2. Referenced Components and Variables

Below follows a list of all Component Variable combinations with a role standardized in this specification.

These Configuration Variables replace the Configuration Keys from OCPP 1.x

The list is split by functionality: General, Security, Authorization, Local Authorization List Management related, Authorization Cache, Transaction, Metering, Reservation, Smart Charging, Tariff & Cost, Diagnostics, Display Message and Charging Infrastructure related.

A required Configuration Variable mentioned under a particular function block only has to be supported by the Charging Station if it supports that functional block.

Please see chapter 4 in "Part 1 - Architecture & Topology" about the addressing of Components and Variables in the Device Model.

Requirements for all the Configuration Variables in this document:

- All variables that are writable SHALL have the VariableAttribute field: *persistence* = true, and SHALL thus be stored in a persistent way.
- Any fields not defined SHALL be left empty.
- Any field marked with a * (Asterisk) can be of any possible value.
- When the AttributeType is NOT given, the CSMS and Charging Station SHALL assume the AttributeType to be Actual.

NOTE See *'OCPP 2.0 Part 4 - JSON over Websockets implementation guide'* for a number of Configuration Variables that are specific to controlling the JSON/Websocket behavior.

2.1. General

2.1.1. `ActiveNetworkProfile`

Required	no		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	ActiveNetworkProfile	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Indicates the configuration profile the station uses at that moment to connect to the network. This configuration variable only has to be implemented when NetworkConnectionProfile is implemented.		

2.1.2. `AllowNewSessionsPendingFirmwareUpdate`

Required	no		
Component	**componentName**	ChargingStation	
Variable	**variableName**	AllowNewSessionsPendingFirmwareUpdate	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	Indicates whether new sessions can be started on EVSEs, while Charging Station is waiting for all EVSEs to become Available in order to start a pending firmware update. When a firmware update is waiting to be installed and this variable exists and has the value *true*, then, the Charging Station will not set free EVSEs to Unavailable, pending the update. This means that it may take longer until there is a point in time when all EVSEs of the Charging Station are free and it can perform the firmware update.		

2.1.3. `DefaultMessageTimeout`

Required	yes	
Component	**componentName**	OCPPCommCtrlr

Variable	**variableName**	MessageTimeout	
	variableInstance	Default	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**unit**	seconds
		dataType	integer
Description	The purpose of the message timeout is to be able to consider a request message as not sent and continue with other tasks when the message did not arrive due to communication errors or software failure. The message timeout setting in a Charging Station can be configured in the messageTimeout field in the *NetworkConnectionProfile*.		

2.1.4. FileTransferProtocols

Required	yes		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	FileTransferProtocols	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	MemberList
Description	List of supported file transfer protocols. Possible values: FTP, FTPS, HTTP, HTTPS, SFTP.		

2.1.5. HeartbeatInterval

Required	no		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	HeartbeatInterval	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	seconds
		dataType	integer
		minLimit	1
Description	Interval of inactivity (no OCPP exchanges) with CSMS after which the Charging Station should send HeartbeatRequest.		

2.1.6. NetworkConfigurationPriority

Required	yes		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	NetworkConfigurationPriority	
	variableAttributes	**attributeType**	Actual
		mutability	ReadWrite
	variableCharacteristics	**dataType**	SequenceList
		valueList	List of possible values
Description	A comma separated ordered list of the priority of the possible Network Connection Profiles. The list of possible available profile slots for the network configuration profiles SHALL be reported, via the valueList characteristic of this Variable.		

2.1.7. NetworkProfileConnectionAttempts

Required	yes		
Component	**componentName**	OCPPCommCtrlr	

Variable	**variableName**	NetworkProfileConnectionAttempts	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	integer
Description	Specifies the number of connection attempts the Charging Station executes before switching to a different profile.		

2.1.8. `OfflineThreshold`

Required	yes		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	OfflineThreshold	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	seconds
		dataType	integer
Description	When the offline period of a Charging Station exceeds the `OfflineThreshold` it is recommended to send a StatusNotificationRequest for all its Connectors when the Charging Station is back online.		

2.1.9. `QueueAllMessages`

Required	no		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	QueueAllMessages	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	When this variable is set to *true*, the Charging Station will queue all message until they are delivered to the CSMS. When set to *false* the Charging Station will only queue Transaction related messages as required in: E04.FR.01. and other requirements When this variable is the to *true*, and the Charging Station is running low on memory, the Charging Station SHALL drop TransactionEvent messages last, and when dropping measurements/meter data, the Charging Station SHALL drop intermediate values first (1st value, 3th value, 5th etc), not start dropping values from the beginning or end of the measurements/meter data. Default = false		

2.1.10. `MessageAttemptsTransactionEvent`

Required	yes		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	MessageAttempts	
	variableInstance	TransactionEvent	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	integer
Description	How often the Charging Station should try to submit a TransactionEventRequest message when the CSMS fails to process it.		

2.1.11. `MessageAttemptIntervalTransactionEvent`

Required	yes		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	MessageAttemptInterval	
	variableInstance	TransactionEvent	
	variableAttributes	**attributeType**	Actual
		mutability	ReadWrite
	variableCharacteristics	**unit**	seconds
		dataType	integer

Description	How long the Charging Station should wait before resubmitting a TransactionEventRequest message that the CSMS failed to process.

2.1.12. UnlockOnEVSideDisconnect

Required	yes		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	UnlockOnEVSideDisconnect	
	variableAttributes	**mutability**	ReadWrite/ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	When set to true, the Charging Station SHALL unlock the cable on the Charging Station side when the cable is unplugged at the EV. For an EVSE with only fixed cables, the mutability SHALL be ReadOnly and the actual value SHALL be false. For a charging station with fixed cables and sockets, the variable is only applicable to the sockets.		

2.1.13. WebSocketPingInterval

Required	no		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	WebSocketPingInterval	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	seconds
		dataType	integer
Description	Only relevant for websocket implementations. 0 disables client side websocket Ping/Pong. In this case there is either no ping/pong or the server initiates the ping and client responds with Pong. Positive values are interpreted as number of seconds between pings. Negative values are not allowed. SetConfiguration is expected to return a *Rejected* result.		

2.1.14. ResetRetries

Required	yes		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	ResetRetries	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	integer
Description	Number of times to retry a reset of the Charging Station when a reset was unsuccessful.		

2.1.15. MessageFieldLength

Required	no		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	FieldLength	
	variableInstance	<message>.<field>	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	This variable is used to report the length of <field> in <message> when it is larger than the length that is defined in the standard OCPP message schema.		

2.1.16. ItemsPerMessageGetReport

Required	yes		
Component	**componentName**	DeviceDataCtrlr	

Variable	**variableName**	ItemsPerMessage	
	variableInstance	GetReport	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Maximum number of ComponentVariable entries that can be sent in one getReportRequest or GetMonitoringReportRequest message.		

2.1.17. ItemsPerMessageGetVariables

Required	yes		
Component	**componentName**	DeviceDataCtrlr	
Variable	**variableName**	ItemsPerMessage	
	variableInstance	GetVariables	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Maximum number of GetVariableData objects in GetVariablesRequest.		

2.1.18. BytesPerMessageGetReport

Required	yes		
Component	**componentName**	DeviceDataCtrlr	
Variable	**variableName**	BytesPerMessage	
	variableInstance	GetReport	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Message Size (in bytes) - puts constraint on getReportRequest or GetMonitoringReportRequest message size.		

2.1.19. BytesPerMessageGetVariables

Required	yes		
Component	**componentName**	DeviceDataCtrlr	
Variable	**variableName**	BytesPerMessage	
	variableInstance	GetVariables	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Message Size (in bytes) - puts constraint on GetVariablesRequest message size.		

2.1.20. ConfigurationValueSize

Required	no		
Component	**componentName**	DeviceDataCtrlr	
Variable	**variableName**	ConfigurationValueSize	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	1000
Description	This Configuration Variable can be used to limit the following fields: SetVariableData.attributeValue and VariableCharacteristics.valueList. The max size of these values will always remain equal.		

2.1.21. ReportingValueSize

Required	no

Component	**componentName**	DeviceDataCtrlr	
Variable	**variableName**	ReportingValueSize	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	2500
Description	This Configuration Variable can be used to limit the following fields: GetVariableResult.attributeValue, VariableAttribute.value and EventData.actualValue. The max size of these values will always remain equal.		

2.1.22. ItemsPerMessageSetVariables

Required	yes		
Component	**componentName**	DeviceDataCtrlr	
Variable	**variableName**	ItemsPerMessage	
	variableInstance	SetVariables	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Maximum number of SetVariableData objects in SetVariablesRequest.		

2.1.23. BytesPerMessageSetVariables

Required	yes		
Component	**componentName**	DeviceDataCtrlr	
Variable	**variableName**	BytesPerMessage	
	variableInstance	SetVariables	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Message Size (in bytes) - puts constraint on SetVariablesRequest message size.		

2.1.24. DateTime

Required	yes		
Component	**componentName**	ClockCtrlr	
Variable	**variableName**	DateTime	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	DateTime
Description	Contains the current date and time.		

2.1.25. NtpSource

Required	no		
Component	**componentName**	ClockCtrlr	
Variable	**variableName**	NtpSource	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	OptionList
		valuesList	DHCP, manual
Description	When an NTP client is implemented, this variable can be used to configure the client: Use the NTP server provided via DHCP, or use the manually configured NTP server.		

2.1.26. NtpServerUri

Required	no

Component	**componentName**	ClockCtrlr	
Variable	**variableName**	NtpServerUri	
	variableInstance	Single digit, multiple servers allowed, primary NtpServer has instance '1', the secondary has instance '2'. etc	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	When an NTP client is implemented, this variable can be used to configure the client: This contains the address of the NTP server. Multiple NTP servers can be configured. These can be back-up NTP servers. If the NTP client supports it, it can also connect to multiple NTP servers simultaneous to get a more reliable time source.		

2.1.27. `TimeOffset`

Required	no		
Component	**componentName**	ClockCtrlr	
Variable	**variableName**	TimeOffset	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	Configured current local time offset in the format: "+01:00", "-02:00" etc. When a TimeOffset is used, it is advised not to implement: `TimeZone`. If a Charging Station has implemented both `TimeOffset` and `TimeZone` it is RECOMMENDED to not use both at the same time. The time offset is for display purposes.		

2.1.28. `NextTimeOffsetTransitionDateTime`

Required	no		
Component	**componentName**	ClockCtrlr	
Variable	**variableName**	NextTimeOffsetTransitionDateTime	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	DateTime
Description	Date time of the next time offset transition. On this date time, the clock displayed to the EV driver will be given the new offset as configured via 'TimeOffsetNextTransition'. This can be used to manually configure the next start or end of a daylight saving time period.		

2.1.29. `TimeOffsetNextTransition`

Required	no		
Component	**componentName**	ClockCtrlr	
Variable	**variableName**	TimeOffset	
	variableInstance	NextTransition	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	Next local time offset in the format: "+01:00", "-02:00" etc. New offset that will be set on the next time offset transition as configured via 'NextTimeOffsetTransitionDateTime'. This can be used to manually configure the offset for the start or end of the daylight saving time period.		

2.1.30. `TimeSource`

Required	yes

Component	**componentName**	ClockCtrlr	
Variable	**variableName**	TimeSource	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	SequenceList
		valuesList	List of all implemented time sources. Possible values: Heartbeat, NTP, GPS, RealTimeClock, MobileNetwork, RadioTimeTransmitter
Description	Via this variable, the Charging Station provides the CSMS with the option to configure a clock source, if more than 1 are implemented. By providing a list of possible sources, the CSO can configure fallback sources. Example: "NTP,Heartbeat" means, use NTP, but when none of the NTP servers responses, use time synchronization via Heartbeat. NOTE: RadioTimeTransmitter: At various locations around the globe, low-frequency radio transmitters provide accurate local time information e.g. DCF77 in Germany, MSF in the United Kingdom, JJY in Japan etc. Such a radio time clock can be used as a time source for a Charging Station. The Charging Station shall convert the broadcasted time to UTC. For this `TimeZone`, `TimeOffset`, 'NextTimeOffsetTransitionDateTime' and 'TimeOffsetNextTransition' can be used.		

2.1.31. `TimeZone`

Required	no		
Component	**componentName**	ClockCtrlr	
Variable	**variableName**	TimeZone	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	Configured current local time zone in the format: "Europe/Oslo", "Asia/Singapore" etc. When a time zone is used, it is advised not to implement: `TimeOffset`. If a Charging Station has implemented both `TimeOffset` and `TimeZone` it is RECOMMENDED to not use both at the same time. The time zone is for display purposes.		

2.1.32. `TimeAdjustmentReportingThreshold`

Required	no		
Component	**componentName**	ClockCtrlr	
Variable	**variableName**	TimeAdjustmentReportingThreshold	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	integer
Description	When the clock time is adjusted forwards or backwards for more then TimeAdjustmentReportingThreshold number of seconds, a SecurityEventNotification("SettingSystemTime") is sent by the charging station. A reasonable value is 20 seconds.		

2.1.33. `CustomImplementationEnabled`

Required	no		
Component	**componentName**	CustomizationCtrlr	
Variable	**variableName**	CustomImplementationEnabled	
	variableInstance	<VendorId>	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean

Description	This standard configuration variable can be used to enable/disable custom implementations that the Charging Station supports. It is recommended to first check if the custom behavior is able to be implemented using the device model, otherwise DataTransfer message(s) and/or CustomData fields can be used.

2.2. Security related

2.2.1. BasicAuthPassword

The basic authentication password is used for HTTP Basic Authentication. The configuration value is write-only, so that it cannot be accidentally stored in plaintext by the CSMS when it reads out all configuration values.

Required	no		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	BasicAuthPassword	
	variableAttributes	**mutability**	WriteOnly
	variableCharacteristics	**dataType**	passwordString
		maxLimit	40 (Max length of the BasicAuthPassword)
Description	The basic authentication password is used for HTTP Basic Authentication. The password SHALL be a randomly chosen passwordString with a sufficiently high entropy, consisting of minimum 16 and maximum 40 characters (alpha-numeric characters and the special characters allowed by passwordString). The password SHALL be sent as a UTF-8 encoded string (NOT encoded into octet string or base64). This configuration variable is write-only, so that it cannot be accidentally stored in plaintext by the CSMS when it reads out all configuration variables. This configuration variable is required unless only "security profile 3 - TLS with client side certificates" is implemented.		

2.2.2. Identity

Required	no		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	Identity	
	variableAttributes	**mutability**	ReadOnly or ReadWrite
	variableCharacteristics	**dataType**	identifierString
		maxLimit	48 (Charging Station Identity)
Description	The Charging Station identity. identity is an identifierString, however because this value is also used as the basic authentication username, the colon character ':' SHALL not be used. Maximum length was chosen to ensure compatibility with EVSE ID from [EMI3-BO] "Part 2: business objects".		

2.2.3. OrganizationName

Required	yes		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	OrganizationName	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	This configuration variable is used to set the organization name of the CSO or an organization trusted by the CSO. It is used to set the O (*organizationName*) RDN in the subject field of the client certificate. See also A00.FR.509.		

2.2.4. CertificateEntries

Required	yes	
Component	**componentName**	SecurityCtrlr

Variable	**variableName**	CertificateEntries	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	Maximum number of Certificates installed at any time.
Description	Amount of Certificates currently installed on the Charging Station.		

2.2.5. SecurityProfile

Required	yes		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	SecurityProfile	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	This configuration variable is used to report the security profile used by the Charging Station.		

2.2.6. AdditionalRootCertificateCheck

Required	no		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	AdditionalRootCertificateCheck	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	When set to true, only one certificate (plus a temporarily fallback certificate) of certificateType CSMSRootCertificate is allowed to be installed at a time. When installing a new CSMS Root certificate, the new certificate SHALL replace the old one AND the new CSMS Root Certificate MUST be signed by the old CSMS Root Certificate it is replacing. This configuration variable is required unless only "security profile 1 - Unsecured Transport with Basic Authentication" is implemented. Please note that security profile 1 SHOULD only be used in trusted networks. *Note: When using this additional security mechanism please be aware that the Charging Station needs to perform a full certificate chain verification when the new CSMS Root certificate is being installed. However, once the old CSMS Root certificate is set as the fallback certificate, the Charging Station needs to perform a partial certificate chain verification when verifying the server certificate during the TLS handshake. Otherwise the verification will fail once the old CSMS Root (fallback) certificate is either expired or removed.* *Note 2: The statement that the variable is required, means that the configuration variable must be present, but does NOT indicate that the feature must be implemented. This is an optional feature. By setting the value to false, the Charging Station indicates that it does not support this feature, whereas true means that it does support the feature.*		

2.2.7. MaxCertificateChainSize

Required	no		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	MaxCertificateChainSize	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	10000
Description	This configuration variable can be used to limit the size of the 'certificateChain' field from the CertificateSignedRequest PDU. This value SHOULD NOT be set too small. The smaller this value, the less security architectures the Charging Station will support. It is RECOMMENDED to set at least a size of 5600. This will allow the Charging Station to support most security architectures.		

2.2.8. CertSigningWaitMinimum

Required	no		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	CertSigningWaitMinimum	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	seconds
		dataType	integer
Description	This configuration variable defines how long the Charging Station has to wait before generating another CSR, in the case the CSMS accepts the SignCertificateRequest, but never returns the signed certificate back. This value will be doubled after every attempt. The amount of attempts is configured at CertSigningRepeatTimes If the certificate signing process is slow, this setting allows the CSMS to tell the Charging Station to allow more time.		

2.2.9. CertSigningRepeatTimes

Required	no		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	CertSigningRepeatTimes	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	integer
Description	This variable can be used to configure the amount of times the Charging Station SHALL double the previous back-off time, starting with the number of seconds configured at CertSigningWaitMinimum, every time the back-off time expires without having received the CertificateSignedRequest containing the from the CSR generated signed certificate. When the maximum number of increments is reached, the Charging Station SHALL stop resending the SignCertificateRequest, until it is requested by the CSMS using a TriggerMessageRequest.		

2.3. Authorization related

2.3.1. AuthEnabled

Required	no		
Component	**componentName**	AuthCtrlr	
Variable	**variableName**	Enabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If set to *false*, then no authorization is done before starting a transaction or when reading an idToken. If an idToken was provided, then it will be put in the *idToken* field of the TransactionEventRequest. If no idToken was provided, then *idToken* in TransactionEventRequest will be left empty and *type* is set to `NoAuthorization`.		

2.3.2. AdditionalInfoItemsPerMessage

Required	no		
Component	**componentName**	AuthCtrlr	
Variable	**variableName**	AdditionalInfoItemsPerMessage	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Maximum number of AdditionalInfo items that can be sent in one message. This configuration variable only has to be implemented when AdditionalInfo is implemented.		

2.3.3. OfflineTxForUnknownIdEnabled

Required	no		
Component	**componentName**	AuthCtrlr	

Variable	**variableName**	OfflineTxForUnknownIdEnabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this key exists, the Charging Station supports Unknown Offline Authorization. If this key reports a value of *true*, Unknown Offline Authorization is enabled.		

2.3.4. AuthorizeRemoteStart

Required	yes		
Component	**componentName**	AuthCtrlr	
Variable	**variableName**	AuthorizeRemoteStart	
	variableAttributes	**mutability**	ReadOnly or ReadWrite. Choice is up to Charging Station implementation.
	variableCharacteristics	**dataType**	boolean
Description	Whether a remote request to start a transaction in the form of RequestStartTransactionRequest message should be authorized beforehand like a local action to start a transaction.		

2.3.5. LocalAuthorizeOffline

Required	yes		
Component	**componentName**	AuthCtrlr	
Variable	**variableName**	LocalAuthorizeOffline	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	Whether the Charging Station, when *Offline*, will start a transaction for locally-authorized identifiers.		

2.3.6. LocalPreAuthorize

Required	yes		
Component	**componentName**	AuthCtrlr	
Variable	**variableName**	LocalPreAuthorize	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	Whether the Charging Station, when online, will start a transaction for locally-authorized identifiers without waiting for or requesting an AuthorizeResponse from the CSMS.		

2.3.7. MasterPassGroupId

Required	no		
Component	**componentName**	AuthCtrlr	
Variable	**variableName**	MasterPassGroupId	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
		maxLimit	36 (The maximum string length of MasterPassGroupId)
Description	IdTokens that have this id as groupId belong to the Master Pass Group. Meaning they can stop any ongoing transaction, but cannot start transactions. This can, for example, be used by law enforcement personal to stop any ongoing transaction when an EV has to be towed away.		

2.3.8. DisableRemoteAuthorization

Required	no

Component	**componentName**	AuthCtrlr	
Variable	**variableName**	DisableRemoteAuthorization	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	When set to *true* this instructs the Charging Station to not issue any AuthorizationRequests, but only use Authorization Cache and Local Authorization List to determine validity of idTokens. *Note: The difference with `DisablePostAuthorize` is that this variable disables all authorization with CSMS, whereas `DisablePostAuthorize` only disables re-authorization of tokens that are as not-Accepted in the Authorization Cache or Local Authorization List.*		

2.4. Authorization Cache related

2.4.1. `AuthCacheEnabled`

NOTE When the value of this variable is changed, the content of the authorization cache should not be altered.

Required	no		
Component	**componentName**	AuthCacheCtrlr	
Variable	**variableName**	Enabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable exists and reports a value of *true*, Authorization Cache is enabled.		

2.4.2. `AuthCacheAvailable`

Required	no		
Component	**componentName**	AuthCacheCtrlr	
Variable	**variableName**	Available	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	If this variable exists and reports a value of *true*, Authorization Cache is supported, but not necessarily enabled.		

2.4.3. `AuthCacheLifeTime`

Required	no		
Component	**componentName**	AuthCacheCtrlr	
Variable	**variableName**	LifeTime	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	Seconds
		dataType	integer
Description	Indicates how long it takes until a token expires in the authorization cache since it is last used.		

2.4.4. `AuthCacheStorage`

Required	no		
Component	**componentName**	AuthCacheCtrlr	
Variable	**variableName**	Storage	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	The maximum number of bytes

Description	Indicates the number of bytes currently used by the Authorization Cache. MaxLimit indicates the maximum number of bytes that can be used by the Authorization Cache.

2.4.5. AuthCachePolicy

Required	no		
Component	**componentName**	AuthCacheCtrlr	
Variable	**variableName**	Policy	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	OptionList
		valuesList	LRU, LFU, FIFO, CUSTOM
Description	Cache Entry Replacement Policy: least recently used, least frequently used, first in first out, other custom mechanism.		

2.4.6. AuthCacheDisablePostAuthorize

Required	no		
Component	**componentName**	AuthCacheCtrlr	
Variable	**variableName**	DisablePostAuthorize	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	When set to *true* this variable disables the behavior to request authorization for an idToken that is stored in the cache with a status other than `Accepted`, as stated in C10.FR.03 and C12.FR.05.		

2.5. Local Authorization List Management related

2.5.1. LocalAuthListEnabled

Required	no		
Component	**componentName**	LocalAuthListCtrlr	
Variable	**variableName**	Enabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable exists and reports a value of *true*, Local Authorization List is enabled.		

2.5.2. LocalAuthListEntries

Required	when `LocalAuthListAvailable` is *true*		
Component	**componentName**	LocalAuthListCtrlr	
Variable	**variableName**	Entries	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	The maximum number of IdTokens that can be stored in the Local Authorization List.
Description	Amount of IdTokens currently in the Local Authorization List. The maxLimit of this variable SHALL be provided to report the maximum number of IdTokens that can be stored in the Local Authorization List.		

2.5.3. LocalAuthListAvailable

Required	no		
Component	**componentName**	LocalAuthListCtrlr	

Variable	**variableName**	Available	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	If this variable exists and reports a value of *true*, Local Authorization List is supported.		

2.5.4. ItemsPerMessageSendLocalList

Required	when `LocalAuthListAvailable` is *true*		
Component	**componentName**	LocalAuthListCtrlr	
Variable	**variableName**	ItemsPerMessage	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer

2.5.5. BytesPerMessageSendLocalList

Required	when `LocalAuthListAvailable` is *true*		
Component	**componentName**	LocalAuthListCtrlr	
Variable	**variableName**	BytesPerMessage	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer

2.5.6. LocalAuthListStorage

Required	no		
Component	**componentName**	LocalAuthListCtrlr	
Variable	**variableName**	Storage	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	The maximum number of bytes
Description	Indicates the number of bytes currently used by the Local Authorization List. MaxLimit indicates the maximum number of bytes that can be used by the Local Authorization List.		

2.5.7. LocalAuthListDisablePostAuthorize

Required	no		
Component	**componentName**	LocalAuthListCtrlr	
Variable	**variableName**	DisablePostAuthorize	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	When set to *true* this variable disables the behavior to request authorization for an idToken that is stored in the local authorization list with a status other than `Accepted`, as stated in C14.FR.03.		

2.6. Transaction related

2.6.1. EVConnectionTimeOut

Required	yes	
Component	**componentName**	TxCtrlr

Variable	**variableName**	EVConnectionTimeOut	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	seconds
		dataType	integer
Description	Interval from between "starting" of a transaction until incipient transaction is automatically canceled, due to failure of EV driver to (correctly) insert the charging cable connector(s) into the appropriate socket(s). The Charging Station SHALL go back to the original state, probably: 'Available'. "Starting" might be the swiping of the RFID, pressing a start button, a RequestStartTransactionRequest being received etc.		

2.6.2. StopTxOnEVSideDisconnect

Required	yes		
Component	**componentName**	TxCtrlr	
Variable	**variableName**	StopTxOnEVSideDisconnect	
	variableAttributes	**mutability**	ReadWrite or ReadOnly, depending on Charging Station implementation.
	variableCharacteristics	**dataType**	boolean
Description	When set to *true*, the Charging Station SHALL deauthorize the transaction when the cable is unplugged from the EV.		

2.6.3. TxBeforeAcceptedEnabled

Required	no		
Component	**componentName**	TxCtrlr	
Variable	**variableName**	TxBeforeAcceptedEnabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	With this configuration variable the Charging Station can be configured to allow charging before having received a BootNotificationResponse with RegistrationStatus: Accepted. See: Transactions before being accepted by a CSMS.		

2.6.4. TxStartPoint

Required	yes		
Component	**componentName**	TxCtrlr	
Variable	**variableName**	TxStartPoint	
	variableAttributes	**mutability**	ReadOnly or ReadWrite. Choice is up to Charging Station implementation.
	variableCharacteristics	**dataType**	MemberList
		valueList	See TxStartStopPoint values for allowed values. It is not required to implement all possible values.
Description	Defines when the Charging Station starts a new transaction: first transactioneventRequest: eventType = Started. When any event in the given list occurs, the Charging Station SHALL start a transaction. The Charging Station SHALL only send the `Started` event once for every transaction. It is advised to put all events that should be part of a transaction in the list, in case the start event never occurs. Because the possible events don't always have to come in the same order it is possible to provide a list of events. Which ever comes first will then cause a transaction to be started. For example: EVConnected, Authorized would mean that a transaction is started when an EV is detected (Cable is connected), or when an EV Driver swipes his RFID card en the CSMS successfully authorizes the ID for charging.		

2.6.5. TxStopPoint

Required	yes	
Component	**componentName**	TxCtrlr

Variable	**variableName**	TxStopPoint	
	variableAttributes	**mutability**	ReadOnly or ReadWrite. Choice is up to Charging Station implementation.
	variableCharacteristics	**dataType**	MemberList
		valueList	See TxStartStopPoint values for allowed values. It is not required to implement all possible values.
Description	Defines when the Charging Station ends a transaction: last transactioneventRequest: eventType = Ended. When any event in the given list is no longer valid, the Charging Station SHALL end the transaction. The Charging Station SHALL only send the `Ended` event once for every transaction.		

2.6.6. TxStartStopPoint values

2.6.6.1. TxStartPoint values

The following table lists the values allowed for the `TxStartPoint` variable. These values represent logical steps or events that (may) occur during a charging session. When such an event occurs, and it is listed in in the `TxStartPoint` variable, then this marks the start of a transaction.

Value	Description
ParkingBayOccupancy	An object (probably an EV) is detected in the parking/charging bay.
EVConnected	Both ends of the Charging Cable have been connected (if this can be detected, else detection of a cable being plugged into the socket), or for wireless charging: initial communication between EVSE and EV is established.
Authorized	Driver or EV has been authorized, this can also be some form of anonymous authorization like a start button.
PowerPathClosed	All preconditions for charging have been met, power can flow. This event is the logical AND of `EVConnected` and `Authorized` and should be used if a transaction is supposed to start when EV is connected and authorized. Despite its name, this event is not related to the state of the power relay. Note: There may be situations where `PowerPathClosed` does not imply that charging starts at that moment, e.g. because of delayed charging or a battery that is too hot.
EnergyTransfer	Energy is being transferred between EV and EVSE.
DataSigned	The moment when the signed meter value is received from the fiscal meter, that is used in the TransactionEventRequest with *context* = `Transaction.Begin` and *triggerReason* = `SignedDataReceived`. This TxStartPoint might be applicable when legislation exists that only allows a billable transaction to start when the first signed meter value has been received.

2.6.6.2. TxStopPoint values

The following table lists the values allowed for the `TxStopPoint` variable. These values represent logical steps or events that (may) occur during a charging session. When such an event occurs, and it is listed in in the `TxStopPoint` variable, then this marks the end of a transaction.

The values are the same as for `TxStartPoint`, but in this case the meaning is different, since it refers to the ending of the event, rather than the start. For use with `TxStopPoint` each value should be interpreted as if it had "Not" prefixed to it. See the following table:

Value	Description
ParkingBayOccupancy	An object (probably an EV) is **no longer** detected in the parking/charging bay.
EVConnected	One or both ends of the Charging Cable have been **disconnected** (if this can be detected, else detection of a cable being **unplugged** from the socket), or for wireless charging: communication between EVSE and EV is **lost**.

Value	Description
Authorized	Driver or EV **is no longer** authorized, this can also be some form of anonymous authorization like a start button.
PowerPathClosed	All preconditions for charging **are no longer met**, power **cannot** flow. This event is the logical OR of `EVConnected` and `Authorized` and should be used if a transaction is supposed to end when EV is **disconnected** and/or **deauthorized**. It is exactly the same as having the values `EVConnected`, `Authorized` in `TxStopPoint`. Despite its name, this event is not related to the state of the power relay.
EnergyTransfer	Energy is **not** being transferred between EV and EVSE. This is not recommended to use as a `TxStopPoint`, because it will stop the transaction as soon as EV or EVSE (temporarily) suspend the charging.
DataSigned	This condition has no meaning as a `TxStopPoint` and should not be used as such.

2.6.7. `MaxEnergyOnInvalidId`

Required	no		
Component	**componentName**	TxCtrlr	
Variable	**variableName**	MaxEnergyOnInvalidId	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	Wh
		dataType	integer
Description	Maximum amount of energy in Wh delivered when an identifier is deauthorized by the CSMS after start of a transaction.		

2.6.8. `StopTxOnInvalidId`

Required	yes		
Component	**componentName**	TxCtrlr	
Variable	**variableName**	StopTxOnInvalidId	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	whether the Charging Station will deauthorize an ongoing transaction when it receives a non- *Accepted* authorization status in TransactionEventResponse for this transaction.		

2.7. Metering related

2.7.1. `SampledDataEnabled`

Required	no		
Component	**componentName**	SampledDataCtrlr	
Variable	**variableName**	Enabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable reports a value of *true*, Sampled Data is enabled.		

2.7.2. `SampledDataAvailable`

Required	no	
Component	**componentName**	SampledDataCtrlr

Variable	**variableName**	Available	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	If this variable reports a value of *true*, Sampled Data is supported.		

2.7.3. SampledDataSignReadings

Required	no		
Component	**componentName**	SampledDataCtrlr	
Variable	**variableName**	SignReadings	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If set to *true*, the Charging Station SHALL include signed meter values in the TransactionEventRequest to the CSMS. Some Charging Stations might only be able to sign `Transaction.Begin` and `Transaction.End` meter values. When a Charging Station does not support signed meter values it SHALL NOT report this variable.		

2.7.4. SampledDataTxEndedMeasurands

Required	yes		
Component	**componentName**	SampledDataCtrlr	
Variable	**variableName**	TxEndedMeasurands	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	MemberList
		maxLimit	The maximum length of the CSV formatted string, to be defined by the implementer.
Description	Sampled measurands to be included in the *meterValues* element of TransactionEventRequest (eventType = Ended), every `SampledDataTxEndedInterval` seconds from the start of the transaction. The Charging Station reports the list of supported Measurands in VariableCharacteristicsType.valuesList of this variable. This way the CSMS knows which Measurands it can put in the `TxEndedSampledData`. When left empty, no sampled measurands SHALL be put into the TransactionEventRequest (eventType = Ended).		

2.7.5. SampledDataTxEndedInterval

Required	yes		
Component	**componentName**	SampledDataCtrlr	
Variable	**variableName**	TxEndedInterval	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	seconds
		dataType	integer
Description	Interval between sampling of metering (or other) data, intended to be transmitted in the TransactionEventRequest (eventType = Ended) message. For transaction data (evseId>0), samples are acquired and transmitted only in the TransactionEventRequest (eventType = Ended) message. A value of "0" (numeric zero), by convention, is to be interpreted to mean that only the values taken at the *start* and *end* of a transaction should be transmitted (no intermediate values).		

2.7.6. SampledDataTxStartedMeasurands

Required	yes		
Component	**componentName**	SampledDataCtrlr	

Variable	**variableName**	TxStartedMeasurands	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	MemberList
		maxLimit	The maximum length of the CSV formatted string, to be defined by the implementer.
Description	Sampled measurand(s) to be taken at the start of any transaction to be included in the meterValues field of the first TransactionEventRequest message send at the start of a transaction (eventType = Started). The Charging Station reports the list of supported Measurands in VariableCharacteristicsType.valuesList of this variable. This way the CSMS knows which Measurands it can put in the `SampledDataTxStartedMeasurands`. If the Charging Station has a meter, recommended to use as default: "Energy.Active.Import.Register"		

2.7.7. `SampledDataTxUpdatedMeasurands`

Required	yes		
Component	**componentName**	SampledDataCtrlr	
Variable	**variableName**	TxUpdatedMeasurands	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	MemberList
		maxLimit	The maximum length of the CSV formatted string, to be defined by the implementer.
Description	Sampled measurands to be included in the *meterValues* element of every TransactionEventRequest (eventType = Updated), every `SampledDataTxUpdatedInterval` seconds from the start of the transaction. The Charging Station reports the list of supported Measurands in VariableCharacteristicsType.valuesList of this variable. This way the CSMS knows which Measurands it can put in the `SampledDataTxUpdatedMeasurands`. If the Charging Station has a meter, recommended to use as default: "Energy.Active.Import.Register"		

2.7.8. `SampledDataTxUpdatedInterval`

Required	yes		
Component	**component Name**	SampledDataCtrlr	
Variable	**variableName**	TxUpdatedInterval	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	seconds
		dataType	integer
Description	Interval between sampling of metering (or other) data, intended to be transmitted via TransactionEventRequest (eventType = Updated) messages. For transaction data (evseId>0), samples are acquired and transmitted periodically at this interval from the start of the charging transaction. A value of "0" (numeric zero), by convention, is to be interpreted to mean that no sampled data should be transmitted during the transaction.		

2.7.9. `AlignedDataEnabled`

Required	no		
Component	**componentName**	AlignedDataCtrlr	
Variable	**variableName**	Enabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable reports a value of *true*, Aligned Data is enabled.		

2.7.10. AlignedDataAvailable

Required	no		
Component	**componentName**	AlignedDataCtrlr	
Variable	**variableName**	Available	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	If this variable reports a value of *true*, Aligned Data is supported.		

2.7.11. AlignedDataMeasurands

Required	yes		
Component	**componentName**	AlignedDataCtrlr	
Variable	**variableName**	Measurands	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	MemberList
		maxLimit	The maximum length of the CSV formatted string, to be defined by the implementer.
Description	Clock-aligned measurand(s) to be included in MeterValuesRequest, every `AlignedDataInterval` seconds. For all the allowed values see: Measurand. The Charging Station reports the list of supported Measurands in VariableCharacteristicsType.valuesList of this variable. This way the CSMS knows which Measurands it can put in the `AlignedDataMeasurands`.		

2.7.12. AlignedDataInterval

Required	yes		
Component	**componentName**	AlignedDataCtrlr	
Variable	**variableName**	Interval	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	seconds
		dataType	integer
Description	Size (in seconds) of the clock-aligned data interval, intended to be transmitted in the MeterValuesRequest message. This is the size (in seconds) of the set of evenly spaced aggregation intervals per day, starting at 00:00:00 (midnight). For example, a value of 900 (15 minutes) indicates that every day should be broken into 96 15-minute intervals. When clock aligned data is being transmitted, the interval in question is identified by the start time and (optional) duration interval value, represented according to the ISO8601 standard. All "per-period" data (e.g. energy readings) should be accumulated (for "flow" type measurands such as energy), or averaged (for other values) across the entire interval (or partial interval, at the beginning or end of a transaction), and transmitted (if so enabled) at the end of each interval, bearing the interval start time timestamp. A value of "0" (numeric zero), by convention, is to be interpreted to mean that no clock-aligned data should be transmitted.		

2.7.13. AlignedDataSendDuringIdle

Required	no		
Component	**componentName**	AlignedDataCtrlr	
	evse	*	
Variable	**variableName**	SendDuringIdle	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If set to *true*, the Charging Station SHALL NOT send clock aligned meter values when a transaction is ongoing. When an EVSE is specified, it SHALL stop sending the clock aligned meter values for this EVSE when it has an ongoing transaction. When no EVSE is specified, it SHALL stop sending the clock aligned meter values when any transaction is ongoing on this Charging Station.		

2.7.14. AlignedDataSignReadings

Required	no		
Component	**componentName**	AlignedDataCtrlr	
Variable	**variableName**	SignReadings	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If set to *true*, the Charging Station SHALL include signed meter values in the SampledValueType in the TransactionEventRequest to the CSMS for those measurands defined in `AlignedDataTxEndedMeasurands`. When a Charging Station does not support signed meter values it SHALL NOT report this variable.		

2.7.15. AlignedDataTxEndedMeasurands

Required	yes		
Component	**componentName**	AlignedDataCtrlr	
Variable	**variableName**	TxEndedMeasurands	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	MemberList
		maxLimit	The maximum length of the CSV formatted string, to be defined by the implementer.
Description	Clock-aligned periodic measurand(s) to be included in the *meterValues* element of TransactionEventRequest (eventType = Ended) for every `AlignedDataTxEndedInterval` of the transaction. The Charging Station reports the list of supported Measurands in VariableCharacteristicsType.valuesList of this variable. This way the CSMS knows which Measurands it can put in the `TxEndedAlignedData`. When left empty, no Clock-aligned measurands SHALL be put into the TransactionEventRequest (eventType = Ended).		

2.7.16. AlignedDataTxEndedInterval

Required	yes		
Component	**componentName**	AlignedDataCtrlr	
Variable	**variableName**	TxEndedInterval	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	seconds
		dataType	integer
Description	Size (in seconds) of the clock-aligned data interval, intended to be transmitted in the TransactionEventRequest (eventType = Ended) message. This is the size (in seconds) of the set of evenly spaced aggregation intervals per day, starting at 00:00:00 (midnight). For example, a value of 900 (15 minutes) indicates that every day should be broken into 96 15-minute intervals. When clock aligned data is being collected, the interval in question is identified by the start time and (optional) duration interval value, represented according to the ISO8601 standard. All "per-period" data (e.g. energy readings) should be accumulated (for "flow" type measurands such as energy), or averaged (for other values) across the entire interval (or partial interval, at the beginning or end of a transaction), and transmitted (if so enabled) at the end of the transaction in 1 TransactionEventRequest (eventType = Ended) message.		

2.7.17. PublicKeyWithSignedMeterValue

Required	no		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	PublicKeyWithSignedMeterValue	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	OptionList
		valueList	Never,OncePerTransaction,EveryMeterValue

Description	This Configuration Variable can be used to configure whether a public key needs to be sent with a signed meter value. Note, that the field is required, so it needs to be present as an empty string when the public key is not sent.

2.7.18. SampledDataRegisterValuesWithoutPhases

Required	no		
Component	**componentName**	SampledDataCtrlr	
Variable	**variableName**	RegisterValuesWithoutPhases	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable reports a value of *true*, then meter values of measurand `Energy.Active.Import.Register` will only report the total energy over all phases without reporting the individual phase values. If this variable is absent or *false*, then the value for each phase is reported, possibly also with a total value (depending on the meter).		

2.8. Reservation related

2.8.1. ReservationEnabled

Required	no		
Component	**componentName**	ReservationCtrlr	
Variable	**variableName**	Enabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	Whether Reservation is enabled.		

2.8.2. ReservationAvailable

Required	no		
Component	**componentName**	ReservationCtrlr	
Variable	**variableName**	Available	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	Whether Reservation is supported.		

2.8.3. ReservationNonEvseSpecific

Required	no		
Component	**componentName**	ReservationCtrlr	
Variable	**variableName**	NonEvseSpecific	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	If this configuration variable is present and set to *true*: Charging Station supports Reservation where EVSE id is not specified.		

2.9. Smart Charging related

2.9.1. SmartChargingEnabled

Required	no	
Component	**componentName**	SmartChargingCtrlr

Variable	**variableName**	Enabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	Whether Smart Charging is enabled.		

2.9.2. SmartChargingAvailable

Required	no		
Component	**componentName**	SmartChargingCtrlr	
Variable	**variableName**	Available	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	Whether Smart Charging is supported.		

2.9.3. ACPhaseSwitchingSupported

Required	no		
Component	**componentName**	SmartChargingCtrlr	
Variable	**variableName**	ACPhaseSwitchingSupported	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	This variable can be used to indicate an on-load/in-transaction capability. If defined and true, this EVSE supports the selection of which phase to use for 1 phase AC charging.		

2.9.4. ChargingProfileMaxStackLevel

Required	yes		
Component	**componentName**	SmartChargingCtrlr	
Variable	**variableName**	ProfileStackLevel	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Maximum acceptable value for *stackLevel* in a ChargingProfile. Since the lowest *stackLevel* is 0, this means that if SmartChargingCtrlr.ProfileStackLevel = 1, there can be at most 2 valid charging profiles per Charging Profile Purpose per EVSE.		

2.9.5. ChargingScheduleChargingRateUnit

Required	yes		
Component	**componentName**	SmartChargingCtrlr	
Variable	**variableName**	RateUnit	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	MemberList
Description	A list of supported quantities for use in a ChargingSchedule. Allowed values: 'A' and 'W'		

2.9.6. PeriodsPerSchedule

Required	yes		
Component	**componentName**	SmartChargingCtrlr	

Variable	**variableName**	PeriodsPerSchedule	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Maximum number of periods that may be defined per ChargingSchedule.		

2.9.7. ExternalControlSignalsEnabled

Required	no		
Component	**componentName**	SmartChargingCtrlr	
Variable	**variableName**	ExternalControlSignalsEnabled	
	variableAttributes	**mutability**	ReadOnly or ReadWrite. Choice is up to Charging Station implementation.
	variableCharacteristics	**dataType**	boolean
Description	Indicates whether a Charging Station should respond to external control signals that influence charging.		

2.9.8. NotifyChargingLimitWithSchedules

Required	no		
Component	**componentName**	SmartChargingCtrlr	
Variable	**variableName**	NotifyChargingLimitWithSchedules	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	Indicates if the Charging Station should include the externally set charging limit/schedule in the message when it sends a NotifyChargingLimitRequest message. This might increase the data usage significantly, especially when an external system sends new profiles/limits with a short interval. Default is false when omitted.		

2.9.9. Phases3to1

Required	no		
Component	**componentName**	SmartChargingCtrlr	
Variable	**variableName**	Phases3to1	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	If defined and true, this Charging Station supports switching from 3 to 1 phase during a transaction.		

2.9.10. ChargingProfileEntries

Required	yes		
Component	**componentName**	SmartChargingCtrlr	
Variable	**variableName**	Entries	
	VariableInstance	ChargingProfiles	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	Maximum number of Charging profiles installed at any time.
Description	Amount of Charging profiles currently installed on the Charging Station.		

2.9.11. LimitChangeSignificance

Required	yes	
Component	**componentName**	SmartChargingCtrlr

Variable	**variableName**	LimitChangeSignificance	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	decimal
Description	If at the Charging Station side a change in the limit in a ChargingProfile is lower than this percentage, the Charging Station MAY skip sending a NotifyChargingLimitRequest or a TransactionEventRequest message to the CSMS. It is RECOMMENDED to set this key to a low value. See Smart Charging signals to a Charging Station from multiple actors.		

2.10. Tariff & Cost related

2.10.1. TariffEnabled

Required	no		
Component	**componentName**	TariffCostCtrlr	
Variable	**variableName**	Enabled	
	variableInstance	Tariff	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	Whether Tariff is enabled.		

2.10.2. TariffAvailable

Required	no		
Component	**componentName**	TariffCostCtrlr	
Variable	**variableName**	Available	
	variableInstance	Tariff	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	Whether Tariff is supported.		

2.10.3. TariffFallbackMessage

Required for Charging Stations supporting Tariff Information.

Required	yes		
Component	**componentName**	TariffCostCtrlr	
Variable	**variableName**	TariffFallbackMessage	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
		maxLimit	255
Description	Message (and/or tariff information) to be shown to an EV Driver when there is no driver specific tariff information available.		

2.10.4. CostEnabled

Required	no		
Component	**componentName**	TariffCostCtrlr	
Variable	**variableName**	Enabled	
	variableInstance	Cost	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	Whether Cost is enabled.		

2.10.5. CostAvailable

Required	no		
Component	**componentName**	TariffCostCtrlr	
Variable	**variableName**	Available	
	variableInstance	Cost	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	Whether Cost is supported.		

2.10.6. TotalCostFallbackMessage

Required for Charging Stations supporting Tariff Information.

Required	yes		
Component	**componentName**	TariffCostCtrlr	
Variable	**variableName**	TotalCostFallbackMessage	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
		maxLimit	255
Description	Message to be shown to an EV Driver when the Charging Station cannot retrieve the cost for a transaction at the end of the transaction.		

2.10.7. Currency

Required for Charging Stations supporting Tariff Information.

Required	yes		
Component	**componentName**	TariffCostCtrlr	
Variable	**variableName**	Currency	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
		maxLimit	3
Description	Currency used by this Charging Station in a ISO 4217 [ISO4217] formatted currency code.		

2.11. Diagnostics related

2.11.1. MonitoringEnabled

Required	no		
Component	**componentName**	MonitoringCtrlr	
Variable	**variableName**	Enabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	Whether Monitoring is enabled.		

2.11.2. MonitoringAvailable

Required	no		
Component	**componentName**	MonitoringCtrlr	

Variable	**variableName**	Available	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	Whether Monitoring is supported.		

2.11.3. ItemsPerMessageClearVariableMonitoring

Required	no		
Component	**componentName**	MonitoringCtrlr	
Variable	**variableName**	ItemsPerMessage	
	variableInstance	ClearVariableMonitoring	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Maximum number of IDs in a ClearVariableMonitoringRequest.		

2.11.4. ItemsPerMessageSetVariableMonitoring

Required	yes		
Component	**componentName**	MonitoringCtrlr	
Variable	**variableName**	ItemsPerMessage	
	variableInstance	SetVariableMonitoring	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Maximum number of setMonitoringData elements that can be sent in one setVariableMonitoringRequest message.		

2.11.5. BytesPerMessageClearVariableMonitoring

Required	no		
Component	**componentName**	MonitoringCtrlr	
Variable	**variableName**	BytesPerMessage	
	variableInstance	ClearVariableMonitoring	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Message Size (in bytes) - puts constraint on ClearVariableMonitoringRequest message size.		

2.11.6. BytesPerMessageSetVariableMonitoring

Required	yes		
Component	**componentName**	MonitoringCtrlr	
Variable	**variableName**	BytesPerMessage	
	variableInstance	SetVariableMonitoring	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Message Size (in bytes) - puts constraint on setVariableMonitoringRequest message size.		

2.11.7. OfflineMonitoringEventQueuingSeverity

Required	no	
Component	**componentName**	MonitoringCtrlr

Variable	**variableName**	OfflineQueuingSeverity	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	integer
Description	When set and the Charging Station is *offline*, the Charging Station shall queue any notifyEventRequest messages triggered by a monitor with a severity number equal to or lower than the severity configured here. Value ranging from 0 (Emergency) to 9 (Debug).		

2.11.8. `ActiveMonitoringBase`

Required	no		
Component	**componentName**	MonitoringCtrlr	
Variable	**variableName**	ActiveMonitoringBase	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	OptionList
Description	Shows the currently used MonitoringBase. Valid values according MonitoringBaseEnumType: `All`, `FactoryDefault`, `HardwiredOnly`.		

2.11.9. `ActiveMonitoringLevel`

Required	no		
Component	**componentName**	MonitoringCtrlr	
Variable	**variableName**	ActiveMonitoringLevel	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Shows the currently used MonitoringLevel. Valid values are severity levels of SetMonitoringLevelRequest: 0-9.		

2.12. Display Message related

2.12.1. `DisplayMessageEnabled`

Required	no		
Component	**componentName**	DisplayMessageCtrlr	
Variable	**variableName**	Enabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	Whether Display Message is enabled.		

2.12.2. `DisplayMessageAvailable`

Required	no		
Component	**componentName**	DisplayMessageCtrlr	
Variable	**variableName**	Available	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	Whether Display Message is supported.		

2.12.3. `NumberOfDisplayMessages`

Required	yes	
Component	**componentName**	DisplayMessageCtrlr

Variable	**variableName**	DisplayMessages	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	Maximum number of different messages that can configured in this Charging Station simultaneous, via SetDisplayMessageRequest.
Description	Amount of different messages that are currently configured in this Charging Station, via SetDisplayMessageRequest		

2.12.4. DisplayMessageSupportedFormats

Required	yes		
Component	**componentName**	DisplayMessageCtrlr	
Variable	**variableName**	SupportedFormats	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	MemberList
Description	List of message formats supported by this Charging Station. Possible values: MessageFormat.		

2.12.5. DisplayMessageSupportedPriorities

Required	yes		
Component	**componentName**	DisplayMessageCtrlr	
Variable	**variableName**	SupportedPriorities	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	MemberList
Description	List of the priorities supported by this Charging Station. Possible values: MessagePriority.		

2.13. Charging Infrastructure related

2.13.1. Available

Required	yes		
Components	**componentName**	ChargingStation	
		EVSE	
		Connector	
	evse	* (*for EVSE and Connector*)	
Variable	**variableName**	Available	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	When *true* the Component exists and is locally configured/wired for use, but may not be (remotely) Enabled. This variable is required on any Component that can be reported by the Charging Station. As a minimum it shall exist on ChargingStation, EVSE and Connector.		
Note	If any other variables are reported for a Component, then reporting *Available* does not add much value and may be omitted. However, the variable needs to exist, because it can be queried for by a GetCustomReport request for all Components that are 'available'. EVSE and Connector components are addressed on their respective tier. So, EVSE #1 is addressed as component EVSE on tier "*evse* = 1" and connector #1 on this EVSE is addressed as component Connector on tier "*evse* = 1, *connector* = 1.		

2.13.2. AvailabilityState

Required	yes		
Components	**componentName**	ChargingStation	
		EVSE	
	evse	* (*for EVSE*)	
Variable	**variableName**	AvailabilityState	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	optionList
		valuesList	Available, Occupied, Reserved, Unavailable, Faulted
Description	This variable reports current availability state for the ChargingStation and EVSE. If a Connector has its own availability state independent of the EVSE, then this variable may be used to report the Connector's availability state. As such it replicates ConnectorStatus values reported in StatusNotification messages. An EVSE component is addressed on its own tier. So, EVSE #1 is addressed as component EVSE on tier "*evse* = 1.		

2.13.3. AllowReset

Required	no		
Component	**componentName**	EVSE	
	evse	*	
Variable	**variableName**	AllowReset	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	Component can be reset. Can be used to announce that an EVSE can be reset individually.		

2.13.4. ConnectorType

Required	yes		
Component	**componentName**	Connector	
	evse	*	
Variable	**variableName**	ConnectorType	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	string
Description	Value of the type of connector as defined by ConnectorEnumType in "Part 2 - Specification" plus additionally: `cGBT`, `cChaoJi`, `OppCharge`.		

2.13.5. PhaseRotation

Required	no		
Component	**componentName**	*	
	evse	*	
Variable	**variableName**	PhaseRotation	
	variableAttributes	**mutability**	ReadOnly or ReadWrite.
	variableCharacteristics	**dataType**	String

Description	This variable describes the phase rotation of a Component relative to its parent Component, using a three letter string consisting of the letters: R, S, T and x. The letter 'R' can be identified as phase 1 (L1), 'S' as phase 2 (L2), 'T' as phase 3 (L3). The lower case 'x' is used to designate a phase that is not connected. An empty string means that phase rotation is not applicable or not known. Certain measurands, like voltage and current, are reported with a phase relative to the grid connection. In order to support this, all components in the chain from Connector to ElectricalFeed need to have a value for PhaseRotation. Some examples: "" (unknown) "RST" (Standard Reference Phasing) "RTS" (Reversed Reference Phasing) "SRT" (Reversed 240 degree rotation) "STR" (Standard 120 degree rotation) "TRS" (Standard 240 degree rotation) "TSR" (Reversed 120 degree rotation) "RSx" (Two phases connected) "Rxx" (One phase connected)

2.13.6. SupplyPhases

Required	yes		
Components	**componentName**	ChargingStation	
		EVSE	
		Connector	
	evse	* (*for EVSE and Connector*)	
Variable	**variableName**	SupplyPhases	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Number of alternating current phases connected/available. 1 or 3 for AC, 0 means DC (no alternating phases). Null value indicates that the number of phases (e.g. in use) is unknown.		

2.13.7. Power

Required	yes (*maxLimit only*)		
Component	**componentName**	EVSE	
	evse	*	
Variable	**variableName**	Power	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	decimal
		maxLimit	decimal
Description	The variableCharacteristic *maxLimit*, that holds the maximum power that this EVSE can provide, is required. The *Actual* value of the instantaneous (real) power is desired, but not required.		

2.13.8. Example Reporting of EVSEs and Connectors via device model

The following example illustrates how the device model reports EVSEs and Connectors for an example charging station that has two EVSEs, of which EVSE #1 has one Type2 connector and EVSE #2 has two connectors: CCS and CHAdeMO.

Component				Variable		VariableAttribute		VariableCharacteristics		
name	**evse id**	**evse connectorId**	**instance**	**name**	**instance**	**type**	**value**	**dataType**	**maxLimit**	**supportsMonitoring**
ChargingStation				Available		Actual	true	boolean		false
ChargingStation				AvailabilityState		Actual	Available	boolean		false
ChargingStation				SupplyPhases		Actual	integer	3		false
ChargingStation				ACCurrent	"L1"	Actual	decimal	45.0		true
ChargingStation				ACCurrent	"L2"	Actual	decimal	44.9		true
ChargingStation				ACCurrent	"L3"	Actual	decimal	44.9		true
EVSE	1		"left"	Available		Actual	true	boolean		false
EVSE	1		"left"	AvailabilityState		Actual	Available	optionList		false
EVSE	1		"left"	SupplyPhases		Actual	3	integer		false
EVSE	1		"left"	Power		Actual	0.0	decimal	22000.0	true
Connector	1	1		Available		Actual	true	boolean		false
Connector	1	1		ConnectorType		Actual	sType2	string		false
Connector	1	1		SupplyPhases		Actual	3	integer		false
EVSE	2		"right"	Available		Actual	true	boolean		false
EVSE	2		"right"	AvailabilityState		Actual	Occupied	optionList		false
EVSE	2		"right"	SupplyPhases		Actual	0	integer		false
EVSE	2		"right"	Power		Actual	41000.0	decimal	50000.0	true
Connector	2	1		Available		Actual	true	boolean		false
Connector	2	1		AvailabilityState		Actual	Occupied	optionList		false
Connector	2	1		ConnectorType		Actual	cCCS2	string		false
Connector	2	1		SupplyPhases		Actual	0	integer		false
Connector	2	2		Available		Actual	true	boolean		false
Connector	2	2		AvailabilityState		Actual	Unavailable	optionList		false
Connector	2	2		ConnectorType		Actual	cG105	string		false
Connector	2	2		SupplyPhases		Actual	0	integer		false

NOTE

An instance name has been given to the EVSEs in this example. This is to illustrate that it is allowed to provide an instance name even if only one instance of the component exists. It is not required to do so.
The variable Voltage of ChargingStation has been added to show an example of a multi-instance variable.
Not all VariableAttributes and VariableCharacteristics are shown in the table.

2.14. ISO 15118 Related

2.14.1. CentralContractValidationAllowed

Required	no		
Component	**componentName**	ISO15118Ctrlr	
Variable	**variableName**	CentralContractValidationAllowed	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable exists and has the value *true*, then Charging Station can provide a contract certificate that it cannot validate, to the CSMS for validation as part of the AuthorizeRequest.		

2.14.2. ContractValidationOffline

Required	yes	
Component	**componentName**	ISO15118Ctrlr

Variable	**variableName**	ContractValidationOffline	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable is *true*, then Charging Station will try to validate a contract certificate when it is offline.		

2.14.3. ProtocolSupportedByEV

Required	no		
Component	**componentName**	ConnectedEV	
	evse	*	
Variable	**variableName**	ProtocolSupportedByEV	
	variableInstance	<Priority>	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	A string with the following comma-separated items: "<uri>,<major>,<minor>". This is information from the supportedAppProtocolReq message from ISO 15118. Each priority is given its own variable instance. Example: "urn:iso:15118:2:2013:MsgDef,2,0"		

2.14.4. ProtocolAgreed

Required	no		
Component	**componentName**	ConnectedEV	
	evse	*	
Variable	**variableName**	ProtocolAgreed	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	A string with the following comma-separated items: "<uri>,<major>,<minor>". This is the protocol uri and version information that was agreed upon between EV and EVSE in the supportedAppProtocolReq handshake from ISO 15118. Example: "urn:iso:15118:2:2013:MsgDef,2,0"		

2.14.5. ISO15118PnCEnabled

Required	no		
Component	**componentName**	ISO15118Ctrlr	
Variable	**variableName**	PnCEnabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable is *true*, then ISO 15118 plug and charge as described by use case C07 - Authorization using Contract Certificates is enabled. If this variable is *false*, then ISO 15118 plug and charge as described by use case C07 - Authorization using Contract Certificates is disabled.		

2.14.6. ISO15118V2GCertificateInstallationEnabled

Required	no		
Component	**componentName**	ISO15118Ctrlr	

Variable	**variableName**	V2GCertificateInstallationEnabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable is *true*, then ISO 15118 V2G Charging Station certificate installation as described by use case A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station is enabled. If this variable is *false*, then ISO 15118 V2G Charging Station certificate installation as described by use case A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station is disabled.		

2.14.7. ISO15118ContractCertificateInstallationEnabled

Required	no		
Component	**componentName**	ISO15118Ctrlr	
Variable	**variableName**	ContractCertificateInstallationEnabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable is *true*, then ISO 15118 contract certificate installation/update as described by use case M01 - Certificate installation EV and M02 - Certificate Update EV is enabled. If this variable is *false*, then ISO 15118 contract certificate installation/update as described by use case M01 - Certificate installation EV and M02 - Certificate Update EV is disabled.		

2.14.8. ISO15118RequestMeteringReceipt

Required	no		
Component	**componentName**	ISO15118Ctrlr	
Variable	**variableName**	RequestMeteringReceipt	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable is *true*, then Charging Station shall request a metering receipt from EV before sending a fiscal meter value to CSMS.		

2.14.9. ISO15118SeccId

Required	no		
Component	**componentName**	ISO15118Ctrlr	
	evse	* (optional)	
Variable	**variableName**	SeccId	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	The name of the SECC in the string format as required by ISO 15118. It is used as the **commonName** (CN) of the SECC leaf certificate. Example: "DE-ICE-S-0003C4D5578786756453309675436-2"		

2.14.10. ISO15118CountryName

Required	no		
Component	**componentName**	ISO15118Ctrlr	
	evse	* (optional)	
Variable	**variableName**	CountryName	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string

Description	The countryName of the SECC in the ISO 3166-1 format. It is used as the **countryName** (C) of the SECC leaf certificate. Example: "DE"

2.14.11. ISO15118OrganizationName

Required	no		
Component	**componentName**	ISO15118Ctrlr	
	evse	* (optional)	
Variable	**variableName**	OrganizationName	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	The organizationName of the CSO operating the charging station. It is used as the **organizationName** (O) of the SECC leaf certificate. Example: "John Doe Charging Services Ltd" Note: This value will usually be identical to SecurityCtrlr.OrganizationName, but it does not have to be.		

2.14.12. ISO15118EvseId

Required	no		
Component	**componentName**	EVSE	
	evse	*	
Variable	**variableName**	ISO15118EvseId	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	The name of the EVSE in the string format as required by ISO 15118 and IEC 63119-2. Example: "DE*ICE*E*1234567890*1"		