

Open Charge Point Protocol 1.6

edition 2 FINAL, 2017-09-28

Table of Contents

1. Scope	4
2. Terminology and Conventions	5
2.1. Conventions	5
2.2. Definitions	5
2.3. Abbreviations	6
2.4. References	7
3. Introduction	8
3.1. Edition 2	8
3.2. Document structure	8
3.3. Feature Profiles	8
3.4. General views of operation	10
3.5. Local Authorization & Offline Behavior	12
3.6. Transaction in relation to Energy Transfer Period	14
3.7. Transaction-related messages	17
3.8. Connector numbering	17
3.9. ID Tokens	18
3.10. Parent idTag	18
3.11. Reservations	18
3.12. Vendor-specific data transfer	18
3.13. Smart Charging	19
3.14. Time zones	26
3.15. Time notations	26
3.16. Metering Data	27
4. Operations Initiated by Charge Point	29
4.1. Authorize	29
4.2. Boot Notification	29
4.3. Data Transfer	30
4.4. Diagnostics Status Notification	30
4.5. Firmware Status Notification	31
4.6. Heartbeat	31
4.7. Meter Values	32
4.8. Start Transaction	33
4.9. Status Notification	33
4.10. Stop Transaction	37
5. Operations Initiated by Central System	39
5.1. Cancel Reservation	39
5.2. Change Availability	39
5.3. Change Configuration	39
5.4. Clear Cache	40
5.5. Clear Charging Profile	40
5.6. Data Transfer	41
5.7. Get Composite Schedule	41
5.8. Get Configuration	41
5.9. Get Diagnostics	42
5.10. Get Local List Version	42
5.11. Remote Start Transaction	43
5.12. Remote Stop Transaction	43
5.13. Reserve Now	44
5.14. Reset	45
5.15. Send Local List	45
5.16. Set Charging Profile	46
5.17. Trigger Message	47
5.18. Unlock Connector	48
5.19. Update Firmware	49
6. Messages	51
6.1. Authorize.req	51
6.2. Authorize.conf	51
6.3. BootNotification.req	51
6.4. BootNotification.conf	51

6.5. CancelReservation.req	52
6.6. CancelReservation.conf	52
6.7. ChangeAvailability.req	52
6.8. ChangeAvailability.conf	52
6.9. ChangeConfiguration.req	53
6.10. ChangeConfiguration.conf	53
6.11. ClearCache.req	53
6.12. ClearCache.conf	53
6.13. ClearChargingProfile.req	53
6.14. ClearChargingProfile.conf	54
6.15. DataTransfer.req	54
6.16. DataTransfer.conf	54
6.17. DiagnosticsStatusNotification.req	55
6.18. DiagnosticsStatusNotification.conf	55
6.19. FirmwareStatusNotification.req	55
6.20. FirmwareStatusNotification.conf	55
6.21. GetCompositeSchedule.req	55
6.22. GetCompositeSchedule.conf	55
6.23. GetConfiguration.req	56
6.24. GetConfiguration.conf	56
6.25. GetDiagnostics.req	56
6.26. GetDiagnostics.conf	57
6.27. GetLocalListVersion.req	57
6.28. GetLocalListVersion.conf	57
6.29. Heartbeat.req	57
6.30. Heartbeat.conf	57
6.31. MeterValues.req	57
6.32. MeterValues.conf	58
6.33. RemoteStartTransaction.req	58
6.34. RemoteStartTransaction.conf	58
6.35. RemoteStopTransaction.req	58
6.36. RemoteStopTransaction.conf	58
6.37. ReserveNow.req	59
6.38. ReserveNow.conf	59
6.39. Reset.req	59
6.40. Reset.conf	59
6.41. SendLocalList.req	60
6.42. SendLocalList.conf	60
6.43. SetChargingProfile.req	60
6.44. SetChargingProfile.conf	60
6.45. StartTransaction.req	61
6.46. StartTransaction.conf	61
6.47. StatusNotification.req	61
6.48. StatusNotification.conf	62
6.49. StopTransaction.req	62
6.50. StopTransaction.conf	62
6.51. TriggerMessage.req	62
6.52. TriggerMessage.conf	63
6.53. UnlockConnector.req	63
6.54. UnlockConnector.conf	63
6.55. UpdateFirmware.req	63
6.56. UpdateFirmware.conf	64
7. Types	65
7.1. AuthorizationData	65
7.2. AuthorizationStatus	65
7.3. AvailabilityStatus	65
7.4. AvailabilityType	65
7.5. CancelReservationStatus	66
7.6. ChargePointErrorCode	66
7.7. ChargePointStatus	67
7.8. ChargingProfile	67
7.9. ChargingProfileKindType	69

7.10. ChargingProfilePurposeType	69
7.11. ChargingProfileStatus	69
7.12. ChargingRateUnitType	69
7.13. ChargingSchedule	70
7.14. ChargingSchedulePeriod	70
7.15. CiString20Type	71
7.16. CiString25Type	71
7.17. CiString50Type	71
7.18. CiString255Type	71
7.19. CiString500Type	71
7.20. ClearCacheStatus	72
7.21. ClearChargingProfileStatus	72
7.22. ConfigurationStatus	72
7.23. DataTransferStatus	72
7.24. DiagnosticsStatus	73
7.25. FirmwareStatus	73
7.26. GetCompositeScheduleStatus	74
7.27. IdTagInfo	74
7.28. IdToken	74
7.29. KeyValue	74
7.30. Location	75
7.31. Measurand	75
7.32. MessageTrigger	76
7.33. MeterValue	77
7.34. Phase	77
7.35. ReadingContext	78
7.36. Reason	78
7.37. RecurrencyKindType	79
7.38. RegistrationStatus	79
7.39. RemoteStartStopStatus	79
7.40. ReservationStatus	80
7.41. ResetStatus	80
7.42. ResetType	80
7.43. SampledValue	80
7.44. TriggerMessageStatus	81
7.45. UnitOfMeasure	81
7.46. UnlockStatus	82
7.47. UpdateStatus	82
7.48. UpdateType	83
7.49. ValueFormat	83
8. Firmware and Diagnostics File Transfer	84
8.1. Download Firmware	84
8.2. Upload Diagnostics	84
9. Standard Configuration Key Names & Values	85
9.1. Core Profile	85
9.2. Local Auth List Management Profile	93
9.3. Reservation Profile	94
9.4. Smart Charging Profile	94
Appendix A: New in OCPP 1.6	96
A.1. Updated/New Messages	96

Charge Point와 Central System 간의 인터페이스 설명

문서 버전	1.6 edition 2
문서 상태	FINAL
문서 릴리스 날짜	2017-09-28

저작권 © 2010 – 2017 Open Charge Alliance. 모든 권리 보유.

이 문서는 *Creative Commons Attribution-NoDerivatives 4.0 International Public License* (<https://creativecommons.org/licenses/by-nd/4.0/legalcode>)에 따라 제공됩니다.

버전 이력

VERSION	DATE	AUTHOR	DESCRIPTION
1.6 edition 2	2017-09-28	Robert de Leeuw <i>IHomer</i> Brendan McMahon <i>ESB ecars</i> Klaas van Zuuren <i>ElaadNL</i>	OCPP 1.6 edition 2 최종 릴리스. 알려진 모든 오류(v3.0 포함)와 개선된 스타일이 포함되어 있습니다.
1.6	2015-10-08	Robert de Leeuw <i>IHomer</i> Reinier Lamers <i>The New Motion</i> Brendan McMahon <i>ESB ecars</i> Lambert Muhlenberg <i>Alfen</i> Patrick Rademakers <i>IHomer</i> Sergiu Tcaciuc <i>smartlab</i> Klaas van Zuuren <i>ElaadNL</i>	1.6 최종 릴리스. 1.5와 관련된 변경 사항은 부록 New in OCPP 1.6 을 참조하세요.
1.5	2012-06-01	Franc Buve	배포 가능한 버전의 스펙 문서입니다. 이 문서는 다음을 포함합니다: CR-01 Authentication/authorization lists CR-02 Interval meter readings CR-03 Charge point reservation CR-04 Generic data transfer CR-05 More detailed status notifications CR-06 Query configuration parameters CR-07 Timestamp in BootNotification mandatory CR-08 Response to StartTransaction.req with status other than Accepted is not clearly defined CR-09 Increase size of firmwareVersion in BootNotification
1.2	2011-02-21	Franc Buve	
1.0	2010-10-19	Franc Buve	e-laad.nl에서 승인한 최종 버전입니다. 버전 0.12와 동일한 버전입니다.

Chapter 1. Scope

이 문서는 **충전소(Charge Point)**와 **CSMS(Central System)** 간에 사용되는 프로토콜을 정의합니다. 프로토콜이 한쪽 또는 다른 쪽에서 특정 작업이나 응답을 요구하는 경우 이 문서에 명시됩니다.

본 스펙 문서는 통신 기술을 정의하지 않습니다. TCP/IP 연결을 지원하는 한 어떤 기술이든 가능합니다.

Chapter 2. Terminology and Conventions

2.1. Conventions

이 문서의 핵심 단어 "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", "OPTIONAL"은 [\[RFC2119\]](#)에 설명된 대로 해석해야 하며, 다음 추가 설명 조항을 따라야 합니다.

"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED"라는 용어 사용과 관련된 "특정 상황에서의 타당한 이유"라는 문구는 충전기 설계의 기능을 지원하는 데 필요한 하드웨어가 없는 것과 같은 기술적으로 타당한 이유를 의미하는 것으로 간주됩니다. 이 사양의 목적을 위해 구현 비용이나 사용 가능성과 같은 상업적 또는 기타 비기술적 이유로 내린 결정은 구체적으로 제외합니다.

"범위"와 "용어 및 규칙"을 제외한 모든 섹션과 부록은 명확하게 정보 제공 목적(informative)으로 표시된 경우가 아니면 규범적(normative)입니다.

2.2. Definitions

이 섹션에는 이 문서 전체에서 사용되는 용어가 포함되어 있습니다.

Central System	충전소 관리 시스템(Charge Point Management System, CSMS): 충전소를 관리, 사용자에게 충전기 사용을 허가하는 정보를 제공하는 CSMS입니다.
CiString	대소문자 구분 없는 문자열입니다. 인쇄 가능한 ASCII만 허용됩니다.
Charge Point	충전기는 전기차를 충전할 수 있는 물리적 시스템입니다. 충전기에는 하나 이상의 커넥터가 있습니다.
Charging Profile	다양한 유형의 프로파일에 사용되는 일반 충전 프로파일입니다. 프로파일에 대한 정보를 포함하고 충전 스케줄을 보관합니다. 향후 버전의 OCPP에서는 2개 이상의 충전 스케줄을 보관할 수 있습니다.
Charging Schedule	충전 프로파일의 일부입니다. 충전 전력 또는 전류 제한 블록을 정의합니다. 시작 시간과 길이를 포함할 수 있습니다.
Charging Session	충전 세션은 사용자 또는 EV와의 첫 번째 상호 작용이 발생할 때 시작됩니다. 이는 카드 스와이프, 원격 충전 시작, 케이블 및/또는 EV 연결, 주차 공간 점유 감지기 등일 수 있습니다.
Composite Charging Schedule	충전기에서 계산한 충전 스케줄입니다. 이는 충전기에 있는 모든 활성 스케줄과 가능한 지역 제한을 계산한 결과입니다. 지역 제한(Local Limits)이 고려될 수 있습니다.
Connector	이 사양에서 사용되는 용어 "커넥터"는 충전기에서 독립적으로 작동하고 관리되는 전기 콘센트를 말합니다. 이는 일반적으로 단일 물리적 커넥터에 해당하지만, 어떤 경우에는 단일 콘센트에 여러 물리적 소켓 유형 및/또는 고정 케이블/커넥터 배열이 있어 다양한 차량 유형(예: 4륜 EV 및 전기 스쿠터)에 이용할 수 있도록 합니다.
Control Pilot signal	충전기에서 EV에 최대 충전 전력 또는 전류 한도를 알리는 데 사용하는 신호로, [IEC61851-1] 에서 정의합니다.
Energy Offer Period	에너지 제공 기간은 EVSE가 에너지를 공급할 준비가 되었을 때 시작됩니다.
Energy Offer SuspendPeriod	트랜잭션 중에 EVSE에서 EV로 에너지 전달이 EVSE에 의해 중단되는 기간이 있을 수 있습니다(예: 스마트 충전 또는 로컬 밸런싱).
Energy Transfer Period	EV가 제공된 에너지를 받거나 반환하기로 선택하는 시간입니다. 한 트랜잭션 내에 여러 에너지 전송 기간이 존재할 수 있습니다.
Local Controller	스마트 충전 인프라의 옵션 장치입니다. 여러 개의 충전기가 실제로 연결된 곳에 위치합니다. 충전기와 CSMS 사이에 위치하며, OCPP 메시지를 수신하고 전송합니다. OCPP 스마트 충전 메시지를 사용하여 다른 충전기의 전력 또는 전류를 제어합니다. 충전기 자체가 로컬 컨트롤러가 될 수 있습니다.
OCPP-J	WebSocket을 통한 JSON 형식의 OCPP

OCPP-S	SOAP을 통한 OCPP
Phase Rotation	계량기(또는 없는 경우 그리드 연결)와 충전기 커넥터 사이의 위상 배선 순서를 정의합니다.
Transaction	모든 관련 전제 조건(예: 인증, 플러그 연결)이 충족되면 시작되고, 전제 조건을 만족하지 못하며, 이전 상태로 돌아가지 못하는 순간에 종료되는 충전 프로세스의 일부입니다.
String	대소문자를 구분하는 문자열입니다. 인쇄 가능한 ASCII만 허용됩니다. 직접적으로 명시되지 않는 한 메시지와 열거형(Enumeration)의 모든 문자열은 대소문자를 구분합니다.

2.3. Abbreviations

CSL	심표로 구분된 목록
CPO	충전소 운영자
DNS	도메인 이름 시스템
DST	일광 절약 시간제
EV	전기차, BEV(배터리 EV) 또는 PHEV(플러그인 하이브리드 EV)일 수 있음
EVSE	전기차 공급 장비(커넥터) [IEC61851-1]
FTP(S)	파일 전송 프로토콜(보안)
HTTP(S)	하이퍼텍스트 전송 프로토콜(보안)
ICCID	집적 회로 카드 식별자
IMSI	국제 모바일 가입 ID
JSON	JavaScript 객체 표기법
NAT	네이티브 주소 변환
PDU	프로토콜 데이터 단위
SC	스마트 충전
SOAP	Simple Object Access Protocol
URL	Uniform Resource Locator
RST	3상 전원 연결, 표준 참조 페이징
RTS	3상 전원 연결, 역방향 참조 페이징
SRT	3상 전원 연결, 역방향 240도 회전
STR	3상 전원 연결, 표준 120도 회전
TRS	3상 전원 연결, 표준 240도 회전

TSR	3상 전원 연결, 역방향 120도 회전
UTC	협정 세계시

2.4. References

[IEC61851-1]	"IEC 61851-1 2010: 전기차 전도성 충전 시스템 - 1부: 일반 요구 사항" https://webstore.iec.ch/publication/
[OCPP1.5]	"OCPP 1.5: Open Charge Proint Protocol 1.5" http://www.openchargealliance.org/downloads/
[OCPP_1.6CT]	"OCPP 1.6 준수 테스트" http://www.openchargealliance.org/downloads/
[OCPP_IMP_J]	"OCPP JSON 사양" http://www.openchargealliance.org/downloads/
[OCPP_IMP_S]	"OCPP SOAP 사양" http://www.openchargealliance.org/downloads/
[RFC2119]	"RFC에서 사용할 키워드 요구 사항 수준 표시". S. Bradner. 1997년 3월. http://www.ietf.org/rfc/rfc2119.txt

Chapter 3. Introduction

이 문서는 OCPP 버전 1.6의 사양 문서입니다.

OCPP는 충전소와 CSMS 간 통신을 위한 표준 개방 프로토콜이며 모든 유형의 충전 기술을 수용하도록 설계되었습니다.

OCPP 1.6은 시장에 맞게 새로운 기능을 도입했습니다. 새로운 기능에는 스마트 충전, 웹소켓을 통한 JSON을 사용하는 OCPP, 더 나은 진단 가능성(이유), 더 많은 충전소 상태 및 TriggerMessage가 포함됩니다. OCPP 1.6은 OCPP 1.5를 기반으로 하며, 몇 가지 새로운 기능과 많은 텍스트 개선, 설명 및 모호하게 작성된 내용에 대한 수정 사항이 있습니다. 개선 사항과 새로운 기능으로 인해 OCPP 1.6은 OCPP 1.5와 이전 버전과 호환되지 않습니다.

변경 사항의 전체 목록은 OCPP 1.6의 새로운 기능을 참조하세요.

이 장의 아래 섹션에서 몇 가지 기본 개념을 설명합니다. 충전기에 의해 통신이 시작된 경우 및 CSMS에 의해 통신이 시작된 경우 장은 프로토콜에서 지원하는 작업을 설명합니다. 정확한 메시지와 해당 매개변수는 메시지 장에서 자세히 설명하고 데이터 유형은 타입 장에서 설명합니다. 정의된 구성 키는 표준 설정 키 및 밸류 값 장에서 설명합니다.

3.1. Edition 2

이 문서는 OCPP 1.6 2판입니다. 이 문서는 여전히 동일한 프로토콜인 OCPP 1.6을 설명하며, 문서만 개선된 버전입니다. 메시지 수준에서는 2015년 10월의 OCPP 1.6 원본 릴리스와 비교하여 변경 사항이 없습니다. 알려진 모든 오류(이전에 발간된 별도 문서에 게시됨)가 이 문서에 병합되어 구현자가 사양을 사용하기가 더 쉬워졌습니다. OCPP 1.6을 구현하는 방법에 대해 의심이 드는 경우 이 문서가 원본 문서보다 우선합니다.

3.2. Document structure

OCPP 1.6이 도입되면서 두 가지 다른 유형의 OCPP가 있습니다. SOAP 기반 구현 외에도 훨씬 더 컴팩트한 JSON 대안을 사용할 수 있습니다. 구현 유형에 대한 의사 소통에서 혼란을 피하기 위해 JSON 또는 SOAP를 나타내는 고유한 접미사 -J 및 -S를 사용하는 것이 좋습니다. 일반적으로 JSON의 경우 OCPP-J, SOAP의 경우 OCPP-S입니다.

다양한 요구사항을 지원하기 위해 OCPP 표준은 여러 문서로 나뉩니다. 기본 문서(본 문서)에는 기술 프로토콜 사양이 포함되어 있습니다. 기술 프로토콜 사양은 전송 프로토콜 사양 중 하나와 함께 사용해야 합니다. OCPP SOAP 사양에는 OCPP-S 구현에 필요한 구현 사양이 포함되어 있습니다. OCPP-J의 경우 OCPP JSON 사양을 사용해야 합니다.

CSMS와 충전소 간의 상호 운용성을 개선하기 위해 OCPP 1.6 컴플라이언스 테스트 문서에 명시된 요구 사항을 충족하는 것이 좋습니다.

3.3. Feature Profiles

이 섹션은 규범적입니다.

OCPP 1.6에서 기능과 관련 메시지는 _프로파일_로 그룹화됩니다. 필요한 기능에 따라 구현자는 다음 프로파일 중 하나 이상을 구현하도록 선택할 수 있습니다.

프로파일 이름	설명
코어	펌웨어 업데이트, 로컬 권한 목록 관리 및 예약 지원 기능이 제거된 OCPP 1.5 [OCPP1.5]와 비슷한 기본 충전기 기능.
펌웨어 관리	펌웨어 업데이트 관리 및 진단 로그 파일 다운로드 지원.
로컬 권한 목록 관리	충전기에서 로컬 권한 목록을 관리하는 기능.
예약	충전기 예약 지원.
스마트 충전	기본 스마트 충전 지원(예: 컨트롤 파일럿 사용).

프로파일 이름	설명
원격 트리거	충전기가 CSMS로 메시지를 전송하는 이벤트 발생을 위한 원격 트리거 지원

이러한 프로파일은 고객이 OCPP 1.6 제품에 비즈니스 사례에 필요한 기능이 있는지 확인하는 데 사용할 수 있습니다. 규정 준수 테스트는 프로파일별로 제품이 OCPP 1.6 사양을 준수하는지 테스트합니다.

핵심 프로파일 구현은 필수입니다. 다른 프로파일은 선택 사항입니다.

Core, Firmware Management, Local Auth List Management 및 **Reservation** 프로파일이 구현되면 OCPP 1.5 [OCPP1.5]에서 지원하는 모든 기능이 포함됩니다.

프로파일의 모든 메시지 그룹화는 아래 표에서 찾을 수 있습니다.

메시지	코어	펌웨어 관리	로컬 인증 리스트 관리	원격 트리거	예약	스마트 차징
Authorize	X					
BootNotification	X					
ChangeAvailability	X					
ChangeConfiguration	X					
ClearCache	X					
DataTransfer	X					
GetConfiguration	X					
Heartbeat	X					
MeterValues	X					
RemoteStartTransaction	X					
RemoteStopTransaction	X					
Reset	X					
StartTransaction	X					
StatusNotification	X					
StopTransaction	X					
UnlockConnector	X					
GetDiagnostics		X				
DiagnosticsStatusNotification		X				
FirmwareStatusNotification		X				

메시지	코어	펌웨어 관리	로컬 인증 리스트 관리	원격 트리거	예약	스마트 차징
UpdateFirmware		X				
GetLocalListVersion			X			
SendLocalList			X			
CancelReservation				X		
ReserveNow				X		
ClearChargingProfile					X	
GetCompositeSchedule					X	
SetChargingProfile					X	
TriggerMessage						X

특정 기능 프로파일에 대한 지원은 **SupportedFeatureProfiles**의 구성 키(configuration key)의 응답에서 확인할 수 있습니다.

3.4. General views of operation

이 섹션은 정보 제공용입니다.

다음 그림은 두 가지 경우에 대한 Charge Point와 Central System 간의 작업에 대한 일반적인 보기를 설명합니다.

1. Charge Point가 카드 인증을 요청하고 충전 트랜잭션 상태를 전송
2. Central System이 Charge Point에 펌웨어 업데이트를 요청

다음 그림의 화살표 레이블은 작업 호출 중에 교환된 PDU를 나타냅니다. 이러한 PDU는 **Messages** 섹션에서 자세히 정의됩니다.

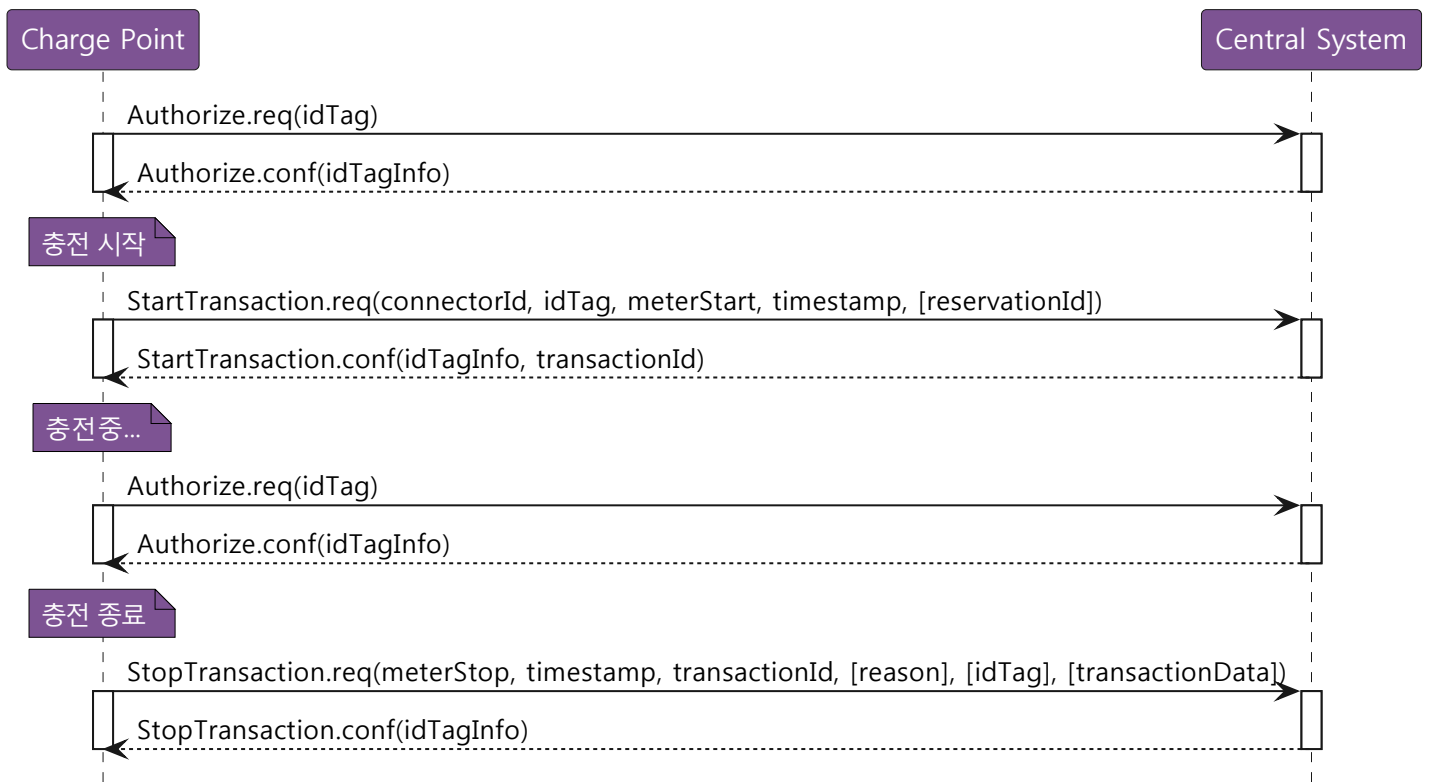


Figure 1. 시퀀스 다이어그램: 트랜잭션 시작 및 종지의 예

충전기에서 전기차를 충전해야 하는 경우, 충전을 시작하기 전에 먼저 사용자를 인증해야 합니다. 사용자 인증이 완료되면 충전소는 CSMS에 충전을 시작했다고 알립니다.

사용자가 충전기에서 전기차를 분리하려는 경우 충전기는 사용자가 충전을 시작한 사용자인지 또는 사용자가 동일한 그룹에 속하여 충전을 종료할 수 있는지 확인해야 합니다. 승인되면 충전기는 CSMS에 충전이 중지되었다고 알립니다.



충전기는 트랜잭션을 종료하기 전에 제시된 idTag가 트랜잭션을 시작하기 위해 제시된 idTag와 동일한 경우에는 Authorize.req를 보내서는 안 됩니다.

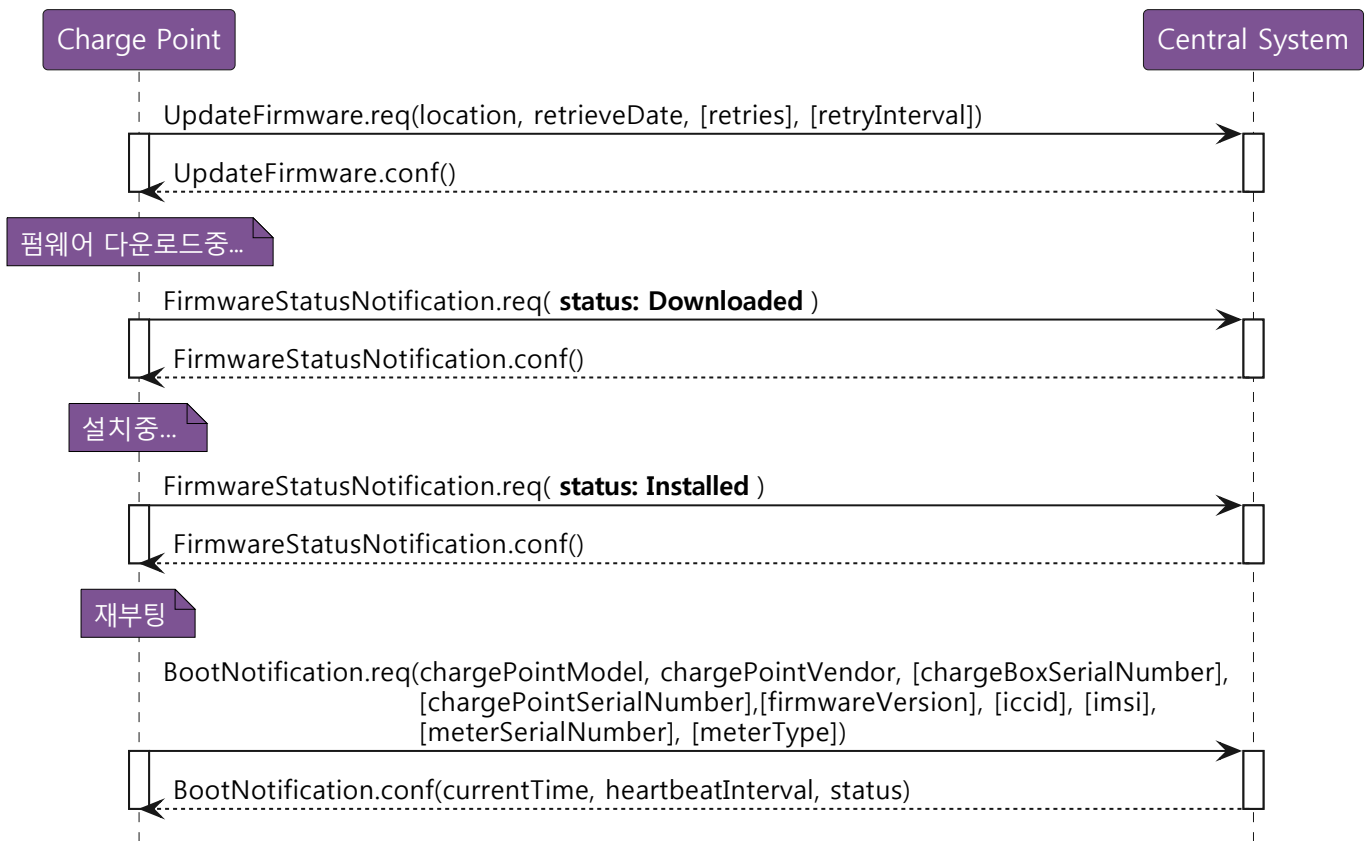


Figure 2. 시퀀스 다이어그램: 펌웨어 업데이트의 예

충전기가 새 펌웨어로 업데이트되어야 하는 경우, CSMS는 충전기가 새 펌웨어를 다운로드할 수 있는 시간을 충전기에 알립니다. 충전기는 새 펌웨어를 다운로드하고 설치하는 과정이 완료 될 때마다 이를 CSMS에 알려야 합니다.

3.5. Local Authorization & Offline Behavior

이 섹션은 규범적입니다.

통신 등으로 CSMS와 연결할 수 없는 경우 충전기는 독립적으로 작동하도록 설계되었습니다. 이러한 상황의 충전기 상태를 **오프라인** 상태라고 합니다.

사용자 경험을 개선하기 위해 충전기는 **Authorization Cache** 및/또는 **Local Authorization List**를 사용하여 식별자의 로컬 인증을 지원할 수 있습니다.

이를 통해 (a) **오프라인** 상태일 때 사용자를 인증하고 (b) 충전기와 CSMS 간의 통신이 느릴 때, 사용자 인증 요청에 대한 응답이 더 빠르게(빠르게 보이도록) 설정할 수 있습니다.

LocalAuthorizeOffline 구성 키는 충전기가 인증 캐시(Authorization Cache) 및/또는 로컬 인증 목록(Local Authorization List)을 사용하여 **오프라인** 상태일 때 사용자를 인증할지 여부를 제어합니다.

LocalPreAuthorize 구성 키는 충전기가 CSMS의 승인 응답을 기다리지 않고 트랜잭션을 시작하기 위해 인증 캐시 및/또는 로컬 인증 목록을 사용할지 여부를 제어합니다.

충전기는 로컬 승인 목록/승인 캐시 항목에서 명시적으로 승인할 수 없는 진짜 사용자에게 요금을 청구하는 것을 거부하지 않기 위해 **오프라인** 일 때 제시된 식별자의 (자동) 승인을 지원할 수 있습니다. 이 기능은 **Unknown Offline Authorization**에서 더 자세히 설명합니다.

3.5.1. Authorization Cache

충전기는 이전에 사용된 카드 중 CSMS로부터 인증 성공 응답을 받은 카드의 기록을 자율적으로 유지하는 **인증 캐시(Authorization Cache)**를 구현할 수 있습니다. (성공 인증을 받은 의미: idTag가 포함된 인증 성공 메시지에서 수신된 응답)

구현된 경우 인증 캐시는 다음 의미 체계를 준수해야 합니다.

- 캐시에는 모든 수신된 최신 식별자(즉, 유효 및 유효하지 않음)가 포함됩니다.
- 캐시는 수신된 모든 `IdTagInfo`(`Authorize.conf`, `StartTransaction.conf` 및 `StopTransaction.conf`에서 수신됨)를 사용하여 업데이트됩니다.
- 캐시 항목의 유효성이 만료되면 캐시에서 만료로 변경해야 합니다.
- 캐시의 식별자에 대해 `IdTagInfo`가 수신되면 업데이트해야 합니다.
- 새 인증 데이터가 수신되고 인증 캐시가 가득 찬 경우 충전기는 유효하지 않은 모든 항목을 제거한 다음, 필요한 경우 가장 오래된 유효 항목을 제거하여 새 항목을 위한 공간을 확보해야 합니다.
- 캐시 값은 비휘발성 메모리에 저장해야 하며 재부팅 및 정전 시에도 유지해야 합니다.
- 캐시에 유효하지 않은 것으로 저장된 식별자로 인증을 시도할 때, 충전기가 온라인 인 경우 `Authorize.req`를 CSMS로 보내 식별자의 현재 상태를 확인해야 합니다.

인증 캐시가 있는 경우의 작동은 `AuthorizationCacheEnabled` 구성 키에 의해 보고되고(가능한 경우 제어됨) 제어됩니다.

3.5.2. Local Authorization List

로컬 인증 목록(Local Authorization List)은 CSMS와 동기화할 수 있는 식별자 목록입니다.

목록에는 모든(또는 일부) 식별자의 권한 상태와 권한 상태/만료 날짜가 포함됩니다.

로컬 인증 목록의 식별자는 **유효(valid)**, **만료(expired)**, **(일시적으로) 차단됨(temporarily blocked)** 또는 **블랙리스트(blacklisted)**로 표시될 수 있으며, 각각 `IdTagInfo` 상태 값 `Accepted / ConcurrentTx`, `Expired`, `Blocked` 및 `Invalid`에 해당합니다.

이러한 값은 로컬 인증 과정에서 사용자에게 보다 세부적인 정보(예: 표시 메시지)를 제공하는 데 사용될 수 있습니다.

로컬 인증 목록은 충전기에서 비휘발성 메모리에 유지 관리해야 하며 재부팅 및 정전 시에도 목록이 유지되어야 합니다.

로컬 인증 목록을 지원하는 충전기는 `LocalAuthListMaxLength` 구성 키를 구현해야 합니다. 이를 통해 CSMS는 충전기에서 가능한 최대 로컬 인증 목록 요소 수를 알 수 있습니다.

충전기는 로컬 인증 목록이 `SupportedFeatureProfiles` 구성 키 값의 `LocalAuthListManagement` 요소의 존재 또는 부재에 따라 지원되는지 여부를 나타냅니다.

로컬 인증 목록이 활성화되었는지 여부는 `LocalAuthListEnabled` 구성 키에 의해 보고되고 제어됩니다.

CSMS는 (1) 로컬 인증 목록을 대체할 식별자의 전체 목록을 보내거나 (2) 로컬 인증 목록에 적용할 변경 사항(추가, 업데이트, 삭제) 목록을 보내 이 목록을 동기화할 수 있습니다. 이를 지원하는 작업은 `Get Local List Version` 및 `Send Local List`입니다.

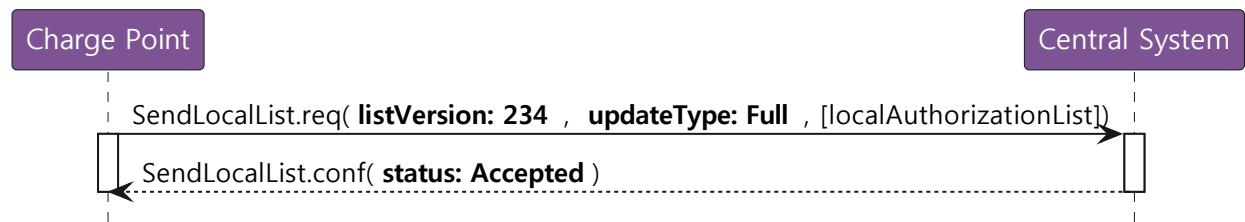


Figure 3. 시퀀스 다이어그램: 전체 로컬 인증 목록 업데이트의 예

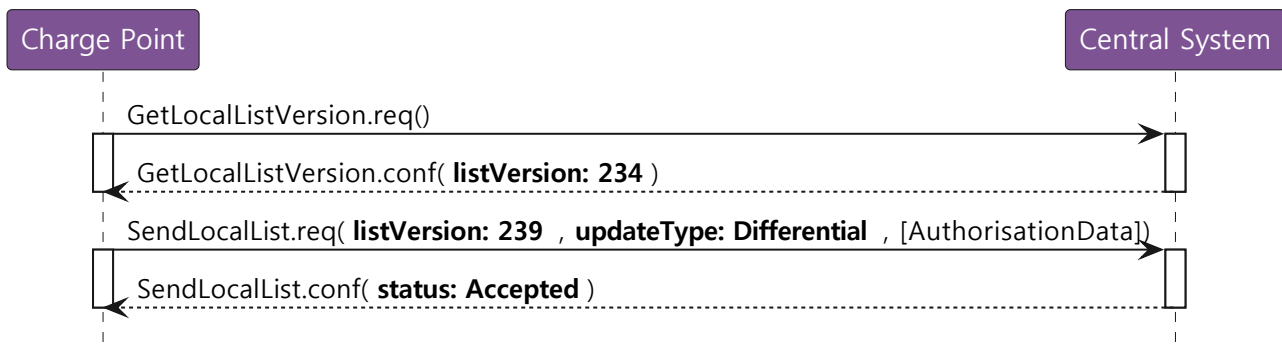


Figure 4. 시퀀스 다이어그램: 차등 로컬 인증 목록 업데이트의 예

충전소는 CSMS에서 SendLocalList PDU를 수신하는 경우를 제외하고는 다른 방법으로 인증 목록의 내용을 수정해서는 안 됩니다.



예를 들어 `StartTransaction.conf` 메시지에서 보고된 로컬 인증 목록과 유효성 간의 충돌이 발생할 수 있습니다. 이런 경우 충전소는 ConnectorId를 0으로 설정하고 ErrorCode를 'LocalListConflict'로 설정한 `StatusNotification`을 보내 CSMS에 알려야 합니다.

3.5.3. Relation between Authorization Cache and Local Authorization List

인증 캐시와 로컬 인증 목록은 별개의 논리적 데이터 구조입니다. 로컬 인증 목록에 알려진 식별자는 인증 캐시에 추가되어서는 안 됩니다.

인증 캐시와 로컬 인증 목록이 모두 지원되는 경우, 충전기는 로컬 인증 목록 항목을 동일한 식별자에 대한 인증 캐시 항목보다 우선시해야 합니다.

3.5.4. Unknown Offline Authorization

충전기가 오프라인 상태 일 때, 충전기는 로컬 인증 목록 또는 인증 캐시 항목에서 명시적으로 승인할 수 없는 "알 수 없는" 식별자의 자동 승인을 허용할 수 있습니다. 로컬 인증 목록에 있는 식별자 중 "허용됨(Allowed)"이 아닌 상태인 식별자(유효하지 않음(Invalid), 차단됨(Blocked), 만료됨(Expired))는 거부해야 합니다.

과거에는 유효했지만 유효 기간이 경과된 식별자도 거부해야 합니다.

지원되는 경우 알 수 없는 식별자에 대한 오프라인 인증(Unknown Offline Authorization) 기능의 작동은 `AllowOfflineTxForUnknownId` 구성 키에 의해 보고됩니다(가능한 경우 제어됨).

CSMS에 대한 연결이 복구되면 충전기는 `transaction-related message handling`에서 요구하는 대로 오프라인으로 승인된 모든 거래에 대해 `Start Transaction` 요청을 보내야 합니다. `StartTransaction.conf`의 승인 상태가 `Accepted` 가 아니고 트랜잭션이 아직 진행 중인 경우, 충전기는 다음을 수행해야 합니다.

- `StopTransactionOnInvalidId`가 `true` 로 설정된 경우: `Stop Transaction`에 명시된 대로 트랜잭션을 정상적으로 중지합니다. `Stop Transaction` 요청의 Reason 필드는 `DeAuthorized`로 설정해야 합니다. 충전기가 충전 케이블을 잠글 가능성이 있는 경우 소유자가 식별자를 제시할 때까지 충전 케이블을 잠금 상태로 유지해야 합니다.
- `StopTransactionOnInvalidId`가 `false` 로 설정된 경우: 차량으로의 에너지 공급만 중지합니다.



잘못된 식별자의 경우, 운영자는 EV를 충전 장소에서 이동시키기 위한 최소한의 에너지만 EV에 충전하게 선택할 수 있습니다. 이 양은 선택적 구성 키 `MaxEnergyOnInvalidId`에 의해 제어됩니다.

3.6. Transaction in relation to Energy Transfer Period

이 섹션은 정보 제공 용도입니다.

`Energy Transfer Period`는 EV와 EVSE 간에 에너지가 전송되는 기간입니다.

`Transaction` 중에 여러 에너지 전송 기간이 있을 수 있습니다.

여러 에너지 전송 기간은 다음 중 하나로 구분할 수 있습니다.

- EVSE가 에너지 전송을 제공하지 않는 EVSE에서 시작된 전송 중단
- EV가 EVSE에 전기적으로 연결된 상태에서 EV에 의해 시작된 전송 중단
- EV가 EVSE에 전기적으로 연결되지 않은 상태에서 EV에 의해 시작된 전송 중단.

CSMS는 트랜잭션 중에 전송된 MeterValues, 상태 알림(Status notifications)(충전중(Charging), SuspendedEV 및/또는 SuspendedEVSE 등)에서 에너지 전송 기간의 시작과 끝을 추론할 수 있습니다.

CSMS 구현은 일부 EV가 SuspendedEV 상태로 전환되지 않아 계속해서 충전이 진행될 수 있는 경우, 일부 충전기에는 계량기도 없는 경우 등과 같은 요소를 고려해야 합니다.

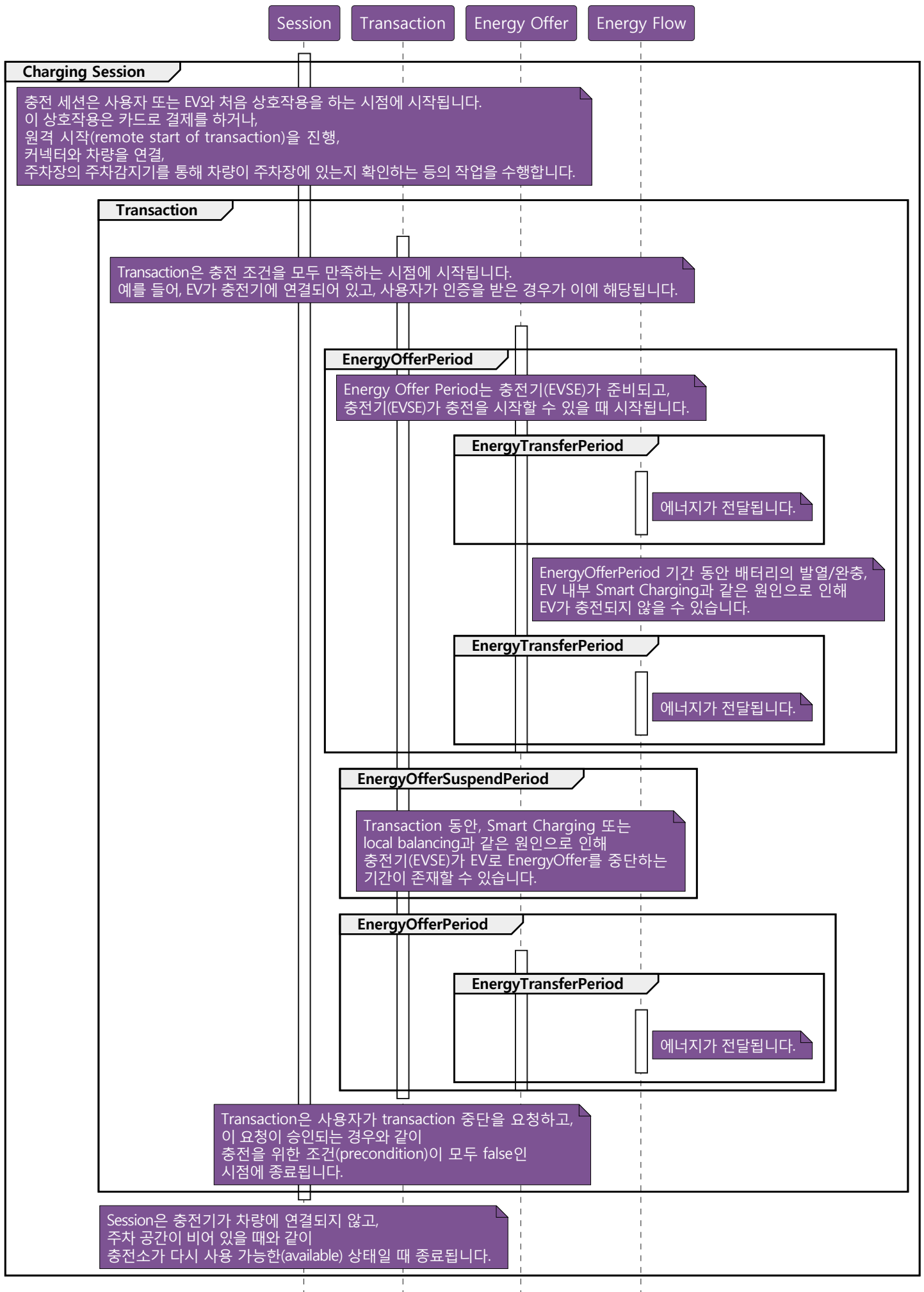


Figure 5. OCPP 충전 세션 및 트랜잭션 정의

3.7. Transaction-related messages

이 섹션은 규범적입니다.

충전기는 가능한 한 빨리 트랜잭션 관련 메시지를 시간순으로 CSMS에 전달해야 합니다. 트랜잭션 관련 메시지는 `StartTransaction.req`, `StopTransaction.req` 및 주기적 또는 시간 동기화가 된 `MeterValues.req` 메시지입니다.

오프라인 일 때 충전기는 충전기가 온라인 상태였다면 CSMS에 전송했을 트랜잭션 관련 메시지를 큐에 넣어야 합니다.

충전기는 CSMS에 전송하기 위해 큐에 넣은 트랜잭션 관련 메시지가 있는 경우, 트랜잭션과 관련이 없는 새 메시지는 큐가 비워질 때까지 기다리지 않고 즉시 전달할 수 있습니다. 따라서 트랜잭션 관련 메시지 대기열이 비워지기 전에, 예를 들어 인증 요청이나 알림 요청을 보낼 수 있습니다. 따라서, 고객이 기다리지 않고 긴급 알림이 지연되지 않을 수 있습니다.

새로운 트랜잭션 관련 메시지 전달은 대기열이 비워질 때까지 기다려야 합니다. 이는 트랜잭션 관련 메시지가 항상 시간순으로 전달되도록 하기 위한 것입니다.

CSMS가 충전기에 일정 기간 동안 큐에 대기한 트랜잭션 관련 메시지를 수신할 때, 다양한 타임스탬프가 상당히 과거라는 추론을 제외하고는 이것이 과거 메시지라는 것을 알지 못합니다. CSMS는 이러한 메시지를 다른 메시지와 동일하게 처리해야 합니다.

3.7.1. Error responses to transaction-related messages

충전기가 트랜잭션 관련 메시지를 건너뛸 수 있는 것은 유일하게 CSMS가 '메시지 처리 실패'를 반복적으로 보고하는 경우에만 허용됩니다. 모든 트랜잭션 관련 메시지를 시간순으로 전달해야 한다는 요구 사항은 소프트웨어 버그로 인해 CSMS가 충전기의 트랜잭션 관련 메시지 중 하나를 확인하지 못한 경우, 이후 충전기가 CSMS에 트랜잭션 관련 메시지를 전달할 수 없다는 것을 의미하기 때문에 이러한 규정이 필요합니다.

어떤 종류의 응답 또는 응답 실패가 '메시지 처리 실패'를 구성하는지는 [OCPP JSON Specification](#) 및 [OCPP SOAP Specification](#) 문서에 정의되어 있습니다.

충전기가 이러한 실패한 트랜잭션 관련 메시지를 재시도해야 하는 횟수와 간격은 `TransactionMessageAttempts` 및 `TransactionMessageRetryInterval` 구성 키를 사용하여 구성할 수 있습니다.

충전기에서 특정 트랜잭션 관련 메시지를 전달하는 데 첫 번째 실패가 발생하면, 메시지 처리 실패가 계속 발생하고 `TransactionMessageAttempts` 구성 키에 지정된 만큼의 메시지 처리 실패가 발생하기 전까지는 이 메시지를 다시 전송해야 합니다. 모든 재전송 전에 `TransactionMessageRetryInterval` 키에 지정된 시간(이 동일한 메시지의 이전 전송 횟수에 곱함)만큼 기다려야 합니다.

예를 들어, `TransactionMessageAttempts` 구성 키에 "3" 값이 있고 `TransactionMessageRetryInterval` 구성 키에 "60" 값이 있는 충전기가 있습니다. `StopTransaction` 메시지를 보내고 CSMS에서 메시지를 처리하는 데 실패한 것을 감지합니다. 충전기는 60초 동안 기다렸다가 메시지를 다시 보내야 합니다. 두 번째 실패가 발생하는 경우, 충전기는 메시지를 다시 보내기 전에 120초 동안 기다려야 합니다. 이 마지막 시도가 실패하면, 충전기는 메시지를 삭제하고 다음 트랜잭션 관련 메시지가 있는 경우 계속 진행해야 합니다.

3.8. Connector numbering

이 섹션은 규범적입니다.

CSMS에서 충전기의 모든 커넥터를 처리할 수 있도록 하려면 `ConnectorId`는 항상 동일한 방식으로 번호가 매겨져야 합니다.

커넥터 번호 매기기(`ConnectorId`)는 다음과 같아야 합니다.

- 첫 번째 커넥터의 ID는 1이어야 합니다.
- 추가 커넥터는 순차적으로 번호가 매겨져야 합니다(번호를 건너뛸 수 없음).
- `ConnectorId`는 충전기의 총 커넥터 수보다 절대로 큰 값을 가질 수 없습니다.
- CSMS에서 시작한 작업의 경우 `ConnectorId` 0은 충전기 전체의 커넥터를 처리하기 위해 예약되어 있습니다.
- 충전기에서 시작한 작업(보고 시)의 경우 `ConnectorId` 0은 충전기 메인 컨트롤러에 예약되어 있습니다.

예: 커넥터가 3개인 충전기: 모든 커넥터는 ID 1, 2, 3으로 번호가 매겨져야 합니다. 충전기의 커넥터는 왼쪽에서 오른쪽으로, 위에서 아래로 증가하는 논리적인 방식으로 번호를 매기는 것이 좋습니다.

3.9. ID Tokens

이 섹션은 규범적입니다.

대부분의 경우 로컬 토큰 리더 하드웨어를 통해 수집된 **IdToken** 데이터는 일반적으로 8/14 16진수 문자로 표현되는 실제 RFID 카드의 (4 또는 7바이트) UID 값입니다.

그러나 원격으로 시작된 충전 세션을 위해 CSMS에서 충전기로 전송된 **IdTokens**는 일반적으로 (일회용) 가상 트랜잭션 승인 코드이거나 실제 UID 값과의 충돌 가능성을 피하기 위해 의도적으로 비표준 UID 형식을 사용하는 가상 RFID 토큰일 수 있습니다.

또한 **ParentIds**로 사용되는 **IdToken** 데이터는 계정의 첫 번째/마스터 RFID 카드의 UID 대신 **ParentId**에 대한 공유 중앙 계정 식별자를 사용할 수 있습니다.

따라서 **IdToken** 클래스의 메시지 데이터 요소(**ParentId** 포함)는 데이터 유형(**CiString20Type**)의 제약 조건에 따라 CSMS에 의미 있는 모든 데이터를 포함할 수 있습니다(예: 충전 활동의 개시자를 식별하는 목적). 그리고 충전기는 이러한 데이터의 형식이나 내용에 대해 어떠한 가정도 해서는 안 됩니다(예: UID와 유사한 값은 반드시 16진수 문자만 있거나/또는 짝수 숫자만 있어야 한다는 가정).



상호 운용성을 촉진하기 위해 물리적 ISO 14443 호환 RFID 카드 UID를 나타내는 **IdToken** 데이터의 경우 현재까지의 일반적인 관행에 따라, 이러한 UID는 UID 바이트의 16진수 표현으로 표현되는 것이 좋습니다. ISO14443-3에 따르면, 16진수 문자열에서 바이트 0이 먼저 나와야 합니다.

3.10. Parent idTag

이 섹션은 규범적입니다.

CSMS는 일련의 ID 토큰을 "그룹"으로 처리할 수 있는 기능을 갖추고 있어 그룹 내의 모든 토큰이 트랜잭션을 시작하고, 동일한 토큰 또는 동일한 그룹의 다른 토큰에 대해 트랜잭션을 중지할 수 있습니다. 이는 단일 충전 계약 계정에서 하나 이상의 공유 전기차를 사용하는 여러 운전자가 있는 가족이나 기업의 일반적인 사용 사례를 지원합니다.

토큰(idTags)은 **IdTagInfo**의 선택적 **ParentId** 요소에 공통 그룹 식별자를 지정하여 권한 부여 목적으로 그룹화됩니다. 두 개의 idTags는 **ParentId** 태그가 일치하면 동일한 그룹에 있는 것으로 간주됩니다.



ParentId가 idTag와 동일한 명목 데이터 유형(**IdToken**)을 가지고 있더라도 이 요소의 값은 **IdTokens**의 일반적인 형식이 아닐 수 있고/있거나 실제 유효한 **IdToken**을 나타내지 않을 수 있습니다(예: 공통 공유 "계정 번호"일 수 있음). 따라서 **ParentId** 값은 제시된 토큰 값과 비교하는 데 사용해서는 안 됩니다(idTag 값으로도 나타나는 경우 제외).

3.11. Reservations

이 섹션은 정보 제공용입니다.

Reserve Now 작업을 사용하여 충전기를 예약할 수 있습니다. 이 작업은 특정 idTag에 대한 특정 만료 시간까지 충전기를 예약합니다. 부모 idTag는 '그룹' 예약을 지원하기 위해 예약에 포함될 수 있습니다. 충전기에서 특정 커넥터를 예약하거나 충전기에서 모든 커넥터를 예약할 수 있습니다. 예약된 idTag가 예약된 커넥터(지정된 경우) 또는 모든 커넥터(지정되지 않은 경우)에서 사용되거나 만료 시간에 도달하거나 예약이 명시적으로 취소되면 예약이 해제됩니다.

3.12. Vendor-specific data transfer

이 섹션은 정보 제공용입니다.

공급업체별 데이터 전송 메커니즘은 OCPP에서 표준화되지 않은 데이터 또는 메시지를 교환할 수 있도록 합니다. 따라서 OCPP 내에서 실험적 기능을 위한 프레임워크를 제공하며, 이는 향후 OCPP 버전에 적용될 수 있습니다. 새로운(호환되지 않을 수 있는) OCPP 비표준 규격을 만들지 않고도 실험할 수 있습니다. 둘째, 특정 CSMS와 충전기 공급업체 간에 합의된 추가 기능을 구현할 수 있는 가능성을 제공합니다.

공급업체별 데이터 정의는 CSMS가나 충전기에서 시작할 수 있습니다.



이 옵션을 사용하지 않는 다른 시스템과의 호환성에 영향을 미치므로, 매우 신중하게 선택적 기능에만 사용하십시오. 설명서 및/또는 커뮤니케이션에서 사용 방법을 명시적으로 언급하는 것이 좋습니다. 이 옵션을 사용하여 기능을 추가하기 전에 Open Charge Alliance에 문의하는 것을 고려하십시오.

3.13. Smart Charging

이 섹션은 규범적입니다.

스마트 충전을 사용하면 CSMS가 특정 EV의 충전 전력 또는 전류, 전체 충전기 / 충전기 그룹의 총 허용 에너지 소비량에 영향을 미칠 수 있습니다. 예를 들어, 그리드 연결, 그리드의 에너지 가용성 또는 건물 배선을 기반으로 합니다. 충전 전력 또는 전류에 영향을 미치는 것은 특정 시점의 에너지 전송 한도에 따라 달라집니다. 이러한 한도는 충전 프로파일에 결합됩니다.

3.13.1. Charging profile purposes

충전 프로파일은 충전 스케줄로 구성되며, 이는 기본적으로 최대 충전 전력 또는 전류가 있는 시간 간격 목록과 일정의 기간 및 반복을 지정하는 몇 가지 값입니다.

목적에 따라 세 가지 유형의 충전 프로파일이 있습니다.

- *ChargePointMaxProfile*

부하 분산(load balancing) 시나리오에서 충전기에는 충전기의 모든 커넥터가 공유할 전력 또는 전류를 제한하는 하나 이상의 로컬 충전 프로파일이 있습니다. CSMS는 ChargingProfilePurpose를 "*ChargePointMaxProfile*"로 설정하여 이러한 프로파일을 구성해야 합니다. *ChargePointMaxProfile*은 충전기의 ConnectorId 0에서만 설정할 수 있습니다.

- *TxDefaultProfile*

새 트랜잭션에 대한 기본 스케줄은 충전 정책을 부과하는 데 사용될 수 있습니다. 이러한 정책의 예로, 낮 동안 요금을 청구하지 않도록 정책을 설정할 수 있습니다. 이러한 목적으로 스케줄을 설정하는 경우, ChargingProfilePurpose를 *_TxDefaultProfile_*로 설정해야 합니다.

*TxDefaultProfile*이 ConnectorId 0으로 설정된 경우 *TxDefaultProfile*은 모든 커넥터에 적용됩니다.

ConnectorId가 >0으로 설정된 경우, 해당 특정 Connector에만 적용됩니다.

Connector 0에 대한 *TxDefaultProfile*이 설정되고 CSMS가 ConnectorId >0인 새 프로파일을 보내는 경우, *TxDefaultProfile*은 해당 특정 Connector에 대해서만 교체해야 합니다.

- *TxProfile*

목적이 *TxProfile*인 트랜잭션별 프로파일이 있는 경우, 현재 트랜잭션 기간 동안만 목적이 *TxDefaultProfile*인 기본 충전 프로파일을 무시해야 합니다. 트랜잭션이 중지된 후에는 프로파일을 삭제해야 합니다. *TxProfile* 유형의 충전 프로파일에 지정된 커넥터에서 활성화된 트랜잭션이 없는 경우 충전기는 이를 삭제하고 *SetChargingProfile.conf*에서 오류 상태를 반환해야 합니다.

트랜잭션에 적용되는 최종 스케줄 제약 조건은 목적이 *ChargePointMaxProfile*인 프로파일을 프로파일 *TxProfile* 또는 목적이 *TxProfile*인 프로파일이 제공되지 않은 경우 *TxDefaultProfile*과 병합하여 결정됩니다. *TxProfile*은 충전기 ConnectorId >0에서만 설정해야 합니다.

3.13.2. Stacking charging profiles

복잡한 일정을 설명하기 위해 동일한 충전 프로파일 목적의 충전 프로파일을 스택킹할 수 있습니다. 예를 들어, 주중 23:00h에서 06:00h까지, 주말 00:00h에서 24:00h까지 최대 전력 또는 전류량을 허용하고 다른 시간에는 전력 또는 전류량을 줄이는 목적 *TxDefaultProfile*의 충전 프로파일을 정의할 수 있습니다. 그 위에 휴일과 같이 이 규칙에 대한 예외를 정의하는 다른 *TxDefaultProfiles*를 정의할 수 있습니다.

충전 프로파일의 우선 순위는 StackLevel 매개변수 값에 따라 결정됩니다. 언제든지 유효한 충전 프로파일은 validFrom 및 validTo 매개변수에 따라 해당 시점에 유효한 프로파일 중에서 stackLevel이 가장 높은 충전 프로파일이어야 합니다.

충돌을 피하기 위해 충전기에 동일한 stackLevel 및 용도가 있는 여러 충전 프로파일이 존재하는 것은 허용되지 않습니다. 충전기가

충전기에 이미 존재하는 `stackLevel` 및 용도가 있는 충전 프로파일을 수신할 때마다 충전기는 기존 프로파일을 대체해야 합니다.



동일한 `stackLevel` 및 용도가 있는 업데이트된 충전 프로파일이 미래의 `validFrom dateTime`와 함께 전송되는 경우, 충전기는 설치된 프로파일을 대체하고 `validFrom`에 도달할 때까지 기본 동작으로 돌아가야 합니다. 프로파일 적용 공백(gaps)을 방지하기 위해, 시작 시각을 과거로 설정하여 충전 프로파일을 충전기로 전송하는 것을 권장합니다.



지속 시간 없이 가장 높은 스택 레벨에서 스택킹을 사용하는 경우 충전기는 더 낮은 스택 레벨 프로파일로 절대 떨어지지 않습니다.

3.13.3. Combining charging profile purposes

충전 레벨을 안내하는 복합 일정은 다양한 `chargingProfilePurposes`의 기존 충전 프로필을 조합한 것입니다.

이 복합 일정은 각 시간 간격의 최소값을 취하여 계산합니다. 시간 간격은 고정된 길이일 필요가 없으며 모든 충전 프로필 목적에 대해 동일할 필요도 없습니다.

즉, 결과 복합 일정에는 길이가 다른 간격이 포함될 수 있습니다.

어느 시점에서든 충전 프로필 `ChargePointMaxProfile` 및 `TxDefaultProfile`(또는 `TxProfile`)의 일정을 병합한 결과인 복합 일정의 사용 가능한 전력 또는 전류는 병합된 일정에서 사용 가능한 전력 또는 전류의 가장 낮은 값보다 작거나 같아야 합니다.

충전 포인트에 두 개 이상의 커넥터가 장착된 경우 `ChargePointMaxProfile`의 한계 값은 모든 커넥터를 합친 한계입니다. 모든 커넥터의 합친 에너지 흐름은 `ChargePointMaxProfile`에서 설정한 한계보다 커서는 안 됩니다.

3.13.4. Smart Charging Use Cases

이 섹션은 정보 제공용입니다.

스마트 충전에는 여러 가지 다른 용도가 있을 수 있습니다. 다음 세 가지 일반적인 스마트 충전 종류를 사용하여 스마트 충전의 가능한 동작을 설명합니다.

- Load balancing
- Central smart charging
- Local smart charging

위의 사용 사례 중 두 개 이상을 하나의 더 복잡한 시스템으로 결합하는 더 복잡한 사용 사례가 가능합니다.

Load Balancing

이 섹션은 정보 제공용입니다.

부하 분산 사용 사례는 충전기 내의 내부 부하 분산에 관한 것이며, 충전기는 커넥터당 충전 스케줄을 제어합니다. 충전기는 고정된 한도(예: 그리드 연결의 최대 전류)로 구성됩니다.

옵셔널 값인 충전 스케줄 필드 `minChargingRate`는 충전기에서 커넥터 간의 전력 분배를 최적화하는 데 사용될 수 있습니다. 이 매개변수는 충전기에 `minChargingRate` 이하의 충전은 비효율적임을 알려 다른 밸런싱 전략을 선택할 수 있는 가능성을 제공합니다.

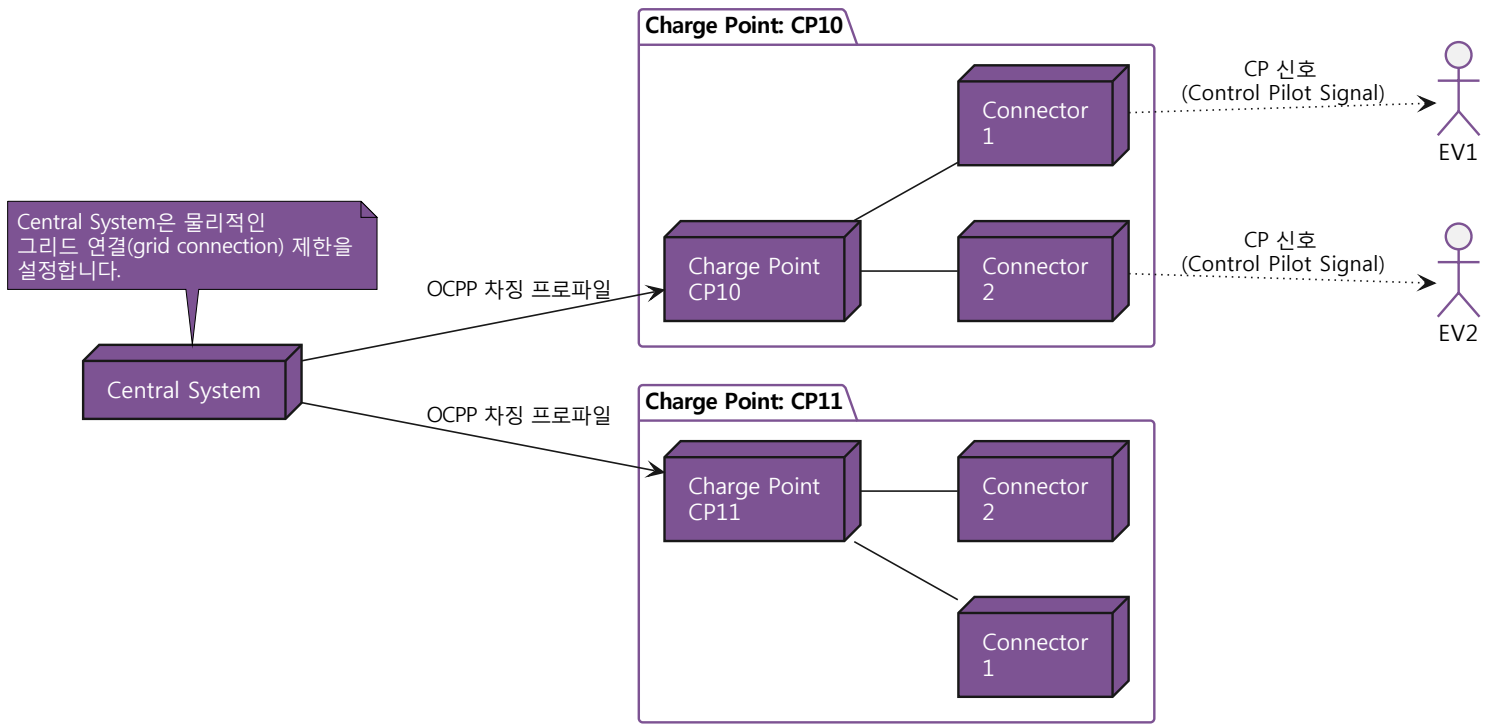


Figure 6. 부하 분산 스마트 충전 토폴로지

Central Smart Charging

이 섹션은 정보 제공용입니다.

중앙 스마트 충전을 사용하면 트랜잭션당 충전 일정에 대한 제약은 CSMS가 결정합니다. CSMS는 충전기가 외부 시스템에서 부과하는 전력량 제한 내에 머무를 수 있도록 이러한 일정을 사용합니다.

CSMS는 충전기 커넥터의 제한을 직접 제어합니다.

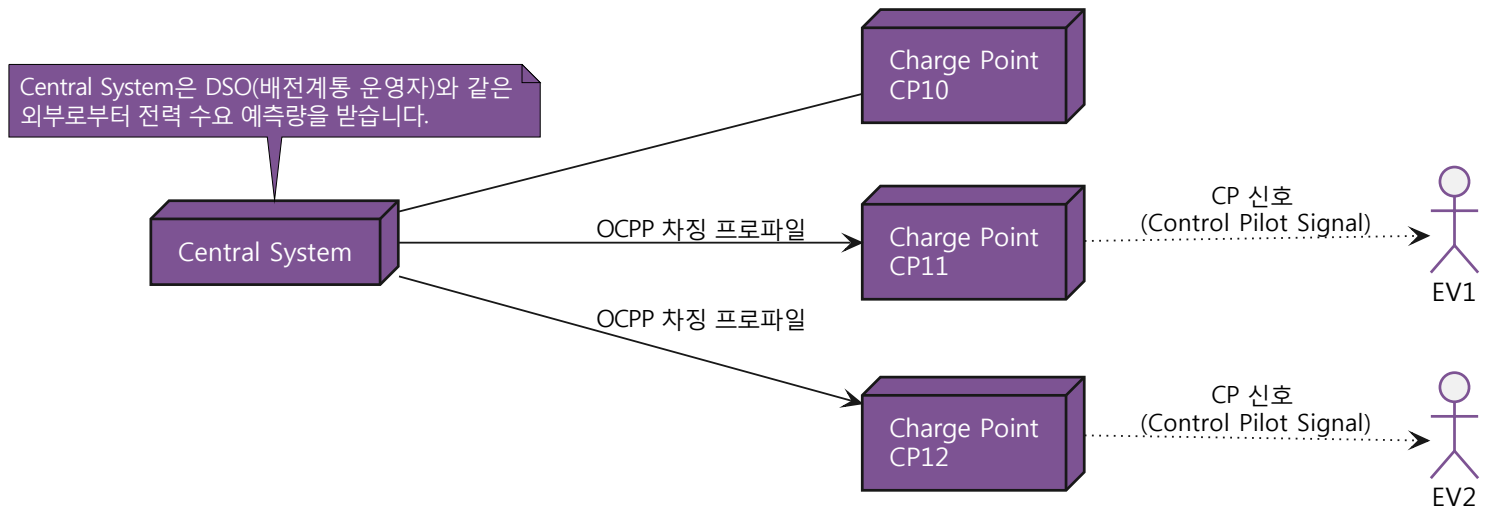


Figure 7. 중앙 스마트 충전 토폴로지

중앙 스마트 충전은 요금 한도가 CSMS에서 제어된다고 가정합니다. CSMS는 그리드 운영자(DSO) 또는 다른 소스에서 어떤 형태로든 용량 예측을 수신하고 일부 또는 모든 충전 트랜잭션에 대한 충전 일정을 계산합니다. 용량 예측과 관련된 세부 정보는 이 사양의 범위를 벗어나므로, 이 문서에서 설명하지 않습니다.

CSMS는 커넥터에 충전량 제한을 설정합니다. `StartTransaction.req` PDU에 대한 응답으로 CSMS는 TxProfile을 사용하여 트랜잭션에 대한 충전 한도를 설정할 수 있습니다.

중앙 스마트 충전은 제어 파일럿 신호로 수행할 수 있지만, EV가 제어 파일럿 신호를 통해 충전을 통신할 수 없기 때문에 몇 가지 제한이 있습니다. **Local Smart Charging** 사용 사례와 유사하게 커넥터는 제어 파일럿 신호로 충전 일정을 실행할 수 있습니다. 이는 아래 그림에

설명되어 있습니다.

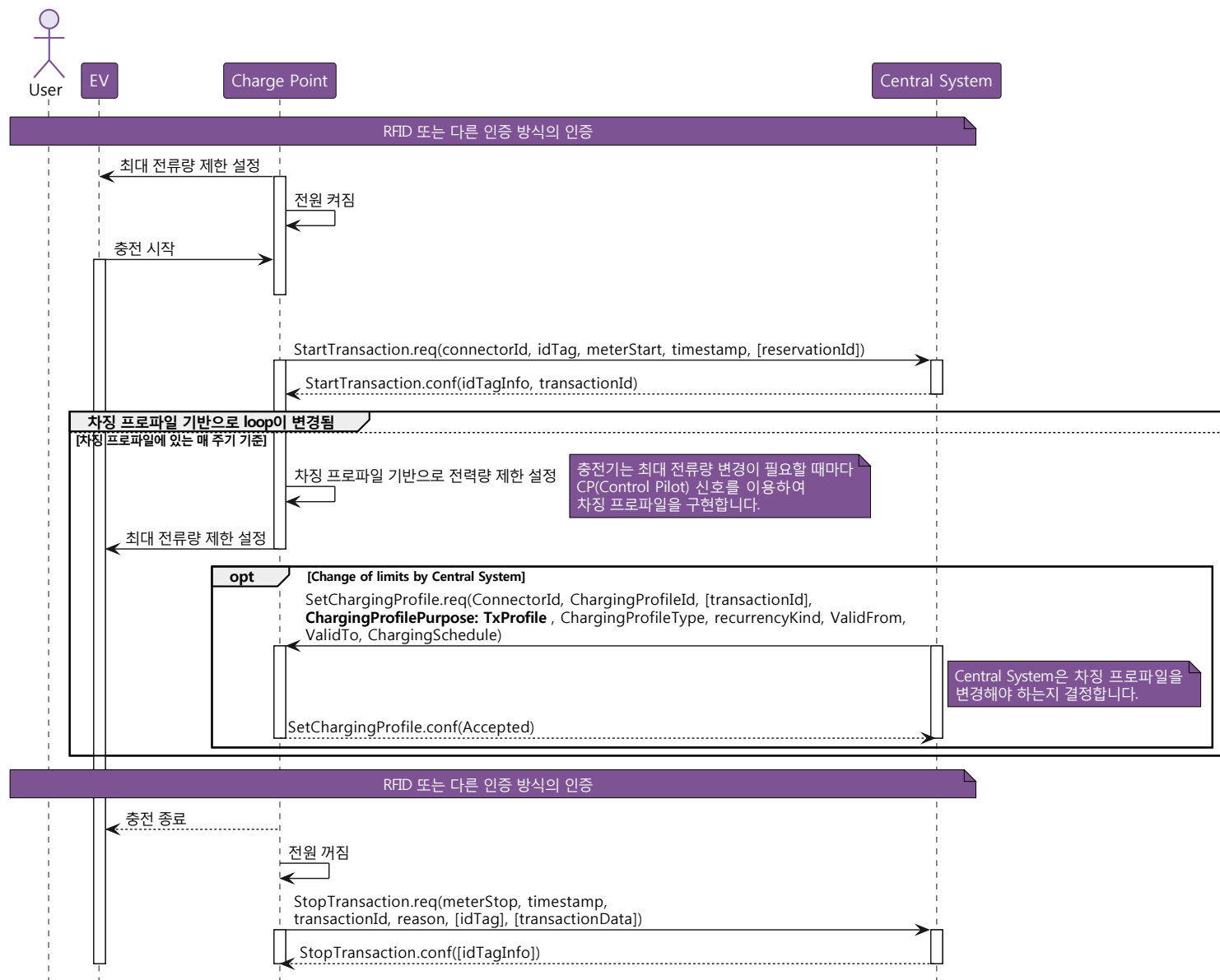


Figure 8. 시퀀스 다이어그램: 중앙 스마트 충전

위 그림에 대한 설명:

- 사용자 인증 후 커넥터는 제어 파일럿 신호를 통해 사용할 최대 전류를 설정합니다. 이 제한은 커넥터가 CSMS에서 이전에 수신한 (기본) 충전 프로파일을 기반으로 합니다. EV가 충전을 시작하고 **StartTransaction.req**가 CSMS로 전송됩니다.
- 충전이 진행되는 동안 커넥터는 충전 프로파일에 따라 최대 전류 또는 전력을 지속적으로 조정합니다. 선택적으로, CSMS는 언제든지 EV에 대한 제한 일정으로 사용될 커넥터에 대한 새로운 충전 프로파일을 보낼 수 있습니다.

Local Smart Charging

로컬 스마트 충전 사용 사례는 스마트 충전이 가능한 충전기가 CSMS가 아닌 로컬 컨트롤러에 의해 로컬에서 제어되는 충전 제한을 갖는 사용 사례를 설명합니다. 로컬 스마트 충전에 대한 사용 사례는 충전기 그룹에서 사용할 수 있는 전력량을 특정 최대치로 제한하는 것입니다. 일반적인 사용 사례는 그리드 연결 정격이 충전기 정격의 합보다 작은 주차장의 여러 충전기입니다. 또 다른 응용 분야는 로컬 컨트롤러가 DSO 또는 로컬 스마트 그리드 노드에서 전력 가용성에 대한 정보를 수신하는 것입니다.

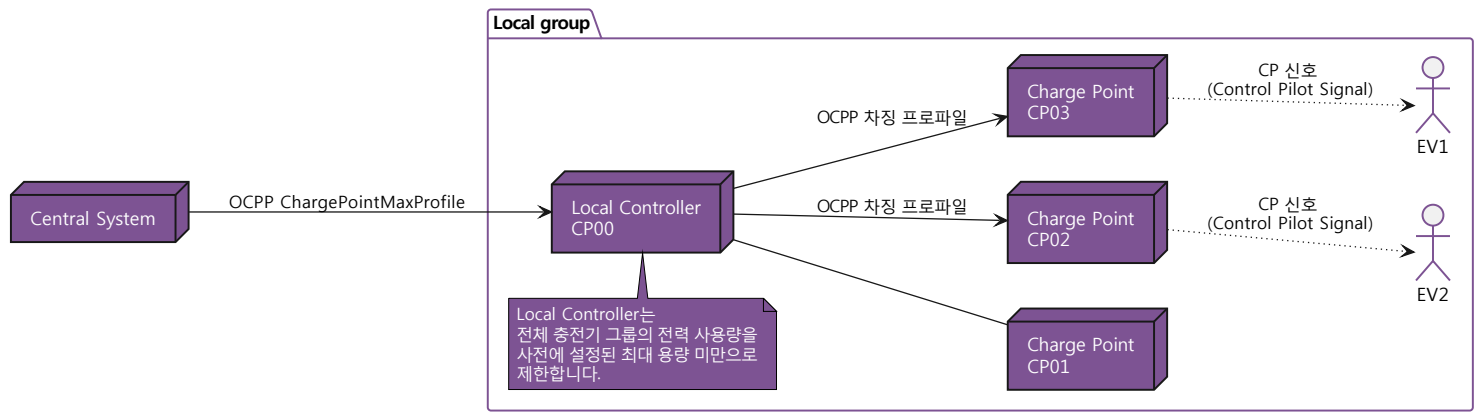


Figure 9. 로컬 스마트 충전 토폴로지

로컬 스마트 충전은 충전기 그룹을 제어하는 로컬 컨트롤러가 있다고 가정합니다. 로컬 컨트롤러는 논리적 구성 요소입니다. 별도의 물리적 구성 요소로 구현되거나 여러 다른 충전기를 제어하는 '마스터' 충전기의 일부로 구현될 수 있습니다. 로컬 컨트롤은 OCPP 프로토콜을 구현하고 그룹 구성원의 OCPP 메시지에 대한 프록시이며 자체 커넥터가 있을 수도 있고 없을 수도 있습니다.

로컬 스마트 충전의 경우 로컬 컨트롤러는 충전기에 충전 제한을 부과합니다. 이러한 제한은 충전기 그룹의 전력 소비를 그룹 제한 내로 유지하기 위해 충전 프로세스 중에 동적으로 변경될 수 있습니다. 그룹 제한은 로컬 컨트롤러에서 미리 구성되거나 CSMS에서 구성되었을 수 있습니다.

선택 사항인 충전 일정 필드 `minChargingRate`는 로컬 컨트롤러에서 커넥터 간의 전력 분배를 최적화하는 데 사용될 수 있습니다. 매개변수는 로컬 컨트롤러에 `minChargingRate` 이하의 충전은 비효율적임을 알려 다른 밸런싱 전략을 선택할 수 있는 가능성을 제공합니다.

다음 다이어그램은 로컬 스마트 충전 그룹의 충전기에 대한 충전 한도를 설정하는 메시지 시퀀스를 보여줍니다. 이러한 한도는 로컬 컨트롤러에서 어떤 식으로든 미리 구성하거나 CSMS에서 설정할 수 있습니다. 로컬 컨트롤러에는 필요에 따라 한도를 조정하여 연결된 커넥터 간에 이 용량을 분배하는 논리가 포함되어 있습니다.

1. 로컬 그룹 한도 사전 설정 image::images/figure_10.svg[로컬 그룹 한도 사전 설정]

다음 다이어그램은 일반적인 로컬 스마트 충전 사례에 대한 메시지 시퀀스를 설명합니다. 단순화를 위해 이 사례에는 커넥터가 하나만 포함됩니다.

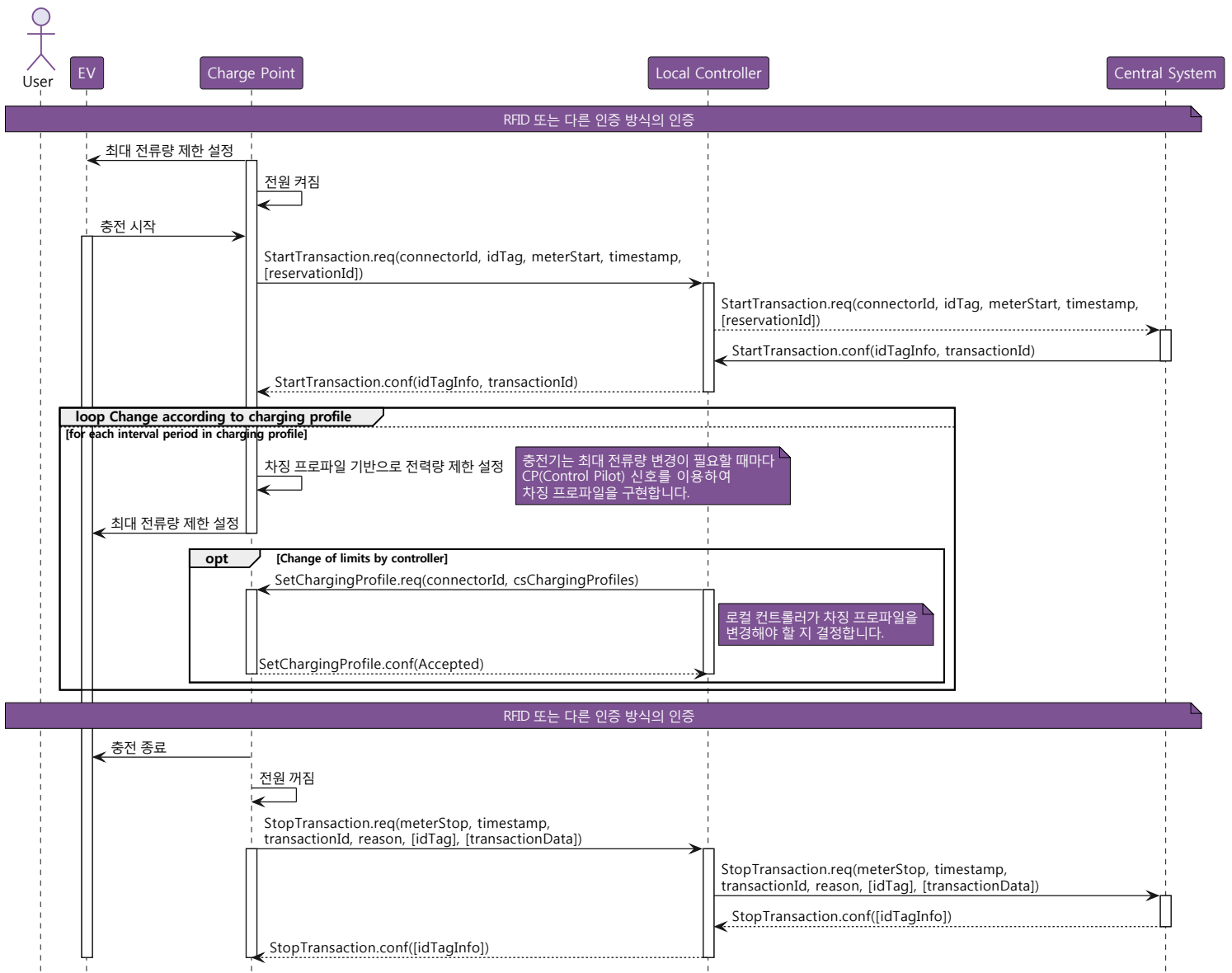


Figure 10. 시퀀스 다이어그램: 로컬 스마트 충전

위 그림에 대한 설명:

- 인증 후 커넥터는 Control Pilot 신호를 통해 사용할 최대 전류를 설정합니다. 이 제한은 커넥터가 로컬 컨트롤러에서 이전에 수신한 (기본) 충전 프로파일을 기반으로 합니다. EV는 충전을 시작하고 **StartTransaction.req**를 보냅니다.
- StartTransaction.req**는 로컬 컨트롤러를 통해 CSMS로 전송되므로 로컬 컨트롤러도 트랜잭션이 시작되었음을 알 수 있습니다. 로컬 컨트롤러는 충전기와 CSMS 간에 메시지를 전달하기만 하므로 CSMS는 모든 로컬 스마트 충전 그룹 구성원을 개별적으로 처리할 수 있습니다.
- 충전이 진행되는 동안 커넥터는 충전 프로파일에 따라 최대 전류를 지속적으로 조정합니다.
선택적으로, 언제든지 로컬 컨트롤러는 EV에 대한 제한 일정으로 사용될 새로운 충전 프로파일을 커넥터에 보낼 수 있습니다.

3.13.5. Discovery of Charge Point Capabilities

이 섹션은 규범적입니다.

정의된 스마트 충전 옵션은 광범위하게 사용될 수 있습니다. 충전기 간의 제한 및 기능 차이 존재 가능성으로 인해 CSMS는 충전기 특정 기능을 발견할 수 있어야 합니다. 이는 이 장에서 정의된 표준화된 구성 키에 의해 보장됩니다. 스마트 충전이 가능한 충전기는 **GetConfiguration.req** PDU를 통해 다음 구성 키를 구현하고 보고를 지원해야 합니다.

스마트 충전 구성 키

ChargeProfileMaxStackLevel

ChargingScheduleAllowedChargingRateUnit

ChargingScheduleMaxPeriods

MaxChargingProfilesInstalled

모든 표준화된 구성 키의 전체 목록은 [Standard Configuration Key Names & Values](#)에서 확인할 수 있습니다.

3.13.6. Offline behavior of smart charging

이 섹션은 규범적입니다.

충전기가 목적이 TxProfile인 트랜잭션별 충전 프로파일을 수신한 후, 오프라인 이 되면 트랜잭션 기간 동안 이 프로파일을 계속 사용해야 합니다.

트랜잭션이 시작되기 전이나 목적이 TxProfile인 트랜잭션별 충전 프로파일을 수신하기 전에 충전기가 오프라인 이 되면 사용 가능한 충전 프로파일을 사용해야 합니다. 다음 충전 프로파일 목적 중 0개 이상이 이전에 CSMS에서 수신되었을 수 있습니다.

- ChargePointMaxProfile
- TxDefaultProfile

다른 목적의 충전 프로파일을 결합하는 방법에 대한 설명은 섹션 [Combining Charging Profile Purposes](#)를 참조하세요.

충전기가 충전 프로파일 없이 오프라인 이 되면 제약 조건이 적용되지 않는 것처럼 트랜잭션을 실행해야 합니다.

3.13.7. Example data structure for smart charging

이 섹션은 정보 제공용입니다.

다음 데이터 구조는 08:00h와 20:00h 사이에 전력을 6kW로 제한하는 일일 기본 프로파일을 설명합니다.

CHARGINGPROFILE	
chargingProfileId	100
stackLevel	0
chargingProfilePurpose	TxDefaultProfile
chargingProfileKind	Recurring
recurrencyKind	Daily
chargingSchedule	(List of 1 ChargingSchedule elements)
ChargingSchedule	
duration	86400 (= 24 hours)
startSchedule	2013-01-01T00:00Z

CHARGINGPROFILE

chargingRateUnit	W
chargingSchedulePeriod	(List of 3 ChargingSchedulePeriod elements)
ChargingSchedulePeriod	
startPeriod	0 (=00:00)
limit	11000
numberPhases	3
ChargingSchedulePeriod	
startPeriod	28800 (=08:00)
limit	6000
numberPhases	3
ChargingSchedulePeriod	
startPeriod	72000 (=20:00)
limit	11000
numberPhases	3



충전 중 사용되는 위상의 양은 충전기, EV 및 충전기와 EV 간 케이블에 의해 제한됩니다. 이 3개 중 하나라도 3상 충전이 불가능한 경우 EV는 1상만 사용하여 충전됩니다.



스케줄 또는 충전 세션 중에 사용되는 위상의 수를 전환하는 것은 신중하게 수행해야 합니다. 일부 EV는 이를 지원하지 않을 수 있으며, 위상의 양을 변경하면 물리적 손상이 발생할 수 있습니다. 구성 키 [ConnectorSwitch3to1PhaseSupported](#)를 이용하여 충전기가 트랜잭션 중에 위상의 양을 전환하는 것을 지원하는지 여부를 알 수 있습니다.



DST(일광 절약 시간대, 섬머타임)가 적용되거나 적용되지 않는 날에는 특수 프로파일이 필요할 수 있습니다(예: 상대 프로파일의 경우)

3.14. Time zones

이 섹션은 정보 제공용입니다.

OCPP는 시간 값에 특정 시간대를 사용하도록 규정하지 않습니다. 그러나 CSMS와 충전소 간의 상호 운용성을 개선하기 위해 모든 시간 값에 UTC를 사용하는 것이 좋습니다.

3.15. Time notations

이 섹션은 규범적입니다.

구현은 ISO 8601 날짜 시간 표기법을 사용해야 합니다. 메시지 수신자는 소수 초와 시간대 오프셋을 처리할 수 있어야 합니다(다른

구현에서 사용할 수 있음). 메시지 발신자는 중요하지 않은 소수 점 아래의 초 단위를 생략하여 데이터 사용량을 절약할 수 있습니다.

3.16. Metering Data

이 섹션은 규범적입니다.

충전 세션과 관련된 광범위한 계량 데이터는 의도된 목적에 따라 다양한 방식으로 기록 및 전송될 수 있으며, 아래와 같이 두 가지 명백한 사용 사례가 있습니다(하지만 미터 값의 사용은 이 두 가지에 국한되지 않습니다):

- [Charging Session Meter Values](#)
- [Clock-Aligned Meter Values](#)

두 가지 유형의 미터 판독값은 독립형 [MeterValues.req](#) 메시지(트랜잭션 중) 및/또는 [StopTransaction.req](#) PDU의 `transactionData` 요소의 일부로 보고될 수 있습니다.

3.16.1. Charging Session Meter Values

빈번한(예: 1-5분 간격) 미터 판독값을 측정하여 CSMS로 전송(일반적으로 "실시간")하여 웹, 앱, SMS 등을 통해 EV 사용자(일반적으로 충전기에 없음)에게 충전 세션 진행 상황에 대한 정보 업데이트를 제공할 수 있습니다. OCPP에서는 이를 "샘플링된 미터 데이터"라고 합니다. 판독값의 정확한 빈도와 시간은 "충분히 빈번"하기만 하면 크게 중요하지 않기 때문입니다. "샘플링된 미터 데이터"는 다음 구성 키로 구성할 수 있습니다.

- [MeterValuesSampledData](#)
- [MeterValuesSampledDataMaxLength](#)
- [MeterValueSampleInterval](#)
- [StopTxnSampledData](#)
- [StopTxnSampledDataMaxLength](#)

[MeterValueSampleInterval](#)는 "MeterValues" PDU에서 전송하도록 의도된 미터링(또는 기타) 데이터의 샘플링 사이의 시간(초)입니다. 샘플은 충전 트랜잭션 시작부터 이 간격으로 주기적으로 수집되어 전송됩니다.

관례에 따라 "0"(숫자 0) 값은 샘플링된 데이터를 전송하지 않아야 함을 의미하는 것으로 해석됩니다.

[MeterValuesSampledData](#)는 [MeterValues.req](#) PDU에 [MeterValueSampleInterval](#)초마다 포함될 측정량 세트를 규정하는 심표로 구분된 목록입니다. [MeterValuesSampledData](#) 목록의 최대 요소 수는 충전기에서 다음을 통해 보고할 수 있습니다. [MeterValuesSampledDataMaxLength](#)

[StopTxnSampledData](#)는 샘플링된 측정값이 [StopTransaction.req](#) PDU의 `TransactionData` 요소에 포함될 것을 규정하는 심표로 구분된 목록이며, Transaction 시작부터 [MeterValueSampleInterval](#)초마다 포함됩니다. [StopTxnSampledData](#) 목록에 있는 요소의 최대 개수는 충전기에서 [StopTxnSampledDataMaxLength](#)를 통해 보고될 수 있습니다.

3.16.2. Clock-Aligned Meter Values

그리드 운영자는 특정 시간 동기화 간격(일반적으로 15분 또는 30분)에 공인 인증된 전력계에서 미터 판독값을 가져오도록 요구할 수 있습니다.

"Clock-Aligned Billing Data"는 다음 구성 키로 구성할 수 있습니다.

- [ClockAlignedDataInterval](#)
- [MeterValuesAlignedData](#)

- `MeterValuesAlignedDataMaxLength`
- `StopTxnAlignedData`
- `StopTxnAlignedDataMaxLength`

`ClockAlignedDataInterval`는 클록 정렬 데이터 간격의 크기(초)입니다. 이는 00:00:00(자정)부터 시작하여 하루에 균일하게 간격을 둔 미터 데이터 집계 간격 세트를 정의합니다.

예를 들어, 900(15분)의 값은 매일을 96개의 15분 간격으로 나누어야 함을 의미합니다.

관례에 따라 "0"(숫자 0)의 값은 시계 정렬된 데이터를 전송하지 않아야 함을 의미하는 것으로 해석합니다.

`MeterValuesAlignedData`는 `MeterValues.req` PDU에 `ClockAlignedDataInterval`초마다 포함될 측정량 세트를 규정하는 심표로 구분된 목록입니다. `MeterValuesAlignedData` 목록의 최대 요소 수는 Charge Point에서 다음을 통해 보고할 수 있습니다.

`MeterValuesAlignedDataMaxLength`

`StopTxnAlignedData`는 Transaction의 모든 `ClockAlignedDataInterval`에 대해 `StopTransaction.req` PDU의 `TransactionData` 요소에 포함될 클록 정렬 주기적 측정값 세트를 규정하는 심표로 구분된 목록입니다. `StopTxnAlignedData` 목록의 최대 요소 수는 충전기에서 다음을 통해 보고할 수 있습니다. `StopTxnAlignedDataMaxLength`

3.16.3. Multiple Locations/Phases

충전기가 여러 위치 또는 위상에서 동일한 측정량을 측정할 수 있는 경우, 관련 구성 키 중 하나에서 모든 가능한 위치 및/또는 위상을 보고해야 합니다.

예: `Inlet`(모든 3개 위상)(그리드 연결) 및 `Outlet`(두 커넥터 모두에서 커넥터당 3개 위상)에서 `Current.Import` 를 측정할 수 있는 충전기. `Current.Import` 는 `MeterValuesSampledData`에서 설정됩니다. `MeterValueSampleInterval`은 300(초)으로 설정됩니다. 이러한 경우 충전기는 아래의 값을 전송해야 합니다.

- `MeterValues.req: connectorId = 0`; 3개의 `SampledValue` 요소, `location = Inlet` 인 단계당 하나씩.
- `MeterValues.req: connectorId = 1`; 3개의 `SampledValue` 요소, `location = Outlet` 인 단계당 하나씩.
- `MeterValues.req: connectorId = 2`; 3개의 `SampledValue` 요소, `location = Outlet` 인 단계당 하나씩.

3.16.4. Unsupported measurands

CSMS가 다음 구성 키 중 하나를 사용하여 `ChangeConfiguration.req`를 충전소로 보낼 때:

- `MeterValuesAlignedData`
- `MeterValuesSampledData`
- `StopTxnAlignedData`
- `StopTxnSampledData`

심표로 구분된 목록에 이 충전소에서 지원하지 않는 측정량이 하나 이상 포함된 경우 충전소는 `ChangeConfiguration.conf`와 함께 `status = Rejected` 로 응답해야 합니다. 현재 구성은 변경되지 않아야 합니다.

3.16.5. No metering data in a Stop Transaction

구성 키: `StopTxnAlignedData` 및 `StopTxnSampledData`가 빈 문자열로 설정된 경우, 충전기는 `StopTransaction.req` PDU에 미터 값(`meter values`)을 넣지 않아야 합니다.

Chapter 4. Operations Initiated by Charge Point

4.1. Authorize

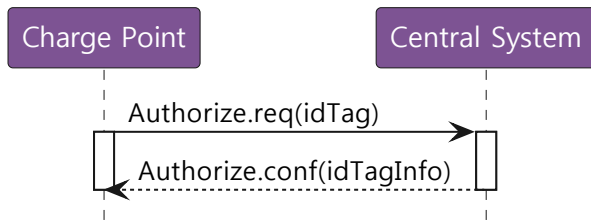


Figure 11. 시퀀스 다이어그램: 인증

전기차 소유자가 충전을 시작하거나 중지하려면 충전기는 인증 과정을 진행해야 합니다. 충전기는 인증 후에만 에너지를 공급해야 합니다. 트랜잭션을 중지할 때, 충전기는 트랜잭션을 중지하는 데 사용된 식별자가 트랜잭션을 시작한 식별자와 다른 경우에만 **Authorize.req**를 보내야 합니다.

Authorize.req는 충전을 위한 식별자의 인증에만 사용해야 합니다.

충전기는 로컬 인증 목록에 설명된 대로 CSMS를 포함하지 않고 로컬에서 식별자를 인증할 수 있습니다. 사용자가 제시한 idTag가 **Local Authorization List** 또는 인증 캐시에 없는 경우, 충전기는 CSMS에 **Authorize.req** PDU를 보내 인증을 요청해야 합니다. idTag가 로컬 인증 목록 또는 인증 캐시에 있는 경우, 충전기는 CSMS에 **Authorize.req** PDU를 보낼지 선택할 수 있습니다.

Authorize.req PDU를 수신하면 CSMS는 **Authorize.conf** PDU로 응답해야 합니다. 이 응답 PDU는 CSMS에서 idTag를 수락했는지 여부를 나타냅니다. CSMS에서 idTag를 수락하면 응답 PDU에 **parentIdTag**가 포함될 수 있으며 수락 또는 거부 이유를 나타내는 권한 상태 값이 포함되어야 합니다.

충전기에서 인증 캐시를 구현한 경우 **Authorize.conf** PDU를 수신하면 충전기는 idTag가 **Local Authorization List**에 없으면 인증 캐시에 설명된 대로 응답의 **IdTagInfo** 값으로 캐시 항목을 업데이트해야 합니다.

4.2. Boot Notification

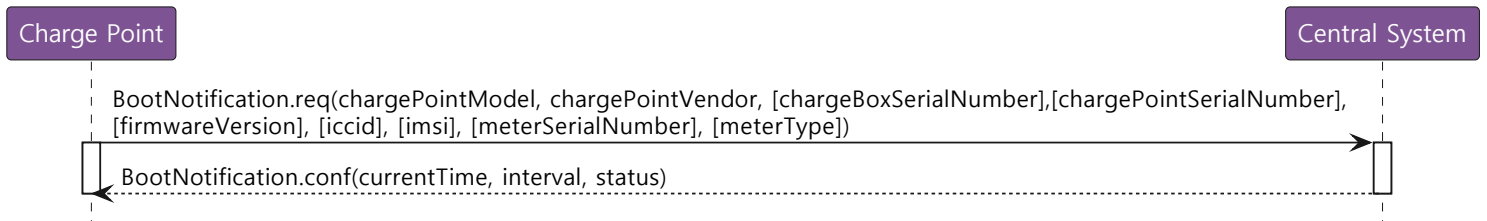


Figure 12. 시퀀스 다이어그램: 부팅 알림

부팅 후 충전기는 구성에 대한 정보(예: 버전, 공급업체 등)와 함께 CSMS에 요청을 보내야 합니다. CSMS는 충전기를 수락할지 여부를 나타내기 위해 응답해야 합니다.

충전기는 부팅 또는 재부팅할 때마다 **BootNotification.req** PDU를 보내야 합니다. 물리적 전원 켜기/재부팅과 CSMS가 **Accepted** 또는 **Pending** 을 반환하는 **BootNotification** 정상적인 요청 - 응답 사이클 사이에 충전기는 CSMS에 다른 요청을 보내지 않아야 합니다. 여기에는 이전에 충전기에 여전히 있는 캐시된 메시지가 포함됩니다.

CSMS가 **BootNotification.conf**와 **Accepted** 상태로 응답하면 충전기는 응답 PDU의 간격에 따라 하트비트 간격을 조정하고 제공된 CSMS의 현재 시간과 내부 시계를 동기화하는 것이 좋습니다. CSMS가 **Accepted** 이외의 값을 반환하는 경우, 인터벌 필드의 값은 다음 **BootNotification** 요청을 보내기 전의 최소 대기 시간을 나타냅니다. 해당 간격 값이 0이면 충전기는 CSMS에 요청이 과도하게 전송되는 것을 방지하는 방식으로 자체적으로 대기 간격을 선택합니다. 충전기는 **TriggerMessage.req**로 요청하지 않는 한 더 일찍 **BootNotification.req**를 보내서는 안 됩니다.

CSMS가 **Rejected** 상태를 반환하는 경우, 충전기는 앞서 언급한 재시도 간격이 만료될 때까지 CSMS에 OCPP 메시지를 보내서는 안 됩니다. 이 간격 동안 충전기는 CSMS와 더 이상 통신할 수 없습니다. 예를 들어 통신 채널을 닫거나 통신 하드웨어를 종료할 수 있습니다.

또한 CSMS는 예를 들어 시스템 리소스를 확보하기 위해 통신 채널을 닫을 수 있습니다. *Rejected* 상태인 동안 충전기는 CSMS에서 시작한 메시지에 응답해서는 안 됩니다. CSMS는 어떤 메시지도 시작해서는 안 됩니다.

CSMS는 또한 CSMS가 충전기를 수락하기 전에 충전기에서 특정 정보를 검색하거나 설정하려고 한다는 것을 나타내기 위해 *Pending* 상태를 반환할 수 있습니다. CSMS가 *Pending* 상태를 반환하는 경우, 충전기나 CSMS는 통신 채널을 닫아서는 안 됩니다. CSMS는 충전기에서 정보를 검색하거나 구성을 변경하기 위한 요청 메시지를 보낼 수 있습니다. 충전기는 이러한 메시지에 응답해야 합니다. 충전소는 CSMS에서 *TriggerMessage.req* 요청으로 지시하지 않는 한 CSMS에 요청 메시지를 보내면 안 됩니다.

Pending 상태인 동안 다음의 CSMS에서 시작한 메시지는 허용되지 않습니다. *RemoteStartTransaction.req* 및 *RemoteStopTransaction.req*

4.2.1. Transactions before being accepted by a Central System

충전기 운영자는 충전기가 CSMS에서 등록되기 전에 트랜잭션을 수락하도록 구성할 수 있습니다. 이러한 동작을 구현하려는 당사자는 해당 트랜잭션이 CSMS에 전달될 수 있는지 여부가 불확실하다는 것을 알아야 합니다.

재시작(예: 원격 재설정 명령, 정전, 펌웨어 업데이트, 소프트웨어 오류 등으로 인해) 후 충전기는 CSMS에 다시 *BootNotification* 요청을 보내야 합니다. 충전기가 CSMS에서 *BootNotification.conf*를 수신하지 못한 경우, 올바르게 사전 설정된 내장형 비휘발성 실시간 클록 하드웨어가 없는 경우 충전기에 유효한 날짜/시간 설정이 없으므로, 트랜잭션 날짜/시간을 확인할 수 없습니다.

또한 CSMS가 장시간 또는 무기한으로 *Accepted* 가 아닌 상태를 표시하는 경우도 있습니다(예: 구성 오류).

충전기가 CSMS에 등록된 적이 없는 경우(현재 연결 설정, URL 등을 사용하여), 충전기에서 모든 충전 서비스를 거부하는 것이 좋습니다. 사용자를 인증할 수 없고 실행 중인 트랜잭션이 프로비저닝 프로세스와 충돌할 수 있기 때문입니다.

4.3. Data Transfer

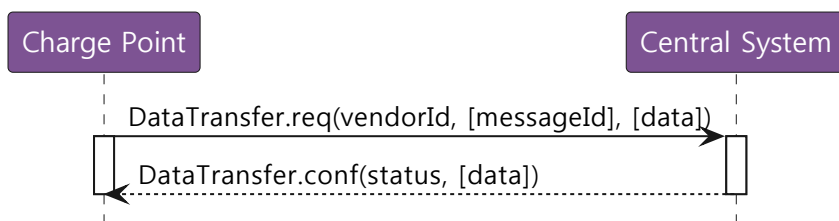


Figure 13. 시퀀스 다이어그램: 데이터 전송

충전기가 OCPP에서 지원하지 않는 기능에 대한 정보를 CSMS로 보내야 하는 경우 *DataTransfer.req* PDU를 사용해야 합니다.

요청의 *vendorId*는 CSMS가 알고 있어야 하며 공급업체별 구현을 고유하게 식별해야 합니다. *VendorId*는 역순 DNS 네임스페이스의 값이어야 하며, 이름의 최상위 계층은 역순으로 공급업체 조직의 공개적으로 등록된 기본 DNS 이름과 일치해야 합니다.

선택적으로 요청 PDU의 *messageId*는 특정 메시지 또는 구현을 나타내는 데 사용할 수 있습니다.

요청 및 응답 PDU의 데이터 길이는 정의되지 않았으며 관련된 모든 당사자가 합의해야 합니다.

요청 수신자가 특정 *vendorId*에 대한 구현이 없는 경우 상태 'UnknownVendor'를 반환하고 데이터 요소는 존재하지 않아야 합니다. *messageId*가 일치하지 않는 경우(사용된 경우) 수신자는 상태 'UnknownMessageId'를 반환해야 합니다. 다른 모든 경우에는 상태 'Accepted' 또는 'Rejected'를 사용하고 데이터 요소는 관련 당사자 간의 공급업체별 계약의 일부입니다.

4.4. Diagnostics Status Notification

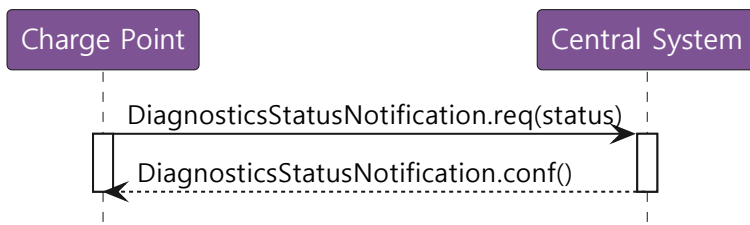


Figure 14. 시퀀스 다이어그램: 진단 상태 알림

충전기는 CSMS에 진단 업로드 상태를 알리는 알림을 보냅니다. 충전기는 `DiagnosticsStatusNotification.req` PDU를 보내 CSMS에 진단 업로드가 바쁘거나, 성공적으로 완료되었거나 실패했음을 알려야 합니다. 충전기는 진단 상태 알림에 대한 `TriggerMessage`를 수신한 후에만 Idle 상태를 보내야 하며, 이는 진단을 업로드하는 데 바쁘지 않을 때입니다.

`DiagnosticsStatusNotification.req` PDU를 수신하면 CSMS는 `DiagnosticsStatusNotification.conf`로 응답해야 합니다.

4.5. Firmware Status Notification

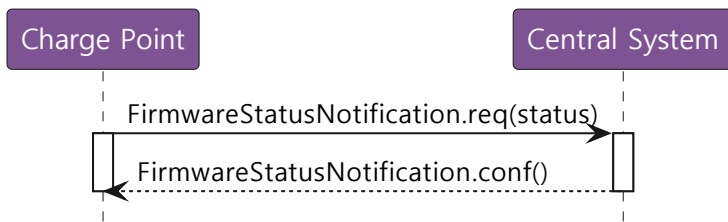


Figure 15. 시퀀스 다이어그램: 펌웨어 상태 알림

충전기는 펌웨어 업데이트 진행 상황을 CSMS에 알리기 위해 알림을 보냅니다. 충전기는 펌웨어 업데이트 다운로드 및 설치 진행 상황을 CSMS에 알리기 위해 `FirmwareStatusNotification.req` PDU를 보내야 합니다. 충전기는 펌웨어 다운로드/설치에 바쁘지 않을 때 펌웨어 상태 알림에 대한 `TriggerMessage`를 수신한 후에만 Idle 상태를 보내야 합니다.

`FirmwareStatusNotification.req` PDU를 수신하면 CSMS는 `FirmwareStatusNotification.conf`로 응답해야 합니다.

`FirmwareStatusNotification.req` PDU는 CSMS가 `FirmwareUpdate.req` PDU로 시작한 업데이트 프로세스의 상태를 CSMS에 최신 상태로 유지하기 위해 전송되어야 합니다.

4.6. Heartbeat



Figure 16. 시퀀스 다이어그램: Heartbeat

CSMS에 충전기가 여전히 연결되어 있음을 알리기 위해 충전기는 구성 가능한 시간 간격 후에 하트비트를 보냅니다.

충전기는 CSMS가 충전기가 여전히 살아 있음을 알 수 있도록 `Heartbeat.req` PDU를 보내야 합니다.

`Heartbeat.req` PDU를 수신하면 CSMS는 `Heartbeat.conf`로 응답해야 합니다. 응답 PDU에는 CSMS의 현재 시간이 포함되어야 하며, 충전기가 내부 시계를 동기화하는 데 사용하는 것이 좋습니다.

충전기는 구성된 하트비트 간격 내에 다른 PDU가 CSMS로 전송된 경우 `Heartbeat.req` PDU 전송을 건너뛸 수 있습니다. 즉, CSMS는 PDU가 수신될 때마다 충전기의 가용성을 가정해야 하며, `Heartbeat.req` PDU를 수신했을 때와 같은 방식으로 가정해야 합니다.



JSON을 WebSocket을 통해 사용하는 경우 하트비트를 보내는 것은 필수가 아닙니다. 그러나 시간 동기화의 경우 24시간당 최소 하나의 하트비트를 보내는 것이 좋습니다.

4.7. Meter Values

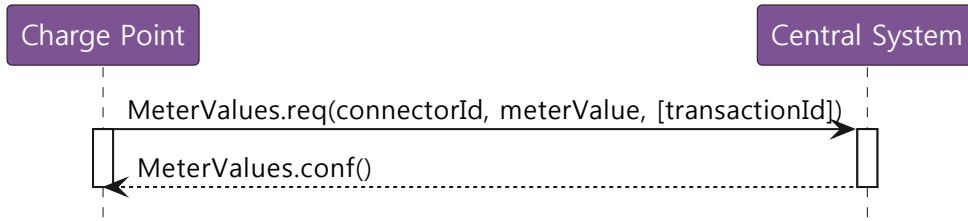


Figure 17. 시퀀스 다이어그램: 미터 값

충전기는 미터 값에 대한 추가 정보를 제공하기 위해 전기 미터 또는 기타 센서/변환기 하드웨어를 샘플링할 수 있습니다. 미터 값을 언제 보낼지는 충전기에서 결정합니다. 이는 [ChangeConfiguration.req](#) 메시지를 사용하여 데이터 수집 간격을 구성하고 수집 및 보고할 데이터를 지정할 수 있습니다.

충전기는 미터 값을 오프로드하기 위해 [MeterValues.req](#) PDU를 보내야 합니다. 요청 PDU는 각 샘플에 대해 다음을 포함해야 합니다.

1. 샘플을 가져온 커넥터의 ID입니다. `connectorId`가 0이면 전체 충전기와 연결됩니다. `connectorId`가 0이고 `Measurand`가 에너지와 관련된 경우 샘플은 메인 계량기에서 가져와야 합니다.
2. 해당되는 경우 이러한 값과 관련된 트랜잭션의 `transactionId`입니다. 진행 중인 트랜잭션이 없거나 값이 기본 미터에서 가져온 경우 트랜잭션 ID를 생략할 수 있습니다.
3. 하나 이상의 **meterValue** 요소, 유형 [MeterValue](#), 각각은 특정 시점에서 가져온 하나 이상의 데이터 값 세트를 나타냅니다.

각 [MeterValue](#) 요소에는 타임스탬프와 하나 이상의 개별 [sampledValue](#) 요소 세트가 포함되며, 모두 동일한 시점에서 캡처됩니다. 각 `sampledValue` 요소에는 단일 값 데이터가 포함됩니다. 각 `sampledValue`의 특성은 선택적 `measurand`, `context`, `location`, `unit`, `phase`, `format` 필드에 의해 결정됩니다.

선택 사항인 `measurand` 필드는 측정/보고되는 값의 유형을 지정합니다.

선택 사항인 `context` 필드는 판독을 트리거하는 이유/이벤트를 지정합니다.

선택 사항인 `location` 필드는 측정이 수행되는 위치(예: 입구, 출구)를 지정합니다.

선택 사항인 `phase` 필드는 값이 적용되는 전기 설비의 위상 또는 위상을 지정합니다. 충전기는 계량기(또는 없는 경우 그리드 연결) 관점에서 모든 위상 번호 종속 값을 보고해야 합니다.



위상 필드는 모든 [Measurands](#)에 적용되지 않습니다.



엄밀히 말하면 측정된 값이 없는 두 가지 측정량([Current.Offered](#) 및 [Power.Offered](#))을 사용할 수 있습니다. 이는 EV에 제공되는 최대 전류/전력량을 나타내며 스마트 충전 애플리케이션에서 사용하도록 의도되었습니다.

개별 커넥터 위상 회전 정보의 경우 CSMS는 [GetConfiguration](#)을 통해 충전기에서 [ConnectorPhaseRotation](#) 구성 키를 쿼리할 수 있습니다. 충전기는 그리드 연결과 관련하여 위상 회전을 보고해야 합니다. 커넥터당 가능한 값은 NotApplicable, Unknown, RST, RTS, SRT, STR, TRS 및 TSR입니다. 자세한 내용은 [Standard Configuration Key Names & Values](#) 섹션을 참조하세요.

EXPERIMENTAL 선택 사항인 `format` 필드는 데이터가 일반(기본) 형식으로 간단한 숫자 값("Raw")으로 표현되는지, 아니면 불투명한 디지털 서명된 이진 데이터 블록인 "SignedData"로 표현되는지를 지정합니다. 이 실험적 필드는 더 성숙한 솔루션 대안이 제공되면 이후 버전에서 더 이상 사용되지 않으며 이후 제거될 수 있습니다.

이전 버전과의 호환성을 유지하기 위해 `sampledValue` 요소의 모든 선택 사항 필드의 기본값은 추가 필드가 없는 **value**가 Wh(와트시) 단위의 활성 에너지에 대한 레지스터 판독으로 해석되도록 합니다.

MeterValues.req PDU를 수신하면 CSMS는 MeterValues.conf로 응답해야 합니다.

CSMS가 수신한 MeterValues.req에 포함된 데이터의 유효성 검사를 적용할 수도 있습니다. 이러한 유효성 검사의 결과로 인해 CSMS가 MeterValues.conf로 응답하지 않는 일이 절대 있어서는 안 됩니다. MeterValues.conf로 응답하지 않으면 충전기가 Error responses to transaction-related messages에 지정된 대로 동일한 메시지를 다시 시도하게 됩니다.

4.8. Start Transaction

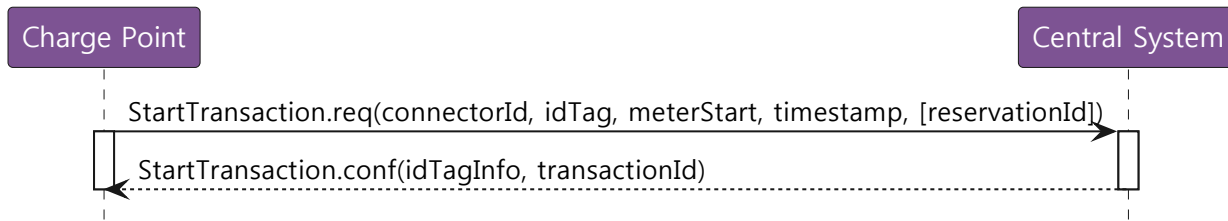


Figure 18. 시퀀스 다이어그램: 트랜잭션 시작

충전기는 CSMS에 StartTransaction.req PDU를 보내 시작된 트랜잭션에 대해 알려야 합니다. 이 트랜잭션이 예약을 종료하는 경우(Reserve Now 작업 참조), StartTransaction.req는 반드시 reservationId를 포함해야 합니다.

StartTransaction.req PDU를 수신하면 CSMS는 StartTransaction.conf PDU로 응답해야 합니다. 이 응답 PDU에는 트랜잭션 ID와 권한 부여 상태 값이 포함되어야 합니다.

CSMS는 StartTransaction.req PDU에 있는 식별자의 유효성을 확인해야 합니다. 로컬에서 충전기에 저장된 유효기간이 지난 데이터를 이용하여 식별자가 승인되었을 수 있기 때문입니다. 예를 들어, 식별자는 충전소의 Authorization Cache에 추가된 이후 차단되었을 수 있습니다.

충전기에서 인증 캐시를 구현한 경우 StartTransaction.conf PDU를 수신하면 충전소는 로컬 인증 목록에 idTag가 없는 경우 인증 캐시에 설명된 대로 응답의 IdTagInfo 값으로 캐시 항목을 업데이트해야 합니다.

CSMS는 수신한 StartTransaction.req에 포함된 데이터에 유효성 검사를 적용할 가능성이 있습니다. 이러한 유효성 검사의 결과는 CSMS가 StartTransaction.conf로 응답하지 않는 원인이 되어서는 안 됩니다. StartTransaction.conf로 응답하지 않으면 충전기가 Error responses to transaction-related messages에 지정된 대로 메시지 재전송을 시도하게 됩니다.

4.9. Status Notification

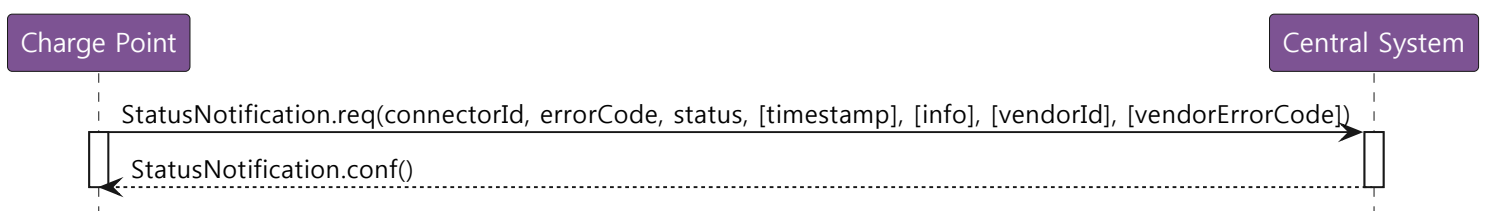


Figure 19. 시퀀스 다이어그램: 상태 알림

충전기는 CSMS에 알림을 보내 충전기의 상태 변경이나 오류를 CSMS에 알립니다. 다음 표는 이전 상태(왼쪽 열)에서 새 상태(위쪽 행)로의 변경 사항을 보여줍니다. 이 변경 사항에 따라 충전기는 StatusNotification.req PDU를 CSMS로 보낼 수 있습니다.



이전 OCPP 버전에서 정의한 점유 상태는 더 이상 관련이 없습니다. 점유 상태는 *Preparing*, *Charging*, *SuspendedEV*, *SuspendedEVSE*, *Finishing* 의 다섯 가지 새로운 상태로 나뉩니다.



EVSE는 향후 호환성을 위해 소켓이나 충전기 대신 상태 알림에 사용됩니다.

다음 표는 가능한 상태 전환을 설명합니다.

	상태 시작 / 종료:	1 Availab le	2 Prepari ng	3 Chargi ng	4 Suspen dedEV	5 Suspen dedEVS E	6 Finishin g	7 Reserve d	8 Unavail able	9 Faulted
A	Available		A2	A3	A4	A5		A7	A8	A9
B	Preparing	B1		B3	B4	B5	B6			B9
C	Charging	C1			C4	C5	C6		C8	C9
D	SuspendedEV	D1		D3		D5	D6		D8	D9
E	SuspendedEVSE	E1		E3	E4		E6		E8	E9
F	Finishing	F1	F2						F8	F9
G	Reserved	G1	G2						G8	G9
H	Unavailable	H1	H2	H3	H4	H5				H9
I	Faulted	I1	I2	I3	I4	I5	I6	I7	I8	



위의 표는 ConnectorId > 0에만 적용됩니다. ConnectorId 0의 경우 Available, Unavailable, Faulted라는 제한된 집합만 적용됩니다.

다음 표는 상태 변경으로 이어질 수 있는 이벤트를 설명합니다.

	설명
A2	사용이 시작됩니다(예: 플러그 삽입, 충전 공간 점유, idTag 제시, 시작 버튼 누르기, RemoteStartTransaction.req 수신)
A3	인증 수단 없이도 충전기에서 가능
A4	A3 과 유사하지만 EV가 충전을 시작하지 않음
A5	A3 과 유사하지만 EVSE가 충전을 허용하지 않음
A7	커넥터를 예약하는 Reserve Now 메시지가 수신됨
A8	커넥터를 사용할 수 없음(Unavailable)으로 전환하는 Change Availability 메시지가 수신됨
A9	추가 충전 작업을 방해하는 오류가 감지됨
B1	예상된 사용이 종료됨(예: 플러그 제거, 충전 공간이 더 이상 사용되지 않음, idTag의 두 번째 제시, 예상된 사용자 작업에서 시간 초과(구성 키 ConnectionTimeOut 에 의해 구성됨))
B3	충전을 위한 모든 전제 조건이 충족되고 충전 프로세스가 시작됨

설명	
B4	충전을 위한 모든 전제 조건이 충족되었지만 EV가 충전을 시작하지 않음
B5	충전을 위한 모든 전제 조건이 충족되었지만 EVSE가 충전을 허용하지 않음
B6	시간 초과됨. 사용이 시작되었지만(예: 플러그 삽입, 충전 공간 점유) idTag가 제한 시간 내에 제공되지 않았습니다.
B9	추가 충전 작업을 방해하는 오류가 감지됨
C1	사용자 작업이 필요하지 않은 상태에서 충전 세션이 종료됨(예: EV 측에서 고정 케이블이 제거됨)
C4	EV 요청 시 충전이 중단됨(예: S2가 열림)
C5	EVSE 요청 시 충전이 중단됨(예: 스마트 충전 제한, <code>StartTransaction.conf</code> 에서 <code>AuthorizationStatus</code> 에 의해 트랜잭션이 무효화됨)
C6	사용자 또는 <code>Remote Stop Transaction</code> 메시지에 의해 트랜잭션이 중단되고 추가 사용자 작업이 필요함(예: 케이블 제거, 주차 공간 비우기)
C8	충전 세션이 종료되고 사용자 작업이 필요하지 않으며 커넥터가 사용 불가 상태가 되도록 예약됨
C9	추가 충전 작업을 방해하는 오류가 감지됨
D1	사용자 작업이 필요하지 않은 상태에서 충전 세션이 종료됨
D3	EV 요청 시 충전이 재개됨(예: S2가 닫힘)
D5	EVSE에 의해 충전이 중단됨(예: 스마트 충전 제한으로 인해)
D6	트랜잭션이 중단되었으며 추가 사용자 작업이 필요함
D8	충전 세션이 종료되고 사용자 작업이 필요하지 않으며 커넥터가 사용 불가능 상태가 되도록 예약됨
D9	추가 충전 작업을 방해하는 오류가 감지됨
E1	사용자 작업이 필요하지 않은 상태에서 충전 세션이 종료됨
E3	EVSE 제한이 해제되어 충전이 재개됨
E4	EVSE 제한이 해제되었지만 EV가 충전을 시작하지 않음
E6	트랜잭션이 중단되었으며 추가 사용자 작업이 필요함
E8	충전 세션이 종료되고 사용자 작업이 필요하지 않으며 커넥터가 사용 불가능 상태가 되도록 예약됨
E9	추가 충전 작업을 방해하는 오류가 감지됨
F1	모든 사용자 작업이 완료됨

설명	
F2	사용자가 충전 세션을 다시 시작(예: 케이블 재연결, idTag 다시 제시)하여 새 트랜잭션을 생성
F8	모든 사용자 작업이 완료되고 커넥터가 사용 불가능 상태가 되도록 예약됨
F9	추가 충전 작업을 방해하는 오류가 감지됨
G1	예약이 만료되거나 Cancel Reservation 메시지가 수신됨
G2	예약 ID가 제시됨
G8	예약이 만료되거나 Cancel Reservation 메시지가 수신되고 커넥터 _사용 불가능_이 되도록 예약됨
G9	추가 충전 작업을 방해하는 오류가 감지됨
H1	커넥터가 Change Availability 메시지에 의해 사용 가능 으로 설정됨
H2	사용자가 충전기와 상호 작용한 후 커넥터가 사용 가능 으로 설정됨
H3	커넥터가 사용 가능 으로 설정되었으며 충전을 시작하기 위해 사용자 작업이 필요하지 않음
H4	H3와 유사하지만 EV가 충전을 시작하지 않음
H5	H3와 유사하지만 EVSE가 충전을 허용하지 않음
H9	추가 충전 과정을 방해하는 오류가 감지됨
I1-I8	오류가 해결되고 상태가 오류 전 상태로 돌아갑니다.



충전기 커넥터는 위 표에 표시된 9가지 상태 중 하나를 가질 수 있습니다. ConnectorId 0의 경우 Available, Unavailable, Faulted 라는 제한된 집합만 적용됩니다. ConnectorId 0의 상태는 개별 커넥터의 상태와 직접 관련이 없습니다(>0).



EV와 EVSE 모두에서 충전이 중단된 경우 상태 *SuspendedEVSE* 가 상태 *SuspendedEV* 보다 우선합니다.



Change Availability 명령으로 충전기 또는 커넥터가 사용 불가능 상태로 설정된 경우, 'Unavailable' 상태는 재부팅 시에도 지속되어야 합니다. 충전기는 *Unavailable* 상태를 다른 목적(예: 펌웨어 업데이트 중 또는 초기 **Accepted RegistrationStatus** 대기 중)으로 내부적으로 사용할 수 있습니다.

Occupied 상태가 5개의 새로운 상태(*Preparing*, *Charging*, *SuspendedEV*, *SuspendedEVSE* 및 *Finishing*)로 분할되었으므로 더 많은 **StatusNotification.req** PDU가 충전기에서 CSMS로 전송됩니다. 예를 들어, 트랜잭션이 시작되면 커넥터 상태가 *Preparing* 에서 *Charging* 으로 차례로 변경되고 그 사이에 짧은 *SuspendedEV* 및/또는 *SuspendedEVSE* 가 표시되며 몇 초 이내에 변경됩니다.

전환 횟수를 제한하기 위해 충전기는 선택적 구성 키 **MinimumStatusDuration**에서 정의된 시간보다 짧은 시간 동안 활성화된 경우 **StatusNotification.req** 전송을 생략할 수 있습니다. 이런 방식으로 충전기는 특정 **StatusNotification.req** PDU를 전송하지 않기로 선택할 수 있습니다.



충전기 제조업체는 **MinimumStatusDuration** 설정과 별도로 특정 상태 전환에 대한 최소 상태 기간을 구현했을 수 있습니다. **MinimumStatusDuration**에 설정된 시간이 이 기본 지연에 추가됩니다. **MinimumStatusDuration**을 0으로 설정해도 기본 제조업체의 최소 상태 기간이 무시되지 않습니다.



MinimumStatusDuration 시간을 높게 설정하면 모든 StatusNotification이 지연되어 전송될 수 있습니다. 충전기는 **MinimumStatusDuration** 시간이 지난 후에만 StatusNotification.req를 전송하기 때문입니다.

충전기는 StatusNotification.req PDU를 보내 CSMS에 오류 상태를 알릴 수 있습니다. 'status' 필드가 *Faulted* 가 아닌 경우, 충전 작업이 여전히 가능하므로 해당 상태는 경고로 간주해야 합니다.



ChargePointErrorCode EVCommunicationError 는 Preparing, SuspendedEV, SuspendedEVSE 및 Finishing 상태에서만 사용해야 하며 경고로 처리해야 합니다.

충전기가 StopTransactionOnEVSideDisconnect가 *false* 로 설정된 상태로 구성되고, 트랜잭션이 실행 중이고 EV가 EV 측에서 연결이 끊어지면, StatusNotification.req가 상태가 *SuspendedEV* 인 상태로 CSMS로 전송되어야 하며, 'errorCode' 필드는 'NoError'로 설정되어야 합니다. 충전기는 'info' 필드에 추가 정보를 추가해야 하며, CSMS에 중단 사유를 알려야 합니다: 'EV side disconnected'. 현재 트랜잭션은 중단되지 않습니다.

StopTransactionOnEVSideDisconnect가 *true* 로 설정된 충전기가 구성되고 트랜잭션이 실행 중이고 EV 측에서 EV 연결이 끊어지면, 상태가 'Finishing'인 StatusNotification.req가 CSMS로 전송되어야 하며, 'errorCode' 필드는 'NoError'로 설정되어야 합니다. 충전기는 'info' 필드에 추가 정보를 추가해야 하며, CSMS에 중단 사유를 알려야 합니다: 'EV side disconnected'. 현재 트랜잭션은 중단됩니다.

충전기가 *offline* 상태가 된 후 CSMS에 연결되면 다음 규칙에 따라 CSMS에 상태를 업데이트합니다.

1. 충전기는 충전기가 오프라인 상태일 때 상태가 변경된 경우 현재 상태와 함께 StatusNotification.req PDU를 보내야 합니다.
2. 충전기는 충전기가 오프라인 상태일 때 발생한 오류를 보고하기 위해 StatusNotification.req PDU를 보낼 수 있습니다.
3. 충전기는 충전기가 오프라인 상태일 때 발생하고 CSMS에 충전기 오류나 충전기의 현재 상태를 알리지 않는 이전 상태 변경 이벤트에 대해 StatusNotification.req PDU를 보내서는 안 됩니다.
4. StatusNotification.req 메시지는 설명된 이벤트가 발생한 순서대로 보내야 합니다.

StatusNotification.req PDU를 수신하면 CSMS는 StatusNotification.conf PDU로 응답해야 합니다.

4.10. Stop Transaction

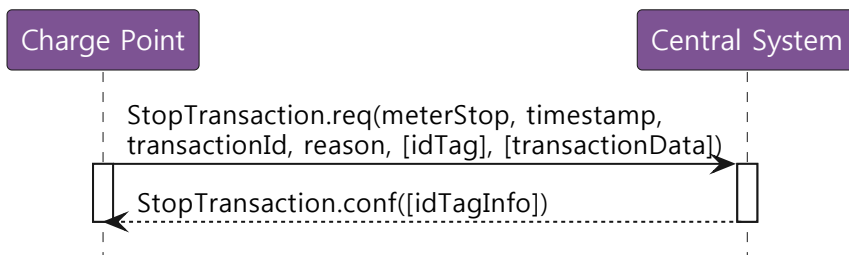


Figure 20. 시퀀스 다이어그램: 트랜잭션 중지

트랜잭션이 중단되면 충전기는 StopTransaction.req PDU를 보내 CSMS에 트랜잭션이 중지되었음을 알려야 합니다.

StopTransaction.req PDU에는 트랜잭션 사용에 대한 자세한 내용을 제공하기 위한 선택적 TransactionData 요소가 포함될 수 있습니다. 선택적인 TransactionData 요소는 MeterValues의 모든 수에 대한 컨테이너이며, MeterValues.req PDU의 meterValue 요소와 동일한 데이터 구조를 사용합니다(섹션 MeterValues 참조)

StopTransaction.req PDU를 수신하면 CSMS는 StopTransaction.conf PDU로 응답해야 합니다.



ChargePointErrorCode EVCommunicationError 는 Preparing, SuspendedEV, SuspendedEVSE 및 Finishing 상태에서만 사용해야 하며 경고로 처리해야 합니다.

CSMS는 트랜잭션이 중단되는 것을 막을 수 없습니다. 충전기에 StopTransaction.req를 수신했다는 사실만 알리고 트랜잭션을 중단하는 데 사용된 idTag에 대한 정보를 보낼 수 있습니다. 이 정보는 구현된 경우 Authorization Cache를 업데이트하는 데 사용해야 합니다.

요청 PDU의 idTag는 충전기 자체가 트랜잭션을 중단해야 하는 경우 생략할 수 있습니다. 예를 들어, 충전기에 재설정을 요청하는 경우입니다.

트랜잭션이 정상적인 방식으로 종료된 경우(예: EV 운전자가 트랜잭션을 중단하기 위해 식별자를 제시한 경우) Reason 요소를 생략할 수 있으며 Reason을 'Local'로 가정해야 합니다. 트랜잭션이 정상적으로 종료되지 않은 경우 Reason을 올바른 값으로 설정해야 합니다. 정상적인 트랜잭션 종료의 일부로 충전기는 케이블의 잠금을 해제해야 합니다(영구적으로 연결되지 않은 경우).

충전기는 케이블이 EV에서 분리될 때 케이블의 잠금을 해제할 수 있습니다(영구적으로 연결되지 않은 경우). 지원되는 경우 이 기능은 구성 키 UnlockConnectorOnEVSideDisconnect에서 보고되고 제어됩니다.

충전기는 케이블이 EV에서 분리될 때 실행 중인 트랜잭션을 중지할 수 있습니다. 지원되는 경우 이 기능은 구성 키 StopTransactionOnEVSideDisconnect에 의해 보고되고 제어됩니다.

StopTransactionOnEVSideDisconnect가 false로 설정된 경우 케이블이 EV에서 분리될 때 트랜잭션이 중지되지 않아야 합니다. EV가 다시 연결되면 에너지 전송이 다시 허용됩니다. 이 경우 동일한 진행 중인 트랜잭션 중에 다른 EV가 충전되고 분리되는 것을 방지하는 메커니즘이 없습니다. UnlockConnectorOnEVSideDisconnect가 false로 설정된 경우 커넥터는 사용자가 식별자를 제시할 때까지 충전기에서 잠긴 상태로 유지되어야 합니다.

StopTransactionOnEVSideDisconnect를 true로 설정하면 케이블이 EV에서 분리될 때 트랜잭션이 중지되어야 합니다. EV가 다시 연결되면 트랜잭션이 중지되고 새 트랜잭션이 시작될 때까지 에너지 전송이 허용되지 않습니다. UnlockConnectorOnEVSideDisconnect를 true로 설정하면 충전기의 커넥터도 잠금 해제됩니다.



StopTransactionOnEVSideDisconnect가 false로 설정된 경우, 이는 UnlockConnectorOnEVSideDisconnect보다 우선합니다. 즉, StopTransactionOnEVSideDisconnect가 false일 때 EV 측에서 케이블이 분리되면 케이블은 항상 잠긴 상태로 유지됩니다.



StopTransactionOnEVSideDisconnect를 true로 설정하면 EV 측에서 잠기지 않은 케이블을 뽑아 에너지 흐름을 중단하는 방해 행위를 방지할 수 있습니다.

CSMS가 수신한 StopTransaction.req에 포함된 데이터에 대한 건전성 검사를 적용할 가능성이 있습니다. 이러한 건전성 검사의 결과로 인해 CSMS가 StopTransaction.conf로 응답하지 않는 일이 발생해서는 안 됩니다. StopTransaction.conf로 응답하지 않으면 충전기가 Error responses to transaction-related messages에 지정된 대로 동일한 메시지를 다시 시도할 뿐입니다.

충전기에서 권한 부여 캐시를 구현한 경우 StopTransaction.conf PDU를 수신하면 충전기는 idTag가 Local Authorization List에 없으면 Authorization Cache에서 설명한 대로 응답의 IdTagInfo 값으로 캐시 항목을 업데이트해야 합니다.

Chapter 5. Operations Initiated by Central System

5.1. Cancel Reservation



Figure 21. 시퀀스 다이어그램: 예약 취소

CSMS는 예약을 취소하기 위해 `CancelReservation.req` PDU를 충전기로 보내야 합니다.

충전기에 요청 PDU의 `reservationId`와 일치하는 예약이 있는 경우 'Accepted' 상태를 반환해야 합니다. 그렇지 않으면 'Rejected'를 반환해야 합니다.

5.2. Change Availability



Figure 22. 시퀀스 다이어그램: 가용성 변경

CSMS는 충전기에 가용성을 변경하도록 요청할 수 있습니다. 충전기는 충전 중이거나 충전 준비가 되면 사용 가능한("작동 중") 것으로 간주됩니다. 충전기는 충전을 허용하지 않으면 사용할 수 없는 것으로 간주됩니다. CSMS는 충전기의 가용성을 변경하도록 요청하기 위해 `ChangeAvailability.req` PDU를 보내야 합니다. CSMS는 가용성을 사용 가능 또는 사용 불가능으로 변경할 수 있습니다.

`ChangeAvailability.req` PDU를 수신하면 충전기는 `ChangeAvailability.conf` PDU로 응답해야 합니다. 응답 PDU는 충전기가 요청한 가용성으로 변경할 수 있는지 여부를 나타내야 합니다. 트랜잭션이 진행 중이면 충전기는 트랜잭션이 완료된 후 발생하도록 예약되었음을 나타내기 위해 가용성 상태 'Reserved'으로 응답해야 합니다.

CSMS가 충전기에 이미 있는 상태로 변경하도록 요청하는 경우 충전기는 가용성 상태 'Accepted'으로 응답해야 합니다.

`ChangeAvailability.req` PDU로 요청된 가용성 변경이 발생하면 충전기는 해당 설명에 따라 `StatusNotification.req`로 CSMS에 새로운 가용성 상태를 알려야 합니다.



`ChangeAvailability.req`에 `ConnectorId = 00`이 포함된 경우 상태 변경은 충전기와 모든 커넥터에 적용됩니다.



지속 상태: 예를 들어, 커넥터를 사용할 수 없음으로 설정하면 재부팅이 지속됩니다.

5.3. Change Configuration

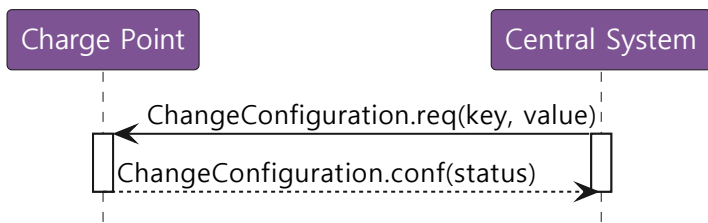


Figure 23. 시퀀스 다이어그램: 구성 변경

CSMS는 충전기에 구성 매개변수를 변경하도록 요청할 수 있습니다. 이를 위해 CSMS는 `ChangeConfiguration.req`를 보내야 합니다. 이 요청에는 키-값 쌍이 포함되어 있으며, 여기서 "키"는 변경할 구성 설정의 이름이고 "값"은 구성 설정에 대한 새 설정을 포함합니다.

충전기가 `ChangeConfiguration.req`를 수신하면 해당 구성에 변경 사항을 적용할 수 있었는지 여부를 나타내는 `ChangeConfiguration.conf`로 응답해야 합니다. "키"와 "값"의 내용은 규정되지 않습니다. 충전기는 다음 규칙에 따라 `ChangeConfiguration.conf`의 상태 필드를 설정해야 합니다.

- 변경 사항이 성공적으로 적용되고 변경 사항이 즉시 적용되는 경우 충전기는 'Accepted' 상태로 응답해야 합니다.
- 변경 사항이 성공적으로 적용되었지만 적용하려면 재부팅이 필요한 경우 충전기는 'RebootRequired' 상태로 응답해야 합니다.
- "키"가 충전기에서 지원하는 구성 설정과 일치하지 않는 경우 'NotSupported' 상태로 응답해야 합니다.
- 충전기가 구성을 설정하지 않았고 이전 상태 중 어느 것도 적용되지 않는 경우 충전기는 'Rejected' 상태로 응답해야 합니다.



충전기가 'Rejected' 상태의 `ChangeConfiguration.conf`로 응답하는 변경 구성 요청의 예로는 범위를 벗어난 값이 있는 요청과 예상 형식에 맞지 않는 값이 있는 요청이 있습니다.

키 값이 CSL로 정의된 경우, 항목에서 CSL의 최대 길이를 나타내는 `[KeyName]MaxLength` 키와 함께 제공될 수 있습니다. 이 키가 설정되지 않은 경우 안전한 값인 1(하나) 항목을 가정해야 합니다.

5.4. Clear Cache

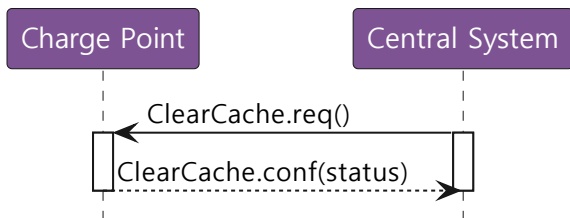


Figure 24. 시퀀스 다이어그램: 캐시 지우기

CSMS는 충전기에 `Authorization Cache`를 지우도록 요청할 수 있습니다. CSMS는 충전기의 인증 캐시를 지우기 위해 `ClearCache.req` PDU를 보내야 합니다.

`ClearCache.req` PDU를 수신하면 충전기는 `ClearCache.conf` PDU로 응답해야 합니다. 응답 PDU는 충전기가 인증 캐시를 지울 수 있었는지 여부를 나타냅니다.

5.5. Clear Charging Profile

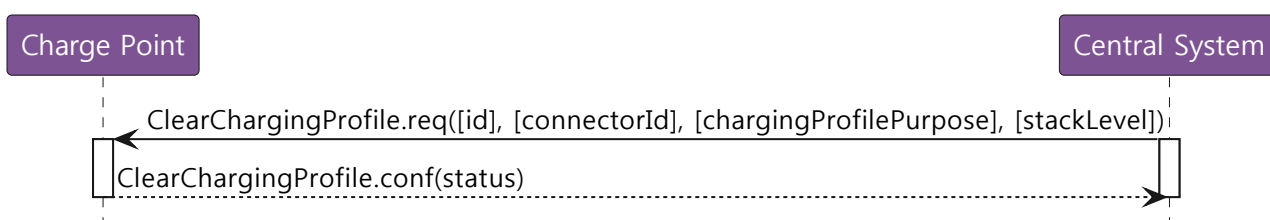


Figure 25. 시퀀스 다이어그램: 충전 프로파일 지우기

CSMS가 이전에 충전기에 전송된 일부 또는 모든 충전 프로파일을 지우려면 `ClearChargingProfile.req` PDU를 사용해야 합니다.

충전기는 요청을 처리할 수 있었는지 여부를 지정하는 `ClearChargingProfile.conf` PDU로 응답해야 합니다.

5.6. Data Transfer

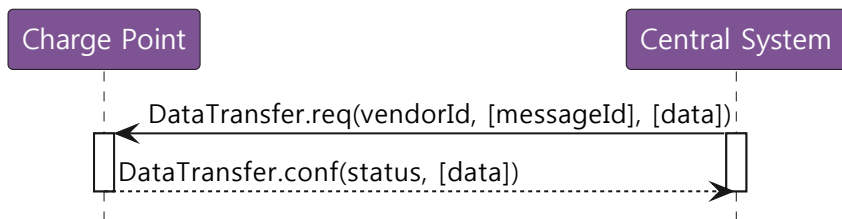


Figure 26. 시퀀스 다이어그램: 데이터 전송

CSMS가 OCPP에서 지원하지 않는 기능에 대한 정보를 충전소로 보내야 하는 경우, `DataTransfer.req` PDU를 사용해야 합니다.

이 작업의 동작은 충전기에서 시작한 데이터 전송 작업과 동일합니다. 자세한 내용은 [Data Transfer](#)를 참조하세요.

5.7. Get Composite Schedule

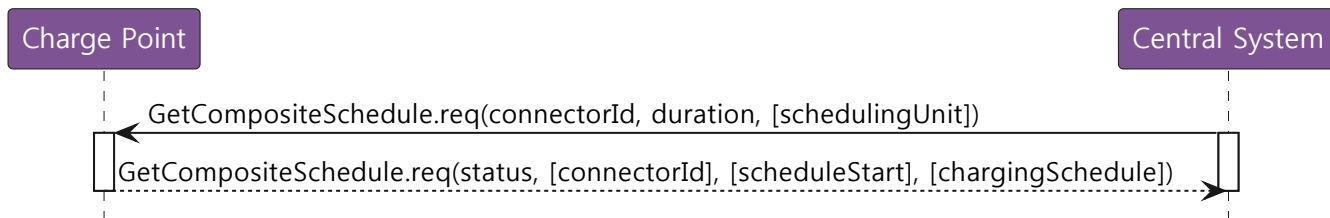


Figure 27. 시퀀스 다이어그램: 복합 일정 가져오기

CSMS는 `GetCompositeSchedule.req` PDU를 보내 충전기에 복합 충전 일정을 보고하도록 요청할 수 있습니다. `GetCompositeSchedule.conf` PDU에 보고된 일정은 충전기에 있는 모든 활성 일정과 가능한 지역 제한을 계산한 결과입니다. 지역 제한이 고려될 수 있습니다.

`GetCompositeSchedule.req`를 수신하면 충전기는 요청 PDU를 수신한 순간부터 시간 X에서 X + 기간까지 복합 충전 일정 간격을 계산하여 `GetCompositeSchedule.conf` PDU에 넣어 CSMS로 전송해야 합니다.

요청의 ConnectorId가 '0'으로 설정된 경우 충전기는 요청된 기간 동안 충전기가 그리드에서 소비할 것으로 예상하는 총 예상 전력 또는 전류를 보고해야 합니다.



충전기에서 보낸 충전 일정은 해당 시점에만 표시된다는 점에 유의하세요. 이 일정은 외부 원인(예: 그리드 연결 용량에 따른 로컬 밸런싱이 활성화되고 하나의 커넥터를 사용할 수 있게 됨)으로 인해 시간이 지남에 따라 변경될 수 있습니다.

충전기에서 요청된 일정을 보고할 수 없는 경우(예: connectorId를 알 수 없는 경우)에는 거부됨(Rejected) 상태로 응답해야 합니다.

5.8. Get Configuration

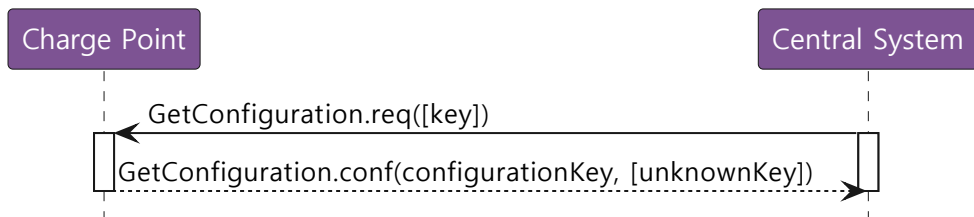


Figure 28. 시퀀스 다이어그램: 구성 가져오기

구성 설정 값을 검색하기 위해 CSMS는 `GetConfiguration.req` PDU를 충전기로 보내야 합니다.

요청 PDU의 키 목록이 비어 있거나 누락된 경우(선택 사항), 충전기는 `GetConfiguration.conf`에 있는 모든 구성 설정 목록을 반환해야 합니다. 그렇지 않으면 충전기는 인식된 키와 해당 값 및 읽기 전용 상태 목록을 반환해야 합니다. 인식되지 않은 키는 `GetConfiguration.conf`의 선택적 unknown key list 요소의 일부로 응답 PDU에 배치해야 합니다.

단일 PDU에서 요청된 구성 키 수는 충전기에 의해 제한될 수 있습니다. 이 최대값은 구성 키 `GetConfigurationMaxKeys`를 읽어서 검색할 수 있습니다.

5.9. Get Diagnostics

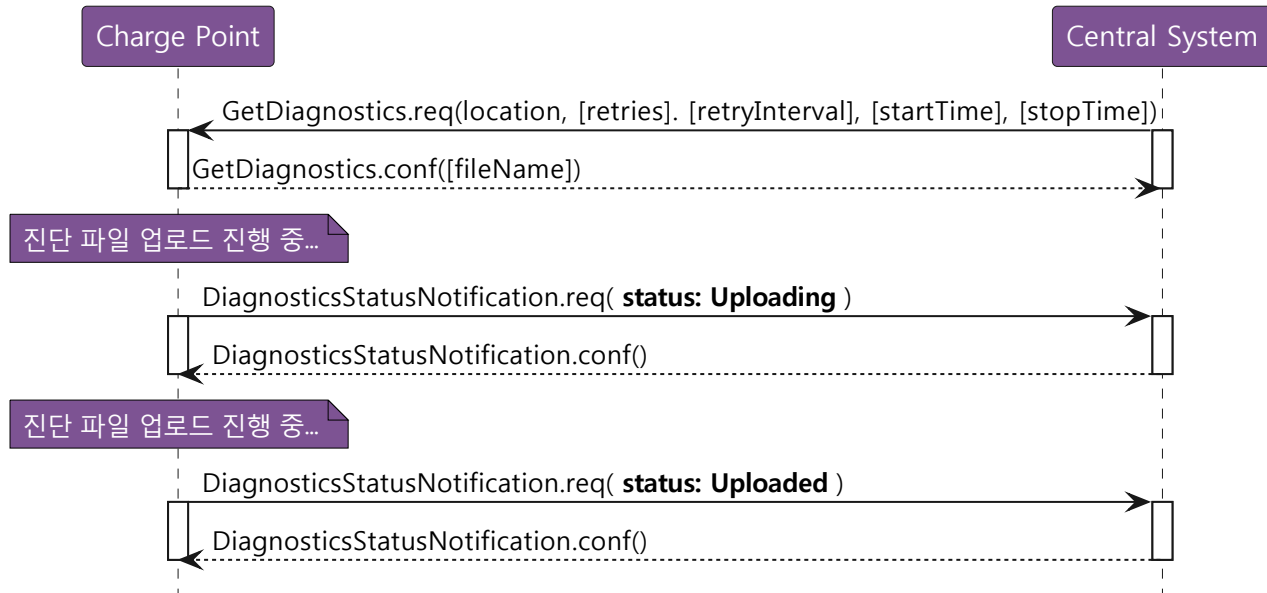


Figure 29. 시퀀스 다이어그램: 진단 받기

CSMS는 충전기에 진단 정보를 요청할 수 있습니다. CSMS는 충전기의 진단 정보를 얻기 위해 `GetDiagnostics.req` PDU를 보내야 하며, 충전기는 진단 데이터를 업로드해야 하는 위치와 요청된 진단 정보의 시작 및 종료 시간을 선택적으로 포함해야 합니다.

`GetDiagnostics.req` PDU를 수신하고 진단 정보를 사용할 수 있는 경우 충전기는 `GetDiagnostics.conf` PDU로 응답해야 하며, 여기에는 업로드될 진단 정보가 포함된 파일의 이름이 명시되어야 합니다. 충전기는 단일 파일을 업로드해야 합니다. 진단 파일의 형식은 규정되지 않습니다. 진단 파일을 사용할 수 없는 경우, `GetDiagnostics.conf`에는 파일 이름이 포함되지 않아야 합니다.

진단 파일을 업로드하는 동안, 충전기는 `DiagnosticsStatusNotification.req` PDU를 보내 CSMS에 업로드 프로세스 상태를 최신 상태로 유지해야 합니다.

5.10. Get Local List Version



Figure 30. 시퀀스 다이어그램: 로컬 목록 버전 가져오기

`Local Authorization List`의 동기화를 지원하기 위해, CSMS는 충전기에 로컬 인증 목록 버전 번호를 요청할 수 있습니다. CSMS는 이 값을 요청하기 위해 `GetLocalListVersion.req` PDU를 보내야 합니다.

`GetLocalListVersion.req` PDU를 수신하면 충전소는 로컬 인증 목록의 버전 번호가 포함된 `GetLocalListVersion.conf` PDU로 응답해야 합니다. 로컬 인증 목록이 비어 있음을 나타내는 버전 번호 0(영)을 사용하고 충전소가 로컬 인증 목록을 지원하지 않음을 나타내는 버전 번호 -1을 사용해야 합니다.

5.11. Remote Start Transaction

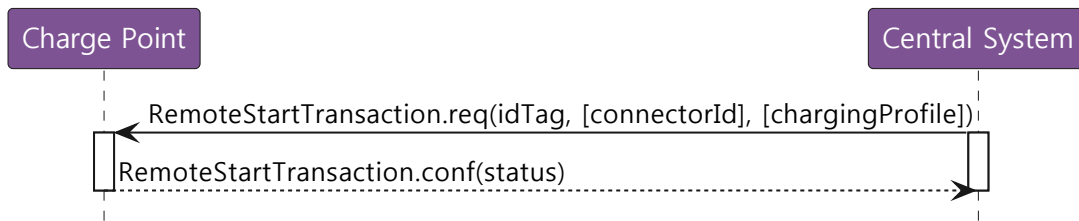


Figure 31. 시퀀스 다이어그램: 원격 트랜잭션 시작

CSMS는 `RemoteStartTransaction.req`를 보내서 충전기에 트랜잭션을 시작하도록 요청할 수 있습니다. 수신 시, 충전기는 `RemoteStartTransaction.conf`와 요청을 수락했는지 여부를 나타내는 상태로 응답해야 하며 트랜잭션을 시작하려고 시도합니다.

`RemoteStartTransaction.req` 메시지의 효과는 충전기의 `AuthorizeRemoteTxRequests` 구성 키 값에 따라 달라집니다.

- `'AuthorizeRemoteTxRequests'`의 값이 `true` 이면, 충전기는 `RemoteStartTransaction.req` 메시지에 제공된 `idTag`로 트랜잭션을 시작하기 위해 충전기에서 로컬 작업에 대한 응답으로 동작해야 합니다. 즉, 충전기는 먼저 `Local Authorization List`, `Authorization Cache` 및/또는 `Authorize.req` 요청을 사용하여 `idTag`를 승인하려고 시도합니다. 승인을 획득한 후에만 트랜잭션이 시작됩니다.
- `'AuthorizeRemoteTxRequests'`의 값이 `false` 이면, 충전기는 `RemoteStartTransaction.req` 메시지에 제공된 `idTag`에 대한 트랜잭션을 즉시 시작하려고 시도해야 합니다. 트랜잭션이 시작된 후 충전기는 CSMS에 `StartTransaction` 요청을 보내고 CSMS는 이 `StartTransaction` 요청을 처리할 때 `idTag`의 승인 상태를 확인합니다.

다음은 Remote Start Transaction의 일반적인 사용 사례입니다.

- CPO 운영자가 트랜잭션을 시작하는 데 문제가 있는 EV 운전자를 도울 수 있도록 합니다.
- 모바일 앱이 CSMS를 통해 충전 트랜잭션을 제어할 수 있도록 합니다.
- SMS를 사용하여 CSMS를 통해 충전 트랜잭션을 제어할 수 있도록 합니다.

`RemoteStartTransaction.req`에는 식별자(`idTag`)가 포함되어야 하며, 충전기는 트랜잭션을 시작할 수 있는 경우 이를 사용하여 CSMS로 `StartTransaction.req`를 보냅니다. 트랜잭션은 `StartTransaction`에서 설명한 것과 같은 방식으로 시작됩니다. `RemoteStartTransaction.req`에는 특정 커넥터에서 트랜잭션을 시작해야 하는 경우 커넥터 ID가 포함될 수 있습니다. 커넥터 ID가 제공되지 않으면 충전기가 커넥터 선택을 제어합니다. 충전기는 커넥터 ID가 없는 `RemoteStartTransaction.req`를 거부할 수 있습니다.

CSMS는 `RemoteStartTransaction` 요청에 `ChargingProfile`을 포함할 수 있습니다. 이 `ChargingProfile`의 목적은 `TxProfile`로 설정해야 합니다. 수락된 경우 충전기는 이 `ChargingProfile`을 트랜잭션에 사용해야 합니다.



스마트 충전을 지원하지 않는 충전기가 충전 프로파일이 있는 `RemoteStartTransaction.req`를 수신하는 경우 이 매개변수는 무시해야 합니다.

5.12. Remote Stop Transaction

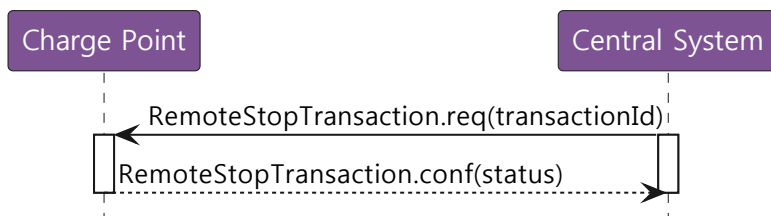


Figure 32. 원격 트랜잭션 중지

CSMS는 충전기에 트랜잭션 식별자와 함께 `RemoteStopTransaction.req`를 충전기로 보내 트랜잭션을 중지하도록 요청할 수 있습니다. 충전기는 `RemoteStopTransaction.conf`와 요청을 수락했는지 여부와 주어진 `transactionId`를 가진 트랜잭션이 진행 중이며 중지될 것인지를 나타내는 상태로 응답해야 합니다.

트랜잭션을 중지하라는 이 원격 요청은 트랜잭션을 중지하는 로컬 작업과 동일합니다. 따라서 트랜잭션은 중지되어야 합니다. 충전기는 `StopTransaction.req`를 보내고 해당되는 경우 커넥터를 잠금 해제해야 합니다.

다음 두 가지 예시는 원격 트랜잭션 중지의 주요 사용 사례입니다.

- CPO 운영자가 트랜잭션을 중지하는 데 문제가 있는 EV 운전자를 도울 수 있도록 합니다.
- 모바일 앱이 CSMS를 통해 충전 트랜잭션을 제어할 수 있도록 합니다.

5.13. Reserve Now

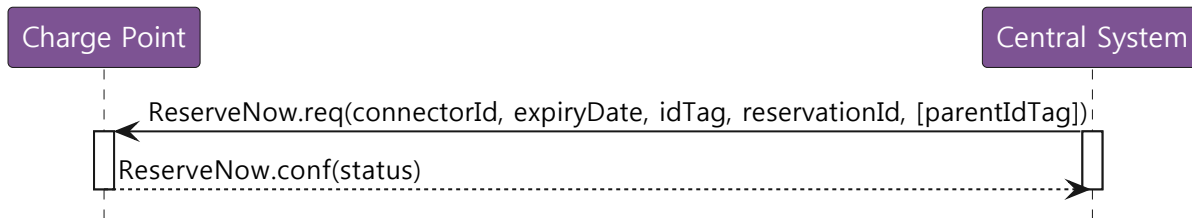


Figure 33. 시퀀스 다이어그램: 지금 예약하기

CSMS는 특정 idTag에서 사용할 커넥터를 예약하기 위해 충전기에 `ReserveNow.req`를 발급할 수 있습니다.

예약 요청하려면 CSMS가 충전기에 `ReserveNow.req` PDU를 보내야 합니다. CSMS는 예약할 커넥터를 지정할 수 있습니다. `ReserveNow.req` PDU를 수신하면 충전기는 `ReserveNow.conf` PDU로 응답해야 합니다.

요청의 `reservationId`가 충전기의 예약과 일치하는 경우 충전기는 해당 예약을 요청의 새 예약으로 대체해야 합니다.

`reservationId`가 충전기의 어떤 예약과도 일치하지 않는 경우 충전기는 커넥터를 예약하는 데 성공하면 상태 값 'Accepted'를 반환해야 합니다. 충전기는 충전기 또는 지정된 커넥터가 점유된 경우 'Occupied'를 반환해야 합니다. 충전기는 충전기 또는 커넥터가 동일하거나 다른 idTag에 예약된 경우에도 'Occupied'를 반환해야 합니다. 충전기는 충전기 또는 커넥터가 Faulted 상태인 경우 'Faulted'를 반환해야 합니다. 충전기는 충전기 또는 커넥터가 Unavailable 상태인 경우 'Unavailable'을 반환해야 합니다. 충전기는 예약을 수락하지 않도록 구성된 경우 'Rejected'를 반환해야 합니다.

충전기가 예약 요청을 수락하면 예약된 커넥터에서 들어오는 모든 idTag에 대한 요금 청구를 거부해야 합니다. 들어오는 idTag 또는 부모 idTag가 예약의 idTag 또는 부모 idTag와 일치하는 경우는 예외입니다.

구성 키: `ReserveConnectorZeroSupported`가 `true`로 설정된 경우 충전기는 커넥터 0에서 예약을 지원합니다. 예약 요청의 `connectorId`가 0인 경우 충전기는 특정 커넥터를 예약하지 않아야 하지만, 예약의 유효 기간 동안 언제든지 예약된 idTag에 대해 하나의 커넥터를 사용할 수 있도록 해야 합니다. 구성 키: `ReserveConnectorZeroSupported`가 설정되지 않았거나 `false`로 설정된 경우 충전기는 'Rejected'를 반환해야 합니다.

예약에 부모 idTag에 값이 있는 경우(선택 사항), 들어오는 idTag와 연관된 부모 idTag를 확인하기 위해 충전기는 로컬 인증 목록 또는 인증 캐시에서 조회할 수 있습니다. 로컬 인증 목록 또는 인증 캐시에서 찾을 수 없는 경우 충전기는 들어오는 idTag에 대한 `Authorize.req`를 CSMS로 보내야 합니다. `Authorize.conf` 응답에는 부모 id가 포함될 수 있습니다.

예약은 (1) 예약된 idTag 또는 부모 idTag에 대한 트랜잭션이 시작되고 예약된 커넥터 또는 예약된 `connectorId`가 0인 모든 커넥터에서 시작되거나 (2) `expiryDate`에 지정된 시간에 도달하거나 (3) 충전기 또는 커넥터가 'Faulted' 또는 'Unavailable'으로 설정될 때 충전소에서 종료되어야 합니다.

예약된 idTag에 대한 트랜잭션이 시작되면 충전기는 `StartTransaction.req` PDU(`Start Transaction` 참조)에서 `reservationId`를 보내 CSMS에 예약이 종료되었음을 알립니다.

예약이 만료되면 충전기는 예약을 종료하고 커넥터를 사용 가능하게 합니다. 충전기는 CSMS에 예약된 커넥터를 사용할 수 있음을 알리는 상태 알림을 보냅니다.

충전기가 인증 캐시를 구현한 경우 `ReserveNow.conf` PDU를 수신하면 충전기는 idTag가 `Local Authorization List`에 없으면 `IdTagInfo` 값을 `Authorization Cache`에 설명된 대로 응답에서 업데이트합니다.



ReserveNow.req를 수신한 후와 트랜잭션 시작 전에 authorize.req로 식별자를 검증하는 것이 좋습니다.

5.14. Reset

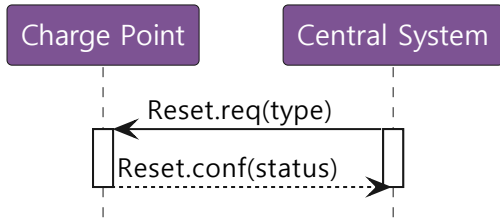


Figure 34. 시퀀스 다이어그램: 재설정

CSMS는 충전기가 스스로 재설정하도록 요청하기 위해 **Reset.req** PDU를 보내야 합니다. CSMS는 하드 또는 소프트 리셋을 요청할 수 있습니다. **Reset.req** PDU를 수신하면 충전기는 **Reset.conf** PDU로 응답해야 합니다. 응답 PDU에는 충전기가 자체적으로 재설정을 시도할지 여부가 포함되어야 합니다.

Reset.req를 수신한 후 충전기는 재설정을 수행하기 전에 진행 중인 모든 트랜잭션에 대해 **StopTransaction.req**를 보내야 합니다. 충전기가 CSMS에서 **StopTransaction.conf**를 수신하지 못하면 **StopTransaction.req**를 대기열에 넣어야 합니다.

소프트 리셋을 수신하면 충전기는 진행 중인 트랜잭션을 정상적으로 중지하고 진행 중인 모든 트랜잭션에 대해 **StopTransaction.req**를 보내야 합니다. 그런 다음 애플리케이션 소프트웨어를 다시 시작해야 합니다(가능하면 프로세서/컨트롤러를 다시 시작해야 함).

하드 리셋을 받으면 충전기는 (모든) 하드웨어를 다시 시작해야 하며, 진행 중인 트랜잭션을 정상적으로 중지할 필요는 없습니다. 가능하다면 충전기는 **BootNotification.conf**를 통해 재시작하고 CSMS에서 수락한 후 이전에 진행 중이던 트랜잭션에 대해 **StopTransaction.req**를 보냅니다. 이것은 "하드" 리셋을 보내면 올바르게 작동하지 않는 충전기에 대한 최후의 해결책이며, (대기 중인) 정보가 손실될 수 있습니다.



지속 상태: 예를 들어, "Unavailable"로 설정된 커넥터 상태는 지속되어야 합니다.

5.15. Send Local List

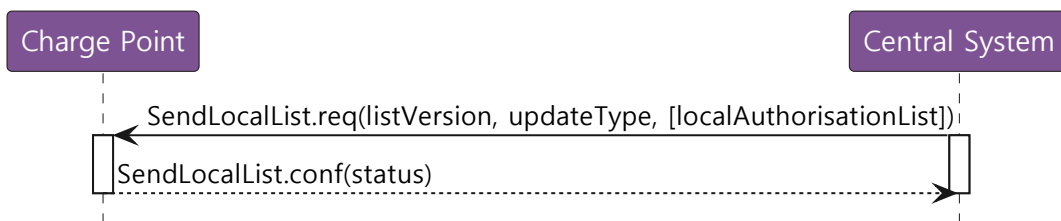


Figure 35. 시퀀스 다이어그램: 로컬 목록 전송

CSMS는 충전기가 idTags의 권한 부여에 사용할 수 있는 **Local Authorization List**를 전송할 수 있습니다. 목록은 충전기의 현재 목록을 대체하는 전체 목록이거나 충전기의 현재 목록에 적용할 업데이트가 있는 차등 목록(differential list)일 수 있습니다.

CSMS는 충전기로 목록을 전송하기 위해 **SendLocalList.req** PDU를 전송해야 합니다. **SendLocalList.req** PDU에는 업데이트 유형(전체 또는 차등)과 충전기가 업데이트된 후 로컬 인증 목록과 연관시켜야 하는 버전 번호가 포함되어야 합니다.

SendLocalList.req PDU를 수신하면 충전기는 **SendLocalList.conf** PDU로 응답해야 합니다. 응답 PDU는 충전기가 로컬 인증 목록의 업데이트를 수락했는지 여부를 나타내야 합니다. 상태가 실패 또는 버전 불일치이고 updateType이 차등인 경우 CSMS는 updateType Full로 전체 로컬 인증 목록을 다시 보내야 합니다.

5.16. Set Charging Profile

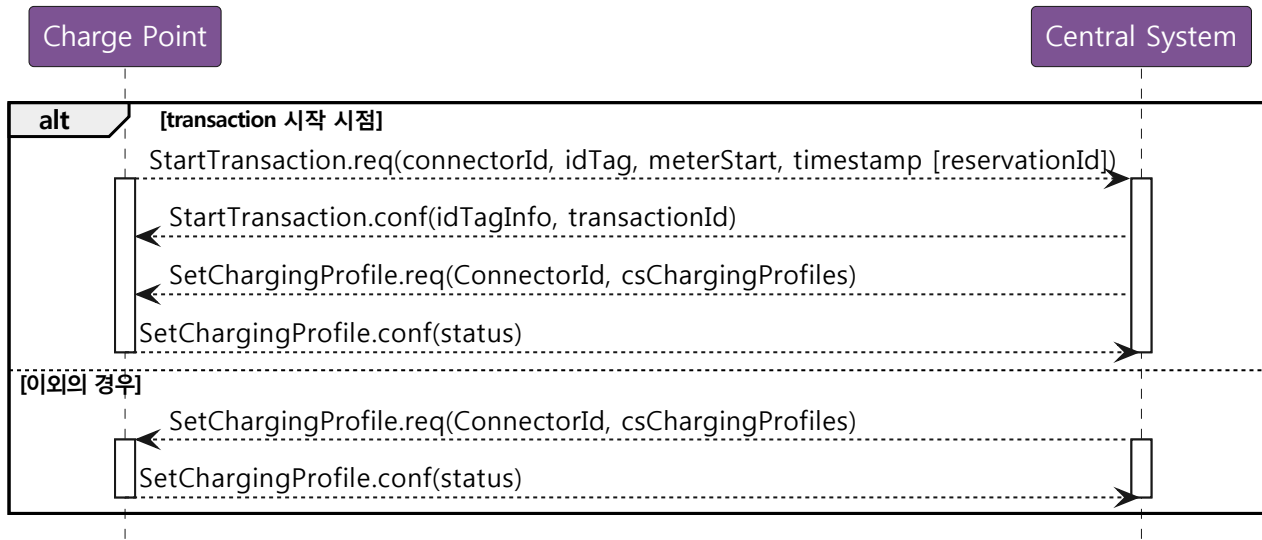


Figure 36. 시퀀스 다이어그램: 충전 프로파일 설정

CSMS는 다음과 같은 상황에서 충전 프로파일을 설정하기 위해 충전기에 `SetChargingProfile.req`를 보낼 수 있습니다.

- 트랜잭션 시작 시 트랜잭션에 대한 충전 프로파일을 설정합니다.
- 충전기에 `RemoteStartTransaction.req`를 보냅니다.
- 트랜잭션 중에 트랜잭션에 대한 활성 프로파일을 변경합니다.
- 트랜잭션 컨텍스트 외부에서 별도의 메시지로 로컬 컨트롤러, 충전기 또는 커넥터에 대한 기본 충전 프로파일을 설정합니다.



트랜잭션과 TxProfile 간의 불일치를 방지하기 위해 CSMS는 프로파일이 특정 트랜잭션에 적용되는 경우 `SetChargingProfile.req`에 `transactionId`를 포함해야 합니다.

이러한 상황은 아래에 설명되어 있습니다.

5.16.1. Setting a charging profile at start of transaction

CSMS가 `StartTransaction.req`를 수신하면 CSMS는 `StartTransaction.conf`로 응답해야 합니다. 충전 프로파일이 필요한 경우 CSMS는 `SetChargingProfile.req`를 충전기로 보내는 것을 선택할 수 있습니다.

충전 프로파일을 보내기 전에 `StartTransaction.req` PDU의 타임스탬프를 확인하여 트랜잭션이 계속 진행 중일 가능성이 있는지 확인하는 것이 좋습니다. `StartTransaction.req`는 오프라인 기간 동안 캐시되었을 수 있습니다.

5.16.2. Setting a charge profile in a RemoteStartTransaction request

CSMS는 `RemoteStartTransaction` 요청에 충전 프로파일을 포함할 수 있습니다.

CSMS가 `ChargingProfile`를 포함한 경우 `ChargingProfilePurpose`는 TxProfile로 설정해야 하며 `transactionId`는 설정해서는 안 됩니다.



충전기는 새로 시작된 트랜잭션에 지정된 프로파일을 적용해야 합니다. 이 트랜잭션은 CSMS에서 `StartTransaction.conf`를 통해 할당된 `transactionId`를 받게 됩니다. 충전기가 `SetChargingProfile.req`를 수신하면, 이 트랜잭션에 대한 `_transactionId`와 `RemoteStartTransaction.req`에 제공된 프로파일과 동일한 `StackLevel`을 갖는 충전기는 기존 충전 프로파일을 대체해야 하며, 그렇지 않은 경우 기존 프로파일 옆에 프로파일을 설치/스택해야 합니다.

5.16.3. Setting a charging profile during a transaction

CSMS는 충전기에 충전 프로파일을 보내 해당 트랜잭션에 대한 충전 프로파일을 업데이트할 수 있습니다. CSMS는 이 목적으로 `SetChargingProfile.req` PDU를 사용해야 합니다. 동일한 `chargingProfileId` 또는 동일한 `stackLevel` / `ChargingProfilePurpose` 조합을 가진

충전 프로파일이 충전기에 있는 경우, 새로운 충전 프로파일은 기존 충전 프로파일을 대체해야 하며, 그렇지 않은 경우 추가해야 합니다. 그런 다음 충전기는 충전 프로파일 컬렉션을 다시 평가하여 어떤 충전 프로파일을 활성화할지 결정해야 합니다. 업데이트된 충전 프로파일이 현재 트랜잭션에만 적용되도록 하려면 `ChargingProfile`의 `chargingProfilePurpose`를 `TxProfile`로 설정해야 합니다. ([Charging Profile Purposes](#) 섹션 참조)

5.16.4. Setting a charging profile outside of a transaction

CSMS는 기본 충전 프로파일로 사용할 충전 프로파일을 충전기로 보낼 수 있습니다. CSMS는 해당 목적으로 `SetChargingProfile.req` PDU를 사용해야 합니다. 이러한 충전 프로파일은 언제든지 보낼 수 있습니다. 동일한 `chargingProfileId` 또는 동일한 `stackLevel` / `ChargingProfilePurpose` 조합을 가진 충전 프로파일이 충전기에 있는 경우 새 충전 프로파일이 기존 충전 프로파일을 대체해야 하며, 그렇지 않은 경우 추가해야 합니다. 그런 다음 충전기는 충전 프로파일 모음을 다시 평가하여 어떤 충전 프로파일이 활성화될지 결정해야 합니다.



`ChargingProfile`를 활성 트랜잭션이 없거나, 트랜잭션이 진행되기 전에 설정하는 경우, `TxProfile`를 설정하기 위한 목적으로 사용할 수 없습니다.



`ChargingProfile`이 실행 중에 업데이트 되는 경우, 새 `ChargingProfile`의 시작 일정을 과거에 두는 것이 좋습니다. 그러면 `ChargingProfiles` 사이에 기본 충전 동작 기간이 없습니다. 충전기는 새 `ChargingProfile`이 설치될 때까지 기존 `ChargingProfile`을 계속 실행해야 합니다.



`chargingSchedulePeriod`가 기간보다 길면 나머지 기간은 실행되지 않아야 합니다. 기간이 `chargingSchedulePeriod`보다 길면 충전기는 `duration`이 끝날 때까지 마지막 `chargingSchedulePeriod`의 값을 유지해야 합니다.



`recurrencyKind`가 `chargingSchedulePeriod` 및/또는 반복 기간보다 긴 기간과 함께 사용되는 경우 나머지 기간은 실행되지 않아야 합니다.



`chargingSchedule`의 첫 번째 `chargingSchedulePeriod`의 `StartSchedule`은 항상 0이어야 합니다.



`recurrencyKind`가 `recurrencyKind` 기간보다 짧은 `chargingSchedule duration`과 함께 사용되는 경우, 충전기는 `chargingSchedule duration`이 끝난 후 기본 동작으로 돌아갑니다. 이 폴백은 충전기가 사용 가능한 경우 더 낮은 `stackLevel`을 가진 `ChargingProfile`을 사용해야 함을 의미합니다. 다른 `ChargingProfile`을 사용할 수 없는 경우 충전기는 `ChargingProfile`이 설치되지 않은 것처럼 충전을 허용해야 합니다. `chargingSchedulePeriod` 및/또는 기간이 반복 기간 기간보다 긴 경우 나머지 기간은 실행되지 않아야 합니다.

5.17. Trigger Message

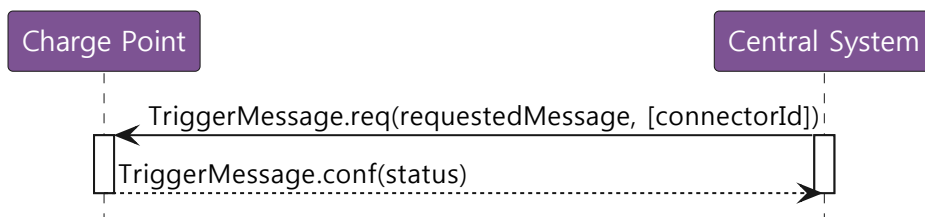


Figure 37. 시퀀스 다이어그램: 트리거 메시지

정상 작동 중에 충전기는 CSMS에 상태와 관련 발생 사항을 알립니다. 보고할 내용이 없으면 충전기는 미리 정의된 간격으로 최소한 하트비트를 보냅니다. 일반적인 상황에서는 괜찮지만, CSMS가 마지막으로 알려진 상태를 의심할 만한 (어떤) 이유가 있다면 어떨까요? 펌웨어 업데이트가 진행 중이고 이에 대해 수신한 마지막 상태 알림이 합리적으로 예상할 수 있는 것보다 훨씬 오래 전인 경우 CSMS는 어떻게 할 수 있을까요? 진단 요청의 진행 상황에 대해서도 동일한 질문을 할 수 있습니다. 이러한 상황에서 문제는 필요한 정보가 기존 메시지에 포함되지 않는다는 것이 아니라, 문제는 순전히 타이밍 문제입니다. 충전기에는 정보가 있지만 CSMS가 업데이트를 원한다는 것을 알 수 있는 방법이 없습니다.

`TriggerMessage.req`를 사용하면 CSMS가 충전기에 요청하여 충전기에서 시작한 메시지를 보낼 수 있습니다. 요청에서 CSMS는 수신하고자

하는 메시지를 표시합니다. CSMS는 이러한 요청된 모든 메시지에 대해 선택적으로 이 요청이 적용되는 커넥터를 표시할 수 있습니다. 요청된 메시지는 다음과 같습니다. 지정된 connectorId가 메시지와 관련이 없으면 무시해야 합니다. 이러한 경우에도 요청된 메시지를 보내야 합니다.

반대로, connectorId가 관련이 있지만 없는 경우, 이는 "허용된 모든 connectorId 값에 대해"로 해석되어야 합니다. 예를 들어, connectorId 0에 대한 statusNotification 요청은 충전기의 상태에 대한 요청입니다. connectorId가 없는 statusNotification에 대한 요청은 여러 statusNotification에 대한 요청입니다. 충전기 자체에 대한 알림과 각 커넥터에 대한 알림입니다.

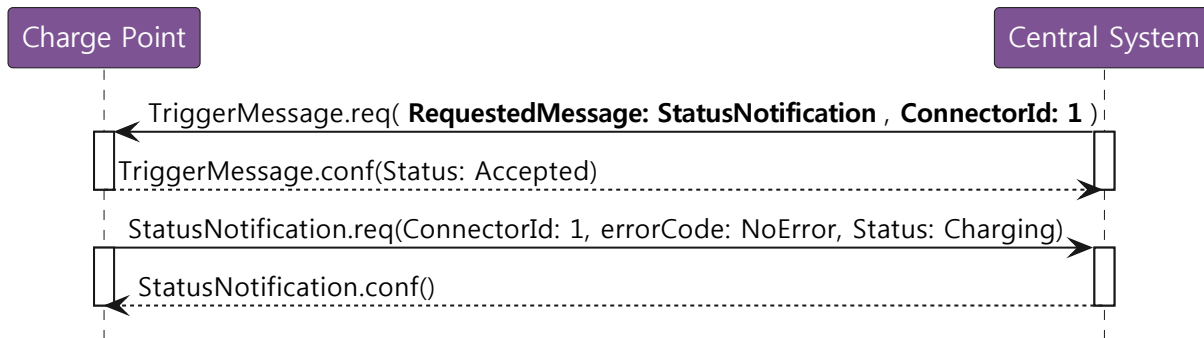


Figure 38. 시퀀스 다이어그램: 트리거 메시지 - StatusNotification 예

충전기는 요청된 메시지를 보내기 전에 먼저 TriggerMessage 응답을 보내야 합니다. TriggerMessage.conf에서 충전기는 ACCEPTED 또는 REJECTED를 반환하여 보낼지 여부를 표시해야 합니다. 충전기가 보내기 요청을 수락하거나 거부하는 것은 충전기에 달려 있습니다. 요청된 메시지가 알려지지 않았거나 구현되지 않은 경우 충전기는 NOT_IMPLEMENTED를 반환해야 합니다.

충전기가 ACCEPTED로 표시한 메시지는 전송해야 합니다. 요청을 수락하고 실제로 요청된 메시지를 전송하는 사이에 정상적인 작업으로 인해 동일한 메시지가 전송되는 상황이 발생할 수 있습니다. 이러한 경우 방금 전송된 메시지는 요청을 준수하는 것으로 간주될 수 있습니다.

TriggerMessage 메커니즘은 과거 데이터를 검색하기 위한 것이 아닙니다. 트리거하는 메시지는 현재 정보만 제공해야 합니다. 예를 들어 이런 방식으로 트리거된 MeterValues.req 메시지는 구성 키 MeterValuesSampledData에 구성된 모든 측정량에 대한 최신 측정값을 반환해야 합니다. StartTransaction 및 StopTransaction은 상태와 관련이 없지만 본질적으로 전환을 설명하기 때문에 이 메커니즘에서 제외되었습니다.

5.18. Unlock Connector



Figure 39. 시퀀스 다이어그램: 커넥터 잠금 해제

CSMS는 충전기에 커넥터 잠금 해제를 요청할 수 있습니다. 이를 위해 CSMS는 UnlockConnector.req PDU를 보내야 합니다. 이 메시지의 목적은 커넥터 케이블 고정 장치의 오작동 시 충전기에서 케이블을 분리하는 데 문제가 있는 EV 운전자를 돕는 것입니다.

EV 운전자가 CPO 헬프 데스크에 전화하면 운영자가 수동으로 UnlockConnector.req를 충전기로 보내 커넥터 잠금 해제를 다시 시도할 수 있습니다. 이번에는 커넥터가 잠금 해제되고 EV 운전자가 케이블을 분리하고 운전할 수 있기를 바랍니다.

UnlockConnector.req는 실행 중인 트랜잭션을 원격으로 중지하는 데 사용해서는 안 되며, 대신 Remote Stop Transaction을 사용해야 합니다.

UnlockConnector.req PDU를 수신하면 충전기는 UnlockConnector.conf PDU로 응답해야 합니다. 응답 PDU는 충전기가 커넥터를 잠금 해제할 수 있었는지 여부를 나타내야 합니다.

특정 커넥터에서 진행 중인 트랜잭션이 있는 경우 충전기는 Stop Transaction에 설명된 대로 먼저 트랜잭션을 완료해야 합니다.



UnlockConnector.req는 충전기 커버를 잠금 해제하는 것이 아니라 커넥터의 케이블 고정 잠금을 해제하기 위한 것입니다.

5.19. Update Firmware

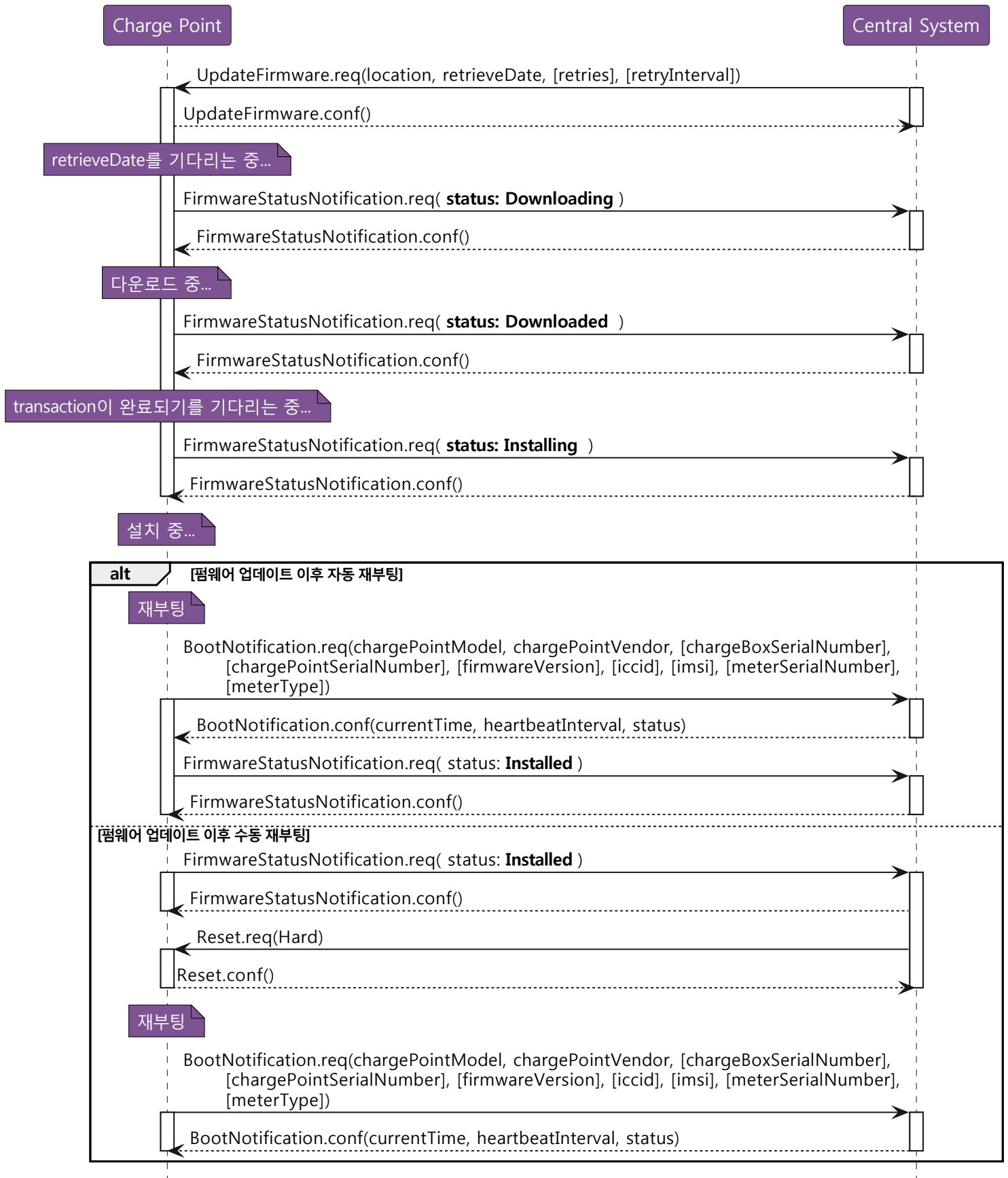


Figure 40. 시퀀스 다이어그램: 펌웨어 업데이트

CSMS는 충전기에 펌웨어를 업데이트해야 한다는 것을 알릴 수 있습니다. CSMS는 충전기에 새 펌웨어를 설치하도록 지시하기 위해 `UpdateFirmware.req` PDU를 보내야 합니다. PDU에는 충전기가 새 펌웨어를 검색할 수 있는 날짜와 시간, 펌웨어를 다운로드할 수 있는 위치가 포함되어야 합니다.

`UpdateFirmware.req` PDU를 수신하면 충전기는 `UpdateFirmware.conf` PDU로 응답해야 합니다. 충전기는 검색 날짜 후 가능한 한 빨리 펌웨어 검색을 시작해야 합니다.

펌웨어를 다운로드하고 설치하는 동안 충전기는 `FirmwareStatusNotification.req` PDU를 보내 CSMS에 업데이트 프로세스 상태를 알려야 합니다.

새 펌웨어 이미지가 "vaild"한 경우 충전기는 가능한 한 빨리 새 펌웨어를 설치해야 합니다.

펌웨어 설치 중에 충전을 계속할 수 없는 경우 설치를 시작하기 전에 충전 세션이 종료될 때까지(충전기 유효) 기다리는 것이 좋습니다. 충전기가 세션이 종료될 때까지 기다리는 동안 사용하지 않는 커넥터를 UNAVAILABLE로 설정하는 것이 좋습니다.



위의 시퀀스 다이어그램은 예시입니다. 상태를 *Installed* 로 보내기 전에 먼저 충전기를 재부팅하여 새 펌웨어가 부팅되고 CSMS에 연결할 수 있는지 확인하는 것이 좋습니다. 이는 필수 사항이 아닙니다.

Chapter 6. Messages

6.1. Authorize.req

여기에는 충전소에서 CSMS로 보낸 Authorize.req PDU의 필드 정의가 들어 있습니다. [Authorize](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
idTag	IdToken	1..1	필수. 여기에는 인증이 필요한 식별자가 들어 있습니다.

6.2. Authorize.conf

여기에는 CSMS가 충전기에서 [Authorize.req](#) PDU에 대한 응답으로 보낸 Authorize.conf PDU의 필드 정의가 들어 있습니다. [Authorize](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
idTagInfo	IdTagInfo	1..1	필수. 여기에는 인증 상태, 만료 및 parent id에 대한 정보가 포함됩니다.

6.3. BootNotification.req

여기에는 충전기에서 CSMS로 보낸 BootNotification.req PDU의 필드 정의가 포함됩니다. [Boot Notification](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
chargeBoxSerialNumber	CiString25Type	0..1	선택 사항. 여기에는 충전기 내부의 충전 단자의 일련 번호를 식별하는 값이 포함됩니다. 이 필드는 더 이상 사용되지 않으며 향후 버전에서 제거됩니다.
chargePointModel	CiString20Type	1..1	필수. 여기에는 충전기의 모델을 식별하는 값이 포함됩니다.
chargePointSerialNumber	CiString25Type	0..1	선택 사항. 여기에는 충전기의 일련 번호를 식별하는 값이 포함됩니다.
chargePointVendor	CiString20Type	1..1	필수. 여기에는 충전기의 공급업체를 식별하는 값이 포함됩니다.
firmwareVersion	CiString50Type	0..1	선택 사항. 여기에는 충전기의 펌웨어 버전이 포함됩니다.
iccid	CiString20Type	0..1	선택 사항. 여기에는 모뎀 SIM 카드의 ICCID가 포함됩니다.
imsi	CiString20Type	0..1	선택 사항. 여기에는 모뎀 SIM 카드의 IMSI가 포함됩니다.
meterSerialNumber	CiString25Type	0..1	선택 사항. 여기에는 충전기의 주 계량기의 일련 번호가 포함됩니다.
meterType	CiString25Type	0..1	선택 사항. 여기에는 충전기의 주 계량기의 유형이 포함됩니다.

6.4. BootNotification.conf

여기에는 [BootNotification.req](#) PDU에 대한 응답으로 CSMS에서 충전기로 보낸 BootNotification.conf PDU의 필드 정의가 들어 있습니다. [Boot Notification](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
currentTime	dateTime	1..1	필수. 여기에는 CSMS의 현재 시간이 포함됩니다.
interval	integer	1..1	필수. <i>RegistrationStatus</i> 가 <i>Accepted</i> 인 경우, 여기에는 초 단위의 하트비트 간격이 포함됩니다. CSMS가 <i>Accepted</i> 가 아닌 다른 값을 반환하는 경우, <i>interval</i> 필드의 값은 다음 <i>BootNotification</i> 요청을 보내기 전 최소 대기 시간을 나타냅니다.
status	<i>RegistrationStatus</i>	1..1	필수. 여기에는 충전기가 CSMS에 등록되었는지 여부가 포함됩니다.

6.5. CancelReservation.req

CSMS에서 충전기로 보낸 CancelReservation.req PDU의 필드 정의가 포함되어 있습니다. *Cancel Reservation*도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
reservationId	정수	1..1	필수. 취소할 예약의 ID입니다.

6.6. CancelReservation.conf

여기에는 *CancelReservation.req* PDU에 대한 응답으로 충전기에서 CSMS로 보낸 CancelReservation.conf PDU의 필드 정의가 포함되어 있습니다. *Cancel Reservation*도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	<i>CancelReservationStatus</i>	1..1	필수. CSMS에서 보낸 예약 취소 메시지에 대한 성공 또는 실패를 나타냅니다.

6.7. ChangeAvailability.req

여기에는 CSMS가 충전기로 보낸 ChangeAvailability.req PDU의 필드 정의가 포함되어 있습니다. *Change Availability*도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
connectorId	integer connectorId >= 0	1..1	필수. 가용성을 변경해야 하는 커넥터의 ID입니다. 충전기와 모든 커넥터의 가용성을 변경해야 하는 경우 ID '0'(영)이 사용됩니다.
type	<i>AvailabilityType</i>	1..1	필수. 여기에는 충전기가 수행해야 하는 가용성 변경 유형이 포함됩니다.

6.8. ChangeAvailability.conf

여기에는 충전기에서 CSMS로 반환한 ChangeAvailability.conf PDU의 필드 정의가 포함되어 있습니다. 또한 *Change Availability*를 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	<i>AvailabilityStatus</i>	1..1	필수. 충전기에서 가용성 변경을 수행할 수 있는지 여부를 나타냅니다.

6.9. ChangeConfiguration.req

여기에는 CSMS에서 충전기로 보낸 ChangeConfiguration.req PDU의 필드 정의가 포함되어 있습니다. '키' 및 '값' 필드의 내용과 의미는 충전기와 CSMS 간에 합의하는 것이 좋습니다. [Change Configuration](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
key	CiString50Type	1..1	필수. 변경할 구성 설정의 이름입니다. 표준 구성 키 이름과 연관된 값은 configuration key를 참조하세요.
value	CiString500Type	1..1	필수. 설정에 대한 문자열인 새 값입니다. 표준 구성 키 이름 및 연관된 값은 configuration key를 참조하세요.

6.10. ChangeConfiguration.conf

여기에는 충전기에서 CSMS로 반환된 ChangeConfiguration.conf PDU의 필드 정의가 포함되어 있습니다. [Change Configuration](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	ConfigurationStatus	1..1	필수. 구성 변경이 수락되었는지 여부를 반환합니다.

6.11. ClearCache.req

여기에는 CSMS에서 충전기로 보낸 ClearCache.req PDU의 필드 정의가 포함되어 있습니다. [Clear Cache](#)도 참조하세요.

정의된 필드가 없습니다.

6.12. ClearCache.conf

여기에는 [ClearCache.req](#) PDU에 대한 응답으로 충전기에서 CSMS로 보낸 ClearCache.conf PDU의 필드 정의가 들어 있습니다. [Clear Cache](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	ClearCacheStatus	1..1	필수. 충전기에서 요청을 실행한 경우 수락되고, 그렇지 않으면 거부됩니다.

6.13. ClearChargingProfile.req

CSMS에서 충전기로 보낸 ClearChargingProfile.req PDU의 필드 정의가 포함되어 있습니다.

CSMS는 이 메시지를 사용하여 특정 충전 프로파일(id로 표시) 또는 선택적 connectorId, stackLevel 및 [chargingProfilePurpose](#) 필드의 값과 일치하는 충전 프로파일을 지울(제거) 수 있습니다. [Clear Charging Profile](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
id	integer	0..1	선택 사항. 지울 충전 프로파일의 ID입니다.

필드 이름	필드 유형	카디널리티	설명
connectorId	integer	0..1	선택 사항. 충전 프로파일을 지울 커넥터의 ID를 지정합니다. 0(0)의 connectorId는 전체 충전기에 대한 충전 프로파일을 지정합니다. 이 매개변수가 없으면 요청의 다른 기준과 일치하는 모든 충전 프로파일에 지우기가 적용됩니다.
chargingProfilePurpose	ChargingProfilePurposeType	0..1	선택 사항. 요청의 다른 기준을 충족하는 경우 지워질 충전 프로파일의 목적을 지정합니다.
stackLevel	integer	0..1	선택 사항. 요청의 다른 기준을 충족하는 경우 충전 프로파일이 지워지는 stackLevel을 지정합니다.

6.14. ClearChargingProfile.conf

여기에는 충전기가 [ClearChargingProfile.req](#) PDU에 대한 응답으로 CSMS에 보낸 ClearChargingProfile.conf PDU의 필드 정의가 포함됩니다. [Clear Charging Profile](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	ClearChargingProfileStatus	1..1	필수. 충전기가 요청을 실행할 수 있었는지 여부를 나타냅니다.

6.15. DataTransfer.req

여기에는 CSMS에서 충전기로 또는 그 반대로 전송한 DataTransfer.req PDU의 필드 정의가 포함됩니다. [Data Transfer](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
vendorId	CiString255Type	1..1	필수. 공급업체별 구현을 식별합니다.
messageId	CiString50Type	0..1	선택 사항. 추가 식별 필드
data	텍스트 길이 정의되지 않음	0..1	선택 사항. 길이나 형식이 지정되지 않은 데이터.

6.16. DataTransfer.conf

여기에는 충전기에서 CSMS로 전송한 DataTransfer.conf PDU의 필드 정의가 포함되어 있으며, [DataTransfer.req](#) PDU에 대한 응답으로 CSMS 또는 그 반대로 전송됩니다. [Data Transfer](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	DataTransferStatus	1..1	필수. 데이터 전송의 성공 또는 실패를 나타냅니다.
data	텍스트 길이 정의되지 않음	0..1	선택 사항. 요청에 대한 응답 데이터

6.17. DiagnosticsStatusNotification.req

여기에는 충전기에서 CSMS로 보낸 DiagnosticsStatusNotification.req PDU의 필드 정의가 포함되어 있습니다. [Diagnostics Status Notification](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	DiagnosticsStatus	1..1	필수. 여기에는 진단 업로드 상태가 포함되어 있습니다.

6.18. DiagnosticsStatusNotification.conf

여기에는 CSMS가 [DiagnosticsStatusNotification.req](#) PDU에 대한 응답으로 충전기로 보낸 DiagnosticsStatusNotification.conf PDU의 필드 정의가 포함되어 있습니다. [Diagnostics Status Notification](#)도 참조하세요.

정의된 필드가 없습니다.

6.19. FirmwareStatusNotification.req

여기에는 충전기가 CSMS로 보낸 FirmwareStatusNotifitacion.req PDU의 필드 정의가 포함되어 있습니다. 또한 [Firmware Status Notification](#)을 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	FirmwareStatus	1..1	필수. 여기에는 펌웨어 설치의 진행 상태가 포함됩니다.

6.20. FirmwareStatusNotification.conf

여기에는 CSMS가 [FirmwareStatusNotification.req](#) PDU에 대한 응답으로 충전기로 보낸 FirmwareStatusNotification.conf PDU의 필드 정의가 포함됩니다. [Firmware Status Notification](#)도 참조하세요.

정의된 필드가 없습니다.

6.21. GetCompositeSchedule.req

여기에는 CSMS가 충전기로 보낸 GetCompositeSchedule.req PDU의 필드 정의가 들어 있습니다. 또한 [Get Composite Schedule](#)을 참조하세요.

필드 이름	필드 유형	카디널리티	설명
connectorId	integer	1..1	필수. 일정이 요청된 커넥터의 ID입니다. ConnectorId=0인 경우 충전소는 그리드 연결에 대한 예상 소비량을 계산합니다.
duration	integer	1..1	필수. 초 단위 시간. 요청된 일정의 길이
chargingRateUnit	ChargingRateUnitType	0..1	선택 사항. 전력 또는 전류 프로파일을 강제로 적용하는 데 사용할 수 있습니다.

6.22. GetCompositeSchedule.conf

여기에는 충전소가 GetCompositeSchedule.req PDU에 대한 응답으로 CSMS에 보낸 GetCompositeSchedule.conf PDU의 필드 정의가 포함됩니다. [Get Composite Schedule](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	GetCompositeScheduleStatus	1..1	필수. 요청의 상태. 충전기는 요청을 처리할 수 있었는지 여부를 나타냅니다.
connectorId	integer	0..1	선택 사항. 이 알림에 포함된 충전 일정은 커넥터에 적용됩니다.
scheduleStart	dateTime	0..1	선택 사항. 시간. 충전 프로파일에 포함된 기간은 이 시점을 기준으로 합니다. 상태가 "Rejected"인 경우 이 필드가 없을 수 있습니다.
chargingSchedule	ChargingSchedule	0..1	선택 사항. 계획된 복합 충전 일정, 시간 경과에 따른 에너지 소비량. 항상 ScheduleStart를 기준으로 합니다. 상태가 "Rejected"인 경우 이 필드가 없을 수 있습니다.

6.23. GetConfiguration.req

여기에는 CSMS에서 충전기로 보낸 GetConfiguration.req PDU의 필드 정의가 포함되어 있습니다. [Get Configuration](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
key	CiString50Type	0..*	선택 사항. 구성 값이 요청되는 키 목록.

6.24. GetConfiguration.conf

여기에는 충전기에서 CSMS으로 보낸 GetConfiguration.conf PDU의 필드 정의가 포함되어 있으며, [GetConfiguration.req](#)에 대한 응답으로 전송됩니다. [Get Configuration](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
configurationKey	KeyValue	0..*	선택 사항. 요청되거나 알려진 키 목록
unknownKey	CiString50Type	0..*	선택 사항. 알려지지 않은 요청된 키

6.25. GetDiagnostics.req

여기에는 CSMS에서 충전기로 보낸 GetDiagnostics.req PDU의 필드 정의가 들어 있습니다. [Get Diagnostics](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
location	anyURI	1..1	필수. 여기에는 진단 파일을 업로드할 위치(디렉토리)가 들어 있습니다.
retries	integer	0..1	선택 사항. 충전기가 진단 업로드를 시도해야 하는 횟수를 지정합니다. 이 필드가 없으면 충전기에서 재시도할 횟수를 결정합니다.
retryInterval	integer	0..1	선택 사항. 재시도를 시도하는 간격(초). 이 필드가 없으면 충전기에서 재시도 대기 시간을 결정합니다.
startTime	dateTime	0..1	선택 사항. 진단에 포함할 가장 오래된 로깅 정보의 날짜와 시간이 포함됩니다.
stopTime	dateTime	0..1	선택 사항. 진단에 포함할 가장 최신 로깅 정보의 날짜와 시간이 포함됩니다.

6.26. GetDiagnostics.conf

여기에는 충전기에서 [GetDiagnostics.req](#) PDU에 대한 응답으로 CSMS로 보낸 [GetDiagnostics.conf](#) PDU의 필드 정의가 포함됩니다. [GetDiagnostics](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
fileName	CiString255Type	0..1	선택 사항. 여기에는 업로드될 진단 정보가 있는 파일의 이름이 포함됩니다. 진단 정보를 사용할 수 없는 경우 이 필드는 존재하지 않습니다.

6.27. GetLocalListVersion.req

여기에는 CSMS에서 충전기로 보낸 [GetLocalListVersion.req](#) PDU의 필드 정의가 포함됩니다. [Get Local List Version](#)도 참조하세요.

정의된 필드가 없습니다.

6.28. GetLocalListVersion.conf

여기에는 충전기에서 [GetLocalListVersion.req](#) PDU에 대한 응답으로 CSMS에 보낸 [GetLocalListVersion.conf](#) PDU의 필드 정의가 포함되어 있습니다. [Get Local List Version](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
listVersion	integer	1..1	필수. 여기에는 충전소의 로컬 인증 목록의 현재 버전 번호가 포함되어 있습니다.

6.29. Heartbeat.req

여기에는 충전기에서 CSMS으로 보낸 [Heartbeat.req](#) PDU의 필드 정의가 들어 있습니다. [Heartbeat](#)도 참조하세요.

정의된 필드가 없습니다.

6.30. Heartbeat.conf

여기에는 [Heartbeat.req](#) PDU에 대한 응답으로 CSMS에서 충전기로 보낸 [Heartbeat.conf](#) PDU의 필드 정의가 포함되어 있습니다. [Heartbeat](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
currentTime	dateTime	1..1	필수. 여기에는 CSMS의 현재 시간이 포함되어 있습니다.

6.31. MeterValues.req

여기에는 충전기에서 CSMS로 보낸 [MeterValues.req](#) PDU의 필드 정의가 포함되어 있습니다. [Meter Values](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
connectorId	integer + connectorId >= 0	1..1	필수. 여기에는 충전기의 커넥터를 지정하는 숫자(>0)가 포함됩니다. '0'(영)은 주 전력계를 지정하는 데 사용됩니다.
transactionId	integer	0..1	선택 사항. 미터 샘플과 관련된 트랜잭션입니다.

필드 이름	필드 유형	카디널리티	설명
meterValue	MeterValue	1..*	필수. 타임스탬프가 있는 샘플링된 미터 값입니다.

6.32. MeterValues.conf

여기에는 MeterValues.req PDU에 대한 응답으로 CSMS에서 충전기로 보낸 MeterValues.conf PDU의 필드 정의가 포함되어 있습니다. Meter Values도 참조하세요.

정의된 필드가 없습니다.

6.33. RemoteStartTransaction.req

여기에는 CSMS에서 충전기로 보낸 RemoteStartTransaction.req PDU의 필드 정의가 포함되어 있습니다. Remote Start Transaction도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
connectorId	integer	0..1	선택 사항. 트랜잭션을 시작할 커넥터 번호입니다. connectorId는 0보다 커야 합니다.
idTag	IdToken	1..1	필수. 충전기에서 트랜잭션을 시작하는 데 사용해야 하는 식별자입니다.
chargingProfile	ChargingProfile	0..1	선택 사항. 충전기에서 요청한 트랜잭션에 사용할 충전 프로파일입니다. ChargingProfilePurpose는 TxProfile로 설정해야 합니다.

6.34. RemoteStartTransaction.conf

여기에는 충전기에서 CSMS로 전송된 RemoteStartTransaction.conf PDU의 필드 정의가 포함됩니다. Remote Start Transaction도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	RemoteStartStopStatus	1..1	필수. 충전기에서 트랜잭션 시작 요청을 수락하는지 여부를 나타내는 상태입니다.

6.35. RemoteStopTransaction.req

여기에는 CSMS에서 충전기로 보낸 RemoteStopTransaction.req PDU의 필드 정의가 포함되어 있습니다. Remote Stop Transaction도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
transactionId	integer	1..1	필수. 충전기에서 중지하도록 요청한 트랜잭션의 id입니다.

6.36. RemoteStopTransaction.conf

여기에는 충전기에서 CSMS으로 보낸 RemoteStopTransaction.conf PDU의 필드 정의가 포함되어 있습니다. Remote Stop Transaction도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	RemoteStartStopStatus	1..1	필수. 충전소에서 트랜잭션 중지 요청을 수락하는지 여부를 나타내는 상태.

6.37. ReserveNow.req

여기에는 CSMS에서 충전기로 보낸 ReserveNow.req PDU의 필드 정의가 포함됩니다. [Reserve Now](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
connectorId	integer connectorId >= 0	1..1	필수. 여기에는 예약할 커넥터의 ID가 포함됩니다. 값 0은 예약이 특정 커넥터를 위한 것이 아님을 의미합니다.
expiryDate	dateTime	1..1	필수. 여기에는 예약이 종료되는 날짜와 시간이 포함됩니다.
idTag	IdToken	1..1	필수. 충전소가 커넥터를 예약해야 하는 식별자입니다.
parentIdTag	IdToken	0..1	선택 사항. 부모 idTag입니다.
reservationId	integer	1..1	필수. 이 예약에 대한 고유 ID입니다.

6.38. ReserveNow.conf

여기에는 충전기에서 [ReserveNow.req](#) PDU에 대한 응답으로 CSMS에 보낸 ReserveNow.conf PDU의 필드 정의가 포함되어 있습니다. [Reserve Now](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	ReservationStatus	1..1	필수. 예약의 성공 또는 실패를 나타냅니다.

6.39. Reset.req

여기에는 CSMS가 충전기로 보낸 Reset.req PDU의 필드 정의가 들어 있습니다. [Reset](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
type	ResetType	1..1	필수. 여기에는 충전기가 수행해야 하는 재설정 유형이 들어 있습니다.

6.40. Reset.conf

여기에는 충전기가 [Reset.req](#) PDU에 대한 응답으로 CSMS에 보낸 Reset.conf PDU의 필드 정의가 들어 있습니다. [Reset](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	ResetStatus	1..1	필수. 충전기에서 재설정을 수행할 수 있는지 여부를 나타냅니다.

6.41. SendLocalList.req

여기에는 CSMS에서 충전기로 보낸 SendLocalList.req PDU의 필드 정의가 포함됩니다.

(비어 있는) localAuthorizationList가 제공되지 않고 updateType이 Full인 경우, 모든 식별 정보가 목록에서 제거됩니다. (비어 있는) localAuthorizationList 없이 차등 업데이트를 요청해도 목록에 영향을 미치지 않습니다. localAuthorizationList의 모든 idTags는 고유해야 하며 중복 값은 허용되지 않습니다. [Send Local List](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
listVersion	integer	1..1	필수. 전체 업데이트의 경우 전체 목록의 버전 번호입니다. 차등 업데이트의 경우 업데이트가 적용된 후의 목록 버전 번호입니다.
localAuthorizationList	AuthorizationData	0..*	선택 사항. 전체 업데이트의 경우 새 로컬 인증 목록을 구성하는 값 목록이 포함됩니다. 차등 업데이트의 경우 충전기의 로컬 인증 목록에 적용할 변경 사항이 포함됩니다. AuthorizationData 요소의 최대 수는 구성 키 SendLocalListMaxLength 에서 사용할 수 있습니다.
updateType	UpdateType	1..1	필수. 여기에는 이 요청의 업데이트 유형(전체 또는 차등)이 포함됩니다.

6.42. SendLocalList.conf

여기에는 [SendLocalList.req](#) PDU에 대한 응답으로 충전기에서 CSMS로 보낸 SendLocalList.conf PDU의 필드 정의가 포함됩니다. [Send Local List](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	UpdateStatus	1..1	필수. 충전기가 로컬 인증 목록 업데이트를 성공적으로 수신하고 적용했는지 여부를 나타냅니다.

6.43. SetChargingProfile.req

여기에는 CSMS에서 충전기로 보낸 SetChargingProfile.req PDU의 필드 정의가 포함됩니다.

CSMS는 이 메시지를 사용하여 충전기로 충전 프로파일을 보냅니다. [Set Charging Profile](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
connectorId	integer	1..1	필수. 충전 프로파일이 적용되는 커넥터입니다. connectorId = 0인 경우 메시지에는 충전기에 대한 전체 제한이 포함됩니다.
csChargingProfiles	ChargingProfile	1..1	필수. 충전기에서 설정할 충전 프로파일입니다.

6.44. SetChargingProfile.conf

여기에는 충전기가 [SetChargingProfile.req](#) PDU에 대한 응답으로 CSMS에 보낸 SetChargingProfile.conf PDU의 필드 정의가 포함됩니다. [Set Charging Profile](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	ChargingProfileStatus	1..1	필수. 충전기에서 메시지를 성공적으로 처리할 수 있었는지 여부를 반환합니다. 이는 일정이 문자 그대로 준수될 것이라는 보장은 없습니다. 충전기에서 고려해야 할 다른 제약 조건이 있을 수 있습니다.

6.45. StartTransaction.req

이 섹션에는 충전기에서 CSMS로 보낸 StartTransaction.req PDU의 필드 정의가 포함되어 있습니다. [Start Transaction](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
connectorId	integer connectorId > 0	1..1	필수. 이는 충전기의 어떤 커넥터가 사용되는지 식별합니다.
idTag	IdToken	1..1	필수. 이는 트랜잭션을 시작해야 하는 ID를 포함합니다.
meterStart	integer	1..1	필수. 이는 트랜잭션 시작 시 커넥터의 미터 값(Wh)을 포함합니다.
reservationId	integer	0..1	선택 사항. 이는 이 트랜잭션의 결과로 종료되는 예약의 ID를 포함합니다.
timestamp	dateTime	1..1	필수. 이는 트랜잭션이 시작된 날짜와 시간을 포함합니다.

6.46. StartTransaction.conf

여기에는 CSMS가 StartTransaction.req PDU에 대한 응답으로 충전기로 보낸 StartTransaction.conf PDU의 필드 정의가 포함됩니다. [Start Transaction](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
idTagInfo	IdTagInfo	1..1	필수. 여기에는 승인 상태, 만료 및 부모 ID에 대한 정보가 포함됩니다.
transactionId	integer	1..1	필수. 여기에는 CSMS가 제공한 트랜잭션 ID가 포함됩니다.

6.47. StatusNotification.req

여기에는 충전기에서 CSMS로 보낸 StatusNotification.req PDU의 필드 정의가 포함되어 있습니다. [Status Notification](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
connectorId	integer connectorId >= 0	1..1	필수. 상태가 보고되는 커넥터의 ID입니다. 상태가 충전기 메인 컨트롤러에 대한 경우 ID '0'(영)이 사용됩니다.
errorCode	ChargePointErrorcode	1..1	필수. 충전기에서 보고한 오류 코드가 포함되어 있습니다.
info	CiString50Type	0..1	선택 사항. 오류와 관련된 추가 자유 형식 정보.
status	ChargePointStatus	1..1	필수. 여기에는 충전기의 현재 상태가 포함됩니다.

필드 이름	필드 유형	카디널리티	설명
timestamp	dateTime	0..1	선택 사항. 상태가 보고되는 시간입니다. 없는 경우 메시지 수신 시간으로 간주됩니다.
vendorId	CiString255Type	0..1	선택 사항. 공급업체별 구현을 식별합니다.
vendorErrorCode	CiString50Type	0..1	선택 사항. 여기에는 공급업체별 오류 코드가 포함됩니다.

6.48. StatusNotification.conf

여기에는 CSMS가 [StatusNotification.req](#) PDU에 대한 응답으로 충전기로 보낸 StatusNotification.conf PDU의 필드 정의가 포함되어 있습니다. [Status Notification](#)도 참조하세요.

정의된 필드가 없습니다.

6.49. StopTransaction.req

여기에는 충전기가 CSMS로 보낸 StopTransaction.req PDU의 필드 정의가 포함되어 있습니다. [Stop Transaction](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
idTag	IdToken	0..1	선택 사항. 여기에는 충전을 중지하도록 요청한 식별자가 포함됩니다. 충전기가 idTag 없이도 충전을 종료할 수 있으므로 선택 사항입니다(예: 재설정). 충전기는 알려진 경우 idTag를 보내야 합니다.
meterStop	integer	1..1	필수. 여기에는 트랜잭션이 끝날 때 커넥터에 대한 미터 값(Wh)이 포함됩니다.
timestamp	dateTime	1..1	필수. 여기에는 트랜잭션이 중지된 날짜와 시간이 포함됩니다.
transactionId	integer	1..1	필수. 여기에는 StartTransaction.conf 에서 수신한 트랜잭션 ID가 포함됩니다.
reason	Reason	0..1	선택 사항. 여기에는 트랜잭션이 중지된 이유가 포함됩니다. Reason 이 "Local"일 때만 생략할 수 있습니다.
transactionData	MeterValue	0..*	선택 사항. 여기에는 청구 목적에 관련된 트랜잭션 사용 세부 정보가 포함됩니다.

6.50. StopTransaction.conf

여기에는 [StopTransaction.req](#) PDU에 대한 응답으로 CSMS에서 충전기로 보낸 StopTransaction.conf PDU의 필드 정의가 포함되어 있습니다. [Stop Transaction](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
idTagInfo	IdTagInfo	0..1	선택 사항. 여기에는 승인 상태, 만료 및 부모 ID에 대한 정보가 포함되어 있습니다. 식별자 없이 트랜잭션이 중단되었을 수 있으므로 선택 사항입니다.

6.51. TriggerMessage.req

여기에는 CSMS가 충전기로 보낸 TriggerMessage.req PDU의 필드 정의가 들어 있습니다. [Trigger Message](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
requestedMessage	MessageTrigger	1..1	필수.
connectorId	integer connectorId > 0	0..1	선택 사항. 요청이 특정 커넥터에 적용되는 경우에만 채워집니다.

6.52. TriggerMessage.conf

여기에는 충전기에서 [TriggerMessage.req](#) PDU에 대한 응답으로 CSMS에 보낸 [TriggerMessage.conf](#) PDU의 필드 정의가 포함되어 있습니다. [Trigger Message](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	TriggerMessageStatus	1..1	필수. 충전기에서 요청된 알림을 보낼지 여부를 나타냅니다.

6.53. UnlockConnector.req

CSMS에서 충전기로 보낸 [UnlockConnector.req](#) PDU의 필드 정의가 포함되어 있습니다. [Unlock Connector](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
connectorId	integer connectorId > 0	1..1	필수. 잠금 해제할 커넥터의 식별자가 포함되어 있습니다.

6.54. UnlockConnector.conf

여기에는 [UnlockConnector.req](#) PDU에 대한 응답으로 충전기에서 CSMS로 보낸 [UnlockConnector.conf](#) PDU의 필드 정의가 포함되어 있습니다. [Unlock Connector](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
status	UnlockStatus	1..1	필수. 충전기가 커넥터를 잠금 해제했는지 여부를 나타냅니다.

6.55. UpdateFirmware.req

CSMS에서 충전기로 보낸 [UpdateFirmware.req](#) PDU의 필드 정의가 포함되어 있습니다. [Update Firmware](#)도 참조하세요.

필드 이름	필드 유형	카디널리티	설명
location	anyURI	1..1	필수. 펌웨어를 검색할 위치를 가리키는 URI가 포함된 문자열이 포함되어 있습니다.
retries	integer	0..1	선택 사항. 이것은 충전기가 포기하기 전에 펌웨어 다운로드를 시도해야 하는 횟수를 지정합니다. 이 필드가 없으면 충전기가 재시도할 횟수를 결정합니다.
retrieveDate	dateTime	1..1	필수. 이것은 충전기가 (새로운) 펌웨어를 검색할 수 있는 날짜와 시간을 포함합니다.

필드 이름	필드 유형	카디널리티	설명
retryInterval	integer	0..1	선택 사항. 재시도를 시도할 수 있는 간격(초)입니다. 이 필드가 없으면 충전기가 시도 사이에 기다릴 시간을 결정합니다.

6.56. UpdateFirmware.conf

여기에는 충전기가 [UpdateFirmware.req](#) PDU에 대한 응답으로 CSMS에 보낸 [UpdateFirmware.conf](#) PDU의 필드 정의가 포함됩니다. [Update Firmware](#)도 참조하세요.

정의된 필드가 없습니다.

Chapter 7. Types

7.1. AuthorizationData

Class

로컬 인증 목록 업데이트의 항목을 구성하는 요소입니다.

필드 이름	필드 유형	카디널리티	설명
idTag	IdToken	1..1	필수. 이 권한이 적용되는 식별자입니다.
idTagInfo	IdTagInfo	0..1	선택 사항입니다. (UpdateType이 Full인 경우 필수) 여기에는 인증 상태, 만료 및 부모 ID에 대한 정보가 포함됩니다. 차등 업데이트의 경우 다음이 적용됩니다. 이 요소가 있는 경우 이 항목은 로컬 인증 목록에 추가되거나 업데이트되어야 합니다. 이 요소가 없는 경우 로컬 인증 목록에서 이 ID 태그에 대한 항목을 삭제해야 합니다.

7.2. AuthorizationStatus

Enumeration

Authorize.req에 대한 응답의 상태입니다.

값	설명
Accepted	ID가 충전에 허용됩니다.
Blocked	ID가 차단되었습니다. 충전에 허용되지 않습니다.
Expired	ID가 만료되었습니다. 충전에 허용되지 않습니다.
Invalid	ID를 알 수 없습니다. 충전에 허용되지 않습니다.
ConcurrentTx	ID가 이미 다른 트랜잭션에 관련되어 있으며 여러 트랜잭션이 허용되지 않습니다. (StartTransaction.req에만 해당)

7.3. AvailabilityStatus

Enumeration

ChangeAvailability.req에 대한 응답으로 반환된 상태입니다.

값	설명
Accepted	요청이 수락되었으며 실행됩니다.
Rejected	요청이 수락되지 않았으며 실행되지 않습니다.
Scheduled	요청이 수락되었으며 진행 중인 트랜잭션이 완료되면 실행됩니다.

7.4. AvailabilityType

Enumeration

ChangeAvailability.req에서 요청된 가용성 변경입니다.

값	설명
Inoperative	충전기를 충전에 이용할 수 없습니다.
Operative	충전기를 충전에 이용할 수 있습니다.

7.5. CancelReservationStatus

Enumeration

CancelReservation.conf의 상태입니다.

값	설명
Accepted	식별자에 대한 예약이 취소되었습니다.
Rejected	식별자에 대해 활성화된 예약이 없으므로 예약을 취소할 수 없습니다.

7.6. ChargePointErrorCode

Enumeration

충전기 상태가 StatusNotification.req에서 보고되었습니다.

값	설명
ConnectorLockFailure	커넥터 잠금 또는 잠금 해제 실패.
EVCommunicationError	차량과의 통신 실패, 모드 3 또는 다른 통신 프로토콜 문제일 수 있습니다. 이는 충전기가 오류 상태로 전환될 필요가 없다는 의미에서 실제 오류가 아닙니다. 대신 SuspendedEVSE 상태로 전환되어야 합니다.
GroundFailure	접지 고장 회로 차단기가 활성화되었습니다.
HighTemperature	충전기 내부 온도가 너무 높습니다.
InternalError	내부 하드 또는 소프트웨어 구성 요소 오류.
LocalListConflict	CSMS에서 수신한 인증 정보가 LocalAuthorizationList와 충돌합니다.
NoError	보고할 오류가 없습니다.
OtherError	다른 유형의 오류입니다. 자세한 내용은 vendorErrorCode에서 확인하세요.
OverCurrentFailure	과전류 보호 장치가 작동했습니다.
OverVoltage	전압이 허용 수준 이상으로 상승했습니다.
PowerMeterFailure	전기/에너지/전력 미터를 읽지 못했습니다.
PowerSwitchFailure	전원 스위치를 제어하지 못했습니다.

값	설명
ReaderFailure	idTag 리더에 오류가 발생했습니다.
ResetFailure	재설정을 수행할 수 없습니다.
UnderVoltage	전압이 허용 수준 아래로 떨어졌습니다.
WeakSignal	무선 통신 장치에서 약한 신호를 보고합니다.

7.7. ChargePointStatus

Enumeration

[StatusNotification.req](#)에서 보고된 상태입니다. 충전기 메인 컨트롤러(connectorId = 0) 또는 특정 커넥터에 대한 상태를 보고할 수 있습니다. 충전기 메인 컨트롤러의 상태는 다음 열거형의 하위 집합입니다. *Available*, *Unavailable* 또는 *Faulted*.

작동 상태로 간주되는 상태는 다음과 같습니다. *Available*, *Preparing*, *Charging*, *SuspendedEVSE*, *SuspendedEV*, *Finishing*, *Reserved*. 작동 불능 상태로 간주되는 상태는 다음과 같습니다. *Unavailable*, *Faulted*.

상태	조건
Available	Connector가 새 사용자에게 사용 가능해질 때(작동 중)
Preparing	Connector가 새 사용자에게 더 이상 사용 가능하지 않지만 진행 중인 트랜잭션이 없을 때(아직). 일반적으로 사용자가 태그를 제시하거나 케이블을 삽입하거나 차량이 주차 공간을 차지할 때 Connector는 준비 상태에 있습니다(작동 중)
Charging	Connector의 접촉부가 차량에 연결되어 차량이 충전될 수 있을 때(작동 중)
SuspendedEVSE	EV가 EVSE에 연결되었지만 EVSE가 EV에 에너지를 공급하지 않을 때(예: 스마트 충전 제한, 지역 공급 전력 제약 또는 StartTransaction.conf 에서 충전이 허용되지 않음) 등을 나타내는 결과로 인해 발생합니다. (작동 중)
SuspendedEV	EV가 EVSE에 연결되어 있고 EVSE가 에너지를 제공하지만 EV가 에너지를 사용하지 않는 경우입니다. (작동 중)
Finishing	Connector에서 트랜잭션이 중단되었지만 Connector를 아직 새 사용자가 사용할 수 없는 경우입니다. (예: 케이블이 제거되지 않았거나 차량이 주차장을 떠나지 않은 경우) (작동 중)
Reserved	Connector가 Reserve Now 명령의 결과로 예약된 경우입니다. (작동 중)
Unavailable	Connector가 Change Availability 명령 또는 충전기가 재량에 따라 사용할 수 없게 전환되는 이벤트의 결과로 사용할 수 없게 된 경우입니다. Change Availability 명령을 수신하면 상태가 즉시 변경될 수 있으며 변경이 예약될 수 있습니다. 예약된 경우, Status Notification 은 가용성 변경이 유효해질 때 전송됩니다. (작동 불가)
Faulted	충전기 또는 커넥터가 오류를 보고하고 에너지 공급을 받을 수 없는 경우입니다. (작동 불가)

7.8. ChargingProfile

Class

충전 프로파일은 [ChargingSchedule](#)으로 구성되며, 시간 간격당 공급할 수 있는 전력 또는 전류량을 설명합니다.

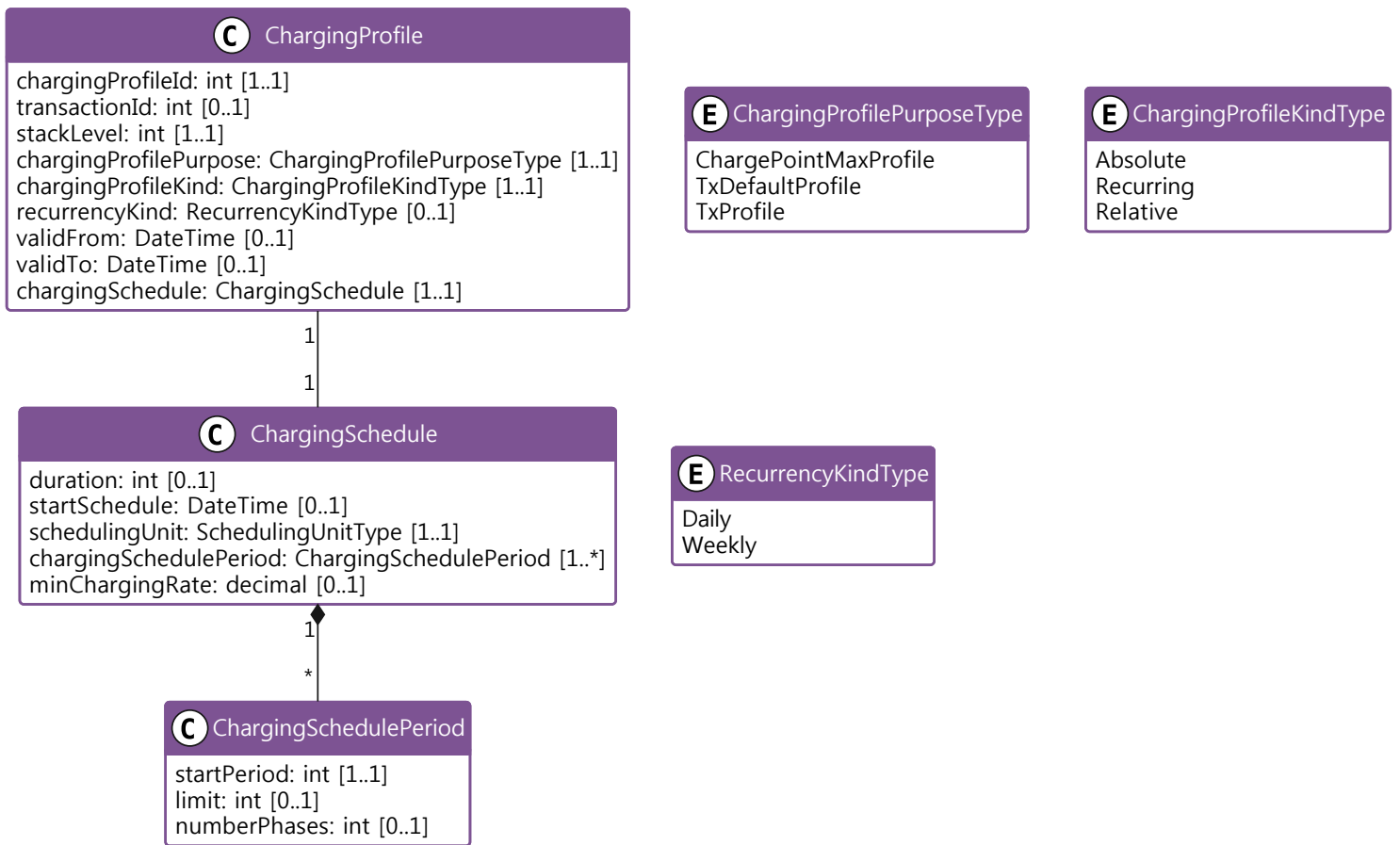


Figure 41. 클래스 다이어그램: 충전 프로파일

필드 이름	필드 유형	카디널리티	설명
chargingProfileId	integer	1..1	필수. 이 프로파일의 고유 식별자입니다.
transactionId	integer	0..1	선택 사항입니다. 충전 프로파일 용도가 TxProfile로 설정된 경우에만 유효하며, 트랜잭션 ID를 사용하여 프로파일을 특정 트랜잭션과 일치시킬 수 있습니다.
stackLevel	integer >=0	1..1	필수. 프로파일의 계층 스택에서 레벨을 결정하는 값입니다. 높은 값은 낮은 값보다 우선합니다. 가장 낮은 레벨은 0입니다.
chargingProfilePurpose	ChargingProfilePurposeType	1..1	필수. 이 메시지에서 전송된 일정의 목적을 정의합니다.
chargingProfileKind	ChargingProfileKindType	1..1	필수. 일정의 종류를 나타냅니다.
recurrencyKind	RecurrencyKindType	0..1	선택 사항. 반복의 시작점을 나타냅니다.
validFrom	dateTime	0..1	선택 사항. 프로파일이 유효해지기 시작하는 시점입니다. 없는 경우, 프로파일은 충전기에서 수신되는 즉시 유효합니다.
validTo	dateTime	0..1	선택 사항. 프로파일이 더 이상 유효하지 않게 되는 시점. 없는 경우, 프로파일은 다른 프로파일로 대체될 때까지 유효합니다.
chargingSchedule	ChargingSchedule	1..1	필수. 시간 경과에 따른 사용 가능한 전력 또는 전류에 대한 제한을 포함합니다.

7.9. ChargingProfileKindType

Enumeration

ChargingProfile에서 사용되는 충전 프로파일의 종류입니다.

값	설명
Absolute	일정 기간은 일정에 정의된 고정된 시간 지점을 기준으로 합니다.
Recurring	일정은 첫 번째 일정 기간에서 주기적으로 다시 시작됩니다.
Relative	일정 기간은 요금 지점에서 결정되는 상황별 시작 지점(예: 트랜잭션 시작)을 기준으로 합니다.

7.10. ChargingProfilePurposeType

Enumeration

ChargingProfile에서 사용되는 충전 프로파일의 목적 타입입니다.

값	설명
ChargePointMaxProfile	전체 충전기에 사용 가능한 최대 전력 또는 전류에 대한 구성입니다.
TxDefaultProfile	충전기에서 구성할 수 있는 기본 프로파일입니다. 새 트랜잭션이 시작되면 이 프로파일을 사용해야 합니다. 단, 충전기에서 수락한 ChargeProfile이 있는 RemoteStartTransaction.req에서 시작된 트랜잭션은 예외입니다.
TxProfile	충전기에서 현재 트랜잭션에 부과할 제약 조건이 있는 프로파일이거나, ChargeProfile이 있는 RemoteStartTransaction.req를 통해 시작될 때 새 트랜잭션에 부과할 제약 조건이 있는 프로파일입니다. 이 목적을 가진 프로파일은 트랜잭션이 종료되면 더 이상 유효하지 않습니다.

7.11. ChargingProfileStatus

Enumeration

SetChargingProfile.req에 대한 응답으로 반환된 상태입니다.

값	설명
Accepted	요청이 수락되었으며 실행됩니다.
Rejected	요청이 수락되지 않았으며 실행되지 않습니다.
NotSupported	충전기에서 요청이 지원되지 않음을 나타냅니다.

7.12. ChargingRateUnitType

Enumeration

GetCompositeSchedule.req 및 ChargingSchedule에서 사용되는 충전 일정이 정의된 단위입니다.

값	설명
W	<p>와트(전력).</p> <p>이것은 총 허용 충전 전력입니다.</p> <p>AC 충전에 사용되는 경우 위상 전류는 다음을 통해 계산해야 합니다. 위상당 전류 = 전력 / (라인 전압 * 위상 수). 계산에 사용되는 "라인 전압"은 측정된 전압이 아니라 해당 영역의 설정 전압(따라서 110볼트의 230)입니다. "위상 수"는 ChargingSchedulePeriod의 numberPhases입니다.</p> <p>일반적으로 DC 충전에 사용하는 것이 더 편리합니다.</p> <p>ChargingSchedulePeriod의 numberPhases가 없는 경우 3으로 가정해야 합니다.</p>
A	<p>암페어(전류).</p> <p>모든 위상의 합이 아닌 위상당 암페어 양입니다.</p> <p>일반적으로 AC 충전에 사용하는 것이 더 편리합니다.</p>

7.13. ChargingSchedule

Class

충전 일정 구조는 [GetCompositeSchedule.conf](#) 및 [ChargingProfile](#)에서 사용되는 충전 기간 목록을 정의합니다.

필드 이름	필드 유형	카디널리티	설명
duration	integer	0..1	선택 사항. 초 단위의 충전 일정 기간입니다. 기간이 비어 있으면 마지막 기간은 무기한 계속되거나 startSchedule이 없는 경우 트랜잭션이 끝날 때까지 계속됩니다.
startSchedule	dateTime	0..1	선택 사항. 절대 일정의 시작점입니다. 없는 경우 일정은 충전 시작을 기준으로 합니다.
chargingRateUnit	ChargingRateUnitType	1..1	필수. 측정 단위 Limit는 다음과 같이 표현됩니다.
chargingSchedulePeriod	ChargingSchedulePeriod	1..*	필수. 시간 경과에 따른 최대 전력 또는 전류 사용량을 정의하는 ChargingSchedulePeriod 요소의 목록입니다. 첫 번째 ChargingSchedulePeriod 의 startSchedule은 항상 0이어야 합니다.
minChargingRate	decimal	0..1	선택 사항. 전기차에서 지원하는 최소 충전 속도입니다. 측정 단위는 chargingRateUnit에 의해 정의됩니다. 이 매개변수는 충전 프로세스가 낮은 충전 속도에서 비효율적인 경우 전력 할당을 최적화하기 위해 로컬 스마트 충전 알고리즘에서 사용하도록 의도되었습니다. 최대 1자리 소수(예: 8.1) 허용

7.14. ChargingSchedulePeriod

Class

충전 일정 기간 구조는 [ChargingSchedule](#)에서 사용되는 것처럼 충전 일정의 기간을 정의합니다.

필드 이름	필드 유형	카디널리티	설명
startPeriod	integer	1..1	필수. 일정 시작부터 초 단위로 기간의 시작입니다. StartPeriod의 값은 이전 기간의 종료 시간도 정의합니다.
limit	decimal	1..1	필수. 해당 chargingRateUnit에서 일정 기간 동안의 충전 속도 제한(예: 암페어 또는 와트)입니다. 최대 1자리 분수 허용(예: 8.1).

필드 이름	필드 유형	카디널리티	설명
numberPhase s	integer	0..1	선택 사항. 충전에 사용할 수 있는 단계 수입니다. 단계 수가 필요한 경우 다른 숫자가 지정되지 않는 한 numberPhases=3으로 가정합니다.

7.15. CiString20Type

Type

대소문자를 구분하지 않는 일반적인 20자 문자열입니다.

필드 유형	설명
CiString[20]	문자열은 대소문자를 구분하지 않습니다.

7.16. CiString25Type

Type

대소문자를 구분하지 않는 일반적인 문자열(25자).

필드 유형	설명
CiString[25]	문자열은 대소문자를 구분하지 않습니다.

7.17. CiString50Type

Type

대소문자를 구분하지 않는 일반적인 문자열(50자).

필드 유형	설명
CiString[50]	문자열은 대소문자를 구분하지 않습니다.

7.18. CiString255Type

Type

대소문자를 구분하지 않는 일반적인 문자열(255자).

필드 유형	설명
CiString[255]	문자열은 대소문자를 구분하지 않습니다.

7.19. CiString500Type

Type

대소문자를 구분하지 않는 일반적인 500자 문자열입니다.

필드 유형	설명
CiString[500]	문자열은 대소문자를 구분하지 않습니다.

7.20. ClearCacheStatus

Enumeration

ClearCache.req에 대한 응답으로 반환된 상태입니다.

값	설명
Accepted	명령이 실행되었습니다.
Rejected	명령이 실행되지 않았습니다.

7.21. ClearChargingProfileStatus

Enumeration

ClearChargingProfile.req에 대한 응답으로 반환된 상태입니다.

값	설명
Accepted	요청이 수락되었으며 실행됩니다.
Unknown	요청과 일치하는 충전 프로파일이 없습니다.

7.22. ConfigurationStatus

Enumeration

ChangeConfiguration.conf의 상태입니다.

값	설명
Accepted	구성 키가 지원되고 설정이 변경되었습니다.
Rejected	구성 키가 지원되지만 설정을 변경할 수 없습니다.
RebootRequired	구성 키가 지원되고 설정이 변경되었지만 재부팅 후에 변경이 가능합니다(충전기가 스스로 재부팅되지 않음).
NotSupported	구성 키가 지원되지 않습니다.

7.23. DataTransferStatus

Enumeration

DataTransfer.conf의 상태입니다.

값	설명
Accepted	메시지가 승인되었고 포함된 요청이 승인되었습니다.
Rejected	메시지가 수락되었지만 포함된 요청이 거부되었습니다.
UnknownMessageId	알 수 없는 메시지 ID 문자열로 인해 메시지를 해석할 수 없습니다.
UnknownVendorId	알 수 없는 공급업체 ID 문자열로 인해 메시지를 해석할 수 없습니다.

7.24. DiagnosticsStatus

Enumeration

DiagnosticsStatusNotification.req의 상태입니다.

값	설명
Idle	충전기가 진단 관련 작업을 수행하지 않습니다. 상태 Idle은 TriggerMessage.req에 의해 트리거된 DiagnosticsStatusNotification.req에서만 사용해야 합니다.
Uploaded	진단 정보가 업로드되었습니다.
UploadFailed	진단 업로드에 실패했습니다.
Uploading	파일이 업로드되고 있습니다.

7.25. FirmwareStatus

Enumeration

FirmwareStatusNotification.req에서 보고된 펌웨어 다운로드 상태입니다.

값	설명
Downloaded	Charge Point에서 새 펌웨어를 다운로드했습니다.
DownloadFailed	Charge Point에서 펌웨어를 다운로드하지 못했습니다.
Downloading	펌웨어를 다운로드하고 있습니다.
Idle	Charge Point에서 펌웨어 업데이트 관련 작업을 수행하지 않습니다. 상태 Idle은 TriggerMessage.req에서 트리거된 FirmwareStatusNotification.req에서만 사용해야 합니다.
InstallationFailed	새 펌웨어 설치에 실패했습니다.
Installing	펌웨어를 설치 중입니다.
Installed	새 펌웨어가 충전기에 성공적으로 설치되었습니다.

7.26. GetCompositeScheduleStatus

Enumeration

GetCompositeSchedule.req에 대한 응답으로 상태가 반환되었습니다.

값	설명
Accepted	요청이 수락되었으며 실행됩니다.
Rejected	요청이 수락되지 않았으며 실행되지 않습니다.

7.27. IdTagInfo

Class

식별자에 대한 상태 정보가 포함되어 있습니다. Authorize, Start Transaction 및 Stop Transaction 응답에서 반환됩니다.

expiryDate가 제공되지 않으면 상태에 종료 날짜가 없음을 의미합니다.

필드 이름	필드 유형	카디널리티	설명
expiryDate	dateTime	0..1	선택 사항. 여기에는 idTag가 Authorization Cache에서 제거되어야 하는 날짜가 포함됩니다.
parentIdTag	IdToken	0..1	선택 사항. 여기에는 parent-identifier가 포함됩니다.
status	AuthorizationStatus	1..1	필수. 여기에는 idTag가 CSMS에서 승인되었는지 여부가 포함됩니다.

7.28. IdToken

Type

인증에 사용할 식별자를 포함합니다. 대소문자를 구분하지 않는 문자열입니다. 향후 릴리스에서는 여러 형태의 식별자를 지원하기 위해 복잡한 유형이 될 수 있습니다.

필드 유형	설명
CiString20Type	IdToken은 대소문자를 구분하지 않습니다.

7.29. KeyValue

Class

특정 구성 키에 대한 정보를 포함합니다. GetConfiguration.conf에서 반환됩니다.

필드 이름	필드 유형	카디널리티	설명
key	CiString50Type	1..1	필수.
readonly	boolean	1..1	필수. ChangeConfiguration 메시지로 값을 설정할 수 있는 경우 False

필드 이름	필드 유형	카디널리티	설명
value	CiString500Type	0..1	선택 사항. 키가 알려져 있지만 설정되지 않은 경우 이 필드가 없을 수 있습니다.

7.30. Location

Enumeration

SampledValue의 값 요소의 선택적 "위치" 필드에 허용되는 값입니다.

값	설명
Body	충전기 본체 내부 측정(예: 온도)
Cable	EV와 충전기 사이의 케이블에서 측정
EV	EV에서 측정
Inlet	네트워크("그리드") 인렛 연결에서 측정
Outlet	커넥터에서 측정. 기본값

7.31. Measurand

Enumeration

MeterValues.req 및 StopTransaction.req 메시지에서 사용되는 값 요소의 선택적 "측정량" 필드의 허용 값입니다. "measurand"의 기본값은 항상 "Energy.Active.Import.Register"입니다.



가져오기는 그리드에서 충전소, EV 또는 기타 부하로의 에너지 흐름입니다. 내보내기는 EV에서 충전소로 및/또는 충전소에서 그리드로의 에너지 흐름입니다.

값	설명
Current.Export	EV에서 순간 전류 흐름
Current.Import	EV로 순간 전류 흐름
Current.Offered	EV에 제공되는 최대 전류
Energy.Active.Export.Register	(그리드로) 수출된 에너지를 측정하는(가장 대표적인) 전기 미터의 "활성 전기 에너지"(Wh 또는 kWh) 레지스터에서 읽은 숫자 값.
Energy.Active.Import.Register	(그리드 공급에서) 수입된 에너지를 측정하는(가장 대표적인) 전기 미터의 "활성 전기 에너지"(Wh 또는 kWh) 레지스터에서 읽은 숫자 값.
Energy.Reactive.Export.Register	(가장 대표적인) 전기 계량기의 "무효 전기 에너지"(VARh 또는 kVARh) 레지스터에서 읽은 숫자 값(그리드로 내보낸 에너지 측정).
Energy.Reactive.Import.Register	(가장 대표적인) 전기 계량기의 "무효 전기 에너지"(VARh 또는 kVARh) 레지스터에서 읽은 숫자 값(그리드 공급에서 가져온 에너지 측정).

값	설명
Energy.Active.Export.Interval	Metervalues ReadingContext에서 지정한 관련 시간 "간격" 동안(그리드로) 내보낸 "유효 전기 에너지"(Wh 또는 kWh)의 절대량, "ClockAlignedDataInterval" 및 "MeterValueSampleInterval"에 대한 해당 간격 기간 구성 값(초)
Energy.Active.Import.Interval	연관된 시간 "간격" 동안 가져온(그리드 공급에서) "활성 전기 에너지"(Wh 또는 kWh)의 절대량으로, Metervalues ReadingContext에서 지정하고, "ClockAlignedDataInterval" 및 "MeterValueSampleInterval"에 대한 해당 간격 기간 구성 값(초)입니다.
Energy.Reactive.Export.Interval	연관된 시간 "간격" 동안 내보낸(그리드로) "무효 전기 에너지"(VARh 또는 kVARh)의 절대량으로, Metervalues ReadingContext에서 지정하고, "ClockAlignedDataInterval" 및 "MeterValueSampleInterval"에 대한 해당 간격 기간 구성 값(초)입니다.
Energy.Reactive.Import.Interval	연관된 시간 "간격" 동안 그리드 공급에서 가져온 "무효 전기 에너지"(VARh 또는 kVARh)의 절대량으로, Metervalues ReadingContext와 "ClockAlignedDataInterval" 및 "MeterValueSampleInterval"에 대한 해당 간격 기간 구성 값(초)으로 지정됩니다.
Frequency	전력선 주파수의 순간 판독값. 참고: OCPP 1.6에는 주파수에 대한 UnitOfMeasure가 없으며, 측정 대상인 SampledValue에 대한 UnitOfMeasure는 주파수가 헤르츠입니다.
Power.Active.Export	EV에서 내보낸 순간 유효 전력입니다. (W 또는 kW)
Power.Active.Import	EV에서 가져온 순간 유효 전력입니다. (W 또는 kW)
Power.Factor	전체 에너지 흐름의 순간 역률(power factor)
Power.Offered	EV에 제공된 최대 전력
Power.Reactive.Export	EV에서 내보낸 순간 무효 전력(var 또는 kvar)
Power.Reactive.Import	EV에서 가져온 순간 무효 전력(var 또는 kvar)
RPM	RPM 단위의 팬 속도
SoC	충전 차량의 충전 상태(백분율)
Temperature	충전기 내부의 온도 판독값
Voltage	순간 AC RMS 공급 전압



단일 충전 트랜잭션 또는 비트랜잭션 소비자(예: 충전소 내부 전원 공급, 전체 공급)와 관련된 모든 "등록" 값은 시간에 따라 단조롭게 증가해야 합니다.

보고된 "레지스터" 값에 해당하는 실제 에너지 양은 문제의 레지스터 값에서 거래 시작 시점이나 기타 관련 시작 참조 시점에 기록/보고된 레지스터 값을 뺀 값으로 계산됩니다. 무결성을 향상시키기 위해 "레지스터" 값은 전기 계측 하드웨어의 비휘발성 레지스터에서 직접 읽은 그대로 보고되어야 하며, 거래 시작 시점에 0으로 재기반되면 안 됩니다. 이는 하드웨어 결함, 배선 오류, 사기 등으로 인해 연속적인 거래 간에 발생할 수 있는 "누락된 에너지"를 식별할 수 있게 해줍니다. 이를 통해 중앙 시스템은 동일한 커넥터에서 이전 거래의 종료 레지스터 값과 이후 거래의 시작 레지스터 값이 동일함을 확인할 수 있습니다.

7.32. MessageTrigger

Enumeration

TriggerMessage.req에서 트리거할 요청 유형입니다.

값	설명
BootNotification	BootNotification 요청을 트리거합니다.
DiagnosticsStatusNotification	DiagnosticsStatusNotification 요청을 트리거합니다.
FirmwareStatusNotification	FirmwareStatusNotification 요청을 트리거합니다.
Heartbeat	Heartbeat 요청을 트리거합니다.
MeterValues	MeterValues 요청을 트리거합니다.
StatusNotification	StatusNotification 요청을 트리거합니다.

7.33. MeterValue

Class

[MeterValues.req](#) 및 [StopTransaction.req](#)에서 하나 이상의 샘플링된 값의 컬렉션입니다. [MeterValue](#)의 모든 샘플링된 값은 동일한 시점에서 샘플링됩니다.

필드 이름	필드 유형	카디널리티	설명
timestamp	dateTime	1..1	필수. 측정된 값의 타임스탬프
sampledValue	SampledValue	1..*	필수. 하나 이상의 측정된 값

7.34. Phase

Enumeration

[SampledValue](#)에서 사용되는 Phase입니다. Phase는 측정된 값을 해석하는 방법을 지정합니다. 모든 Phase 값이 모든 [Measurands](#)에 적용되는 것은 아니라는 점에 유의하세요.

값	설명
L1	L1에서 측정
L2	L2에서 측정
L3	L3에서 측정
N	중립에서 측정
L1-N	중립 도체를 기준으로 L1에서 측정
L2-N	중립 도체를 기준으로 L2에서 측정
L3-N	중립 도체를 기준으로 L3에서 측정
L1-L2	L1과 L2 사이에서 측정

값	설명
L2-L3	L2와 L3 사이에서 측정
L3-L1	L3과 L1 사이에서 측정

7.35. ReadingContext

Enumeration

값의 컨텍스트 필드 값 [SampledValue](#)입니다.

값	설명
Interruption.Begin	인터럽트 시작 시 가져온 값입니다.
Interruption.End	인터럽트 후 재개할 때 가져온 값입니다.
Other	다른 상황에 대한 값입니다.
Sample.Clock	클럭 정렬 간격에서 가져온 값입니다.
Sample.Periodic	트랜잭션 시작 시간을 기준으로 주기적 샘플로 가져온 값입니다.
Transaction.Begin	트랜잭션 시작 시 가져온 값입니다.
Transaction.End	트랜잭션 종료 시 가져온 값입니다.
Trigger	TriggerMessage.req 에 대한 응답으로 가져온 값입니다.

7.36. Reason

Enumeration

[StopTransaction.req](#)에서 트랜잭션을 중단한 이유입니다.

값	설명
DeAuthorized	StartTransaction.conf 에서 승인 상태로 인해 트랜잭션이 중단되었습니다.
EmergencyStop	비상 정지 버튼이 사용되었습니다.
EVDIsconnected	케이블이 분리되었고 차량이 유도 충전 장치에서 멀어졌습니다.
HardReset	하드 리셋 명령이 수신되었습니다.
Local	충전기에서 사용자의 요청에 따라 로컬에서 중단되었습니다. 이는 트랜잭션의 정기적 종료입니다. 예: RFID 태그 제시, 중지 버튼을 누름.
Other	기타 이유입니다.
PowerLoss	전원이 완전히 끊겼습니다.

값	설명
Reboot	로컬에서 시작된 재설정/재부팅이 발생했습니다. (예를 들어 감시자(watchdog)가 작동했습니다.)
Remote	사용자의 요청에 따라 원격으로 중단되었습니다. 이는 트랜잭션의 정기적 종료입니다. 예: 스마트폰 앱을 사용한 종료, (비로컬) 선불 크레딧 초과.
SoftReset	소프트 리셋 명령이 수신되었습니다.
UnlockCommand	CSMS에서 커넥터 잠금 해제 명령을 보냈습니다.

7.37. RecurrencyKindType

Enumeration

[ChargingProfile](#)에서 사용되는 충전 프로파일의 반복 유형입니다.

값	설명
Daily	일정은 startSchedule과 같은 시간에 24시간마다 다시 시작됩니다.
Weekly	일정은 startSchedule과 같은 시간과 요일에 7일마다 다시 시작됩니다.

7.38. RegistrationStatus

Enumeration

[BootNotification.req](#)에 대한 응답으로 등록한 결과입니다.

값	설명
Accepted	충전기가 CSMS에서 수락되었습니다.
Pending	CSMS에서 아직 충전기를 수락할 준비가 되지 않았습니다. CSMS에서 정보를 검색하거나 충전기를 준비하기 위해 메시지를 보낼 수 있습니다.
Rejected	충전기가 CSMS에서 수락되지 않았습니다. 이는 CSMS에서 충전기 ID를 알지 못할 때 발생할 수 있습니다.

7.39. RemoteStartStopStatus

Enumeration

[RemoteStartTransaction.req](#) 또는 [RemoteStopTransaction.req](#) 요청의 결과입니다.

값	설명
Accepted	명령이 실행됩니다.
Rejected	명령이 실행되지 않습니다.

7.40. ReservationStatus

Enumeration

ReserveNow.conf의 상태입니다.

값	설명
Accepted	예약이 이루어졌습니다.
Faulted	커넥터 또는 지정된 커넥터가 오류 상태이므로 예약이 이루어지지 않았습니다.
Occupied	예약이 이루어지지 않았습니다. 모든 커넥터 또는 지정된 커넥터가 점유되었습니다.
Rejected	예약이 이루어지지 않았습니다. 충전소가 예약을 수락하도록 구성되지 않았습니다.
Unavailable	커넥터 또는 지정된 커넥터가 사용할 수 없는 상태이므로 예약이 이루어지지 않았습니다.

7.41. ResetStatus

Enumeration

Reset.req의 결과입니다.

값	설명
Accepted	명령이 실행됩니다.
Rejected	명령이 실행되지 않습니다.

7.42. ResetType

Enumeration

Reset.req에서 요청한 리셋 유형입니다.

값	설명
Hard	모든 하드웨어를 다시 시작합니다. 충전기는 진행 중인 트랜잭션을 일반적인 과정으로 중지할 필요가 없습니다. 가능한 경우 충전기는 BootNotification.conf를 통해 CSMS에서 다시 시작하고 수락한 후 이전에 진행 중이던 트랜잭션에 대해 StopTransaction.req를 보냅니다. 이것은 제대로 작동하지 않는 충전기에 대한 최후의 수단으로, "하드" 재설정을 보내면 (대기 중인) 정보가 손실될 수 있습니다.
Soft	진행 중인 트랜잭션을 일반적인 과정으로 중지하고 모든 진행 중인 트랜잭션에 대해 StopTransaction.req를 보냅니다. 그런 다음 애플리케이션 소프트웨어를 다시 시작해야 합니다(가능하면 경우, 그렇지 않는 경우에는 프로세서/컨트롤러를 다시 시작합니다).

7.43. SampledValue

Class

MeterValues에 단일 샘플링된 값입니다. 각 값에는 선택적 필드가 수반될 수 있습니다.

필드 이름	필드 유형	카디널리티	설명
value	String	1..1	필수. "Raw"(10진수) 숫자 또는 "SignedData" 값입니다. 필드 유형은 디지털 서명된 데이터 판독을 허용하기 위한 "문자열"입니다. 온도 및 전류와 같은 측정량에 대한 분수 값을 허용하기 위해 10진수 숫자 값도 허용됩니다.
context	ReadingContext	0..1	선택 사항. 세부 값의 유형: 시작, 끝 또는 샘플. 기본값 = "Sample.Periodic"
format	ValueFormat	0..1	선택 사항. 원시 또는 서명된 데이터. 기본값 = "Raw"
measurand	Measurand	0..1	선택 사항. 측정 유형. 기본값 = "Energy.Active.Import.Register"
phase	Phase	0..1	선택 사항. 측정된 값을 해석하는 방법을 나타냅니다. 예를 들어 L1과 중립(L1-N) 사이 모든 위상 값이 모든 <i>Measurands</i> 에 적용되는 것은 아니라는 점에 유의하세요. 위상이 없는 경우 측정된 값은 전체 값으로 해석됩니다.
location	Location	0..1	선택 사항. 측정 위치. 기본값="Outlet"
unit	UnitOfMeasure	0..1	선택 사항. 값의 단위. (기본) 측정량이 "Energy" 유형인 경우 기본값 = "Wh".

7.44. TriggerMessageStatus

Enumeration

TriggerMessage.conf의 상태입니다.

값	설명
Accepted	요청된 알림이 전송됩니다.
Rejected	요청된 알림이 전송되지 않습니다.
NotImplemented	요청된 알림은 구현되지 않았거나 알 수 없기 때문에 전송할 수 없습니다.

7.45. UnitOfMeasure

Enumeration

SampledValue에서 사용되는 Value 요소의 선택적 "unit" 필드의 허용 가능한 값입니다. "unit"의 기본값은 항상 "Wh"입니다.

값	설명
Wh	와트시(에너지). 기본값.
kWh	킬로와트시(에너지).
varh	Var시(무효 에너지).
kvarh	킬로와트시(무효 에너지).
W	와트(전력).

값	설명
kW	킬로와트(전력).
VA	볼트암페어(피상 전력, apparent power).
kVA	킬로볼트 암페어(피상 전력).
var	바(무효 전력, reactive power).
kvar	킬로바(무효 전력).
A	암페어(전류).
V	전압(r.m.s. AC).
Celsius	섭씨(온도).
Fahrenheit	화씨(온도).
K	켈빈(온도).
Percent	퍼센트.

7.46. UnlockStatus

Enumeration

[UnlockConnector.req](#)에 대한 응답 상태입니다.

값	설명
Unlocked	커넥터가 성공적으로 잠금 해제되었습니다.
UnlockFailed	커넥터 잠금 해제 실패: 충전기가 커넥터 잠금을 해제하려고 했지만 커넥터가 여전히 잠겨 있거나 잠금 해제 메커니즘이 실패했음을 감지했습니다.
NotSupported	충전기에 커넥터 잠금이 없거나 ConnectorId를 알 수 없습니다.

7.47. UpdateStatus

Enumeration

[SendLocalList.req](#)에 대한 업데이트 유형입니다.

값	설명
Accepted	로컬 인증 목록이 성공적으로 업데이트되었습니다.
Failed	로컬 인증 목록을 업데이트하지 못했습니다.
NotSupported	충전기에서 로컬 인증 목록의 업데이트를 지원하지 않습니다.

값	설명
VersionMismatch	차등 업데이트 요청의 버전 번호가 현재 목록의 버전 번호보다 작거나 같습니다.

7.48. UpdateType

Enumeration

`SendLocalList.req`에 대한 업데이트 유형입니다.

값	설명
Differential	현재 로컬 인증 목록을 이 메시지의 값으로 업데이트해야 함을 나타냅니다.
Full	현재 로컬 인증 목록을 이 메시지의 값으로 바꿔야 함을 나타냅니다.

7.49. ValueFormat

Enumeration

`SampledValue`의 값 요소를 해석하는 방법을 지정하는 형식입니다.

값	설명
Raw	데이터는 정수/10진수 숫자 데이터로 해석해야 합니다.
SignedData	데이터는 16진수 데이터로 인코딩된 부호 있는 이진 데이터 블록으로 표현됩니다.

Chapter 8. Firmware and Diagnostics File Transfer

이 섹션은 규범적입니다.

지원되는 전송 프로토콜은 구성 키 `SupportedFileTransferProtocols`에 의해 제어됩니다. FTP, FTPS, HTTP, HTTPS(CSL)

8.1. Download Firmware

충전기에 새 펌웨어에 대한 알림이 전송되면 이 펌웨어를 다운로드할 수 있어야 합니다. CSMS는 펌웨어를 다운로드할 수 있는 URL을 요청 메시지에 제공합니다. URL에는 펌웨어를 다운로드하는 데 사용해야 하는 프로토콜도 포함되어 있습니다.

펌웨어는 FTP 또는 FTPS를 통해 다운로드하는 것이 좋습니다. FTP(S)는 HTTP보다 대용량 바이너리 데이터에 더 잘 최적화되어 있습니다. 또한 FTP(S)에는 다운로드를 재개할 수 있는 기능이 있습니다. 다운로드가 중단된 경우 충전기는 이미 다운로드한 부분 다음 부분부터 다운로드를 재개할 수 있습니다. FTP URL은 `ftp://user:password@host:port/path` 형식이며, 여기서 `user:password@`, `:password` 또는 `:port` 부분은 제외될 수 있습니다.

올바른 펌웨어가 다운로드되었는지 확인하려면 펌웨어도 디지털 서명을 적용하는 것이 좋습니다.

8.2. Upload Diagnostics

충전기에 진단 파일을 업로드하라는 요청이 있을 경우, CSMS는 충전기가 파일을 업로드해야 하는 URL을 요청에 제공합니다. URL에는 파일을 업로드하는 데 사용해야 하는 프로토콜도 포함되어 있습니다.

진단 파일은 FTP 또는 FTPS를 통해 다운로드하는 것이 좋습니다. FTP(S)는 HTTP보다 대용량 바이너리 데이터에 더 잘 최적화되어 있습니다. 또한 FTP(S)는 업로드를 재개할 수 있습니다. 업로드가 중단된 경우, 충전기는 이미 업로드한 부분 다음 부분부터 업로드를 재개할 수 있습니다. FTP URL은 `ftp://user:password@host:port/path` 형식이며, 여기서 `user:password@`, `:password` 또는 `:port` 부분은 제외될 수 있습니다.

Chapter 9. Standard Configuration Key Names & Values

다음은 이 사양에서 표준화된 역할이 있는 모든 구성 키 목록입니다. 목록은 [Feature Profiles](#)로 구분됩니다. 특정 프로파일에 언급된 필수 구성 키는 해당 프로파일을 지원하는 경우에만 충전기에서 지원해야 합니다.

부울 유형이 있는 선택적 구성 키의 경우 키 목록이 없는 [GetConfiguration.req](#)에 대한 응답에서 구성 키에 다음 규칙이 적용됩니다.

- 키가 있는 경우 충전기는 키로 구성된 기능을 제공하며 키 값을 설정하여 활성화하거나 비활성화할 수 있습니다.
- 키가 없는 경우 충전기는 키로 구성할 수 있는 기능을 제공하지 않습니다.

"접근성" 속성은 특정 구성 키의 값이 읽기 전용("R")인지 읽기-쓰기("RW")인지 보여줍니다. 키가 읽기 전용인 경우, CSMS는 [GetConfiguration](#)을 사용하여 키의 값을 읽을 수 있지만 쓸 수는 없습니다. 접근성이 읽기-쓰기인 경우, CSMS는 [ChangeConfiguration](#)을 사용하여 키의 값을 쓸 수도 있습니다.

9.1. Core Profile

9.1.1. AllowOfflineTxForUnknownId

Required/optional	optional
Accessibility	RW
Type	boolean
Description	이 키가 있으면 충전기는 Unknown Offline Authorization 을 지원합니다. 이 키가 <i>true</i> 값을 보고하면 Unknown Offline Authorization 이 활성화됩니다.

9.1.2. AuthorizationCacheEnabled

Required/optional	optional
Accessibility	RW
Type	boolean
Description	이 키가 있으면 충전기에서 Authorization Cache 를 지원합니다. 이 키가 <i>true</i> 값을 보고하면 Authorization Cache 가 활성화됩니다.

9.1.3. AuthorizeRemoteTxRequests

Required/optional	required
Accessibility	R 또는 RW. 선택은 충전기 구현에 달려 있습니다.
Type	boolean
Description	RemoteStartTransaction.req 메시지 형태로 트랜잭션을 시작하라는 원격 요청이 트랜잭션을 시작하기 위한 로컬 작업처럼, 사전에 승인되어야 하는지 여부입니다.

9.1.4. BlinkRepeat

Required/optional	optional
Accessibility	RW
Type	integer
Unit	times
Description	신호를 보낼 때 충전기 조명을 깜박이는 횟수입니다.

9.1.5. ClockAlignedDataInterval

Required/optional	required
Accessibility	RW
Type	integer
Unit	seconds
Description	<p>클럭 정렬 데이터 간격의 크기(초). 이것은 00:00:00(자정)부터 시작하여 하루에 균등하게 간격을 둔 집계 간격 세트의 크기(초)입니다. 예를 들어, 값 900(15분)은 매일을 96개의 15분 간격으로 나누어야 함을 나타냅니다.</p> <p>클럭 정렬 데이터가 전송되는 경우 해당 간격은 ISO8601 표준에 따라 표현된 시작 시간과(선택 사항) 기간 간격 값으로 식별됩니다. 모든 "기간별" 데이터(예: 에너지 판독값)는 전체 간격(또는 트랜잭션의 시작 또는 끝에서 부분 간격)에 걸쳐 누적(에너지와 같은 "흐름" 유형 측정값의 경우)되거나 평균(다른 값의 경우)이 되어야 하며, 각 간격의 끝에서 전송(활성화된 경우)되어야 하며, 간격 시작 시간 타임스탬프가 포함되어야 합니다.</p> <p>규칙에 따라 "0"(숫자 0) 값은 클럭 정렬 데이터가 전송되지 않아야 함을 의미하는 것으로 해석됩니다.</p>

9.1.6. ConnectionTimeOut

Required/optional	required
Accessibility	RW
Type	integer
Unit	seconds
Description	간격 *상태 시작부터: '준비 중' 초기 트랜잭션까지 EV 운전자가 충전 케이블 커넥터를 적절한 소켓에 (올바르게) 삽입하지 못하여 트랜잭션이 자동으로 취소됩니다. 충전기는 원래 상태, 아마도 '사용 가능'으로 돌아가야 합니다.

9.1.7. ConnectorPhaseRotation

Required/optional	required
Accessibility	RW
Type	CSL

Description	<p>커넥터의 전기 미터(또는 없는 경우 그리드 연결)에 대한 커넥터당 위상 회전입니다. 커넥터당 가능한 값은 다음과 같습니다.</p> <p>NotApplicable(단상 또는 DC 충전기의 경우)</p> <p>Unknown(아직 알려지지 않음)</p> <p>RST(표준 참조 위상)</p> <p>RTS(역방향 참조 위상)</p> <p>SRT(역방향 240도 회전)</p> <p>STR(표준 120도 회전)</p> <p>TRS(표준 240도 회전)</p> <p>TSR(역방향 120도 회전)</p> <p>R은 위상 1(L1), S는 위상 2(L2), T는 위상 3(L3)으로 식별할 수 있습니다.</p> <p>알려진 경우 충전기는 인덱스 번호 0(0)을 사용하여 그리드 연결과 주 에너지계 사이의 위상 회전도 보고할 수 있습니다.</p> <p>값은 CSL로 보고되며 형식은 다음과 같습니다. 0.RST, 1.RST, 2.RTS</p>
--------------------	--

9.1.8. ConnectorPhaseRotationMaxLength

Required/optional	optional
Accessibility	R
Type	integer
Description	ConnectorPhaseRotation 구성 키의 최대 항목 수입니다.

9.1.9. GetConfigurationMaxKeys

Required/optional	required
Accessibility	R
Type	integer
Description	GetConfiguration.req PDU에서 요청된 구성 키의 최대 개수입니다.

9.1.10. HeartbeatInterval

Required/optional	required
Accessibility	RW
Type	integer
Unit	seconds
Description	충전기가 Heartbeat.req PDU를 보내야 하는 CSMS과의 비활성 간격(OCPP 교환 없음)입니다.

9.1.11. LightIntensity

Required/optional	optional
Accessibility	RW
Type	integer
Unit	%
Description	충전기를 비추는 최대 강도의 백분율 조명합니다.

9.1.12. LocalAuthorizeOffline

Required/optional	required
Accessibility	RW
Type	boolean
Description	충전기가 <i>offline</i> 일 때 로컬에서 승인된 식별자에 대한 트랜잭션을 시작할지 여부입니다.

9.1.13. LocalPreAuthorize

Required/optional	required
Accessibility	RW
Type	boolean
Description	온라인 상태일 때 충전기가 CSMS에서 Authorize.conf 를 기다리거나 요청하지 않고 로컬로 승인된 식별자에 대한 트랜잭션을 시작할지 여부입니다.

9.1.14. MaxEnergyOnInvalidId

Required/optional	optional
Accessibility	RW
Type	integer
Unit	Wh
Description	트랜잭션 시작 후 CSMS에서 식별자가 무효화될 때 전달되는 최대 에너지(Wh)입니다.

9.1.15. MeterValuesAlignedData

Required/optional	required
-------------------	----------

Accessibility	RW
Type	CSL
Description	매 <code>ClockAlignedDataInterval</code> 초마다 <code>MeterValues.req</code> PDU에 포함될 시계 정렬 측정량(Clock-aligned measurand(s))입니다.

9.1.16. MeterValuesAlignedDataMaxLength

Required/optional	optional
Accessibility	R
Type	integer
Description	<code>MeterValuesAlignedData</code> 구성 키의 최대 항목 수입니다.

9.1.17. MeterValuesSampledData

Required/optional	required
Accessibility	RW
Type	CSL
Description	<p><code>MeterValues.req</code> PDU에 <code>MeterValueSampleInterval</code> 초마다 포함될 샘플링된 측정량입니다. 해당되는 경우 측정량은 Voltage.L1과 같은 선택적 단계와 결합됩니다.</p> <p>기본값: "Energy.Active.Import.Register"</p>

9.1.18. MeterValuesSampledDataMaxLength

Required/optional	optional
Accessibility	R
Type	integer
Description	<code>MeterValuesSampledData</code> 구성 키의 최대 항목 수입니다.

9.1.19. MeterValueSampleInterval

Required/optional	required
Accessibility	RW
Type	integer
Unit	seconds

Description	"MeterValues" PDU에서 전송하도록 의도된 측정(또는 기타) 데이터 샘플링 간격입니다. 요금 청구 세션 데이터(ConnectorId>0)의 경우, 샘플은 요금 청구 트랜잭션 시작부터 이 간격으로 주기적으로 수집되어 전송됩니다. 관례에 따라 "0"(숫자 0) 값은 샘플링된 데이터가 전송되지 않아야 함을 의미하는 것으로 해석됩니다.
--------------------	---

9.1.20. MinimumStatusDuration

Required/optional	optional
Accessibility	RW
Type	integer
Unit	seconds
Description	StatusNotification.req PDU가 CSMS로 전송되기 전에 충전기 또는 커넥터 상태가 안정되는 최소 기간입니다.

9.1.21. NumberOfConnectors

Required/optional	required
Accessibility	R
Type	integer
Description	이 충전기의 물리적 충전 커넥터 수입니다.

9.1.22. ResetRetries

Required/optional	required
Accessibility	RW
Type	integer
Unit	times
Description	충전기 재설정을 재시도하는 횟수입니다.

9.1.23. StopTransactionOnEVSideDisconnect

Required/optional	required
Accessibility	RW
Type	boolean
Description	<i>true</i> 로 설정하면 충전기는 케이블이 EV에서 분리될 때 트랜잭션을 관리적으로 중지해야 합니다.

9.1.24. StopTransactionOnInvalidId

Required/optional	required
Accessibility	RW
Type	boolean
Description	충전기가 이 트랜잭션에 대해 StartTransaction.conf 에서 <i>Accepted</i> 가 아닌 승인 상태를 수신할 때 진행 중인 트랜잭션을 중지할지 여부입니다.

9.1.25. StopTxnAlignedData

Required/optional	required
Accessibility	RW
Type	CSL
Description	TransactionData 요소에 포함될 시계 정렬 주기적 측정값 StopTransaction.req MeterValues.req 트랜잭션의 모든 'ClockAlignedDataInterval'에 대한 PDU입니다.

9.1.26. StopTxnAlignedDataMaxLength

Required/optional	optional
Accessibility	R
Type	integer
Description	StopTxnAlignedData 구성 키의 최대 항목 수입니다.

9.1.27. StopTxnSampledData

Required/optional	required
Accessibility	RW
Type	CSL
Description	충전 세션 시작부터 MeterValueSampleInterval 초마다 StopTransaction.req PDU의 TransactionData 요소에 포함될 샘플링된 측정량입니다.

9.1.28. StopTxnSampledDataMaxLength

Required/optional	optional
Accessibility	R

Type	integer
Description	StopTxnSampledData 구성 키의 최대 항목 수입니다.

9.1.29. SupportedFeatureProfiles

Required/optional	required
Accessibility	R
Type	CSL
Description	지원되는 Feature Profiles 목록입니다. 가능한 프로파일 식별자는 다음과 같습니다: Core, FirmwareManagement, LocalAuthListManagement, Reservation, SmartCharging 및 RemoteTrigger.

9.1.30. SupportedFeatureProfilesMaxLength

Required/optional	optional
Accessibility	R
Type	integer
Description	SupportedFeatureProfiles 구성 키의 최대 항목 수입니다.

9.1.31. TransactionMessageAttempts

Required/optional	required
Accessibility	RW
Type	integer
Unit	times
Description	CSMS에서 처리하지 못할 때 충전기에서 트랜잭션 관련 메시지를 전송하려고 시도해야 하는 빈도입니다.

9.1.32. TransactionMessageRetryInterval

Required/optional	required
Accessibility	RW
Type	integer
Unit	seconds
Description	CSMS에서 처리하지 못한 트랜잭션 관련 메시지를 다시 제출하기 전에 충전기에서 기다려야 하는 시간입니다.

9.1.33. UnlockConnectorOnEVSideDisconnect

Required/optional	required
Accessibility	RW
Type	boolean
Description	<i>true</i> 로 설정하면 EV에서 케이블이 분리될 때 충전기가 충전기 쪽의 케이블을 잠금 해제해야 합니다.

9.1.34. WebSocketPingInterval

Required/optional	optional
Accessibility	RW
Type	integer
Unit	seconds
Description	웹소켓 구현에만 해당합니다. 0은 클라이언트 측 웹소켓 Ping/Pong을 비활성화합니다. 이 경우 ping/pong이 없거나, 서버가 ping을 시작하고 클라이언트가 Pong으로 응답합니다. 양수 값은 ping 사이의 초 수로 해석됩니다. 음수 값은 허용되지 않습니다. ChangeConfiguration은 REJECTED 결과를 반환해야 합니다.

9.2. Local Auth List Management Profile

9.2.1. LocalAuthListEnabled

Required/optional	required
Accessibility	RW
Type	boolean
Description	Local Authorization List 가 활성화되어 있는지 여부입니다.

9.2.2. LocalAuthListMaxLength

Required/optional	required
Accessibility	R
Type	integer
Description	Local Authorization List 에 저장할 수 있는 최대 식별 수 목록입니다.

9.2.3. SendLocalListMaxLength

Required/optional	required
-------------------	----------

Accessibility	R
Type	integer
Description	단일 <code>SendLocalList.req</code> 에서 보낼 수 있는 최대 식별 수입입니다.

9.3. Reservation Profile

9.3.1. ReserveConnectorZeroSupported

Required/optional	optional
Accessibility	R
Type	boolean
Description	이 구성 키가 있고 <code>true</code> 로 설정: 충전기가 커넥터 0에서 <code>reservations</code> 를 지원합니다.

9.4. Smart Charging Profile

9.4.1. ChargeProfileMaxStackLevel

Required/optional	required
Accessibility	R
Type	integer
Description	충전 프로파일의 최대 StackLevel입니다. 정의된 숫자는 <code>Charging Profile Purposes</code> 당 설치된 충전 일정의 최대 허용 개수도 나타냅니다.

9.4.2. ChargingScheduleAllowedChargingRateUnit

Required/optional	required
Accessibility	R
Type	CSL
Description	<code>ChargingSchedule</code> 에서 사용할 수 있는 지원 수량 목록입니다. 허용 값: 'Current' 및 'Power'

9.4.3. ChargingScheduleMaxPeriods

Required/optional	required
Accessibility	R
Type	integer

Description	<code>ChargingSchedule</code> 당 정의할 수 있는 최대 기간 수입니다.
--------------------	--

9.4.4. `ConnectorSwitch3to1PhaseSupported`

Required/optional	optional
Accessibility	R
Type	boolean
Description	정의되고 true인 경우, 이 충전기는 트랜잭션 중에 3상에서 1상으로 전환을 지원합니다.

9.4.5. `MaxChargingProfilesInstalled`

Required/optional	required
Accessibility	R
Type	integer
Description	한 번에 설치할 수 있는 최대 충전 프로파일 수입니다.

Appendix A: New in OCPP 1.6

OCPP 1.5와 비교하여 OCPP 1.6에서 다음과 같은 변경 사항이 적용되었습니다[OCPP1.5]:

- 스마트 충전이 추가되었습니다.
- 전송 프로토콜로 WebSocket을 통한 JSON에 대한 바인딩이 추가되어 데이터 사용량이 줄어들고 NAT 라우터를 통한 OCPP 통신이 가능해졌습니다. [OCPP JSON Specification](#)을 참조하세요.
- [ChargePointStatus](#) 열거형에 추가 상태가 추가되어 CPO와 최종 사용자에게 충전기의 현재 상태에 대한 추가 정보를 제공합니다
- [MeterValues.req](#)의 구조가 XML 속성 사용을 제거하기 위해 변경되었습니다. 이는 JSON 지원에 필요합니다(JSON에서는 속성 지원 없음).
- [Measurand](#) 열거형에 추가 값이 추가되어 충전기 제조업체가 EV의 충전 상태와 같은 새로운 정보를 CSMS에 보낼 수 있습니다.
- [TriggerMessage](#) 메시지가 추가되어 CSMS가 충전기에 정보를 요청할 수 있습니다.
- [BootNotification.conf](#)에서 사용되는 [RegistrationStatus](#) 열거형에 새로운 *Pending* 멤버가 추가되었습니다.
- 더 많고 명확한 구성 키가 추가되어 CPO가 충전기에서 다양한 비즈니스 사례를 구성하는 방법을 더 명확하게 알 수 있습니다.
- 메시지와 구성 키가 프로파일로 분할되어 OCPP를 점진적으로 또는 일부만 구현하기가 더 쉬워졌습니다.
- 알려진 모호성이 제거되었습니다(예: [UnlockConnector.req](#)를 사용하는 시기, 응답하는 방법 [RemoteStart/Stop](#), 커넥터 번호)

A.1. Updated/New Messages

- [BootNotification.req](#)
 - 최대 길이를 적용하기 위해 `IccId`와 `Imsi`를 `CiString[]`으로 변경합니다.
- [BootNotification.conf](#)
 - `heartbeatInterval`에서 `interval`로, `interval`은 이제 `heartbeat` 이외의 다른 용도로도 사용되며, 사양에서 수정해야 합니다.
 - `Pending` 상태가 추가되었습니다.
- [ChargePointErrorCode](#)
 - 열거형 값 추가: `InternalError`, `LocalListConflict` 및 `UnderVoltage`
 - 열거형 값 ``Mode3Error``를 ``EVCommunicationError``로 이름 변경합니다.
- [ChargePointStatus](#)
 - 열거형 값 `Occupied`` ``Preparing`, `Charging`, `SuspendedEVSE`, `SuspendedEV` 및 ``Finishing``으로 대체했습니다.
- [ChargingRateUnitType](#)
 - 새로 추가됨
- [ConfigurationStatus](#)
 - 열거형 `RebootRequired` 추가했습니다.
- [ClearChargingProfile.req](#)
 - 새로 추가됨
- [ClearChargingProfile.conf](#)
 - 새로 추가됨
- [DiagnosticsStatus](#)
 - 열거형 업로드 및 유틸 상태 추가했습니다.
- [FirmwareStatus](#)
 - 열거형 다운로드, 설치 및 유틸 상태 추가했습니다.

- [GetCompositeSchedule.req](#)
 - 새로 추가됨
- [GetCompositeSchedule.conf](#)
 - 새로 추가됨
- [Location](#)
 - 열거형 케이블 및 EV 추가했습니다.
- [Measurand](#)
 - 열거형 `Current.Offered`, `Frequency`, `Power.Factor`, `Power.Offered`, `RPM` 및 `'SoC'`를 추가했습니다.
- [MeterValues.req](#)
 - 복잡한 데이터 구조의 전면 개편했습니다.
 - 'phase' 필드 추가했습니다.
- [ReadingContext](#)
 - 열거형 `Trigger` 및 `Other` 추가했습니다.
- [RemoteStartTransaction.req](#)
 - `ChargingProfile` 선택 사항 추가했습니다.
- [SendLocalList.req](#)
 - 해시를 제거했습니다.
- [SendLocalList.conf](#)
 - 해시를 제거했습니다.
- [SetChargingProfile.req](#)
 - 새로 추가됨
- [SetChargingProfile.conf](#)
 - 새로 추가됨
- [StatusNotification.req](#)
 - `states` 값을 전면 개편했습니다.
 - 새로운 오류 코드를 추가했습니다.
 - 커넥터 ID 0은 사용 가능, 사용 불가능 및 오류 상태만 가질 수 있습니다.
- [StopTransaction.req](#)
 - 명시적이고 필요한 중지 이유 추가했습니다.
- [TriggerMessage.req](#)
 - 새로 추가됨
- [TriggerMessage.conf](#)
 - 새로 추가됨
- [UnlockConnector.conf](#)
 - `UnlockStatus` 열거형의 전면 개편했습니다.
- [UnitOfMeasure](#)
 - 화씨, K, 퍼센트, VA, kVA 추가했습니다.
 - 전압을 V로, 암페어를 A로 이름을 변경했습니다.