Old text	RemoteStopTransaction.req to Charge Point with the identifier of the transaction. Charge Point SHALL reply with RemoteStopTransaction.conf **to indicate whether it is indeed able to stop the transaction.**
New text	RemoteStopTransaction.req to Charge Point with the identifier of the transaction. Charge Point SHALL reply with RemoteStopTransaction.conf **and a status indicating whether it has accepted the request and a transaction with the given transactionId is ongoing and will be stopped.**

3.34. page 54, section: 5.16. Not defined how a Charge Point should process a Charging Profile with more phases then being used.

Added in errata sheet v4.0

Additional Text	When the Charge Point receives a Charging Profile with 'numberPhases' higher then the currently used amount of phases or maximum amount of phases this Charge Point can use, The Charge Point SHALL use the 'limit' as is given, for the amount of phases that are/can be used for charging. The Charge Point SHALL NOT calculate a new 'limit' based on the amount of phases possible and given in the 'numberPhases' field.

3.35. page 55, section: 5.17. TriggerMessage(BootNotification) after being accepted may be rejected

Added in errata sheet v4.0

The following scenario is not defined in OCPP 1.6: "When the Charge Point is accepted by the Central System and the Central System sends a TriggerMessage for requesting a BootNotification. The Charge Point will accept this TriggerMessage and sends a BootNotification.req to the Central System. What should be the behavior of the Charge Point when the Central System is responding with the status Pending or Rejected?"

As this can cause a lot of confusing situations, what to do with ongoing transaction etc. It is allowed to reject such a TriggerMessage.req.

Additional text	After the Charge Point has received a BootNotification.conf(Accepted), until the next reset/reboot/reconnect, the Charge Point is RECOMMENDED to Reject a TriggerMessage request for BootNotification.

3.36. Page 56, section: 5.14: Improved description of Soft/Hard Reset

Added in errata sheet v3.0

The descriptions of Soft/Hard reset can be improved, not very clear what is the difference between the two.

Additional text, between par 1 & 2	After receipt of a Reset.req, The Charge Point SHALL send a StopTransaction.req for any ongoing transaction before performing the reset. If the Charge Point fails to receive a StopTransaction.conf from the Central System, it SHALL queue the StopTransaction.req.

Old text	At receipt of a soft reset, the Charge Point SHALL return to a state that behaves as just having been booted. If any transaction is in progress it SHALL be terminated normally, before the reset, as in Stop Transaction.

New text	At receipt of a soft reset request, the Charge Point SHALL stop ongoing transactions gracefully and send StopTransaction.req for every ongoing transaction. It should then restart the application software (if possible, otherwise restart the processor/controller).

Old text	At receipt of a hard reset the Charge Point SHALL attempt to terminate any transaction in progress normally as in StopTransaction and then perform a reboot.
New text	At receipt of a hard reset request, the Charge Point SHALL restart (all) the hardware, it is not required to gracefully stop ongoing transaction. If possible the Charge Point sends a StopTransaction.req for previously ongoing transactions after having restarted and having been accepted by the Central System via a BootNotification.conf. This is a last resort solution for a not correctly functioning Charge Point, by sending a "hard" reset, (queued) information might get lost.

3.37. Page 53, section: 5.16: Unclear ChargingProfiles should persist reboots

Added in errata sheet v4.0

There is no text that explain that it is expected of a Charge Point to store ChargingProfiles in persistent memory.

Additional text	ChargingProfiles set via SetChargingProfile.req SHALL be persistent across reboots/power cycles.

3.38. Page 55, section 5.16: Missing requirement about ChargingSchedule’s ChargingRateUnit

Added in errata sheet v4.0

The following note must be added at the end of this section:

> If an invalid value is used for an enumeration in ChargingProfile or ChargingSchedule, e.g. when using a value for chargingRateUnit that is not 'A' or 'W', then Charge Point SHALL respond with RPC Framework CALLERROR: PropertyConstraintViolation (JSON) or SOAP Fault: Sender, ProtocolError (SOAP).

3.39. Page 55, section: 5.16: SetChargingProfile may be rejected

Added in errata sheet v4.0

The following notes must be added at the end of this section:

> If an invalid value for connectorId is used in SetChargingProfile.req, then the Charge Point SHALL respond with RPC Framework CALLERROR: PropertyConstraintViolation (JSON) or SOAP Fault: Sender, ProtocolError (SOAP).

> If the Charge Point does not support smart charging, then it SHALL respond with RPC Framework CALLERROR: NotSupported (JSON) or SOAP Fault: Receiver, NotSupported (SOAP).

3.40. Page 56, section: 5.18: TriggerMessage(BootNotification) not allowed after BootNotificationResponse(Accepted)

Added in errata sheet v4.0

Add the following note at the end of this section:

> ℹ Once a CSMS has sent a BootNotification.conf message with status registrationStatus = Accepted to the Charge Point, then CSMS SHALL not send a TriggerMessage to request for a new BootNotification until the Charge Point sends a BootNotification.req message.

3.41. Page 58, section: 5.16.2: RemoteStart with ChargingProfile: TransactionId should not be set

It is not clear that in a remoteStartTransaction.req with ChargingProfile, the transactionId should not be set.

Old text	If the Central System includes a ChargingProfile, the ChargingProfilePurpose MUST be set to TxProfile.
New text	If the Central System includes a ChargingProfile, the ChargingProfilePurpose MUST be set to TxProfile **and the transactionId SHALL NOT be set**.

3.42. Page 58, section: 5.16.2: Meaning of note on RemoteStart with ChargingProfile is not clear

In the description of RemoteStartTransaction with a ChargingProfile there is a note, but the meaning of the note is not clear.

Old text	The Charge Point SHOULD add the TransactionId to the received profile once the transaction is reported to the central system.
New text	The Charge Point SHALL apply the given profile to the newly started transaction. This transaction will get a transactionId assigned by Central System via a startTransaction.conf. When the Charge Point receives a setChargingProfile.req, with the transactionId for this transaction, with the same StackLevel as the profile given in the remoteStartTransaction.req, the Charge Point SHALL replace the existing charging profile, otherwise it SHALL install/stack the profile next to the already existing profile(s).

3.43. Page 58, Section 5.16.4: Smart Charging fall back to default unclear

Added in errata sheet v3.0

Old text	When recurrencyKind is used in combination with a chargingSchedule duration shorter than the recurrencyKind period, the Charge Point SHALL fall back to default behavior after the chargingSchedule duration ends.
New text	When recurrencyKind is used in combination with a chargingSchedule duration shorter than the recurrencyKind period, the Charge Point SHALL fall back to default behavior after the chargingSchedule duration ends. **This fall back means that the Charge Point SHALL use a ChargingProfile with a lower stackLevel if available. If no other ChargingProfile is available, the Charge Point SHALL allow charging as if no ChargingProfile is installed. If the chargingSchedulePeriod and/or duration is longer than the recurrencyKind period, the remainder periods SHALL NOT be executed.**

3.44. Page 58, Section 5.16.4: Not defined what to do with a charging schedule period longer then recurrence.

Added in errata sheet v3.0

It is not defined what to do with a chargingSchedulePeriod and/or duration that is longer than the recurrence period.

Add the following note after the NOTE about "chargingSchedulePeriod longer than duration"

New text	NOTE: When recurrencyKind is used in combination with a chargingSchedulePeriod and/or duration that is longer than the recurrence period duration, the remainder periods SHALL NOT be executed.

3.45. Page 58, Section 5.16.4: First ChargingSchedulePeriod should start with StartSchedule = 0

Added in errata sheet v3.0

The obvious is not defined: The first ChargingSchedulePeriod StartSchedule in a ChargingSchedule should be 0.

Add the following note after the NOTE about "chargingSchedulePeriod longer than duration"

New text	NOTE: The startPeriod of the first ChargingSchedulePeriod in a ChargingSchedule SHALL always be 0.

3.46. Page 60, section: 5.17: Description TriggerMessage for MeterValues not clear

The description of what a Charge Point should do when it receives a TriggerMessage.req PDU with requestedMessage: MeterValues, is not clear.

Old text	A MeterValues message triggered in this way for instance **SHOULD** return the most recent measurements for all measurands configured in configuration key MeterValuesSampledData.
New text	A MeterValues message triggered in this way for instance **SHALL** return the most recent measurements for all measurands configured in configuration key MeterValuesSampledData.

3.47. Page 61, section: 5.19: Relation between FirmwareUpdate.req and FirmwareStatusNotification.req is missing.

There is no description about the relation between FirmwareUpdate.req and FirmwareStatusNotification.req in the specification. In paragraph 3.3 on page 15, there is a more elaborate diagram of firmware update, including the relationship between UpdateFirmware.req and FirmwareStatusNotification.req, but that paragraph is informative.

The following diagram should replace the diagram in 5.19.

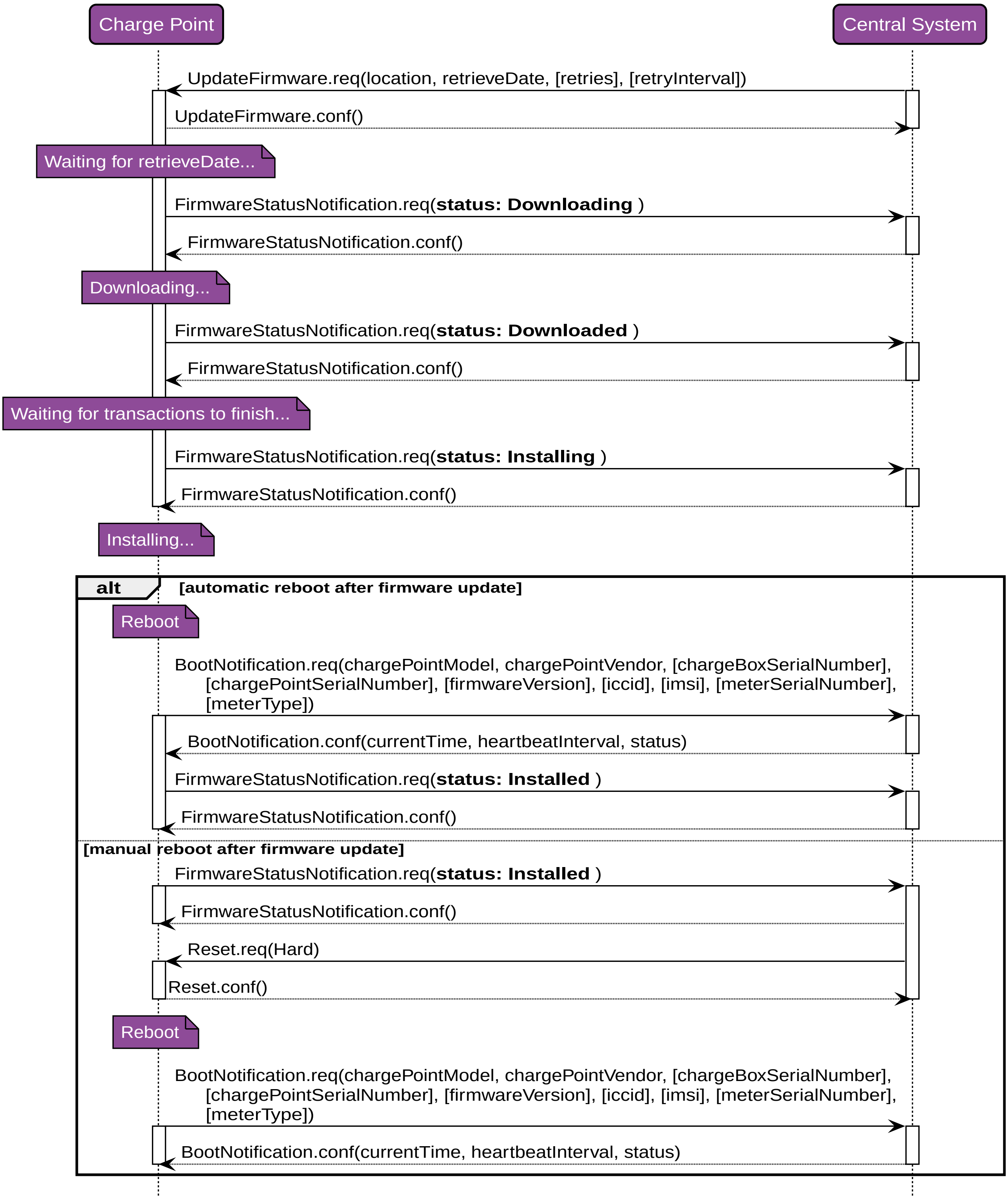
Charge Point
Central System
UpdateFirmware.req(location, retrieveDate, [retries], [retryInterval])
UpdateFirmware.conf()
Waiting for retrieveDate...
FirmwareStatusNotification.req(status: Downloading)
FirmwareStatusNotification.conf()
Downloading...
FirmwareStatusNotification.req(status: Downloaded)
FirmwareStatusNotification.conf()
Waiting for transactions to finish...
FirmwareStatusNotification.req(status: Installing)
FirmwareStatusNotification.conf()
Installing...
alt
[automatic reboot after firmware update]
Reboot
BootNotification.req(chargePointModel, chargePointVendor, [chargeBoxSerialNumber], [chargePointSerialNumber], [firmwareVersion], [iccid], [imsi], [meterSerialNumber], [meterType])
BootNotification.conf(currentTime, heartbeatInterval, status)
FirmwareStatusNotification.req(status: Installed)
FirmwareStatusNotification.conf()
[manual reboot after firmware update]
FirmwareStatusNotification.req(status: Installed)
FirmwareStatusNotification.conf()
Reset.req(Hard)
Reset.conf()
Reboot
BootNotification.req(chargePointModel, chargePointVendor, [chargeBoxSerialNumber], [chargePointSerialNumber], [firmwareVersion], [iccid], [imsi], [meterSerialNumber], [meterType])
BootNotification.conf(currentTime, heartbeatInterval, status)

Figure 2. Sequence Diagram: firmware update

Additional text	During downloading and installation of the firmware, the Charge Point MUST send FirmwareStatusNotification.req PDUs to keep the Central System updated with the status of the update process. The sequence diagram above is an example. It is good practice to first reboot the Charge Point to check the new firmware is booting and able to connect to the Central System, before sending the status: **Installed**. It is not a requirement.

3.48. Page 61, section: 5.19: No description new firmware should be installed.

There is no requirement on the installation of new firmware.

Additional text	The Charge Point SHALL, if the new firmware image is "valid", install the new firmware as soon as it is able to.

3.49. page 61, section: 5.19. Firmware installation during charging session

It is advised to not stop charging session to install new firmware, but wait until session has ended.

Additional text	If it is not possible to continue charging during installation of firmware, it is RECOMMENDED to wait until Charging Session has ended (Charge Point idle) before commencing installation. It is RECOMMENDED to set connectors that are not in use to UNAVAILABLE while the Charge Point waits for the Session to end.

This errata note is also applicable for OCPP versions 1.2 and 1.5.

3.50. Page 66, section: 6.25: Description of retries incorrect

Added in errata sheet v4.0

In the description of the field **retries** the word "try" should be "retry".

Old text	Optional. This specifies how many times Charge Point must try to upload the diagnostics before giving up. If this field is not present, it is left to Charge Point to decide how many times it wants to retry.
New text	Optional. This specifies how many times Charge Point must retry to upload the diagnostics before giving up. If this field is not present, it is left to Charge Point to decide how many times it wants to retry.

3.51. Page 70, section: 6.22: Unclear when to use the fields: scheduleStart and chargingSchedule in GetCompositeSchedule.conf.

The message: GetCompositeSchedule.conf contains 3 optional fields, for two of these fields: "scheduleStart" and "chargingSchedule" it is not clear when they should or should not be used:

Add to the description the field: scheduleStart	If status is "Rejected", this field may be absent.
Add to the description the field: chargingSchedule	If status is "Rejected", this field may be absent.

3.52. Page 70, section: 6.41: SendLocalList list version should never be 0

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 77

In GetLocalListVersion.conf listVersion = 0 and -1 have a special meaning, so they should not be used in SetLocalList.req

Old text	Required. In case of a full update this is the version number of the full list. In case of a differential update it is the version number of the list after the update has been applied.

New text	Required. In case of a full update this is the version number of the full list. In case of a differential update it is the version number of the list after the update has been applied. **SHALL NOT be -1 or 0 as these have a special meaning in GetLocalListVersion.conf**

3.53. Page 70, section: 6.43: Description connectorId = 0 in SetChargingProfile.req causes confusion

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 78

The description of the connectorId in the SetChargingProfile.req does not take into account that it can also be used for setting a TxDefaultProfile. It seems to only take into account a ChargePointMaxProfile.

Old text	If connectorId = 0, the message contains an overall limit for the Charge Point.
New text	If connectorId = 0, and the message contains a ChargePointMaxProfile it contains an overall limit for the Charge Point. If connectorId = 0, and the message contains a TxDefaultProfile it contains limits that are to be used for any new transaction on any connector of that Charge Point.

3.54. Page 71, section: 6.46: Useful values for Transaction ID not clear

Added in errata sheet v4.0

It is not clear to some that the expected value for Transaction IDs are unique positive integers.

Old description	Required. This contains the transaction id supplied by the Central System.
New description	Required. This contains the transaction id supplied by the Central System. It is RECOMMENDED to use unique positive numbers for transactionIds. Negative numbers and zero (0) are possible, but don't have any special meaning to the Charge Point (they don't mean the transaction is rejected or something like that.) Note that the Charge Point might use transactionId = -1 in transaction related message when the Charge Point was not able to successfully deliver the StartTransaction.req.

3.55. Page 77, section: 7.7: Definition of SuspendedEV is too confusing

Added in errata sheet v4.0

The new text for SuspendedEV is too confusing. The way it is written now implies that, even if the EVSE is in charging state (C2 with contactor closed) and the EV is not consuming any power that the state should be SuspendedEV.

SuspendedEV	Old text	When the EV is connected to the EVSE and the EVSE is offering energy but the EV is not taking any energy. (Operative)
SuspendedEV	New text	When the EV is connected to the EVSE and the EVSE is willing and ready to offer energy, but EV is not asking or taking energy. For example: A Charge Point using mode 3: if the EV is in B2 this also means SuspendedEV. (Operative)

3.56. Page 79, section: 7.8: recurrencyKind only to be used with chargingProfileKind: Recurring

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 90

There is no description/explanation that the field "recurrencyKind" should only be used when "chargingProfileKind" is "Recurring"

Old description	Optional. Indicates the start point of a recurrence.
New description	Optional. Indicates the start point of a recurrence. SHALL only be used when the field: recurrencyKind is set to: Recurring.

3.57. Page 80, section: 7.13: GetCompositeScheduleResponse unclear how/when to use which start field

Added in errata sheet v4.0

The description of the field **startSchedule** must be replaced by the following.

Old text	Optional. Starting point of an absolute schedule. If absent the schedule will be relative to start of charging.
New text	Optional. Starting point of an absolute schedule. If absent the schedule will be relative to start of charging. When ChargingSchedule is used as part of a GetCompositeSchedule.conf message, then this field must be omitted.

3.58. Page 81, section: 7.15: Improved definition of string types: CiString20Type.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 94

Description of: CiString20Type can now be read as: "A String that has to be exactly 20 characters long." This is a maximum length, not a required length.

Old text	Generic used case insensitive string of 20 characters.
New text	A case insensitive string with a maximum length of 20 characters.

3.59. Page 81, section: 7.16: Improved definition of string types: CiString25Type.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 94

Description of: CiString25Type can now be read as: "A String that has to be exactly 25 characters long." This is a maximum length, not a required length.

Old text	Generic used case insensitive string of 25 characters.
New text	A case insensitive string with a maximum length of 25 characters.

3.60. Page 82, section: 7.17: Improved definition of string types: CiString50Type.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 94

Description of: CiString50Type can now be read as: "A String that has to be exactly 50 characters long." This is a maximum length, not a required length.

Old text	Generic used case insensitive string of 50 characters.
New text	A case insensitive string with a maximum length of 50 characters.

3.61. Page 82, section: 7.18: Improved definition of string types: CiString255Type.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 95

Description of: CiString255Type can now be read as: "A String that has to be exactly 255 characters long." This is a maximum length, not a required length.

Old text	Generic used case insensitive string of 255 characters.
New text	A case insensitive string with a maximum length of 255 characters.

3.62. Page 82, section: 7.19: Improved definition of string types: CiString500Type.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 95

Description of: CiString500Type can now be read as: "A String that has to be exactly 500 characters long." This is a maximum length, not a required length.

Old text	Generic used case insensitive string of 500 characters.
New text	A case insensitive string with a maximum length of 500 characters.

3.63. Page 84, section: 6.55: Description for UpdateFirmware.req: retrieveDate is ambiguous

The description of the field: "retrieveDate" is ambiguous. It should not be "must", but: "is allowed to"

Old text	Required. This contains the date and time after which the Charge Point **must** retrieve the (new) firmware.
New text	Required. This contains the date and time after which the Charge Point **is allowed to** retrieve the (new) firmware.

3.64. Page 88, section: 7.7: Description SuspendedEVSE and SuspendedEV too strict, not all chargers have a contactor

Descriptions for SuspendedEVSE and SuspendedEV seems to imply that an EVSE has a contactor, but that is not always the case, for example: wireless charging.

SuspendedEVSE	Old text	When the contactor of a Connector opens upon request of the EVSE, e.g. due to a smart charging restriction or as the result of StartTransaction.conf indicating that charging is not allowed (Operative)
SuspendedEVSE	New text	When the EV is connected to the EVSE but the EVSE is not offering energy to the EV, e.g. due to a smart charging restriction, local supply power constraints, or as the result of StartTransaction.conf indicating that charging is not allowed etc. (Operative)
SuspendedEV	Old text	When the EVSE is ready to deliver energy but contactor is open, e.g. the EV is not ready.
SuspendedEV	New text	When the EV is connected to the EVSE and the EVSE is offering energy but the EV is not taking any energy. (Operative)

3.65. Page 90, section: 7.8: validFrom fields are allowed for TxProfiles.

In the class definition of ChargingProfile, the field: "validFrom", is defined as: "Not to be used when ChargingProfilePurpose is TxProfile." The specification denotes that the field ValidTo and ValidFrom are not to be used in combination with profiletype TxProfile. This note should have been deleted in the final version. With the decision to support stacking in combination with ProfileType TxProfile, the use of ValidFrom and ValidTo fields is unavoidable, since otherwise the profile with the highest StackLevel will be active until it is uninstalled.

Old text	Optional. Point in time at which the profile starts to be valid. If absent, the profile is valid as soon as it is received by the Charge Point. **Not to be used when ChargingProfilePurpose is TxProfile.**
New text	Optional. Point in time at which the profile starts to be valid. If absent, the profile is valid as soon as it is received by the Charge Point.

3.66. Page 91, section: 7.8: validTo fields are allowed for TxProfiles.

In the class definition of ChargingProfile, the field: "validTo", is defined as: "Not to be used when ChargingProfilePurpose is TxProfile." The specification denotes that the field ValidTo and ValidFrom are not to be used in combination with profiletype TxProfile. This note should have been deleted in the final version. With the decision to support stacking in combination with ProfileType TxProfile, the use of ValidFrom and ValidTo fields is unavoidable, since otherwise the profile with the highest profile will be active until it is uninstalled.

Old text	Optional. Point in time at which the profile stops to be valid. If absent, the profile is valid until it is replaced by another profile. **Not to be used when ChargingProfilePurpose is TxProfile.**
New text	Optional. Point in time at which the profile stops to be valid. If absent, the profile is valid until it is replaced by another profile.

3.67. Page 91, section: 7.10: Description of TxProfile/TxDefaultProfile in ChargingProfilePurposeType in relation with RemoteStartTransaction unclear

It is not completely clear what the correct ProfilePurpose should be in a remoteStartTransaction.req.

TxDefaultProfile	Old text	Default profile **to be used for new transactions**.
TxDefaultProfile	New text	Default profile **that can be configured in the Charge Point. When a new transaction is started, this profile SHALL be used, unless it was a transaction that was started by a remoteStartTransaction.req with a ChargeProfile that is accepted by the Charge Point**.
TxProfile	Old text	Profile with constraints to be imposed by the Charge Point on the current transaction. A profile with this purpose SHALL cease to be valid when the transaction terminates.
TxProfile	New text	Profile with constraints to be imposed by the Charge Point on the current transaction **, or on a new transaction when this is started via a RemoteStartTransaction.req with a ChargingProfile**. A profile with this purpose SHALL cease to be valid when the transaction terminates.

3.68. Page 91, section: 7.38: BootNotification Rejected because of unknown ID not logical for JSON

NEW: errata sheet v4.1

With OCPP-J an unknown Charge Point will not receive a BootNotification Rejected, but an HTTP 404, as specified in the OCPP-J specification (Section 3.2).

Therefor another example in the Rejected description is better.

Old text	Charge point is not accepted by Central System. This may happen when the **Charge Point id** is not known by Central System.
New text	Charge point is not accepted by Central System. This may happen when the **imsi** is not known by Central System.

3.69. Page 92, section: 7.12: ChargingRateUnit value descriptions need more clarification

Using a ChargingRateUnit W for AC charging is potentially very complicated, and, if used, the calculation is tricky The description of the values of the ChargingRateUnit should be improved.

VALUE	OLD DESCRIPTION	NEW DESCRIPTION
W	Watts (power).	Watts (power). This is the TOTAL allowed charging power. If used for AC Charging, the phase current should be calculated via: Current per phase = Power / (Line Voltage * Number of Phases). The "Line Voltage" used in the calculation is not the measured voltage, but the set voltage for the area (hence, 230 or 110 volt). The "Number of Phases" is the numberPhases from the ChargingSchedulePeriod. It is usually more convenient to use this for DC charging. Note that if numberPhases in a ChargingSchedulePeriod is absent, 3 SHALL be assumed.
A	Amperes (current).	Amperes (current). The amount of Ampere per phase, not the sum of all phases. It is usually more convenient to use this for AC charging.

3.70. Page 93, section: 7.13: First ChargingSchedulePeriod should start with StartSchedule = 0

Added in errata sheet v3.0

There is no requirement that explains the obvious: The first ChargingSchedulePeriod should start with StartSchedule = 0.

Updated description for the chargingSchedulePeriod field:

Old text	Required. List of ChargingSchedulePeriod elements defining maximum power or current usage over time.
New text	Required. List of ChargingSchedulePeriod elements defining maximum power or current usage over time. **The StartSchedule of the first ChargingSchedulePeriod SHALL always be 0.**

3.71. page 94, section: 7.14 ChargingSchedulePeriod Limit can also be in Watts

ChargingSchedulePeriod field: Limit can be in Watts or Ampere.

Old text	Required. Power limit during the schedule period, expressed in Amperes. Accepts at most one digit fraction (e.g. 8.1).
New text	Required. Charging rate limit during the schedule period, in the applicable chargingRateUnit, for example in Amperes or Watts. Accepts at most one digit fraction (e.g. 8.1).

3.72. page 94 - 95, section: 7.15 - 7.19 CiStringXXXType should be defined as type

The CiStringXXTypes are defined as Class, that contain a Field Name, but they should have been defined as Type (without Field Name)

Paragraphs effected:

- 7.15 CiString20Type
- 7.16 CiString25Type
- 7.17 CiString50Type
- 7.18 CiString255Type
- 7.19 CiString500Type

Old definition:

7.15 CiString20Type

Class

Generic used case insensitive string of 20 characters.

FIELD NAME	FIELD TYPE	DESCRIPTION
cistring20	CiString[20]	String is case insensitive.

New definition:

7.15 CiString20Type

Type

Generic used case insensitive string of 20 characters.

FIELD TYPE	DESCRIPTION
CiString[20]	String is case insensitive.

These Changes have no effect on the WSDL and JSON definitions, they are defined correct in WSDL and JSON Schemas.

3.73. Page 98, section: 7.27: Definition of IdTagInfo misses cardinality

Every class definition in OCPP 1.6 contains a column called: "Card." (cardinality), but this column is missing in the definition of IdTagInfo.

New definition:

7.27 IdTagInfo

Class

Contains status information about an identifier. It is returned in Authorize, Start Transaction and Stop Transaction responses.

If expiryDate is not given, the status has no end date.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
expiryDate	dateTime	0..1	Optional. This contains the date at which idTag should be removed from the Authorization Cache.
parentIdTag	IdToken	0..1	Optional. This contains the parent-identifier.
status	AuthorizationStatus	1..1	Required. This contains whether the idTag has been accepted or not by the Central System.

3.74. page 98, section: 7.28. idToken: idToken field is of wrong type

The field: idToken should have been of the type CiString20Type, it is case insensitive.

Old text	String[20]
New text	CiString20Type

To prevent interoperability issues: Do NOT update WSDL files!

Note: For future version: idToken was missed when added the CiStringXXTypes. Might be removed as type, and all field of the type idToken in other Classes will then be replaced by CiString20Type.

3.75. Page 98, section: 7.28: IdToken should be defined as type

The IdToken is defined as Class, that contain a Field Name, but they should have been defined as Type (without Field Name) The WSDL and JSON Schema are correct, was only wrong in the specification. (was also wrong in OCPP 1.5)

New definition:

7.28 IdToken

Type

Contains the identifier to use for authorization. It is a case insensitive string. In future releases this may become a complex type to support multiple forms of identifiers.

FIELD TYPE	DESCRIPTION
CiString20Type	IdToken is case insensitive.

3.76. Page 98, section: 9.1.5: Incorrect text about duration in MeterValue

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 111

There is some text in the description of the configuration key: ClockAlignedDataInterval, talking about MeterValue duration. There is no duration of a MeterValue. This is old text from OCPP 1.5, which was already incorrect in OCPP 1.5. This text should have been removed.

Ignore incorrect text:

"~~ and (optional) duration interval value, represented according to the ISO8601 standard~~"

3.77. Page 98, section: 9.1.6: Improved description configuration key: "ConnectionTimeOut"

Added in errata sheet v4.0

The description of the configuration key: "ConnectionTimeOut" can be improved even further

Old text	Interval **from beginning of status: 'Preparing'** until incipient session is automatically canceled, due to failure of EV driver to (correctly) insert the charging cable connector(s) into the appropriate socket(s). **The Charge Point SHALL go back to the original state, probably: 'Available'**
New text	Interval **from beginning of status: 'Preparing'** until incipient session is automatically canceled, due to failure of EV driver to (correctly) insert the charging cable connector(s) into the appropriate socket(s). **The Charge Point SHALL go back to the original state, typically: 'Available'**

3.78. Page 99, section: 7.31: No definition of .register and .interval.

The are a couple of measurands that are defined as .register or .interval. But there is no definition of what that means.

Old:

VALUE	DESCRIPTION
Energy.Active.Export.Register	Energy exported by EV (Wh or kWh)
Energy.Active.Import.Register	Energy imported by EV (Wh or kWh)