": 0,
          "exclusiveMinimum": false,
          "exclusiveMaximum": false
        },
        "AlignStatusCode": {
          "type": "string",
          "description": "Status of VA.",
          "enum": [
            "Ongoing",
            "Aligned",
            "Fail"
          ]
        },
        "VANaturalOffset": {
          "type": "integer",
          "maximum": 32767,
          "minimum": -32768
```

```
            },
            "LFMethod": {
              "type": "object",
              "description": "Parameters for LF positioning method.",
              "additionalProperties": false,
              "properties": {
                "VAIsTx": {
                  "type": "boolean",
                  "description": "Does the VA contain the transmitters?"
                },
                "VANumTxRx": {
                  "type": "integer",
                  "description": "Number of transmitters/receivers on VA",
                  "maximum": 255,
                  "minimum": 0
                },
                "VATxRx": {
                  "type": "array",
                  "items": {
                    "type": "object",
                    "additionalProperties": false,
                    "maxProperties": 255,
                    "minProperties": 0,
                    "properties": {
                      "TxRxID": {
                        "type": "integer",
                        "description": "Transmitter/Receiver ID",
                        "maximum": 255,
                        "minimum": 0
                      },
                      "TxRxPosition": {
                        "type": "object",
                        "description": "Positiion in mm of the transmitter/receiver on the VA relative to the coil center.",
                        "additionalProperties": false,
                        "properties": {
                          "X": {
                            "type": "integer",
                            "description": "X location in mm relative to center of VA coil.",
                            "maximum": 32767,
                            "minimum": -32768
                          },
                          "Y": {
                            "type": "integer",
                            "description": "Y location in mm relative to center of VA coil.",
                            "maximum": 32767,
                            "minimum": -32768
                          },
                          "Z": {
                            "type": "integer",
```

```
                "description": "Z location in mm relative to center of VA coil.",
                "maximum": 32767,
                "minimum": -32768
              }
            },
            "required": [
              "X",
              "Y",
              "Z"
            ]
          },
          "TxRxOrientation": {
            "type": "object",
            "description": "Unit vector given the direction of measurement.  If no direction is applicable then all thr
ee values are set to zero.",
            "additionalProperties": false,
            "properties": {
              "XO": {
                "type": "number",
                "description": "X orientation vector component in mm relative to center of VA coil.",
                "multipleOf": 0.001,
                "maximum": 1,
                "minimum": -1
              },
              "YO": {
                "type": "number",
                "description": "Y orientation vector component in mm relative to center of VA coil.",
                "multipleOf": 0.001,
                "maximum": 1,
                "minimum": -1
              },
              "ZO": {
                "type": "number",
                "description": "Z orientation vector component in mm relative to center of VA coil.",
                "multipleOf": 0.001,
                "maximum": 1,
                "minimum": -1
              }
            },
            "required": [
              "XO",
              "YO",
              "ZO"
            ]
          }
        }
      }
    },
    "VAPulseSequenceOrder": {
```