OCPP 2.0.1

Table of Contents

Disclaimer . . . 2
Generic . . . 3
Version History . . . 4
1. Scope . . . 5
1.1. OCPP 2.0.1 . . . 5
1.2. OCPP 2.0.1 Edition 3 . . . 5
2. Conventions, Terminology and Abbreviations . . . 6
2.1. Conventions . . . 6
2.2. Terminology . . . 7
2.3. Abbreviations . . . 10
2.4. Actors . . . 13
2.5. References . . . 13
2.6. Definition of Transaction . . . 16
2.7. ISO 15118 support . . . 17
3. Generic Requirements . . . 19
3.1. Time Format Requirements . . . 19
3.2. Message Timeouts . . . 20
3.3. Language support . . . 20
A. Security . . . 21
1. OCPP Security . . . 22
1.1. Security Objectives . . . 22
1.2. Design Considerations . . . 22
1.3. Security Profiles . . . 23
1.4. Keys used in OCPP . . . 32
1.5. Certificate Revocation . . . 34
2. Use cases & Requirements . . . 36
A01 - Update Charging Station Password for HTTP Basic Authentication . . . 36
A02 - Update Charging Station Certificate by request of CSMS . . . 38
A03 - Update Charging Station Certificate initiated by the Charging Station . . . 43
A04 - Security Event Notification. . . . 46
A05 - Upgrade Charging Station Security Profile . . . 47
B. Provisioning . . . 50
1. Introduction . . . 51
1.1. Transactions before being accepted by a CSMS . . . 51
2. Use cases & Requirements . . . 52
2.1. Booting a Charging Station . . . 52
B01 - Cold Boot Charging Station . . . 52
B01 - Cold Boot Charging Station - Requirements . . . 54
B02 - Cold Boot Charging Station - Pending . . . 55
B03 - Cold Boot Charging Station - Rejected . . . 58
B04 - Offline Behavior Idle Charging Station . . . 60
2.2. Configuring a Charging Station . . . 62
B05 - Set Variables . . . 62
B06 - Get Variables . . . 65
B07 - Get Base Report . . . 68
B09 - Setting a new NetworkConnectionProfile . . . 74
2.3. Resetting a Charging Station . . . 78
B11 - Reset - Without Ongoing Transaction . . . 78
B12 - Reset - With Ongoing Transaction . . . 82
C. Authorization . . . 86
1. Introduction . . . 87
1.1. ID Tokens . . . 87
1.2. Group ID Tokens . . . 87
1.3. Authorization Cache . . . 88
1.4. Local Authorization List . . . 88
1.5. Unknown Offline Authorization . . . 89
2. Use cases & Requirements . . . 90
2.1. Authorization options . . . 90
C01 - EV Driver Authorization using RFID . . . 90

C02 - Authorization using a start button 95
C03 - Authorization using credit/debit card 97
C04 - Authorization using PIN-code 100
C05 - Authorization for CSMS initiated transactions 102
C06 - Authorization using local id type 105
2.2. ISO 15118 Authorization 108
C07 - Authorization using Contract Certificates 108
C08 - Authorization at EVSE using ISO 15118 External Identification Means (EIM) 112
2.3. GroupId 113
C09 - Authorization by GroupId 113
2.4. Authorization Cache 117
C10 - Store Authorization Data in the Authorization Cache 117
C11 - Clear Authorization Data in Authorization Cache 119
C12 - Start Transaction - Cached Id 121
2.5. Local Authorization list 123
C13 - Offline Authorization through Local Authorization List 123
C14 - Online Authorization through Local Authorization List 125
2.6. Offline Authorization 127
C15 - Offline Authorization of unknown Id 127
2.7. Master Pass 131
C16 - Stop Transaction with a Master Pass 131
D. LocalAuthorizationList Management 134
1. Introduction 135
2. Use cases & Requirements 136
D01 - Send Local Authorization List 136
D02 - Get Local List Version 138
E. Transactions 140
1. Introduction 141
1.1. Flexible transaction start/stop 141
1.2. TransactionId generation 142
1.3. Delivering transaction-related messages 142
1.4. Authorization 143
1.5. Clarification for optional fields in TransactionEventRequest 143
2. Use cases & Requirements 145
2.1. OCPP transaction mechanism 145
E01 - Start Transaction options 145
E03 - Start Transaction - IdToken First 156
E04 - Transaction started while Charging Station is offline 160
E05 - Start Transaction - Id not Accepted 164
E06 - Stop Transaction options 167
E07 - Transaction locally stopped by IdToken 173
E08 - Transaction stopped while Charging Station is offline 178
E09 - When cable disconnected on EV-side: Stop Transaction 181
E10 - When cable disconnected on EV-side: Suspend Transaction 184
E12 - Inform CSMS of an Offline Occurred Transaction 190
E14 - Check Transaction status 194
2.2. Interrupting and Stopping ISO 15118 Charging 196
E15 - End of charging process 196
F. RemoteControl 198
1. Introduction 199
2. Use cases & Requirements 200
2.1. Remote Transaction Control 200
F01 - Remote Start Transaction - Cable Plugin First 200
F02 - Remote Start Transaction - Remote Start First 205
F04 - Remote Stop ISO 15118 Charging from CSMS 214
2.2. Unlock Connector 216
F05 - Remotely Unlock Connector 216
2.3. Remote Trigger 218
F06 - Trigger Message 218
G. Availability 221
1. Introduction 222
2. Use cases & Requirements 223

G01 - Status Notification 223
G02 - Heartbeat 226
G02 - Heartbeat - Requirements 226
G03 - Change Availability EVSE/Connector 228
G04 - Change Availability Charging Station 231
G05 - Lock Failure 233
H. Reservation 235
1. Introduction 236
2. Use cases & Requirements 237
H01 - Reservation 237
H02 - Cancel Reservation 242
H03 - Use a reserved EVSE 244
H04 - Reservation Ended, not used 247
I. TariffAndCost 249
1. Introduction 250
1.1. Why no structured tariff information? 250
2. Show EV Driver-specific Tariff Information 251
I01 - Show EV Driver-specific Tariff Information 251
I02 - Show EV Driver Running Total Cost During Charging 252
I03 - Show EV Driver Final Total Cost After Charging 253
I04 - Show Fallback Tariff Information 255
I05 - Show Fallback Total Cost Message 257
I06 - Update Tariff Information During Transaction 258
J. MeterValues 260
1. Introduction 261
2. Configuration 262
2.1. Transaction Meter Values 262
2.2. Clock-Aligned Meter Values 262
2.3. Multiple Locations/Phases 263
2.4. Signed Meter Values 263
2.5. Configuration Examples 264
3. Use cases & Requirements 265
3.1. MeterValues 265
J01 - Sending Meter Values not related to a transaction 265
J02 - Sending transaction related Meter Values 268
3.2. ISO 15118 MeterValue signing 270
J03 - Charging Loop with metering information exchange 271
K. SmartCharging 273
1. Introduction 274
2. Types of Smart Charging 275
2.1. Internal Load Balancing 275
2.2. Central Smart Charging 275
2.3. Local Smart Charging 276
2.4. External Smart Charging Control Signals 276
3. Charging Profiles 278
3.1. Introduction 278
3.2. Charging profile purposes 278
3.3. Charging profile recurrency 278
3.4. Stacking charging profiles 279
3.5. Combining Charging Profile Purposes 279
3.6. Example Charging Profile 280
4. Smart Charging Signals to a Charging Station from Multiple Actors 283
5. Use cases & Requirements 285
5.1. General Smart Charging 285
K01 - SetChargingProfile 285
K02 - Central Smart Charging 290
K03 - Local Smart Charging 295
K04 - Internal Load Balancing 298
K05 - Remote Start Transaction with Charging Profile 300
K06 - Offline Behavior Smart Charging During Transaction 302
K07 - Offline Behavior Smart Charging at Start of Transaction 303
K08 - Get Composite Schedule 306

K09 - Get Charging Profiles 307
K10 - Clear Charging Profile 309
5.2. External Charging Limit based Smart Charging 312
K11 - Set / Update External Charging Limit With Ongoing Transaction 312
K12 - Set / Update External Charging Limit Without Ongoing Transaction 314
K13 - Reset / Release External Charging Limit 316
K14 - External Charging Limit with Local Controller 317
5.3. ISO 15118 based Smart Charging 320
K15 - Charging with load leveling based on High Level Communication 320
K16 - Renegotiation initiated by CSMS 323
K17 - Renegotiation initiated by EV 327
L. FirmwareManagement 331
1. Introduction 332
2. Use cases & Requirements 333
L01 - Secure Firmware Update 333
L02 - Non-Secure Firmware Update 339
L03 - Publish Firmware file on Local Controller 343
L04 - Unpublish Firmware file on Local Controller 345
M. ISO 15118 CertificateManagement 347
1. Introduction 348
2. ISO 15118 Certificates 351
2.1. ISO 15118 Certificate structure 351
2.2. Using ISO 15118 Certificates in OCPP 353
2.3. 15118 communication set-up 353
2.4. Certificate - Use Case mapping 354
2.5. Use cases from ISO 15118 relevant for OCPP 355
3. Use cases & Requirements 357
M01 - Certificate installation EV 357
M02 - Certificate Update EV 358
M03 - Retrieve list of available certificates from a Charging Station 359
M04 - Delete a specific certificate from a Charging Station 360
M05 - Install CA certificate in a Charging Station 362
M06 - Get V2G Charging Station Certificate status 364
N. Diagnostics 367
1. Introduction 368
2. Use cases & Requirements 369
2.1. Logging 369
N01 - Retrieve Log Information 369
2.2. Configure Monitoring 372
N02 - Get Monitoring report 372
N03 - Set Monitoring Base 375
N04 - Set Variable Monitoring 376
N05 - Set Monitoring Level 380
N06 - Clear / Remove Monitoring 381
2.3. Monitoring Events 382
N07 - Alert Event 382
N08 - Periodic Event 385
2.4. Customer Information 387
N09 - Get Customer Information 387
N10 - Clear Customer Information 388
O. DisplayMessage 391
1. Introduction 392
2. Use cases & Requirements 393
O01 - Set DisplayMessage 393
O02 - Set DisplayMessage for Transaction 396
O03 - Get All DisplayMessages 398
O04 - Get Specific DisplayMessages 401
O05 - Clear DisplayMessage 403
O06 - Replace DisplayMessage 405
P. DataTransfer 407
1. Introduction 408
2. Use cases & Requirements 409

P01 - Data Transfer to the Charging Station 409
P02 - Data Transfer to the CSMS 411
Messages, Datatypes & Enumerations 413
1. Messages 414
1.1. Authorize 414
1.2. BootNotification 414
1.3. CancelReservation 415
1.4. CertificateSigned 416
1.5. ChangeAvailability 416
1.6. ClearCache 417
1.7. ClearChargingProfile 417
1.8. ClearDisplayMessage 418
1.9. ClearedChargingLimit 419
1.10. ClearVariableMonitoring 419
1.11. CostUpdated 420
1.12. CustomerInformation 420
1.13. DataTransfer 421
1.14. DeleteCertificate 422
1.15. FirmwareStatusNotification 422
1.16. Get15118EVCertificate 423
1.17. GetBaseReport 424
1.18. GetCertificateStatus 424
1.19. GetChargingProfiles 425
1.20. GetCompositeSchedule 426
1.21. GetDisplayMessages 426
1.22. GetInstalledCertificateIds 427
1.23. GetLocalListVersion 428
1.24. GetLog 428
1.25. GetMonitoringReport 429
1.26. GetReport 429
1.27. GetTransactionStatus 430
1.28. GetVariables 431
1.29. Heartbeat 431
1.30. InstallCertificate 431
1.31. LogStatusNotification 432
1.32. MeterValues 432
1.33. NotifyChargingLimit 433
1.34. NotifyCustomerInformation 433
1.35. NotifyDisplayMessages 434
1.36. NotifyEVChargingNeeds 434
1.37. NotifyEVChargingSchedule 435
1.38. NotifyEvent 436
1.39. NotifyMonitoringReport 436
1.40. NotifyReport 437
1.41. PublishFirmware 438
1.42. PublishFirmwareStatusNotification 438
1.43. ReportChargingProfiles 439
1.44. RequestStartTransaction 440
1.45. RequestStopTransaction 440
1.46. ReservationStatusUpdate 441
1.47. ReserveNow 441
1.48. Reset 442
1.49. SecurityEventNotification 443
1.50. SendLocalList 443
1.51. SetChargingProfile 444
1.52. SetDisplayMessage 445
1.53. SetMonitoringBase 445
1.54. SetMonitoringLevel 446
1.55. SetNetworkProfile 447
1.56. SetVariableMonitoring 448
1.57. SetVariables 449
1.58. SignCertificate 449

1.59. StatusNotification . . . 450
1.60. TransactionEvent . . . 450
1.61. TriggerMessage . . . 452
1.62. UnlockConnector . . . 453
1.63. UnpublishFirmware . . . 454
1.64. UpdateFirmware . . . 454
2. Datatypes . . . 456
2.1. ACChargingParametersType . . . 456
2.2. AdditionalInfoType . . . 456
2.3. APNType . . . 456
2.4. AuthorizationData . . . 457
2.5. CertificateHashDataChainType . . . 457
2.6. CertificateHashDataType . . . 458
2.7. ChargingLimitType . . . 458
2.8. ChargingNeedsType . . . 458
2.9. ChargingProfileCriterionType . . . 459
2.10. ChargingProfileType . . . 459
2.11. ChargingSchedulePeriodType . . . 460
2.12. ChargingStationType . . . 462
2.13. ClearChargingProfileType . . . 462
2.14. ClearMonitoringResultType . . . 462
2.15. ComponentType . . . 463
2.16. ComponentVariableType . . . 463
2.17. CompositeScheduleType . . . 464
2.18. ConsumptionCostType . . . 464
2.19. CostType . . . 464
2.20. DCChargingParametersType . . . 465
2.21. EventDataType . . . 465
2.22. EVSEType . . . 466
2.23. FirmwareType . . . 467
2.24. GetVariableDataType . . . 467
2.25. GetVariableResultType . . . 467
2.26. IdTokenInfoType . . . 468
2.27. IdTokenType . . . 469
2.28. LogParametersType . . . 469
2.29. MessageContentType . . . 470
2.30. MessageInfoType . . . 470
2.31. MeterValueType . . . 471
2.32. ModemType . . . 471
2.33. MonitoringDataType . . . 471
2.34. NetworkConnectionProfileType . . . 472
2.35. OCSPRequestDataType . . . 473
2.36. RelativeTimeIntervalType . . . 473
2.37. ReportDataType . . . 473
2.38. SalesTariffEntryType . . . 474
2.39. SalesTariffType . . . 474
2.40. SampledValueType . . . 475
2.41. SetMonitoringDataType . . . 475
2.42. SetMonitoringResultType . . . 477
2.43. SetVariableDataType . . . 478
2.44. SetVariableResultType . . . 479
2.45. SignedMeterValueType . . . 479
2.46. StatusInfoType . . . 480
2.47. TransactionType . . . 480
2.48. UnitOfMeasureType . . . 481
2.49. VariableAttributeType . . . 481
2.50. VariableCharacteristicsType . . . 482
2.51. VariableMonitoringType . . . 483
2.52. VariableType . . . 484
2.53. VPNType . . . 485
3. Enumerations . . . 486
3.1. APNAuthenticationEnumType . . . 486

3.2. AttributeEnumType 486
3.3. AuthorizationStatusEnumType 486
3.4. AuthorizeCertificateStatusEnumType 487
3.5. BootReasonEnumType 487
3.6. CancelReservationStatusEnumType 488
3.7. CertificateActionEnumType 488
3.8. CertificateSignedStatusEnumType 488
3.9. CertificateSigningUseEnumType 489
3.10. ChangeAvailabilityStatusEnumType 489
3.11. ChargingLimitSourceEnumType 489
3.12. ChargingProfileKindEnumType 490
3.13. ChargingProfilePurposeEnumType 490
3.14. ChargingProfileStatusEnumType 490
3.15. ChargingRateUnitEnumType 491
3.16. ChargingStateEnumType 491
3.17. ClearCacheStatusEnumType 491
3.18. ClearChargingProfileStatusEnumType 492
3.19. ClearMessageStatusEnumType 492
3.20. ClearMonitoringStatusEnumType 492
3.21. ComponentCriterionEnumType 493
3.22. ConnectorEnumType 493
3.23. ConnectorStatusEnumType 494
3.24. CostKindEnumType 495
3.25. CustomerInformationStatusEnumType 495
3.26. DataEnumType 495
3.27. DataTransferStatusEnumType 496
3.28. DeleteCertificateStatusEnumType 496
3.29. DisplayMessageStatusEnumType 496
3.30. EnergyTransferModeEnumType 497
3.31. EventNotificationEnumType 497
3.32. EventTriggerEnumType 497
3.33. FirmwareStatusEnumType 498
3.34. GenericDeviceModelStatusEnumType 499
3.35. GenericStatusEnumType 499
3.36. GetCertificateIdUseEnumType 499
3.37. GetCertificateStatusEnumType 500
3.38. GetChargingProfileStatusEnumType 500
3.39. GetDisplayMessagesStatusEnumType 500
3.40. GetInstalledCertificateStatusEnumType 500
3.41. GetVariableStatusEnumType 501
3.42. HashAlgorithmEnumType 501
3.43. IdTokenEnumType 501
3.44. InstallCertificateStatusEnumType 502
3.45. InstallCertificateUseEnumType 502
3.46. Iso15118EVCertificateStatusEnumType 503
3.47. LocationEnumType 503
3.48. LogEnumType 503
3.49. LogStatusEnumType 503
3.50. MeasurandEnumType 504
3.51. MessageFormatEnumType 505
3.52. MessagePriorityEnumType 506
3.53. MessageStateEnumType 506
3.54. MessageTriggerEnumType 506
3.55. MonitorEnumType 507
3.56. MonitoringBaseEnumType 508
3.57. MonitoringCriterionEnumType 508
3.58. MutabilityEnumType 508
3.59. NotifyEVChargingNeedsStatusEnumType 508
3.60. OCPPInterfaceEnumType 509
3.61. OCPPTransportEnumType 509
3.62. OCPPVersionEnumType 509
3.63. OperationalStatusEnumType 510

3.64. PhaseEnumType . . . 510
3.65. PublishFirmwareStatusEnumType . . . 511
3.66. ReadingContextEnumType . . . 511
3.67. ReasonEnumType . . . 512
3.68. RecurrencyKindEnumType . . . 513
3.69. RegistrationStatusEnumType . . . 513
3.70. ReportBaseEnumType . . . 513
3.71. RequestStartStopStatusEnumType . . . 514
3.72. ReservationUpdateStatusEnumType . . . 514
3.73. ReserveNowStatusEnumType . . . 514
3.74. ResetEnumType . . . 515
3.75. ResetStatusEnumType . . . 515
3.76. SendLocalListStatusEnumType . . . 515
3.77. SetMonitoringStatusEnumType . . . 516
3.78. SetNetworkProfileStatusEnumType . . . 516
3.79. SetVariableStatusEnumType . . . 516
3.80. TransactionEventEnumType . . . 517
3.81. TriggerMessageStatusEnumType . . . 517
3.82. TriggerReasonEnumType . . . 517
3.83. UnlockStatusEnumType . . . 518
3.84. UnpublishFirmwareStatusEnumType . . . 519
3.85. UpdateEnumType . . . 519
3.86. UpdateFirmwareStatusEnumType . . . 519
3.87. UploadLogStatusEnumType . . . 520
3.88. VPNEnumType . . . 520
Referenced Components and Variables . . . 521
1. Controller Components . . . 522
2. Referenced Components and Variables . . . 523
2.1. General . . . 523
2.2. Security related . . . 534
2.3. Authorization related . . . 538
2.4. Authorization Cache related . . . 540
2.5. Local Authorization List Management related . . . 542
2.6. Transaction related . . . 545
2.7. Metering related . . . 549
2.8. Reservation related . . . 555
2.9. Smart Charging related . . . 556
2.10. Tariff & Cost related . . . 559
2.11. Diagnostics related . . . 562
2.12. Display Message related . . . 565
2.13. Charging Infrastructure related . . . 566
2.14. ISO 15118 Related . . . 571

Part 2 - Specification

Edition 3 FINAL, 2024-05-06

Disclaimer

Copyright © 2010 – 2024 Open Charge Alliance. All rights reserved.

This document is made available under the *Creative Commons Attribution-NoDerivatives 4.0 International Public License* (https://creativecommons.org/licenses/by-nd/4.0/legalcode).

Generic

Version History

버전	날짜	설명
2.0.1 Edition 3	2024-05-06	OCPP 2.0.1 Edition 3. OCPP 2.0.1 Part 2에서 2024-04까지의 모든 정오표가 이 버전의 사양에 병합되었습니다.
2.0.1 Edition 2	2022-12-15	OCPP 2.0.1 Edition 2. OCPP 2.0.1 Part 2 - Errata v2.0의 모든 정오표가 이 버전의 사양에 병합되었습니다.
2.0.1	2020-03-31	OCPP 2.0의 최종 버전.
2.0	2018-04-11	OCPP 2.0 2018년 4월 1.0 이후 첫 번째 주요 릴리스. 많은 새로운/개선된/개정된 기능 개정된 문서

1. Scope

이 문서는 EV 충전 인프라에서 **충전기** 와 **충전기 관리 시스템** 간에 사용되는 프로토콜을 사용 사례 형태로 정의합니다. 프로토콜에 한쪽 또는 다른 쪽에서 특정 작업이나 응답이 필요한 경우 이 문서에 명시됩니다.

이 사양의 이 부분은 통신 기술을 정의하지 않습니다. 광범위한 호환성을 보장하기 위해 OCPP 2.0.1은 JSON으로 제한됩니다. JSON 구현에 대한 사양은 "Part 4 - JSON over WebSockets implementation guide"에 있습니다.

1.1. OCPP 2.0.1

이 사양은 OCPP 버전 2.0.1을 정의합니다.

OCPP 2.0이 출시된 후 OCPP 2.0에서 몇 가지 문제가 발견되었습니다. 이러한 문제 중 일부는 OCPP 1.6에서 수행된 것처럼 사양 텍스트에만 오류를 발행하여 수정할 수 없었지만, 이전 버전과 호환되지 않는 프로토콜의 기계 판독 가능 스키마 정의 파일을 변경해야 했습니다.

시장에서의 혼란과 현장에서 발생할 수 있는 상호 운용성 문제를 방지하기 위해 OCA는 이 버전의 이름을 2.0.1로 지정하기로 했습니다. OCPP 2.0.1에는 메시지에 대한 수정 사항뿐만 아니라 현재까지 알려진 모든 문제에 대한 수정 사항이 포함되어 있습니다.

이 버전은 OCPP 2.0을 대체합니다. OCA는 OCPP 구현자에게 더 이상 OCPP 2.0을 구현하지 않고 앞으로는 버전 2.0.1만 사용하도록 조언합니다.

원칙적으로 기존 번호가 매겨진 요구 사항은 업데이트되거나 제거될 뿐이며, 이전에 사용된 요구 사항 번호는 완전히 다른 요구 사항에 재사용되지 않습니다.

특별히 달리 명시되어 있지 않는 한 "OCPP 2.0"이라는 언급은 개정판 2.0.1을 말합니다.

1.2. OCPP 2.0.1 Edition 3

2022년에 출시된 OCPP 2.0.1 Edition 2에 대한 여러 오류 문서가 출시되었습니다. 2024-04까지의 오류는 이 문서 "OCPP-2.0.1_edition3_part2_specification"에 통합되어 더 이상 사양과 함께 이러한 오류 정오표를 같이 읽을 필요가 없습니다. 버전 3에 오류 통합은 OCPP 메시지의 스키마에 영향을 미치지 않습니다. 특정 오류에는 요구 사항의 변경 사항 또는 새로운 요구 사항이 포함되었지만 요구 사항에 명백한 오류가 포함되어 있고 설명된 대로 구현되지 않거나 구현할 수 없는 경우에만 해당됩니다. 새로운 요구 사항은 이미 암묵적으로 존재하는 경우에만 추가되었습니다. 이러한 변경 사항은 Open Charge Alliance의 기술 작업 그룹에서 논의되었거나 제안되었습니다.

OCPP 2.0.1 part 2의 부록은 새로운 OCPP 릴리스 없이 업데이트될 수 있습니다. 이는 주로 OCPP 장치 모델의 구성 요소와 변수에 관한 것이며, 선택 사항인 한 새로운 구성 요소나 변수로 확장이 가능합니다.

2. Conventions, Terminology and Abbreviations

2.1. Conventions

2.1.1. Normative

모든 절과 부록은 명확하게 정보 제공을 목적으로 한다고 표시된 경우를 제외하고는 규범적입니다.

2.1.2. Requirements take precedence over text

설명적(narrative) 텍스트와 사양 문서의 요구 사항 사이에 (겉보기에) 충돌이 있는 경우, 요구 사항이 우선합니다.

2.1.3. Requirement Keywords

이 문서의 핵심 단어 "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", "OPTIONAL"은 RFC-2119 [RFC2119]에 설명된 대로 해석되어야 하며, 다음 추가 설명 조항이 적용됩니다.

"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED"라는 용어 사용과 관련된 "특정 상황에서의 타당한 이유(valid reasons in particular circumstances)"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.

2.1.4. Primitive Datatypes

사양에는 다음과 같은 기본 데이터 유형(primitive datatype)이 언급되어 있습니다.

Table 1. 기본 데이터 유형

데이터 유형	설명
string	UTF-8 문자 집합에 정의된 문자를 사용할 수 있습니다.
integer	32비트(31비트 해상도, 1부호 비트) 선행 0 없음(No leading 0's) 더하기 기호 없음 허용되는 값 예: 1234, -1234 허용되지 않음: 01234, +1234
decimal	충전기에서 보고하는 데이터의 경우 소스 데이터의 전체 해상도를 유지해야 합니다. 충전기로 전송되는 소수점에는 소수점 여섯 자리를 초과할 수 없습니다.
identifierString	대소문자를 구분하지 않는 dataType이며 다음 문자 집합의 문자만 포함할 수 있습니다. a-z, A-Z, 0-9, '*', '-', '_', '=', ':', '+', '|', '@', '.'
dateTime	CSMS와 충전기 간에 교환된 모든 시간 값은 [RFC3339]에 정의된 대로 포맷해야 합니다. 또한 소수점 이하 초에는 추가 제한이 있습니다. 소수점 이하 자릿수는 최대 3을 초과해서는 안 됩니다. 예 1: 2019-04-12T23:20:50.52Z는 UTC 기준으로 2019년 4월 12일 23시 이후 20분 50.52초를 나타냅니다. 예 2: 2019-12-19T16:39:57+01:00은 UTC(중부 유럽 시간)에서 +01:00 오프셋을 적용하여 2019년 12월 19일 16시 39분 57초를 나타냅니다.
passwordString	다음 문자 집합의 문자만 포함할 수 있는 UTF-8 인코딩된 대소문자 구분 문자열입니다. "a-z", "A-Z", "0-9" 또는 다음 제한된 기호 집합 중 하나를 포함합니다: * - _ : + | @.
AnyType	지정된 길이 또는 형식이 없는 데이터입니다.

데이터 유형	설명
boolean	다음 값만 허용됩니다: "false" 및 "true"

2.1.5. Normal communication

달리 명시하지 않는 한, 모든 사용 사례와 요구 사항은 충전기와 CSMS(*Online*) 간의 정상적인 통신을 가정합니다.

2.1.6. Field description

대부분의 경우, 메시지와 데이터 유형에서 특정 필드를 사용하는 방법 또는 시기에 대한 추가 설명은 필드 설명에 제공됩니다. Messages 장을 참조하세요.

2.2. Terminology

2.2.1. General Terminology

이 섹션에는 이 문서 전체에서 사용되는 용어가 포함되어 있습니다.

Table 2. 용어

용어	설명
Application	계층 OSI-계층 5-7.
Authentication	인증은 신원이나 속성을 확인하는 프로세스입니다. 인증에 대해 말할 때 사용자 인증(예: 발신자/수신자)과 메시지 인증을 구별해야 합니다.
Block cipher	고정된 블록 길이의 메시지를 암호화/복호화하는 암호화 기본 요소입니다. 예: AES는 한 번에 128비트(16바이트) 블록을 암호화합니다.
Cable Plugged in	이 문서에서는 다음을 의미할 수 있습니다. - 충전기 쪽에 고정된 케이블, EV에 연결된 케이블 - 충전기와 EV에 연결된 케이블 - 무선 충전기가 EV를 감지
Certificate	디지털 인증서는 공개 키 또는 엔터티를 인증합니다. 공개 키 인프라도 참조하세요.
Certificate Management Protocol	PKI 내에서 X.509 디지털 인증서를 관리하는 데 사용되는 인터넷 프로토콜입니다. RFC 4210에 설명되어 있으며 RFC 4211에 설명된 인증서 요청 메시지 형식(CRMF)을 사용합니다.
Charging Cable	EV에서 허용하는 플러그가 장착된 케이블 어셈블리로, EV와 EVSE 간 연결에 사용하도록 의도되었습니다. 한쪽은 EVSE에 영구적으로 부착되거나 EVSE에서 허용하는 플러그가 장착될 수도 있습니다.
Charging Loop	이 사양에서는 ISO 15118-2의 충전 루프 정의를 사용합니다. *ISO 15118에 따라 충전 프로세스를 제어하기 위한 V2G 메시징 단계.*
Charging Profile	다양한 유형의 프로필에 사용되는 일반 충전 프로필입니다. 프로필에 대한 정보를 포함하고 ChargingSchedule을 보유합니다.
Charging Schedule	충전 프로필의 일부입니다. 충전 전력 또는 전류 제한 블록을 정의합니다. 시작 시간과 길이를 포함할 수 있습니다.
Charging Station	충전기는 EV를 충전할 수 있는 물리적 시스템입니다. 충전기에는 하나 이상의 EVSE가 있습니다.

용어	설명
Composite Charging Schedule	충전기에서 계산한 충전 일정입니다. 이는 충전기에 있는 모든 활성 일정과 가능한 지역 제한을 계산한 결과입니다. 지역 제한을 고려할 수 있습니다.
Confidentiality	권한이 있는 엔티티만 기밀 데이터에 액세스할 수 있습니다. 데이터를 무단 액세스로부터 보호하기 위해 암호화할 수 있습니다. 그런 다음 비밀 키에 액세스할 수 있는 엔터티만 데이터를 해독한 후 데이터에 액세스할 수 있습니다.
Connector	이 사양에서 사용되는 용어 커넥터는 충전기에서 독립적으로 작동하고 관리되는 전기 콘센트를 말합니다. 즉, 이는 단일 물리적 커넥터에 해당합니다. 어떤 경우에는 EVSE에 여러 물리적 소켓 유형 및/또는 고정 케이블/커넥터 배열(예: 커넥터)이 있어 다양한 차량 유형(예: 4륜 EV 및 전기 스쿠터)을 용이하게 할 수 있습니다.
Contactor	일반적으로 충전기에서 충전 전원을 켜거나 끄는 데 사용되는 전기 제어 스위칭 장치입니다.
Contract Certificate	15118 통신을 위한 EV의 충전 계약에 대한 유효한 인증서입니다.
Control Pilot signal	충전기에서 EV에 IEC61851-1에서 정의한 최대 전류 한도를 알리는 데 사용하는 신호입니다.
Cost	EV 운전자가 소비한 에너지/시간 등에 대해 지불해야 하는 비용입니다. 세금 포함.
Cryptographic hash function	암호 해시 함수는 단방향 함수처럼 동작해야 합니다. 이 함수는 사전 이미지 저항성, 2차 사전 이미지 저항성 및 충돌 저항성이 있어야 합니다. 입력의 변경은 출력에서 명확하게 다른 결과를 생성해야 합니다. 예: SHA-256. ENISA OCPP Security [1]도 참조하세요.
Cryptography	ENISA 알고리즘, 키 크기 및 매개변수 보고서 [1]는 최신 기술에 대한 개요를 제공합니다.
CSMS	충전기 관리 시스템. 충전기를 관리하고 사용자에게 충전기 사용을 허가하는 정보를 제공하는 시스템입니다.
Data Integrity	무결성 및 메시지 인증을 참조하세요.
Digital Signature	발신자를 인증합니다. 실제로 디지털 서명은 타원 곡선(EC)을 사용하여 구현됩니다.
Encryption	암호화 체계를 사용하여 메시지를 무작위로 보이는 해독 불가능한 문자열(암호문)에 매핑합니다. 복호화는 암호화 프로세스를 역전하며 해당 복호화 키로만 수행할 수 있습니다. 이 복호화 키는 암호화 키(대칭 암호화)와 동일하거나 공개 키 암호 시스템의 개인 키입니다. 키가 비밀로 유지되는 동안에만 메시지의 기밀성을 보장할 수 있습니다.
Energy Management System	지역 및/또는 계약적 제약 및/또는 계약적 인센티브에 따라 지역 부하(소비 및 생산)를 관리하는 장치입니다. 여기에는 PV, 배터리 저장 장치 등의 센서 및 제어와 같은 추가 입력이 있습니다.
Energy Offer Period	충전기가 EV에 에너지를 제공할 준비가 되어 있는 시간입니다.
Energy Transfer Period	EV가 제공된 에너지를 가져가거나 반환하기로 선택하는 시간입니다.
EVSE	EVSE는 한 번에 하나의 EV에 에너지를 공급할 수 있는 충전기의 독립적으로 작동 및 관리되는 부분으로 간주됩니다.
Hash function	메시지를 고정 길이의 비트 문자열(해시 값)에 매핑하는 함수입니다. 암호화 해시 함수도 참조하세요.
Hash value	(암호화) 해시 함수의 출력입니다. 길이는 해시 함수의 사양에 고정되어 있습니다.
High level communication	ISO 15118 시리즈 [ISO15118-1]에 명시된 프로토콜 및 메시지와 물리적 및 데이터 링크 계층을 사용하는 양방향 디지털 통신

용어	설명
Idle State	사용 사례와 시퀀스 다이어그램 모두에서 유휴 상태는 충전기가 사용 사례 관련 작업을 수행하지 않는 상태를 말합니다. 장비가 주요 기능을 즉시 제공할 수 있지만 그렇게 하지 않는 상태입니다.
Integrity	데이터는 권한 없이 변경할 수 없습니다. 메시지 인증도 참조하세요.
Local Controller	CSMS와 하나 이상의 충전기 사이의 논리적 엔티티로, CSMS의 입력에 따라 충전기 그룹의 충전을 제어할 수 있으며, CSMS와 독립적으로 충전기로 메시지를 보낼 수 있습니다.
Master Pass	진행 중인 모든(또는 모든) 트랜잭션을 중지하는 데 사용할 수 있는 ID 토큰입니다. 예를 들어 법 집행 기관 직원이 트랜잭션을 중지하는 데 사용할 수 있습니다.
Master Pass UI	마스터 패스 사용자 인터페이스는 풀 컬러 터치스크린일 수도 있지만, 사용자가 중지할 트랜잭션을 선택할 수 있도록 하는 몇 개의 버튼과 LED 및/또는 사운드일 수도 있습니다.
Message authentication	메시지는 무단 수정으로부터 보호되어야 합니다. 메시지는 항상 진위성을 제공하는 인증 태그와 함께 보내야 합니다. 이러한 인증 태그는 AES-CCM 또는 AES-GCM과 같은 인증된 암호의 두 번째 출력이거나 메시지 인증 코드일 수 있습니다.
Mode of Operation	작동 모드는 블록 암호가 메시지 블록을 처리하는 방식을 지정합니다. CBC 또는 CTR 모드에서 블록 암호를 사용하면 암호화만 제공되지만, CCM 또는 GCM 모드에서 블록 암호를 사용하면 일반 텍스트가 암호화되고 암호 텍스트에 대한 메시지 인증 태그가 생성됩니다.
OCPP-J	WebSocket을 통한 JSON을 통한 OCPP.
Offline	충전기와 CSMS 간에 통신이 불가능합니다. OCPP-J 연결의 경우 WebSocket 연결이 열리지 않습니다.
Password authentication	사용자는 비밀번호 또는 PIN을 사용하여 신원을 증명합니다.
Phase Rotation	전기 계량기(또는 없는 경우 그리드 연결)와 충전기 커넥터 사이의 위상 배선 순서를 정의합니다.
Price	단일 요금 항목의 특정 가격 태그, 예: 18% VAT 포함 kWh당 0.35.
Public-key cryptography	"공개 키가 공개되고 이후 메시지 암호화 또는 디지털 서명 검증에 사용될 수 있는 암호화 방식입니다. 각 공개 키에는 해당 개인 키인 대응 키가 있습니다. 이 키는 비밀로 유지해야 하며 메시지의 암호 해독 또는 디지털 서명에 사용됩니다. 공개 키 기본 요소는 암호화에 대한 계산 복잡도가 높기 때문에 대부분 공개 키가 모든 추가 통신이 암호화되는 공통 대칭 세션 키를 전달하는 데 사용되는 하이브리드 암호화 방식의 일부로 사용됩니다. 공개 키 인프라에서 관리하는 인증서는 공개 키의 진위성을 확인하는 데 사용됩니다. ENISA OCPP Security [12]도 참조하세요. 가장 널리 사용되는 공개 키 암호화 방식은 RSA입니다. 디지털 서명은 타원 곡선 기반(EC) 메커니즘을 사용하여 가장 효율적으로 생성할 수 있습니다."
Public-key infrastructure	인증서를 생성, 관리 및 해지하는 시스템입니다.
Resume regular transaction	시퀀스 다이어그램에서 이 사용 사례/시퀀스 다이어그램이 종료되었음을 나타내는 데 사용되지만, 트랜잭션은 종료되지 않았고 계속 진행되며, 이는 해당 특정 사용 사례의 범위를 벗어납니다.
Requirement	충족해야 할 기준을 전달하는 조항. ISO/IEC 가이드 2:2004, 7.5.
Security Event	장치의 안전한 작동과 관련된 모든 이벤트.
Security Function	액세스 제어, 인증 및 암호화를 포함하여 안전하게 작동하는 데 필요한 장치의 모든 기능.
Session	OCPP의 세션은 EV의 충전 프로세스를 나타내는 일반적인 용어로, 트랜잭션을 포함할 수 있습니다.
Session key	수명이 제한된 대칭 키.

용어	설명
Symmetric cryptography	송신자와 수신자가 동일한 키를 보유합니다. 대칭 기본 요소의 예로는 블록 암호 또는 MAC이 있습니다.
Transaction	OCPP의 트랜잭션은 구성 가능한 매개변수에 따라 시작 및 중지되는 EV 충전의 전체 프로세스의 일부입니다. 이러한 구성 가능한 매개변수는 EV가 연결되거나 EV 운전자가 승인되는 것과 같은 충전 프로세스의 순간을 나타냅니다.
Tariff	충전 시간, 전력 사용량 및 가격에 영향을 미치는 기타 매개변수에 따른 가격 모음.
Use case	사용 사례는 특정 목표를 달성하는 데 필요한 (상호)작용을 설명하는 구조화된 방법입니다. 이 문서에서 사용 사례는 액터 목록, 시나리오 설명, 사후 조건 및 시퀀스 다이어그램으로 구성되며 항상 번호가 매겨진 요구 사항 목록이 뒤따릅니다.
User Authentication	통신 파트너(예: 장치의 사용자)의 신원을 확인합니다. 또한 세션 내내 통신 파트너가 여전히 살아 있는지 확인합니다.

2.2.2. ISO 15118 and OCPP terminology mapping

이 섹션은 정보 제공을 목적으로 합니다.

ISO 15118 용어는 EV 및 충전기 내의 특정 구성 요소를 언급할 때 더 포괄적입니다. 다음 표는 이러한 용어의 "매핑"을 보여줍니다.

Table 3. ISO 15118 및 OCPP 용어 매핑

ISO 15118	OCPP
ChargingProfile(EV가 소비할 예정인 시간별 전력 포함)	NotifyEVChargingSchedule 메시지의 ChargingSchedule과 대략적으로 일치합니다.
SASchedule(특정 시간 동안 EV를 충전하기 위한 보조 액터의 전력 제한)	SetChargingProfile 메시지의 ChargingProfile과 대략적으로 일치합니다.
EVCC(전기 자동차 통신 컨트롤러, Electric Vehicle Comunication Controller)	ISO 15118 통신에 사용되는 EV의 컨트롤러.
Outlet	Connector
SECC(공급 장비 통신 컨트롤러 Supply Equipment Communication Controller)	ISO 15118 통신에 사용되는 충전기의 EVSE의 컨트롤러.
SA(보조 액터, Secondary Actor)	CSMS(또는 다른 백엔드 시스템)

2.3. Abbreviations

2.3.1. General Abbreviations

이 섹션에는 이 문서 전체에서 사용되는 약어가 포함되어 있습니다.

Table 4. 약어

약어	설명
AES	고급 암호화 표준(Advanced Encryption Standard). 이 블록 암호의 원래 이름은 설계자 Vincent Rijmen과 Joan Daemen의 이름을 딴 Rijndael이었습니다.

약어	설명
BEV	배터리 전기 자동차(Battery Electric Vehicle)
CMP	인증서 관리 프로토콜(Certificate Management Protocol)
CS	충전기(Charging Station)
CSL	쉼표로 구분된 목록(Comma Separated List)
CSMS	충전기 관리 시스템(Charging Station Management System)
CSO	충전기 운영자(Charging Station Operator)
DHCP	동적 호스트 구성 프로토콜(Dynamic Host Configuration Protocol)
DNS	도메인 이름 시스템(Domain Name System)
DSO	분배 시스템 운영자(Distribution System Operator)
DST	일광 절약 시간제(Daylight Saving Time)
EC Elliptic	곡선. ENISA OCPP Security [1]도 참조하세요.
ECDSA	타원 곡선 디지털 서명 알고리즘(Elliptic Curve Digital Signature Algorithm).
EMS	에너지 관리 시스템(Energy Management System)
ENISA	유럽 연합 네트워크 및 정보 보안 기관(European Union Agency for Network and Information Security).
EV	전기차(Electric Vehicle)
EVSE	EV 공급 장비(EV Supply Equipment) IEC61851-1
FQDN	정규화된 도메인 이름(Fully Qualified Domain Name)
FTP(S)	파일 전송 프로토콜(보안)(File Transport Protocol (Secure))
HTTP(S)	하이퍼텍스트 전송 프로토콜(보안)(HyperText Transport Protocol (Secure))
ICCID	집적 회로 카드 식별자(Integrated Circuit Card Identifier)
IMSI	국제 모바일 가입 ID(International Mobile Subscription Identity)
JSON	JavaScript 단순 개체 표기법(JavaScript Simple Object Notation)
MAC	메시지 인증 코드(Message authentication code). 데이터 무결성을 제공합니다. 예: CMAC, GMAC. ENISA OCPP Security [1]도 참조하세요.
NAT	네트워크 주소 변환(Network Address Translation)
NIST	국립표준기술원(National Institute of Standards and Technology).
NTP	네트워크 시간 프로토콜(Network Time Protocol)

약어	설명
PDU	프로토콜 데이터 단위(Protocol Data Unit)
PHEV	플러그인 하이브리드 전기 자동차(Plugin Hybrid Electric Vehicle)
RDN	상대적 고유 이름(Relative Distinguished Name)
RSA	발명가인 리베스트(Rivest), 샤미르(Shamir), 애들먼(Adleman)의 이름을 딴 공개 키 암호 시스템.
RSA-PSS	RSA-PSS는 RSA 암호 시스템을 기반으로 한 새로운 서명 방식으로, 보안성을 강화한 것이 특징입니다. 이는 PKCS #1 버전 2.1에서 추가되었으며, OCPP Security [23]를 따릅니다.
RST	3상 전원 연결, 표준 참조 위상(Standard Reference Phasing)
RTS	3상 전원 연결, 역방향 참조 위상(Reversed Reference Phasing)
SRT	3상 전원 연결, 역방향 240도 회전(Reversed 240 degree rotation)
STR	3상 전원 연결, 표준 120도 회전(Standard 120 degree rotation)
TRS	3상 전원 연결, 표준 240도 회전(Standard 240 degree rotation)
TSR	3상 전원 연결, 역방향 120도 회전(Reversed 120 degree rotation)
SC	스마트 충전(Smart Charging)
TLS	전송 계층 보안(Transport Layer Security)
TSO	전송 시스템 운영자(Transmission System Operator)
URI	Uniform Resource Identifier RFC-3986 [RFC3986]
URL	Uniform Resource Locator - 리소스를 식별하는 것 외에도 리소스를 설명하는 방법을 제공하는 URI의 하위 집합을 말합니다. 주요 액세스 메커니즘(예: 네트워크 "위치").
UTC	협정 세계시(Coordinated Universal Time)
WAN	광역 네트워크(Wide Area Network).

2.3.2. ISO 15118 Abbreviations

이 섹션에는 이 문서에서 사용되는 ISO 15118의 약어가 들어 있습니다.

Table 5. ISO 15118 약어

EIM	외부 식별 수단(External Identification Means)
EMAID	E-Mobility 계정 식별자(E-Mobility Account Identifier)
EVCC	EV 통신 컨트롤러(EV Communication Controller)
HLC	고수준 통신(High Level Communication)
HMI	인간-기계 인터페이스(Human Machine Interface)

LAN	로컬 영역 네트워크(Local Area Network)
MO	모빌리티 운영자(Mobility Operator)
OEM	원래 장비 제조업체(Original Equipment Manufacturer)
OCSP	온라인 인증서 상태 프로토콜(Online Certificate Status Protocol)
PWM	펄스 폭 변조(Pulse Width Modulation)
SA	2차 액터(Secondary Actor)
SECC	공급 장비 통신 컨트롤러(Supply Equipment Communication Controller)
V2G	차량-그리드 연결(Vehicle to Grid)

2.4. Actors

이 섹션은 정보 제공을 목적으로 합니다.

OCPP에서 시스템 액터는 기능 또는 장치를 포함합니다.

Table 6. 액터

액터 이름	액터 유형	액터 설명
EV 운전자	액터	충전기에서 EV를 충전하려는 EV 운전자.
커넥터	장치	이 사양에서 사용되는 용어 "커넥터"는 충전기에서 독립적으로 작동하고 관리되는 전기 콘센트를 말합니다. 즉, 이는 단일 물리적 커넥터에 해당합니다. 어떤 경우 EVSE에 여러 개의 물리적 소켓 유형 및/또는 유형(예: 4륜 EV 및 전기 스쿠터)과 같이 여러 개의 커넥터가 있을 수 있습니다.
CSMS	시스템	충전기 관리 시스템: 충전기를 관리하고 사용자가 충전기를 사용하도록 허가하는 정보를 보유합니다.
충전기	장치	충전기는 EV를 충전할 수 있는 물리적 시스템입니다. 충전기에는 하나 이상의 EVSE가 있습니다.
충전기 운영자	행위자	CSMS를 관리하는 당사자.
전기 자동차	장치	전기 자동차, 원격 배터리와 소켓이 있는 분산 에너지 리소스.
로컬 컨트롤러	장치	CSMS와 하나 이상의 충전기 사이의 논리적 엔티티로, CSMS의 입력에 따라 충전기 그룹의 충전을 제어할 수 있습니다.
외부 제어 시스템	행위자	충전기 또는 CSMS에 충전 제한/제약을 부과할 수 있는 외부 시스템(예: DSO 또는 EMS).

2.5. References

2.5.1. Generic references

Table 7. 참조

참조	설명
[DNP3]	분산 네트워크 프로토콜(Distributed Network Protocol). https://www.dnp.org/About/Overview-of-DNP3-Protocol
[EMI3-BO]	"eMI3 표준 버전 V1.0" http://emi3group.com/documents-links/
[IEC60870-5-104]	전기 엔지니어링 및 전력 시스템 자동화 애플리케이션에서 원격 제어(감독 제어 및 데이터 수집)에 사용되는 시스템을 정의하는 표준 세트입니다. https://webstore.iec.ch/publication/3755
[IEC61850-7-420]	분산형 에너지 자원(DER,Distributed Energy Resources)을 위한 통신 표준. https://webstore.iec.ch/publication/6019
[IEC61851-1]	"IEC 61851-1 2017: EV 전도성 충전 시스템 - 1부: 일반 요구 사항" https://webstore.iec.ch/publication/33644
[IEC62196]	IEC 62196: 플러그, 소켓-아웃렛, 차량 커플러 및 차량 인렛 - 전기 자동차의 전도성 충전. https://webstore.iec.ch/publication/6582
[ISO15118-1]	ISO 15118-1은 ISO 15118의 다른 부분을 위한 기초로서 용어와 정의, 일반 요구 사항 및 사용 사례를 지정합니다. 이는 요금 프로세스, 지불 및 부하 평준화에 영향을 미치는 측면에 대한 일반적인 개요와 공통된 이해를 제공합니다. https://webstore.iec.ch/publication/9272
[ISO15118-2]	도로 차량 - 차량 대 그리드 통신 인터페이스 - 2부: 기술 프로토콜 설명 및 OSI(개방형 시스템 상호 연결) 계층 요구 사항, 문서 식별자: 69/216/CDV. https://webstore.iec.ch/publication/9273
[ISO4217]	"ISO 4217: 통화 코드" http://www.iso.org/iso/home/standards/currency_codes.htm
[OCPP2.0-PART4]	"OCPP 2.0.1: 4부 - WebSockets를 통한 JSON 구현 가이드". http://www.openchargealliance.org/downloads/
[OpenADR]	"오픈 자동 수요 대응" http://www.openadr.org/
[RFC1321]	"MD5 메시지 다이제스트 알고리즘" https://tools.ietf.org/html/rfc1321
[RFC2119]	"요구 사항 수준을 나타내는 RFC에서 사용되는 키워드". S. Bradner. 1997년 3월. http://www.ietf.org/rfc/rfc2119.txt
[RFC3339]	"인터넷의 날짜 및 시간: 타임스탬프" https://tools.ietf.org/html/rfc3339
[RFC3986]	"Uniform Resource Identifier(URI): 일반 구문" https://tools.ietf.org/html/rfc3986
[RFC5646]	"언어 식별을 위한 태그" https://tools.ietf.org/html/rfc5646

2.5.2. Security related references

Table 8. 보안 관련 참조

참조	설명
[1]	ENISA 유럽 네트워크 및 정보 보안 기관, 알고리즘, 키 크기 및 매개변수 보고서 2014, 2014. (2016년 1월 17일에 마지막 액세스) https://www.enisa.europa.eu/publications/algorithms-key-size-and-parameters-report-2014
[2]	국립 표준 기술 연구소. FIPS PUB 140-2, 암호화 모듈에 대한 보안 요구 사항, 2001년 5월. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf
[3]	Cooper, D. 등, 인터넷 X.509 공개 키 인프라 인증서 및 인증서 해지 목록(CRL, Certificate Revocation List) 프로필, 인터넷 엔지니어링 태스크포스, 의견 요청 5280, 2008년 5월, http://www.ietf.org/rfc/rfc5280.txt

참조	설명
[4]	Dierks, T. 및 Rescorla, E., 전송 계층 보안(TLS) 프로토콜 버전 1.2, 인터넷 엔지니어링 태스크포스, 의견 요청 5246, 2008년 8월, http://www.ietf.org/rfc/rfc5246.txt
[5]	Eastlake, D., 전송 계층 보안(TLS) 확장: 확장 정의, 인터넷 엔지니어링 태스크포스, 의견 요청 6066, 2011년 1월, http://www.ietf.org/rfc/rfc6066.txt
[6]	McGrew, D. 및 Bailey, D., 전송 계층 보안(TLS)을 위한 AES-CCM 암호 모음, 인터넷 엔지니어링 태스크포스, 의견 요청 6655, 2012년 7월, http://www.ietf.org/rfc/rfc6655.txt
[7]	Rescorla E. 등, 전송 계층 보안(TLS) 재협상 표시 확장, 인터넷 엔지니어링 태스크포스, 의견 요청 5746, 2월 2010, http://www.ietf.org/rfc/rfc5746.txt
[8]	"Russel Housley, Tim Polk, Warwick Ford, David Solo. 인터넷 공개 키 인프라: X.509 인증서 및 인증서 해지 목록(CRL) 프로필, RFC 3280, 2002년 4월." https://www.ietf.org/rfc/rfc3280.txt
[9]	Pettersen. "전송 계층 보안(TLS) 다중 인증서 상태 요청 확장." RFC 6961, 2013년 6월. https://tools.ietf.org/html/rfc6961.
[10]	Hollenbeck, S., "전송 계층 보안 프로토콜 압축 방법", RFC 3749, 2004년 5월. https://www.ietf.org/rfc/rfc3749.txt
[11]	미국 국립표준기술원. 부록 C: 승인된 난수 생성기 FIPS PUB 140-2 [25], 2012년 2월. https://csrc.nist.gov/csrc/media/publications/fips/140/2/final/documents/fips1402annexc.pdf
[12]	Bundesamt für Sicherheit in der Informationstechnik: Anwendungshinweise und Interpretationen zum Schema, AIS 20, Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren, 버전 3.0, 본, 독일, 2013년 5월. (in 독일어) https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_20_pdf.html
[13]	Bundesamt für Sicherheit in der Informationstechnik: Anwendungshinweise und Interpretationen zum Schema, AIS 31, Funktionalitätsklassen und Evaluationsmethodologie fürphykalische Zufallszahlengeneratoren, 버전 3.0, 독일 본, 2013년 5월.(독일어) https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_pdf.html
[14]	"OWASP - 전송 계층 보호 치트 시트. https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet#Extended_Validation_Certificates "
[15]	P. Hoffman 및 W.C.A. Wijngaards, DNNSEC을 위한 타원 곡선 디지털 서명 알고리즘(DSA), 인터넷 엔지니어링 태스크포스(IETF) RFC 6605, 2012년 4월. http://www.ietf. org/rfc/rfc6605.txt
[16]	Adams, C., Farrell, S., Kause, T., 및 T. Mononen, "인터넷 X.509 공개 키 인프라 인증서 관리 프로토콜(CMP)", RFC 4210, 2005년 9월. https://www.ietf.org/rfc/rfc4210.txt
[17]	미국 국립표준기술원. 특별 간행물 800-57 Part 1 Rev. 4, 주요 관리 권장 사항. 2016년 1월. https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-4/final
[[ocpp_security_18]][18]	RFC 2617. HTTP 인증: 기본 및 다이제스트 액세스 인증. https://www.ietf.org/rfc/rfc2617.txt
[19]	RFC 5280 . 인터넷 X.509 공개 키 인프라 인증서 및 인증서 해지 목록(CRL) 프로필. https://www.ietf.org/rfc/rfc5280.txt
[20]	OCPP 1.6. 충전 간 인터페이스 설명 스테이션 및 CSMS. 2015년 10월. http://www.openchargealliance.org/downloads/
[21]	Eekelen, M. van, Poll, E., Hubbers, E., Vieira, B., Broek, F. van den: LaQuSo1의 Enexis 및 ElaadNL을 위한 스마트 EV 충전을 위한 종단간 보안 설계. 2014년 12월 2일. https://www.elaad.nl/smart-charging-end2end-security-design/
[[ocpp_security_22]][22]	RFC 2986. PKCS #10: 인증 요청 구문 사양, 버전 1.7. https://www.ietf.org/rfc/rfc2986.txt

참조	설명
[23]	RSA-PSS. https://tools.ietf.org/html/rfc8017
[24]	Santesson 등 "X.509 인터넷 공개 키 인프라 온라인 인증서 상태 프로토콜 - OCSP" RFC 6960. 2013년 6월
[[ocpp_security_25]] [25]	RFC 2818. TLS를 통한 HTTP. https://tools.ietf.org/html/rfc2818

2.6. Definition of Transaction

이 섹션은 정보 제공을 목적으로 합니다.

가능한 한 많은 비즈니스 사례를 지원하고 특정 비즈니스 사례에 필요하지 않은 너무 많은 메시지가 전송되는 것을 방지하기 위해 OCPP 2.0.1은 트랜잭션 시작 및 중지의 유연한 구성을 지원합니다. 이를 통해 시장 수요에 따라 트랜잭션 시작 및 중지를 정의할 수 있습니다.

자세한 내용은 Flexible transaction start/stop을 참조하세요.

2.6.1. Transaction in relation to Energy Transfer Period

에너지 전송 기간은 EV와 EVSE 간에 에너지가 전송되는 기간입니다. Transaction 동안 여러 개의 Energy Transfer Periods가 있을 수 있습니다.

여러 에너지 전송 기간은 다음 중 하나로 구분할 수 있습니다.

- EVSE가 에너지 전송을 제공하지 않는 EVSE에서 시작된 전송 중단, *또는*
- EV가 EVSE에 전기적으로 연결된 상태로 유지되는 EV에서 시작된 전송 중단, *또는*
- EV가 EVSE에 전기적으로 연결되지 않은 EV에서 시작된 전송 중단.

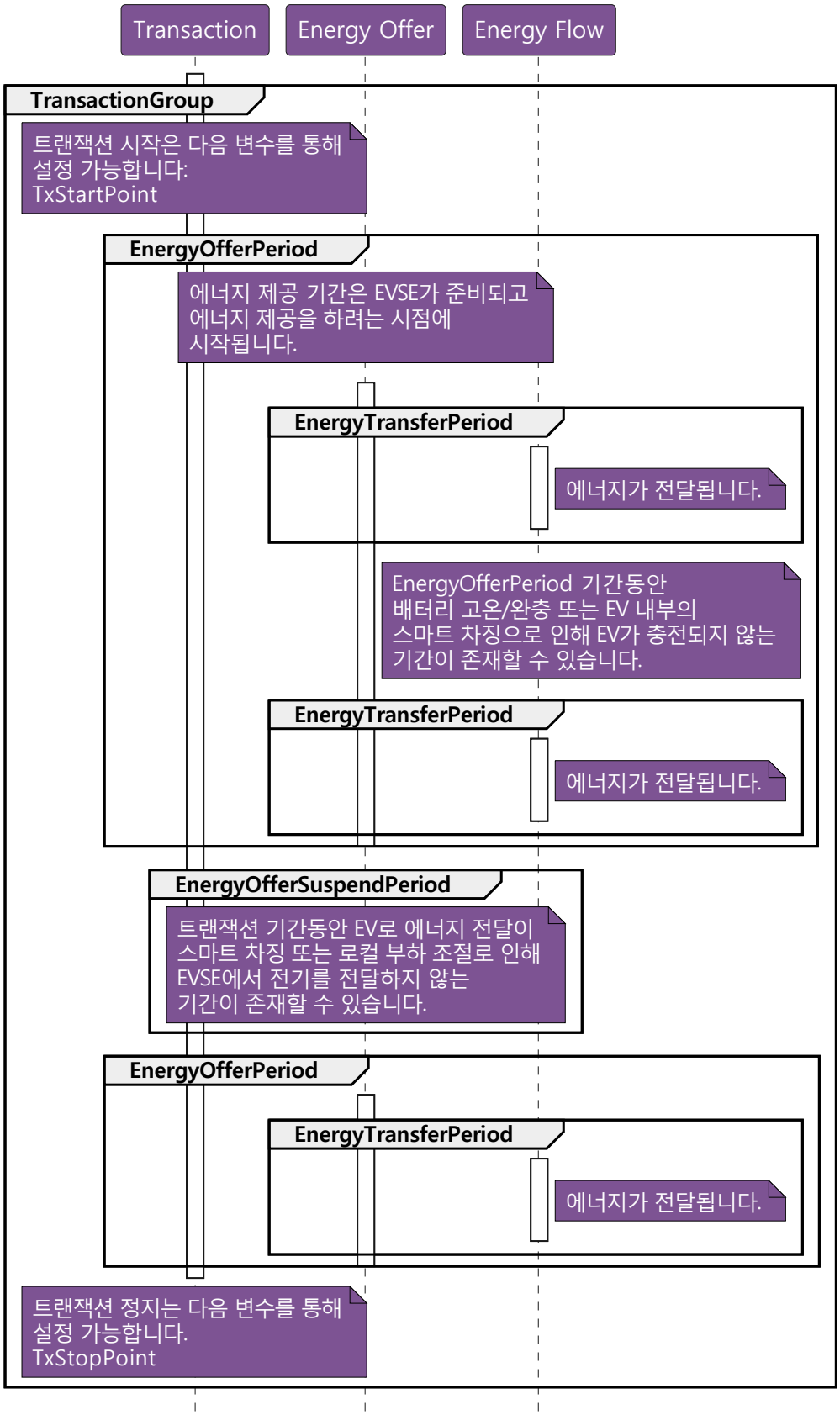

Figure 1. OCPP 충전 트랜잭션 정의

2.7. ISO 15118 support

이 섹션은 정보 제공을 목적으로 합니다.

이 버전의 OCPP는 ISO 15118 인증(일명 "플러그 앤 차지")과 ISO 15118 기반 스마트 충전을 지원합니다. ([ISO15118-2] 참조) 또한 ISO 15118 인증서를 설치하고 업데이트하는 방법을 설명합니다. 이 3가지 기능은 하나의 기능 블록으로 포함되지 않지만 사양 전체에 걸쳐 여러 장에 포함되어 있습니다. ISO 15118 인증은 기능 블록 Authorization에 포함되고 ISO 15118의 스마트 충전 사용 사례는 Smart Charging 장에 포함됩니다. 인증서 처리에 대한 설명은 별도의 기능 블록에 나와 있습니다.

15118 구현자는 15118에서 시행하는 시간 제한 제약을 알고 있어야 합니다. [ISO15118-1] (페이지: 127, 표: 109)
참고로, 15118 버전 1의 현재 시간 제한은 다음과 같습니다.

Table 9. ISO 15118 시간 제한

시간 제한	기본값
Sequence Timeouts	60초
Sequence Performance Timeouts	40초
PaymentDetailsReq/Res	5초
CertificateUpdateReq/Res	5초
CertificateInstallationReq/Res	5초

3. Generic Requirements

이 섹션은 규범적입니다.

일반 요구 사항은 기능 블록에 설명된 사용 사례 요소를 정의하기 위한 기반을 구축합니다.

Table 10. 일반 요구 사항

ID	전제 조건	요구 사항 정의	참고
FR.01		<message>요청의 발신자는 다른 요청 메시지를 보내기 전에 <message>응답 또는 시간 초과를 기다려야 합니다.	
FR.02	충전기가 JSON 스키마/RPC 프레임워크에 따라 유효한 OCPP 요청 메시지를 수신하고 다른 시스템이 보안 위반을 일으키지 않는 경우	충전기는 RPC 프레임워크인 CALLRESULT로 응답해야 합니다.	충전기/CSMS가 추가 정보를 제공해야 하는 경우 응답 메시지의 *statusInfo* 요소에서 이를 수행할 수 있습니다.
FR.03	충전기/CSMS가 JSON 스키마/RPC 프레임워크에 따라 잘못된 OCPP 메시지를 수신하거나 다른 시스템이 보안 위반을 발생시키는 경우	충전기/CSMS는 RPC 프레임워크 CALLERROR로 응답해야 합니다.	
FR.04	CSMS가 충전기에서 BootNotificationRequest를 수락하지 않고 충전기가 BootNotificationRequest가 아닌 다른 메시지를 보내는 경우	CSMS는 RPC 프레임워크인 CALLERROR: SecurityError로 응답해야 합니다.	
FR.05	응답 메시지에 결과를 제공하지 않지만 결과가 포함된 하나 이상의 메시지를 보내는 몇 가지 메시지가 있습니다. 다음 메시지 중 하나를 수신하는 경우; GetReport, GetBaseReport, GetMonitoringReport, GetDisplayMessages, CustomerInformation, GetChargingProfiles, GetLog, UpdateFirmware, PublishFirmware, TriggerMessage(<메시지>)	충전기는 아래 목록의 요청을 요청과 동일한 *requestId* 를 사용하여 응답 메시지(화살표 "→" 뒤에 표시됨)로 확인해야 합니다. GetReport → NotifyReport GetBaseReport → NotifyReport GetMonitoringReport → NotifyMonitoringReport GetDisplayMessages → NotifyDisplayMessage CustomerInformation → NotifyCustomerInformation GetChargingProfiles → ReportChargingProfiles GetLog → LogStatusNotification UpdateFirmware → FirmwareStatusNotification PublishFirmware → PublishFirmwareStatusNotification TriggerMessage(<message>) → <requested message>	CSMS는 *requestId = X* 에 대한 요청이 수락되었음을 알아야 하므로 이 *requestId* 에 대한 결과 메시지를 기대할 수 있습니다. TriggerMessage에는 *requestId* 가 없지만, TriggerMessageResponse는 요청된 메시지를 보내기 전에 보내야 한다는 의미에서 여전히 요구 사항이 적용됩니다.

3.1. Time Format Requirements

이 섹션은 규범적입니다.

CSMS와 충전소 간에 교환되는 모든 시간 값은 RFC-3339 [RFC3339]에 정의된 대로 형식화되어야 합니다.
또한 소수점 이하 초에는 추가 제한이 주어졌습니다. 소수점 이하 자릿수는 최대 3을 초과해서는 안 됩니다. 그러나 소수점 이하 초는 사용이 제한적이고 이를 생략하면 데이터 사용량이 감소하므로 완전히 생략하는 것이 좋습니다.

CSMS와 충전소 간의 모든 시간 값은 RFC-3339 [RFC3339]에 지정된 대로 시간대 지정자 'Z'를 사용하여 UTC로 교환하는 것이 좋습니다. 이렇게 하면 CSMS와 충전소 간의 상호 운용성이 향상됩니다.

3.1.1. Displaying local time

충전기에서 CSO에 내부 시계 구성에 대한 세부적인 제어권을 주고 싶을 때 다음 구성 변수 중 하나 이상을 구현할 수 있습니다. `TimeSource`, `TimeZone`, `TimeOffset`, `NtpSource`, `NtpServerUri`.

3.1.2. Daylight Saving Time

충전기가 "표준 시간"과 "일광 절약 시간(서머타임)" 기간 간의 정시에 자동으로 반기별 전환을 지원할 수 있는 방법은 두 가지가 있습니다.

- 전환 날짜와 오프셋은 구성된 `TimeZone`을 기반으로 충전기에서 알 수 있습니다.
- 전환 날짜와 오프셋은 `NextTimeOffsetTransitionDateTime` 및 `TimeOffsetNextTransition`을 통해 모든 전환에 대해 수동으로 구성됩니다.

일광 절약 시간은 EV 운전자에게 현재 시간을 표시하는 데 사용됩니다.

3.2. Message Timeouts

이 섹션은 규범적입니다.

OCPP는 메시지에 대한 타이밍 요구 사항을 지정하지 않습니다. 메시지 타이밍은 사용되는 기본 네트워크에 따라 크게 영향을 받습니다. GPRS 네트워크는 유선 전화와 다른 타이밍 특성을 가지고 있습니다. OCPP는 특정 유형의 네트워크를 요구하지 않지만 CSO가 선택할 수 있도록 열어두므로 OCPP는 타이밍 제약을 요구할 수 없습니다.

지침을 찾고 있다면 메시지 요청에 30초 시간 초과로 시작하여 사용되는 네트워크에 맞게 조정하세요.

충전기의 메시지 시간 초과 설정은 NetworkConnectionProfile의 messageTimeout 필드에서 구성할 수 있습니다. messageTimeout의 목적은 요청 메시지를 전송되지 않은 것으로 간주하고 통신 오류나 소프트웨어 오류로 인해 메시지가 도착하지 않았을 때 다른 작업을 계속할 수 있도록 하는 것입니다. 트랜잭션 관련 이벤트의 경우, 사용 사례 E13 - Transaction-related message not accepted by CSMS은 이런 경우 재시도 절차를 설명합니다. 기능 블록 E의 섹션 Delivering transaction-related messages도 참조하세요.

충전기는 요청에 대한 시간 초과가 발생하거나 웹소켓 ping에 응답하지 않을 때 CSMS에 대한 연결이 제대로 작동하지 않는다는 것을 발견할 수 있습니다. 이러한 상황에서는 충전기가 연결을 끊은 다음 CSMS에 다시 연결하는 것이 좋습니다. 이렇게 하면 새 세션이 생성되고 다중 인스턴스 CSMS의 다른 엔드포인트에 연결되어 오류가 해결될 수 있습니다.

3.3. Language support

이 섹션은 정보 제공을 목적으로 합니다.

CSMS는 충전기에 EV 운전자가 선호하는 언어를 제공하여 충전기가 EV 운전자의 선호도에 따라 언어로 통신할 수 있도록 합니다.

디스플레이에 메시지를 표시하는 모든 충전기의 경우 최소한 "영어"로 이를 구현하는 것이 좋습니다. EV 운전자의 선호 언어(CSMS에서 제공)가 "영어"가 아니고 충전기에 구현된 다른 언어와 일치하지 않는 경우 대체 언어로 "영어"를 사용하는 것이 좋습니다.

A. Security

1. OCPP Security

이 기능 블록은 OCPP 프로토콜에 대한 보안 요구 사항을 설명합니다. 보안 부분은 OCPP의 미래 개발 및 표준화를 강화하고 성숙시키기 위해 개발되었습니다. 이는 LaQuSo [21]의 종단 간 보안 설계를 기반으로 합니다. 보안 요구 사항은 OCPP 사용자를 지원하기 위해 충전기 및 CSMS의 보안 조치에 포함됩니다.

1.1. Security Objectives

이 섹션은 정보제공을 목적으로 합니다.

OCPP 보안은 다음 보안 목표를 충족하도록 설계되었습니다.

1. CSMS와 충전기 간에 안전한 통신 채널을 생성할 수 있도록 합니다. 이 채널의 메시지 무결성과 기밀성은 강력한 암호화 조치로 보호해야 합니다.
2. 충전기와 CSMS 간에 상호 인증을 제공합니다. 양측은 통신하는 사람을 식별할 수 있어야 합니다.
3. 충전기가 펌웨어 이미지의 출처와 무결성을 확인할 수 있도록 하고, 해당 이미지에 대한 부인 방지를 가능하게 함으로써 안전한 펌웨어 업데이트 프로세스를 제공합니다.
4. 스마트 충전 시스템의 보안을 모니터링하기 쉽게 보안 이벤트 로깅을 허용합니다. 보안 관련 이벤트 목록과 해당 '중요도'는 부록에 제공됩니다.

1.2. Design Considerations

이 섹션은 정보 제공을 목적으로 합니다.

보안 기능 블록은 OCPP에서 취한 접근 방식에 맞게 설계되었습니다. 가능한 한 표준 웹 기술을 사용하여 사용 가능한 웹 라이브러리와 소프트웨어를 사용하여 비용 효율적인 구현을 허용합니다. 애플리케이션 계층 보안 조치는 포함되지 않습니다. 이러한 고려 사항에 따라 OCPP 보안은 X.509 인증서를 사용하는 TLS 및 공개 키 암호화를 기반으로 합니다. CSMS는 일반적으로 서버 역할을 하므로 충전기의 다른 사용자 또는 역할 기반 액세스 제어는 이 표준에 구현되지 않습니다. 이를 완화하려면 CSMS에 액세스 제어(Access control)를 구현하는 것이 좋습니다. 구현된 메커니즘을 우회할 수 없도록 하려면 충전기에서 로컬로 충전기에 대한 유지 관리를 수행하는 자격을 갖춘 인력이 OCPP를 사용해서는 안 됩니다. 로컬 유지 관리 목적으로 다른 프로토콜을 사용할 수 있기 때문입니다.

1.3. Security Profiles

이 섹션에서는 다양한 OCPP 보안 프로필과 해당 요구 사항을 정의합니다. OCPP 2.0.1은 세 가지 보안 프로필을 지원합니다. 아래 표는 프로필에서 사용하는 보안 조치를 보여줍니다.

Table 11. OCPP 보안 프로필 개요

프로필	충전기 인증	CSMS 인증	통신 보안
1. 기본 인증을 사용한 보안되지 않은 전송	HTTP 기본 인증	-	-
2. 기본 인증을 사용한 TLS	HTTP 기본 인증	인증서를 사용한 TLS 인증	전송 계층 보안(TLS)
3. 클라이언트 측 인증서가 있는 TLS	인증서를 사용한 TLS 인증	인증서를 사용한 TLS 인증	전송 계층 보안(TLS)

- Unsecured Transport with Basic Authentication Profile에는 CSMS에 대한 인증이나 보안 통신 채널을 설정하는 조치가 포함되지 않습니다. 따라서 신뢰할 수 있는 네트워크, 예를 들어 CSMS와 충전기 사이에 VPN이 있는 네트워크에서만 사용해야 합니다. 현장 작업의 경우 TLS가 있는 보안 프로필을 사용하는 것이 좋습니다.
- 어떤 경우(예: 실험실 설치, 테스트 설정 등)에는 보안을 구현하지 않고 OCPP 2.0.1을 사용하는 것을 선호할 수 있습니다. 이는 가능하지만 유효한 OCPP 2.0.1 구현으로 간주되지 않습니다.
- 충전기에 올바른 날짜와 시간이 설정되어 있지 않으면 서버 인증서를 검증할 수 없습니다. 이 문제를 해결하려면 NTP를 사용하거나, 모바일 네트워크를 사용하여 시간을 자동으로 설정하거나, 첫 번째 연결 전에 시간을 설정하는 설치 도구를 사용하는 것이 좋습니다.

1.3.1. Generic Security Profile requirements

Table 12. 일반 보안 프로필 요구 사항

ID	전제 조건	요구 사항 정의
A00.FR.001		충전기와 CSMS는 한 번에 하나의 보안 프로필만 사용해야 합니다.
A00.FR.002	충전기가 CSMS가 사용하는 것과 다른 프로필로 연결을 시도하는 경우	CSMS는 연결을 종료해야 합니다.
A00.FR.003	CSMS가 충전기가 사용하는 것과 다른 프로필로 연결을 시도하는 경우	충전기는 연결을 종료해야 합니다.
A00.FR.004		OCPP 통신이 가능하기 전에 보안 프로필을 구성해야 합니다.
A00.FR.005		사용하는 보안 프로필을 보안 수준이 낮은 프로필로 낮추는 것은 보안상의 이유로 OCPP 사양의 일부가 아니며 OCPP를 통해서가 아닌 다른 방법을 통해 수행해야 합니다. OCPP 메시지는 이를 위해 사용되어서는 안 됩니다(예: SetVariablesRequest 또는 DataTransferRequest).
A00.FR.006	CSMS가 다른 보안 프로필 또는 다른 버전의 OCPP를 사용하는 충전기와 통신하는 경우.	CSMS는 CSMS의 다른 주소 또는 포트를 통해 충전기를 운영할 수 있습니다. 예를 들어, CSMS 서버에는 기본 인증을 사용하는 TLS용 TCP 포트 하나와 클라이언트 측 인증서를 사용하는 TLS용 TCP 포트 하나가 있을 수 있습니다. 이 경우 CSMS 포트당 사용되는 보안 프로필이 하나뿐이며 이는 허용됩니다.

1.3.2. Unsecured Transport with Basic Authentication Profile - 1

Table 13. 보안 프로필 1 - 기본 인증을 사용한 보안되지 않은 전송

번호	유형	설명
1	**이름**	기본 인증을 사용한 보안되지 않은 전송
2	**프로필 번호**	1
3	**설명**	기본 인증을 사용한 보안되지 않은 전송 프로필은 낮은 수준의 보안을 제공합니다. 충전기 인증은 사용자 이름과 비밀번호를 통해 수행됩니다. 통신 채널을 보호하기 위한 조치는 포함되지 않습니다.
4	**충전기 인증**	충전기 인증에는 HTTP 기본 인증이 사용됩니다.
5	**CSMS 인증**	이 프로필에서 CSMS는 충전기에 자신을 인증하지 않습니다. 충전기는 연결하는 서버가 실제로 CSMS라고 신뢰해야 합니다.
6	**통신 보안**	프로필에 통신 보안 조치가 포함되지 않습니다.

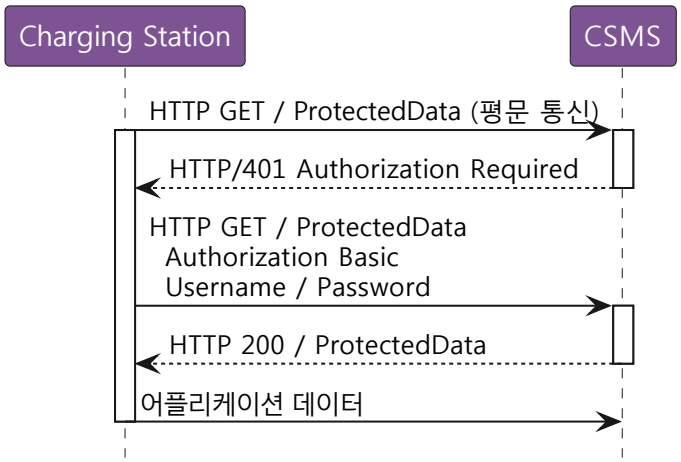

Figure 2. 시퀀스 다이어그램: HTTP 기본 인증 시퀀스 다이어그램

7	**참고**	OCPP 2.0.1(A00.FR.205)의 기본 인증 비밀번호 인코딩은 OCPP 1.6에서 수행된 방식과 다릅니다.

1.3.3. Unsecured Transport with Basic Authentication Profile - Requirements

Table 14. 보안 프로필 1 - 기본 인증을 사용한 보안되지 않은 전송 - 요구 사항

ID	전제 조건	요구 사항 정의
A00.FR.201		Unsecured Transport with Basic Authentication Profile은 신뢰할 수 있는 네트워크에서만 사용해야 합니다.
A00.FR.202		충전기는 HTTP 기본 인증을 사용하여 CSMS에 자체 인증을 해야 합니다. [18]
A00.FR.203	A00.FR.202	클라이언트, 즉 충전기는 모든 연결 요청에 사용자 이름과 비밀번호를 제공해야 합니다.

ID	전제 조건	요구 사항 정의
A00.FR.204	A00.FR.203	사용자 이름은 충전기 ID와 동일해야 하며, 이는 충전기가 OCPP-J 연결 URL에서 사용하는 식별 문자열입니다. 기본 인증을 사용하는 경우 충전기 ID에 ":" 문자가 포함될 수 없습니다. 그렇지 않으면 CSMS가 사용자 이름과 비밀번호를 분리하지 못할 수 있습니다.
A00.FR.205		비밀번호는 `BasicAuthPassword` 구성 변수에 저장해야 합니다. 충분히 높은 엔트로피를 갖는 임의로 선택된 passwordString이어야 하며, 최소 16자에서 최대 40자(영숫자 문자 및 passwordString에서 허용하는 특수 문자)로 구성되어야 합니다. 비밀번호는 UTF-8 인코딩된 문자열(옥텟 문자열 또는 base64로 인코딩되지 않음)로 전송해야 합니다.
A00.FR.206	A00.FR.203	HTTP Basic을 사용하면 사용자 이름과 비밀번호는 일반 텍스트로 전송되며 base64로만 인코딩됩니다. 따라서 이 메커니즘은 VPN과 같은 다른 수단으로 이미 보안된 연결에서만 사용하는 것이 좋습니다.
A00.FR.207	A00.FR.202	CSMS는 충전기 ID와 기본 인증 비밀번호가 연결 요청의 권한 헤더에 있는 사용자 이름과 비밀번호와 일치하는지 확인해야 합니다.

1.3.4. TLS with Basic Authentication Profile - 2

Table 15. 보안 프로필 2 - 기본 인증을 사용한 TLS

번호	유형	설명
1	**이름**	기본 인증을 사용한 TLS
2	**프로필 번호**	2
3	**설명**	기본 인증을 사용한 TLS 프로필에서 통신 채널은 TLS(전송 계층 보안)를 사용하여 보안됩니다. CSMS는 TLS 서버 인증서를 사용하여 자체 인증합니다. 충전기는 HTTP 기본 인증을 사용하여 자체 인증합니다.
4	**충전기 인증**	충전기 인증에는 HTTP 기본 인증이 사용됩니다. 이 프로필에서 TLS를 사용하므로 비밀번호가 암호화되어 전송되어 이 인증 방법을 사용할 때의 위험이 줄어듭니다.
5	**CSMS 인증**	충전기는 TLS 서버 인증서를 통해 CSMS를 인증합니다.
6	**통신 보안**	충전기와 CSMS 간의 통신은 TLS를 사용하여 보안됩니다.

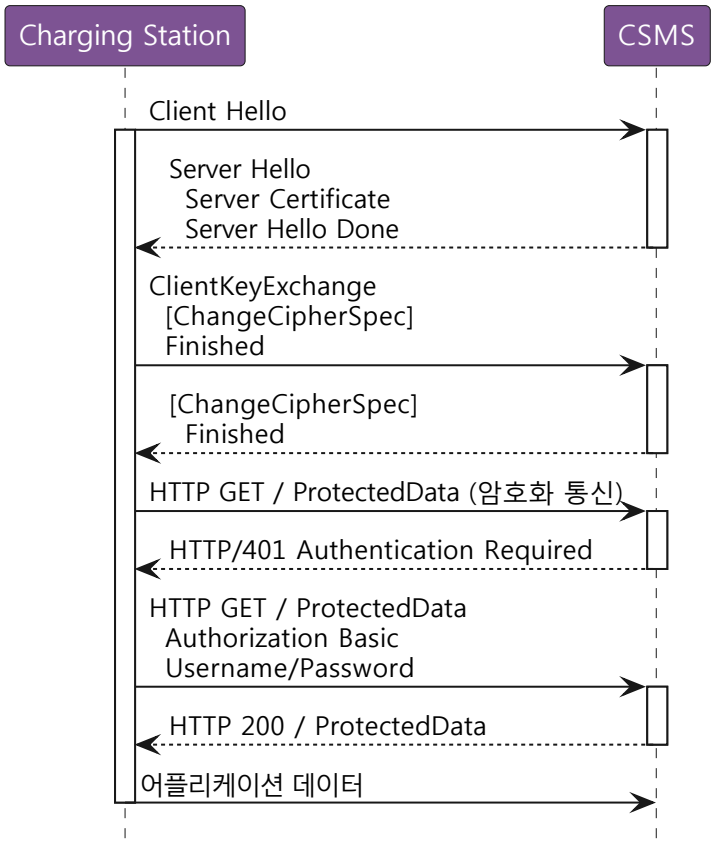

Figure 3. 시퀀스 다이어그램: 기본 인증이 있는 TLS 시퀀스 다이어그램

7	참고	TLS는 여러 구성을 허용하지만, 모든 구성이 충분한 보안을 제공하는 것은 아닙니다. 아래 요구 사항은 OCPP에 허용되는 구성을 설명합니다. 충전기는 RFC 6455에 설명된 대로 http 연결을 웹소켓 연결로 업그레이드하도록 요청하는 동안 기본 인증 RFC 2617에서 사용된 것과 동일한 헤더를 포함해야 합니다. 서버는 연결을 업그레이드하기 전에 먼저 인증 헤더를 검증해야 합니다. **예:** *GET /ws HTTP/1.1* *(Remote-Addr: 127.0.0.1)* *UPGRADE: websocket* *CONNECTION: Upgrade* *HOST: 127.0.0.1:9999* *ORIGIN: http://127.0.0.1:9999* *SEC-WEBSOCKET-KEY: Pb4obWo2214EfaPQuazMjA==* *SEC-WEBSOCKET-VERSION: 13* *AUTHORIZATION: Basic <Base64로 인코딩됨(<ChargePointId>:<AuthorizationKey>)>* OCPP 2.0.1(A00.FR.304)의 기본 인증 비밀번호 인코딩은 OCPP 1.6에서 수행된 방식과 다릅니다.

1.3.5. TLS with Basic Authentication Profile - Requirements

Table 16. 보안 프로필 2 - 기본 인증을 사용한 TLS - 요구 사항

ID	전제 조건	요구 사항 정의
A00.FR.301		충전기는 HTTP 기본 인증을 사용하여 CSMS에 자체 인증을 해야 합니다 [18]

ID	전제 조건	요구 사항 정의
A00.FR.302	A00.FR.301	클라이언트, 즉 충전기는 모든 연결 요청에 사용자 이름과 비밀번호를 제공해야 합니다.
A00.FR.303	A00.FR.302	사용자 이름은 충전기 ID와 동일해야 합니다. 충전기 ID는 OCPP-J 연결 URL에서 사용하는 충전기의 식별 문자열입니다. 기본 인증을 사용할 때 충전기 ID에 ":" 문자가 포함되어서는 안 됩니다. 그렇지 않으면 CSMS가 사용자 이름과 비밀번호를 분리하지 못할 수 있습니다.
A00.FR.304	A00.FR.302	비밀번호는 `BasicAuthPassword` 구성 변수에 저장해야 합니다. 최소 16자에서 최대 40자(영숫자 문자 및 passwordString에서 허용하는 특수 문자)로 구성된 충분히 높은 엔트로피를 갖는 임의로 선택된 passwordString이어야 합니다. 비밀번호는 UTF-8 인코딩된 문자열로 전송해야 합니다(옥텟 문자열이나 base64로 인코딩되지 않음).
A00.FR.306		CSMS는 TLS 서버 역할을 해야 합니다.
A00.FR.307		CSMS는 CSMS 인증서를 서버 측 인증서로 사용하여 자체를 인증해야 합니다.
A00.FR.308		충전기는 [3]의 Section 6에 설정된 경로 검증 규칙에 따라 CSMS 인증서의 인증 경로를 검증해야 합니다.
A00.FR.309		충전기는 commonName에 CSMS의 FQDN이 포함되어 있는지 검증해야 합니다.
A00.FR.310	CSMS가 유효한 인증서를 소유하지 않거나 인증 경로가 유효하지 않은 경우	충전기는 InvalidCsmsCertificate 보안 이벤트를 트리거해야 합니다(보안 이벤트의 전체 목록은 Part 2 - Appendices 참조).
A00.FR.311	A00.FR.310	충전기는 연결을 종료해야 합니다.
A00.FR.312		통신 채널은 TLS(전송 계층 보안) [4]를 사용하여 보호해야 합니다.
A00.FR.313		충전기와 CSMS는 TLS v1.2 이상만 사용해야 합니다.
A00.FR.314		두 엔드포인트 모두 사용된 TLS 버전을 확인해야 합니다.
A00.FR.315	A00.FR.314 AND CSMS가 충전기에서 이전 버전의 TLS를 사용하는 연결만 허용하거나 SSL만 허용한다는 것을 감지합니다.	CSMS는 연결을 종료해야 합니다.
A00.FR.316	A00.FR.314 AND 충전기에서 CSMS가 이전 버전의 TLS를 사용하는 연결만 허용하거나 SSL만 허용한다는 것을 감지합니다.	충전기는 InvalidTLSVersion 보안 이벤트를 트리거하고 연결을 종료해야 합니다(보안 이벤트의 전체 목록은 Part 2 - Appendices 참조). 참고: 이는 성공적인 연결이 이루어지면 큐에 넣고 CSMS로 보내야 하는 중요한 보안 이벤트입니다(사용 사례 A04에서 설명). 보안 이벤트는 오프라인 큐의 오버플로우를 방지하기 위해 반복적으로 실패한 연결 시도에 대해 한 번만 보내면 됩니다.
A00.FR.317		TLS는 수정 없이 [4] 또는 후속 표준과 같이 구현되어야 합니다.

ID	전제 조건	요구 사항 정의
A00.FR.318		CSMS는 최소한 다음 네 가지 암호화 제품군을 지원해야 합니다. **TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256** **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384** **TLS_RSA_WITH_AES_128_GCM_SHA256** **TLS_RSA_WITH_AES_256_GCM_SHA384** 참고: CSMS는 두 암호화 제품군을 모두 지원하기 위해 두 가지 다른 인증서를 제공해야 합니다. 보안 프로필 3을 사용하는 경우에도 CSMS는 두 암호 제품군에 대한 클라이언트 측 인증서를 생성할 수 있어야 합니다.
A00.FR.319		충전기는 최소한 다음 암호 제품군을 지원해야 합니다. (**TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256** 및 **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384**) 또는 (**TLS_RSA_WITH_AES_128_GCM_SHA256** 및 **TLS_RSA_WITH_AES_256_GCM_SHA384**) 참고 1: TLS_RSA는 순방향 비밀성(forward secrecy)을 지원하지 않으므로 TLS_ECDHE를 권장합니다. 또한 충전기에서 보안되지 않은 알고리즘을 사용한 것을 감지하면 InvalidTLSCipherSuite 보안 이벤트를 트리거해야 합니다(보안 이벤트의 전체 목록은 Part 2 - Appendices 참조). 참고 2: ISO15118-2는 EV와 충전기 간 통신을 위해 다음 암호 제품군을 구현하도록 규정합니다. TLS_ECDH_ECDSA_WITH_AES_128_**CBC***SHA256*, *TLS_ECDHE_ECDSA_WITH_AES_128***CBC**_SHA256
A00.FR.320		충전기와 CSMS는 [1]에서 레거시 사용에 적합하지 않은 것으로 표시된 암호화 기본을 사용하는 암호 제품군을 사용해서는 안 됩니다. 이는 이 사양에 설명된 암호화 제품군 중 하나(또는 그 이상)가 레거시 사용에 적합하지 않다고 표시되면 더 이상 사용해서는 안 됨을 의미합니다.
A00.FR.321		TLS 서버와 클라이언트는 압축 사이드 채널 공격을 방지하고 [10]의 Section 6에 설명된 대로 상호 운용성을 보장하기 위해 TLS 압축 방법을 사용해서는 안 됩니다.
A00.FR.322	A00.FR.320 AND CSMS가 충전기에서 이러한 제품군 중 하나만 사용하는 연결만 허용한다는 것을 감지합니다.	CSMS는 연결을 종료해야 합니다.
A00.FR.323	A00.FR.320 AND 충전기는 CSMS가 이러한 제품군 중 하나만을 사용하여 연결을 허용한다는 것을 감지합니다.	충전기는 InvalidTLSCipherSuite 보안 이벤트를 트리거하고 연결을 종료해야 합니다(보안 이벤트의 전체 목록은 Part 2 - Appendices 참조).
A00.FR.324	A00.FR.302	CSMS는 충전기 ID와 기본 인증 암호가 연결 요청의 권한 헤더에 있는 사용자 이름 및 암호와 일치하는지 확인해야 합니다.

1.3.6. TLS with Client Side Certificates Profile - 3

Table 17. 보안 프로필 3 - 클라이언트 측 인증서가 있는 TLS

번호	유형	설명
1	**이름**	클라이언트 측 인증서가 있는 TLS
2	**프로필 번호**	3
3	**설명**	클라이언트 측 인증서가 있는 TLS 프로필에서 통신 채널은 TLS(전송 계층 보안)를 사용하여 보안됩니다. 충전기와 CSMS는 모두 인증서를 사용하여 자체 인증합니다.
4	**충전기 인증**	CSMS는 TLS 클라이언트 인증서를 통해 충전기를 인증합니다.
5	**CSMS 인증**	충전기는 TLS 서버 인증서를 통해 CSMS를 인증합니다.
6	**통신 보안**	충전기와 CSMS 간의 통신은 TLS를 사용하여 보안됩니다.

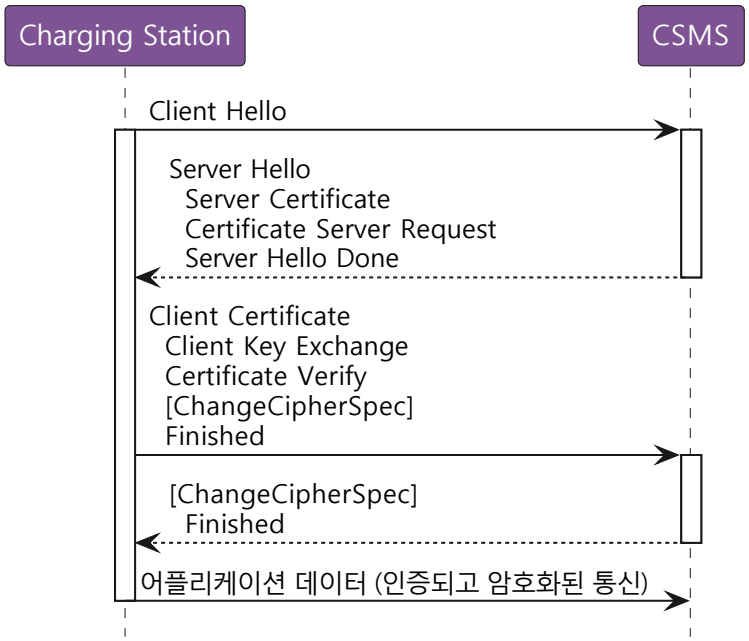

Figure 4. 시퀀스 다이어그램: 클라이언트 측 인증서가 있는 TLS

7	**비고**	N/a

1.3.7. TLS with Client Side Certificates Profile - Requirements

Table 18. 보안 프로필 3 - 클라이언트 측 인증서가 있는 TLS - 요구 사항

ID	전제 조건	요구 사항 정의
A00.FR.401		충전기는 충전기 인증서를 사용하여 CSMS에 자체 인증을 해야 합니다.
A00.FR.402		충전기 인증서는 TLS 클라이언트 측 인증서로 사용해야 합니다.
A00.FR.403		CSMS는 [3]의 Section 6에 설정된 경로 검증 규칙에 따라 충전기 인증서의 인증 경로를 확인해야 합니다.

ID	전제 조건	요구 사항 정의
A00.FR.404		CSMS는 인증서의 제목 필드에 있는 O(organizationName) RDN에 CSO 이름이 포함되어 있는지 확인하여 인증서가 CSO(또는 CSO가 신뢰하는 조직)의 소유인지 확인해야 합니다.
A00.FR.405		CSMS는 인증서의 제목 필드에 있는 CN(commonName) RDN에 충전기의 고유한 일련 번호가 포함되어 있는지 확인하여 인증서가 이 충전기에 속하는지 확인해야 합니다(Certificate Properties 참조).
A00.FR.406	충전기 인증서가 CSO 소유가 아닌 경우(예: 설치 직후)	충전기와 통신을 계속하기 전에 인증서를 업데이트하는 것이 좋습니다(Installation 참조)
A00.FR.407	A00.FR.429가 아닌 경우 및 충전기가 유효한 인증서를 소유하지 않거나 인증 경로가 잘못된 경우	CSMS는 연결을 종료해야 합니다.
A00.FR.408	A00.FR.407 또는 A00.FR.429	CSMS에서 보안 이벤트 InvalidChargingStationCertificate를 기록하는 것이 좋습니다.
A00.FR.409		CSMS는 TLS 서버 역할을 해야 합니다.
A00.FR.410		CSMS는 CSMS 인증서를 서버 측 인증서로 사용하여 자체 인증을 해야 합니다.
A00.FR.411		충전기는 [3]의 Section 6에 설정된 경로 검증 규칙에 따라 CSMS 인증서의 인증 경로를 확인해야 합니다.
A00.FR.412		충전기는 commonName이 CSMS의 FQDN과 일치하는지 확인해야 합니다.
A00.FR.413	CSMS가 유효한 인증서를 소유하지 않거나 인증 경로가 유효하지 않은 경우	충전기는 InvalidCsmsCertificate 보안 이벤트를 트리거해야 합니다(보안 이벤트의 전체 목록은 Part 2 - Appendices 참조).
A00.FR.414	A00.FR.413	충전기는 연결을 종료해야 합니다.
A00.FR.415		통신 채널은 TLS(전송 계층 보안) [4]를 사용하여 보안해야 합니다.
A00.FR.416		충전기와 CSMS는 TLS v1.2 이상만 사용해야 합니다.
A00.FR.417		이 두 엔드포인트는 사용된 TLS 버전을 확인해야 합니다.
A00.FR.418	A00.FR.417 AND CSMS가 충전기에서 이전 버전의 TLS를 사용하는 연결만 허용하거나 SSL만 허용한다는 것을 감지합니다.	CSMS는 연결을 종료해야 합니다.
A00.FR.419	A00.FR.417 AND 충전기에서 CSMS가 이전 버전의 TLS를 사용하는 연결만 허용하거나 SSL만 허용한다는 것을 감지합니다.	충전기는 InvalidTLSVersion 보안 이벤트를 트리거하고 연결을 종료해야 합니다(전체 보안 이벤트 목록은 Part 2 - Appendices 참조). 참고: 이것은 연결이 이루어지면 큐에 넣고 CSMS로 보내야 하는 중요한 보안 이벤트입니다(사용 사례 A04에서 설명). 보안 이벤트는 오프라인 큐의 오버플로우를 방지하기 위해 반복적으로 실패한 연결 시도에 대해 한 번만 보내면 됩니다. A00.FR.420 TLS는 수정 없이 [4] 또는 후속 표준과 같이 구현해야 합니다.

ID	전제 조건	요구 사항 정의
A00.FR.421		CSMS는 최소한 다음 네 가지 암호화 제품군을 지원해야 합니다. **TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256** **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384** **TLS_RSA_WITH_AES_128_GCM_SHA256** **TLS_RSA_WITH_AES_256_GCM_SHA384** 참고: CSMS는 두 암호화 제품군을 모두 지원하기 위해 두 가지 다른 인증서를 제공해야 합니다. 또한 보안 프로필 3을 사용하는 경우 CSMS는 두 암호화 제품군에 대한 클라이언트 측 인증서를 생성할 수 있어야 합니다.
A00.FR.422		충전기는 최소한 다음 암호 제품군을 지원해야 합니다. (**TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256** 및 **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384**) 또는 (**TLS_RSA_WITH_AES_128_GCM_SHA256** 및 **TLS_RSA_WITH_AES_256_GCM_SHA384**) 참고 1: TLS_RSA는 순방향 비밀성을 지원하지 않으므로 TLS_ECDHE를 권장합니다. 또한 충전기에서 보안되지 않은 알고리즘을 사용한 것을 감지하면 InvalidTLSCipherSuite 보안 이벤트를 트리거해야 합니다(보안 이벤트의 전체 목록은 Part 2 - Appendices 참조). 참고 2: ISO15118-2는 EV와 충전기 간 통신을 위해 다음 암호 제품군을 구현하도록 규정합니다. TLS_ECDH_ECDSA_WITH_AES_128_**CBC***SHA256*, *TLS_ECDHE_ECDSA_WITH_AES_128***CBC**_SHA256
A00.FR.423		충전기와 CSMS는 [1]에서 레거시 사용에 적합하지 않은 것으로 표시된 암호화 기본을 사용하는 암호 제품군을 사용해서는 안 됩니다. 이는 이 사양에 설명된 암호화 제품군 중 하나(또는 그 이상)가 레거시 사용에 적합하지 않다고 표시되면 더 이상 사용해서는 안 됨을 의미합니다.
A00.FR.424		TLS 서버와 클라이언트는 압축 사이드 채널 공격을 방지하고 [10]의 Section 6에 설명된 대로 상호 운용성을 보장하기 위해 TLS 압축 방법을 사용해서는 안 됩니다.
A00.FR.425	A00.FR.424 AND CSMS가 충전기에서 이러한 제품군 중 하나만을 사용하는 연결만 허용한다고 감지하는 경우	CSMS는 연결을 종료해야 합니다.
A00.FR.426	A00.FR.424 AND 충전기에서 CSMS가 이러한 제품군 중 하나만을 사용하는 연결을 허용한다는 것을 감지합니다.	충전기는 InvalidTLSCipherSuite 보안 이벤트를 트리거하고 연결을 종료해야 합니다(보안 이벤트의 전체 목록은 Part 2 - Appendices 참조).
A00.FR.427		각 충전기에 고유한 충전기 인증서를 사용해야 합니다.
A00.FR.428		충전기 인증서는 ISO15118-2의 SECC 인증서와 동일한 인증서일 수 있으며, 충전기와 전기 자동차 간의 TLS 연결을 설정하는 데 사용됩니다.

ID	전제 조건	요구 사항 정의
A00.FR.429	충전기 인증서가 만료된 경우 AND CSMS가 만료된 인증서가 있는 이 특정 충전기의 연결을 허용하도록 명시적으로 구성되었습니다.	CSMS는 BootNotification - Pending 상태에서 이 충전기를 수락할 수 있습니다(사용 사례 B02). 그런 다음 즉시 A02 - Update Charging Station Certificate by request of CSMS를 실행하여 인증서를 갱신해야 합니다.

1.4. Keys used in OCPP

이 섹션은 규범적입니다.

OCPP는 보안을 위해 여러 개의 공개 개인 키 쌍을 사용합니다. 아래 표를 참조하세요. 충전기의 키를 관리하기 위해 OCPP에 메시지가 추가되었습니다. CSMS 또는 제조업체에서 키를 업데이트하는 것은 OCPP의 범위를 벗어납니다. 클라이언트 측 인증서가 있는 TLS를 사용하는 경우 충전기에는 CSMS에 대한 인증을 위해 "충전기 인증서"가 필요합니다.

Table 19. OCPP 보안 사양에 사용된 인증서

인증서	개인 키 저장 위치	설명
CSMS 인증서	CSMS	CSMS를 인증하는 데 사용되는 키입니다.
충전기 인증서	충전기	충전기를 인증하는 데 사용되는 키입니다.
펌웨어 서명 인증서	제조업체	펌웨어 서명을 확인하는 데 사용되는 키입니다.
SECC 인증서	충전기	ISO15118-2에서 충전기와 전기 자동차 간의 TLS 연결을 설정하는 데 사용하는 인증서입니다.

1.4.1. Certificate Properties

이 섹션은 규범적입니다.

Table 20. 인증서 속성 요구 사항

ID	전제 조건	요구 사항 정의
A00.FR.501		모든 인증서는 [17]의 Section 5.6.1에 따라 최소 112비트의 대칭 키와 동등한 보안을 제공하는 개인 키를 사용해야 합니다. 이는 NIST가 2011-2030년 기간에 권장하는 키 크기입니다.
A00.FR.502	A00.FR.501 AND RSA 또는 DSA	이는 최소 2048비트 길이의 키로 변환됩니다.
A00.FR.503	A00.FR.501 AND 타원 곡선 암호화	이것은 최소 224비트 길이의 키로 변환됩니다.
A00.FR.504		모든 암호화 작업의 경우, 향후 시스템에서 사용하기에 적합한 [12]에서 BSI가 권장하는 알고리즘만 사용해야 합니다. 이 제한에는 인증서 계층 구조의 인증서 서명이 포함됩니다.
A00.FR.505		인증 기관의 서명에는 RSA-PSS 또는 ECDSA를 사용해야 합니다.
A00.FR.506		해시 값을 계산하려면 SHA256 알고리즘을 사용해야 합니다.

ID	전제 조건	요구 사항 정의
A00.FR.507		인증서는 Privacy-Enhanced Mail(PEM) 형식으로 인코딩된 X.509 형식으로 저장 및 전송되어야 합니다.
A00.FR.508		모든 인증서에는 일련 번호가 포함되어야 합니다.
A00.FR.509		인증서의 제목 필드에는 O(organizationName) RDN에 인증서 소유자의 조직 이름이 포함되어야 합니다.
A00.FR.510		CSMS 인증서의 경우 제목 필드에는 CN(commonName) RDN에 서버 엔드포인트의 FQDN이 포함되어야 합니다.
A00.FR.511		충전기 인증서의 경우 제목 필드에는 충전기의 고유한 일련 번호로 구성된 CN(commonName) RDN이 포함되어야 합니다. 이 일련 번호는 충전기 인증서와 CSMS 인증서를 구별할 수 있도록 URL 또는 IP 주소 형식이어서는 안 됩니다. 참고: RFC 2818에 따르면 dnsName 유형의 subjectAltName 확장자가 있는 경우 이를 ID로 사용해야 합니다. 이는 OCPP 및 ISO 15118와 호환되지 않습니다. 따라서 충전기 및 CSMS 인증서에서 사용하면 안 됩니다. CSMS에 여러 네트워크 경로(예: VPN에서 IP 주소를 사용하는 개인 APN + VPN을 통해, 명명된 URL을 사용하는 공용 인터넷을 통해)가 있는 경우 CSMS에 dnsName 유형의 subjectAltName 확장자를 사용할 수 있습니다.
A00.FR.512		모든 인증서에 대해 X.509 키 사용 확장 [19]을 사용하여 인증서 사용을 해당 작업에 제한해야 합니다.
A00.FR.513		충전기 인증서가 ISO 15118 프로토콜에서 SECC 인증서로도 사용되는 경우 인증서는 ISO15118-2의 요구 사항도 충족해야 합니다.
A00.FR.514		모든 인증서에 대해 ISO 15118 표준과 호환되도록 X.509 확장 키 사용 확장을 사용하지 않는 것이 좋습니다. 사용 가능한 대체 메커니즘이 있습니다.

1.4.2. Certificate Hierarchy

이 섹션은 규범적입니다.

OCPP 프로토콜은 두 개의 별도 인증서 계층 사용을 지원합니다.

1. CSMS 및 충전기 인증서를 포함하는 충전기 운영자 계층.
2. 펌웨어 서명 인증서를 포함하는 제조업체 계층.

CSMS는 InstallCertificateRequest 메시지를 사용하여 충전기에 저장된 CSO 루트 인증서를 업데이트할 수 있습니다.

Table 21. 인증서 계층 요구 사항

ID	전제 조건	요구 사항 정의
A00.FR.601		충전기 운영자는 충전기 운영자 계층에 대한 인증 기관 역할을 할 수 있습니다.

ID	전제 조건	요구 사항 정의
A00.FR.602	A00.FR.601	충전기 운영자는 예를 들어 ISO15118-2의 부록 E 및 F에 설명된 인증서 계층을 따르고 CSO Sub-CA 2 인증서를 사용하여 CSMS 및 충전기 인증서에 서명할 수 있습니다. 이를 통해 충전기 클라이언트 측 인증서의 온라인 검증을 충전기 운영자의 네트워크 내에서 수행할 수 있어 네트워크 아키텍처가 간소화될 수 있습니다.
A00.FR.603		CSO 루트 인증서에 속하는 개인 키는 반드시 잘 보호해야 합니다.
A00.FR.604		제조업체는 일반적으로 충전기 운영자와 별도의 조직이므로 신뢰할 수 있는 제3자를 인증 기관으로 사용해야 합니다. 이는 펌웨어 이미지의 부인 방지(non-repudiation)를 위해 필수적입니다.

1.5. Certificate Revocation

이 섹션은 규범적입니다.

어떤 경우에는 유효 기간이 만료되기 전에 인증서가 무효화될 수 있습니다. 이러한 사례에는 조직 이름 변경, 인증서 개인 키의 손상 또는 의심되는 손상이 포함됩니다. 이러한 경우 인증서를 해지하거나 더 이상 유효하지 않음을 표시해야 합니다. 인증서가 해지되었다고 해서 연결을 닫아야 하는 것은 아닙니다. 연결이 24시간 이상 열려 있을 수 있기 때문입니다.

인증서 해지에는 다양한 방법이 권장됩니다. 아래 표를 참조하세요.

Table 22. 다양한 인증서에 대한 권장 해지 방법.

인증서	폐지
CSMS 인증서	빠른 만료
충전기 인증서	온라인 검증
펌웨어 서명 인증서	온라인 검증

Table 23. 인증서 폐지 요구 사항

ID	전제 조건	요구 사항 정의
A00.FR.701		빠른 만료는 CSMS 인증서를 폐지하는 데 사용해야 합니다. (참고 1 참조)
A00.FR.702		CSMS는 온라인 인증서 검증을 사용하여 충전기 인증서의 유효성을 확인해야 합니다.
A00.FR.703		인증서를 관리하기 위해 별도의 인증 기관 서버를 사용하는 것이 좋습니다.
A00.FR.704	A00.FR.703	이 서버는 또한 어떤 인증서가 취소되었는지 추적해야 합니다.
A00.FR.705		CSMS는 인증 기관 서버로 인증서의 유효성을 확인해야 합니다. (참고 2 참조)
A00.FR.707		충전기에 펌웨어 검증을 위한 인증서를 제공하기 전에 CSMS는 인증서와 서명된 펌웨어 업데이트를 모두 검증해야 합니다.

참고 1: 빠른 만료의 경우 인증서는 24시간 미만의 짧은 기간 동안만 유효합니다. 그 후 서버는 인증 기관(CSO 자체일 수 있음)에 새

인증서를 요청해야 합니다(섹션 Certificate Hierarchy 참조). 이렇게 하면 충전기에서 해지 목록이나 온라인 인증서 검증을 구현할 필요가 없습니다. 이렇게 하면 충전기에서 인증서 관리를 구현하는 것이 간소화되고 충전기 측에서 통신 비용이 줄어듭니다. 빠른 만료를 요구함으로써 인증서가 손상되더라도 영향은 짧은 기간으로 줄어듭니다.

인증서 체인이 손상되면 공격자는 위조된 인증서를 사용하여 충전기를 속여 "가짜" CSMS에 연결하도록 할 수 있습니다. 빠른 만료를 사용하면 충전기가 취약해지는 시간이 크게 줄어듭니다.

충전기는 항상 CSMS를 통해 인증 기관과 통신하므로 충전기가 손상되더라도 충전기는 CA를 직접 공격할 수 없습니다.

참고 2: 이렇게 하면 충전기 인증서를 즉시 해지할 수 있습니다. 충전기 인증서 해지는 예를 들어 충전기가 제거될 때 발생합니다. 이 방법은 CSMS 인증서를 폐지하는 것보다 더 일반적입니다. CSMS 인증서 폐지는 일반적으로 인증서가 손상되었을 때만 수행됩니다.

1.5.1. Installation

이 섹션은 규범적입니다.

고유한 자격 증명은 각 충전기를 CSMS에 인증하는 데 사용해야 하며, 이는 HTTP 기본 인증에 사용되는 비밀번호(Charging Station Authentication 참조) 또는 충전기 인증서인지 여부와 무관합니다. 이러한 고유한 자격 증명은 제조 또는 설치 중 어느 시점에서 충전기에 넣어야 합니다.

Table 24. 인증서 설치 요구 사항

ID	전제 조건	요구 사항 정의
A00.FR.801		제조업체가 제조 중에 고유한 자격 증명으로 충전기를 초기화하는 것이 좋습니다.
A00.FR.802	A00.FR.801	자격 증명은 암호화 난수 생성기를 사용하여 생성되어야 하며, 보안 환경에 설치되어야 합니다.
A00.FR.803	A00.FR.801	보안 채널을 통해 CSO로 전송되어야 CSO가 CSMS에 가져올 수 있습니다.
A00.FR.804	충전기 인증서가 사용되는 경우.	제조업체는 자체 인증서를 사용하여 이에 서명할 수 있습니다.
A00.FR.805	A00.FR.804	CSO가 섹션 A01 - Update Charging Station Password for HTTP Basic Authentication 또는 A02 - Update Charging Station Certificate by request of CSMS에 설명된 방법을 사용하여 설치 후 즉시 자격 증명을 업데이트하는 것이 좋습니다.
A00.FR.806	'공장 자격 증명'이 업데이트되기 전에	CSMS는 충전기에서 사용할 수 있는 기능을 제한할 수 있습니다. CSMS는 이를 위해 BootNotification state: Pending를 사용할 수 있습니다. Pending 상태 동안 CSMS는 자격 증명을 업데이트할 수 있습니다.
A00.FR.807	A00.FR.804 AND 충전기 제조업체 인증서가 만료되었습니다.	CSMS는 BootNotification 이후 `Pending` 상태의 충전기에서 연결을 수락하고 즉시 사용 사례 A02 - Update Charging Station Certificate by request of CSMS를 실행하여 새로운 유효한 CSO 인증서를 설치할 수 있습니다.

2. Use cases & Requirements

A01 - Update Charging Station Password for HTTP Basic Authentication

Table 25. A01 - 비밀번호 관리

번호	유형	설명
1	**이름**	HTTP 기본 인증을 위한 충전기 비밀번호 업데이트
2	**ID**	A01
	기능 블록	A. 보안
3	**목표**	이 사용 사례는 기본 및 TLS에서 기본 인증 보안 프로필로 충전기를 인증하는 데 사용되는 비밀번호인 BasicAuthPassword를 사용하는 방법을 정의합니다.
4	**설명**	CSMS가 HTTP 기본 인증을 위한 새 비밀번호를 구성할 수 있도록 CSMS는 `BasicAuthPassword` 구성 변수에 대한 새 값을 보낼 수 있습니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS가 SetVariablesRequest(ComponentName=SecurityCtrlr, VariableName=BasicAuthPassword)를 충전기로 보냅니다. **2.** 충전기가 SetVariablesResponse와 상태 *Accepted* 로 응답합니다. **3.** 충전기가 현재 연결을 끊습니다. (대기 중인 메시지를 저장합니다.) **4.** 충전기가 새 비밀번호로 CSMS에 연결합니다.
5	**필수 조건**	보안 프로필: Basic Security Profile 또는 TLS with Basic Authentication 사용 중.
6	**사후 조건**	**성공 사후 조건:** 충전기가 새 비밀번호로 CSMS에 다시 연결되었습니다. **실패 사후 조건:** 충전기가 SetVariablesRequest에 *Accepted* 가 아닌 상태의 SetVariablesResponse로 응답하는 경우, 충전기는 이전 자격 증명을 계속 사용합니다. CSMS는 충전기의 부팅 알림을 수락하지 않는 등 충전기를 다르게 처리할 수 있습니다.

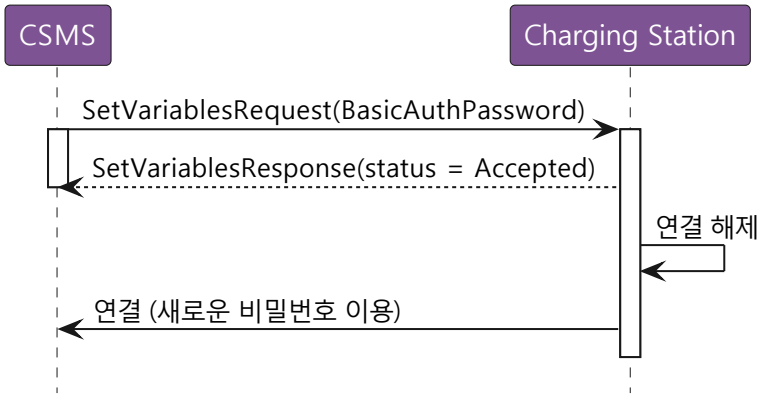

Figure 5. HTTP 기본 인증을 위한 충전기 비밀번호 업데이트(행복한 흐름)

7	**오류 처리**	n/a

8	**참고**	n/a

A01 - Update Charging Station Password for HTTP Basic Authentication - Requirements

Table 26. A01 - HTTP 기본 인증을 위한 충전기 비밀번호 업데이트 - 요구 사항

ID	전제 조건	요구 사항 정의
A01.FR.01		비밀번호는 구성 변수 `BasicAuthPassword`에 저장되어야 합니다.
A01.FR.02		OCPP를 통해 충전기의 기본 인증 비밀번호를 설정하려면 CSMS가 `BasicAuthPassword` 구성 변수와 함께 충전기에 SetVariablesRequest 메시지를 보내야 합니다.
A01.FR.03	A01.FR.02 AND 충전기는 이 SetVariablesRequest에 SetVariablesResponse를 상태 *Accepted* 로 응답합니다.	CSMS는 권한 부여 키 변경이 성공적이라고 가정하고 충전기에서 이전에 사용한 자격 증명을 더 이상 수락하지 않습니다.
A01.FR.04	A01.FR.02 AND 충전기는 이 SetVariablesRequest에 SetVariablesResponse를 상태 *Accepted* 가 아닌 다른 상태로 응답합니다.	CSMS는 충전기에서 비밀번호를 변경하지 않았다고 가정합니다. 따라서 CSMS는 이전 자격 증명을 계속 수락합니다.
A01.FR.05	A01.FR.04	CSMS는 여전히 충전기에서 연결을 수락해야 하지만 충전기에서 사용할 수 있는 기능을 제한할 수 있습니다. CSMS는 이를 위해 BootNotification state :Pending을 사용할 수 있습니다. Pending 상태에서 CSMS는 예를 들어 자격 증명을 업데이트하려고 다시 시도할 수 있습니다.
A01.FR.06		충전기마다 다른 비밀번호를 사용해야 합니다.
A01.FR.07		비밀번호는 충분한 엔트로피를 갖도록 무작위로 생성해야 합니다.
A01.FR.08		CSMS는 비밀번호 자체가 아닌 솔트 처리된 비밀번호 해시만 저장해야 합니다.
A01.FR.09		CSMS는 로그 파일이나 디버그 정보에 비밀번호를 일반 텍스트로 저장해서는 안 됩니다. 이런 식으로 CSMS가 손상되더라도 모든 충전기 비밀번호가 즉시 손상되는 것은 아닙니다.
A01.FR.10		충전기에서 비밀번호는 일반 텍스트로 저장해야 합니다. 안전하게 저장하는 데 각별히 주의해야 합니다. 그러나 자격 증명을 안전하게 저장하는 방법에 대한 메커니즘 정의는 OCPP 보안 프로필의 범위에 포함되지 않습니다.
A01.FR.11	A01.FR.02	충전기는 보안 로그에 `BasicAuthPassword`의 변경 사항을 기록해야 합니다.
A01.FR.12	A01.FR.11	충전기는 로깅에서 BasicAuthPassword의 내용을 공개해서는 안 됩니다. 이는 진단 파일에 액세스할 수 있는 사람에게 키 자료가 노출되는 것을 방지하기 위한 것입니다.

A02 - Update Charging Station Certificate by request of CSMS

Table 27. A02 - CSMS 요청에 따른 충전기 인증서 업데이트

번호	유형	설명
1	**이름**	CSMS 요청에 따른 충전기 인증서 업데이트
2	**ID**	A02
	기능 블록	A. 보안
3	**목표**	충전기 클라이언트 측 인증서 관리를 용이하게 하기 위해 인증서 업데이트 절차가 제공됩니다.
4	**설명**	CSMS는 충전기에 TriggerMessageRequest를 사용하여 키를 업데이트하도록 요청하고 *requestedMessage* 필드는 SignChargingStationCertificate(또는 별도의 15118 인증서의 경우 SignV2GCertificate)로 설정합니다. 충전기에 각 EVSE에 대한 별도의 ISO15118Ctrlr(ISO 15118의 SECC)이 있는 경우 CSMS는 각각에 대한 요청을 보내야 합니다. 충전기의 장치 모델은 ISO15118Ctrlr이 최상위 또는 EVSE 수준에 있는지 알려줍니다. 충전기에 각각 여러 EVSE를 제어하는 여러 SECC가 있는 경우 장치 모델에서 각 EVSE에 대한 ISO15118Ctrlr로 표현됩니다. 동일한 SECC에서 제어하는 EVSE는 동일한 "SeccId"를 사용하여 ISO15118Ctrlr을 보고합니다.
	Actors	충전기, CSMS, 인증 기관 서버
	시나리오 설명	*SignChargingStationCertificate* **1.** CSMS는 TriggerMessageRequest를 사용하여 충전기에 인증서를 업데이트하도록 요청하고 *requestedMessage* 필드는 SignChargingStationCertificate로 설정합니다. **2.** 충전기는 TriggerMessageResponse로 응답합니다. **3.** 충전기는 새로운 공개/비공개 키 쌍을 생성합니다. **4.** 충전기는 *certificateType* = `ChargingStationCertificate` 를 포함하는 SignCertificateRequest를 CSMS로 보냅니다. **5.** CSMS는 *Accepted* 상태의 SignCertificateResponse로 응답합니다. **6.** CSMS는 CSR을 인증 기관 서버로 전달합니다. **7.** 인증 기관 서버가 인증서에 서명합니다. **8.** 인증 기관 서버는 서명된 인증서를 CSMS로 반환합니다. **9.** CSMS는 CertificateSignedRequest를 충전기로 보냅니다. **10.** 충전기는 서명된 인증서를 확인합니다. **11.** 충전기는 CertificateSignedResponse로 CSMS에 *Accepted* 또는 *Rejected* 상태로 응답합니다.

번호	유형	설명
	대체 시나리오	*SignV2GCertificate* **1.** CSMS는 *componentVariable.component* "ISO15118Ctrlr" 및 *componentVariable.variable* "SeccId"에 대한 GetReportRequest를 보내 구성 요소 ISO15118Ctrlr에 대한 정보를 요청합니다. **2.** 반환된 **고유** SeccId마다: **2.1.** CSMS는 충전기에 TriggerMessageRequest를 사용하여 인증서를 업데이트하도록 요청합니다. 여기서 *requestedMessage* 필드는 15118 인증서의 경우 SignV2GCertificate로 설정되고 evse는 ISO15118Ctrlr의 EVSE로 설정됩니다. (ISO15118Ctrlr이 최상위에서 하나의 구성 요소로만 존재하는 경우 evse를 생략할 수 있습니다.) **2.2.** 충전기는 TriggerMessageResponse로 응답합니다. **2.3.** 충전기는 새로운 공개/비공개 키 쌍을 생성합니다. **2.4.** 충전기는 *certificateType* = `V2GCertificate` 와 CommonName(CN)이 SeccId 값으로 설정된 csr을 포함하는 SignCertificateRequest를 CSMS로 전송합니다. **2.5.** CSMS는 *Accepted* 상태의 SignCertificateResponse로 응답합니다. **2.6.** CSMS는 CSR을 인증 기관 서버로 전달합니다. **2.7.** 인증 기관 서버가 인증서에 서명합니다. **2.8.** 인증 기관 서버는 서명된 인증서를 CSMS로 반환합니다. **2.9.** CSMS는 CertificateSignedRequest를 충전기로 전송합니다. **2.10.** 충전기는 서명된 인증서를 확인합니다. **2.11.** 충전기는 CertificateSignedResponse로 CSMS에 *Accepted* 또는 *Rejected* 상태로 응답합니다.
5	**필수 조건**	표준 구성 변수 "OrganizationName"을 설정해야 합니다. SignV2GCertificate의 경우 변수 ISO15118Ctrlr.SeccId를 설정해야 합니다.
6	**사후 조건**	**성공 사후 조건:** 충전기에 새 클라이언트 측 인증서가 설치되었습니다. **실패 사후 조건:** 새 클라이언트 측 인증서가 거부되고 삭제됩니다.

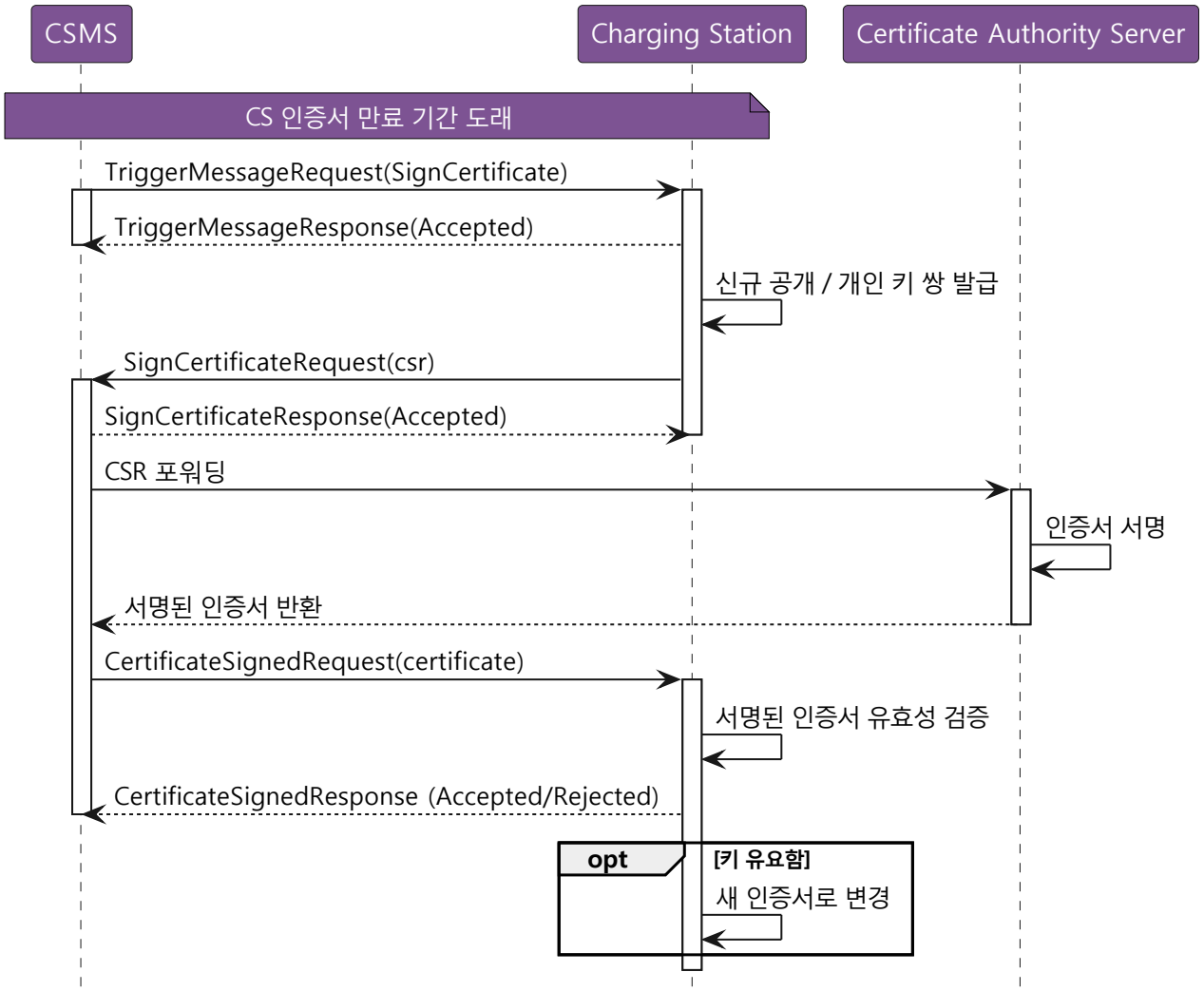

Figure 6. 충전기 인증서 업데이트

7	**오류 처리**	CSMS는 충전기에서 CSR 요청을 수락한 후 CA로 전달합니다. 하지만 CA에 도달할 수 없거나 CSR을 거부하면 충전기는 결코 알 수 없습니다. CSMS는 CSR에 대한 몇 가지 검사를 수행할 수 있지만 CA가 수행하는 모든 검사를 수행할 수 없으며 CA에 대한 연결 시간 초과를 방지하지 못합니다. CA에 따르면, 이와 같은 문제는 CA가 오프라인이거나 충전기에서 보낸 CSR이 올바르지 않을 때 발생합니다. 두 경우 모두 CSO의 운영자에게 알려야 할 사항입니다. 그런 다음 운영자는 문제를 조사해야 합니다. 문제가 해결되면 운영자는 A02를 다시 실행할 수 있습니다. 인증서가 X분 또는 시간 내에 전송되지 않을 때 충전기가 재시도하도록 하는 것은 권장되지 않습니다. CSR이 올바르지 않으면 자동으로 해결되지 않습니다. 새 펌웨어로만 이를 해결할 수 있습니다.

충전기 운영자는 충전기 운영자 계층에 대한 인증 기관 역할을 할 수 있습니다.

해당 인증 기관은 CSR의 정보를 확인해야 합니다.
올바르면 인증 기관은 CSR에 서명하고 CSO로 보내고 CSO는 CertificateSignedRequest 메시지로 충전기로 다시 보냅니다.
인증 기관은 인증서 서명 개인 키를 안전하게 유지하기 위한 강력한 조치를 구현해야 합니다.

8 참고

CertificateSignedRequest 메시지(사용 사례 A02 및 A03 참조)와 InstallCertificateRequest(사용 사례 M05 - Install CA Certificate in a Charging Station)는 모두 인증서를 보내는 데 사용되지만 그 용도는 다릅니다. CertificateSignedRequest는 충전기 자체의 공개 인증서와 인증 기관에서 서명한 V2G 인증서를 반환하는 데 사용됩니다. InstallCertificateRequest는 루트 인증서를 설치하는 데 사용됩니다.

V2G 인증서 처리에 대해서는 사용 사례 M03 - Retrieve list of available certificates from a Charging Station, M04 - Delete a specific certificate from a Charging Station 및 M06 - Get Charging Station Certificate status를 참조하세요.

A02 - Update Charging Station Certificate by request of CSMS - Requirements

Table 28. A02 - 요구 사항

ID	전제 조건	요구 사항 정의
A02.FR.01		충전기 설치 후 키 업데이트를 수행하여 제조업체에서 처음 제공한 키(아마도 기본 키)에서 키를 변경해야 합니다.
A02.FR.02	TriggerMessageResponse를 보낸 후.	충전기는 [16]의 Section 4.2.1.3에 설명된 키 생성 기능 중 하나를 사용하여 새 공개/비공개 키 쌍을 생성해야 합니다.
A02.FR.03	A02.FR.02	충전기는 RFC 2986 [22]에 설명된 대로 공개 키를 인증서 서명 요청(CSR) 형태로 보낸 다음 SignCertificateRequest 메시지를 사용하여 PEM 인코딩해야 합니다.
A02.FR.04		CSMS는 인증서 자체에 서명해서는 안 되지만, 대신 CSR을 충전기 인프라의 인증서를 관리하는 전담 인증 기관 서버로 전달해야 합니다. 전담 기관 서버는 CSO에서 운영할 수 있습니다.
A02.FR.05		키 업데이트 프로세스 중에 충전기에서 생성한 개인 키는 언제든지 충전기를 떠나서는 안 되며, OCPP 또는 기타(원격) 통신 연결을 통해 읽을 수 없어야 합니다.
A02.FR.06		충전기는 CertificateSignedRequest 메시지에서 서명된 인증서의 유효성을 검증해야 하며, 최소한 인증서가 유효한 기간, Certificate Properties의 속성, Certificate Hierarchy에 설명된 대로 충전기 운영자 인증서 계층의 일부인지 확인해야 합니다.
A02.FR.07		인증서가 유효하지 않은 경우. 충전기는 CertificateSignedRequest에 *Rejected* 상태로 응답하고 인증서를 삭제하고 *InvalidChargingStationCertificate* 보안 이벤트를 트리거해야 합니다(보안 이벤트의 전체 목록은 Part 2 - Appendices 참조).
A02.FR.08		충전기는 현재 날짜와 시간이 인증서의 '이전에는 유효하지 않음' 필드 뒤에 오는 즉시 새 인증서로 전환해야 합니다(예: 웹소켓 및 TLS 연결을 닫고 새 인증서로 다시 연결).

ID	전제 조건	요구 사항 정의
A02.FR.09	충전기에 *ChargingStationCertificate* 유형의 유효한 인증서가 두 개 이상 있는 경우.	충전기는 유효 기간의 시작을 기준으로 최신 인증서를 사용해야 합니다.
A02.FR.10	A02.FR.09 AND 충전기가 새 인증서가 작동하는지 검증한 경우	충전기는 이전 인증서를 삭제할 수 있습니다. 대체 수단으로 이전 인증서를 한 달 동안 저장하는 것이 좋습니다.
A02.FR.11	SignCertificateRequest를 수신하고 요청을 처리할 수 있는 경우	CSMS는 SignCertificateResponse에서 상태를 *Accepted* 로 설정해야 합니다.
A02.FR.12	SignCertificateRequest를 수신하고 요청을 처리할 수 없는 경우	CSMS는 SignCertificateResponse에서 상태를 *Rejected* 로 설정해야 합니다.
A02.FR.13	15118 연결과 충전기에서 CSMS 연결에 다른 인증서를 사용하는 경우	충전기는 SignCertificateRequest의 certificateType 필드를 업데이트가 트리거된 인증서로 설정해야 합니다.
A02.FR.14	certificateType이 포함된 SignCertificateRequest를 수신하는 경우	CSMS가 CertificateSignedRequest의 certificateType 필드를 SignCertificateRequest의 인증서 유형으로 설정하는 것이 좋습니다.
A02.FR.15	충전기에 동일한 루트 인증서에서 파생된 두 개 이상의 유효한 V2G 인증서가 있는 경우.	충전기는 유효 기간의 시작을 기준으로 최신 인증서를 사용해야 합니다.
A02.FR.16	구성 변수 MaxCertificateChainSize가 구현되고 충전기가 CertificateSignedRequest 메시지를 수신하고 크기가 MaxCertificateChainSize에서 구성된 설정 값을 초과하는 인증서(체인)가 있는 경우	충전기는 CertificateSignedResponse 메시지를 *Rejected* 상태로 응답해야 합니다.
A02.FR.17	CSMS가 CSR에 대한 SignCertificateRequest를 수락하고 충전기가 아직 이 CSR에 대한 CertificateSignedRequest를 수신하지 못했으며 CertSigningWaitMinimum에서 구성된 초가 경과된 경우	충전기는 CSR에 대한 새 SignCertificateRequest를 보내야 합니다. 선택적으로 이 CSR은 새로 생성된 키 쌍에 대한 것일 수 있습니다.
A02.FR.18	A02.FR.17	충전기는 이 CSR에 대한 CertificateSignedRequest를 수신하지 않고 백오프 시간이 만료될 때마다 CertSigningWaitMinimum에서 구성된 초부터 시작하여 이전 백오프 시간을 두 배로 늘려야 합니다.
A02.FR.19	A02.FR.18 AND 최대 증가 수에 도달했습니다.	충전기는 CSMS가 `SignChargingStationCertificate`, `SignV2GCertificate` 또는 `SignCombinedCertificate` 에 대한 TriggerMessageRequest를 통해 요청할 때까지 SignCertificateRequest 재전송을 중단해야 합니다.
A02.FR.20	A02.FR.07	충전기는 CSMS가 `SignChargingStationCertificate`, `SignV2GCertificate` 또는 `SignCombinedCertificate` 에 대한 TriggerMessageRequest를 통해 요청할 때까지 백오프 메커니즘을 시작하고 SignCertificateRequest를 재전송해서는 안 됩니다.
A02.FR.21	충전기가 signcertificateRequest에 대한 certificateType *V2GCertificate* 에 대한 응답으로 *Rejected* 상태의 SignCertificateResponse를 수신하는 경우	충전기가 재부팅될 때까지 ISO15118PnCEnabled를 끄는 것이 좋습니다.

A03 - Update Charging Station Certificate initiated by the Charging Station

Table 29. A03 - 충전기에서 시작된 충전기 인증서 업데이트

번호	유형	설명
1	**이름**	충전기에서 시작된 충전기 인증서 업데이트
2	**ID**	A03
	기능 블록	A. 보안
3	**목표**	충전기 클라이언트 측 인증서 관리를 용이하게 하기 위해 인증서 업데이트 절차가 제공됩니다.
4	**설명**	충전기에서 사용 중인 인증서(ChargingStationCertificate 또는 V2GCertificate for 15118)가 한 달 후에 만료됨을 감지합니다. 충전기는 SignCertificateRequest를 사용하여 키를 업데이트하는 프로세스를 시작하고, CertificateSigningUse 필드에 요청된 인증서를 표시합니다.
	Actors	충전기, CSMS, 인증 기관 서버
	시나리오 설명	**1.** 충전기는 충전기 인증서가 만료될 예정임을 감지합니다. **2.** 충전기는 새로운 공개/비공개 키 쌍을 생성합니다. **3.** 충전기는 해당 CertificateSigningUse가 포함된 SignCertificateRequest를 CSMS로 보냅니다. **4.** CSMS는 SignCertificateResponse로 응답하고 상태는 *Accepted* 입니다. **5.** CSMS가 CSR을 인증 기관 서버로 전달합니다. **6.** 인증 기관 서버가 인증서에 서명합니다. **7.** 인증 기관 서버가 서명된 인증서를 CSMS로 반환합니다. **8.** CSMS가 CertificateSignedRequest를 충전기로 보냅니다. **9.** 충전기가 서명된 인증서를 확인합니다. **10.** 충전기가 CertificateSignedResponse로 CSMS에 *Accepted* 또는 *Rejected* 상태로 응답합니다.
5	**필수 조건**	표준 구성 변수 `OrganizationName` 을 설정해야 합니다.
6	**사후 조건**	**성공한 사후 조건:** 충전기에 새 클라이언트 측 인증서가 설치되었습니다. **실패한 사후 조건:** 새 클라이언트 측 인증서가 거부되고 삭제됩니다.

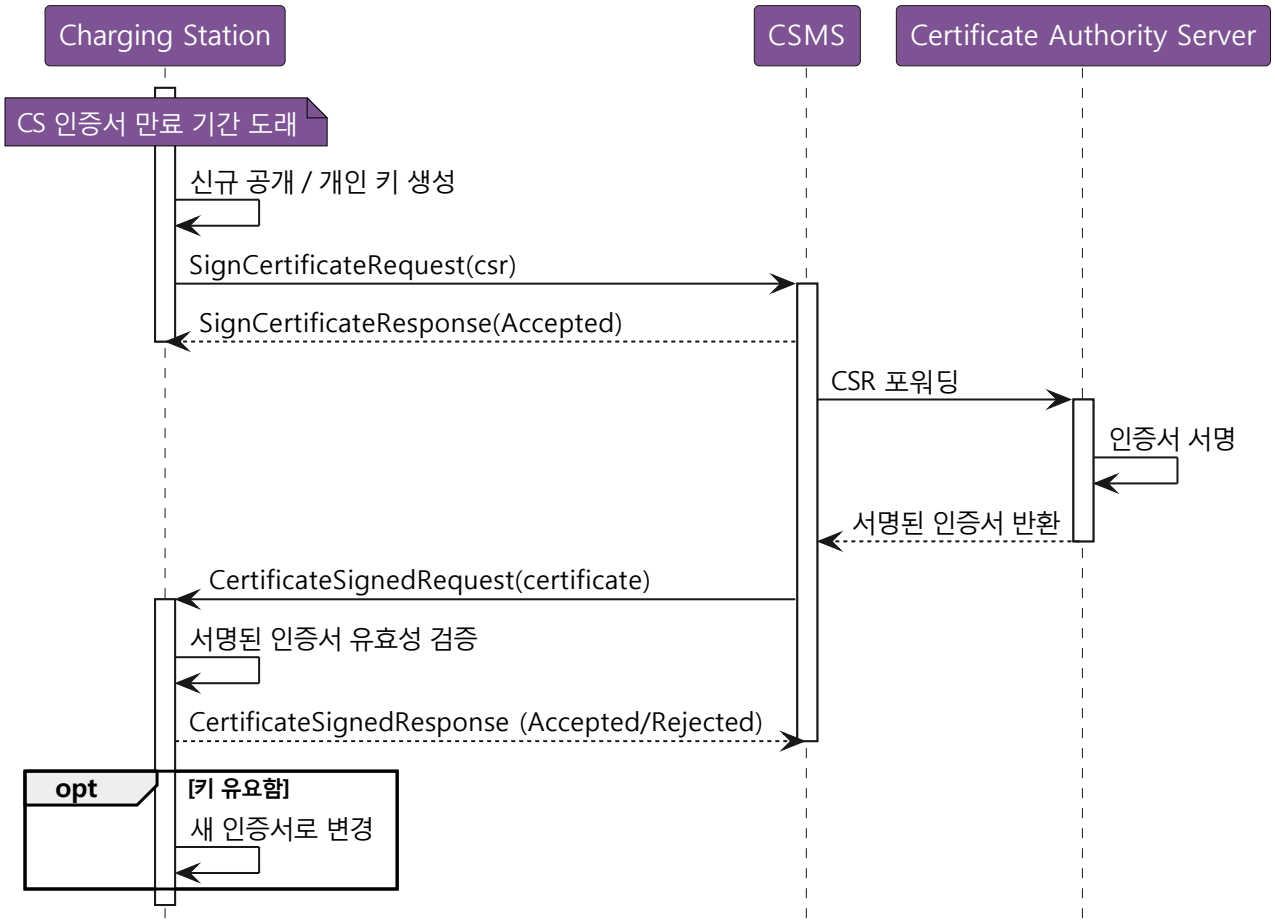

Figure 7. 충전기에서 시작된 충전기 인증서 업데이트

7	**오류 처리**	CSMS는 충전기에서 CSR 요청을 수락한 후 CA로 전달합니다. 하지만 CA에 도달할 수 없거나 CSR을 거부하면 충전기는 알 수 없습니다. CSMS는 CSR에 대한 몇 가지 검사를 수행할 수 있지만 CA가 수행하는 모든 검사를 수행할 수 없으며 CA에 대한 연결 시간 초과를 방지하지 못합니다. CA에 따르면, 이와 같은 문제는 CA가 오프라인이거나 충전기에서 보낸 CSR이 올바르지 않는 경우 발생합니다. 두 경우 모두 CSO의 운영자에게 알려야 할 사항입니다. 그런 다음 운영자는 문제를 조사해야 합니다. 문제가 해결되면 운영자는 A02를 다시 실행할 수 있습니다. 인증서가 X분 또는 시간 내에 전송되지 않을 때 충전기가 재시도하도록 하는 것은 권장되지 않습니다. CSR이 올바르지 않으면 자동으로 해결되지 않습니다. 새 펌웨어로만 이를 수정할 수 있습니다.
8	**참고**	A02 - Update Charging Station Certificate by request of CSMS와 동일한 주의 사항이 적용됩니다.

A03 - Update Charging Station Certificate initiated by the Charging Station - Requirements

Table 30. A03 - 요구 사항

ID	전제 조건	요구 사항 정의
A03.FR.01		충전기 설치 후 키 업데이트를 수행하여 제조업체에서 처음 제공한 키(아마도 기본 키)에서 키를 변경할 수 있습니다.
A03.FR.02	충전기에서 현재 충전기 인증서가 한 달 후에 만료됨을 감지하는 경우.	충전기는 [16]의 Section 4.2.1.3에 설명된 키 생성 기능 중 하나를 사용하여 새 공개/비공개 키 쌍을 생성해야 합니다.
A03.FR.03	A03.FR.02	충전기는 RFC 2986 [22]에 설명된 대로 인증서 서명 요청(CSR) 형태로 공개 키를 전송한 다음 SignCertificateRequest 메시지를 사용하여 PEM 인코딩해야 합니다.

ID	전제 조건	요구 사항 정의
A03.FR.04		CSMS는 인증서 자체에 서명해서는 안 되지만, 대신 CSR을 충전기 인프라의 인증서를 관리하는 전담 인증 기관 서버로 전달해야 합니다. 전담 기관 서버는 CSO에서 운영할 수 있습니다.
A03.FR.05		키 업데이트 프로세스 중에 충전기에서 생성한 개인 키는 절대 충전기를 떠나서는 안 되며, OCPP 또는 기타(원격) 통신 연결을 통해 읽을 수 없어야 합니다.
A03.FR.06		충전기는 CertificateSignedRequest 메시지에서 서명된 인증서의 유효성을 검증해야 하며, 최소한 인증서가 유효한 기간, Certificate Properties의 속성, Certificate Hierarchy에 설명된 대로 충전기 운영자 인증서 계층의 일부인지 확인해야 합니다.
A03.FR.07	인증서가 유효하지 않은 경우.	충전기는 CertificateSignedRequest에 Rejected 상태로 응답하고 인증서를 삭제하고 *InvalidChargingStationCertificate* 보안 이벤트를 트리거해야 합니다(보안 이벤트의 전체 목록은 Part 2 - Appendices 참조).
A03.FR.08		충전기는 현재 날짜와 시간이 인증서의 '이전에는 유효하지 않음' 필드 뒤에 오는 즉시 새 인증서로 전환해야 합니다(예: 웹소켓 및 TLS 연결을 닫고 새 인증서로 다시 연결).
A03.FR.09	충전기에 *ChargingStationCertificate* 유형의 유효한 인증서가 두 개 이상 있는 경우.	충전기는 유효 기간의 시작을 기준으로 최신 인증서를 사용해야 합니다.
A03.FR.10	A03.FR09 AND 충전기에서 새 인증서가 작동하는지 검증한 경우	충전기는 이전 인증서를 삭제할 수 있습니다. 대체 수단으로 이전 인증서를 한 달 동안 저장하는 것이 좋습니다.
A03.FR.11	SignCertificateRequest를 수신하고 요청을 처리할 수 있는 경우	CSMS는 SignCertificateResponse에서 상태를 *Accepted* 로 설정해야 합니다.
A03.FR.12	SignCertificateRequest를 수신하고 요청을 처리할 수 없는 경우	CSMS는 SignCertificateResponse에서 상태를 *Rejected* 로 설정해야 합니다.
A03.FR.13	15118 연결과 충전기에서 CSMS 연결에 다른 인증서를 사용하는 경우	충전기는 SignCertificateRequest에 certificateType 필드를 포함하여 업데이트하려는 인증서를 지정해야 합니다.
A03.FR.14	certificateType이 포함된 SignCertificateRequest를 수신하는 경우	CSMS가 CertificateSignedRequest의 certificateType 필드를 SignCertificateRequest의 인증서 유형으로 설정하는 것이 좋습니다.
A03.FR.15	충전기에 동일한 루트 인증서에서 파생된 두 개 이상의 유효한 V2G 인증서가 있는 경우.	충전기는 유효 기간의 시작을 기준으로 최신 인증서를 사용해야 합니다.
A03.FR.16	구성 변수 MaxCertificateChainSize가 구현되고 충전기가 CertificateSignedRequest 메시지를 수신하고 크기가 MaxCertificateChainSize에서 구성된 설정 값을 초과하는 인증서(체인)가 포함된 경우	충전기는 CertificateSignedResponse 메시지를 Rejected 상태로 응답해야 합니다.
A03.FR.17	CSMS가 CSR에 대한 SignCertificateRequest를 수락하고 충전기가 아직 이 CSR에 대한 CertificateSignedRequest를 수신하지 못하고 CertSigningWaitMinimum에서 구성된 초 수가 만료된 경우	충전기는 CSR에 대한 새 SignCertificateRequest를 보내야 합니다. 선택적으로 이 CSR은 새로 생성된 키 쌍에 대한 것일 수 있습니다.

ID	전제 조건	요구 사항 정의
A03.FR.18	A03.FR.17	충전기는 이 CSR에 대한 CertificateSignedRequest를 수신하지 않고 백오프 시간이 만료될 때마다 CertSigningWaitMinimum에서 구성된 초 수부터 시작하여 이전 백오프 시간을 두 배로 늘려야 합니다.
A03.FR.19	A03.FR.18 AND 최대 증가 수에 도달했습니다.	충전기는 CSMS가 `SignChargingStationCertificate`, `SignV2GCertificate` 또는 `SignCombinedCertificate` 에 대한 TriggerMessageRequest를 통해 요청할 때까지 SignCertificateRequest 재전송을 중단해야 합니다.

A04 - Security Event Notification.

Table 31. A04 - 보안 이벤트 알림

번호	유형	설명
1	**이름**	보안 이벤트 알림
2	**ID**	A04
	기능 블록	A. 보안
3	**목표**	CSMS에 중요한(critical) 보안 이벤트를 알립니다.
4	**설명**	이 사용 사례를 통해 충전기는 시스템 보안의 변경 사항을 CSMS에 즉시 알릴 수 있습니다.
	Actors	CSMS, 충전기
	시나리오 설명	**1.** *중요한(critical)* 보안 이벤트가 발생합니다. **2.** 충전기는 SecurityEventNotificationRequest를 CSMS로 전송합니다. **3.** CSMS는 SecurityEventNotificationResponse로 충전기에 응답합니다.
5	**필수 조건**	해당 없음
6	**사후 조건**	충전기는 SecurityEventNotificationRequest를 CSMS로 전송하여 CSMS에 중요한 보안 이벤트를 *성공적으로* 알립니다.

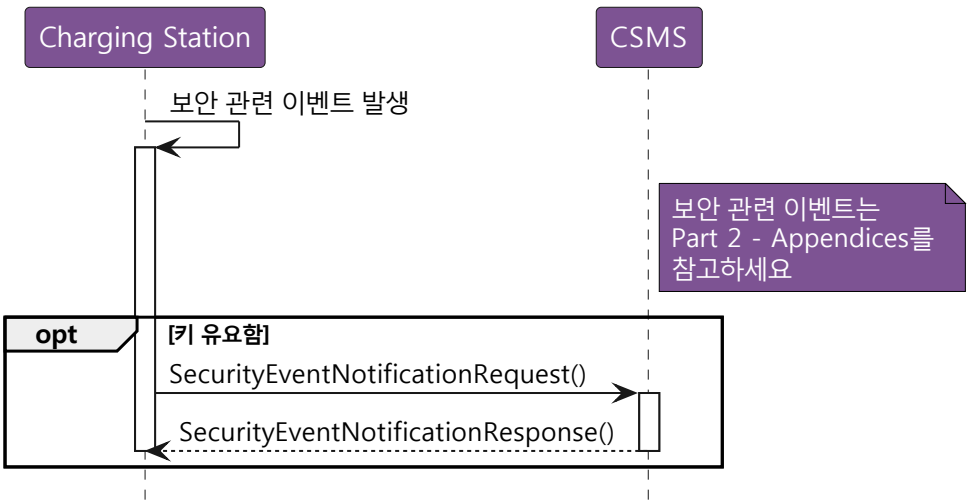

Figure 8. 보안 이벤트 알림

7	**오류 처리**	n/a
8	**참고**	보안 관련 이벤트 목록과 해당 '중요도'는 부록(*Appendix 1. Security Events*)에 제공됩니다.

A04 - Security Event Notification - Requirements

Table 32. A04 - 보안 이벤트 알림 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
A04.FR.01	중요 보안 이벤트가 발생하는 경우	충전기는 CSMS에 SecurityEventNotificationRequest를 보내 보안 이벤트를 알려야 합니다.	
A04.FR.02	A04.FR.01 AND 충전기가 연결 해제됩니다.	보안 이벤트 알림은 CSMS에서 보장된 전달로 대기열에 있어야 합니다.	
A04.FR.03	A04.FR.01	CSMS는 SecurityEventNotificationResponse 메시지를 사용하여 알림 수신을 확인해야 합니다.	
A04.FR.04	보안 이벤트가 발생하는 경우(중요하지 않은 경우)	충전기는 보안 로그에 보안 이벤트를 저장해야 합니다.	이 로그는 롤링 형식으로 구현하는 것이 좋습니다.

A05 - Upgrade Charging Station Security Profile

Table 33. A05 - 충전기 보안 프로필 업그레이드

번호	유형	설명
1	**이름**	충전기 보안 프로필 업그레이드
2	**ID**	A05
	기능 블록	A. 보안
3	**목표**	CSO는 CSMS와 충전기 간 OCPP 연결의 보안을 강화하고자 합니다.
4	**설명**	보안 프로필이 없는 OCPP 1.6에서 보안 프로필이 있는 OCPP 1.6 또는 OCPP 2.0.1로 마이그레이션할 때의 사용 사례 보안 프로필로 마이그레이션하기 전에 설치된 인증서 또는 비밀번호와 같은 필수 조건을 구성해야 합니다.
	Actors	CSMS, 충전기
	시나리오 설명	**1.** CSMS는 SetVariablesRequest를 통해 `NetworkConfigurationPriority` 구성 변수에 대한 새 값을 설정하여 새(더 높은) 보안 프로필에 대한 NetworkConnectionProfile이 목록에서 첫 번째가 되고 기존 연결 프로필이 목록에서 두 번째가 됩니다. **2.** 충전기는 SetVariablesResponse로 응답하고 상태는 *Accepted* 입니다. **3.** CSMS는 ResetRequest(OnIdle)를 보냅니다. **4.** 충전기는 재부팅하고 새 기본 NetworkConnectionProfile을 통해 연결합니다.
5	**필수 조건**	CSO는 NetworkConnectionProfile이 (더 높은) 보안 프로필을 사용하여 설정되었는지 확인합니다. AND 더 높은 보안 프로필로 변경하라는 명령을 보내기 전에 더 높은 보안 프로필로 이동하기 위한 필수 조건이 충족되었는지 확인합니다.

번호	유형	설명
6	사후 조건	충전기가 더 높은 보안 프로필로 성공적으로 업그레이드되었습니다.

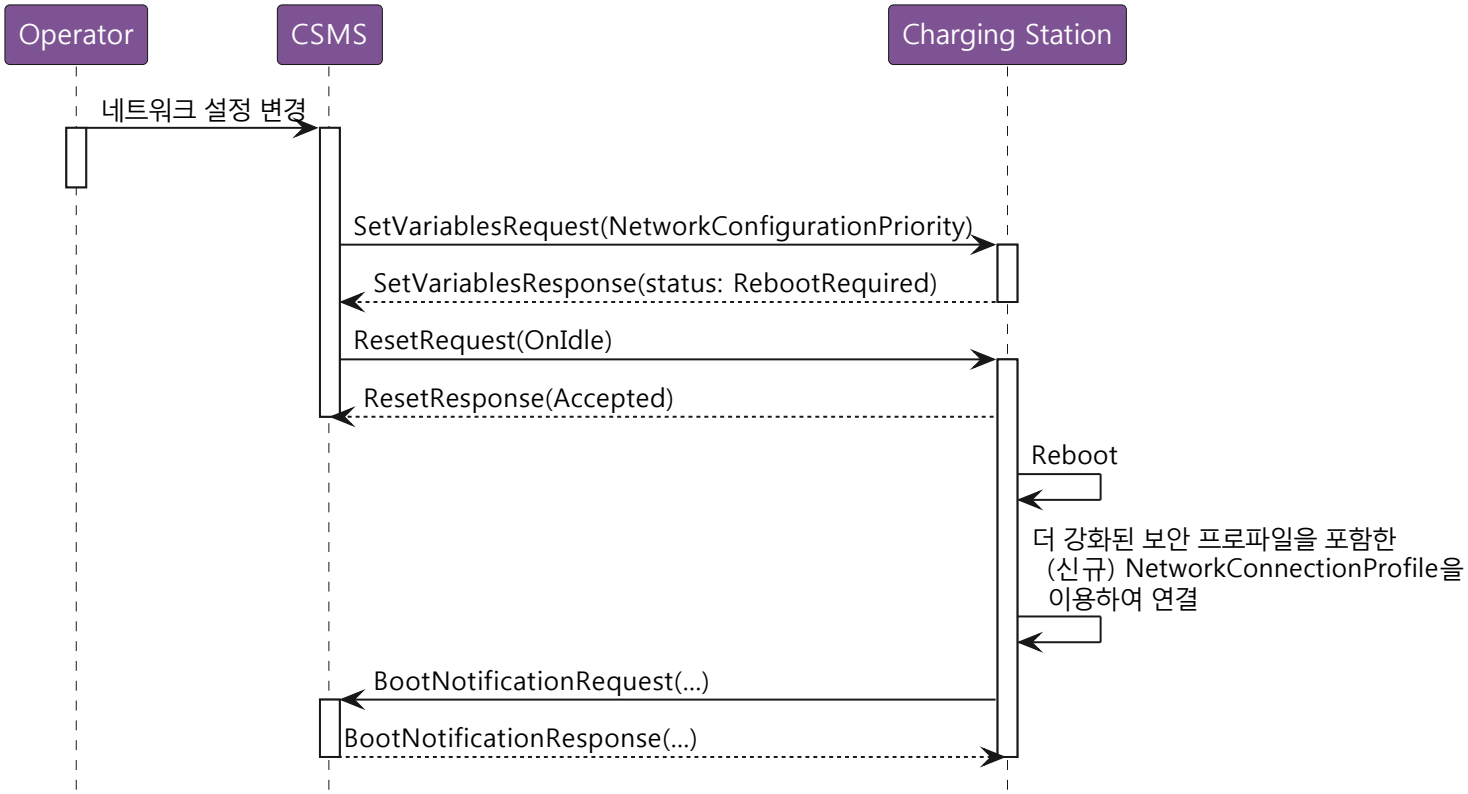

Figure 9. 충전기 보안 프로필 업그레이드

7	오류 처리	해당 없음
8	참고	보안상의 이유로 OCPP를 사용하여 더 낮은 보안 프로필로 되돌릴 수 없습니다.

- Upgrade Charging Station Security Profile - Requirements

Table 34. A05 - 충전기 보안 프로필 업그레이드

ID	전제 조건	요구 사항 정의
A05.FR.02	충전기는 `NetworkConfigurationPriority`에 대한 SetVariablesRequest를 수신합니다. 여기에는 현재 값보다 높은 'securityProfile' 값을 가진 NetworkConnectionProfile에 대한 프로필 슬롯이 포함됩니다. AND 새 값은 2 또는 3입니다. AND 유효한 CSMSRootCertificate가 설치되지 않았습니다.	충전기는 SetVariablesResponse(Rejected)로 응답해야 하며 값을 업데이트하지 않아야 합니다. `SecurityProfile` 및/또는 CSMS에 다시 연결합니다.

ID	전제 조건	요구 사항 정의
A05.FR.03	충전기는 NetworkConfigurationPriority에 대한 SetVariablesRequest를 수신하며, 여기에는 현재 값보다 높은 'securityProfile' 값을 가진 NetworkConnectionProfile에 대한 프로필 슬롯이 포함됩니다. AND 새 값은 3입니다. AND 유효한 ChargingStationCertificate가 설치되지 않았습니다.	충전기는 SetVariablesResponse(Rejected)로 응답해야 하며, SecurityProfile에 대한 값을 업데이트하거나 CSMS에 다시 연결해서는 안 됩니다.
A05.FR.04	충전기는 NetworkConfigurationPriority에 대한 SetVariablesRequest를 수신합니다. 여기에는 현재 값과 같거나 더 높은 'securityProfile' 값을 가진 NetworkConnectionProfile에 대한 프로필 슬롯이 포함됩니다. AND 모든 전제 조건이 충족됩니다.	충전기는 SetVariablesResponse(Accepted)로 응답해야 합니다.
A05.FR.05	A05.FR.04 AND 재부팅 후	충전기는 NetworkConfigurationPriority의 첫 번째 항목에 연결을 시작해야 합니다.
A05.FR.06	A05.FR.05 AND 충전소가 (새로운) NetworkConnectionProfile을 사용하여 CSMS에 성공적으로 연결되었습니다.	충전기는 구성 변수 SecurityProfile의 값을 업데이트해야 하며 SecurityProfile에 저장된 것보다 낮은 securityProfile을 가진 모든 NetworkConnectionProfile을 제거하고 그에 따라 NetworkConfigurationPriority를 업데이트해야 합니다.
A05.FR.07	A05.FR.06	CSMS는 더 이상 충전기가 낮은 보안 프로필로 연결하도록 허용해서는 안 됩니다.

B. Provisioning

1. Introduction

이 기능 블록은 CSO가 충전기를 프로비저닝하고, 네트워크에서 충전기를 허용하고, 충전기에서 구성 정보를 검색하는 데 도움이 되는 모든 기능을 설명합니다. 또한 충전기 구성에 대한 정보를 검색하고, 구성을 변경하는 등의 기능으로 구성됩니다. 이 장에서는 충전기를 재설정하고 새 NetworkConnectionProfile로 마이그레이션하는 방법도 다룹니다.

1.1. Transactions before being accepted by a CSMS

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

재시작(예: 원격 재설정 명령, 정전, 펌웨어 업데이트, 소프트웨어 오류 등으로 인해) 후 충전기는 반드시 CSMS에 다시 연락하여 BootNotification 요청을 보내야 합니다. 충전기가 CSMS에서 BootNotificationResponse를 수신하지 못하고 올바르게 사전 설정된 내장형 비휘발성 실시간 클록 하드웨어가 없는 경우, 충전기에 유효한 날짜 및 시간 설정이 없어 나중에 트랜잭션의 날짜와 시간을 확인하기 어렵거나 불가능할 수 있습니다.

또한 CSMS가 장시간 또는 무기한으로 수락됨이 아닌 상태를 표시하는 경우도 있습니다(예: 구성 오류로 인해).

CSMS가 충전기를 수락(Accepted)한 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.

이것이 지원되는 경우 이 동작은 구성 변수 `TxBeforeAcceptedEnabled`를 통해 구성할 수 있습니다.

2. Use cases & Requirements

2.1. Booting a Charging Station

B01 - Cold Boot Charging Station

Table 35. B01 - 충전기 콜드 부트

번호	유형	설명
1	**이름**	충전기 콜드 부트
2	**ID**	B01
	기능 블록	B. 프로비저닝
3	**목표**	이 사용 사례의 목적은 전원이 켜지는 충전기가 CSMS에 등록하고 올바른 상태 정보를 제공할 수 있도록 하는 것입니다.
4	**설명**	이 사용 사례는 CSMS가 어떤 충전기가 CSMS에 액세스하는지 제어하는 방법을 설명합니다. CSMS에 연결된 충전기를 제어하려면 충전기가 BootNotificationRequest를 보내야 합니다. 이 요청에는 충전기에 대한 몇 가지 일반 정보가 포함되어 있습니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 충전기에 전원이 공급됩니다. **2.** 충전기에서 BootNotificationRequest를 CSMS로 전송합니다. **3.** CSMS에서 BootNotificationResponse를 *Accepted* 상태로 반환합니다. **4.** *선택 사항:* 충전기에서 StatusNotificationRequest를 각 커넥터에 대한 CSMS로 *Unavailable* 상태로 전송합니다. **5.** 충전기에서 각 커넥터에 대한 StatusNotificationRequest를 CSMS로 전송합니다. (재)부팅 전에 CSMS에서 상태가 *Unavailable* 또는 *Reserved* 으로 설정된 경우 커넥터는 이 상태로 돌아가야 하며, 그렇지 않은 경우 상태가 *Available* 이어야 하고, 진행 중이었던 트랜잭션을 재개할 때 상태가 *Occupied* 이어야 합니다. **6.** 정상적인 운영이 재개됩니다. **7.** 충전기가 CSMS에 HeartbeatRequest를 보냅니다.
	대체 시나리오	B02 - Cold Boot Charging Station - Pending B03 - Cold Boot Charging Station - Rejected
5	**필수 조건**	충전기의 전원이 꺼졌습니다.
6	**사후 조건**	**성공한 사후 조건:** 충전기가 *Idle* 상태이고 *Accepted* 입니다. **실패 사후 조건:** 충전기가 *Rejected* 상태를 수신했습니다. B03 - Cold Boot Charging Station - Rejected이 적용됩니다. 충전기가 *Pending* 상태를 수신했습니다. B02 - Cold Boot Charging Station - Pending이 적용됩니다.

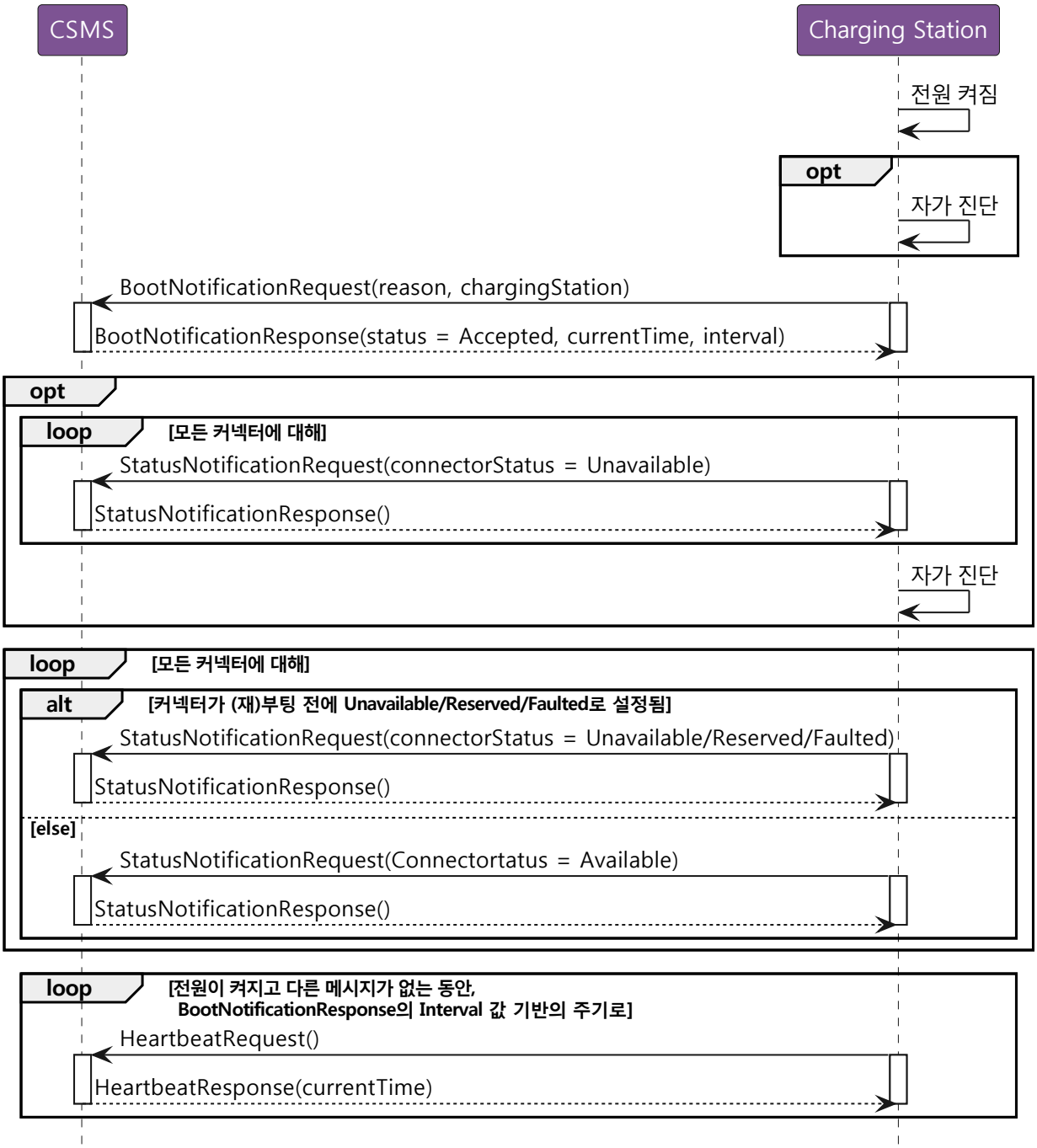

Figure 10. 시퀀스 다이어그램: 충전기 콜드 부트

7	**오류 처리**	**1.** CSMS와 충전기 간의 통신 연결 초기 설정 안 됨: CSMS와의 연결을 다시 시도하세요. **2.** CSMS에서 응답 없음/시간 초과: 충전기는 대기 간격 후에 BootNotificationRequest를 다시 보냅니다. 충전기는 이 간격을 스스로 선택합니다(이 간격을 포함하는 BootNotificationResponse를 받지 못했기 때문에). 이렇게 하면 CSMS에 요청이 넘쳐나는 것을 방지할 수 있습니다.
8	**참고**	자체 검사를 위한 여러 옵션이 가능합니다. 일부 충전기는 부팅하고 *Unavailable* 로 상태 알림을 보낸 다음, 모든 하드웨어를 검사하고 충전기가 가동되면 *Available* 상태로 새 StatusNotifications를 보냅니다. 그러나 자체 검사를 수행하고 BootNotificationRequest를 보내는 데 필요한 순서는 없습니다. 충전기는 BootNotificationRequest를 보내기 *전에* 자체 검사를 수행하고 (모바일) 네트워크 연결이 설정되고 BootNotificationRequest가 전송되기 전에 상태를 확인할 수도 있습니다. 충전기나 EVSE에 문제가 있는 경우 상태는 *Faulted* 로 설정해야 합니다. *Reserved* 및 *Unavailable* 상태는 재부팅 후에도 지속됩니다.

B01 - Cold Boot Charging Station - Requirements

Table 36. B01 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
B01.FR.01	시작 후.	충전기는 구성에 대한 정보와 함께 BootNotificationRequest를 CSMS로 보내야 합니다.	정보: 버전, 공급업체 등
B01.FR.02	B01.FR.01 CSMS는 충전기에서 BootNotificationRequest를 수신했습니다.	CSMS는 충전기를 수락할지 여부를 나타내기 위해 응답해야 합니다.	
B01.FR.03	리부팅 후(예: 원격 재설정 명령, 정전, 펌웨어 업데이트, 소프트웨어 오류 등)	충전기는 CSMS에 다시 연결하고 부팅하거나 재부팅할 때마다 BootNotificationRequest를 전송해야 합니다.	
B01.FR.04	CSMS가 BootNotificationResponse로 응답하고 상태가 `Accepted` 이고 `interval` > 0인 경우	충전기는 응답 메시지의 간격에 따라 heartbeat 간격을 조정해야 합니다.	
B01.FR.05	CSMS가 BootNotificationResponse로 응답하고 상태가 Accepted인 경우	충전기는 각 커넥터에 대해 현재 상태와 함께 StatusNotificationRequest를 보내야 합니다.	
B01.FR.06	충전기는 BootNotificationResponse를 수신했습니다. AND 충전기는 heartbeat를 사용하여 시간 동기화의 `TimeSource`로 설정합니다	충전기는 충전기의 내부 시계를 제공된 CSMS의 현재 시간과 동기화해야 합니다.	
B01.FR.07	충전기 또는 EVSE가 Change Availability 명령에 의해 *Unavailable* 상태로 설정된 경우.	*Unavailable* 상태는 재부팅 시에도 지속되어야 합니다.	
B01.FR.08	물리적 전원 켜기/재부팅과 BootNotification의 성공적인 완료 사이, CSMS가 *Accepted* 또는 *Pending* 을 반환하는 경우.	충전기는 CSMS에 다른 OCPP 요청을 보내서는 안 됩니다(BootNotificationRequest 제외). 여기에는 이전 세션에서 충전기에 여전히 있는 캐시된 OCPP 메시지가 포함됩니다.	*Pending* 상태에서 메시지를 보내는 것에 대한 자세한 내용은 B02 - Cold Boot Charging Station - Pending(예: B02.FR.02)을 참조하세요.
B01.FR.09	B01.FR.01	충전기는 reason 필드에 BootNotificationRequest 메시지를 보낸 이유를 표시해야 합니다.	사용할 이유에 대해서는 BootReasonEnumType을 참조하세요.
B01.FR.10	충전기가 상태가 *Accepted* 가 아닌 BootNotificationResponse를 수신했습니다. AND 충전기가 RPC Framework: CALL 메시지를 전송하는데, 이 메시지는 BootNotificationRequest가 아니거나 다음 메시지 중 하나에 의해 트리거된 메시지가 아닙니다. TriggerMessageRequest, GetBaseReportRequest, GetReportRequest.	CSMS는 RPC Framework로 응답해야 합니다. CALLERROR: SecurityError.	충전기는 수락되기 전에 다른 메시지를 보내는 것을 시작할 수 없습니다.
B01.FR.11	B01.FR.01 및 보안 프로필 3이 사용됩니다.	CSMS는 BootNotificationRequest의 일련 번호를 인증서 일반 이름의 일련 번호와 비교해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
B01.FR.12	B01.FR.11 및 BootNotificationRequest의 일련 번호가 인증서 일반 이름의 일련 번호와 같지 않습니다.	CSMS는 WebSocket 연결을 닫아야 합니다.	
B01.FR.13	EVSE가 예약된 경우	*Reserved* 상태는 재부팅 시에도 지속되어야 합니다.	

B02 - Cold Boot Charging Station - Pending

Table 37. B02 - 충전기 콜드 부트 - 보류 중

번호	유형	설명
1	**이름**	충전기 콜드 부트 - 보류 중
2	**ID**	B02
	기능 블록	B. 프로비저닝
	부모 사용 사례	B01 - Cold Boot Charging Station
3	**목표**	**1.** 충전기에 아직 CSMS에서 수락되지 않았음을 알리는 것: *Pending* 상태. **2.** CSMS에 특정 구성 정보를 검색하거나 설정하는 방법을 제공하는 것. **3.** CSMS가 예를 들어 CSMS 재부팅 후 CSMS의 부하를 제한할 수 있는 방법을 제공합니다.
4	**설명**	이 사용 사례는 CSMS가 *Pending* 상태를 사용하여 아직 수락되지 않았다고 충전기에 알렸을 때 CSMS와 충전기의 동작을 설명합니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 충전기에 전원이 켜졌습니다. **2.** 충전기가 BootNotificationRequest를 CSMS로 보냅니다. **3.** CSMS가 BootNotificationResponse를 *Pending* 상태로 응답합니다. **4.** 그런 다음 CSMS가 충전기에 메시지를 보내 충전기의 구성을 변경할 수 있습니다. **5.** 충전기는 간격 필드에 표시된 초 수 후에 BootNotificationRequest를 다시 전송합니다. (BootNotificationResponse부터의 간격)
5	**필수 조건**	**1.** CSMS는 충전기를 *Pending* 상태로 설정해야 합니다. **2.** 충전기가 시작됩니다(즉, 전원이 꺼진 후 전원이 켜짐).
6	**사후 조건**	**성공 사후 조건:** 충전기가 *Pending* 상태입니다. **실패 사후 조건:** 충전기가 *Rejected* 상태, B03 - Cold Boot Charging Station - Rejected을 수신했습니다.

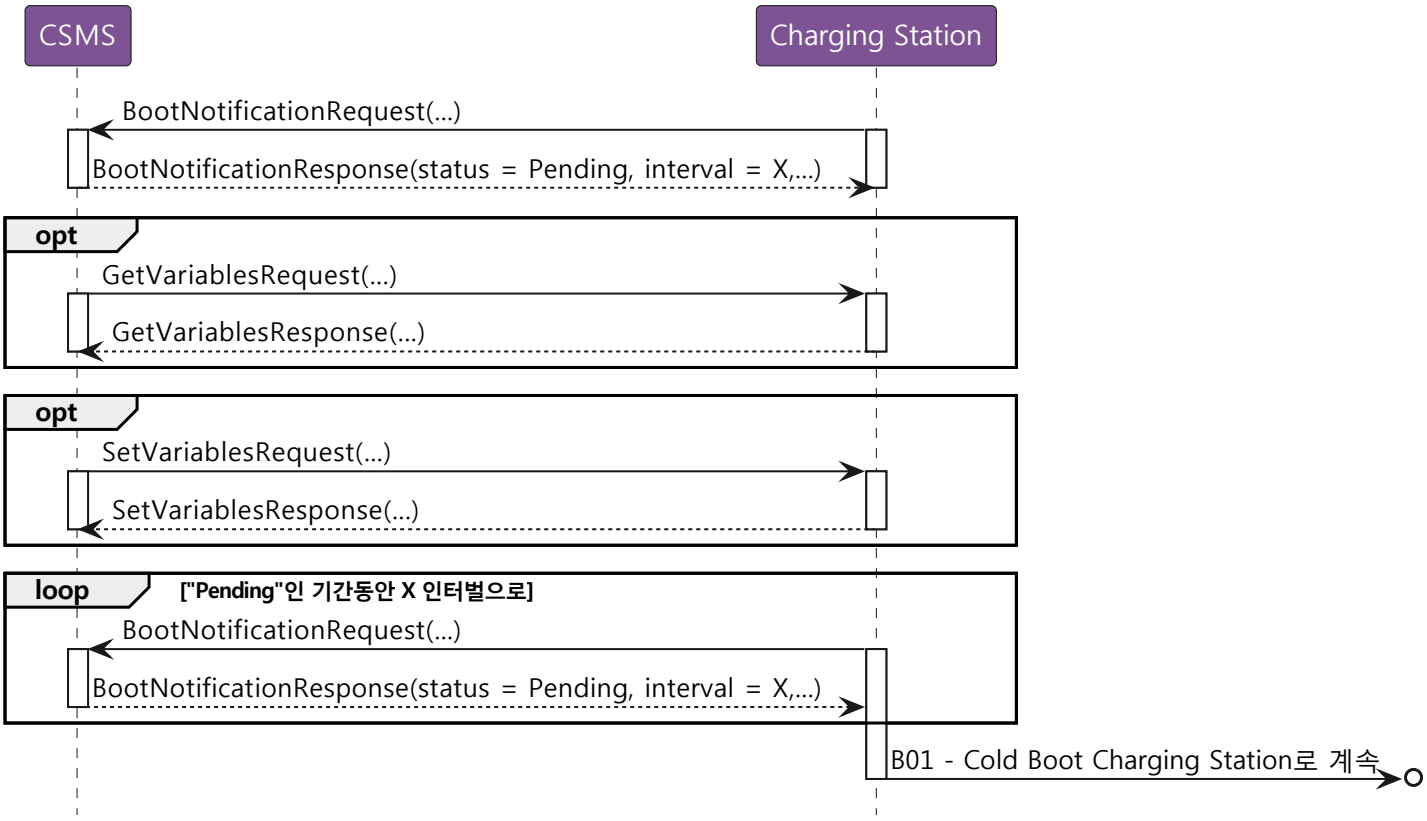

Figure 11. 시퀀스 다이어그램: 충전기 콜드 부트 - 보류 중

7	**오류 처리**	**1.** CSMS와 충전기 간에 초기 연결이 설정되지 않은 경우: CSMS에 연결을 다시 시도하고 BootNotificationRequest를 다시 보냅니다. **2.** CSMS에서 응답 없음/시간 초과: 충전기는 대기 간격 후에 BootNotificationRequest를 다시 보냅니다. 이 대기 간격은 이전 BootNotificationResponse의 간격을 기반으로 하거나 충전기 자체에서 선택할 수 있습니다. 후자의 경우, 충전기는 CSMS에 요청이 범람하는 것을 방지하는 방식으로 이 간격을 선택합니다.
8	**참고**	CSMS가 BootNotificationResponse를 *Accepted* 상태로 반환하면 B01 - Cold Boot Charging Station이 적용됩니다.

B02 - Cold Boot Charging Station - Pending - Requirements

Table 38. B02 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
B02.FR.01	충전기가 *Pending* 상태를 수신한 후.	CSMS는 충전기에서 정보를 검색하기 위해 메시지를 보내거나(사용 사례 B06, B07, B08에서 설명한 대로) SetVariablesRequest를 통해 구성을 변경할 수 있습니다(사용 사례 B05에서 설명한 대로). 충전기는 이러한 메시지에 응답해야 합니다.	Pending 상태는 CSMS가 충전기를 수락하기 전에 충전기에서 특정 정보를 검색하거나 설정하려고 한다는 것을 나타낼 수 있습니다.
B02.FR.02	CSMS가 아직 BootNotificationRequest에 BootNotificationResponse에서 *Accepted* 상태로 응답하지 않은 경우.	충전기는 CSMS가 다음 메시지 중 하나를 사용하여 그렇게 하도록 지시하지 않는 한 CSMS에 RPC Framework: CALL 메시지(BootNotificationRequest 제외)를 보내서는 안 됩니다. TriggerMessageRequest, GetBaseReportRequest, GetReportRequest.	

ID	전제 조건	요구 사항 정의	참고
B02.FR.03	CSMS가 아직 BootNotificationRequest에 BootNotificationResponse에서 *Accepted* 상태를 나타내는 응답을 하지 않은 경우.	충전기 운영자는 충전기를 구성하여 트랜잭션을 수락하고 CSMS로 보낼 TransactionEventRequest 메시지를 대기열에 넣을 수 있습니다.	이 동작을 구현하려는 당사자는 해당 트랜잭션이 CSMS에 전달될 수 있는지 여부가 불확실하다는 것을 알아야 합니다.
B02.FR.04	CSMS가 아직 BootNotificationResponse에서 *Accepted* 상태를 나타내는 BootNotificationRequest에 응답하지 않은 경우.	충전기는 TriggerMessageRequest로 요청하지 않는 한 BootNotificationResponse의 Interval 필드 값보다 일찍 BootNotificationRequest를 보내서는 안 됩니다.	
B02.FR.05	*Pending* 상태이고 RequestStartTransactionRequest 또는 RequestStopTransactionRequest를 수신하는 경우	충전기는 RequestStartTransactionResponse 또는 RequestStopTransactionResponse에 *Rejected* 상태로 응답해야 합니다. (충전기가 트랜잭션을 시작할 수 있도록 허용된 경우에도 B02.FR.03 참조. CSMS가 RequestStartTransaction 등을 사용하려면 먼저 충전기를 수락(accept)해야 함)	
B02.FR.06	CSMS가 Pending 상태를 반환하는 경우	통신 채널은 충전기나 CSMS에 의해 닫히지 않아야 함	
B02.FR.07	BootNotificationResponse의 간격이 0이고 상태가 *Accepted* 가 아닌 경우	충전기는 CSMS에 요청이 범람하지 않도록 자체적으로 대기 간격을 선택해야 함.	
B02.FR.08	BootNotificationResponse의 간격이 0보다 크고 상태가 *Accepted* 가 아닌 경우,	충전기는 설정된 간격이 지난 후 BootNotificationRequest를 보내야 합니다.	
B02.FR.09	충전기에서 BootNotificationResponse를 수신했으며 상태는 *Pending* 입니다. AND 충전기에서 RPC 프레임워크: CALL 메시지를 전송하는데, 이 메시지는 BootNotificationRequest가 아니고 다음 메시지 중 하나에 의해 트리거된 메시지도 아닙니다. TriggerMessageRequest, GetBaseReportRequest, GetReportRequest.	CSMS는 RPC 프레임워크로 응답해야 합니다. CALLERROR: SecurityError.	충전기는 수락되기 전에 다른 메시지를 보내는 것을 시작할 수 없습니다.

B03 - Cold Boot Charging Station - Rejected

Table 39. B03 - 충전기 콜드 부트 - 거부됨

번호	유형	설명
1	**이름**	충전기 콜드 부트 - 거부됨
2	**ID**	B03
	기능 블록	B. 프로비저닝
	부모 사용 사례	B01 - Cold Boot Charging Station
3	**목표**	충전기에 CSMS에서 아직 수락되지 않았다는 사실을 알리는 것: *Rejected* 상태.
4	**설명**	이 사용 사례는 CSMS와 충전기의 동작을 설명하는데, 충전기가 CSMS로부터 *Rejected* 상태를 사용하여 (아직) 수락되지 않았다는 통보를 받을 때입니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 충전기에 전원이 공급됩니다. **2.** 충전기가 BootNotificationRequest를 CSMS로 보냅니다. **3** CSMS가 BootNotificationResponse를 충전기에 *Rejected* 상태로 응답합니다. **4.** 충전기는 간격 필드에 표시된 초 후에 BootNotificationRequest를 다시 보냅니다. (BootNotificationResponse부터의 간격).
5	**필수 조건**	**1.** CSMS는 충전기를 *Rejected* 상태로 설정해야 합니다. **2.** 충전기의 전원이 꺼졌습니다.
6	**사후 조건**	충전기는 *Rejected* 상태를 유지합니다.

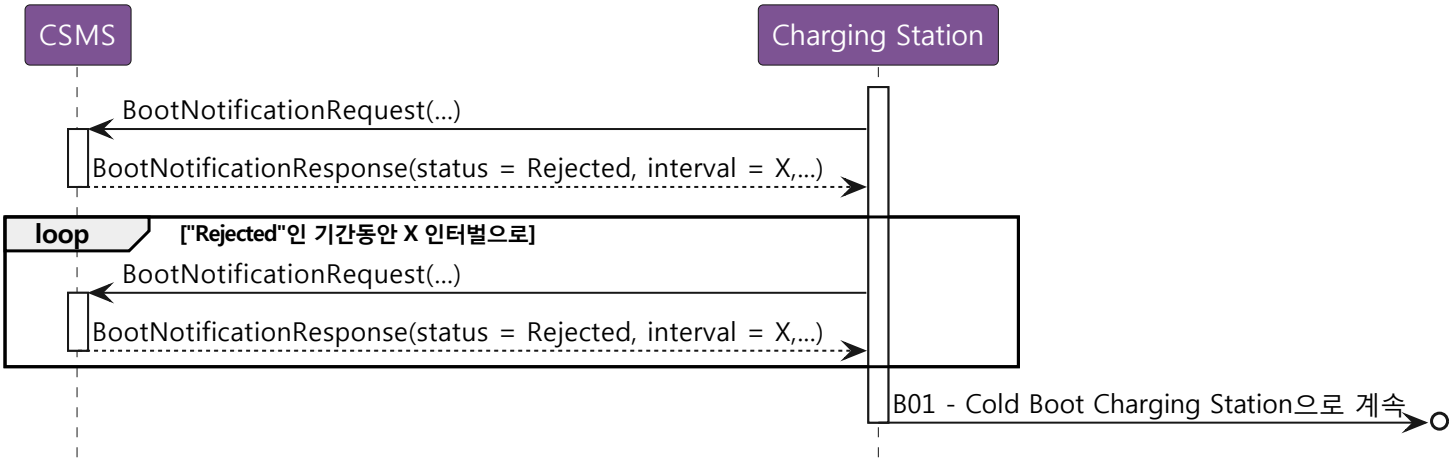

Figure 12. 시퀀스 다이어그램: 충전기 콜드 부트 - 거부됨

7	**오류 처리**	CSMS에서 응답이 없거나 시간 초과가 발생하는 경우: 충전기는 대기 간격 후에 BootNotificationRequest를 다시 보냅니다. 이 대기 간격은 이전 BootNotificationResponse의 간격을 기반으로 하거나 충전기 자체에서 선택할 수 있습니다. 후자의 경우 충전기는 CSMS에 요청이 넘쳐나는 것을 방지하는 방식으로 이 간격을 선택합니다.

		Rejected 상태 동안 충전기는 더 이상 CSMS에서 도달할 수 없습니다. 충전기는 예를 들어 통신 채널을 닫거나 통신 하드웨어를 종료할 수 있습니다. 또한 CSMS는 예를 들어 시스템 리소스를 확보하기 위해 통신 채널을 닫을 수 있습니다.
8	**참고**	충전기의 BootNotification이 CSMS에서 수락될 때까지 어떠한 트랜잭션도 수락하지 않는 것이 좋습니다. 참조: Transactions before being accepted by a CSMS CSMS가 BootNotificationResponse와 *Accepted* 상태를 반환하면 B01 - Cold Boot Charging Station이 적용됩니다.

B03 - Cold Boot Charging Station - Rejected - Requirements

Table 40. B03 - 요구 사항

ID	전제 조건	요구 사항 정의
B03.FR.01	충전기가 Transactions before being accepted by a CSMS를 허용하도록 구성된 경우	충전기는 로컬에서 인증된 트랜잭션을 허용할 수 있습니다.
B03.FR.02	CSMS가 *Rejected* 상태를 반환하는 경우. 예를 들어 충전기가 블랙리스트에 있는 경우.	충전기는 재시도 간격이 만료될 때까지 CSMS에 OCPP 메시지를 보내서는 안 됩니다.
B03.FR.03	CSMS가 충전기에서 BootNotificationRequest를 거부한 경우	CSMS는 어떠한 메시지도 시작해서는 안 됩니다.
B03.FR.04	B03.FR.03	충전기는 다음 BootNotificationRequest를 보내야 할 때까지 연결을 닫을 수 있습니다.
B03.FR.05	BootNotificationResponse의 간격이 0이고 상태가 *Accepted* 가 아닌 경우	충전기는 CSMS에 요청이 범람하지 않도록 자체적으로 대기 간격을 선택해야 합니다.
B03.FR.06	BootNotificationResponse의 간격이 0보다 크고 상태가 *Accepted* 가 아닌 경우	충전기는 설정된 간격이 지난 후 BootNotificationRequest를 보내야 합니다.
B03.FR.07	B03.FR.03 AND 충전기에서 BootNotificationRequest가 아닌 메시지를 보냅니다.	CSMS는 RPC 프레임워크로 응답해야 합니다: CALLERROR: SecurityError.
B03.FR.08	B03.FR.03 AND CSMS에서 충전기에서 BootNotificationRequest에 대한 응답이 아닌 메시지를 보냅니다.	충전기에서 RPC 프레임워크로 응답해야 합니다: CALLERROR: SecurityError.

B04 - Offline Behavior Idle Charging Station

Table 41. B04 - 오프라인 동작 유휴 충전기

번호	유형	설명
1	**이름**	오프라인 동작 유휴 충전기
2	**ID**	B04
	기능 블록	B. 프로비저닝
3	**목표**	충전기의 독립 실행형 작동을 달성합니다.
4	**설명**	이 사용 사례는 통신이 불가능한 경우 충전기가 독립 실행형으로 작동하도록 설계되었음을 설명합니다. 이 상황에서 충전기는 *Offline* 이라고 합니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS 또는 통신을 사용할 수 없습니다. **2.** 충전기는 단독으로 작동합니다. **3.** 연결이 복구되었습니다. **4.** *Offline* 기간이 OfflineThreshold 구성 변수의 값을 초과하는 경우: 충전기는 각 커넥터에 대해 StatusNotificationRequest를 CSMS로 보냅니다. 그렇지 않으면 오프라인 기간 동안 상태가 변경된 커넥터에 대해서만 StatusNotificationRequest를 보냅니다. **5.** 충전기는 HeartbeatRequest를 CSMS로 보냅니다. **6.** CSMS는 HeartbeatResponse로 응답합니다.
5	**필수 조건**	BootNotification이 이전에 수락되었고 충전기가 단독으로 작동할 수 있습니다.
6	**사후 조건**	*Offline* 동작 기간 후 연결이 복구되면 CSMS는 충전기와 EVSE의 상태를 알게 됩니다.

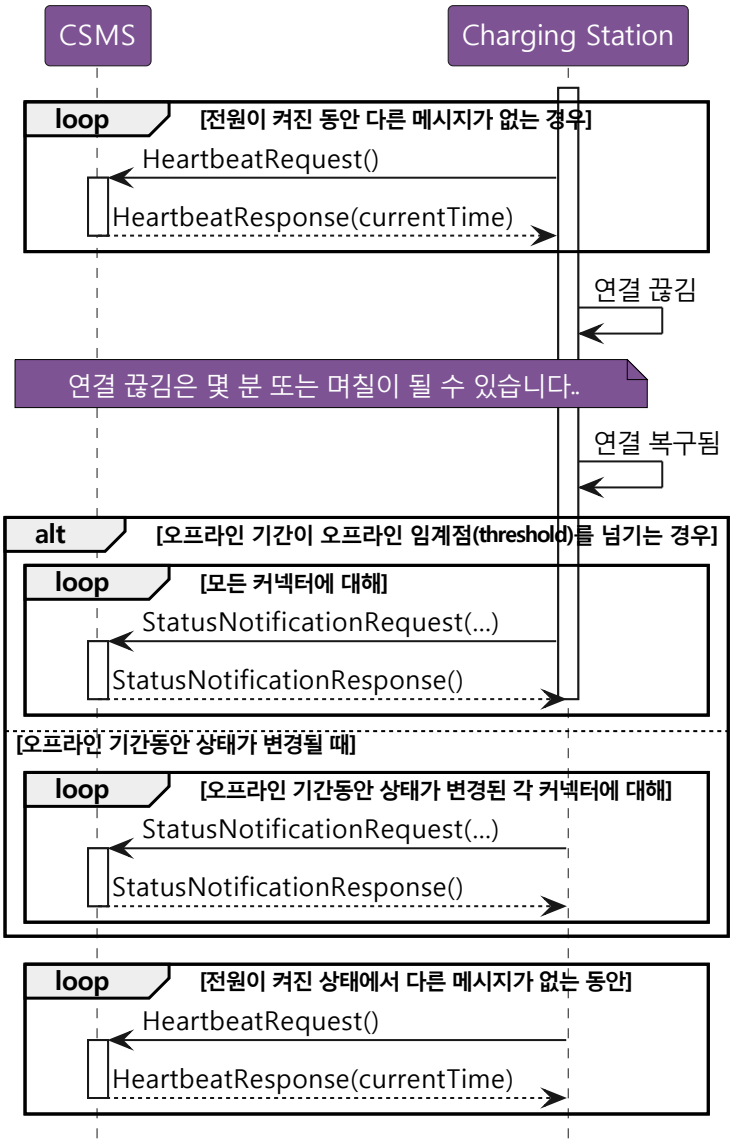

Figure 13. 시퀀스 다이어그램: 오프라인 동작 유휴 충전기

7	**오류 처리**	오프라인 상황은 충전기가 연결을 다시 설정하려고 시도하여 처리해야 하는 선호되지 않는 작동 모드입니다.
8	**비고**	n/a

B04 - Offline Behavior Idle Charging Station - Requirements

Table 42. B04 - 요구 사항

ID	전제 조건	요구 사항 정의
B04.FR.01	*Offline* 이 되고 *Offline* 기간이 OfflineThreshold 구성 변수의 값을 초과한 후.	충전기는 StatusNotificationRequest을 보내 모든 커넥터의 현재 상태를 보고해야 합니다.
B04.FR.02	*Offline* 이 되고 *Offline* 기간이 OfflineThreshold 구성 변수의 값을 초과하지 않은 후.	충전기는 StatusNotificationRequest를 보내서 상태 변경이 발생한 커넥터의 현재 상태만 보고해야 합니다.

2.2. Configuring a Charging Station

NOTE	충전기 구성을 관리하려면 장치 모델 개념에 대한 기본적인 이해가 필수적입니다. 이러한 개념은 "OCPP 2.0.1: Part 1 - Architecture & Topology", chapter 4에서 설명합니다.

B05 - Set Variables

Table 43. B05 - 변수 설정

번호	유형	설명
1	**이름**	변수 설정
2	**ID**	B05
	기능 블록	B. 프로비저닝
3	**목표**	CSMS가 충전기의 변수를 변경할 수 있는 기능을 제공합니다.
4	**설명**	충전기에는 CSMS가 구성/변경할 수 있는 많은 변수가 있을 수 있습니다. CSMS는 이러한 변수를 사용하여 예를 들어 충전기의 동작에 영향을 미칠 수 있습니다. 이 사용 사례는 CSMS가 충전기에 구성 요소의 변수 값을 설정하도록 요청하는 방법을 설명합니다. CSMS는 요청당 두 개 이상의 값을 설정하도록 요청할 수 있습니다.
	Actors	CSMS, 충전기
	시나리오 설명	**1.** CSO가 CSMS를 트리거하여 충전기에서 하나 이상의 변수를 설정하도록 요청합니다. **2.** CSMS가 충전기에 SetVariablesRequest를 보냅니다. **3.** 충전기는 SetVariablesResponse로 응답하여 변경을 실행할 수 있었는지 여부를 나타냅니다.
5	**필수 조건**	n/a
6	**사후 조건**	**성공한 사후 조건:** **1.** 변경이 *성공적으로* 실행되었습니다. **실패한 사후 조건:** **1.** 변수는 지원되지만 설정을 변경할 수 없으며 충전기는 *Rejected* 상태로 응답합니다. **2.** 변수는 *지원되지 않습니다*, 충전기는 *UnknownVariable* 상태로 응답합니다.

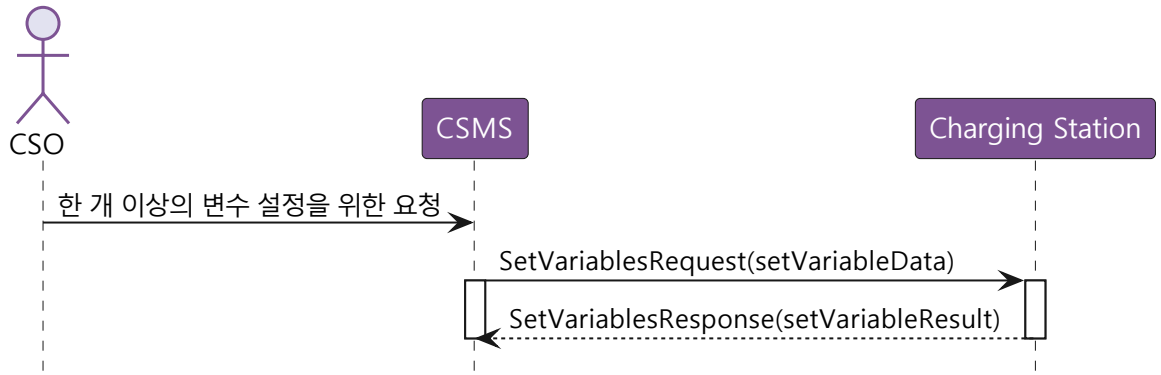

Figure 14. 시퀀스 다이어그램: 변수 설정

7	**오류 처리**	n/a

8	참고 사항	속성 유형 Actual은 변수의 실제 값에 해당하는 반면 속성 유형 Target, MinSet 및 MaxSet은 이 변수에 설정된 대상, 최소 및 최대 값에 해당합니다. 이는 예를 들어 가장 잘 설명할 수 있습니다. 냉각 시스템은 1000~5000rpm 사이의 팬 속도로 작동하도록 구성되어 있습니다. 이러한 경계는 MinSet 및 MaxSet 속성으로 표현됩니다. 현재 팬 속도는 실제 속성으로 표현됩니다. 원하는 팬 속도는 목표 속성으로 표현됩니다.

B05 - Set Variables - Requirements

Table 44. B05 - 요구 사항

ID	전제 조건	요구 사항 정의
B05.FR.01	충전기가 X개의 SetVariableData 요소가 있는 SetVariablesRequest를 수신할 때	충전기는 SetVariablesResponse로 응답해야 하며, SetVariablesRequest의 모든 SetVariableData 요소에 대해 동일한(X)개의 SetVariableResult 요소가 있어야 합니다.
B05.FR.02	B05.FR.01	SetVariablesResponse의 모든 SetVariableResult 요소에는 SetVariablesRequest의 SetVariableData 요소 중 하나와 동일한 구성 요소와 변수 조합이 포함되어야 합니다.
B05.FR.03	B05.FR.02 AND SetVariablesRequest에 *attributeType* 이 포함된 경우	SetVariablesResponse의 해당 SetVariableResult 요소에도 동일한 *attributeType* 이 포함되어야 합니다.
B05.FR.04	충전기가 SetVariablesRequest를 수신하고 SetVariableData에 알 수 없는 Component가 포함된 경우	충전기는 해당 SetVariableResult의 *attributeStatus* 필드를 UnknownComponent로 설정해야 합니다.
B05.FR.05	충전기가 SetVariablesRequest를 수신할 때, Component에 대해 알려지지 않은 변수가 SetVariableData에 있는 경우	충전기는 해당 SetVariableResult의 *attributeStatus* 필드를 UnknownVariable로 설정해야 합니다.
B05.FR.06	충전기가 SetVariablesRequest를 수신하고 attributeType이 SetVariableData의 주어진 Variable에 대해 알려지지 않은 경우	충전기는 해당 SetVariableResult의 *attributeStatus* 필드를 NotSupportedAttributeType으로 설정해야 합니다.
B05.FR.07	충전기가 SetVariablesRequest를 수신하고 SetVariableData의 주어진 Variable에 대해 잘못 포맷된 *value* 를 포함하는 경우	충전기는 해당 SetVariableResult의 *attributeStatus* 필드를 Rejected로 설정해야 합니다. (자세한 내용은 선택 사항인 *statusInfo* 요소에서 제공할 수 있습니다.)
B05.FR.08	충전기가 SetVariablesRequest를 수신하고 *value* 가 SetVariableData의 주어진 Variable의 범위보다 낮거나 높을 때	충전기는 해당 SetVariableResult의 *attributeStatus* 필드를 Rejected로 설정해야 합니다. (자세한 정보는 선택 사항인 *statusInfo* 요소에서 제공할 수 있습니다.)
B05.FR.09	NOT (B05.FR.04 to B05.FR.08) AND 충전기가 SetVariablesRequest를 SetVariableData의 Variable에 대해 수신했지만 설정할 수 없는 경우	충전기는 해당 SetVariableResult의 *attributeStatus* 필드를 Rejected로 설정해야 합니다. (변수가 *ReadOnly* 인 경우 발생하지만, 기술적 문제로 인해 변수 설정이 실패할 때도 발생할 수 있습니다.)
B05.FR.10	충전기가 SetVariableData에서 주어진 *value* 를 설정할 수 있는 경우	충전기는 해당 SetVariableResult의 *attributeStatus* 필드를 Accepted로 설정해야 합니다.
B05.FR.11		CSMS는 `ItemsPerMessageSetVariables`를 통해 충전기에서 보고한 것보다 더 많은 SetVariableData 요소를 SetVariablesRequest에 보내서는 안 됩니다.

ID	전제 조건	요구 사항 정의
B05.FR.12	충전기가 *attributeType* 없이 SetVariablesRequest를 수신하는 경우.	SetVariablesResponse의 해당 SetVariableResult 요소에는 *attributeType* Actual이 포함되어야 합니다.
B05.FR.13		CSMS는 동일한 Component, Variable 및 *AttributeType* 조합을 사용하여 단일 SetVariableData 요소를 여러 개 포함해서는 안 됩니다. 생략된 *AttributeType* 은 *Actual* 값으로 간주됩니다.

B06 - Get Variables

Table 45. B06 - 변수 가져오기

번호	유형	설명
1	**이름**	변수 가져오기
2	**ID**	B06
	기능 블록	B. 프로비저닝
3	**목표**	CSMS에 하나 이상의 구성 요소의 하나 이상의 변수에 대한 속성 값을 검색할 수 있는 기능을 제공합니다.
4	**설명**	이 사용 사례는 CSMS가 충전기에 하나 이상의 구성 요소의 하나 이상의 변수에 대한 속성 값을 보내도록 요청하는 방법을 설명합니다. 한 번의 호출로 모든 변수의 모든 속성을 가져오는 것은 불가능합니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSO가 CSMS를 트리거하여 충전기에서 여러 변수를 요청합니다. **2.** CSMS가 GetVariablesRequest와 요청된 변수 목록을 사용하여 충전기에 여러 변수를 요청합니다. **3.** 충전기가 GetVariablesResponse와 함께 요청된 변수로 응답합니다. **4.** CSMS가 CSO에 선택적 알림을 보냅니다.
5	**필수 조건**	n/a
6	**사후 조건**	**성공한 사후 조건:** 충전기가 요청된 모든 변수를 보낼 수 있었습니다. **실패한 사후 조건:** 충전기가 요청된 모든 변수를 보낼 수 없었습니다.

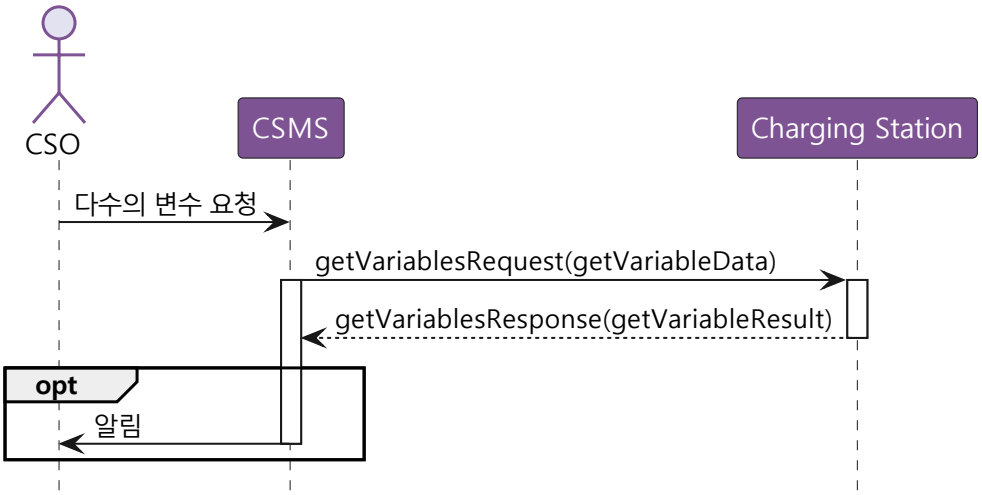

Figure 15. 시퀀스 다이어그램: 변수 가져오기

7	**오류 처리**	n/a
8	**참고**	n/a

B06 - Get Variables - Requirements

Table 46. B06 - 요구 사항

ID	전제 조건	요구 사항 정의
B06.FR.01	충전기가 X개의 GetVariableData 요소가 포함된 GetVariablesRequest를 수신할 때	충전기는 GetVariablesResponse로 응답해야 하며, X개의 GetVariableResult 요소가 동일해야 하며, GetVariablesRequest의 모든 GetVariableData 요소에 대해 하나씩 응답해야 합니다.
B06.FR.02	B06.FR.01	GetVariablesResponse의 모든 GetVariableResult 요소는 GetVariablesRequest의 GetVariableData 요소 중 하나와 동일한 *component* 및 *variable* 조합을 포함해야 합니다.
B06.FR.03	B06.FR.02 AND GetVariablesRequest에 *attributeType* 이 포함된 경우	GetVariablesResponse의 해당 GetVariableResult 요소에도 동일한 *attributeType* 이 포함되어야 합니다.
B06.FR.04	B06.FR.01	GetVariablesResponse의 모든 GetVariableResult 요소에는 GetVariablesRequest의 요청된 *attributeType* 의 속성 값이 있는 *attributeValue* 가 포함되어야 합니다.
B06.FR.05		CSMS는 충전기에서 `ItemsPerMessageGetVariables`를 통해 보고한 것보다 더 많은 GetVariableData 요소를 GetVariablesRequest에 보내서는 안 됩니다.
B06.FR.06	충전기가 GetVariableData에 알 수 없는 Component가 포함된 GetVariablesRequest를 수신하는 경우	충전기는 해당 GetVariableResult의 *attributeStatus* 필드를 UnknownComponent로 설정하고 *attributeValue* 를 생략해야 합니다.
B06.FR.07	충전기가 GetVariablesRequest를 수신하고 Component에 대해 알려지지 않은 GetVariableData 에 있는 GetVariablesRequest를 수신하는 경우	충전기는 해당 GetVariableResult의 *attributeStatus* 필드를 UnknownVariable로 설정하고 *attributeValue* 를 생략해야 합니다.
B06.FR.08	충전기가 GetVariablesRequest를 수신하고 attributeType이 GetVariableData의 주어진 Variable에 대해 알려지지 않은 경우	충전기는 해당 GetVariableResult의 *attributeStatus* 필드를 NotSupportedAttributeType으로 설정하고 *attributeValue* 를 생략해야 합니다.
B06.FR.09	충전기가 GetVariablesRequest를 수신하면 GetVariableData의 Variable이 *WriteOnly* 인 경우	충전기는 해당 GetVariableResult의 *attributeStatus* 필드를 Rejected로 설정해야 합니다.
B06.FR.10	충전기가 GetVariablesRequest에서 요청한 *value* 를 가져올 수 있는 경우	충전기는 해당 GetVariableResult의 *attributeStatus* 필드를 Accepted로 설정하고 *attributeValue* 를 찾은 값으로 설정해야 합니다.
B06.FR.11	충전기가 *attributeType* 없이 GetVariablesRequest를 수신하는 경우.	GetVariablesResponse의 해당 GetVariableResult 요소에는 *attributeType* Actual이 포함되어야 합니다.
B06.FR.13	NOT B06.FR.08 AND 충전기에 요청된 구성 요소 변수의 *attributeType* 에 대한 *attributeValue* 가 없습니다.	충전기는 *attributeValue* 로 빈 문자열을 반환해야 합니다. 참고: 예를 들어 *attributeType* `Target`이 지원되더라도 아직 설정되지 않은 경우 발생할 수 있습니다.

ID	전제 조건	요구 사항 정의
B06.FR.14	B06.FR.01 AND *instance* 에 대한 값이 GetVariableData의 *component* 및/또는 *variable* 에 제공됩니다.	충전기는 GetVariableResult에서 해당 구성 요소 및/또는 변수의 지정된 인스턴스를 반환해야 합니다.
B06.FR.15	B06.FR.01 AND GetVariableData의 *component* 및/또는 *variable* 에 대한 *instance* 에 값이 제공되지 않거나 빈 문자열이 제공되지 않음 AND *instance* 가 없는 구성 요소 및/또는 변수가 존재하지 않음	충전기는 GetVariableResult 항목에서 GetVariableData에 대한 *attributeStatus* `UnknownComponent` 또는 `UnknownVariable` 을 반환해야 합니다.
B06.FR.16	충전기가 `ItemsPerMessageGetVariables`에서 허용하는 것보다 더 많은 GetVariableData 요소가 포함된 GetVariablesRequest를 수신합니다.	충전기는 CALLERROR(OccurenceConstraintViolation)로 응답할 수 있습니다.
B06.FR.17	충전기가 `BytesPerMessageGetVariables`에서 허용하는 것보다 더 많은 바이트 길이의 GetVariablesRequest를 수신합니다.	충전기는 CALLERROR(FormatViolation)로 응답할 수 있습니다.

B07 - Get Base Report

Table 47. B07 - 기본 보고서 가져오기

번호	유형	설명
1	**이름**	기본 보고서 가져오기
2	**ID**	B07
	기능 블록	B. 프로비저닝
3	**목표**	CSMS가 ReportBase에 정의된 대로 미리 정의된 보고서를 요청할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSMS가 충전기에 ReportBase에 정의된 대로 미리 정의된 보고서를 보내도록 요청하는 방법을 설명합니다. 결과는 하나 이상의 NotifyReportRequest 메시지에서 비동기적으로 반환됩니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSO가 CSMS를 트리거하여 충전기에 보고서를 요청합니다. **2.** CSMS가 GetBaseReportRequest로 충전기에 보고서를 요청합니다. **3.** 충전기가 GetBaseReportResponse로 응답합니다. **4.** 충전기가 비동기적으로 하나 이상의 NotifyReportRequest 메시지로 결과를 보냅니다. **5.** CSMS가 NotifyReportRequest마다 NotifyReportResponse로 응답합니다.
5	**필수 조건**	n/a
6	**사후 조건**	**성공한 사후 조건:** 충전기에서 요청한 보고서를 보낼 수 있었습니다. **실패한 사후 조건:** 충전기에서 요청한 보고서를 보낼 수 *없었습니다*.

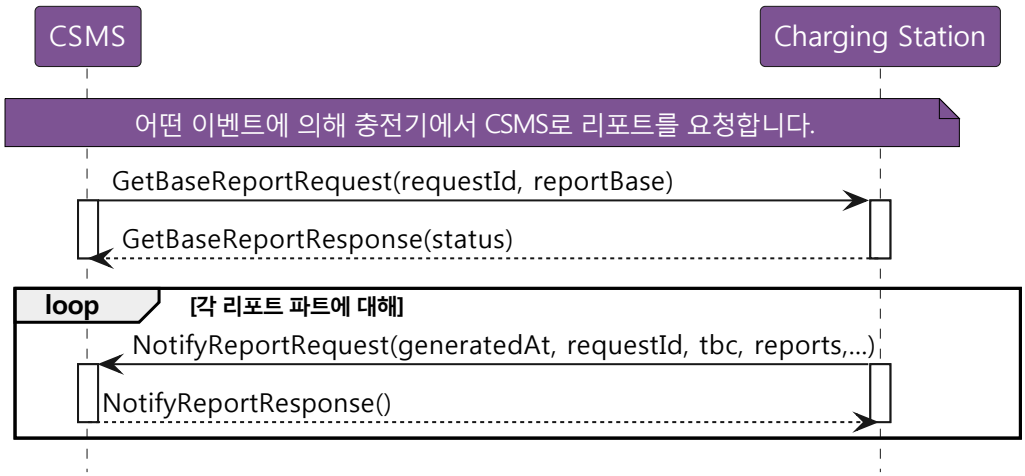

Figure 16. 시퀀스 다이어그램: 기본 보고서 가져오기

7	**오류 처리**	n/a
8	**비고**	n/a

B07 - Get Base Report - Requirements

Table 48. B07 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
B07.FR.01	충전기가 지원되는 *reportBase* 에 대한 getBaseReportRequest를 수신하고 B07.FR.13이 아닌 경우	충전기는 `Accepted` 와 함께 getBaseReportResponse를 보내야 합니다.	
B07.FR.02	충전기가 지원되지 않는 *reportBase* 에 대한 getBaseReportRequest를 수신하는 경우	충전기는 `NotSupported` 로 getBaseReportResponse를 보내야 합니다.	
B07.FR.03	B07.FR.01	충전기는 하나 이상의 NotifyReportRequest 메시지를 통해 요청된 정보를 CSMS로 보내야 합니다.	
B07.FR.04	B07.FR.01 AND getBaseReportRequest에는 *requestId* 가 포함되어 있습니다.	이 getBaseReportRequest에 대한 모든 NotifyReportRequest 전송에는 동일한 *requestId* 가 포함되어야 합니다.	
B07.FR.05	B07.FR.02	충전기는 CSMS에 NotifyReportRequest를 보내서는 안 됩니다.	
B07.FR.07	B07.FR.01 및 *reportBase* 가 ConfigurationInventory인 경우	그러면 충전기는 NotifyReportRequest로 응답하여 운영자가 설정할 수 있는 모든 구성 요소 변수(*VariableCharacteristics* 포함)를 보고해야 합니다.	
B07.FR.08	B07.FR.01 및 *reportBase* 가 FullInventory인 경우	그러면 충전기는 NotifyReportRequest로 응답하여 모든 구성 요소 변수(*VariableCharacteristics* 포함)를 보고해야 합니다.	최소한 Charging Infrastructure related에 언급된 필수 변수는 보고되어야 하며, 구현된 각 기능 블록과 관련된 Section 1 Controller Components에 있는 필수 변수도 보고되어야 합니다.
B07.FR.09	B07.FR.01 AND *reportBase* 가 SummaryInventory인 경우	그러면 충전기는 NotifyReportRequest로 응답하여 충전기의 가용성 및 상태와 관련된 구성 요소 및 변수, 특히 충전기, EVSE 및 커넥터의 운영 상태와 오류 상태를 보고해야 합니다.	충전기의 현재 충전 가용성과 기존 문제 상태와 관련된 구성 요소/변수를 나열하는 (요약) 보고서입니다. 충전기 구성 요소의 경우: - AvailabilityState. 각 EVSE 구성 요소의 경우: - AvailabilityState. 각 커넥터 구성 요소의 경우: - AvailabilityState(알려져 있고 EVSE와 다른 경우). 비정상 상태의 모든 구성 요소의 경우: - Active (Problem, Tripped, Overload, Fallback) 변수. - 구성 요소의 다른 진단 관련 변수.

ID	전제 조건	요구 사항 정의	참고
B07.FR.10		NotifyReportRequest의 seqNo 필드에 포함된 시퀀스 번호는 보고서당 증가합니다. 따라서 첫 번째 보고서 부분을 포함하는 NotifyReportRequest 메시지는 값이 *0* 인 seqNo를 가져야 합니다.	
B07.FR.11	B07.FR.08	충전기에서 지원하는 변수의 모든 속성 유형은 값이 없더라도(설정되지 않은 경우에도) 보고되어야 합니다.	이를 통해 CSMS는 충전기에서 지원하는 속성 유형을 알 수 있습니다.
B07.FR.12		충전기는 최소한 기본 보고서인 ConfigurationInventory 및 FullInventory를 지원해야 합니다.	
B07.FR.13	충전기가 일시적으로 보고서 요청을 실행할 수 없는 경우	충전기는 `Rejected` 와 함께 getBaseReportResponse를 보내야 합니다.	
B07.FR.14	충전기가 처음으로 CSMS에 연결되거나 CSMS가 충전기의 기기 모델이 변경되었을 것으로 의심할 때마다(예: 펌웨어 업데이트 또는 하드웨어 변경 후)	CSMS는 *reportBase* = `FullInventory` 로 GetBaseReportRequest를 요청하여 모든 기기 모델 구성 요소와 변수의 전체 목록을 검색해야 합니다.	고정된 기기 모델이 변경되지 않는 알려진 충전기 세트를 기반으로 하는 구현이 있을 수 있으므로 의무화되지 않습니다.

B08 - Get Custom Report

Table 49. B08 - 사용자 정의 보고서 가져오기

번호	유형	설명
1	**이름**	사용자 정의 보고서 가져오기
2	**ID**	B08
	기능 블록	B. 프로비저닝
3	**목표**	CSMS에 ComponentCriteria 및/또는 ComponentVariables 목록과 일치하는 모든 구성 요소 및 변수에 대한 보고서를 요청할 수 있는 기능을 제공합니다.
4	**설명**	이 사용 사례는 CSMS가 충전기에 ComponentCriteria 및/또는 ComponentVariables 목록과 일치하는 모든 구성 요소 및 변수에 대한 보고서를 보내도록 요청하는 방법을 설명합니다. 결과는 하나 이상의 NotifyReportRequest 메시지로 비동기적으로 반환됩니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSO가 CSMS를 트리거하여 충전기에 보고서를 요청합니다. **2.** CSMS가 GetReportRequest로 충전기에 보고서를 요청합니다. **3.** 충전기가 GetReportResponse로 응답합니다. **4.** 충전기가 하나 이상의 NotifyReportRequest 메시지로 비동기적으로 결과를 보냅니다. **5.** CSMS가 NotifyReportResponse로 응답합니다.
5	**필수 조건**	n/a

번호	유형	설명
6	사후 조건	**성공한 사후 조건:** 충전기에서 요청한 보고서를 보낼 수 있었습니다. **실패한 사후 조건:** 충전기에서 요청한 보고서를 보낼 수 *없었습니다*.

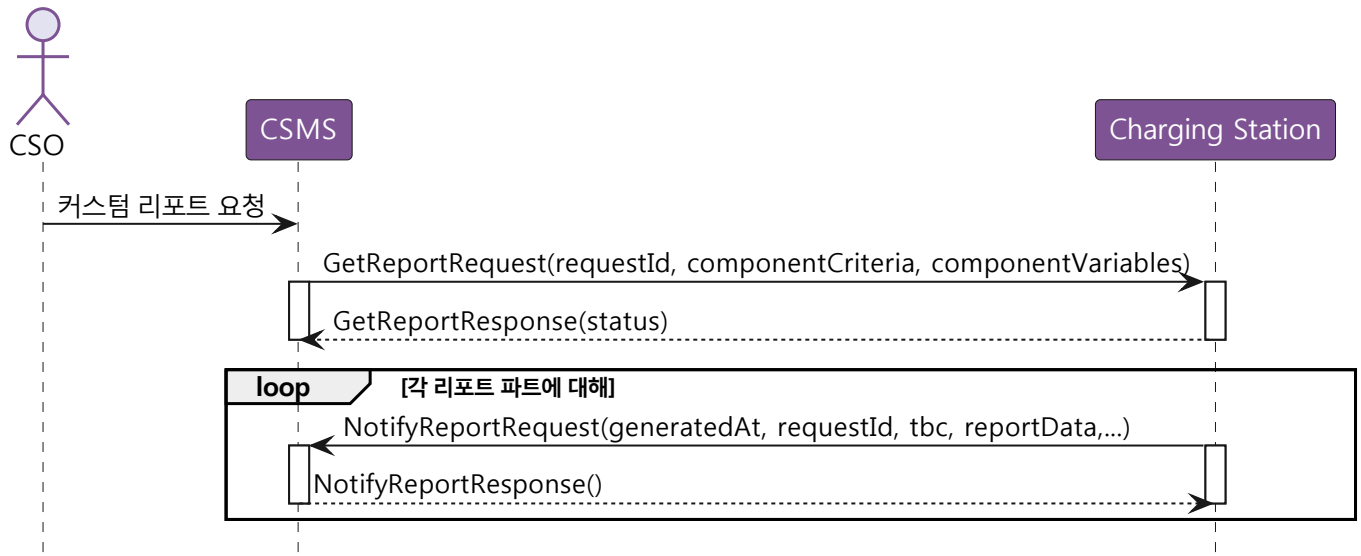

Figure 17. 시퀀스 다이어그램: 사용자 지정 보고서 가져오기

7	오류 처리	n/a
8	비고	n/a

B08 - Get Custom Report - Requirements

Table 50. B08 - 요구 사항

ID	전제 조건	요구 사항 정의
B08.FR.01	NOT B08.FR.15 AND 충전기가 지원되는 *기준(criteria)* 에 대한 getReportRequest를 수신하는 경우	충전기는 Accepted와 함께 getReportResponse를 보내야 합니다.
B08.FR.02	충전기가 지원되지 않는 *기준(criteria)* 에 대한 getReportRequest를 수신하는 경우	충전기는 getReportResponse와 함께 NotSupported
B08.FR.03	B08.FR.01	충전기는 요청된 정보를 하나 이상의 NotifyReportRequest 메시지를 통해 CSMS로 전송해야 합니다.
B08.FR.04	B08.FR.01 및 getReportRequest에는 *requestId* 가 포함되어 있습니다.	이 getReportRequest에 대해 전송된 모든 NotifyReportRequest에는 동일한 *requestId* 가 포함되어야 합니다.
B08.FR.05	B08.FR.01 및 *componentCriteria* 와 *componentVariables* 가 모두 비어 있지 않습니다.	이 getReportRequest에 대해 전송된 모든 NotifyReportRequest는 *componentCriteria* 및 *componentVariables* 집합으로 제한되어야 합니다.

ID	전제 조건	요구 사항 정의
B08.FR.06		한 getReportRequest 메시지에서 *componentVariables* 의 최대 수는 `ItemsPerMessageGetReport` 구성 변수에서 제공됩니다.
B08.FR.07	B08.FR.01 AND *ComponentCriteria* 에는 다음이 포함됩니다. *Active*	충전기는 변수 *Active* 가 *true* 로 설정된 모든 구성 요소 또는 NotifyReportRequest에서 *Active* 변수가 없는 모든 구성 요소를 보고해야 합니다.
B08.FR.08	B08.FR.01 AND *ComponentCriteria* 에는 다음이 포함됩니다. *Available*	충전기는 변수 *Available* 이 *true* 로 설정된 모든 구성 요소 또는 *Available* 변수가 없는 모든 구성 요소를 NotifyReportRequest에 보고해야 합니다.
B08.FR.09	B08.FR.01 AND *ComponentCriteria* 에는 다음이 포함됩니다. *Enabled*	충전기는 변수 *Enabled* 가 *true* 로 설정된 모든 구성 요소 또는 *Enabled* 변수가 없는 모든 구성 요소를 NotifyReportRequest에 보고해야 합니다.
B08.FR.10	B08.FR.01 AND *ComponentCriteria* 에는 다음이 포함됩니다. *Problem*	충전기는 NotifyReportRequest에서 변수 *Problem* 이 *true* 로 설정된 모든 구성 요소를 보고해야 합니다.
B08.FR.11	B08.FR.01 AND *componentCriteria* 가 없고 AND *componentVariables* 가 비어 있지 않습니다.	이 getReportRequest에 대해 전송된 모든 NotifyReportRequest는 *componentVariables* 의 집합으로 제한됩니다.
B08.FR.12	B08.FR.01	NotifyReportRequest에서 보고된 변수는 *variableCharacteristics* 를 포함해야 합니다.
B08.FR.13	B08.FR.01 AND 두 개 이상의 *componentCriteria* 가 제공됩니다.	충전기는 제공된 기준 중 하나 이상을 갖는 모든 구성 요소를 보고해야 합니다(논리적 OR).
B08.FR.14		NotifyReportRequest의 seqNo 필드에 포함된 시퀀스 번호는 보고서마다 증가합니다. 따라서 첫 번째 보고서 부분을 포함하는 NotifyReportRequest 메시지는 값이 *0* 인 seqNo를 가져야 합니다.
B08.FR.15	충전기가 기준 조합으로 GetReportRequest를 수신하여 빈 결과 집합이 생성되는 경우.	충전기는 GetReportResponse(*status*=`EmptyResultSet`)로 응답해야 합니다.
B08.FR.16	충전기가 일시적으로 보고서 요청을 실행할 수 없는 경우	충전기는 `Rejected` 로 getBaseReportResponse를 보내야 합니다.
B08.FR.17	충전기가 `ItemsPerMessageGetReport`에서 허용하는 것보다 더 많은 ComponentVariableType 요소가 포함된 GetReportRequest를 수신합니다.	충전기는 CALLERROR(OccurenceConstraintViolation)로 응답할 수 있습니다.
B08.FR.18	충전기가 `BytesPerMessageGetReport`에서 허용하는 것보다 더 많은 바이트 길이의 GetReportRequest를 수신합니다.	충전기는 CALLERROR(FormatViolation)로 응답할 수 있습니다.
B08.FR.20	충전기가 *variable* 이 누락된 *componentVariable* 요소가 포함된 GetReportRequest를 수신하는 경우	충전기는 *componentVariable* 의 *component* 의 모든 *variable* 에 대해 보고해야 합니다.
B08.FR.21	충전기가 *componentVariable* 요소가 있는 GetReportRequest를 수신하고 *variable* 은 있지만 *instance* 가 없는 경우	충전기는 *componentVariable* 의 *component* 의 *variable* 의 모든 인스턴스에 대해 보고해야 합니다.

ID	전제 조건	요구 사항 정의
B08.FR.22	B08.FR.11 AND 충전기가 *component.evse.id* 가 있는 *componentVariable* 요소에 *component* 가 포함된 GetReportRequest를 수신하지만 *component.evse.connector* 가 없는 경우	충전기는 모든 *component.evse.connector* 에 대해 이 *component.name*, *component.instance* 및 *component.evse.id* 가 있는 구성 요소를 보고해야 하며, 이때 B08.FR.24를 고려해야 합니다.
B08.FR.23	B08.FR.11 AND 충전기가 *component.evse.id* 가 없는 *componentVariable* 요소에 *component* 가 포함된 GetReportRequest를 수신하는 경우	충전기는 *component.evse* 필드(*component.evse* 가 없는 최상위 구성 요소 포함)에 대해 이 *component.name*, *component.instance* 가 있는 구성 요소를 보고해야 하며, B08.FR.24를 고려해야 합니다.
B08.FR.24	B08.FR.11 AND 충전기가 *component.instance* 에 대한 값이 있는 *componentVariable* 요소에 *component* 가 포함된 GetReportRequest를 수신하는 경우	충전기는 B08.FR.22, B08.FR.23을 고려하면서 모든 *component.instance* 필드에 대해 이 *component.name* 이 포함된 구성 요소를 보고해야 합니다.
B08.FR.25	B08.FR.11 AND 충전기가 *component.instance* 필드가 없는 *componentVariable* 요소에 *component* 가 포함된 GetReportRequest를 수신하는 경우	충전기는 모든 *component.instance* 필드에 대해 이 *component.name* 이 있는 구성 요소 또는 *component.instance* 필드가 없는 구성 요소를 보고해야 하며, B08.FR.22, B08.FR.23을 고려합니다.

B09 - Setting a new NetworkConnectionProfile

Table 51. B09 - 새 NetworkConnectionProfile 설정

번호	유형	설명
1	**이름**	새 NetworkConnectionProfile 설정.
2	**ID**	B09
	기능 블록	B. 프로비저닝
3	**목표**	CSMS가 충전기의 연결 세부 정보를 업데이트할 수 있도록 합니다.
4	**설명**	CSMS가 충전기의 연결 세부 정보를 업데이트합니다. 예를 들어 새 CSMS로 마이그레이션을 준비할 때입니다. 이 사용 사례를 완료한 후 충전기에서 CSMS로의 연결 데이터가 업데이트되었습니다.
	Actors	충전기, CSMS
	시나리오 설명	충전기는 숫자로 식별되는 최소 두 개의 네트워크 구성 슬롯을 지원합니다. 사용 가능한 슬롯 번호는 충전기에서 변수 NetworkConfigurationPriority의 *valuesList* 에 보고합니다. 예를 들어: 기본 보고서의 valuesList "0,1,2"는 CSMS에 0, 1, 2로 번호가 매겨진 세 개의 구성 슬롯을 사용할 수 있음을 알려줍니다(반드시 설정되지는 않음). 활성 연결에 사용되는 구성 슬롯 번호는 변수 OCPPCommCtrlr.ActiveNetworkProfile에 의해 보고됩니다. **1.** CSMS는 업데이트된 연결 프로필과 NetworkConfigurationPriority의 *valuesList* 에서 *configurationSlot* 을 포함하는 SetNetworkProfileRequest PDU를 보냅니다. **2.** 충전기는 PDU를 수신하고, 콘텐츠를 검증하고, 새 데이터를 저장합니다. **3.** 충전기는 *Accepted* 상태로 SetNetworkProfileResponse PDU를 보내 응답합니다.
5	**필수 조건**	CSMS에서 제공한 데이터가 충전기의 기능과 일치합니다.
6	**사후 조건**	충전기가 새 연결 데이터를 저장할 수 있었습니다.

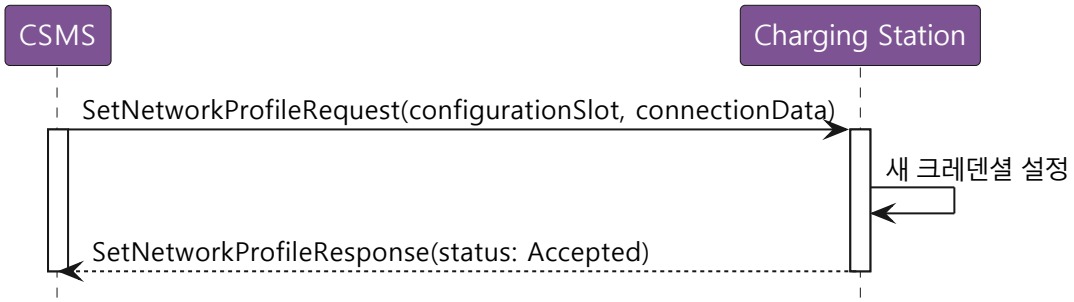

Figure 18. 시퀀스 다이어그램: 네트워크 연결 프로필 설정

8	**오류 처리**	새 NetworkConnectionProfile의 활성화는 B10 - Migrate to new CSMS. 이 사용 사례 중 오류는 현재 데이터 연결에 파괴적이지 않습니다. 오류 처리에 대한 자세한 설명은 B10 - Migrate to new CSMS
9	**참고**	현재 활성화된 NetworkConnectionProfile에 변경 사항이 적용되더라도 B10 - Migrate to new CSMS에서 설명한 대로 재부팅이 발생할 때까지는 활성화되지 않습니다.

B09 - Setting a new NetworkConnectionProfile - Requirements

Table 52. B09 - 요구 사항

ID	전제 조건	요구 사항 정의
B09.FR.01	SetNetworkProfileRequest를 수신하면	충전기는 콘텐츠를 검증하고, 새 데이터를 저장하고, 성공하면 SetNetworkProfileResponse 메시지를 보내 응답해야 합니다. 메시지의 상태는 *Accepted* 입니다.
B09.FR.02	SetNetworkProfileRequest를 수신하면	충전기는 콘텐츠를 검증해야 합니다. 콘텐츠가 유효하지 않은 경우 충전기는 SetNetworkProfileResponse 메시지를 보내서 응답해야 하며, 메시지의 상태는 *Rejected* 입니다.
B09.FR.03	새로운 네트워크 프로필 설정이 실패하는 경우입니다.	충전기는 SetNetworkProfileResponse 메시지를 보내서 응답해야 합니다. 메시지의 상태는 *Failed* 입니다.
B09.FR.04	SetNetworkProfileRequest를 수신하고 NetworkConnectionProfile에 구성 변수 SecurityProfile에 저장된 것보다 낮은 securityProfile이 포함되어 있는 경우	충전기는 SetNetworkProfileResponse 메시지를 보내서 응답해야 합니다. 메시지의 상태는 *Rejected* 입니다.
B09.FR.05	SetNetworkProfileRequest의 *configurationSlot* 값이 NetworkConfigurationPriority의 valuesList에 있는 항목과 일치하지 않는 경우	충전기는 SetNetworkProfileResponse 메시지를 *Rejected* 상태로 보내 응답합니다.
B09.FR.06		충전기는 네트워크 연결 프로필에 대해 최소 두 개의 구성 슬롯을 지원해야 합니다.

B10 - Migrate to new CSMS

Table 53. B10 - 새 CSMS로 마이그레이션

번호	유형	설명
1	**이름**	다른 NetworkConnectionProfile을 사용하여 새 CSMS로 마이그레이션합니다.
2	**ID**	B10
	기능 블록	B. 프로비저닝
3	**목표**	이 사용 사례를 완료한 후 충전기는 새 CSMS에 연결합니다.
4	**설명**	이 사용 사례는 `NetworkConfigurationPriority`에서 NetworkConnectionProfiles의 순서를 변경하여 충전기에 새 CSMS에 연결하도록 지시하는 방법을 설명합니다.
	Actors	충전기, CSMS 1, CSMS 2
	시나리오 설명	충전기는 숫자로 식별되는 최소 두 개의 네트워크 구성 슬롯을 지원합니다. 사용 가능한 슬롯 번호는 충전기에서 변수 NetworkConfigurationPriority의 *valuesList* 에 보고합니다. 예를 들어: 기본 보고서의 *valuesList* = "0,1,2"는 CSMS에 0, 1, 2로 번호가 매겨진 세 개의 구성 슬롯을 사용할 수 있음을 알려줍니다(하지만 반드시 설정되지는 않음). NetworkConfigurationPriority의 *value* 는 네트워크 구성이 연결을 시도하는 순서를 보고합니다. 예를 들어: 값 "1,0"은 충전기가 먼저 구성 슬롯 1을 시도하고 NetworkProfileConnectionAttempts에 구성된 시도 횟수 이후에 실패하면 구성 슬롯 0에 연결을 시도한다는 것을 의미합니다. **1.** CSMS 1은 SetVariablesRequest를 통해 `NetworkConfigurationPriority` 구성 변수에 대한 새 값을 설정하여 CSMS 2의 NetworkConnectionProfile이 목록에서 첫 번째가 되고 CSMS 1에 대한 기존 연결이 목록에서 두 번째가 되도록 합니다. **2.** 충전기는 *Accepted* 상태의 SetVariablesResponse로 응답합니다. **3.** CSMS 1은 충전기에 `OnIdle` 재설정을 수행하도록 지시합니다. **4.** 충전기가 재부팅되고 새로운 기본 NetworkConnectionProfile을 통해 CSMS 2에 연결됩니다.
5	**필수 조건**	사용 사례 B09 - Setting a new NetworkConnectionProfile이 이 사용 사례 이전에 성공적으로 실행되었습니다. CSMS에서 제공한 데이터가 충전기의 기능과 일치합니다.
6	**사후 조건**	충전기가 다른 NetworkConnectionProfiles를 통해 연결됩니다.

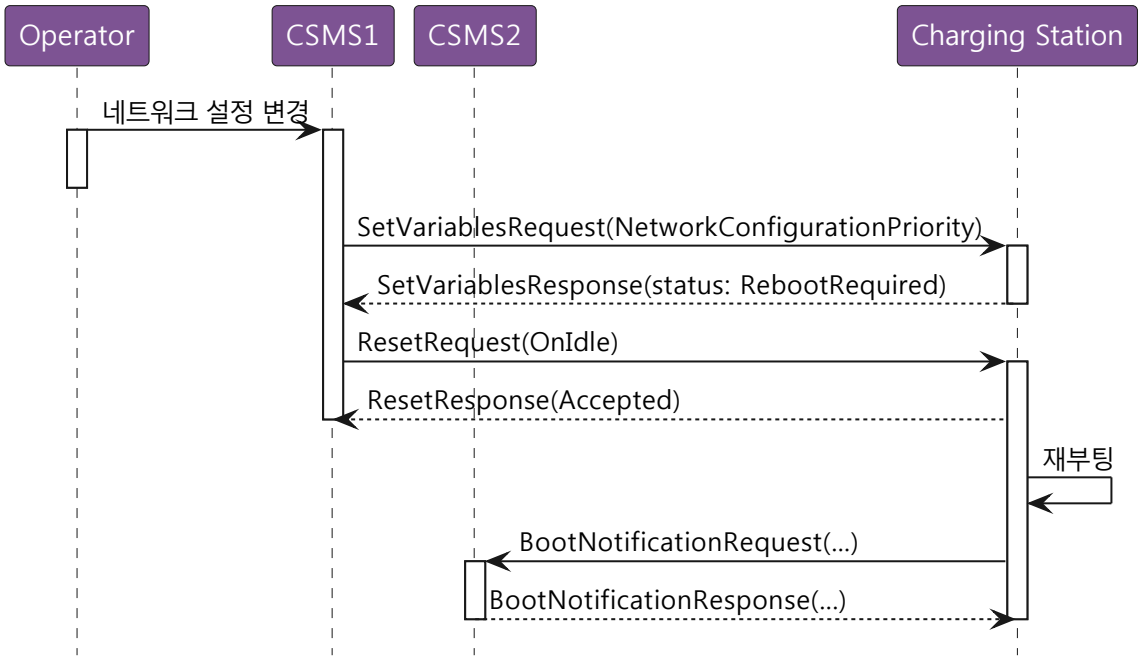

Figure 19. 시퀀스 다이어그램: 새 ConnectionProfile로 마이그레이션

7	**오류 처리**	n/a
8	**참고**	B12 - Reset - With Ongoing Transaction과 마찬가지로 진행 중인 트랜잭션이 있는 경우 충전기는 이러한 트랜잭션이 완료될 때까지 기다렸다가 Reset을 수행한 다음 다른 CSMS에 연결합니다. 운영자가 즉시 전환을 수행하려면 먼저 트랜잭션을 중지해야 합니다.

B10 - Migrate to new NetworkConnectionProfile - Requirements

Table 54. B10 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
B10.FR.01	구성 변수 `NetworkConfigurationPriority`를 포함하는 SetVariablesRequest를 수신하고 메시지의 NetworkProfile 슬롯에 모두 유효한 구성이 포함된 경우	충전기는 SetVariablesResponse를 *Accepted* 또는 *RebootRequired* 상태로 전송해야 합니다.	
B10.FR.02	SetVariablesRequest를 수신하고 구성 변수 `NetworkConfigurationPriority`를 포함하고 메시지의 NetworkProfile 슬롯 중 하나에 유효한 구성이 없는 경우	충전기는 SetVariablesResponse를 *Rejected* 상태로 전송해야 합니다.	선택적 요소 *statusInfo* 를 사용하여 추가 정보를 제공할 수 있습니다.
B10.FR.03	B10.FR.04 AND 연결에 실패하는 경우	충전기는 `NetworkConfigurationPriority`의 항목당 `NetworkProfileConnectionAttempts`에 구성된 횟수만큼 시도해야 합니다.	
B10.FR.04	B10.FR.01 또는 B09.FR.01 AND 재부팅 후	충전기는 `NetworkConfigurationPriority`의 첫 번째 항목에 연결을 시작해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
B10.FR.05		충전기를 Inoperative으로 설정하는 것이 좋습니다(ChangeAvailabilityRequest를 통해). 이렇게 하면 새 트랜잭션을 시작할 수 없고, ResetRequest를 보내기 전에 충전기의 트랜잭션 메시지 대기열이 비어 있을 때까지 기다릴 수 있습니다. 그렇지 않으면 충전기가 트랜잭션 시작을 받지 못한 새 CSMS에 트랜잭션 관련 메시지를 보낼 수 있으며, 이전 시스템은 종료된 메시지를 놓치게 됩니다. 대기열에 진행 중인 트랜잭션에 대한 트랜잭션이 아직 있는지 확인하려면 getTransactionStatusRequest 메시지를 사용할 수 있습니다.	
B10.FR.06		충전기는 새 CSMS에 연결을 시도하기 전에 이전 CSMS에서 연결을 끊어야 합니다.	
B10.FR.07	B10.FR.03 AND `NetworkConfigurationPriority`의 모든 항목에 대한 `NetworkProfileConnectionAttempts`가 모두 실패했습니다.	충전기는 폴백하여 마지막으로 성공적으로 연결한 NetworkConnectionProfile에 'reconnecting'을 시작해야 합니다.	이 요구 사항에서 'reconnecting'은 "Part 4 - JSON over WebSockets implementation guide"의 Section 5.3에서 설명한 재연결 메커니즘을 말합니다.

2.3. Resetting a Charging Station

B11 - Reset - Without Ongoing Transaction

Table 55. B11 - 재설정 - 진행 중인 트랜잭션 없음

번호	유형	설명
1	**이름**	재설정 - 진행 중인 트랜잭션 없음
2	**ID**	B11
	기능 블록	B. 프로비저닝
3	**목표**	CSMS가 진행 중인 트랜잭션이 없는 동안 충전기에 자체 또는 EVSE를 재설정하도록 요청할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSMS가 ResetRequest를 보내 충전기에 자체 또는 EVSE를 재설정하도록 요청하는 방법을 다룹니다. (ResetRequest에 선택적 매개변수 *evseId* 가 포함된 경우 특정 EVSE의 재설정만 요청됩니다.) 예를 들어 충전기가 제대로 작동하지 않는 경우 필요할 수 있습니다.
	Actors	충전기, CSMS, CSO
	시나리오 설명	**1.** CSO가 CSMS에 충전기 또는 EVSE를 재설정하도록 요청합니다. **2.** CSMS가 ResetRequest를 보내 충전기가 자체 또는 EVSE를 재설정하도록 요청합니다. **3.** CSMS가 OnIdle 또는 Immediate 재설정을 요청합니다. **4.** 충전기가 ResetResponse로 응답하여 충전기가 자체 또는 EVSE를 재설정할 수 있는지 여부를 나타냅니다. **5.** CSMS가 CSO에 선택적 알림을 보냅니다. **6.** evseId가 제공되지 않은 경우에만 재설정 후 충전기는 사용 사례 B01과 같이 진행됩니다.

번호	유형	설명
	대체 시나리오	B12 - Reset With Ongoing Transaction
5	**필수 조건**	진행 중인 트랜잭션이 없습니다.
6	**사후 조건**	**성공적인 사후 조건:** 충전기가 자체 또는 EVSE를 재설정할 수 있었습니다. **실패 사후 조건:** 충전기가 자체 또는 EVSE를 재설정할 수 *없었습니다*.

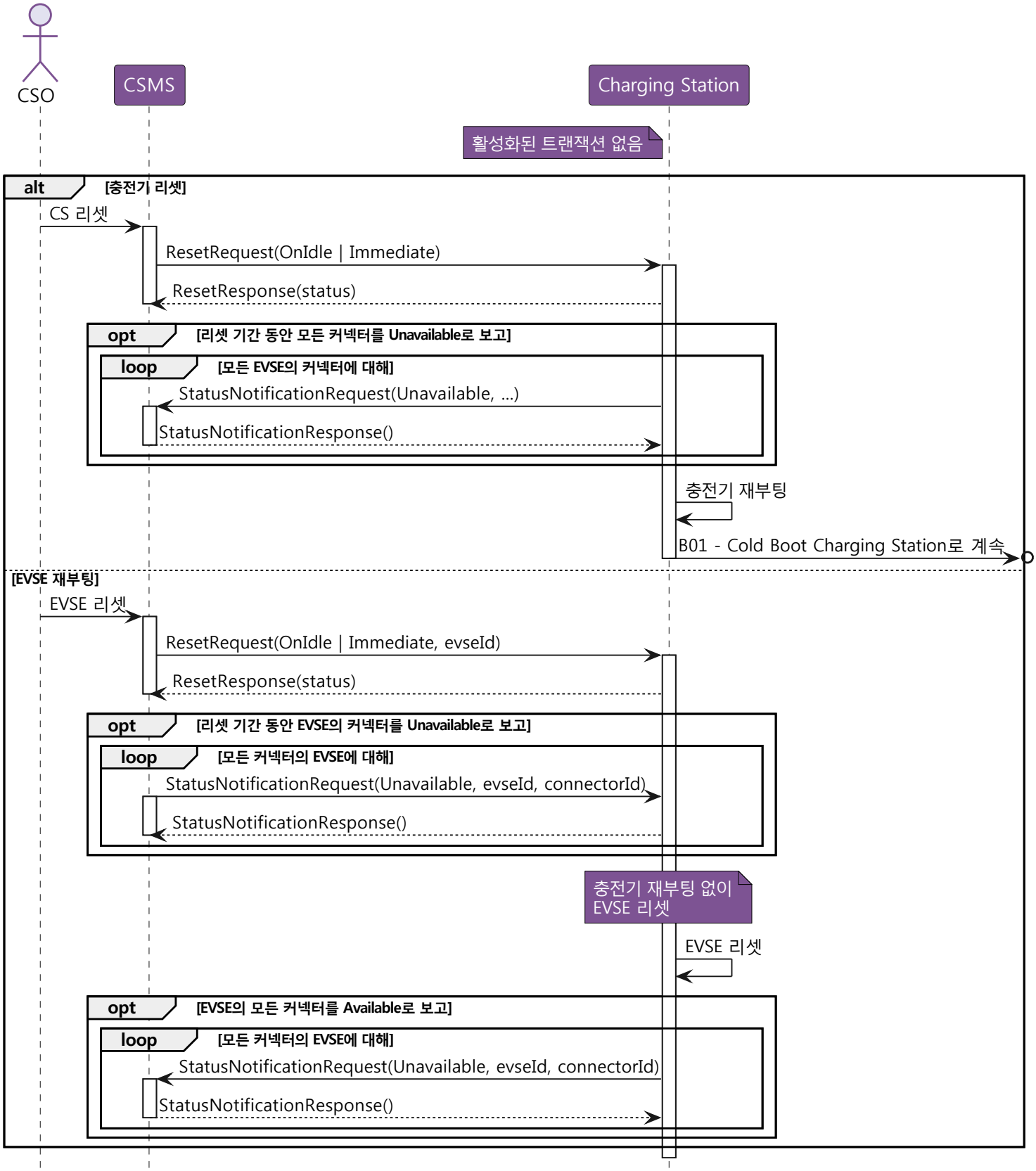

Figure 20. 시퀀스 다이어그램: 트랜잭션 없이 재설정

7	**오류 처리**	n.a
8	**참고**	지속적인 상태: 예를 들어 EVSE가 *Unavailable* 로 설정된 경우 지속되어야 합니다. 충전기는 ResetResponse로 응답합니다.

B11 - Reset - Without Ongoing Transaction - Requirements

Table 56. B11 - 요구 사항

ID	전제 조건	요구 사항 정의
B11.FR.01	충전기가 ResetRequest를 수신할 때.	충전기는 ResetResponse로 응답해야 합니다.
B11.FR.02	CSMS에서 상태가 *Inoperative* 로 설정된 경우.	충전기를 재부팅한 후 EVSE는 재부팅 전과 마찬가지로 *Unavailable* 상태로 돌아가야 합니다.
B11.FR.03	B11.FR.01 AND *evseId* 매개변수가 제공되지 않음 AND ResetResponse가 *Accepted* 되었습니다.	충전기는 StatusNotification(Unavailable)을 보낼 수 있으며 재부팅을 시작해야 합니다.
B11.FR.04	B11.FR.03	충전기는 사용 사례 B01 - Cold Boot Charging Station에 설명된 대로 진행해야 합니다.
B11.FR.05	EVSE의 상태가 *Reserved* 인 경우.	충전기 재부팅 또는 EVSE 재설정 후 EVSE는 *Reserved* 상태로 돌아가야 합니다.
B11.FR.06	B11.FR.01 AND 예를 들어 중단할 수 없는 펌웨어 업데이트가 진행 중입니다.	충전기는 *Rejected* 상태로 응답해야 합니다.
B11.FR.07	B11.FR.01 AND 충전기는 지금 재설정을 수행할 수 없지만 나중에 재설정을 예약했습니다.	충전기는 *Scheduled* 상태로 응답해야 합니다.
B11.FR.08	B11.FR.01 AND *evseId* 매개변수가 제공됩니다. AND ResetResponse가 *Accepted* 이었습니다.	충전기는 EVSE에 대한 StatusNotification(Unavailable)을 보낼 수 있으며 *evseId* 매개변수로 참조되는 EVSE 재설정을 시작해야 합니다.
B11.FR.09	B11.FR.01 AND *evseId* 매개변수가 제공됨 AND 충전기가 개별 EVSE 재설정을 지원하지 않음	충전기는 ResetResponse *Rejected* 를 반환해야 함
B11.FR.10	충전기가 개별 EVSE 재설정을 지원하는 경우	충전기는 EVSE에 대해 장치 모델 변수 AllowReset을 true로 설정해야 함

B12 - Reset - With Ongoing Transaction

Table 57. B12 - 재설정 - 진행 중인 트랜잭션

번호	유형	설명
1	**이름**	재설정 - 진행 중인 트랜잭션
2	**ID**	B12
	기능 블록	B. 프로비저닝
3	**목표**	CSMS가 진행 중인 트랜잭션이 있는 동안 충전기에 자체 또는 EVSE를 재설정하도록 요청할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSMS가 ResetRequest를 보내 충전기에 자체 또는 EVSE를 재설정하도록 요청하는 방법을 다룹니다. (ResetRequest에 선택적 매개변수 evseId가 포함된 경우 특정 EVSE의 재설정만 요청됩니다.) 예를 들어 충전기가 제대로 작동하지 않는 경우 필요할 수 있습니다. CSMS는 충전기가 모든 트랜잭션을 스스로 종료하고 재부팅하거나 모든 진행 중인 트랜잭션이 정상적으로 종료될 때까지(EV 사용자가) 기다린 다음 재부팅할 수 있습니다.
	Actors	충전기, CSMS, CSO
	시나리오 설명	**1.** CSO가 CSMS에 충전기 또는 EVSE를 재설정하도록 요청합니다. **2.** CSMS가 ResetRequest를 보내 충전기가 스스로 또는 EVSE를 재설정하도록 요청합니다. **3a.** OnIdle 재설정을 수신하면 충전기는 ResetResponse(Scheduled)로 응답하여 모든 진행 중인 트랜잭션이 종료된 후 충전기가 자체 또는 EVSE를 재설정하려고 시도할 것임을 나타냅니다. 충전기는 충전을 계속하고 사용 가능한 모든 EVSE(또는 *evseId* 가 제공된 경우 요청에서 제공된 EVSE만)를 상태 *Unavailable* 로 설정하고 모든 트랜잭션이 완료되고 모든 TransactionEventRequest(eventType = Ended) 메시지가 전송될 때까지 기다립니다. **3b.** 즉시 재설정을 수신하면 충전기는 ResetResponse(Accepted)로 응답하여 충전기가 자체 또는 EVSE를 재설정하려고 시도할 것임을 나타냅니다. 충전기는 진행 중인 모든 트랜잭션(또는 요청에서 제공된 EVSE에서 실행 중인 트랜잭션만, *evseId* 가 제공된 경우)를 종료하고 TransactionEventRequest (eventType = Ended) 메시지를 전송하려고 시도합니다. **4.** evseId가 제공되지 않은 경우에만 충전기가 재부팅되고 방금 부팅된 상태로 돌아갑니다. B01 - Cold Boot Charging Station applies.
	대체 시나리오	B11 - Reset Without Ongoing Transaction
5	**필수 조건**	트랜잭션이 진행 중입니다.
6	**사후 조건**	**성공한 사후 조건:** 충전기가 자체 또는 EVSE를 재설정할 수 있었습니다. **실패 사후 조건:** 충전기가 자체 또는 EVSE를 재설정할 수 없습니다.

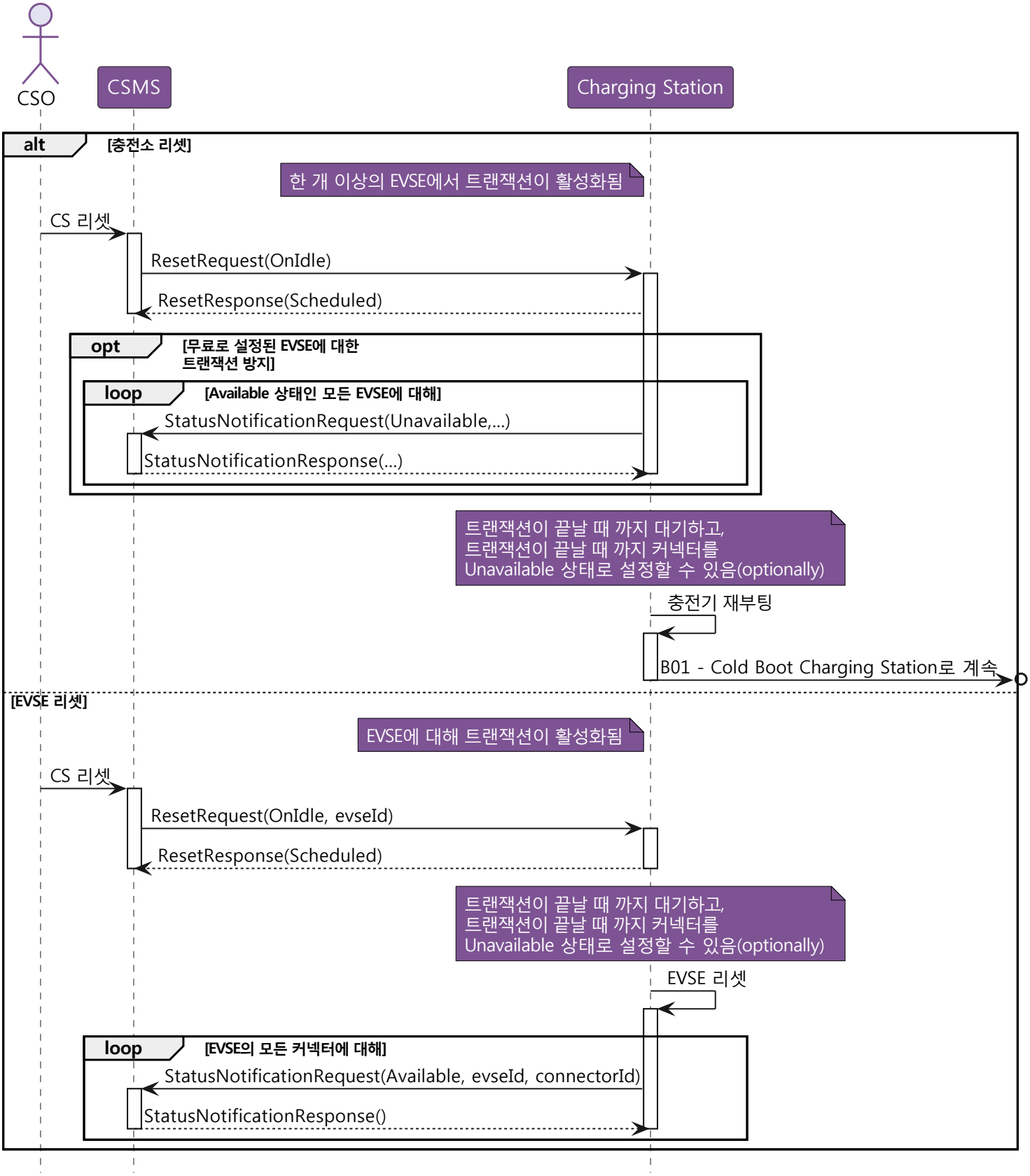

그림 21a: 시퀀스 다이어그램: 진행 중인 트랜잭션으로 OnIdle 재설정

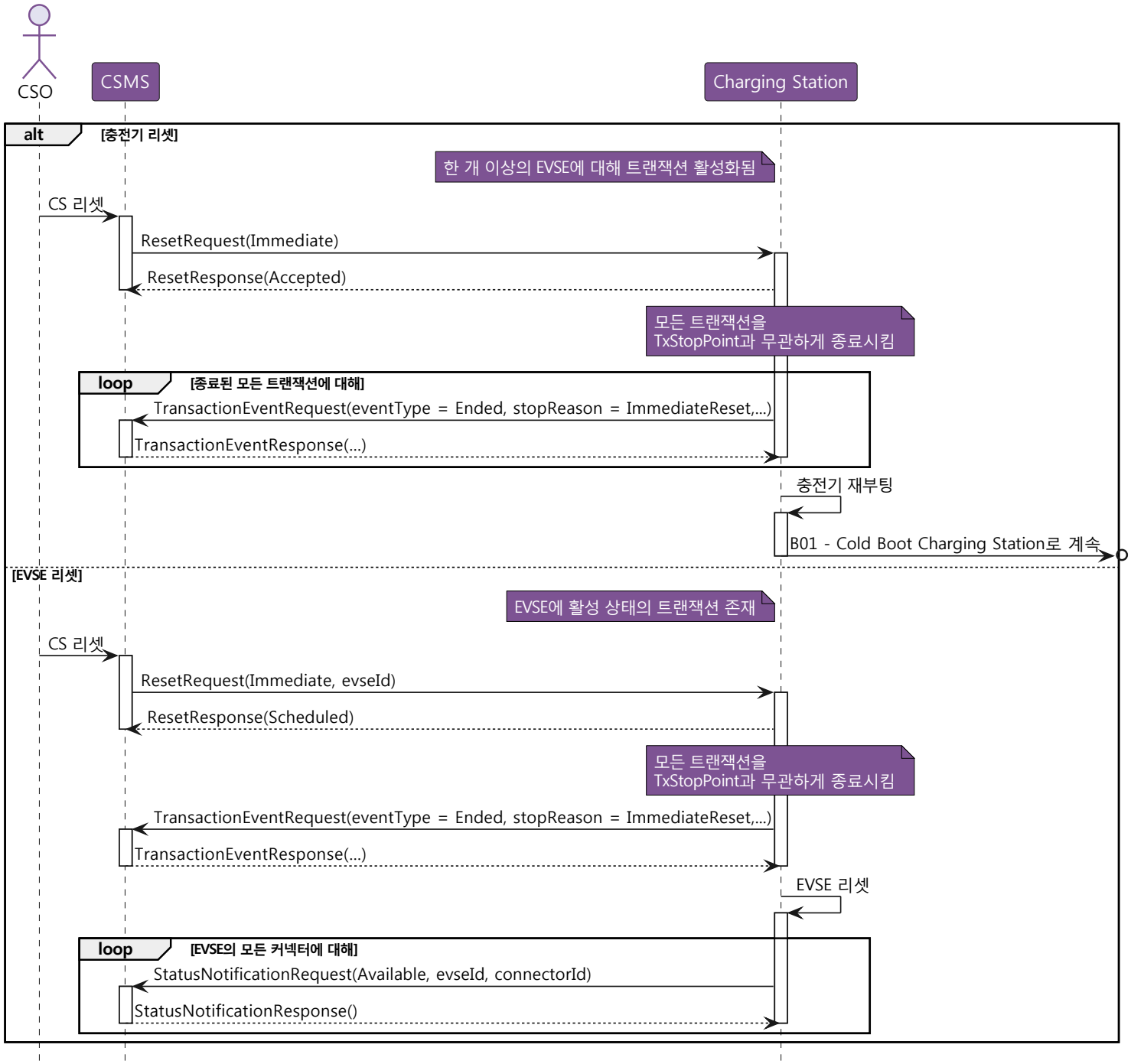

그림 21b: 시퀀스 다이어그램: 진행 중인 트랜잭션으로 즉시 재설정

7	**오류 처리**	ResetRequest를 수락한 후, CSMS에 전달할 수 없는 TransactionEventRequest 메시지는 반드시 큐에 넣어야 합니다.
8	**참고**	n/a

B12 - Reset - With Ongoing Transaction - Requirements

Table 58. B12 - 요구 사항

ID	전제 조건	요구 사항 정의
B12.FR.01	충전기가 ResetRequest(OnIdle)를 수신하고 트랜잭션이 진행 중인 경우	충전기는 ResetResponse(Scheduled)로 응답하여 충전기 또는 EVSE의 모든 트랜잭션이 종료된 후 충전기가 자체 또는 EVSE를 재설정할지 여부를 표시해야 합니다.

ID	전제 조건	요구 사항 정의
B12.FR.02	충전기가 ResetRequest(Immediate)를 수신하고 트랜잭션이 진행 중인 경우	충전기는 ResetResponse(Accepted)로 응답하여 충전기가 자체 또는 EVSE를 재설정할지 여부를 나타냅니다.
B12.FR.03	*evseId* 가 제공되지 않은 경우 AND 트랜잭션이 진행 중이고 OnIdle 재설정이 수신된 경우.	충전기의 트랜잭션은 E06 - Stop Transaction과 같이 재부팅 전에 정상적으로 종료되어야 합니다.
B12.FR.04	*evseId* 가 제공되지 않은 경우 AND 트랜잭션이 진행 중이고 Immediate Reset이 수신된 경우.	충전기는 진행 중인 모든 트랜잭션을 종료하고 재부팅을 수행하기 전에 TransactionEventRequest (eventType = Ended) 메시지를 보내야 합니다.
B12.FR.05	*evseId* 없이 Immediate Reset을 수신하고 TransactionEventResponse가 시간 초과 내에 수신되지 않는 경우.	충전기는 TransactionEventRequest를 대기열에 넣고 재부팅한 후 TransactionEventRequest를 다시 보내야 합니다.
B12.FR.06	CSMS에서 상태가 *Inoperative* 로 설정된 경우.	충전기 재부팅 또는 EVSE 재설정 후 EVSE는 충전기 재부팅 또는 EVSE 재설정 이전과 마찬가지로 *Unavailable* 상태로 돌아가야 합니다.
B12.FR.07	*evseId* 가 제공된 경우 AND EVSE에서 트랜잭션이 진행 중이고 OnIdle 재설정이 수신된 경우.	EVSE의 트랜잭션은 E06 - Stop Transaction과 같이 재설정 전에 정상적으로 종료되어야 합니다.
B12.FR.08	*evseId* 가 제공된 경우 AND EVSE에서 트랜잭션이 진행 중이고 Immediate Reset이 수신된 경우.	충전기는 EVSE에서 진행 중인 트랜잭션을 종료하고 재설정을 수행하기 전에 TransactionEventRequest (eventType = Ended) 메시지를 보내야 합니다.
B12.FR.09	B12.FR.01 AND *evseId* 매개변수가 제공됨 AND 충전기는 개별 EVSE 재설정을 지원하지 않음	충전기는 ResetResponse *Rejected* 를 반환해야 합니다.

C. Authorization

1. Introduction

이 기능 블록은 모든 인증 관련 기능을 설명하며, 사용자 인증을 위한 다양한 방법(온라인 및/또는 오프라인)과 AuthorizeRequest 메시지 처리/동작, 인증 캐시 기능 등을 포함합니다.

사용자가 전기차를 충전기에서 분리하려고 할 때, 충전기는 사용자가 충전을 시작한 사람인지 또는 동일한 그룹에 속해 있어 충전을 종료할 수 있는 권한이 있는지를 확인해야 합니다. 권한이 확인되면, 충전기는 CSMS에 충전이 중지되었음을 알립니다.

1. 사용자 경험을 향상시키기 위해, 충전기는 Authorization Cache를 사용하여 식별자의 로컬 인증을 지원할 수 있습니다.
2. `LocalAuthorizeOffline`구성 변수는 충전기가 ***오프라인 상태*** 에서 인증 캐시를 사용하여 사용자를 인증할지 여부를 결정합니다.
3. `LocalPreAuthorize`구성 변수는 충전기가 CSMS와의 인증을 수행하지 않고 인증 캐시를 사용하여 충전을 시작할지 여부를 결정합니다.

1.1. ID Tokens

이 섹션은 규범적입니다.

OCPP는 이제 다양한 유형의 인증을 사용할 수 있게 되었습니다. OCPP 1.X는 RFID만 지원했지만, 이제 OCPP는 신용카드, PIN 코드, 즉시 시작 버튼 등도 지원합니다.

IDTokenType에는 인증에 사용할 식별자가 포함되어 있습니다. 이는 대/소문자를 구분하지 않는 문자열과 형식의 조합으로 정의됩니다. IDTokenType 클래스의 메시지 데이터 요소(GroupId 포함)는 CSMS에 의미 있는 모든 데이터(예: 충전 활동의 시작자를 식별하기 위한 목적)를 포함할 수 있으며(MAY) 충전기는 IdTokenType의 설명에 제공된 것 외에 이러한 데이터의 형식이나 내용에 대해 어떠한 추정도 해서는 안 됩니다(예: UID와 같은 값으로 가정하여 16진수 문자만 포함하거나 짝수 자릿수여야 한다고 가정하는 것). 로컬 토큰 판독기 하드웨어를 통해 획득한 IdToken 데이터는 일반적으로 물리적 IdToken의 (4, 7 또는 10바이트) UID 값으로, 일반적으로 8, 14 또는 20개의 16진수 문자로 표시됩니다.

NOTE	상호 운용성을 촉진하기 위하여 IdTokenType 데이터 유형이 *ISO14443*인 경우 현재까지의 일반적인 관행에 따라, 이러한 UID를 UID바이트의 16진수 표현으로 나타내는것을 권장합니다. ISO 14443-3에 따르면 바이트 0은 16진수 문자열 가장 앞에 와야 합니다. (최상위 니블의 바이트에 0이 먼저)

1.1.1. Additional Info

AdditionalInfo는 *IdToken* 을 사용한 일반적인 인증 외에도 CSMS에서 유효성을 검증될 수 있는 추가 정보를 전송하는데 사용될 수 있습니다.

AdditionalInfo 에는 관련된 모든 당사자가 동의해야하는 하나 이상의 사용자 지정 형식이 포함되어 있습니다. AdditionalInfo가 구현된다면 충전기는 일반 작동 중 *AdditionalInfo* 를 캐싱하고 포함해야 하며 구성 변수 `AdditionalInfoItemsPerMessage`를 설정해야 합니다. *AdditionalInfo* 가 구현되지 않았거나, 지원하지 않는 *AdditionalInfo.type* 가 사용될 경우, CSMS/충전기는 *AdditionalInfo* 를 무시할 수(MAY) 있습니다.

1.2. Group ID Tokens

이 섹션은 규범적입니다

CSMS는 일련의 신원 토큰을 "그룹"으로 취급할 수 있는 기능을 가지고 있어, 그룹 내의 어떤 토큰이든 트랜잭션을 시작하고 동일한 토큰 또는 동일한 그룹 내의 다른 토큰이 트랜잭션을 종료할 수 있습니다. 이는 여러 운전자가 하나 이상의 공유 전기차를 단일 충전 계약 계정으로 사용하는 가족이나 비즈니스의 일반적인 유즈케이스를 지원합니다. "GroupId"로 사용되는 IdTokenTypes는 종종 계정의 첫 번째/주 RFID 카드의 UID 대신 GroupId를 위한 공유 중앙 계정 식별자를 사용할 수 있습니다.

토큰(idTags)은 IdTokenInfo의 선택적 *groupIdToken* 요소에 공통 그룹 식별자를 지정하여 인증 목적으로 그룹화됩니다. 두 IdToken의 GroupIdTokens가 일치하고 비어 있지 않은 경우 동일한 그룹에 속하는 것으로 간주됩니다.

NOTE	GroupId는 idToken과 동일한 명목 데이터 유형(IdTokenType)을 가지고 있지만, 이 요소의 값은 IdTokenTypes의 일반 형식에 있지 않거나 실제 유효한 IdTokenType을 나타내지 않을 수 있습니다(예: 공통 공유 "계정 번호"일 수 있음): 따라서 GroupId 값은 제시된 토큰 값과 비교하는 데 사용해서는 안 됩니다(동일한 값이 idToken 값으로 발생하지 않는 한).

1.3. Authorization Cache

충전기는 CSMS에서 성공적으로 승인된 이전에 제시된 식별자의 기록을 **자율적으로** 유지하는 인증 캐시를 구현할 수 있습니다. 인증 캐시는 충전기에서 인증 프로세스를 가속화하는 데 사용할 수 있습니다. 로컬에 저장된 캐시를 사용하면 사용자가 충전기가 CSMS에서 인증을 확인할 때까지 기다릴 필요가 없기 때문입니다. 인증 캐시가 있는 경우 작동은 `AuthCacheEnabled` 구성 변수에 의해 보고되고(가능한 경우 제어됨) 일반 인증 캐시 항목의 선택적 만료 시간은 구성 변수 `AuthCacheLifeTime`에서 설정할 수 있습니다. 특정 항목에 대해 다른 만료 시간이 필요한 경우, 예를 들어 AuthorizeResponse의 iDTokenInfo에서 반환되는 cacheExpiryDateTime에서 이를 설정할 수 있습니다.

인증 캐시 기능을 구현/사용하는 방법에 대한 자세한 내용은 유즈케이스 C10 - Store Authorization Data in the Authorization Cache 및 C12 - Start Transaction - Cached Id를 참조하세요.

충전기가 인증 캐시와 단가 정보를 모두 지원하는 경우(참조: Tariff & Cost), 단가 정보를 인증 캐시에 저장해서는 안 됩니다. 오래된(outdated) 정보일 수 있기 때문입니다.
충전기는 Authorization Cache 항목으로 명시적으로 승인할 수 없는 진짜 사용자에게 충전하는 것을 거부하지 않기 위해 ***오프라인*** 일 때 제시된 ***모든*** 식별자의 승인을 지원할 수 있습니다. 이 기능은 Unknown Offline Authorization에서 더 자세히 설명합니다.
인증 캐시의 개인 정보를 안전하게 저장하는 것을 권장합니다.(예: 캐시에 해시된 idToken만 저장).

1.4. Local Authorization List

로컬 인증 목록은 CSMS와 동기화할 수 있는 식별자 목록입니다. 오프라인일 때 사용자를 승인하고 충전기와 CSMS 간의 통신이 느릴 때(겉보기에) 승인 응답 시간을 더 빠르게 할 수 있습니다. CSMS는 로컬 인증 목록을 대체할 식별자의 전체 목록을 보내거나 로컬 인증 목록에 적용할 변경 사항(추가, 업데이트, 삭제) 목록을 보내 목록을 동기화할 수 있습니다. 이를 지원하는 작업은 GetLocalListVersion 및 SendLocalList입니다.

이 목록에는 모든(또는 일부) 식별자의 인증 상태와 *cacheExpiryDateTime* 의 해당 만료 날짜가 포함되어 있습니다. 이러한 값은 로컬 인증 중에 사용자에게 보다 세부적인 정보를 제공하는 데 사용될 수 있습니다(예: 표시 메시지).

로컬 인증 목록 기능을 구현/사용하는 방법에 대한 자세한 내용은 D01 - Send Local Authorization List, C13 - Offline Authorization through Local Authorization List 및 C14 - Online Authorization through Local Authorization List 유즈케이스를 참조하세요.

NOTE	Authorization Cache와 Local Authorization List 메커니즘의 차이점에 유의하세요. Authorization Cache는 충전기에서 자율적인 메커니즘인 반면, Local Authorization List는 CSMS와 충전기(CSMS에서 시작) 간에 동기화되는 목록입니다.

NOTE	Authorization Cache와 Local Authorization List는 **별개의** 논리적 데이터 구조입니다. Authorization Cache와 Local Authorization List가 모두 지원되는 경우 충전기는 Local Authorization List 항목을 Authorization Cache 항목보다 우선시해야 합니다.

다음 구성 변수는 충전기에서 로컬 인증 목록에 대한 정보를 제공하는 데 사용됩니다.

- `LocalAuthListEntries` (로컬 인증 목록에 있는 최대 IdToken 수를 보고합니다)
- `LocalAuthListEnabled`
- `LocalAuthListAvailable`
- `ItemsPerMessageSendLocalList`
- `BytesPerMessageSendLocalList`

1.5. Unknown Offline Authorization

오프라인 일 때 충전기는 Local Authorization List 및/또는 Authorization Cache에서 찾을 수 없는 "알 수 없는" 식별자의 자동 인증을 허용할 수 있습니다.알 수 없는 오프라인 인증 기능이 지원되는 경우 작동은 `OfflineTxForUnknownIdEnabled` 구성 변수에 의해 보고되고 가능한 경우 제어됩니다. CSMS에 대한 연결이 복구되면 충전기는 대기 중인 TransactionEventRequest 메시지를 보내야 합니다. 여기에는 transaction-related message handling에서 설명한 대로 ***오프라인*** 에서 승인된 충전이 포함될 수 있습니다. 이 상황에서 충전기가 트랜잭션을 계속하거나 중지해야 하는 옵션은 C15 - Unknown Offline Authorization을 참조하세요.

2. Use cases & Requirements

2.1. Authorization options

C01 - EV Driver Authorization using RFID

Table 59. C01 - EV Driver Authorization using RFID

번호	유형	설명
1	**이름**	RFID를 사용한 EV 운전자 인증
2	**ID**	C01
	기능 블록	C. 인증
3	**목표**	충전기가 충전을 시작하거나 중지하기 위해 EV 운전자 인증 요청을 CSMS에 보낼 수 있어야 합니다.
4	**설명**	충전기가 EV를 충전해야 하는 경우 충전을 시작하거나 중지하기 전에 먼저 EV 운전자를 인증해야 합니다.
	Actors	충전기, CSMS, EV 운전자
	시나리오 설명	**1.** EV 운전자가 EV 충전을 시작하거나 중지하고 싶어하며 RFID 카드를 제시합니다. **2.** 충전기가 AuthorizeRequest를 CSMS로 보내 인증을 요청합니다. **3.** AuthorizeRequest를 수신하면 CSMS가 AuthorizeResponse로 응답합니다. 이 응답 메시지는 IdToken이 CSMS에서 수락되는지 여부를 나타냅니다.
	대체 시나리오	C02 - Authorization using a start button C03 - Authorization using credit/debit card C04 - Authorization using PIN-code C05 - Authorization for CSMS initiated transactions C06 - Authorization using local id type C07 - Authorization using Contract Certificates C08 - Authorization at EVSE using ISO 15118 External Identification Means (EIM) C15 - Unknown Offline Authorization
5	**필수 조건**	해당 없음
6	**사후 조건**	**성공 사후 조건:** EV 운전자가 인증되었으며 충전을 시작하거나 중지할 수 있습니다. **실패 사후 조건:** 인증 메시지가 *Invalid*, *Blocked*, *Expired* 또는 *Unknown* 인 경우 EV 운전자는 충전을 시작하거나 중지할 수 없습니다. 단, EV 운전자가 충전을 시작하는 데 사용된 것과 동일한 토큰을 제시하는 경우는 예외입니다.

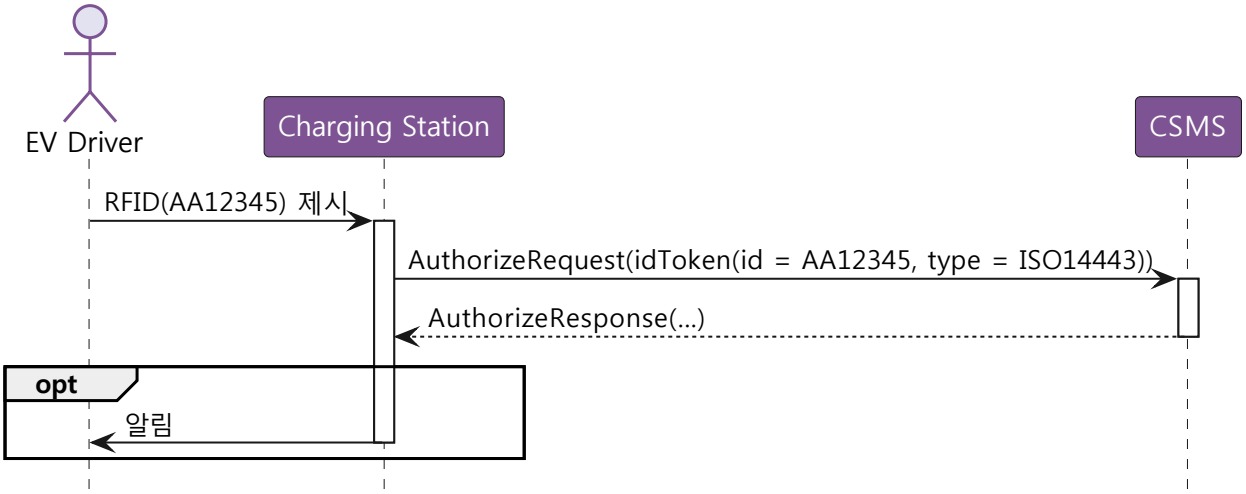

Figure 21. 시퀀스 다이어그램: EV 운전자 인증

7	**오류 처리**	인증이 'Accepted'가 아닌 경우 AuthorizeResponse에는 거부 이유를 나타내는 인증 상태 값이 포함됩니다.
8	**비고**	충전에 idToken이 유효하고 충전기에 EVSE가 3개 있다고 가정할 때, idToken이 충전을 허용할 때 *idTokenInfo* 의 내용은 다음과 같습니다: . EVES에서: *idTokenInfo.status* = Accepted. . EVSE 1에서: *idTokenInfo.status* = Accepted, *idTokenInfo.evseId* = [1]. . EVSE 1 + 2에서: *idTokenInfo.status* = Accepted, *idTokenInfo.evseId* = [1, 2]. . EVSE 중 어느 것도 아님: *idTokenInfo.status* = NotAtThisLocation.

C01 - EV Driver Authorization using RFID - Requirements

Table 60. C01 - Requirements

ID	전제 조건	요구 사항 정의	참고
C01.FR.01	구성 설정 AuthEnabled가 true입니다.	충전기는 인증 후에만 에너지를 제공해야 합니다.	
C01.FR.02	EV 운전자가 제시한 idToken이 Local Authorization List 또는 Authorization Cache에 없는 경우	충전기는 CSMS에 AuthorizeRequest를 보내 승인을 요청해야 합니다.	
C01.FR.03	idToken이 인증된 충전 중에 제시될 때 AND (a) 제시된 idToken이 인증을 시작한 idToken과 동일한 경우 OR (b) 제시된 idToken이 로컬 인증 목록 또는 인증 캐시에 있고 유효하며 인증을 시작한 IdToken과 동일한 GroupIdToken을 갖는 경우.	충전소는 AuthorizeRequest를 먼저 보내지 않고 충전 종료를 승인해야 합니다.	인증을 시작한 idToken은 항상 인증을 종료하는 데 사용할 수 있습니다. 종료 승인을 하면 충전이 종료됩니다. TxStopPoint에 따라 종료 인증을하면 충전이 종료될 수도 있습니다.
C01.FR.04		AuthorizeRequest는 식별자의 인증에만 사용해야 합니다.	
C01.FR.05	Local Authorization List 또는 Authorization Cache에 IdToken이 있는 경우.	충전기는 AuthorizeRequest를 CSMS로 보낼 수 있습니다.(MAY)	
C01.FR.06	CSMS가 *idToken* 에 대한 AuthorizeRequest를 수신하고 idToken에 연관된 groupIdToken이 있는 경우.	CSMS에서 충전기로 보낸 AuthorizeResponse에는 groupIdToken이 포함될 수 있습니다.	
C01.FR.07		AuthorizeResponse에는 승인 또는 거부 이유를 나타내는 인증 상태 값이 포함되어야 합니다.	거부 가능한 이유는 AuthorizationStatusEnumType을 참조하세요.
C01.FR.08	필드 language1이 설정되어 있고 충전기에 해당 *언어* 로 된 메시지가 있는 경우.	충전기는 사용자에게 **언어1** 로 메시지를 표시해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
C01.FR.09	필드 language1이 설정되어 있고 충전기에 해당 언어로 된 메시지가 없고 필드 language2가 설정되어 있고 충전기에 해당 *언어* 로 된 메시지가 있는 경우	충전기는 사용자에게 **언어2** 로 메시지를 표시해야 합니다.	
C01.FR.10	필드 language1이 설정되어 있지 않은 경우	필드 language2는 설정되지 않아야 합니다.	
C01.FR.11		필드 language1은 필드 language2와 달라야 합니다.	
C01.FR.12		대체로 **영어** 로 메시지를 구현하는 것을 권장합니다.	
C01.FR.13	language1과 language2가 충전기에 설치된 언어와 일치하지 않는 경우	EV 운전자에게 **영어** 로 메시지를 표시하는 것을 권장합니다.	
C01.FR.17		언어는 RFC-4646 태그로 지정해야 합니다. 참조: [RFC5646], 예: 미국 영어는 "en-US"입니다.	
C01.FR.18	IdToken이 유효하고 EV 운전자가 이 충전기에서 제공하는 EVSE 유형에서 충전할 수 없는 경우.	CSMS는 idTokenInfo.status *NotAllowedTypeEVSE* 를 사용하여 AuthorizeResponse를 보내야 합니다.	
C01.FR.19	*idToken* 가 충전기의 모든 EVSE에서 허용된 경우.	CSMS는 *idTokenInfo* 에 빈(또는 없는) *evseId* 목록이 있는 AuthorizeResponse를 보내야 합니다.	가장 일반적인 경우입니다. *idToken* 은 모든 EVSE에서 허용될 수 있지만 여전히 충전이 허용되기 전에 *idTokenInfo.status* 가 `Accepted` 여야 합니다.
C01.FR.20	*idToken* 이 충전기의 EVSE 하위 집합에서 허용될 경우.	CSMS는 *IdTokenInfo* 에 허용된 EVSE가 있는 *evseId* 목록이 있는 AuthorizeResponse를 보내야 합니다.	*idToken* 의 유효성과 이 (유형의) 토큰이 EVSE에서 허용되는지 여부의 차이에 유의하십시오. 충전이 허용되기 전에 *idTokenInfo.status* 가 여전히 `Accepted` 여야 합니다.
C01.FR.21	C01.FR.20	충전기는 AuthorizeResponse에 언급된 EVSE에서만 충전을 허용해야 합니다.	
C01.FR.22	*idToken* 이 충전기의 어떤 EVSE에도 허용되지 않은 경우.	CSMS는 *idTokenInfo.status* 가 `NotAtThisLocation` 이고 evseId 목록이 비어 있거나 없는(또는 없는) AuthorizeResponse를 보내야 합니다.	*idToken* 이 모든 EVSE에서 허용되는 상황과 구별하기 위해 `NotAtThisLocation` 상태가 필요합니다.

ID	전제 조건	요구 사항 정의	참고
C01.FR.23	충전이 아직 활성화되어 있고, 이전에 idToken에 의해 승인되었지만, 이제는 더 이상 충전이 승인되지 않고, 새로운 idToken이 승인을 위해 충전기에 제시되었으며, 이는 초기 idToken과 **다를경우**	충전기는 다른 idToken의 인증을 허용해서는 안 됩니다.	충전에 대한 여러 *idToken* 은 CSMS에서 지원되지 않을 가능성이 큽니다.
C01.FR.24	충전이 아직 활성화되어 있고, 이전에 idToken에 의해 승인되었지만, 이제는 더 이상 충전이 승인되지 않았고 충전기에서 트랜잭션의 초기 idToken과 **다른** 새 idToken에 대한 AuthorizeRequest를 보내는 경우	CSMS는 이 idToken에 대해 *idTokenInfo.status* = `NotAtThisTime` 인 AuthorizeResponse로 응답하는 것을 권장합니다.	충전기에서 두 번째 인증을 하는 경우 CSMS는 *idToken* 을 거부할 수 있습니다.

C02 - Authorization using a start button

Table 61. C02 - Authorization using a start button

번호	유형	설명
1	**이름**	시작 버튼을 사용한 인증
2	**ID**	C02
	기능 블록	C. 인증
3	**목표**	시작 버튼이 있는 충전기가 충전을 시작할 수 있도록 합니다.
4	**설명**	일부 충전기의 경우 사용자 인증이 필요하지 않을 수 있습니다. 간단한 충전기에는 충전을 시작하기 위해 비싼 RFID 리더 대신 버튼이 있을 수 있습니다. 이러한 충전기가 충전을 시작하면 AuthorizeRequest를 보낼 필요가 없습니다. TransactionEventRequest(eventType = Started)에서 IdTokenType 정보를 제공해야 하며, CSMS는 이를 거부할 수 없습니다.
	Actors	EV 운전자, 충전기, CSMS
	시나리오 설명	**1.** EV 운전자가 EV와 충전기 사이에 충전 케이블을 꽂습니다. **2.** 충전기가 StatusNotificationRequest와 TransactionEventRequest(eventType = Started)를 보내 케이블이 꽂혔다는 사실을 CSMS에 알립니다. **3.** EV 운전자가 시작 버튼을 눌러 충전을 시작합니다. **4.** 충전기가 EV 충전을 시작합니다. **5.** 충전기가 TransactionEventRequest (eventType = Updated) 메시지를 IdTokenEnumType: *NoAuthorization*과 함께 CSMS에 보내 충전이 시작되었음을 알립니다. **6.** TransactionEventRequest (eventType = Updated)를 수신하면 CSMS는 TransactionEventResponse로 응답합니다. IdTokenInfo.status가 *Accepted* 로 설정됩니다
	대체 시나리오	C01 - EV Driver Authorization using RFID C03 - Authorization using credit/debit card C04 - Authorization using PIN-code C05 - Authorization for CSMS initiated transactions C06 - Authorization using local id type C07 - Authorization using Contract Certificates C08 - Authorization at EVSE using ISO 15118 External Identification Means (EIM) C15 - Unknown Offline Authorization
5	**필수 조건**	충전기에는 RFID 리더 대신 EV 충전을 시작할 수 있는 시작 버튼이 있습니다.
6	**사후 조건**	충전기에서 충전이 진행 중이며 CSMS에서 충전을 인식합니다.

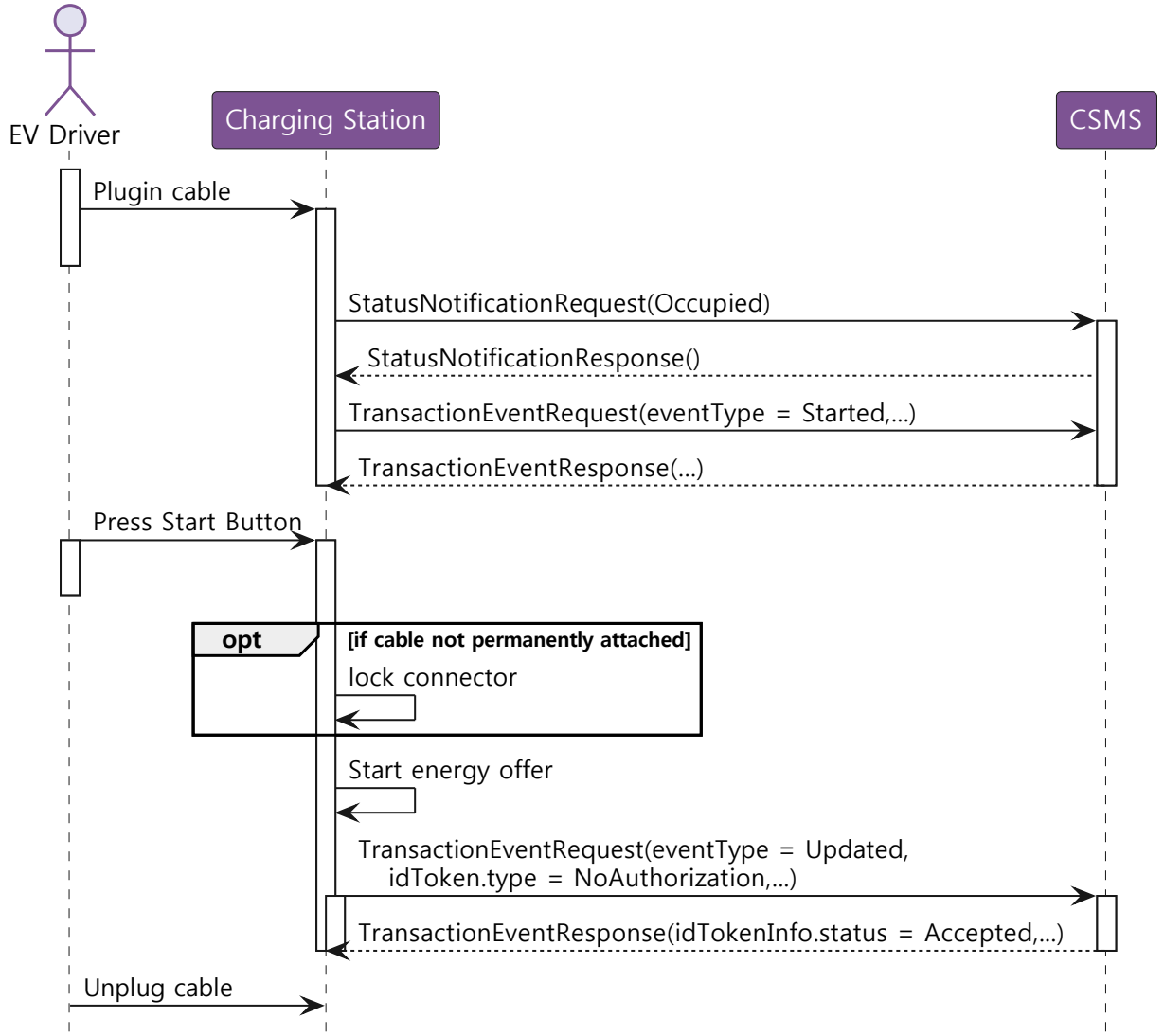

Figure 22. 시퀀스 다이어그램: 시작 버튼을 사용한 인증

7	**오류 처리**	n/a
8	**비고**	시작 버튼은 기계식 키 또는 이와 유사한 것일 수도 있습니다. 충전기가 케이블 연결 시 충전을 시작하도록 구성된 경우 시작 버튼을 생략할 수도 있습니다. 위의 시나리오 설명과 시퀀스 다이어그램은 트랜잭션 시작을 위한 구성 변수가 다음과 같이 구성되는 것을 기반으로 합니다. `TxStartPoint`: EVConnected, Authorized, DataSigned, PowerPathClosed, EnergyTransfer 이 유즈케이스는 다른 구성에도 유효하지만 트랜잭션이 다른 순간에 시작/중지될 수 있으며, 이때 메시지를 보내는 순서가 변경될 수 있습니다. 자세한 내용은 유즈케이스를 참조하세요. E01 - Start Transaction options.

C02 - Authorization using a start button - Requirements

Table 62. C02 - Authorization using a start button - Requirements

ID	전제 조건	요구 사항 정의
C02.FR.01	버튼으로 충전이 시작될 때.	충전기는 IdTokenType 유형: NoAuthorization 및 필드: idToken이 비어 있는(빈 문자열) TransactionEventRequest를 보내야 합니다.

ID	전제 조건	요구 사항 정의
C02.FR.02	CSMS는 IdTokenType 유형 NoAuthorization을 갖는 TransactionEventRequest를 수신합니다.	CSMS는 IdTokenInfo.status가 Accepted로 설정된 TransactionEventResponse로 응답해야 합니다.
C02.FR.03	충전기가 인증 캐시를 구현했고 충전기가 어느 메시지에서 IdTokenType 유형 NoAuthorization에 대한 IdTokenInfo를 수신하는 경우	충전기는 인증 캐시에 정보를 저장해서는 안 됩니다.

C03 - Authorization using credit/debit card

Table 63. C03 - Authorization using credit/debit card

번호	유형	설명
1	**이름**	신용 카드를 사용한 승인
2	**ID**	C03
	기능 블록	C. 승인
3	**목표**	신용 카드를 사용하여 충전을 시작할 수 있도록 합니다.
4	**설명**	하우징 내부에 신용/직불 카드 단말기가 내장된 충전기 또는 중앙 결제 단말기/키오스크가 있는 충전기 그룹에 속하는 충전기. EV 운전자가 카드를 사용하여 충전 비용을 지불합니다. 트랜잭션은 결제 회사에서 승인되고, CSMS는 결제 시스템으로부터 메시지를 받고, RequestStartTransactionRequest를 충전기로 보내 충전을 시작합니다.
	Actors	EV 운전자, 결제 시스템, CSMS, 충전기
	시나리오 설명	**1.** EV 운전자가 충전 케이블을 꽂습니다. **2.** 충전기가 StatusNotificationRequest와 TransactionEventRequest(eventType = Started)를 보내 케이블이 꽂혔다는 것을 CSMS에 알립니다. **3.** 운전자가 신용/직불 카드 단말기를 사용하여 충전을 승인/지불합니다. **4.** 단말기가 자체 서버/백오피스와 통신합니다. **5.** 결제 시스템이 CSMS에 사용자 인증 메시지를 보냅니다. **6.** CSMS가 이 트랜잭션에 대한 IdToken으로 사용할 고유한 ID를 생성합니다. **7.** CSMS가 생성된 IdToken과 함께 RequestStartTransactionRequest를 충전기로 보냅니다. **8.** 충전기는 RequestStartTransactionResponse를 Accepted와 함께 보내 RequestStartTransactionRequest를 수락합니다. **9.** 충전기가 EV 충전을 시작합니다. **10.** 충전기는 TransactionEventRequest (eventType = Updated)를 보내 CSMS에 충전이 시작되었음을 알립니다.
	대체 시나리오	C01 - EV Driver Authorization using RFID C02 - Authorization using a start button C04 - Authorization using PIN-code C05 - Authorization for CSMS initiated transactions C06 - Authorization using local id type C07 - Authorization using Contract Certificates C08 - Authorization at EVSE using ISO 15118 External Identification Means (EIM) C15 - Unknown Offline Authorization

번호	유형	설명
5	**필수 조건**	충전을 시작하기 위하여 충전기에 신용/직불 카드 단말기가 있거나, 중앙 결제 단말기가 있는 충전기 그룹에 속해야 합니다.
6	**사후 조건**	충전기에서 충전 진행

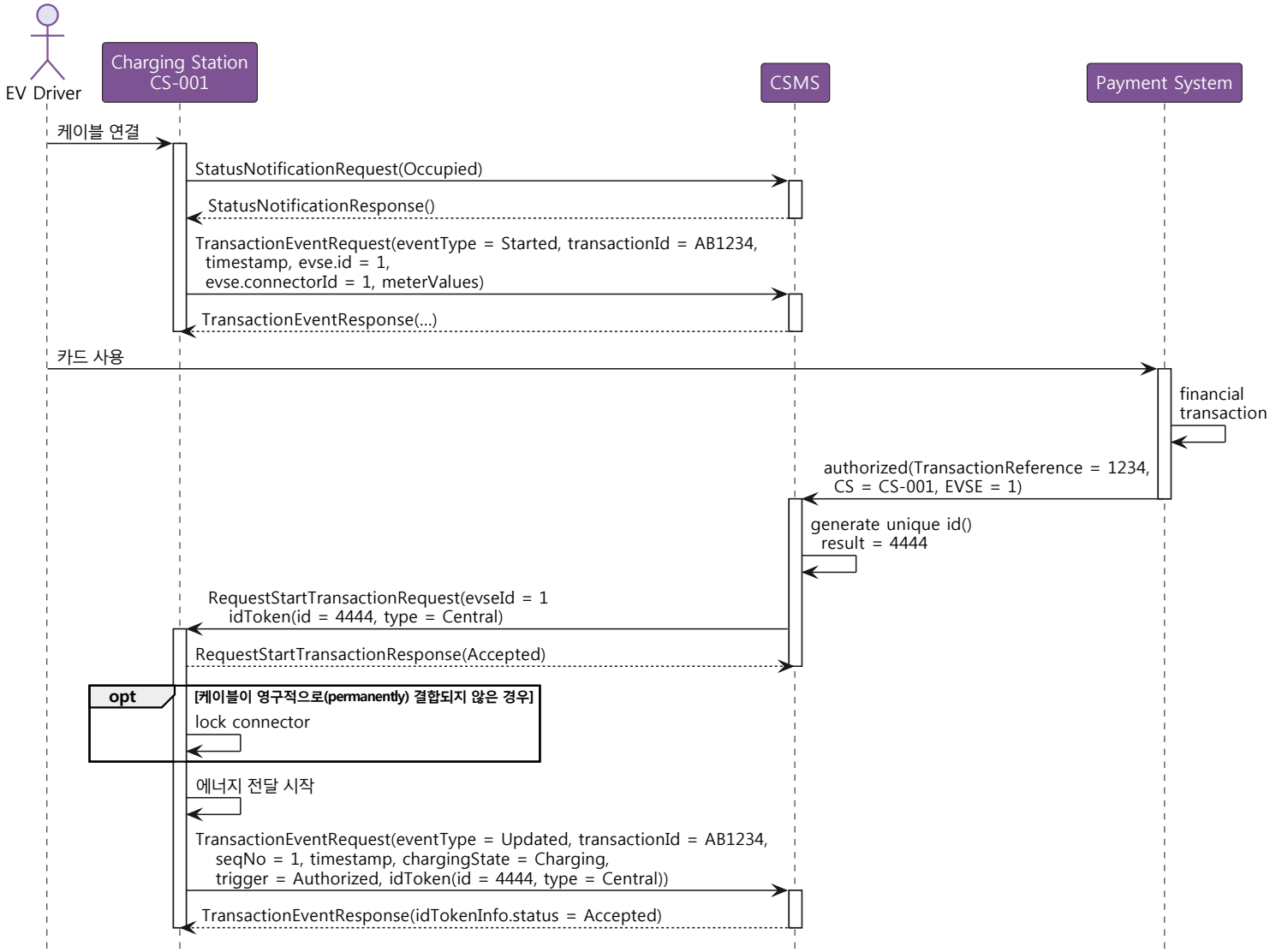

Figure 23. 시퀀스 다이어그램: 신용/직불 카드를 사용한 승인

7	**오류 처리**	n/a

8	설명	이 유즈케이스는 기존 OCPP 메시지를 사용하여 신용/직불 카드로 시작된 트랜잭션을 처리하는 방법의 예이며, 이런 방식으로 신용/직불 카드 결제 솔루션을 구현할 필요는 없습니다. 결제 시스템은 사용자 승인을 처리하는 여러 구성 요소로 구성될 수 있습니다. 이러한 구성 요소의 인터페이스와 결제 시스템과 CSMS 간의 통신은 이 문서의 범위에 포함되지 않습니다. 신용/직불 카드로 시작된 트랜잭션을 중단하는 것은 정의되지 않았으며, 이는 구현자에게 맡겨져 있으며, 예를 들어 EV 측의 케이블 분리 및/또는 정지 버튼 등이 될 수 있습니다. 위의 시나리오 설명과 시퀀스 다이어그램은 시작 트랜잭션을 위한 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStartPoint`: EVConnected, Authorized, DataSigned, PowerPathClosed, EnergyTransfer 이 유즈케이스는 다른 구성에도 유효하지만, 그러면 트랜잭션이 다른 순간에 시작/중지될 수 있으며, 이때 메시지를 보내는 순서가 변경될 수 있습니다. 자세한 내용은 유즈케이스 E01 - Start Transaction options를 참조하세요.

C03 - Authorization using credit/debit card - Requirements

Table 64. C03 - Authorization using credit/debit card - Requirements

ID	전제 조건	요구 사항 정의
C03.FR.01	충전기가 IdTokenType 유형 Central을 가진 RequestStartTransactionRequest를 수신하는 경우	충전기는 수신된 IdTokenType에 대해 AuthorizeRequest를 보내서는 안 됩니다.
C03.FR.02	충전기가 인증 캐시를 구현했고 충전기가 모든 메시지에서 IdTokenInfo를 Central 유형의 IdTokenType에 대해 수신하는 경우	충전기는 Authorization Cache에 정보를 저장해서는 안 됩니다.

C04 - Authorization using PIN-code

이것은 정보 제공적인 유즈케이스이며, KeyCode ID 유형의 사용을 보여주는 것이 목적입니다. KeyCode의 다른 유즈케이스는 차량 번호판과 같은 예시가 있습니다.

Table 65. C04 - Authorization using PIN-code

번호	유형	설명
1	**이름**	PIN 코드를 사용한 인증
2	**ID**	C04
	기능 블록	C. 인증
3	**목표**	키 입력 단말기가 있는 충전기에서 PIN 코드를 승인할 수 있도록 합니다.
4	**설명**	충전기에 PIN 코드 입력 단말기가 있는 경우 EV 운전자는 PIN 코드를 입력합니다. 이 PIN 코드는 AuthorizeRequest를 사용하여 검증을 위해 CSMS로 전송됩니다.
	Actors	EV 운전자, 충전기, CSMS
	시나리오 설명	**1.** EV 운전자가 EV 충전을 시작하거나 중지하고 싶어서 단말기에 PIN 코드를 입력합니다. **2.** 충전기가 AuthorizeRequest 메시지를 CSMS로 전송합니다. 이 메시지의 필드는 IdTokenEnumType이며, *KeyCode*로 설정되어 승인을 요청합니다. **3.** AuthorizeRequest를 수신하면 CSMS가 AuthorizeResponse로 응답합니다. 이 응답은 CSMS가 KeyCode를 수락하는지 여부를 나타냅니다.
	대체 시나리오	C01 - EV Driver Authorization using RFID C02 - Authorization using a start button C03 - Authorization using credit/debit card C05 - Authorization for CSMS initiated transactions C06 - Authorization using local id type C07 - Authorization using Contract Certificates C08 - Authorization at EVSE using ISO 15118 External Identification Means (EIM) C15 - Unknown Offline Authorization
5	**필수 조건**	충전기에 EV 충전을 시작하기 위한 PIN 코드 입력 단말기가 있어야 합니다.
6	**사후 조건**	충전기에서 트랜잭션이 진행 중이며 CSMS가 트랜잭션을 인식하고 있습니다.

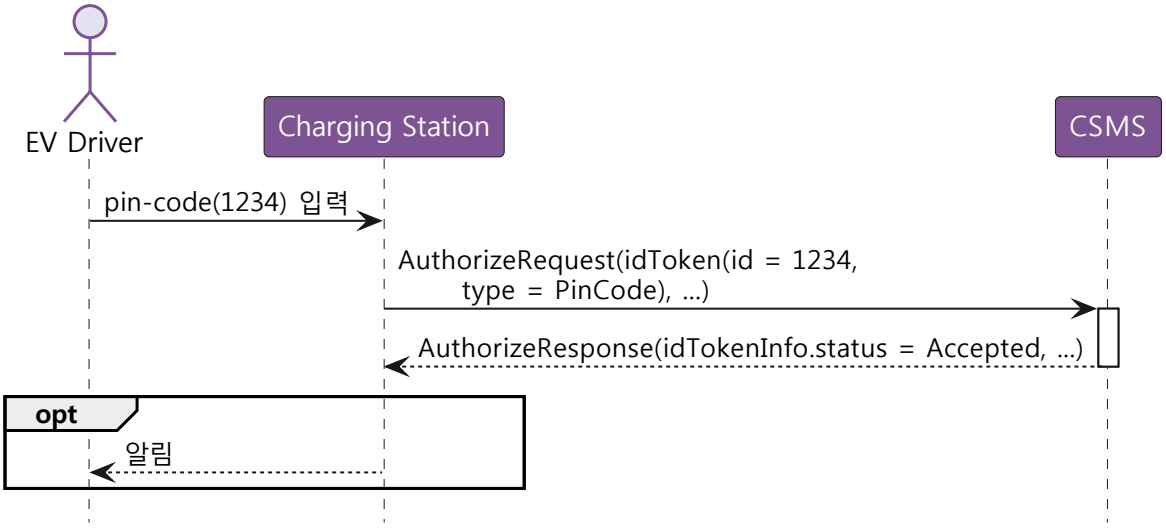

Figure 24. 시퀀스 다이어그램: PIN 코드를 사용한 인증

7	**오류 처리**	해당 없음
8	**비고**	CSMS 대신 충전기에서 PIN 코드가 검증되면 유즈케이스 C02 - Authorization using a start button이 적용됩니다.

C04 - Authorization using PIN-code - Requirements

Table 66. C04 - Authorization using PIN-code - Requirements

ID	전제 조건	요구 사항 정의
C04.FR.01	CSMS가 이 충전기에서 유효하지 않은 *keyCode* 가 포함된 AuthorizeRequest를 수신하는 경우	CSMS는 status = Invalid인 AuthorizeResponse 메시지로 응답해야 합니다.
C04.FR.02	CSMS가 유효한 *keyCode* 가 포함된 AuthorizeRequest를 수신하고 EV 운전자가 이 충전기에서 충전할 수 있는 경우	CSMS는 AuthorizeResponse 메시지로 status = Accepted로 응답해야 합니다.
C04.FR.03		충전기는 *keyCode* 를 인증 캐시에 저장할 수 있습니다(MAY).
C04.FR.04	*keyCode* 유형의 idToken이 사용되는 경우	충전기 또는 CSMS는 어떠한 로깅에도 IdToken을 표시해서는 안 됩니다. *keyCode* 는 로그에 나타나지 않아야 합니다.
C04.FR.05		언어는 RFC-5646 태그로 지정해야 합니다. [RFC5646] 참조, 예를 들어 미국 영어는 "en-US"입니다.
C04.FR.06	keyCode 유형의 idToken을 사용하는 경우	무차별 대입 공격을 방지하기 위한 조치를 취하는 것이 좋습니다. 예를 들어 잘못된 keyCode를 입력하려고 시도한 후 백오프 시간을 늘리는 것이 좋습니다.

C05 - Authorization for CSMS initiated transactions

Table 67. C05 - Authorization for CSMS initiated transactions

번호	유형	설명
1	**이름**	CSMS에서 시작된 트랜잭션에 대한 인증
2	**ID**	C05
	기능 블록	C. 인증
3	**목표**	CSMS가 서버에서 생성한 IdToken으로 충전기에서 트랜잭션을 시작할 수 있도록 합니다.
4	**설명**	CSMS가 RFID가 없거나 RFID를 알 수 없는 운전자를 위해 충전기에서 트랜잭션을 시작해야하는 경우입니다. 예를 들어, EV 운전자가 앱을 사용하여 트랜잭션을 시작합니다. CSMS는 IdToken을 확인하고 충전기에 이것이 RFID가 아니므로 캐시되지 않아야 하며 인증도 필요하지 않습니다.
	Actor	EV 운전자, CSMS, 충전기
	시나리오 설명	**1.** EV 운전자가 앱을 사용하여 충전을 시작합니다. **2.** 앱이 CSMS에 시작 요청을 보냅니다. **3.** CSMS는 IdToken을 결정합니다. 이 트랜잭션에 대한 IdToken으로 사용할 고유한 ID를 생성하거나 앱에서 제공하는 토큰(예: 사용자 계약의 ID)을 사용할 수 있습니다. **4.** CSMS는 이전 단계의 IdToken과 함께 RequestStartTransactionRequest를 충전기로 보냅니다. **5.** 충전기는 RequestStartTransactionResponse를 Accepted와 함께 보내 RequestStartTransactionRequest를 수락합니다. **6.** 충전기는 충전을 시작하고 TransactionEventRequest (eventType = Updated)를 보내 CSMS에 *chargingState* 가 변경되었음을 알립니다.
	대체 시나리오	C01 - EV Driver Authorization using RFID C02 - Authorization using a start button C03 - Authorization using credit/debit card C04 - Authorization using PIN-code C06 - Authorization using local id type C07 - Authorization using Contract Certificates C08 - Authorization at EVSE using ISO 15118 External Identification Means (EIM) C15 - Unknown Offline Authorization
5	**필수 조건**	케이블이 연결되어 있습니다.
6	**사후 조건**	충전 스테이션에서 트랜잭션이 진행 중입니다.

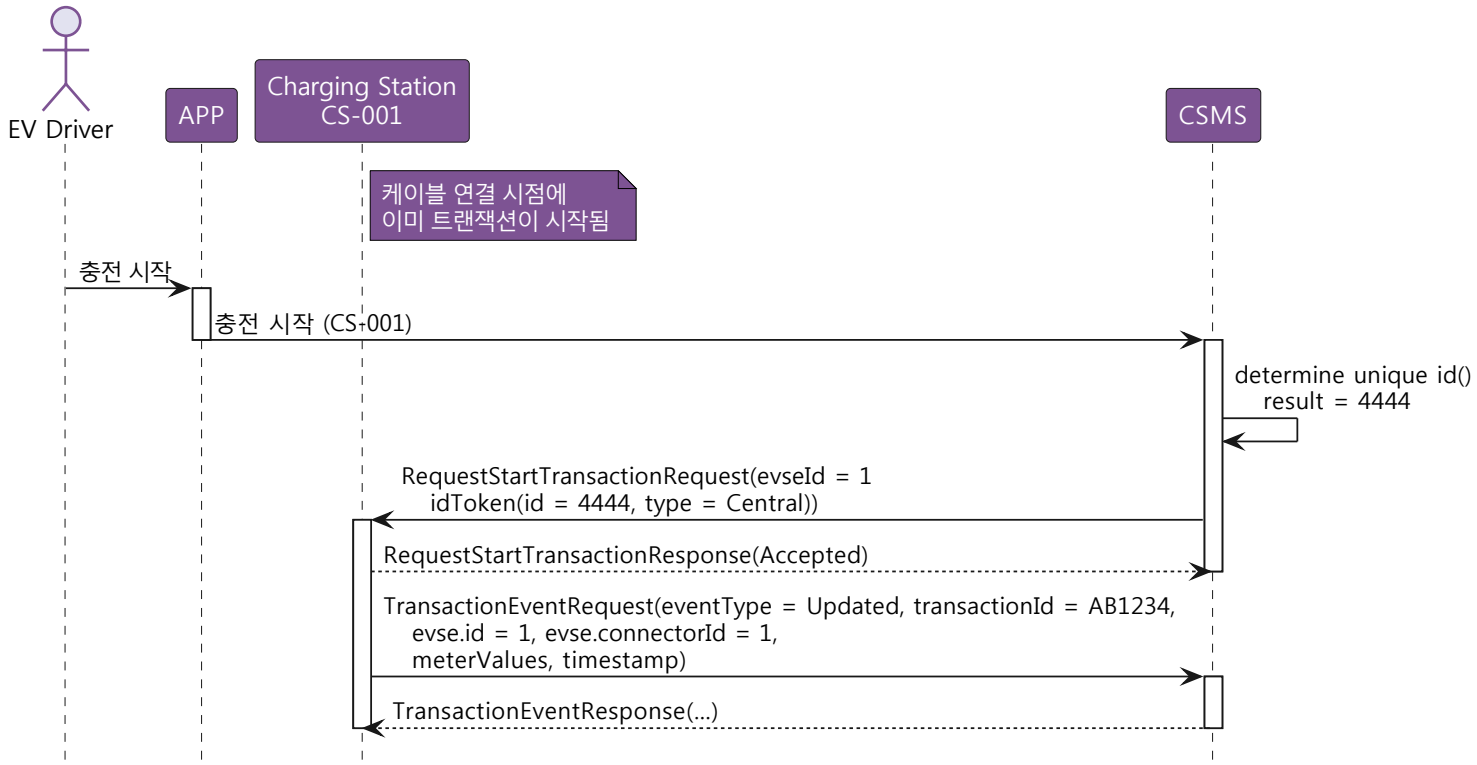

Figure 25. 시퀀스 다이어그램: CSMS에서 시작된 트랜잭션에 대한 인증

7	**오류 처리**	n/a
8	**비고**	IdTokens는 실제 UID 값과의 충돌 가능성을 피하기 위해 비표준 UID 형식을 의도적으로 사용하는 (단일 사용) 가상 트랜잭션 인증 코드 또는 가상 RFID 토큰일 수 있습니다. 이러한 가상 단일 사용 type,IdTokens는 type *Central* 과 함께 전송되며 이러한 토큰을 캐시하거나 인증하는 것은 무의미합니다. 이 유즈케이스는 앱을 예로 사용하지만 이는 필수 사항이 아닙니다. 이 유즈케이스는 서버에서 생성한 IdToken이 있는 모든 RequestStartTransactionRequest에 유효합니다. 위의 시나리오 설명과 시퀀스 다이어그램은 시작 트랜잭션을 위한 구성 변수가 다음과 같이 구성되는 것을 기반으로 합니다. `TxStartPoint`: EVConnected, Authorized, DataSigned, PowerPathClosed, EnergyTransfer 이 유즈케이스는 다른 구성에도 유효하지만 트랜잭션이 다른 순간에 시작/중지될 수 있으며, 이때 메시지를 보내는 순서가 변경될 수 있습니다. 자세한 내용은 유즈케이스 E01 - Start Transaction options를 참조하세요. 이 유즈케이스는 구성 변수 AuthorizeRemoteStart가 *false* 라고 가정합니다. AuthorizeRemoteStart에 대한 요구 사항은 유즈케이스 F01 및 F02를 참조하세요. 다른 idTokenTypes도 원격으로 충전을 시작하는 데 사용할 수 있습니다. 앱에서 제공하는 사용자의 eMAID가 그 예입니다.

C05 - Authorization for CSMS initiated transactions Requirements

Table 68. C05 - Authorization for CSMS initiated transactions Requirements

ID	전제 조건	요구 사항 정의
C05.FR.01	충전기에서 IdTokenType이 Central 유형인 RequestStartTransactionRequest를 수신하는 경우.	충전기는 수신된 IdTokenType에 대해 AuthorizeRequest를 보내서는 안 됩니다.

ID	전제 조건	요구 사항 정의
C05.FR.02	충전기가 인증 캐시를 구현했고 충전기가 메시지에서 Central 유형의 IdTokenType에 대해 IdTokenInfo를 수신하는 경우	충전기는 인증 캐시에 정보를 저장해서는 안 됩니다.
C05.FR.03		RemoteStartId는 TransactionEventRequest에서 최소 한 번 제공되어야 합니다.
C05.FR.04		언어는 RFC-4646 태그로 지정해야 합니다. 참조: [RFC5646], 예: 미국 영어는 "en-US"입니다.
C05.FR.05		idToken은 RequestStartTransactionRequest 다음에 첫 번째 TransactionEventRequest에서도 한 번 제공해야 합니다.

C06 - Authorization using local id type

이것은 정보 제공적인 유즈케이스이며, Local ID 유형의 사용을 보여주는 것이 목적입니다.

Table 69. C06 - Authorization using local id type

번호	유형	설명
1	**이름**	로컬 ID 유형을 사용한 인증
2	**ID**	C06
	기능 블록	C. 인증
3	**목표**	충전기가 로컬에서 생성된 ID 토큰으로 충전을 시작할 수 있도록 합니다.
4	**설명**	충전기가 RFID가 없거나 RFID를 알 수 없는 운전자를 위해 트랜잭션을 시작해야하는 경우입니다. 예를 들어, EV 운전자가 주차 티켓을 사용하여 충전을 시작하는 경우.
	Actor	EV 운전자, 결제 단말기, CSMS, 충전기
	시나리오 설명	**1.** EV 운전자가 차고로 운전하여 입구의 차단기에서 주차 티켓을 받습니다. **2.** 충전기에 EV를 주차합니다. **3.** 충전 케이블을 꽂습니다. **4.** 충전기에서 주차권을 스캔/삽입하여 충전을 시작합니다. **5.** EV가 충전 중이고 운전자가 떠납니다. **6.** EV 운전자가 돌아와 주차권을 결제 키오스크에 삽입합니다. **7.** 주차 및 충전 비용을 지불합니다. **8.** 결제 단말기/키오스크가 CSMS를 통해 충전기로 정지 명령을 보냅니다. **9.** EV 운전자가 충전 케이블을 뽑고 출발합니다.
	대체 시나리오	C01 - EV Driver Authorization using RFID C02 - Authorization using a start button C03 - Authorization using credit/debit card C04 - Authorization using PIN-code C05 - Authorization for CSMS initiated transactions C07 - Authorization using Contract Certificates C08 - Authorization at EVSE using ISO 15118 External Identification Means (EIM) C15 - Unknown Offline Authorization
5	**필수 조건**	통합 주차 및 충전 결제 시스템
6	**사후 조건**	충전기에서 충전이 완료되었고 충전 정보가 CSMS에서 사용 가능합니다.

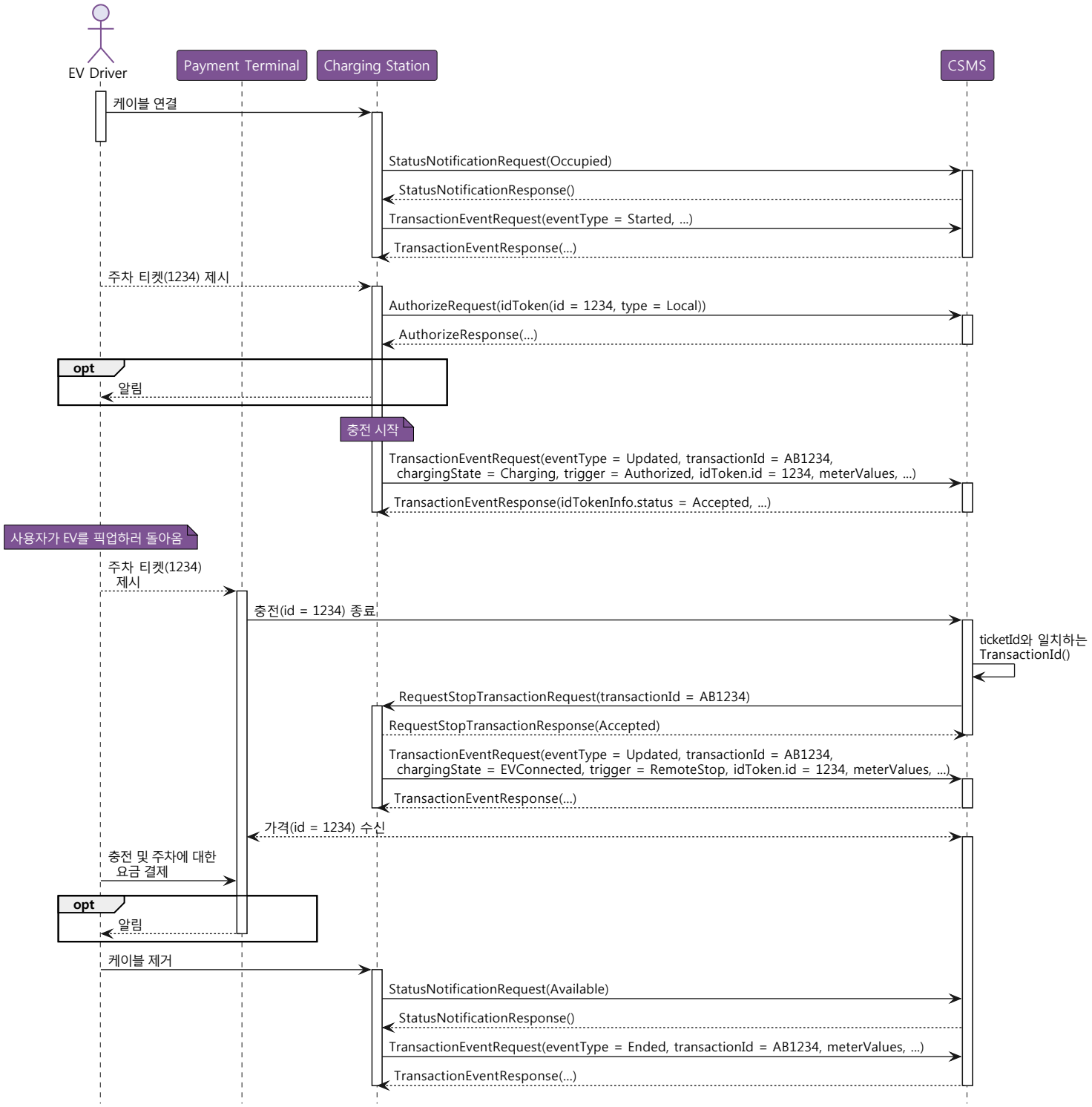

Figure 26. 시퀀스 다이어그램: 로컬 ID 유형을 사용한 승인

7	**오류 처리**	해당 없음
8	**비고**	이 유즈케이스에서는 주차권을 예로 사용하지만 이는 필수 사항이 아닙니다. 결제 단말기와 CSMS 간의 통신은 OCPP 범위를 벗어납니다. 위의 시나리오 설명과 시퀀스 다이어그램은 시작 및 중지 트랜잭션에 대한 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStartPoint`: Authorized, DataSigned, PowerPathClosed, EnergyTransfer `TxStopPoint`: ParkingBayOccupancy, EVConnected 이 유즈케이스는 다른 구성에도 유효하지만 트랜잭션이 다른 순간에 시작/중지될 수 있으며, 이때 메시지를 보내는 순서가 변경될 수 있습니다. 자세한 내용은 다음 유즈케이스를 참조하세요: E01 - Start Transaction options 및 E06 - Stop Transaction options.

C06 - Authorization using local id type - Requirements

Table 70. C06 - Authorization using local id type - Requirements

ID	전제 조건	요구 사항 정의
C06.FR.01		충전기는 인증 후에만 에너지를 제공해야 합니다.
C06.FR.02	EV 운전자가 Local 유형의 IdTokenType을 제시하는 경우.	충전기는 AuthorizeRequest를 CSMS로 보내 인증을 요청해야 합니다.
C06.FR.03		AuthorizeRequest는 충전 식별자의 인증에만 사용해야 합니다.
C06.FR.04		CSMS가 AuthorizeRequest를 수신하면 AuthorizeResponse로 응답해야 하며 수락 또는 거부 이유를 나타내는 인증 상태 값을 포함해야 합니다.

2.2. ISO 15118 Authorization

이 인증 섹션은 Plug & Charge 기능 사용을 위해 ISO15118-1에서 유래되었습니다.

C07 - Authorization using Contract Certificates

Table 71. C07 - Authorization using Contract Certificates

번호	유형	설명
1	**이름**	계약 인증서를 사용한 인증
2	**ID**	C07
	기능 블록	C. 인증
	참조	ISO15118-1 D2
3	**목표**	ISO15118-1 참조, 유즈케이스 목표 D2, 26페이지.
4	**설명**	ISO15118-1 참조, 유즈케이스 설명 D2(첫 번째 글머리 기호), 26페이지.
	Actors	Actors: EV, 충전기, CSMS, OCSP
	시나리오 설명	**15118**: ISO15118-1 참조, 유즈케이스 설명 D2, 시나리오 설명, 첫 2개의 글머리 기호, 26페이지. **OCPP**: **3.** 충전기는 AuthorizeRequest 메시지를 CSMS로 보내는데, 여기에는 계약 인증서 및 인증서 체인과 관련된 OCSP 요청에 필요한 eMAID 및 데이터가 포함됩니다. **4.** CSMS는 인증서 상태값을 동의 또는 비동의로 응답합니다. **5.** ID의 성공적인 인증 후 서비스가 시작됩니다.
	대체 시나리오	C01 - EV Driver Authorization using RFID C02 - Authorization using a start button C03 - Authorization using credit/debit card C04 - Authorization using PIN-code C05 - Authorization for CSMS initiated transactions C06 - Authorization using local id type C08 - Authorization at EVSE using ISO 15118 External Identification Means (EIM) C15 - Unknown Offline Authorization
5	**필수 조건**	EV에 계약 인증서가 설치되어 있습니다.
6	**사후 조건**	계약 인증서의 유효성이 결정됩니다.

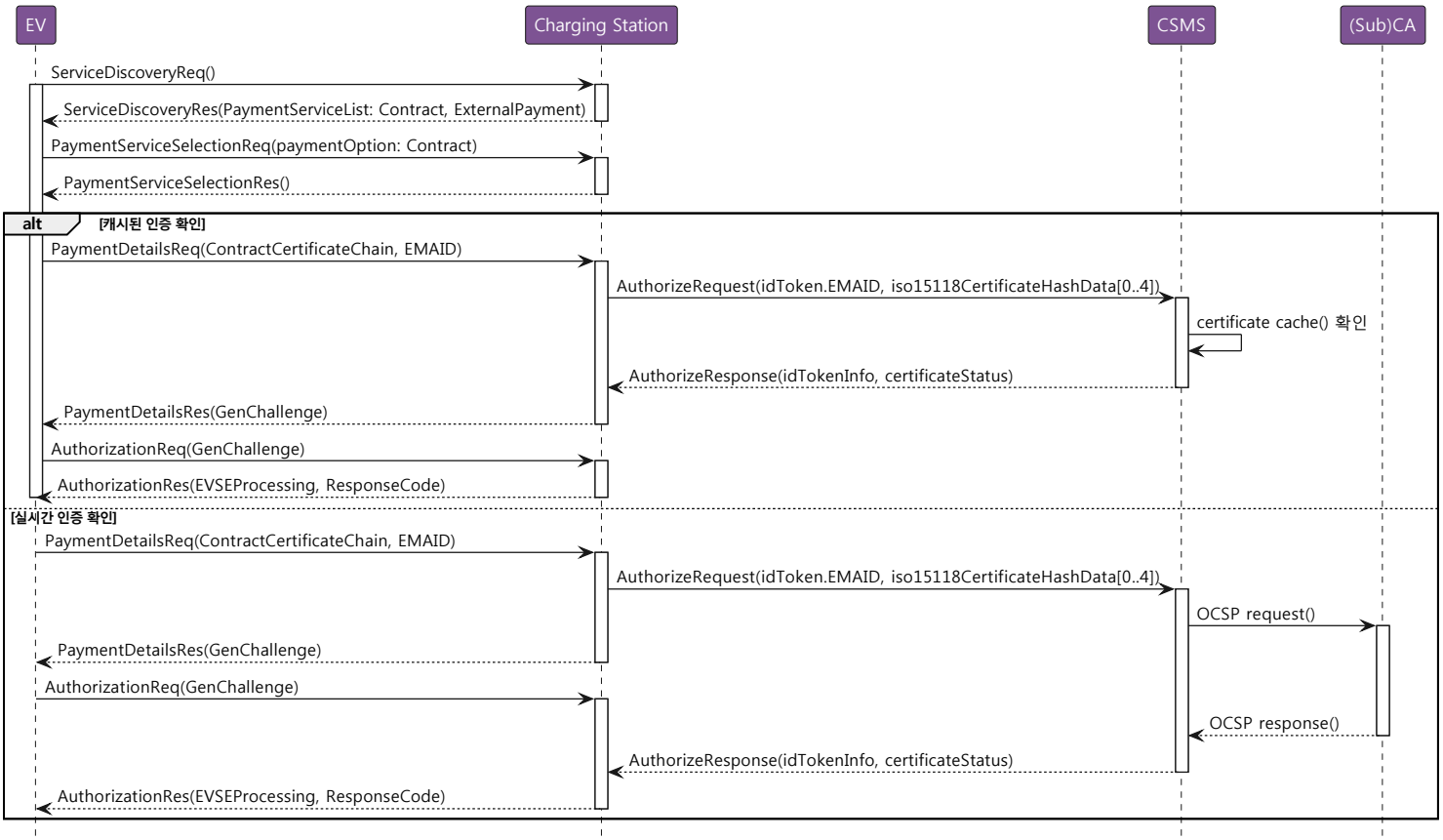

Figure 27. 계약 인증서를 사용한 승인

7	**오류 처리**
	15118의 1번째 버전에서 PaymentDetailsReq/Res 메시지의 메시지 시간 초과는 5초입니다. 해당 시간 내에 인증서 검증을 완료할 수 없는 경우 AuthorizationReq/Res 동안 완료할 수 있으며, 최대 60초까지 연장할 수 있습니다.
8	**참고 사항**
	충전기가 오프라인인 경우 인증서 상태를 확인할 수 없기 때문에 ServiceDiscoveryRes에서 ISO 15118 계약 인증서에 대한 지불 옵션을 생략하고 외부 식별 수단(유즈케이스 C08)으로 되돌리는 것이 좋습니다.

C07 - Authorization using Contract Certificates - Requirements

Table 72. C07 - Requirements

ID	전제 조건	요구 사항 정의	참고
C07.FR.01	충전기가 온라인일 때	충전기는 AuthorizeRequest를 CSMS에 보내서 검증을 받아야 합니다.	
C07.FR.02	C07.FR.01	AuthorizeRequest에는 계약 인증서 및 인증서 체인과 관련된 OCSP 요청에 필요한 eMAID 및 데이터가 포함되어야 합니다.	
C07.FR.04	CSMS가 AuthorizeRequest를 수신하는 경우.	AuthorizeResponse로 응답하고 수락 또는 거부 이유를 나타내는 인증 상태 값을 포함해야 합니다.	
C07.FR.05	C07.FR.02	CSMS는 실시간 또는 캐시된 OCSP 데이터를 통해 인증서 및 인증서 체인의 유효성을 확인해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
C07.FR.06	C07.FR.01 AND 충전기가 연관된 루트 인증서가 없어서 계약 인증서를 검증할 수 없고 `CentralContractValidationAllowed`가 *true* 인 경우	CSMS에서 검증하기 위해 충전기는 AuthorizeRequest의 *certificate* 속성(PEM 형식)에서 계약 인증서 체인을 CSMS에 전달해야 합니다.	
C07.FR.07	충전기가 오프라인이고 `ContractValidationOffline`가 *false* 인 경우	충전기는 충전을 허용하지 않아야 합니다.	
C07.FR.08	충전기가 오프라인이고 `ContractValidationOffline`이 *true* 인 경우	충전기는 계약 인증서를 로컬에서 검증하려고 시도해야 합니다.	
C07.FR.09	C07.FR.08 AND 계약 증명서가 유효하고 LocalAuthorizeOffline이 *true* 인 경우.	충전기는 Local Authorization List 또는 Authorization Cache에서 eMAID를 조회해야 합니다.	
C07.FR.10	C07.FR.09 AND Local Authorization List에서 eMAID를 찾은 경우.	충전기는 유즈케이스 C13 - Offline Authorization through Local Authorization List에 따라 동작해야 합니다.	
C07.FR.11	C07.FR.09 및 Authorization Cache에서 eMAID가 발견된 경우	충전기는 유즈케이스 C12 - Start Transaction - Cached Id에 따라 동작해야 합니다.	
C07.FR.12	C07.FR.09 및 eMAID가 발견되지 않고 `OfflineTxForUnknownIdEnabled` = *true* 인 경우	충전기는 유즈케이스 C15 - Offline Authorization of unknown Id에 따라 충전을 허용해야 합니다.	
C07.FR.13	C07.FR.04 AND 인증서 체인(*certificate* 또는 *iso15118CertificateHashData* 에 제공됨)이 유효하고 AND *idToken* 의 인증 상태가 `Blocked`, `Expired`, `Invalid`, `Unknown` 중 하나인 경우.	CSMS는 *certificateStatus* = `ContractCancelled` 및 *idTokenInfo.status* 의 인증 상태를 포함하는 AuthorizationResponse를 반환해야 합니다.	인증서는 유효하지만 EMAID는 허용되지 않습니다.

ID	전제 조건	요구 사항 정의	참고
C07.FR.14	C07.FR.04 AND 인증서 체인(*certificate* 또는 *iso15118CertificateHashData* 에 제공됨)이 유효하고 AND *idToken* 의 인증 상태가 `Blocked`, `Expired`, `Invalid`, `Unknown` 중 하나가 아닌 경우	CSMS는 *certificateStatus* = `Accepted` 및 *idTokenInfo.status* 의 인증 상태를 포함하는 AuthorizationResponse를 반환해야 합니다.	*idTokenInfo.status* 가 `Accepted` 가 아닌 경우(예: `ConcurrentTx` 또는 `NotAtThisLocation`)에는 여전히 충전이 허락되지 않습니다.
C07.FR.15	C07.FR.04 AND 인증서 체인(*certificate* 또는 *iso15118CertificateHashData* 에서 제공)이 만료되었습니다.	CSMS는 *certificateStatus* = `CertificateExpired` 및 *idTokenInfo.status* = `Expired` 를 포함하는 AuthorizationResponse를 반환해야 합니다.	인증서가 만료된 경우 *idToken* 의 상태도 expired로 보고됩니다.
C07.FR.16	C07.FR.04 AND 인증서 체인(*certificate* 또는 *iso15118CertificateHashData* 에 제공됨)이 해지되었습니다.	CSMS는 *certificateStatus* = `CertificateRevoked` 및 *idTokenInfo.status* = `Invalid` 를 포함하는 AuthorizationResponse를 반환해야 합니다.	인증서가 해지된 경우 *idToken* 의 상태가 invalid로 보고됩니다.
C07.FR.17	C07.FR.04 AND 인증서 체인(*certificate* 또는 *iso15118CertificateHashData* 에 제공됨)을 확인할 수 없거나 invalid일 경우	CSMS는 *certificateStatus* = `CertChainError` 및 *idTokenInfo.status* = `Invalid` 를 포함하는 AuthorizationResponse를 반환해야 함	인증서를 확인할 수 없는 경우 *idToken* 의 상태가 invalid로 보고됨

C08 - Authorization at EVSE using ISO 15118 External Identification Means (EIM)

Table 73. C08 - Authorization at EVSE using ISO 15118 External Identification Means (EIM)

번호	유형	설명
1	**이름**	ISO 15118 외부 식별 수단(EIM)을 사용한 EVSE에서의 승인
2	**ID**	C08 / 15118-1 D4
	기능 블록	C. 인증
	참조	ISO15118-1 D4
3	**목표**	CSMS의 도움을 받아 충전기를 통해 EV를 인증합니다. 또한 ISO15118-1, 유즈케이스 목표 D4, 28페이지를 참조하세요.
4	**설명**	충전기는 EV에서 제공한 정보에 따라 AuthorizeRequest 메시지를 보냅니다. 또한 ISO15118-1, 유즈케이스 설명 D4에서 "참고"까지, 28페이지를 참조하세요.
	Actors	EV, 충전기, CSMS
	시나리오 설명	**15118** ISO15118-1, 유즈케이스 설명(시나리오 설명) D4, 28페이지를 참조하세요. **OCPP** **1.** 충전기는 idToken에 외부 식별 수단(EIM)이 포함하여 AuthorizeRequest를 보냅니다. **2.** CSMS는 AuthorizeResponse로 응답합니다.
	대체 시나리오	C01 - EV Driver Authorization using RFID C02 - Authorization using a start button C03 - Authorization using credit/debit card C04 - Authorization using PIN-code C05 - Authorization for CSMS initiated transactions C06 - Authorization using local id type C07 - Authorization using Contract Certificates C15 - Unknown Offline Authorization
5	**필수 조건**	EV와 EVSE 간의 통신은 성공적으로 성립되어야 합니다.
6	**사후 조건**	인증이 성공적입니다. 또한 ISO15118-1, 유즈케이스 종료 조건 D4, 28페이지를 참조하세요.

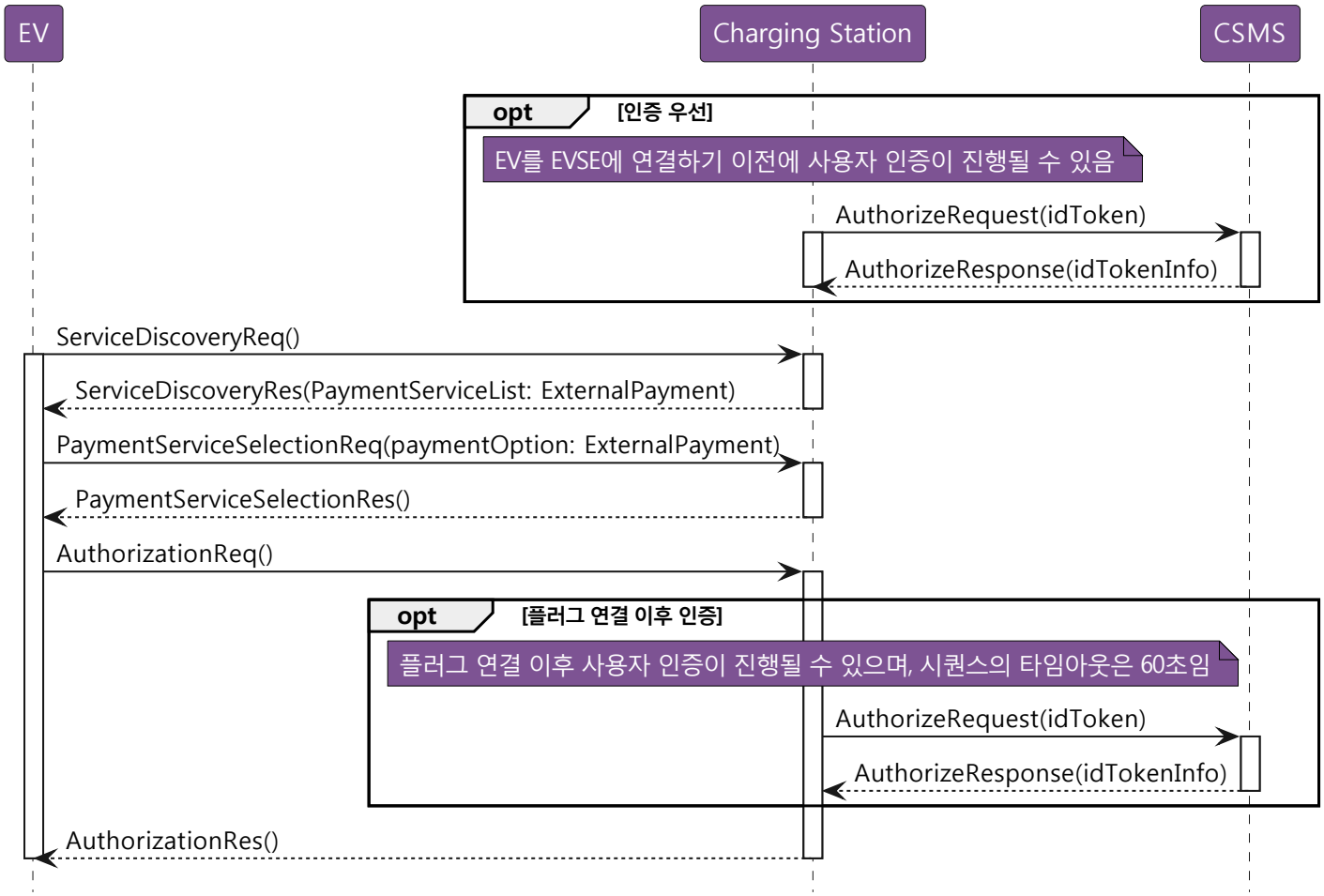

Figure 28. 시퀀스 다이어그램: SA의 도움으로 수행된 외부 자격 증명을 사용한 EVSE에서의 인증.

7	**참고 사항**	이전 섹션에서 언급한 모든 식별 수단은 이 유즈케이스에 적용될 수 있습니다. 유일한 차이점은 15118 통신의 가용성입니다.

출처: ISO15118-1

C08 - Authorization at EVSE using ISO 15118 External Identification Means (EIM) - Requirements

Table 74. C08 - Requirements

ID	전제 조건	요구 사항 정의
C08.FR.01		충전기는 확인을 위해 CSMS에 식별 정보를 보내야 합니다.
C08.FR.02		EV 운전자는 EV를 EVSE에 연결한 후 특정 시간 내에 승인을 활성화해야 하거나 EVSE는 HMI를 통해 식별 프로세스를 다시 시작하도록 승인해야 합니다.

2.3. GroupId

C09 - Authorization by GroupId

Table 75. C09 - Authorization by GroupId

번호	유형	설명
1	**이름**	GroupId에 의한 인증
2	**ID**	C09
	기능 블록	C. 인증
3	**목표**	다른 IdToken을 가진 2명의 EV 운전자가 동일한 GroupId를 사용한 인증을 진행할 수 있도록 합니다.
4	**설명**	이 유즈케이스는 충전기가 GroupId 정보를 기반으로 EV 운전자의 작업을 어떻게 승인할 수 있는지 다룹니다. 예를 들어 2명이 정기적으로 동일한 EV를 사용하는 경우 이를 사용할 수 있습니다. 즉, 각자의 IdToken(예: RFID 카드)을 사용할 수 있으며, 다른 idToken(동일한 GroupId)으로 시작된 트랜잭션을 승인 취소할 수 있습니다.
	Actors	충전기, CSMS, EV Driver1, EV Driver2
	시나리오 설명	**1.** EV Driver 1이 IdToken을 제시합니다. **2.** 충전기가 AuthorizeRequest를 CSMS로 보내 승인을 요청합니다. **3.** AuthorizeRequest를 수신하면 CSMS가 AuthorizeResponse로 응답합니다. 이 응답 메시지에는 GroupId가 포함됩니다. **4.** 충전기는 EV 드라이버 1의 인증 정보와 함께 GroupIdToken을 저장합니다. **5.** EV 드라이버 2는 IdToken을 제시합니다. **6.** 충전기는 AuthorizeRequest를 CSMS로 보내 인증을 요청합니다. **7.** AuthorizeRequest를 수신하면 CSMS는 AuthorizeResponse로 응답합니다. 이 응답 메시지에는 GroupId가 포함됩니다. **8.** 두 응답의 일치하는 GroupId 정보를 기반으로 충전기는 작업을 승인합니다.
5	**필수 조건**	EV 드라이버 1과 EV 드라이버 2는 동일한 GroupId를 갖습니다.
6	**사후 조건**	충전기는 GroupId를 알고 있습니다.

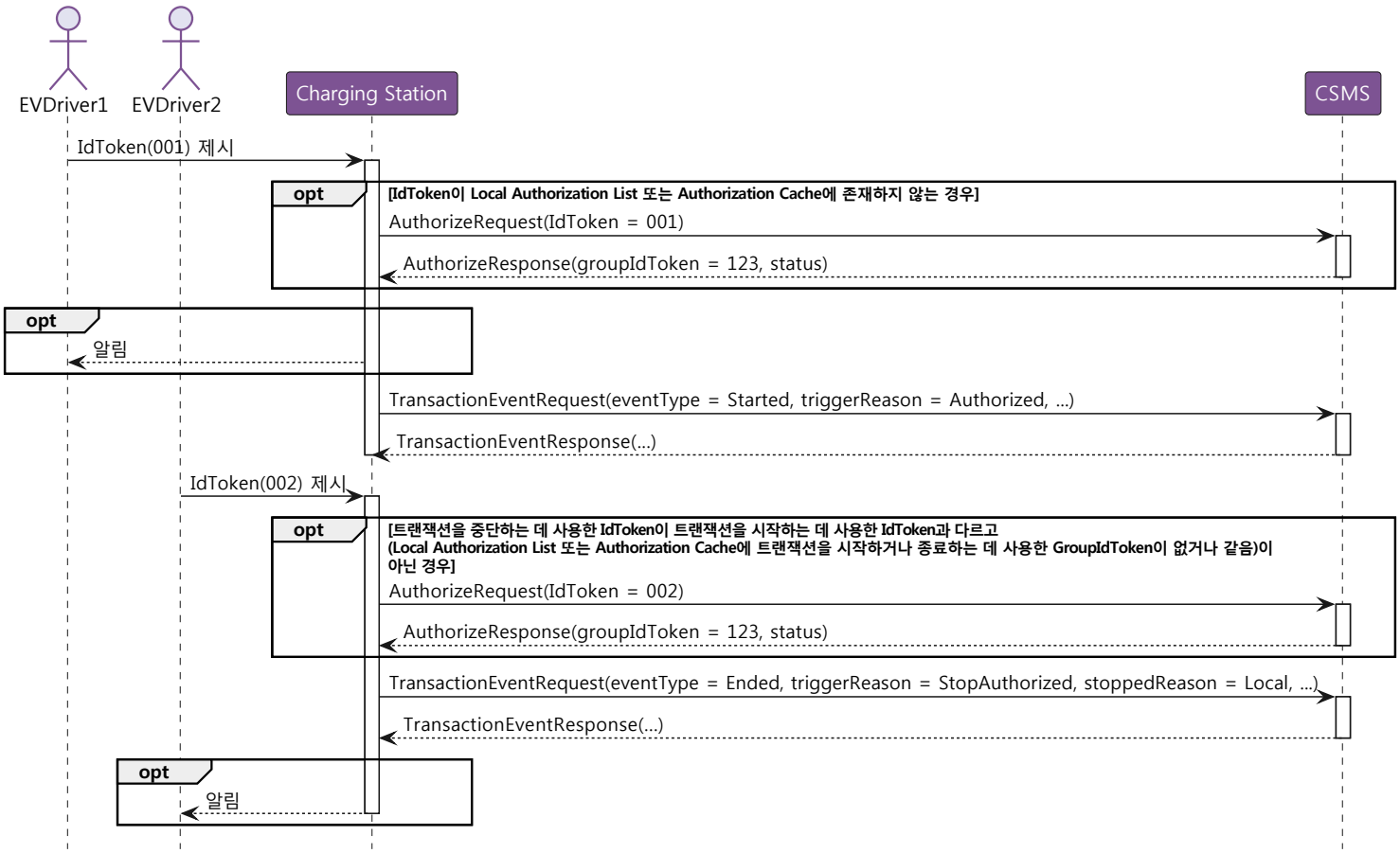

Figure 29. 시퀀스 다이어그램: GroupId에 의한 인증

7	**오류 처리**	n/a
8	**비고**	IdTokenType 데이터는 groupId로 사용되며, 계정에 속한 idToken 중 하나를 사용하는 대신 GroupId에 대한 공유 중앙 계정 식별자를 사용하는 경우가 많습니다. 이 유즈케이스에서 설명한 groupId 메커니즘은 groupId가 캐시에 저장되므로 인증 캐시를 사용할 때도 작동합니다.

C09 - Authorization by GroupId - Requirements

Table 76. C09 - Requirements

ID	전제 조건	요구 사항 정의
C09.FR.02		인증 목적으로 동일한 그룹에 속하는 IdToken은 IdTokenInfo의 선택적 *groupIdToken* 요소에 공통 그룹 식별자를 가져야 합니다.
C09.FR.03	특정 IdToken으로 트랜잭션이 승인/시작된 경우.	다른 유효한 IdToken을 가지고 있지만 동일한 groupIdToken을 가진 EV 운전자는 트랜잭션을 중지할 수 있는 권한이 있어야 합니다.
C09.FR.04	C09.FR.03 AND 두 IdToken과 해당 GroupIdToken이 Local Authorization List 또는 Authorization Cache에 모두 있는 경우.	충전기는 AuthorizeRequest를 CSMS로 보낼 수 있습니다.(MAY)

ID	전제 조건	요구 사항 정의
C09.FR.05	C09.FR.03 AND (C09.FR.07 아님) AND 새로 제시된 IdToken과 해당 GroupIdToken이 Local Authorization List 또는 Authorization Cache에 없는 경우.	충전기는 AuthorizeRequest를 CSMS로 보내야 합니다.
C09.FR.07	idToken이 인증된 충전 중에 제시되고 (a) 제시된 idToken이 승인을 시작한 idToken과 동일한 경우 또는 (b) 제시된 idToken이 로컬 인증 목록 또는 인증 캐시에 있고 유효하며 승인을 시작한 IdToken과 동일한 GroupIdToken을 갖는 경우	충전기는 먼저 AuthorizeRequest를 보내지 않고 충전 승인을 종료해야 합니다.
C09.FR.09	AuthorizeRequest의 IdToken에 연관된 groupIdToken이 있는 경우	CSMS의 AuthorizeResponse에는 groupIdToken이 포함되어야 합니다.
C09.FR.10		AuthorizeResponse에는 수락 또는 거부 이유를 나타내는 승인 상태 값이 포함되어야 합니다.
C09.FR.11	C09.FR.03 AND 동일한 GroupIdToken을 가지고 있지만 status = `Accepted` 가 아닌 다른 IdToken이 중지를 위해 제시된 경우	충전기는 트랜잭션을 중지해서는 안 됩니다.
C09.FR.12	TransactionEventRequest에 IdToken이 포함되어 있고 idToken에 연관된 groupIdToken이 있는 경우	CSMS의 TransactionEventResponse에는 groupIdToken이 포함되어야 합니다.

2.4. Authorization Cache

C10 - Store Authorization Data in the Authorization Cache

Table 77. C10 - Store Authorization Data in the Authorization Cache

번호	유형	설명
1	**이름**	인증 캐시에 인증 데이터를 저장합니다.
2	**ID**	C10
	기능 블록	C. 인증
3	**목표**	인증 캐시에 가장 최근에 수신된 모든 IdToken을 저장합니다.
4	**설명**	이 유즈케이스는 충전기가 CSMS에서 성공적으로 인증된 이전에 제시된 식별자의 기록을 인증 캐시에 자율적으로 저장하는 방법을 다룹니다. (성공적인 의미: IdToken이 포함된 메시지에 대한 응답 수신)
	Actors	충전기, CSMS
	시나리오 설명	**1.** 충전기는 CSMS로부터 AuthorizeResponse, ReserveNowRequest 또는 TransactionEventResponse 응답 메시지를 수신합니다. **2.** 충전기는 CSMS의 응답 메시지에서 수신된 모든 IdTokenInfo를 사용하여 캐시를 업데이트합니다.
	대체 시나리오	해당 없음
5	**필수 조건**	인증 캐시가 구현되고 `AuthCacheEnabled` 구성 변수의 값이 'true' 로 설정됩니다.
6	**사후 조건**	**성공한 사후 조건:** 충전기는 새로 수신한 IdTokenInfo 데이터를 인증 캐시에 저장합니다. **실패한 사후 조건:** 충전기는 인증 캐시를 저장할 수 *없었습니다*.

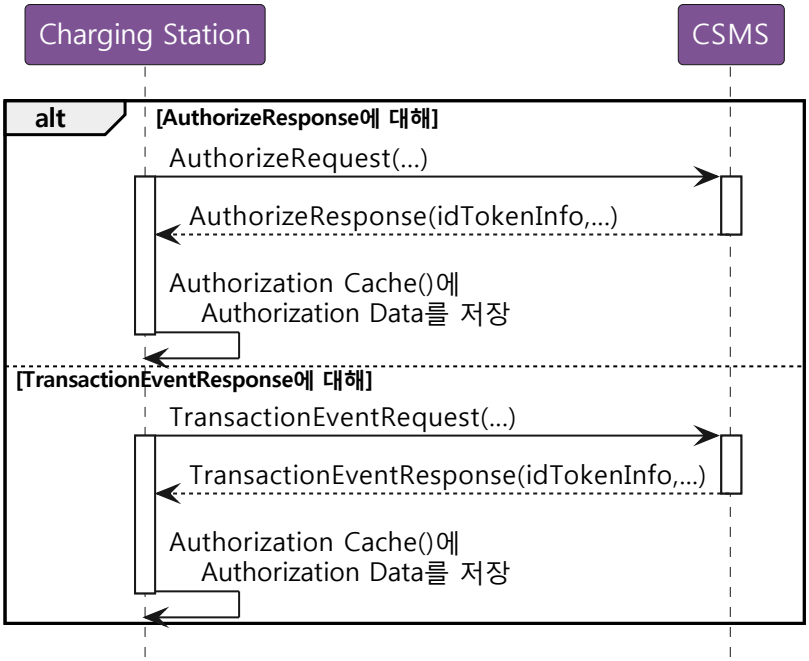

Figure 30. 시퀀스 다이어그램: 인증 캐시에 인증 데이터 저장

7	**오류 처리**	n/a
8	**비고**	n/a

C10 - Store Authorization Data in the Authorization Cache - Requirements

Table 78. C10 - Requirements

ID	전제 조건	요구 사항 정의	참고
C10.FR.01		인증 캐시는 모든 최신 수신 식별자를 포함해야 합니다(상태와 관계없이).	
C10.FR.02		캐시 값은 재부팅 및 정전 시에도 지속되어야 합니다.	따라서 캐시 값은 비휘발성 메모리에 저장되어야 합니다.
C10.FR.03	*Accepted* 가 아닌 상태로 인증 캐시에 저장된 IdToken이 제시되고 충전기가 온라인인 경우.	AuthorizeRequest는 IdToken의 현재 상태를 확인하기 위해 CSMS로 전송되어야 합니다.	식별자의 현재 상태를 확인.
C10.FR.04	AuthorizeResponse를 수신.	충전기는 인증 캐시 항목을 업데이트해야 합니다.	업데이트는 Authorization Cache에서 설명한 대로 응답의 IdTokenInfo 값으로 수행해야 합니다.
C10.FR.05	TransactionEventResponse를 수신.	충전기는 인증 캐시 항목을 업데이트해야 합니다.	업데이트는 Authorization Cache에서 설명한 대로 응답의 IdTokenInfo 값으로 수행해야 합니다.
C10.FR.07		충전기는 캐시 항목이 가득 찼을 때에도 이전 항목을 삭제하여 새 캐시 항목을 수락하는 메커니즘을 가져야 합니다.	먼저 *Accepted* 가 아닌 상태를 가진 항목을 제거한 다음 가장 오래된 유효한 항목을 제거하여 새 항목을 위한 공간을 확보하는 것이 권장됩니다.
C10.FR.08	IdTokenInfoType에 *cacheExpiryDateTime* 값이 없는 경우	토큰이 캐시에 있는 것으로 간주되는 시간은 구성 변수 `AuthCacheLifeTime`에 의해 결정됩니다. 이 변수는 토큰이 마지막으로 사용된 이후 인증 캐시에서 만료될 때까지 걸리는 시간을 나타냅니다.	캐시의 이 만료는 **IdToken에 설정된 만료 날짜(예: RFID 카드 만료 날짜)와 같지 않습니다**.
C10.FR.09	Tariff & Cost를 지원하는 충전기	충전기는 캐시에 tariff 정보를 저장해서는 안 됩니다.	
C10.FR.10	인증 캐시 항목의 유효 기간이 만료되는 경우.	인증 캐시 항목은 캐시에서 제거되거나 `Expired` 으로 변경되어야 합니다.	
C10.FR.11		인증 캐시가 활성화되거나 비활성화되는지 여부는 `AuthCacheEnabled` 구성 변수에 의해 제어되어야 합니다.	
C10.FR.12		개인 정보를 인증 캐시에 안전하게 저장하는 것을 권장합니다.	예시로, 캐시에 해시된 idToken만 저장합니다.
C10.FR.13	IdTokenInfoType에 *cacheExpiryDateTime* 값이 포함된 경우	토큰이 캐시에 있는 것으로 간주되는 시간은 *cacheExpiryDateTime* 에 의해 결정됩니다. 이 변수는 토큰이 인증 캐시에서 만료되는 날짜와 시간을 나타냅니다.	캐시의 이 만료는 **IDToken에 설정된 만료 날짜**(예: RFID 카드 만료 날짜)와 **같지 않습니다**.

C11 - Clear Authorization Data in Authorization Cache

Table 79. C11 - Clear Authorization Data in Authorization Cache

번호	유형	설명
1	**이름**	인증 캐시의 인증 데이터 지우기
2	**ID**	C11
	기능 블록	C. 인증
3	**목표**	인증 캐시의 모든 IDToken을 지웁니다.
4	**설명**	이 유즈케이스는 CSMS가 충전기에 인증 캐시를 지우도록 요청하는 방법을 다룹니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS가 충전기에 ClearCacheRequest를 보내 인증 캐시를 지우도록 요청합니다. **2.** 충전기가 *Accepted* 상태로 응답합니다.
5	**필수 조건**	인증 캐시는 `AuthCacheEnabled` 구성 변수에 의해 지원되고 활성화됩니다.
6	**사후 조건**	**성공한 사후 조건:** 충전기가 인증 캐시를 *성공적으로* 지웠습니다. **실패한 사후 조건:** 충전기가 인증 캐시를 지울 수 *없었습니다.*

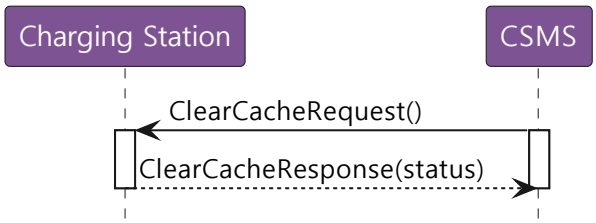

Figure 31. 시퀀스 다이어그램: 인증 캐시에서 인증 데이터 지우기

7	**오류 처리**	n/a
8	**비고**	n/a

C11 - Clear Authorization Data in Authorization Cache - Requirements

Table 80. C11 - Requirements

ID	전제 조건	요구 사항 정의
C11.FR.01	CSMS가 ClearCacheRequest를 보내는 경우.	충전기는 인증 캐시를 지우려고 시도해야 합니다.
C11.FR.02	C11.FR.01	충전기는 인증 캐시를 지울 수 있는지 여부를 나타내는 ClearCacheResponse 메시지를 보내야 합니다.

ID	전제 조건	요구 사항 정의
C11.FR.03	C11.FR.02 AND 충전기가 인증 캐시를 성공적으로 지웠습니다.	충전기는 ClearCacheResponse 메시지를 *Accepted* 상태로 보냅니다.
C11.FR.04	C11.FR.02 AND 구성 변수 `AuthCacheEnabled` 는 false입니다.	충전기는 ClearCacheResponse 메시지를 *Rejected* 상태로 보냅니다.
C11.FR.05	C11.FR.02 AND 충전기가 인증 캐시를 지우지 못했습니다.	충전기는 ClearCacheResponse 메시지를 *Rejected* 상태로 보내야 합니다.

C12 - Start Transaction - Cached Id

Table 81. C12 - Start Transaction - Cached Id

번호	유형	설명
1	**이름**	충전 시작 - 캐시된 ID
2	**ID**	C12
	기능 블록	C. 인증
3	**목표**	EV 운전자가 인증 캐시를 사용하여 온라인에서 트랜잭션을 시작할 수 있도록 합니다. 그러면 충전기가 더 빨리 응답할 수 있습니다. AuthorizeRequest가 전송되지 않기 때문입니다.
4	**설명**	이 유즈케이스는 충전기가 캐시된 IdToken을 사용하는 동안 트랜잭션을 시작하도록 EV 운전자를 인증하는 방법을 설명합니다.
	Actors	충전기, CSMS, EV 운전자
	시나리오 설명	**1.** EV 운전자가 케이블을 꽂습니다. **2.** 충전기가 트랜잭션을 시작합니다. **3.** EV 운전자가 IdToken을 제시합니다. **4.** 충전기가 Authorization Cache로 IdToken을 확인합니다. **5.** 충전기가 트랜잭션을 업데이트합니다. **6.** 충전기가 충전을 시작합니다. **7.** E02 - Start Transaction - Cable Plugin First가 적용됩니다.
5	**필수 조건**	`AuthCacheEnabled` = true `LocalPreAuthorize` = true EV 드라이버의 ID가 Authorization Cache에 캐시됨 ID가 유효함
6	**사후 조건**	**성공한 사후 조건:** EV 드라이버가 인증 캐시를 사용하여 트랜잭션을 시작할 권한이 부여됨. **실패 사후 조건:** Authorization Cache에서 사용자 ID를 찾을 수 없음: * 온라인 충전기: 충전기가 AuthorizeRequest를 CSMS에 요청하고 이 역시 실패함. * 오프라인 상황에서 충전기의 동작은 구성 변수 OfflineTxForUnknownIdEnabled에 의해 정의됩니다.

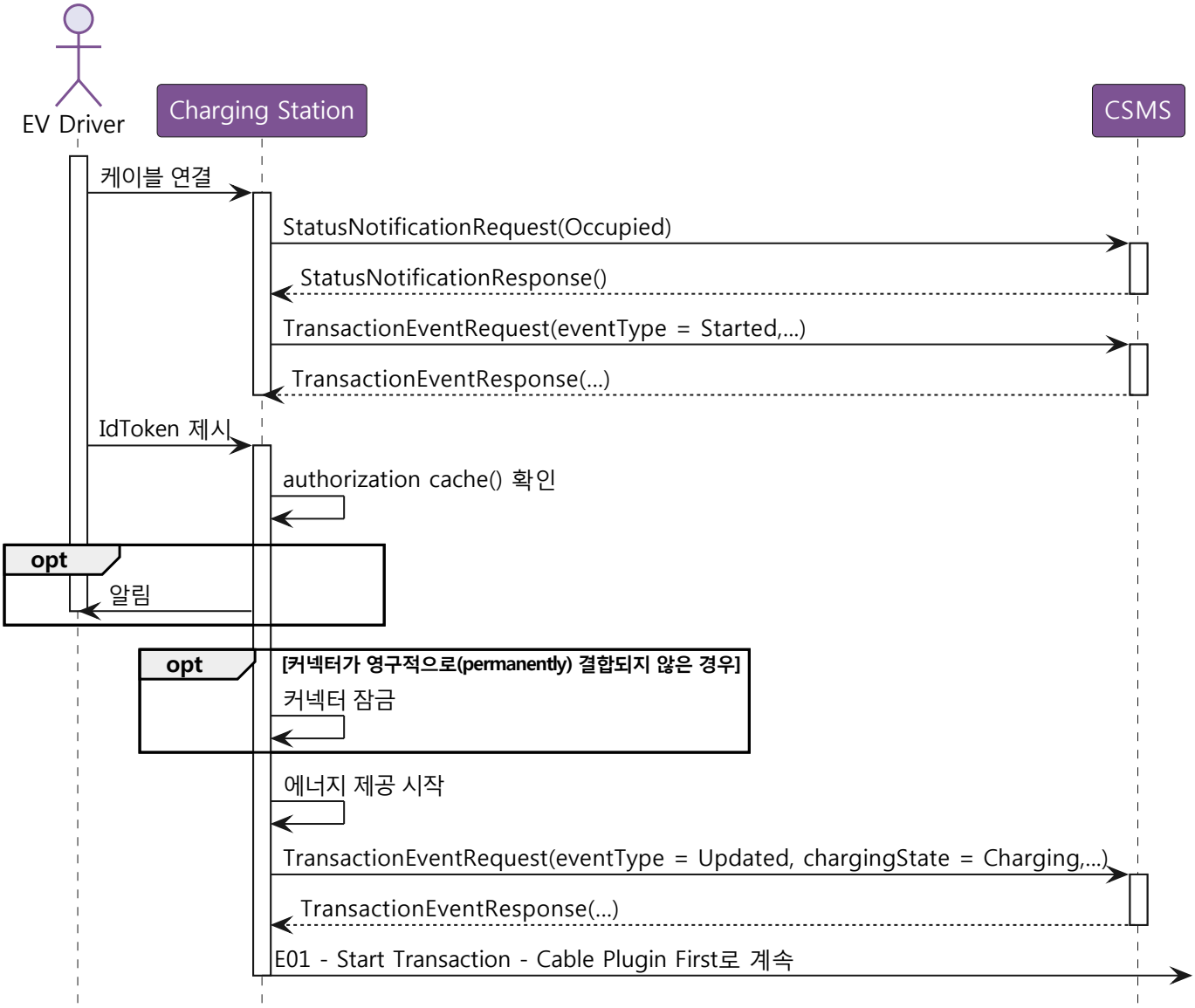

Figure 32. 시퀀스 다이어그램: 시작 트랜잭션 - 캐시된 ID

7	**오류 처리**	충전기 인증 캐시에 IdToken이 있고 인증 캐시에서는 유효하지만 CSMS에서는 더 이상 유효하지 않은 경우: 충전기는 TransactionEventResponse에서 IdTokenInfo를 수신하며 여기에는 최신 Invalid 상태가 포함됩니다. 이런 경우 발생하는 일은 구성 변수 `MaxEnergyOnInvalidId` 및 `StopTxOnInvalidId`에 따라 달라집니다.
8	**참고**	충전기에서 인증 캐시를 구현한 경우 AuthorizeResponse 메시지를 수신하면 충전기가 캐시 항목을 업데이트합니다. 캐시된 유효한 IdToken의 경우 AuthorizeRequest를 보내는 것은 논리적이지 않습니다. TransactionEventResponse 메시지에도 IdToken 정보가 포함되어 있습니다. IdToken이 더 이상 유효하지 않게 되면 충전기는 이 TransactionEventResponse에서 이를 알게 됩니다. 따라서 IdToken이 더 이상 유효하지 않으면 충전기에서 에너지 제공을 중단하기로 결정할 수 있으며 구성에 따라 트랜잭션을 중단할 수도 있습니다. 위의 시나리오 설명과 시퀀스 다이어그램은 시작 트랜잭션을 위한 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStartPoint`: EVConnected, Authorized, DataSigned, PowerPathClosed, EnergyTransfer 이 유즈케이스는 다른 구성에도 유효하지만 트랜잭션이 다른 순간에 시작/중지될 수 있으며, 이때 메시지가 전송되는 순서가 변경될 수 있습니다. 자세한 내용은 유즈케이스를 참조하세요. E01 - Start Transaction options.

C12 - Start Transaction - Cached Id - Requirements

Table 82. C12 - Requirements

ID	전제 조건	요구 사항 정의	참고
C12.FR.02	인증 캐시에 *Accepted* 으로 저장된 식별자가 제시될 때.	충전기는 idToken이 포함된 TransactionEventRequest를 CSMS로 보내야 합니다.	
C12.FR.03	C12.FR.02	CSMS는 이 TransactionEventRequest를 처리할 때 IdToken의 인증 상태를 확인해야 합니다.	
C12.FR.04	C12.FR.02 AND 케이블이 연결되어 있습니다.	충전기는 에너지 제공을 시작해야 합니다.	
C12.FR.05	*Accepted* 가 아닌 상태로 인증 캐시에 저장된 식별자가 제시되고 충전기가 온라인 상태인 경우.	충전기는 CSMS에 AuthorizeRequest를 보내야 합니다.	식별자의 현재 상태를 확인하기 위해.
C12.FR.06	캐시의 식별자에 대해 IdTokenInfo를 수신한 경우.	인증 캐시는 수신된 IdTokenInfo를 사용하여 업데이트해야 합니다.	
C12.FR.09	`MasterPassGroupId` 와 같은 groupId를 갖는 IdToken	트랜잭션을 시작할 수 없습니다.	

2.5. Local Authorization list

C13 - Offline Authorization through Local Authorization List

Table 83. C13 - Offline Authorization through Local Authorization List

번호	유형	설명
1	**이름**	로컬 인증 목록을 통한 오프라인 인증
2	**ID**	C13
	기능 블록	C. 인증
3	**목표**	오프라인 상태에서 Local Authorization List를 사용하여 idToken을 인증합니다.
4	**설명**	이 유즈케이스는 CSMS와 통신할 수 없는 경우 IdToken을 인증하는 방법을 설명합니다. Local Authorization List는 CSMS와 동기화할 수 있는 idToken 목록입니다. 이 목록에는 CSMS에서 관리하는 선택된 idToken 세트의 인증 상태가 포함됩니다.
	Actors	EV 운전자, 충전기

번호	유형	설명
	시나리오 설명	**1.** 충전기가 *Offline* 입니다. **2.** EV 운전자가 IdToken을 제시합니다. **3.** 충전기는 IdToken이 알려져 있고 Local Authorization List에서 *Accepted* 상태인지 확인합니다. **4.** 충전기가 충전을 시작합니다.
5	**필수 조건**	*Local Authorization List* 을 사용할 수 있음 *Local Authorization List* 이 `LocalAuthListEnabled`를 통해 활성화됨 충전기가 *Offline* 상태임 EV 운전자의 ID가 *Local Authorization List* 에 있음 ID가 유효함
6	**사후 조건**	**성공 사후 조건:** 충전기가 오프라인일 때 Local Authorization List의 토큰을 허용합니다. **실패 사후 조건:** 충전기가 오프라인일 때 Local Authorization List의 토큰을 허용하지 않습니다.

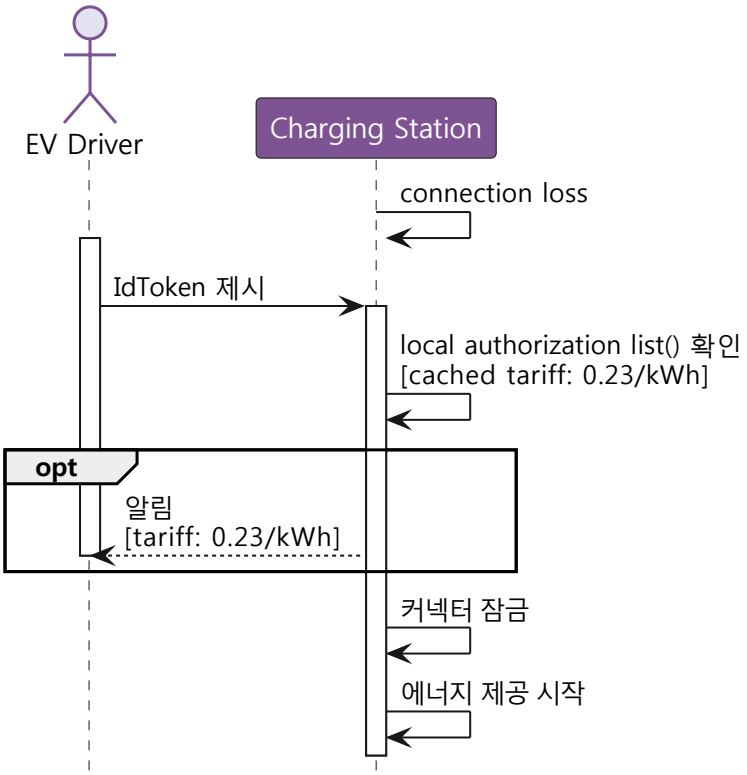

Figure 33. 시퀀스 다이어그램: 로컬 인증 목록을 통한 오프라인 인증

7	**오류 처리**	n/a
8	**비고**	n/a

C13 - Offline Authorization through Local Authorization List - Requirements

Table 84. C13 - Requirements

ID	전제 조건	요구 사항 정의	참고
C13.FR.01		Authorization Cache와 Local Authorization List가 모두 지원되는 경우 충전기는 Local Authorization List 항목을 동일한 식별자에 대한 Authorization Cache 항목보다 우선시해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
C13.FR.02	구성 변수 `OfflineTxForUnknownIdEnabled`가 false이고 충전기가 오프라인이고 `LocalAuthListSupportsExpiryDateTime` 이 존재하지 않거나 false인 경우	Local Authorization List에 있는 식별자 중 상태가 *Accepted* 인 식별자만 트랜잭션을 승인할 수 있습니다.	이는 충전기가 *cacheExpiryDateTime* 을 확인하지 않는다는 것을 의미합니다.
C13.FR.03		충전기는 CSMS를 포함하지 않고 로컬에서 IdToken을 승인할 수 있습니다.(MAY)	Local Authorization List에 설명된 대로.
C13.FR.04	구성 변수 `OfflineTxForUnknownIdEnabled`가 true이고 충전기가 오프라인이고 `LocalAuthListSupportsExpiryDateTime` 이 존재하지 않거나 false인 경우	Authorization Cache와 로컬 인증 목록에 없는 식별자일지라도 트랜잭션이 승인되어야 하고 Local Authorization List에 있는 *Accepted* 상태의 식별자도 트랜잭션이 승인되어야 합니다.	즉, 충전기는 *cacheExpiryDateTime* 을 확인하지 않습니다. C15.FR.08 참조.
C13.FR.05	구성 변수 `OfflineTxForUnknownIdEnabled`가 false이고 충전기가 오프라인이고 `LocalAuthListSupportsExpiryDateTime` = true	Local Authorization List에 있는 식별자 중 상태가 *Accepted* 이고 *cacheExpiryDateTime* 이 지나지 않은 식별자만 트랜잭션을 승인할 수 있습니다.	*cacheExpiryDateTime* 이 없는 경우 idToken은 Local Authorization List에서 만료되지 않습니다.
C13.FR.06	구성 변수 `OfflineTxForUnknownIdEnabled`가 true이고 충전기가 오프라인이고 `LocalAuthListSupportsExpiryDateTime` = true	Authorization Cache와 Local Authorization List에 없는 식별자도 트랜잭션을 승인하도록 허용되어야 하며 Local Authorization List에 있는 식별자 중 상태가 *Accepted* 이고 `cacheExpiryDateTime` 이 지나지 않은 식별자도 트랜잭션을 승인하도록 허용되어야 합니다.	즉, 로컬 인증 목록에서 만료된 토큰은 "알 수 없는 ID"가 아니기 때문에 승인되지 않습니다.

C14 - Online Authorization through Local Authorization List

Table 85. C14 - Online Authorization through Local Authorization List

번호	유형	설명
1	**이름**	로컬 인증 목록을 통한 온라인 인증
2	**ID**	C14

번호	유형	설명
	기능 블록	C. 인증
3	**목표**	*Online* 인 동안 Local Authorization List를 사용하여 idToken을 인증합니다.
4	**설명**	이 유즈케이스는 충전기가 온라인일 때 Local Authorization List를 통해 IdToken을 인증하는 방법을 설명합니다. 온라인일 때 충전기는 IdToken을 로컬로 인증할 수 있으며 알려진 IdToken에 대한 AuthorizeRequest를 보낼 필요가 없습니다.
	Actors	EV 운전자, 충전기
	시나리오 설명	**1.** EV 운전자가 IdToken을 제시합니다. **2.** 충전기는 IdToken이 알려져 있고 Local Authorization List에서 *Accepted* 상태인지 확인합니다. **3.** IdToken이 알려지지 않았거나 IdToken이 *Accepted* 가 아니면 충전기는 AuthorizeRequest를 보냅니다. **4.** 충전기가 충전을 시작합니다.
5	**필수 조건**	*Local Authorization List* 을 사용할 수 있음 *Local Authorization List* 이 `LocalAuthListEnabled`를 통해 활성화됨 EV 드라이버의 ID가 *Local Authorization List* 에 있음 ID가 유효하고 `LocalPreAuthorize`가 *true* 로 설정됨
6	**사후 조건**	**성공 사후 조건:** 충전기가 Local Authorization List에서 토큰을 승인합니다. **실패 사후 조건:** 충전기가 Local Authorization List에서 토큰을 승인하지 않습니다.

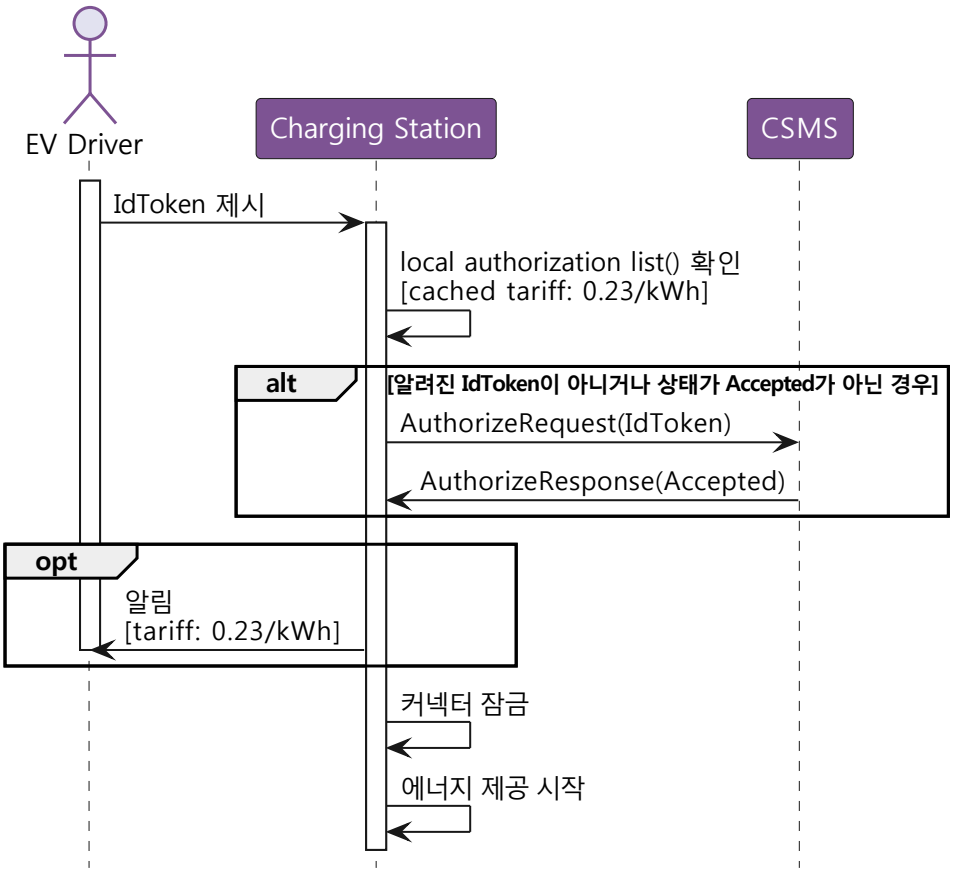

Figure 34. 시퀀스 다이어그램: 로컬 인증 목록을 통한 온라인 인증

7	**오류 처리**	n/a
8	**비고**	n/a

C14 - Online Authorization through Local Authorization List - Requirements

Table 86. C14 - Requirements

ID	전제 조건	요구 사항 정의	참고
C14.FR.01		여기서 Authorization Cache와 Local Authorization List가 모두 지원되는 경우 충전기는 Local Authorization List 항목을 동일한 식별자에 대한 Authorization Cache 항목보다 우선시해야 합니다.	
C14.FR.02	제시된 식별자가 Local Authorization List에 `Accepted` 상태로 있고 `LocalAuthListSupportsExpiryDateTime` 이 존재하지 않거나 false일 경우.	충전기는 AuthorizeRequest를 보내지 않고 충전을 시작해야 합니다.	즉, 충전기는 *cacheExpiryDateTime* 을 확인하지 않습니다.
C14.FR.03	제시된 식별자가 Local Authorization List에 있고 상태가 *Accepted* 가 아닌 경우	충전기는 AuthorizeRequest를 보내 이 IdToken을 승인하려고 시도해야 합니다.	
C14.FR.04	제시된 식별자가 Local Authorization List에 있으며 상태가 `Accepted` 이고 `LocalAuthListSupportsExpiryDateTime` = true이고 *cacheExpiryDateTime* 이 지나지 않았을 경우.	충전기는 AuthorizeRequest를 보내지 않고 충전을 시작해야 합니다.	*cacheExpiryDateTime* 이 없는 경우 idToken은 Local Authorization List에서 만료되지 않습니다.
C14.FR.05	제시된 식별자는 Local Authorization List에 있으며 상태가 `Accepted` 이고 `LocalAuthListSupportsExpiryDateTime` = true이고 *cacheExpiryDateTime* 이 지났을 경우.	충전기는 AuthorizeRequest를 보내 이 IdToken을 승인하려고 시도해야 합니다.	*cacheExpiryDateTime* 이 경과하면 Local Authorization List에 없는 것처럼 IdToken이 무시됩니다.

2.6. Offline Authorization

C15 - Offline Authorization of unknown Id

Table 87. C15 - Offline Authorization of unknown Id

번호	유형	설명
1	**이름**	알 수 없는 ID의 오프라인 인증
2	**ID**	C15
	기능 블록	C. 인증
	부모 유즈케이스	C12 - Start Transaction - Cached Id
3	**목표**	인증 캐시 항목에서 명시적으로 인증할 수 없는 "알 수 없는" 식별자의 자동 인증를 허용합니다.
4	**설명**	이 유즈케이스는 `OfflineTxForUnknownIdEnabled`를 사용하여 인증 캐시 또는 로컬 캐시 항목에 있는 것 외에 "알 수 없는" 식별자를 제시하는 시나리오를 설명합니다.
	Actor	충전기, EV 운전자
	시나리오 설명	**1.** EV 운전자가 EV 충전을 시작하고 IdToken을 제시합니다. **2.** 충전기가 Authorization Cache를 확인하는데, IdToken이 Authorization Cache에 없습니다. **3.** 충전기가 Local Authorization List를 확인하는데, IdToken이 Local Authorization List에 없습니다. **4.** 충전기는 `OfflineTxForUnknownIdEnabled`가 *True* 로 설정된 경우 알 수 없는 IdToken을 수락합니다. **5.** 충전기는 `OfflineTxForUnknownIdEnabled`가 *False* 로 설정된 경우 알 수 없는 IdToken을 거부합니다.
	대체 시나리오	C01 - EV Driver Authorization using RFID C02 - Authorization using a start button C03 - Authorization using credit/debit card C04 - Authorization using PIN-code C05 - Authorization for CSMS initiated transactions C06 - Authorization using local id type C07 - Authorization using Contract Certificates C08 - Authorization at EVSE using ISO 15118 External Identification Means (EIM)
5	**필수 조건**	충전기가 *오프라인* 입니다. 알 수 없는 IdToken이 제시됨(Authorization Cache 및/또는 Local Authorization List에 없음).
6	**사후 조건**	**성공한 사후 조건:** TransactionEventResponse의 인증 상태가 *Accepted* 입니다. **실패한 사후 조건:** `OfflineTxForUnknownIdEnabled`가 *True* 일 때 TransactionEventResponse의 인증 상태가 *Not Accepted* 입니다.

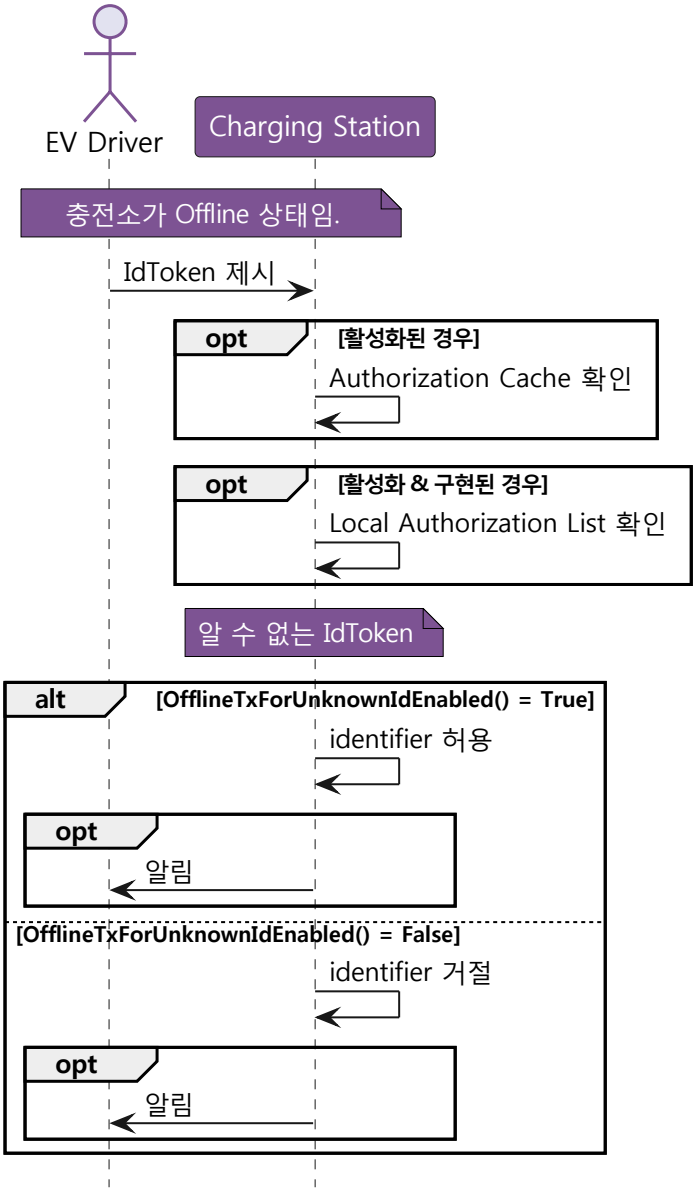

Figure 35. 시퀀스 다이어그램: 시작 트랜잭션 - 알 수 없는 오프라인 인증

7	**오류 처리**	해당 없음
8	**참고**	이는 ISO 15118 계약 인증서의 일부로 제시된 eMAID를 포함한 모든 유형의 식별자에 적용됩니다.

C15 - Offline Authorization of unknown Id - Requirements

Table 88. C15 - Requirements

ID	전제 조건	요구 사항 정의	참고
C15.FR.01	식별자가 `OfflineTxForUnknownIdEnabled`를 통해 승인된 경우	충전기는 토큰을 Authorization Cache에 추가해서는 안 됩니다	
C15.FR.02	CSMS와의 연결이 복구된 경우	충전기는 *Offline* 으로 승인된 모든 트랜잭션에 대해 TransactionEventRequest를 보내야 합니다.	transaction-related message handling에서 설명한 대로

ID	전제 조건	요구 사항 정의	참고
C15.FR.03	C15.FR.02 AND TransactionEventResponse의 승인 상태가 *Accepted* 가 아니고 AND 트랜잭션이 아직 진행 중이며. AND StopTxOnInvalidId 가 *true* 이고 AND TxStopPoint에는 다음이 포함되지 않는 경우 (Authorized 또는 PowerPathClosed 또는 EnergyTransfer)	충전기는 에너지 전송을 중지하고 TransactionEventRequest (eventType = Updated) 메시지의 *triggerReason* 을 Deauthorized 로 설정하고 *chargingState* 를 SuspendedEVSE 또는 바람직하게는 EVConnected 로 설정해서 보내야 합니다.	*chargingState* 를 SuspendedEVSE 또는 EVConnected 로 설정하는 효과는 모두 에너지를 전달하지 않는다는 동일한 효과가 있으므로, 원래 요구 사항을 준수하는 기존 구현을 망치지 않기 위해 이 상황에서도 SuspendedEVSE 를 사용할 수 있습니다. SuspendedEVSE 를 사용하는 것은 다음 OCPP 릴리스에서 더 이상 사용되지 않습니다.
C15.FR.04	C15.FR.02 AND TransactionEventResponse의 승인 상태가 *Accepted* 가 아니고 AND 트랜잭션이 아직 진행 중이며 AND StopTxOnInvalidId 가 *true* 이고 AND TxStopPoint에 다음이 포함될경우. (Authorized 또는 PowerPathClosed 또는 EnergyTransfer)	충전기는 트랜잭션을 중지하고 TransactionEventRequest (eventType = Ended)의 *triggerReason* 이 *Deauthorized* 로 설정되고 stoppedReason이 *DeAuthorized* 로 설정하여 전송해야 합니다.	
C15.FR.05	C15.FR.04 및 충전기에서 충전 케이블을 잠글 수 있는 경우	충전기는 소유자가 식별자를 제시할 때까지 충전 케이블을 잠근 상태로 유지해야 합니다.	
C15.FR.06	C15.FR.02 AND TransactionEventResponse의 승인 상태가 *Accepted* 가 아니고 AND 트랜잭션이 아직 진행 중이며 AND StopTxOnInvalidId 가 *false* 로 설정되었고 AND MaxEnergyOnInvalidId가 구현되지 않았거나 초과되었을 경우. TxStopPoint에는 다음이 포함되지 않습니다. EnergyTransfer	충전기는 EV로의 에너지 공급을 즉시 중단하고 TransactionEventRequest(eventType = Updated)를 보내야 합니다. 이때 *triggerReason* 은 *ChargingStateChanged* 로, chargingState는 *SuspendedEVSE* 로 설정해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
C15.FR.07	C15.FR.02 및 TransactionEventResponse의 승인 상태가 *Accepted* 가 아니고 AND 트랜잭션이 아직 진행 중이고 AND `StopTxOnInvalidId` 가 *false* 로 설정되고 AND `MaxEnergyOnInvalidId`가 설정되었으며 초과되지 않았을 경우.	EV에 대한 에너지 전달은 `MaxEnergyOnInvalidId`에 지정된 에너지 양에 도달할 때까지 허용되어야 합니다.	
C15.FR.08	알 수 없는 식별자가 제시되고 `OfflineTxForUnknownIdEnabled`가 *true* 로 설정된 경우	충전기는 제시된 IdToken을 수락해야 합니다.	

2.7. Master Pass

C16 - Stop Transaction with a Master Pass

Table 89. C16 - Stop Transaction with a Master Pass

번호	유형	설명
1	**이름**	마스터 패스로 트랜잭션 중지
2	**ID**	C16
	기능 블록	C. 인증
3	**목표**	마스터 패스를 사용하여 트랜잭션을 중단할 수 있도록 합니다(예: 법 집행 기관).
4	**설명**	이 유즈케이스는 마스터 패스(사용자)가 있는 사람이 진행 중인 트랜잭션을 중단(선택)하여 케이블이 잠금 해제되는 방법을 다룹니다. 이 마스터 패스는 `MasterPassGroupId`에서 구성할 수 있습니다.
	Actor	충전기, CSMS, 사용자

번호	유형	설명
	시나리오 설명	**1.** 사용자(법 집행 기관)가 충전기에 자신의 IdToken을 제시합니다. **2.** 충전기가 AuthorizeRequest를 CSMS로 보내 인증을 요청합니다. **3.** AuthorizeRequest를 수신하면 CSMS는 AuthorizeResponse로 응답합니다. 이 응답 메시지에는 구성 변수 `MasterPassGroupId`의 값과 동일한 GroupId가 포함되어 있으며 idToken이 유효합니다. **4a.** 충전기에 UI가 있는 경우 충전기는 마스터 패스 UI를 "표시"합니다. **5a.** 사용자는 중지할 트랜잭션을 선택합니다. **6a.** 충전기는 선택한 트랜잭션을 중지하고 중지된 모든 트랜잭션에 대해 TransactionEventRequest (eventType = Ended, stopReason = MasterPass)를 CSMS로 보냅니다. **7a.** TransactionEventRequest를 수신하면 CSMS는 TransactionEventResponse로 응답합니다. **4b.** 충전기에 UI가 없는 경우 충전기는 모든 트랜잭션을 중지하고 중지된 모든 트랜잭션에 대해 TransactionEventRequest(eventType = Ended, stopReason = MasterPass)를 CSMS로 보냅니다. **5b.** TransactionEventRequest를 수신하면 CSMS는 TransactionEventResponse로 응답합니다.
	대체 시나리오	C01 - EV Driver Authorization
5	**필수 조건**	진행 중인 트랜잭션 구성 변수: `MasterPassGroupId` 설정됨. 사용자 IdToken에 구성된 `MasterPassGroupId`와 동일한 그룹 ID가 있습니다.
6	**사후 조건**	(선택된) 트랜잭션이 중단됨.

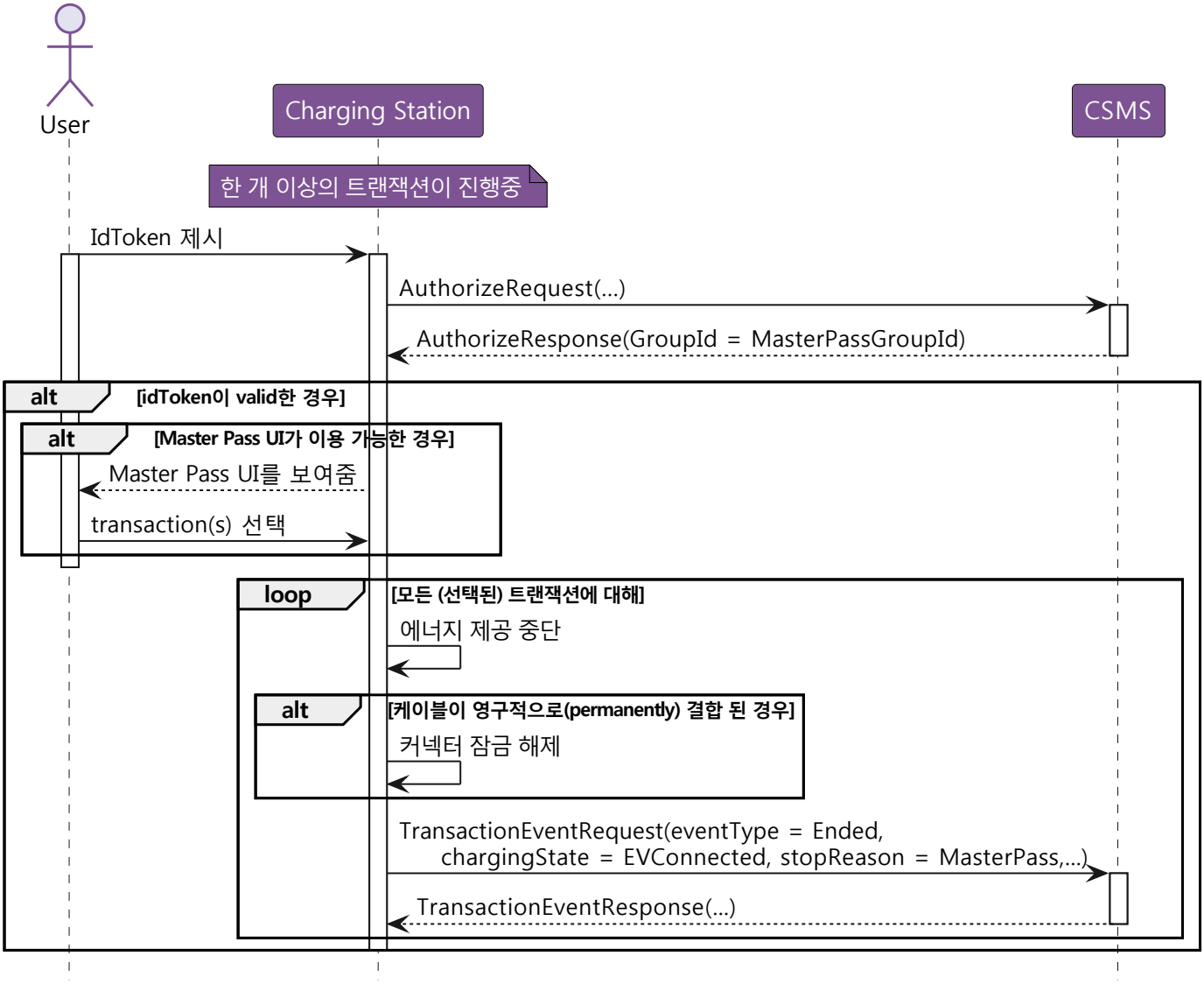

Figure 36. 시퀀스 다이어그램: 마스터 패스로 트랜잭션 중지

7	**오류 처리**	사용자가 허용 가능한 시간 초과 전에 선택을 하지 않으면 충전기는 정상 작동으로 돌아가야 합니다.
8	**비고**	위의 시나리오 설명과 시퀀스 다이어그램은 트랜잭션 중지를 위한 구성 변수가 다음과 같이 구성되는 것을 기반으로 합니다. `TxStopPoint`: Authorized, DataSigned, PowerPathClosed, EnergyTransfer 이 유즈케이스는 다른 구성에도 유효하지만, 그러면 트랜잭션이 다른 순간에 중지될 수 있으며, 이때 메시지가 전송되는 순서가 변경될 수 있습니다. 자세한 내용은 유즈케이스를 참조하세요: E06 - Stop Transaction options

C16 - Stop Transaction with a Master Pass - Requirements

Table 90. C16 - 마스터 패스로 트랜잭션 중지 - Requirements

ID	전제 조건	요구 사항 정의
C16.FR.01	사용자가 `MasterPassGroupId`와 같은 그룹 ID를 가진 IdToken을 제시하고 충전기에 UI가 있을 경우.	충전기는 마스터 패스 UI를 "표시"해야 합니다.
C16.FR.02	사용자가 `MasterPassGroupId`와 같은 그룹 ID를 가진 IdToken을 제시하고 충전기에 UI가 없을 경우.	충전기는 진행 중인 모든 트랜잭션을 중지해야 합니다.
C16.FR.03	`MasterPassGroupId`와 같은 groupId를 갖는 IdToken	트랜잭션을 시작할 수 없습니다.
C16.FR.04	Authorization Cache에 있는 `MasterPassGroupId`와 같은 groupId를 갖는 IdToken.	충전기는 Authorization Cache에 있는 정보를 기반으로 "마스터 패스" 토큰의 승인을 허용할 수도 있습니다.(MAY)
C16.FR.05	Local Authorization List에 있는 `MasterPassGroupId`와 같은 groupId를 갖는 IdToken.	충전기는 Local Authorization List의 정보에 따라 "마스터 패스" 토큰의 승인을 허용할 수도 있습니다.(MAY)

D. LocalAuthorizationList Management

1. Introduction

C1.4 - Local Authorization List에서 설명한 대로 로컬 인증 목록은 CSMS와 동기화할 수 있는 식별자 목록입니다. 오프라인일 때 사용자 인증을 진행하고 온라인일 때 인증 응답 시간을 줄이는 데 사용할 수 있습니다. 이 기능 블록은 로컬 인증 목록을 대체할 식별자의 전체 목록을 보내거나 로컬 인증 목록에 적용할 변경 사항(추가, 업데이트, 삭제) 목록을 보내어 CSMS가 목록을 동기화할 수 있도록 하는 것입니다. 이를 지원하는 작업은 GetLocalListVersion 및 SendLocalList입니다.
목록에는 모든(또는 일부) 식별자의 인증 부여 상태와 해당 만료 날짜가 포함됩니다. 이러한 값은 로컬 인증 중에 사용자에게 보다 세부적인 정보를 제공하는 데 사용될 수 있습니다(예: 표시 메시지).

2. Use cases & Requirements

D01 - Send Local Authorization List

Table 91. D01 - 로컬 인증 목록 전송

번호	유형	설명
1	**이름**	로컬 인증 목록 전송
2	**ID**	D01
	기능 블록	D. 로컬 인증 목록
3	**목표**	CSMS가 충전기가 idToken의 인증 부여에 사용할 수 있는 Local Authorization List를 전송할 수 있도록 합니다.
4	**설명**	CSMS가 충전기가 idToken의 인증 부여에 사용할 수 있는 Local Authorization List를 전송합니다. 목록은 현재 충전기의 목록을 대체하는 전체 목록이거나 현재 충전기의 목록에 적용할 업데이트가 있는 차등 목록일 수 있습니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS는 SendLocalListRequest를 보내 Local Authorization List를 설치하거나 업데이트합니다. **2.** SendLocalListRequest를 수신하면 충전기는 SendLocalListResponse와 함께 상태를 응답합니다.
5	**필수 조건**	로컬 인증 목록은 구성 변수 `LocalAuthListEnabled`로 활성화됩니다.
6	**사후 조건**	**성공한 사후 조건:** - 충전기에 새로운 Local Authorization List가 설치되었습니다. **실패 사후 조건:** - 충전기의 Local Authorization List가 그대로 유지됩니다. - 상태가 *Failed* 또는 *VersionMismatch* 인 경우.

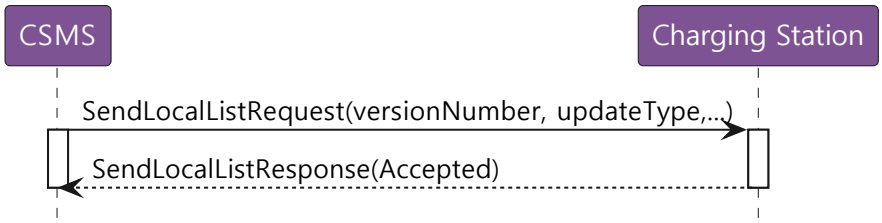

Figure 37. 시퀀스 다이어그램: 로컬 인증 목록 전송

7	**오류 처리**	상태가 *Failed* 또는 *VersionMismatch* 이고 updateType이 Differential인 경우 CSMS는 전체 Local Authorization List를 전송합니다. 이 목록이 한 메시지에 너무 큰 경우 updateType *Full* 로 초기 목록을 보내고 updateType *Differential* 을 사용하여 식별자를 추가하여 목록이 완전히 전송될 때까지 시작합니다(단일 SendLocalListRequest에서 전송할 수 있는 식별자의 양은 요구 사항 D01.FR.11에 설명된 대로 제한됨).
8	**비고**	n/a

D01 - Send Local Authorization List - Requirements

Table 92. D01 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
D01.FR.01		SendLocalListRequest에는 업데이트 유형(*updateType*)과 충전기가 업데이트된 후 Local Authorization List와 연결해야 하는 버전 번호(*versionNumber*)가 포함되어야 합니다.	
D01.FR.02		SendLocalListResponse는 충전기가 Local Authorization List의 업데이트를 수락했는지 여부를 나타냅니다.	
D01.FR.03	SendLocalListResponse의 *status* 가 `Failed` 또는 `VersionMismatch` 이고 *updateType* 이 `Differential` 인 경우	CSMS가 전체 Local Authorization List를 보내는 것이 좋습니다.	이 목록이 하나의 메시지에 넣기에 너무 큰 경우(D01.FR.11 참조), *updateType* `Full` 을 사용하여 초기 목록을 전송하고 목록이 완전히 전송될 때까지 *updateType* `Differential` 을 사용하여 식별자를 추가하는 것으로 시작합니다.
D01.FR.04	*localAuthorizationList* 가 제공되지 않고 *updateType* 이 `Full` 인 경우.	충전기는 목록에서 모든 IdToken을 제거해야 합니다.	목록의 버전 번호는 요청에서 *versionNumber* 값으로 계속 업데이트됩니다.
D01.FR.05		*localAuthorizationList* 가 비어 있거나 없이 차등 업데이트를 요청하는 것은 목록에 영향을 미치지 않습니다.	목록의 버전 번호는 요청에서 *versionNumber* 값으로 계속 업데이트됩니다.
D01.FR.06		Local Authorization List의 모든 IdToken은 고유해야 합니다.	중복된 값은 허용되지 않습니다.
D01.FR.09		충전기는 CSMS에서 SendLocalList 메시지를 수신한 경우를 제외하고는 다른 방법으로 인증 목록의 내용을 수정해서는 안 됩니다.	
D01.FR.10		Local Authorization List는 충전기에서 비휘발성 메모리에 유지 관리해야 하며 재부팅 및 정전 시에도 지속되어야 합니다.	
D01.FR.11		단일 SendLocalListRequest의 크기는 구성 변수 `ItemsPerMessageSendLocalList` 및 `BytesPerMessageSendLocalList`에 의해 제한됩니다.	
D01.FR.12		Local Authorization List를 지원하는 충전기는 구성 변수 `LocalAuthListEntries`를 구현해야 합니다.	이를 통해 CSMS는 충전기에서 Local Authorization List 요소의 현재 양과 최대 가능 개수를 알 수 있습니다.
D01.FR.13		충전기는 Local Authorization List가 활성화되었는지 여부를 나타냅니다. 이는 `LocalAuthListEnabled` 구성 변수에 의해 보고되고 제어됩니다.	

ID	전제 조건	요구 사항 정의	참고
D01.FR.15	충전기가 *updateType* 이 `Full` 이고 *localAuthorizationList* 가 비어 있지 않은 SendLocalListRequest를 수신하는 경우	충전기는 현재 Local Authorization List를 SendLocalListRequest의 목록으로 대체하고 버전 번호를 메시지에 지정된 값으로 설정해야 합니다.	그렇지 않으면 초기 충전기 및 CSMS 목록을 동기화할 방법이 없습니다. 이 목록이 하나의 메시지에 넣기에 너무 큰 경우(D01.FR.11 참조), *updateType* `Full` 로 초기 목록을 보내고 *updateType* `Differential` 을 사용하여 식별자를 추가하여 목록이 완전히 전송될 때까지 시작합니다.
D01.FR.16	충전기가 *updateType* 이 `Differential` 이고 *localAuthorizationList* 에 idTokenInfo가 있는 AuthorizationData 요소가 포함된 SendLocalListRequest를 수신하는 경우	충전기는 이러한 요소로 Local Authorization List를 업데이트하고 버전 번호를 메시지에 지정된 값으로 설정해야 합니다.	아직 없는 경우 추가하고, Local Authorization List에 이미 있는 경우 새 정보로 업데이트합니다.
D01.FR.17	충전기가 *updateType* 이 `Differential` 인 SendLocalListRequest를 수신하고 *localAuthorizationList* 에 idTokenInfo가 없는 AuthorizationData 요소가 포함된 경우	충전기는 Local Authorization List에서 이러한 요소를 제거하고 버전 번호를 메시지에 지정된 값으로 설정해야 합니다.	
D01.FR.18		SendLocalListRequest의 *versionNumber* 는 0보다 커야 합니다.	GetLocalListVersionResponse 에서 *versionNumber* = 0은 특별한 의미를 갖습니다. 이는 로컬 목록이 설치되지 않음을 의미합니다. 따라서 값 0은 절대 사용해서는 안 됩니다.
D01.FR.19	충전기가 *updateType* = `Differential` 인 SendLocalListRequest를 수신하고 *versionNumber* 가 Local Authorization List의 버전 번호보다 작거나 같은 경우	충전기는 Local Authorization List를 업데이트하는 것을 거부하고 상태가 `VersionMismatch` 로 설정된 SendLocalListResponse를 반환해야 합니다.	

D02 - Get Local List Version

Table 93. D02 - 로컬 목록 버전 가져오기

번호	유형	설명
1	**이름**	로컬 목록 버전 가져오기
2	**ID**	D02
	기능 블록	D. 로컬 인증 목록
	부모 사용 사례	D01 - Send Local Authorization List
3	**목표**	Local Authorization List의 동기화를 지원합니다.
4	**설명**	CSMS는 GetLocalListVersionRequest를 보내서 Local Authorization List의 버전 번호에 대한 충전기를 요청할 수 있습니다.

번호	유형	설명
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS는 GetLocalListVersionRequest를 보내 이 값을 요청합니다. **2.** GetLocalListVersionRequest를 수신하면 충전기는 GetLocalListVersionResponse로 응답하며 여기에는 Local Authorization List의 버전 번호가 포함됩니다.
5	**필수 조건**	
6	**사후 조건**	CSMS가 GetLocalListVersionResponse와 Local Authorization List 버전을 수신했습니다.

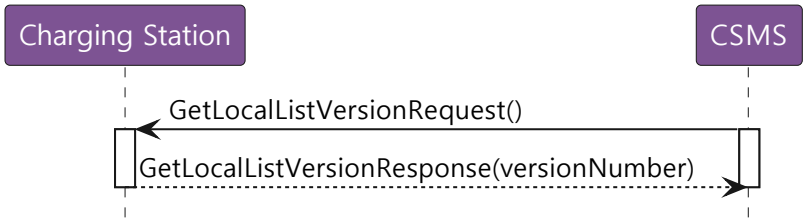

Figure 38. 시퀀스 다이어그램: 로컬 목록 버전 가져오기

7	**오류 처리**	n/a
8	**참고**	0(영)의 *versionNumber* 는 로컬 인증 목록이 존재하지 않음을 나타내기 위해 예약되어 있습니다. 로컬 인증 목록이 활성화되지 않았거나 아직 CSMS에서 업데이트를 수신하지 않아 반환할 버전 번호가 없기 때문입니다. 반대로, CSMS가 빈 *localAuthorizationList* 와 함께 SendLocalListRequest를 보냈기 때문에 비워진 로컬 인증 목록은 *versionNumber* > 0을 갖습니다.

D02 - Get Local List Version - Requirements

Table 94. D02 - 요구 사항

ID	전제 조건	요구 사항 정의
D02.FR.01	`LocalAuthListEnabled` 가 *true* 입니다.	충전기가 GetLocalListVersionRequest를 수신하면 충전기는 Local Authorization List의 버전 번호가 포함된 GetLocalListVersionResponse로 응답해야 합니다.
D02.FR.02	`LocalAuthListEnabled` 가 *true* 이고 CSMS가 아직 충전기에 Local Authorization List에 대한 업데이트를 보내지 않았습니다(SendLocalListRequest를 통해)	충전기가 GetLocalListVersionRequest를 수신하면 충전기는 GetLocalListVersionResponse로 응답해야 하며 *versionNumber* 는 0(영)으로 Local Authorization List가 없음을 나타냅니다.
D02.FR.03	`LocalAuthListEnabled` 가 *true* 가 아닙니다.	충전기가 GetLocalListVersionRequest를 수신하면 충전기는 GetLocalListVersionResponse로 응답해야 하며 *versionNumber* 는 0(영)이어야 합니다. 이는 Local Authorization List가 없음을 나타냅니다.

E. Transactions

1. Introduction

이 기능 블록은 OCPP 트랜잭션 관련 기능을 설명합니다. 트랜잭션은 충전기에서 시작/중지됩니다. 언제든지 EVSE에서는 최대 하나의 트랜잭션만 활성화될 수 있습니다.

1.1. Flexible transaction start/stop

가능한 한 많은 비즈니스 사례를 지원하고 특정 비즈니스 사례에 필요하지 않은 경우 너무 많은 메시지를 보내는 것을 방지하기 위해 OCPP 2.0.1은 트랜잭션 시작 및 중지의 유연한 구성을 지원합니다.

이를 위해 다음 구성 변수가 정의됩니다.

- `TxStartPoint`
- `TxStopPoint`

이 두 구성 변수를 사용하면 트랜잭션을 시작할 시점을 정의할 수 있습니다. TransactionEventRequest (eventType = Started) 및 트랜잭션을 중지할 시점을 정의할 수 있습니다. TransactionEventRequest (eventType = Ended)

OCPP 2.0.1에서 트랜잭션의 유연한 시작/중지 지점이 도입됨에 따라 트랜잭션 정의를 제공하는 것이 중요합니다.

> 트랜잭션은 CSMS에서 기록하는 청구 세션의 일부입니다. 시작 및 종료 시간이 있는 단일 시간 프레임입니다. 운영자는 정보를 청구에 사용할 수 있습니다.

충전기 운영자가 TxStartPoint 및 TxStopPoint의 값을 정의해야 하지만(충전기에서 읽기 전용 값으로 미리 설정된 경우 제외), 모든 조합이 합리적인 것은 아닙니다.

다음 세 가지 변형이 가장 일반적입니다.

- 연결 시간이 청구되는 경우 시작 및 정지 지점은 `EVConnected` 여야 합니다.
- 사용 시간이 청구되는 경우 시작 지점은 `EVConnected`, Authorized, 정지 지점은 `EVConnected` 여야 합니다.
  (인증이 먼저 이루어지는 경우 충전기가 이미 '사용 중'으로 간주됨).
- 충전 시간이 청구되는 경우 시작 및 정지 지점은 `PowerPathClosed` 여야 합니다. (이는 충전기가 전원을 공급할 준비가 되면 시작되고 인증이 취소되거나 차량이 분리되면 중지됩니다.) 그 사이의 일시 중지(예: SuspendedEV(SE))는 트랜잭션을 종료하지 않습니다. 트랜잭션 중에 수집된 미터 값을 기준으로 에너지 또는 전력량에 대한 청구를 수행할 수 있습니다.

WARNING	시작 및 종료 지점의 특정 조합은 시작된 트랜잭션이 결코 중지되지 않는 상황으로 이어질 수 있습니다. 예를 들어, 시작 지점이 `ParkingBayOccupancy` 이고 종료 지점이 `EVConnected` 인 경우 EV가 주차 공간을 점유하면 트랜잭션이 시작되지만 사용자가 EV를 연결하지 않고 단순히 운전해 가는 경우 `ParkingBayOccupancy` 가 종료 지점으로 구성되지 않았기 때문에 트랜잭션이 계속 열려 있습니다.

1.1.1. Readonly or Read/Write

OCPP 2.0.1은 트랜잭션 시작/중지 구성 변수에 대해 두 가지 옵션을 지원합니다. RW(읽기-쓰기) 또는 R(읽기 전용) 중 하나가 될 수 있습니다.

충전기가 RW를 지원하는 경우 CSO가 설정을 구성할 수 있습니다. 가능한 모든 설정을 지원하려면 충전기의 소프트웨어가 더 유연해야 합니다.

R만 있는 경우, 설정은 펌웨어에 고정되어 있고, CSO는 충전기가 어떻게 동작할지 알아내기 위해 설정을 읽을 수 있지만 구성할 수는 없습니다. 이렇게 하면 구현이 더 단순해집니다. 대상 시장의 요구 사항이 잘 알려져 있는 경우 유연한 모델을 구현할 필요가 없을 수 있습니다.

1.1.2. OCPP 1.6 Transaction compatibility

OCPP 1.6과 유사한 트랜잭션이 필요한 경우, 이 섹션에서는 트랜잭션 시작 및 종료 지점을 구성하는 방법을 설명합니다.

OCPP 1.x에서는 충전기가 StartTransaction.req를 보내야 하는 순간이 매우 정확하게 정의되지 않았습니다. 일반적으로 충전기가 전력을 공급할 준비가 되었을 때, 즉 케이블이 연결되고 사용자가 인증되었을 때 이 작업이 수행되었습니다.

유사한 트랜잭션 시작 동작을 지원하려면 *PowerPathClosed* 값을 사용해야 합니다. (그리고 완전성을 위해 다음을 추가하세요: *EnergyTransfer*)

Table 95. OCPP 1.6 호환 트랜잭션 설정

구성 변수	값
`TxStartPoint`	PowerPathClosed
`TxStopPoint`	EVConnected, Authorized

정지 동작의 경우 *ParkingBayOccupancy* 를 추가하면 안 됩니다. OCPP 1.6은 이를 지원하지 않았고, 누군가가 '반대' 커넥터를 사용하는 듀얼 소켓 충전기의 경우 EV는 계속 충전중임에도 트랜잭션이 중지될 수 있습니다.

1.2. TransactionId generation

OCPP 2.0.1의 새로운 기능: 트랜잭션 ID는 이제 충전기에서 생성됩니다.

OCPP 1.x에서는 CSMS가 이를 수행했습니다. 여기에는 몇 가지 단점이 있었습니다. 충전기가 오프라인일 때 트랜잭션 ID가 없는 트랜잭션이 있었습니다.

충전기에서 생성된 트랜잭션 ID는 해당 충전기에서 고유해야 합니다. 충전기의 수명 동안 동일한 트랜잭션 ID를 두 번 사용해서는 안 됩니다. 또한 충전기가 재부팅되거나, 전원을 껐다가 다시 켜거나, 펌웨어를 업데이트하거나, 수리할 때도 마찬가지입니다.

OCPP는 사용할 알고리즘을 지정하지 않지만 UUID를 사용하는 것이 좋습니다.

1.3. Delivering transaction-related messages

TransactionEventRequest 메시지의 주요 목적은 CSMS에 나중에 트랜잭션 과금에 사용할 정보를 제공하는 것입니다. CSMS가 트랜잭션 과금에 필요한 모든 정보를 수신하도록 하기 위해 OCPP는 *retrying* 와 *sequence number* 라는 두 가지 메커니즘을 사용합니다.

1.3.1. Retrying

충전기는 보고해야하는 이벤트가 발생한 후 가능한 한 빨리 CSMS 시스템에 TransactionEventRequest 메시지를 보냅니다.

충전기가 오프라인이거나 전송 중에 메시지를 처리하는 동안 오류가 발생하면 CSMS에 청구 정보가 누락됩니다. CSMS에서 누락된 정보를 복구하기 위해 충전기는 이 정보를 전달하기 위해 재시도해야 합니다.
충전기가 message timeout period 내에 TransactionEventRequest 메시지에 대한 TransactionEventResponse를 수신하지 못하면 충전기는 사용 사례 E13 - Transaction-related message not accepted by CSMS에 설명된 재시도 절차를 따라야 합니다.

1.3.2. Sequence numbers

TransactionEventRequest 메시지 전달이 실패하고 나중에 재시도될 경우, 결과적으로 TransactionEventRequest 메시지가 충전기에서 트랜잭션 이벤트가 발생한 순서와 다른 순서로 CSMS에 도착할 수 있습니다. 이로 인해 CSMS가 트랜잭션에 대한 모든 TransactionEventRequest 메시지를 수신했는지 알기 어려워질 수 있으며, CSMS는 트랜잭션에 대한 청구를 시작하기 전에 이를 알고 싶어할 수 있습니다.

트랜잭션에 대한 모든 TransactionEventRequest 메시지가 수신되었는지 알 수 있도록 하기 위해 OCPP는 TransactionEventRequest 메시지에서 *sequence number* 를 사용합니다. 모든 EVSE에 대해 충전기는 해당 EVSE에 대해 생성된 TransactionEventRequest 메시지 수의

카운터를 유지 관리합니다. 새로운 TransactionEventRequest 메시지를 생성할 때 충전기는 요청의 **seqNo** 필드에 EVSE 카운터의 현재 값을 포함한 다음 카운터를 증가시킵니다. 이 메커니즘을 사용하면 CSMS는 다음을 확인하여 트랜잭션에 대한 전체 정보가 있는지 확인할 수 있습니다.

- **seqNo** *a* 와 함께 트랜잭션 시작에 대한 TransactionEventRequest를 수신했습니다.
- *a* 보다 큰 **seqNo** *o* 인 트랜잭션 중지에 대한 TransactionEventRequest를 수신했습니다.
- *a* 와 *o* 사이의 모든 정수 *n* 에 대해 **seqNo** *n* 이 포함된 트랜잭션에 대한 TransactionEventRequest를 수신했습니다.

1.3.2.1. Sequence number generation

이 섹션은 규범적입니다.

트랜잭션이 시작되면 충전기는 TransactionEventRequest 메시지의 *seqNo* 필드를 0으로 설정해야 합니다.
(*seqNo* 가 지속적으로 증가하는 구현은 여전히 허용됩니다.)

각 TransactionEventRequest 후에 충전기는 *seqNo* 를 1씩 늘려야 합니다.

1.4. Authorization

이 기능 블록의 유즈케이스를 단순화하기 위해 EV 운전자가 인증되는 방식은 이러한 유즈케이스에 포함되지 않습니다. 간단히 "사용자 인증 성공" 또는 "EV 운전자가 충전기 및/또는 CSMS에서 인증 되었습니다."와 같이 호출됩니다. 이는 EV 운전자를 인증하는 어떠한 방식일 수 있습니다. 인증에 대한 모든 옵션과 요구 사항은 기능 블록 C Authorization을 참조하세요.

1.5. Clarification for optional fields in TransactionEventRequest

이 섹션은 정보 제공을 목적으로 합니다.

TransactionEventRequest에는 여러 선택 필드가 포함되어 있습니다. 이러한 필드 중 일부는 한 번만 보내야 하며 모든 TransactionEventRequest에서 반복해서는 안 됩니다. 다음 요약은 이러한 선택 필드와 관련된 요구 사항을 나타냅니다.

evse

(E01.FR.16) *evse* 필드는 EV가 연결된 후 발생하는 첫 번째 TransactionEventRequest에서만 제공됩니다. 이후의 모든 TransactionEventRequest에서 반복되지 않습니다.

idToken

(E03.FR.01) *idToken* 필드는 트랜잭션이 인증된 후 발생하는 첫 번째 TransactionEventRequest에서 한 번 제공됩니다.
(E07.FR.02) *idToken* 필드는 트랜잭션 인증이 완료된 후 발생하는 TransactionEventRequest에서 한 번 제공됩니다.
(C12.FR.02) 위의 내용은 idToken이 `Accepted` 상태로 인증 캐시에 존재해서 인증된 경우에도 해당합니다.
(F02.FR.05): 위의 내용은 idToken이 RequestStartTransactionRequest에서 제공된 경우에도 해당합니다.

reservationId

(E03.FR.03/H01.FR.15) *reservationId* 필드는 충전기에 예약이 존재했던 idToken에 의해 트랜잭션이 인증되었을 때 발생하는 첫 번째 TransactionEventRequest에서만 제공됩니다.
(F02.FR.06) 위의 내용은 idToken이 RequestStartTransactionRequest에서 제공된 경우에도 해당합니다.

meterValue

(E02.FR.09) TransactionEventRequest(*eventType*=`Started`)는 SampledDataCtrlr.TxStartedMeasurands에서 구성된 미터 값을 포함해야 합니다.

(E02.FR.10) TransactionEventRequest(*eventType*=`Updated`)는 SampledDataCtrlr.TxUpdatedInterval에서 구성된 모든 간격에서 전송되어야 하며 SampledDataCtrlr.TxUpdatedMeasurands에서 구성된 미터 값을 포함해야 합니다. TxUpdatedMeasurands == 0이면 미터 값이 전송되지 않습니다.

(E06.FR.11) TransactionEventRequest(*eventType*=`Ended`)는 SampledDataCtrlr.TxEndedMeasurands에서 구성된 미터 값을 포함해야 합니다. SampledDataCtrlr.TxEndedInterval == 0인 경우, 트랜잭션 시작 및 종료 시에 가져온 값만 포함됩니다.

transactionInfo.chargingState

(E02.FR.13) 충전 상태가 변경될 때마다 충전기는 *chargingState* 를 포함하는 TransactionEventRequest를 보내야 합니다. 즉, TransactionEventRequest(*eventType*=`Started`)는 항상 *chargingState* 를 갖는데, 상태가 존재하지 않는 상태에서 값으로 바뀌기 때문입니다.

충전 상태 변경이 유일한 이벤트인 경우 TransactionEventRequest는 *triggerReason* = `ChargingStateChanged` 를 갖지만, 충전 상태 변경이 *triggerReason* = `CablePluggedIn` 또는 (`Stop`)`Authorized` 와 같은 다른 이벤트와 함께 발생하는 경우 *chargingState* 는 해당 메시지의 일부로 보고될 수 있습니다.

triggerReason = `ChargingStateChanged` 인 TransactionEventRequest는 항상 *chargingState* 를 포함해야 합니다.

transactionInfo.stoppedReason

(C15.FR.04, E05.FR.10, E05.FR.08/09, E07.FR.06) *stoppedReason* 은 값이 `Local` 인 경우를 제외하고 TransactionEventRequest(*eventType* =`Ended`)에 항상 제공되어야 하며, Local인 경우는 생략될 수 있습니다.
(F03.FR.03, F03.FR.10, F04.FR.03) 위의 내용은 RequestStopTransactionRequest에 의해 중지된 트랜잭션에도 적용되지만, 이 경우 *stoppedReason* 값은 `Remote` 여야 합니다.

transactionInfo.remoteStartId

(C05.FR.03, F01.FR.25, F02.FR.01) *remoteStartId* 는 동일한 *remoteStartId* 를 사용하여 RequestStartTransactionRequest 이후의 다음 TransactionEventRequest에서 전송되어야 합니다.

2. Use cases & Requirements

2.1. OCPP transaction mechanism

E01 - Start Transaction options

Table 96. E01 - 트랜잭션 시작

번호	유형	설명
1	**이름**	트랜잭션 시작 옵션
2	**ID**	E01
	기능 블록	E. 트랜잭션
3	**목표**	충전기에서 트랜잭션이 시작되었음을 CSMS에 알리는 것.
4	**설명**	이 사용 사례는 충전기의 구성에 따라 (TransactionEventRequest를 eventType = Started와 함께 전송하여) 충전기가 트랜잭션을 시작할 수 있는 여러 순간에 대해 설명합니다.
5	*Actors*	충전기, CSMS, EV 운전자
S1	*시나리오 목표*	**주차장 점유 감지기가 "EV"를 감지하면 트랜잭션을 시작합니다.**
	시나리오 설명	**1.** EV 운전자가 주차장 점유 감지기가 있는 충전기에 "EV"를 주차하면 감지기가 작동합니다. **2.** 충전기가 TransactionEventRequest(eventType = Started)를 보내 CSMS에 시작된 트랜잭션에 대해 알립니다(운전자가 아직 알려지지 않은 경우에도). **3.** CSMS가 TransactionEventRequest가 수신되었음을 확인하는 TransactionEventResponse로 응답합니다.
	필수 조건	EVSE에서 진행 중인 트랜잭션이 없습니다. 구성 변수: `TxStartPoint`에는 다음이 포함됩니다. ParkingBayOccupancy
	사후 조건	**성공한 사후 조건:** 트랜잭션이 진행 중이고 CSMS에 *성공적으로* 통보되었습니다. **실패한 사후 조건:** 트랜잭션이 *진행 중이 아닙니다, 또는* CSMS에 *통보되지 않았습니다.*

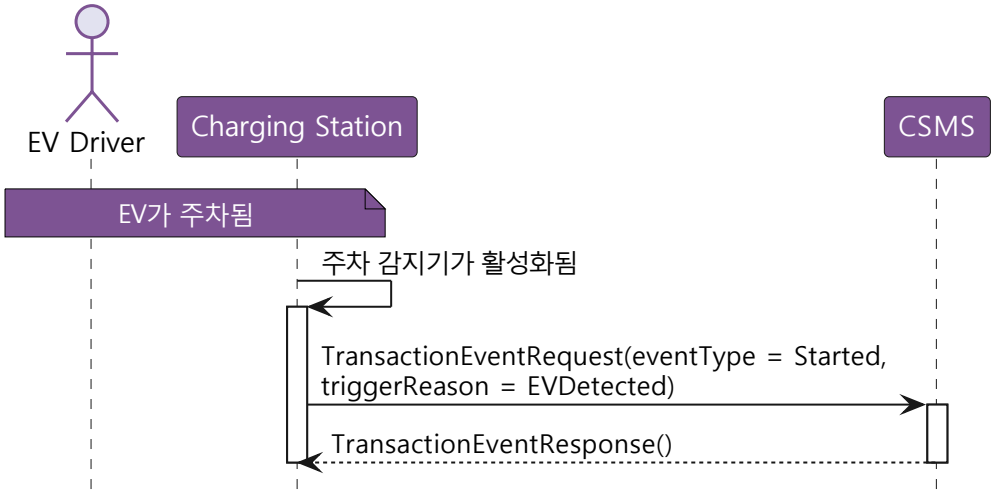

Figure 39. 시퀀스 다이어그램: 시작 트랜잭션 옵션 - ParkingBayOccupancy

S2	*시나리오 목표*	**충전기와 EV + 간에 통신이 설정되면 트랜잭션을 시작합니다(예: 양쪽에 케이블이 올바르게 연결됨)**
	시나리오 설명	**1.** 충전기가 EV와 연결합니다. **2.** 충전기는 TransactionEventRequest (eventType = Started)를 보내 CSMS에 시작된 트랜잭션에 대해 알립니다(운전자가 아직 알려지지 않은 경우에도). **3.** CSMS는 TransactionEventResponse로 응답하여 TransactionEventRequest가 수신되었음을 확인합니다.
	필수 조건	EVSE에서 진행 중인 트랜잭션이 없습니다. 구성 변수: `TxStartPoint` 포함: EVConnected (Not: ParkingBayOccupancy)
	사후 조건	**성공적인 사후 조건:** 트랜잭션이 진행 중이며 CSMS에 *성공적으로* 통보되었습니다. **실패 사후 조건:** 트랜잭션이 *진행되지* 않음, *또는* CSMS에 *알려지지* 않음.

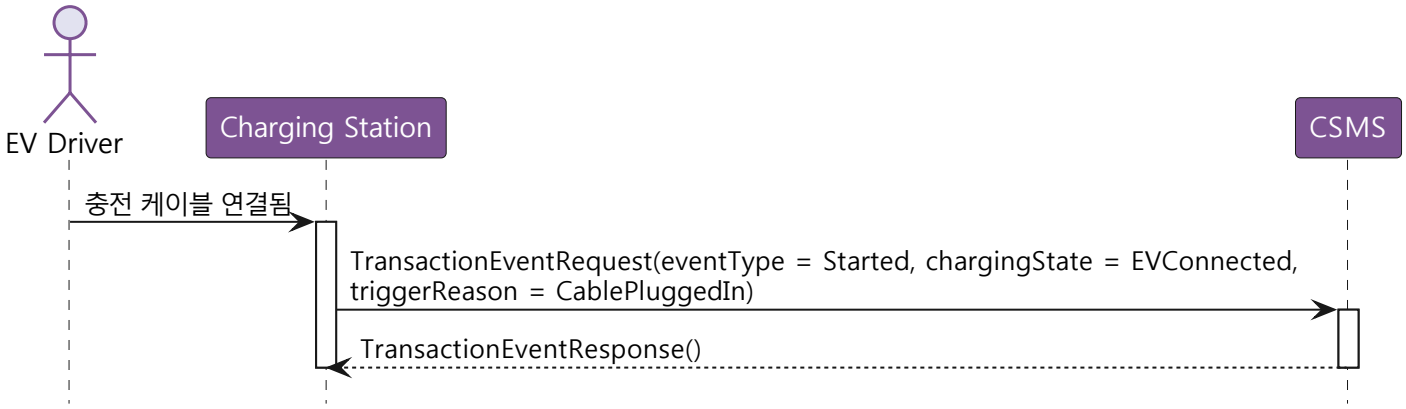

Figure 40. 시퀀스 다이어그램: 트랜잭션 시작 옵션 - EVConnected

S3	*시나리오 목표*	**EV 운전자가 충전을 인증받았을 때 트랜잭션을 시작합니다.**
	시나리오 설명	**1.** EV 운전자가 자신의 신원을 제공합니다 **2.** 충전기가 제공된 신원을 검증합니다(예: 인증 캐시 또는 인증 요청을 통해). **3.** 충전기는 TransactionEventRequest(eventType = Started)를 보내 CSMS에 시작된 트랜잭션에 대해 알립니다. **4.** CSMS는 TransactionEventResponse로 응답하여 TransactionEventRequest가 수신되었음을 확인합니다.

S3	*시나리오 목표*	**EV 운전자가 충전을 인증받았을 때 트랜잭션을 시작합니다.**
	필수 조건	EVSE에서 진행 중인 트랜잭션이 없습니다. 구성 변수: `TxStartPoint`에는 다음이 포함됩니다. Authorized(Not: ParkingBayOccupancy).
	사후 조건	**성공적인 사후 조건:** 트랜잭션이 진행 중이고 CSMS에 *성공적으로* 통보되었습니다. **실패 사후 조건:** 트랜잭션이 *진행 중이 아닙니다, 또는* CSMS에 *통보되지 않았습니다.*

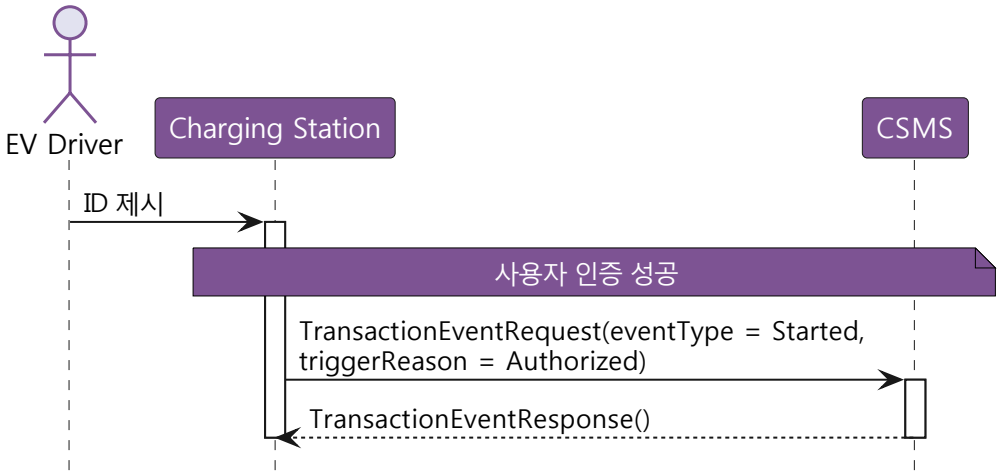

Figure 41. 시퀀스 다이어그램: 트랜잭션 시작 옵션 - 인증됨

S4	*시나리오 목표*	**계량기가 요금 청구를 시작하기 전에 첫 번째 서명된 계량기 값을 제공하면 트랜잭션을 시작합니다.**
	시나리오 설명	**1.** EV 운전자가 충전기와 EV에 케이블을 연결합니다. **2.** 충전기가 미터에 서명된 값을 요청합니다. **3.** 미터가 서명된 값을 제공합니다(시간이 걸릴 수 있음). **4.** 충전기가 TransactionEventRequest(eventType = Started)를 보내 CSMS에 시작된 트랜잭션을 알립니다. **5.** CSMS가 TransactionEventResponse로 응답하여 TransactionEventRequest가 수신되었음을 확인합니다.
	필수 조건	EVSE에서 진행 중인 트랜잭션이 없습니다. 구성 변수: `TxStartPoint`에는 다음이 포함됩니다. DataSigned (Not: ParkingBayOccupancy, EVConnected 또는 Authorized). 충전기에는 측정된 값에 서명할 수 있는 미터가 있습니다. 구성 변수: `SampledDataSignReadings`가 *true* 로 설정됨.
	사후 조건	**성공적인 사후 조건:** 트랜잭션이 진행 중이고 CSMS에 *성공적으로* 통보되었습니다. **실패 사후 조건:** 트랜잭션이 *진행 중이 아닙니다, 또는* CSMS에 *통보되지 않았습니다.*

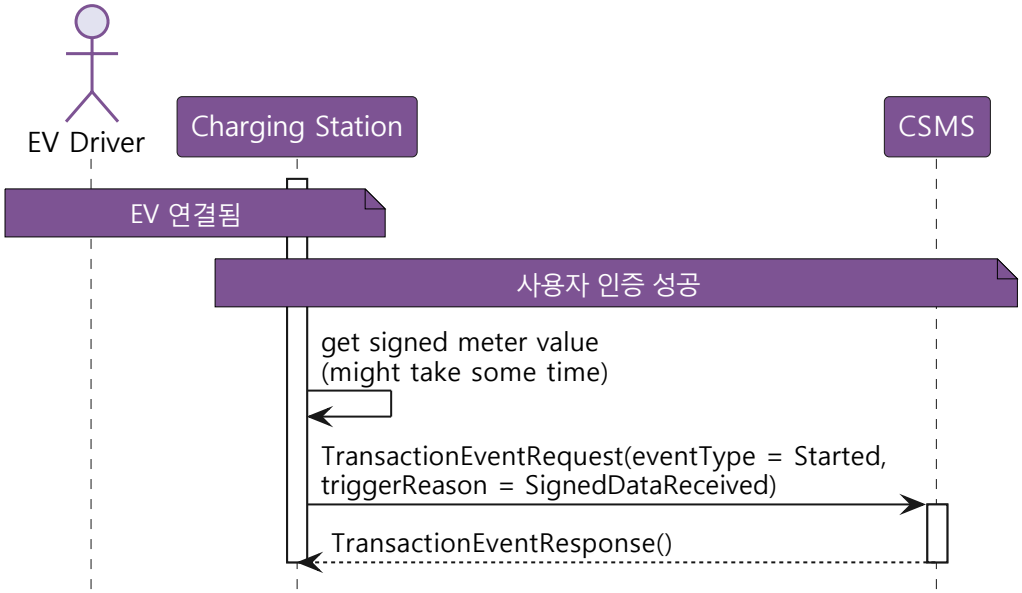

Figure 42. 시퀀스 다이어그램: 트랜잭션 시작 옵션 - DataSigned

S5	*시나리오 목표*	**충전을 시작하기 위한 모든 전제 조건이 충족되었지만(인증 및 연결됨) 아직 에너지가 전송되는 않을 경우에도 트랜잭션을 시작.**
	시나리오 설명	**1.** EV 운전자는 충전기 및/또는 CSMS에서 인증을 받습니다. **2.** 충전기가 EV에 연결됩니다. **3.** 충전기에서 TransactionEventRequest(eventType = Started)를 보내 CSMS에 시작된 트랜잭션을 알립니다. **4.** CSMS에서 TransactionEventResponse로 응답하여 TransactionEventRequest가 수신되었음을 확인합니다.
	필수 조건	EVSE에서 진행 중인 트랜잭션이 없습니다. 구성 변수: `TxStartPoint`에는 다음이 포함됩니다. PowerPathClosed (Not: ParkingBayOccupancy, EVConnected, Authorized 또는 DataSigned). 충전 케이블이 연결되었습니다.
	사후 조건	**성공 사후 조건:** 트랜잭션이 진행 중이고 CSMS에 *성공적으로* 통보되었습니다. **실패 사후 조건:** 트랜잭션이 *진행 중이* 아니거나 CSMS에 *통보되지* 않았습니다.

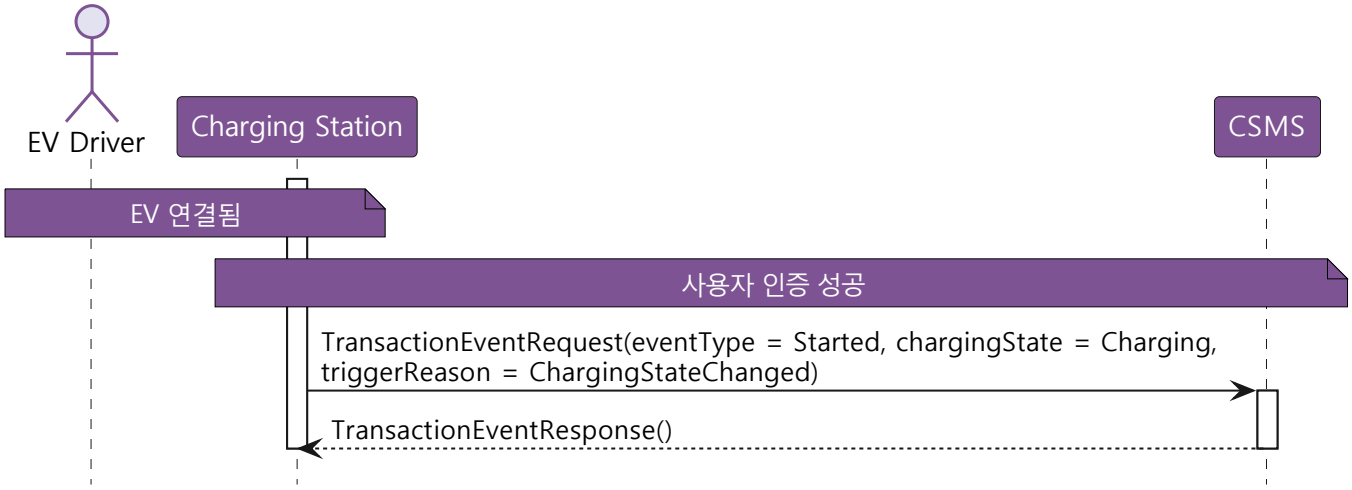

Figure 43. 시퀀스 다이어그램: 시작 트랜잭션 옵션 - PowerPathClosed

S6	*시나리오 목표*	**에너지 전송이 시작될 때 트랜잭션을 시작합니다.**
	시나리오 설명	**1.** EV 운전자는 충전기 및/또는 CSMS에서 인증을 받습니다. **2.** 충전기가 전원 릴레이를 닫습니다. **3.** EV가 충전을 시작하고 에너지 전송이 시작됩니다. **4.** 충전기는 TransactionEventRequest(eventType = Started)를 보내 CSMS에 시작된 트랜잭션에 대해 알립니다. **5.** CSMS는 TransactionEventResponse로 응답하여 TransactionEventRequest가 수신되었음을 확인합니다.
	필수 조건	구성 변수: `TxStartPoint`에는 EnergyTransfer가 포함됩니다(Not: ParkingBayOccupancy, EVConnected, Authorized, DataSigned 또는 PowerPathClosed).
	사후 조건	**성공적인 사후 조건:** 트랜잭션이 진행 중이고 CSMS에 *성공적으로* 통보되었습니다. **실패 사후 조건:** 트랜잭션이 *진행 중이* 아니거나, *또는* CSMS에 *통보되지* 않았습니다.

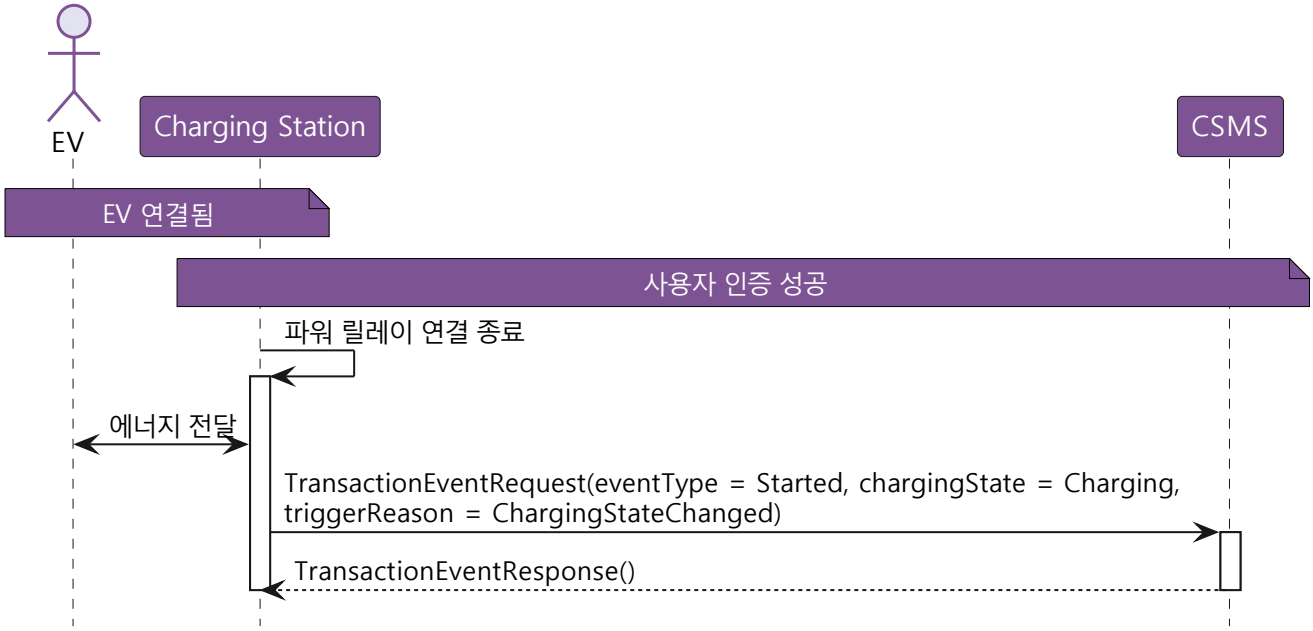

Figure 44. 시퀀스 다이어그램: 시작 트랜잭션 옵션 - EnergyTransfer

7	**오류 처리**	n/a
8	**비고**	n/a

E01 - Start Transaction options - Requirements

Table 97. E01 - 요구 사항

ID	사전 조건	요구 사항 정의
E01.FR.01	TxStartPoint에 다음이 포함됨: ParkingBayOccupancy AND 주차 감지기가 "EV"를 감지합니다. AND 아직 트랜잭션이 시작되지 않았습니다.	충전기는 트랜잭션을 시작하고 TransactionEventRequest (eventType = Started)를 CSMS로 전송해야 합니다.
E01.FR.02	TxStartPoint에는 EVConnected가 포함됩니다. AND 충전기는 EV와 연결되어 있습니다. AND EVSE에서 아직 트랜잭션이 시작되지 않았습니다.	충전기는 트랜잭션을 시작하고 TransactionEventRequest (eventType = Started)를 CSMS로 전송해야 합니다.
E01.FR.03	TxStartPoint에는 다음이 포함됩니다. Authorized AND EV 운전자가 인증됨 AND 아직 트랜잭션이 시작되지 않음	충전기는 트랜잭션을 시작하고 TransactionEventRequest(eventType = Started)를 CSMS로 보내야 합니다.
E01.FR.04	TxStartPoint에는 다음이 포함됩니다. DataSigned AND 충전기에는 측정된 값에 서명할 수 있는 미터가 있습니다. AND 구성 변수: SampledDataSignReadings가 *true* 로 설정됨. AND 충전기에서 서명된 미터 값을 읽었습니다. AND 아직 트랜잭션이 시작되지 않았습니다.	충전기는 트랜잭션을 시작하고 TransactionEventRequest (eventType = Started)를 CSMS로 보내야 합니다.
E01.FR.05	TxStartPoint에는 다음이 포함됩니다. PowerPathClosed AND EV 운전자가 인증되었습니다. AND 충전기가 EV와 연결되어 있습니다. AND EVSE에서 아직 트랜잭션이 시작되지 않았습니다.	충전기는 트랜잭션을 시작하고 TransactionEventRequest (eventType = Started)를 CSMS로 보내야 합니다.
E01.FR.06	TxStartPoint에는 다음이 포함됩니다. EnergyTransfer AND 에너지 전송이 시작됩니다. AND EVSE에서 아직 트랜잭션이 시작되지 않았습니다.	충전기는 트랜잭션을 시작하고 TransactionEventRequest(eventType = Started)를 CSMS로 보내야 합니다.

ID	사전 조건	요구 사항 정의
E01.FR.07	TransactionEventRequest를 만들어야 하는 경우	충전기는 Sequence Number Generation에 지정된 대로 메시지의 **seqNo** 필드를 설정해야 합니다.
E01.FR.08		충전기에서 생성된 transactionId는 충전기가 재부팅되거나 수리되거나 펌웨어가 업데이트되는 경우에도 해당 충전기에서 시작한 각 트랜잭션에 대해 고유해야 하며, 동일한 TransactionId를 두 번 생성하지 않도록 해야 합니다.
E01.FR.09	TransactionEventRequest(eventType = Started)에서 미터 데이터를 보내도록 구성된 경우, Meter Values - Configuration 참조 AND EVSE는 트랜잭션 시작 시 알려져 있음	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Started)에서 *context* = `Transaction.Begin` 인 선택적 meterValue 필드에 추가하여 트랜잭션 중에 더 자세한 정보를 제공해야 합니다.
E01.FR.10	EV 운전자가 이 트랜잭션에 대해 인증을 받은 후	충전기는 IdTokenType 정보가 포함된 TransactionEventRequest를 보내야 합니다.
E01.FR.11	E01.FR.10	CSMS는 TransactionEventRequest의 식별자의 유효성을 확인해야 합니다.
E01.FR.12	E01.FR.11	CSMS는 TransactionEventResponse를 보내야 합니다. 여기에는 *idTokenInfo* 에 인증 상태 값과 *idToken* 에 대한 *groupIdToken* 이 포함되어 있어야 합니다.
E01.FR.13	이 트랜잭션에서 예약을 종료합니다.	다음 TransactionEventRequest에는 reservationId가 포함되어야 합니다.
E01.FR.14	특정 EVSE 및 커넥터에 대해 TransactionEventRequest (eventType = Started)가 전송된 후	충전기는 이 트랜잭션이 종료될 때까지 동일한 EVSE의 다른 커넥터에서 다른 트랜잭션을 시작해서는 안 됩니다.
E01.FR.15	TransactionEventRequest를 전송할 때	충전기는 triggerReason을 설정하여 CSMS에 이벤트를 트리거한 원인을 알려야 합니다. 사용할 이유는 TriggerReasonEnumType의 설명에 설명되어 있습니다.
E01.FR.16	EV가 충전기에 연결된 후	다음 TransactionEventRequest에는 *evse.id* 와 *evse.connectorId* 가 포함되어야 합니다.
E01.FR.17	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Started), 다음을 참조하세요: Meter Values - Configuration AND EVSE는 트랜잭션 시작 시 알려지지 않음	충전기는 *eventType* = `Started` 에 대한 측정값을 충전이 시작될 때 발생하는 TransactionEventRequest(eventType = Updated)의 컨텍스트 `Transaction.Begin` 과 함께 선택적 meterValue 필드에 추가해야 합니다.
E01.FR.18	충전 상태가 변경되는 경우	충전기는 *chargingState* 요소를 포함한 TransactionEventRequest를 보내야 합니다.
E01.FR.19	EV가 에너지 전송을 일시적으로 중단할 때	충전기는 *chargingState* = `SuspendedEV` 인 TransactionEventRequest를 보내야 함
E01.FR.20	E01.FR.19 AND 충전기가 에너지 전송의 일시 중단을 처리할 수 없음	충전기는 *chargingState* = `EVConnected` 인 TransactionEventRequest를 보내야 함.

E02 - Start Transaction - Cable Plugin First

Table 98. E02 - 트랜잭션 시작 - 케이블 플러그인 먼저

번호	유형	설명
1	**이름**	트랜잭션 시작 - 케이블 플러그인 먼저
2	**ID**	E02
	기능 블록	E. 트랜잭션
3	**목표**	충전기에서 트랜잭션이 시작되었음을 CSMS에 알립니다.
4	**설명**	EV 운전자는 먼저 충전 케이블을 연결하여 충전기와 상호 작용을 시작합니다. CSMS는 이에 대한 알림을 전송받습니다. 그런 다음 EV와 EVSE 간의 통신이 설정되면 트랜잭션이 시작되고 CSMS에 이에 대한 알림을 전송받습니다. EV가 충전을 시작합니다.
	Actors	충전기, CSMS, EV 운전자
	시나리오 설명	**1.** EV 운전자가 충전기에 케이블을 연결합니다. **2.** 충전기는 StatusNotificationRequest를 CSMS에 보내서 *Occupied*가 된 Connector에 대해 알립니다. **3.** 충전기는 TransactionEventRequest(eventType = Started)를 보내서 CSMS에 시작된 트랜잭션에 대해 알립니다(운전자가 아직 알려지지 않은 경우에도 해당). **4.** CSMS는 TransactionEventResponse로 응답하여 TransactionEventRequest가 수신되었음을 확인합니다. **5.** EV 운전자는 충전기 및/또는 CSMS에서 인증을 받습니다. **6.** 에너지 전송이 시작됩니다. **7.** 충전기는 TransactionEventRequest (eventType = Updated)를 권한 있는 idToken 정보와 함께 CSMS로 보내 충전 상태와 트랜잭션에 속하는 idToken을 알립니다. **8.** CSMS는 TransactionEventResponse를 IdTokenInfo.status *Accepted* 와 함께 충전기에 응답합니다. **9.** 충전 프로세스 동안 충전기는 트랜잭션 관련 알림을 위해 TransactionEventRequest (Updated) 메시지를 계속 보냅니다.
	대안 시나리오	E03 - Start Transaction - IdToken First E04 - Offline Start Transaction E05 - Start Transaction - Id not Accepted
5	**필수 조건**	먼저 충전 케이블을 꽂습니다.
6	**사후 조건**	**성공한 사후 조건:** 트랜잭션이 진행 중이고 CSMS에 *성공적으로* 알립니다. **실패 사후 조건:** 트랜잭션이 *진행 중이 아닙니다. 또는* CSMS에 *알리지 않습니다. 또는* 트랜잭션 시작 - ID가 수락되지 않습니다.

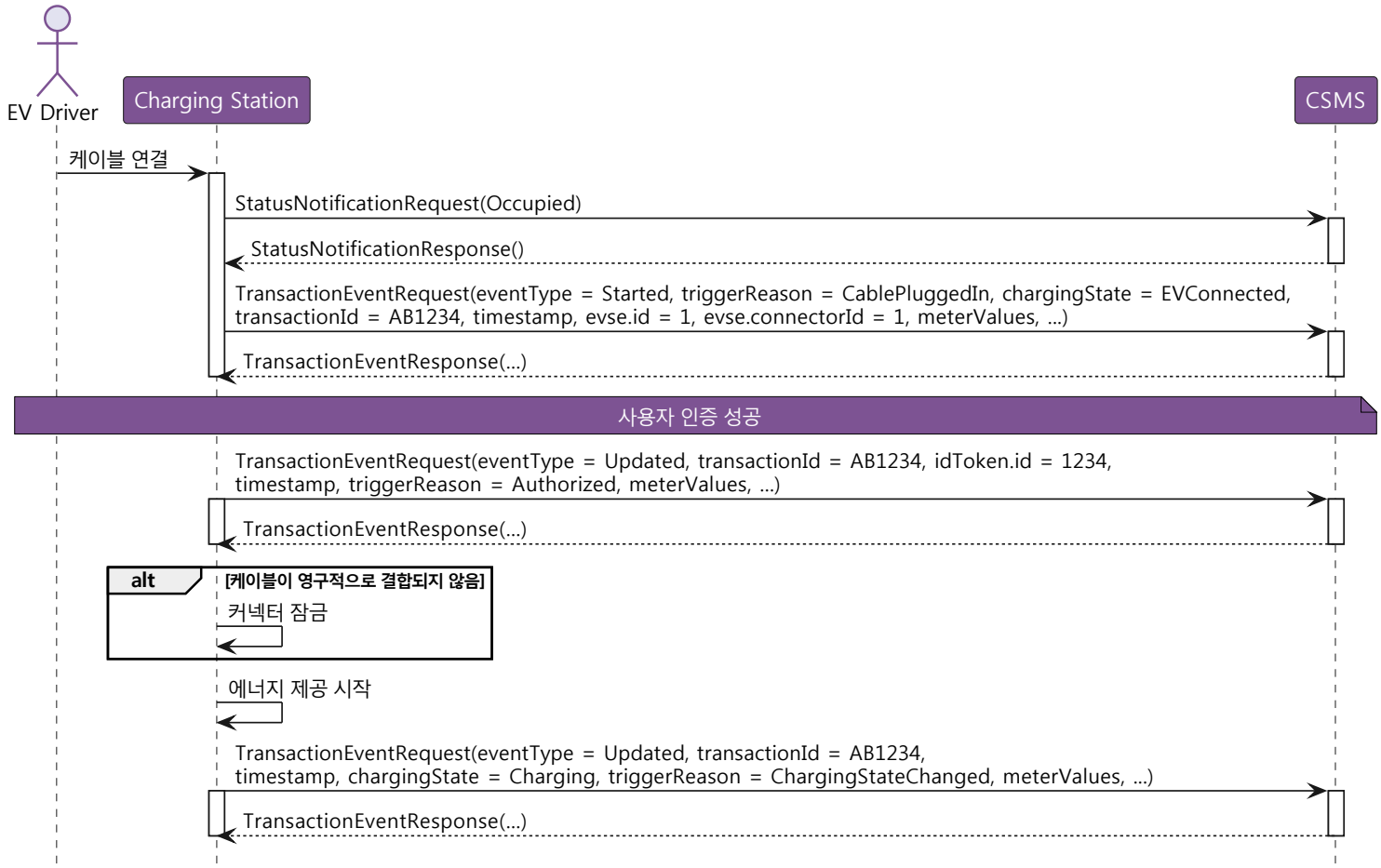

Figure 45. Sequence Diagram: Start Transaction - Cable Plugin First

7	**오류 처리**	TransactionEventResponse로 응답하지 못하면 충전기는 E12 - Transaction-related message not accepted by CSMS에 지정된 대로 동일한 메시지를 다시 시도합니다.
8	**참고**	충전기가 인증 캐시를 구현한 경우 TransactionEventResponse를 수신하면 충전기는 캐시 항목을 업데이트합니다. 위의 시나리오 설명과 시퀀스 다이어그램은 트랜잭션 시작 및 중지에 대한 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStartPoint`: EVConnected, Authorized, DataSigned, PowerPathClosed, EnergyTransfer 이 유즈케이스는 다른 구성에서도 유효하지만, 그 경우 트랜잭션이 다른 시점에서 시작/종료될 수 있으며, 이는 메시지가 전송되는 순서에 영향을 미칠 수 있습니다. 자세한 내용은 사용 사례를 참조하세요. E01 - Start Transaction options 및 E06 - Stop Transaction options.

E02 - Start Transaction - Cable Plugin First - Requirements

Table 99. E02 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
E02.FR.01	EV 운전자가 이 트랜잭션에 대해 인증된 후.	다음 TransactionEventRequest에는 *triggerReason*: Authorized 그리고 IdTokenType 정보가 포함되어야 합니다.	
E02.FR.02	E02.FR.01	CSMS는 인증 상태 값을 포함하는 TransactionEventResponse를 보내야 합니다.	
E02.FR.03	트랜잭션이 예약을 종료합니다.	다음 TransactionEventRequest는 reservationId를 포함해야 합니다.	H. Reservation을 참조하세요.

ID	전제 조건	요구 사항 정의	참고
E02.FR.04		CSMS는 TransactionEventRequest에서 식별자의 유효성을 확인해야 합니다	식별자가 오래된 정보를 사용하여 충전기에서 로컬로 인증되었을 수 있기 때문입니다.
E02.FR.05	케이블이 꽂혀 있을 때	충전기는 StatusNotificationRequest를 상태 *Occupied*로 보내야 합니다.	대신 **component**(name = 'Connector', evse.id = <*x*>, evse.connectorId = <*y*>), variable(name = 'AvailabilityState'), **actualValue** = 'Occupied'에 대한 NotifyEventRequest 메시지를 보내 EVSE <*x*> 의 Connector <*y*> 가 현재 점유되었음을 알릴 수 있습니다.(MAY)
E02.FR.06	케이블이 꽂혀 있고 `TxStartPoint` 에 EVConnected가 포함되어 있을 때	충전기는 TransactionEventRequest를 보내야 합니다.	
E02.FR.07	TransactionEventRequest를 생성해야 하는 경우	충전기는 Sequence Number Generation에 지정된 대로 메시지의 **seqNo** 필드를 설정해야 합니다.	이렇게 하면 CSMS가 트랜잭션 정보의 완전성을 추적할 수 있습니다.
E02.FR.08		충전기에서 생성된 transactionId는 충전기가 재부팅, 수리, 펌웨어 업데이트 등을 수행하더라도 해당 충전기에서 시작한 각 트랜잭션에 대해 고유해야 하며, 동일한 TransactionId를 두 번 생성하지 않도록 해야 합니다.	
E02.FR.09	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Started), 다음을 참조하세요: Meter Values - Configuration AND EVSE는 트랜잭션 시작 시 알려져 있음	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Started)의 *context* = `Transaction.Begin` 인 선택적 meterValue 필드에 추가하여 트랜잭션 중에 더 자세한 정보를 제공해야 합니다.	
E02.FR.10	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Updated), 다음을 참조하세요: Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Updated)의 선택적 meterValue 필드에 추가하여 트랜잭션 중에 더 자세한 정보를 제공해야 합니다.	
E02.FR.11	E02.FR.10 AND TransactionEventRequest에 대한 미터 데이터 양이 너무 많은 경우(eventType = Updated)	충전기는 미터 데이터를 동일한 *timestamp* 를 사용하여 여러 TransactionEventRequest (eventType = Updated) 메시지로 분할할 수 있습니다.	
E02.FR.13	충전 상태가 변경되는 경우	충전기는 chargingState 요소를 포함하여 TransactionEventRequest를 보내야 합니다.	
E02.FR.14	`SampledDataSignReadings`는 *true* 입니다.	충전기는 서명된 미터 값을 검색하여 sampledValues의 *signedMeterValue* 필드에 넣어야 합니다.	

ID	전제 조건	요구 사항 정의	참고
E02.FR.15	TransactionEventRequest를 보낼 때	충전기는 triggerReason을 설정하여 CSMS에 이벤트를 트리거한 원인을 알려야 합니다. 사용할 이유는 TriggerReasonEnumType의 설명에 설명되어 있습니다.	
E02.FR.16	트랜잭션이 시작된 후	충전기는 trigger event가 발생할 때 트랜잭션 중에 추가 TransactionEventRequest(eventType = Updated) 메시지를 보낼 수 있습니다.(MAY)	
E02.FR.17	*TriggerReasonEnumType* 에 나열된 트랜잭션 관련 트리거 이벤트가 발생하고 트랜잭션이 진행 중인 경우.	충전기는 발생한 이벤트에 해당하는 *triggerReason* 과 함께 TransactionEventRequest를 보내야 합니다.	두 개의 트리거 이유가 겹치는 경우 더 구체적인 사유를 사용해야 합니다. 예를 들어 케이블이 연결된 경우 *ChargingStateChanged* 가 아닌 triggerReason = *CablePluggedIn* 을 사용해야 합니다. 하지만 각 트리거에 대해 별도의 TransactionEventRequest 메시지를 보내는 것이 금지된 것은 아닙니다.
E02.FR.18	에너지 전송이 시작될 때 AND 충전기가 사용된 단계 수를 보고할 수 있는 경우	충전기는 *numberOfPhasesUsed* 필드를 사용하여 사용된 단계 수를 제공해야 합니다.	
E02.FR.19	E02.FR.18 AND 트랜잭션 중에 사용된 단계 수가 변경되는 경우	충전기는 *numberOfPhasesUsed* 필드를 사용하여 조정된 사용된 단계 수를 제공해야 합니다.	
E02.FR.20	트랜잭션이 이전에 인증되지 않았고 AND 충전기가 *idToken* 을 인증하여 충전을 시작하는 경우	충전기의 다음 TransactionEventRequest는 *idToken* 을 포함하고 *triggerReason* = `Authorized` 를 가져야 합니다.	인증이 성공하지 못하면 이 이벤트가 실행 중인 트랜잭션에 영향을 미치지 않으므로 TransactionEventRequest가 전송되지 않습니다. (충전 중지를 위한 인증은 E07 참조).
E02.FR.21	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Started), 다음을 참조하세요: Meter Values - Configuration AND EVSE는 트랜잭션 시작 시 알려져있지 않음	충전기는 충전이 시작될 때 발생하는 TransactionEventRequest(eventType = Updated)에서 *context* = `Transaction.Begin` 인 선택적 meterValue 필드에 *eventType* = `Started` 에 대한 측정값을 추가해야 합니다.	

E03 - Start Transaction - IdToken First

Table 100. E03 - 트랜잭션 시작 - IdToken 먼저

번호	유형	설명
1	**이름**	트랜잭션 시작 - IdToken 먼저
2	**ID**	E03
	기능 블록	E. 트랜잭션
3	**목표**	EV 운전자가 충전기에서 IdToken을 먼저 제시하여 트랜잭션을 시작할 수 있도록 합니다.
4	**설명**	이 사용 사례는 EV 운전자가 케이블을 연결해서 트랜잭션이 시작되기 전에 IdToken을 제시하여 먼저 인증되는 방법을 다룹니다.
	Actors	충전기, CSMS, EV 운전자
	시나리오 설명	**1.** EV 운전자가 충전기 및/또는 CSMS에서 인증됩니다. **2.** 충전기는 TransactionEventRequest(eventType = Started)를 보내 트랜잭션이 시작되었음을 CSMS에 알립니다. **3.** EV 운전자가 충전기에 충전 케이블을 연결합니다. **4.** 충전기는 StatusNotificationRequest를 CSMS로 보내고 StatusNotificationResponse를 수신합니다. **5.** 충전기는 TransactionEventRequest(eventType = Updated, chargingState = Charging)를 전송하여 CSMS에 EV가 충전을 시작했다고 알립니다. **6.** CSMS는 TransactionEventResponse로 응답하여 트랜잭션을 수락합니다.
5	**필수 조건**	케이블 연결보다 먼저 IdToken이 제공됩니다.
6	**사후 조건**	**성공한 사후 조건:** 트랜잭션이 시작되고 ChargingState가 *Charging* **실패 사후 조건:** 트랜잭션이 시작되지 않음

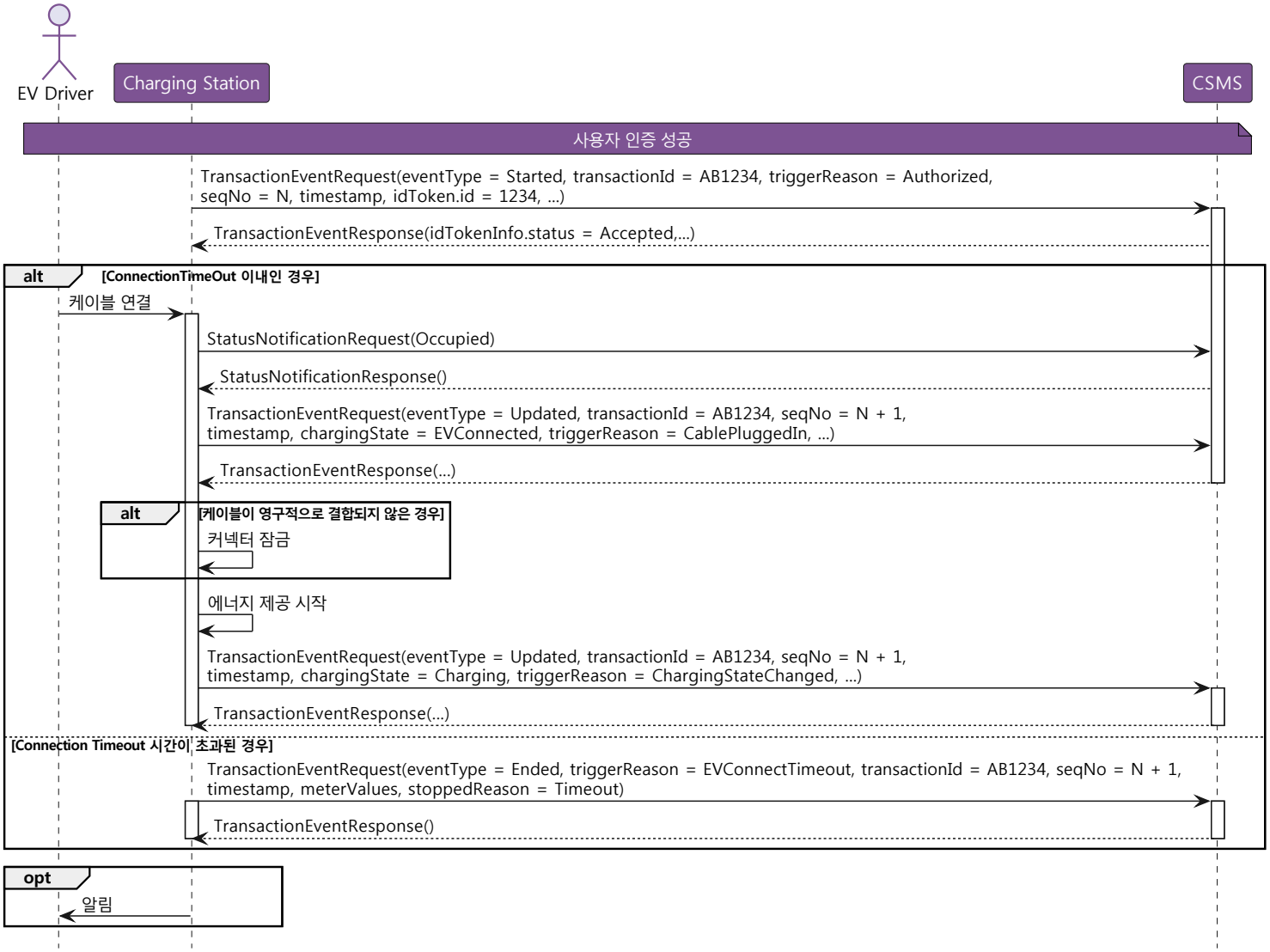

Figure 46. 시퀀스 다이어그램: 트랜잭션 시작 - 먼저 IdToken

7	**오류 처리**	n/a
		CSMS가 수신한 TransactionEventRequest 메시지에 포함된 데이터에 적합성 테스트를 적용할 가능성이 있습니다. 이러한 테스트의 결과는 CSMS가 TransactionEventResponse를 응답하지 않도록 해서는 안 됩니다. 그렇지 않으면 충전기가 E12 - Transaction-related message not accepted by CSMS에 지정된 대로 동일한 메시지를 다시 시도합니다.
8	**참고**	위의 시나리오 설명과 시퀀스 다이어그램은 시작 트랜잭션에 대한 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStartPoint`: Authorized, DataSigned, PowerPathClosed, EnergyTransfer 이 유즈케이스는 다른 구성에서도 유효하지만, 그 경우 트랜잭션이 다른 시점에서 시작/종료될 수 있으며, 이는 메시지가 전송되는 순서에 영향을 미칠 수 있습니다. 자세한 내용은 사용 사례를 참조하세요: E01 - Start Transaction options.

E03 - Start Transaction - IdToken First - Requirements

Table 101. E03 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
E03.FR.01	IdToken 정보가 알려진 경우.	다음 TransactionEventRequest에는 IdTokenType 정보가 포함되어야 합니다.	

ID	전제 조건	요구 사항 정의	참고
E03.FR.02	E03.FR.01	CSMS는 인증 상태를 포함하는 TransactionEventResponse를 보내야 합니다.	
E03.FR.03	이 트랜잭션에서 특정 IdToken에 대한 예약을 종료합니다.	다음 TransactionEventRequest는 reservationId를 포함해야 합니다.	H. Reservation을 참조하세요.
E03.FR.05	EV 운전자가 `EVConnectionTimeOut` 구성 변수에서 설정한 시간 초과 전에 충전 케이블을 연결하지 않은 경우 및 TxStopPoint에 `ParkingBayOccupancy` 가 포함되지 않은 경우	충전기는 트랜잭션을 종료하고 TransactionEventRequest(eventType = Ended, ceaseReason = Timeout, *triggerReason* = `EVConnectionTimeout`)를 CSMS로 보내야 합니다.	이 요구 사항은 `EVConnectionTimeOut`이 트리거될 때 트랜잭션이 종료되도록 하는 추가 안전 조치입니다. 그러나 합리적인 TxStartPoint/TxStopPoint 조합이 구성되었는지 확인하는것은 CSMS에 달려있습니다. 예: Authorized가 TxStartPoint로 사용되는 경우 TxStopPoint로도 사용해야 합니다.
E03.FR.06	TransactionEventRequest를 생성해야 하는 경우	충전기는 Sequence Number Generation에 지정된 대로 메시지의 **seqNo** 필드를 설정해야 합니다.	이렇게 하면 CSMS가 트랜잭션 정보의 완전성을 추적할 수 있습니다.
E03.FR.07	TransactionEventRequest(eventType = Started)에서 미터 데이터를 보내도록 구성된 경우 다음을 참조하세요. Meter Values - Configuration AND EVSE는 트랜잭션 시작 시 알려져 있습니다.	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Started)에서 *context* = `Transaction.Begin` 인 선택적 meterValue 필드에 추가하여 트랜잭션 중에 더 자세한 정보를 제공해야 합니다.	
E03.FR.08	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Updated), 다음을 참조하세요: Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Updated)의 선택적 MeterValue 필드에 추가하여 트랜잭션 중에 더 자세한 정보를 제공해야 합니다.	
E03.FR.09	E03.FR.08 AND TransactionEventRequest에 대한 미터 데이터 양이 너무 많은 경우(eventType = Updated)	충전기는 동일한 *timestamp* 를 사용하여 여러 TransactionEventRequest(eventType = Updated) 메시지로 미터 데이터를 분할할 수 있습니다.	
E03.FR.10	`SampledDataSignReadings`는 *true* 입니다.	충전기는 서명된 미터 값을 검색하여 sampledValues의 *signedMeterValue* 필드에 넣어야 합니다.	
E03.FR.11	TransactionEventRequest(eventType = Started)에서 미터 데이터를 보내도록 구성된 경우 다음을 참조하세요. Meter Values - Configuration AND 트랜잭션 시작 시 EVSE가 알려지지 않음	충전기는 충전이 시작될 때 발생하는 TransactionEventRequest(eventType = Updated)에서 *context* = `Transaction.Begin` 인 선택적 meterValue 필드에 *eventType* = `Started` 에 대한 측정량을 추가해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
E03.FR.12	*TriggerReasonEnumType* 에 나열된 트랜잭션 관련 트리거 이벤트가 발생하고 트랜잭션이 진행 중인 경우	충전기는 발생한 이벤트에 해당하는 triggerReason과 함께 TransactionEventRequest를 보내야 합니다.	두 개의 트리거 사유가 겹치는 경우 더 구체적인 사유를 사용해야 합니다. 예를 들어 케이블이 꽂혀 있는 경우 *ChargingStateChanged* 가 아닌 triggerReason = *CablePluggedIn* 을 사용해야 합니다. 두 이벤트가 동시에 발생하는 경우 두 개의 별도 TransactionEventRequest 메시지를 사용하여 전송해야 합니다. 이는 문제가 발생할 때 정보 손실을 방지하기 위한 것입니다.
E03.FR.13	에너지 전송이 시작될 때 AND 충전기가 사용된 페이즈 수를 보고할 수 있는 경우	충전기는 *numberOfPhasesUsed* 필드를 사용하여 사용된 페이즈 수를 제공해야 합니다.	
E03.FR.14	E03.FR.13 AND 트랜잭션 중에 사용된 페이즈 수가 변경되는 경우	충전기는 *numberOfPhasesUsed* 필드를 사용하여 조정된 사용된 페이즈 수를 제공해야 합니다.	
E03.FR.15	EV 운전자가 구성 변수에서 설정한 시간 초과 전에 충전 케이블을 연결하지 않은 경우: `EVConnectionTimeOut` AND TxStopPoint에 `ParkingBayOccupancy` 가 포함됨	충전기는 트랜잭션을 인증 취소하고 TransactionEventRequest (*triggerReason* = `EVConnectionTimeout`)를 CSMS로 보내야 합니다.	운전자가 주차장을 떠나면 트랜잭션이 정상적으로 종료됩니다.

E04 - Transaction started while Charging Station is offline

Table 102. E04 - 충전기가 오프라인인 동안 트랜잭션이 시작됨.

번호	유형	설명
1	**이름**	충전기가 오프라인일 동안 트랜잭션이 시작됨.
2	**ID**	E04
	기능 블록	E. 트랜잭션
3	**목표**	충전기가 *offline* 인 동안 EV 운전자가 트랜잭션을 시작할 수 있도록 합니다.
4	**설명**	이 사용 사례는 충전기가 *offline* 일 동안 로컬 인증 목록 또는 인증 캐시를 사용하여 트랜잭션을 시작할 수 있는 방법을 다룹니다.
	Actors	충전기, CSMS, EV 운전자
	시나리오 설명	**1.** 트랜잭션이 시작됩니다. **2.** TransactionEventRequest (eventType = Started)가 충전기에 의해 저장/대기됩니다. **3.** 충전기와 CSMS 간의 연결이 복구됩니다. **4.** 충전기가 대기 중인 메시지를 보내기 시작합니다. **5.** 저장된 TransactionEventRequest가 전송되어 CSMS에 시작된 트랜잭션에 대해 알립니다.
	대체 시나리오	E10 - Connection Loss During Transaction
5	**필수 조건**	충전기가 *offline* 입니다. EV 운전자가 충전기에서 오프라인/로컬로 인증되었습니다.
6	**사후 조건**	**성공한 사후 조건:** TransactionEventRequest가 CSMS에 의해 응답되었고 충전기 대기열에서 제거되었습니다. **실패 사후 조건:** TransactionEventRequest가 CSMS에 의해 응답되지 않았고 충전기 대기열에 남아 있습니다.

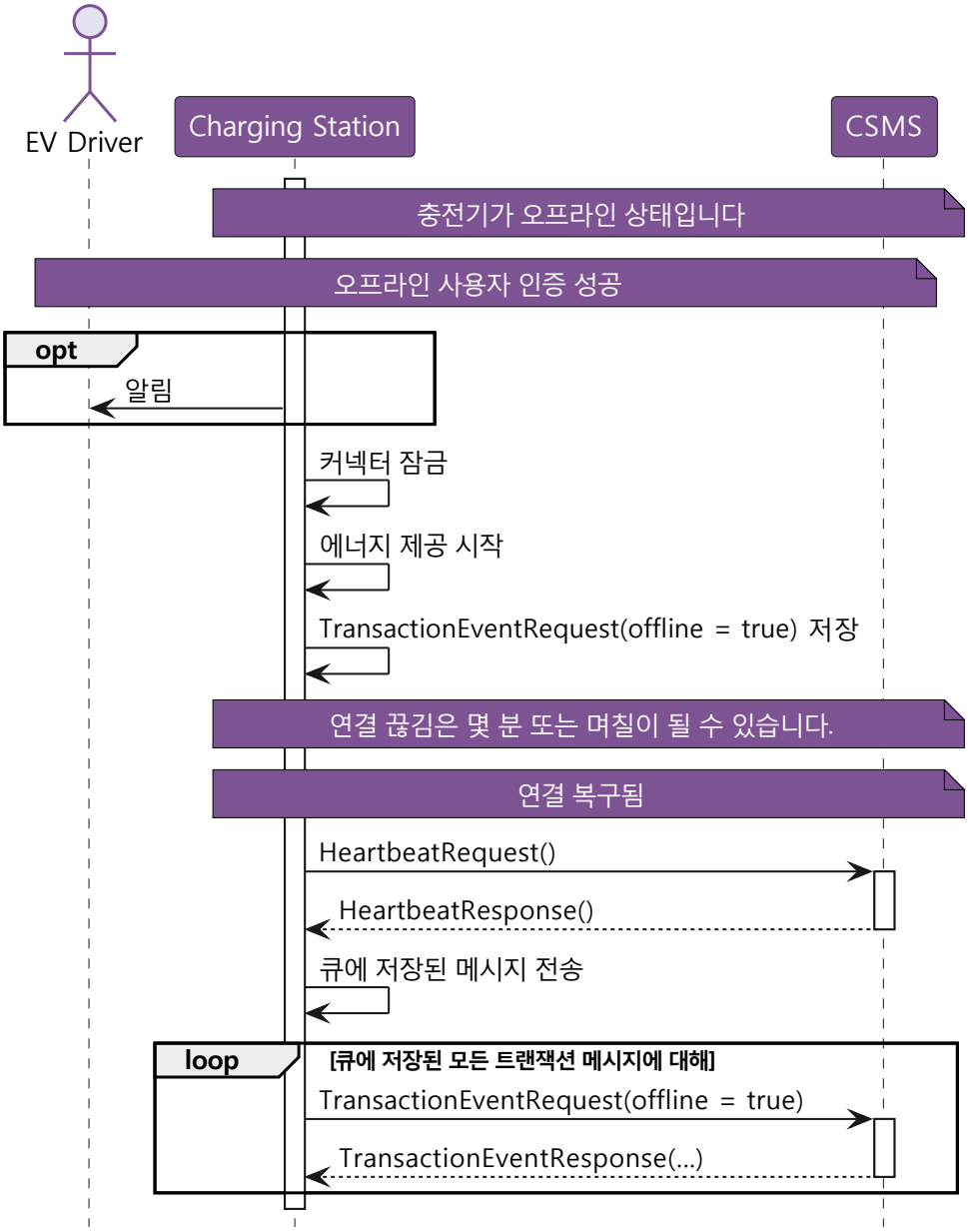

Figure 47. 시퀀스 다이어그램: 충전기가 오프라인일 때 트랜잭션이 시작됨

7	**오류 처리**	n/a
8	**참고**	위의 시나리오 설명과 시퀀스 다이어그램은 시작 트랜잭션에 대한 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStartPoint`: Authorized, DataSigned, PowerPathClosed, EnergyTransfer 이 유즈케이스는 다른 구성에서도 유효하지만, 그 경우 트랜잭션이 다른 시점에서 시작/종료될 수 있으며, 이는 메시지가 전송되는 순서에 영향을 미칠 수 있습니다. 자세한 내용은 사용 사례를 참조하세요. E01 - Start Transaction options.

E04 - Transaction started while Charging Station is offline -Requirements

Table 103. E04 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
E04.FR.01	*offline* 일 때.	충전기는 모든 TransactionEventRequest 메시지를 큐에 넣어야 합니다.	

ID	전제 조건	요구 사항 정의	참고
E04.FR.02	연결이 복구된 후.	충전기는 큐에 넣은 TransactionEventRequest 메시지를 보내야 합니다.	
E04.FR.03	E04.FR.02	플래그: "오프라인"은 충전기가 오프라인 상태일 때 발생한 모든 TransactionEventRequest에 대해 TRUE로 설정되어야 합니다.	
E04.FR.04	TransactionEventRequest를 생성해야 하는 경우	충전기는 Sequence Number Generation에 지정된 대로 메시지의 **seqNo** 필드를 설정해야 합니다.	이렇게 하면 CSMS가 트랜잭션 정보의 완전성을 추적할 수 있습니다.
E04.FR.05	TransactionEventRequest(eventType = Started)에서 미터 데이터를 보내도록 구성된 경우 다음을 참조하세요. Meter Values - Configuration AND EVSE는 트랜잭션 시작 시 알려져 있습니다.	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Started)에서 *context* = `Transaction.Begin` 인 선택적 meterValue 필드에 추가하여 트랜잭션 중에 더 자세한 정보를 제공해야 합니다.	
E04.FR.06	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Updated), 다음을 참조하세요: Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Updated)의 선택적 meterValue 필드에 추가하여 트랜잭션 중에 더 자세한 정보를 제공해야 합니다.	
E04.FR.07	E04.FR.06 AND *Offline* AND 충전기의 메모리가 부족합니다.	충전기는 TransactionEventRequest(eventType = Updated) 메시지를 삭제할 수 있습니다.	
E04.FR.08	E04.FR.07	TransactionEventRequest(eventType = Updated) 메시지를 삭제할 때 충전기는 중간 메시지를 먼저 삭제해야 하며(첫 번째 메시지, 세 번째 메시지, 다섯 번째 메시지 등), 처음부터 메시지를 삭제하거나 큐에 메시지를 추가하는 것을 중단해서는 안 됩니다.	
E04.FR.09	E04.FR.06 AND TransactionEventRequest(eventType = Updated)에 대한 미터 데이터 양이 너무 많은 경우.		충전기는 미터 데이터를 같은 *timestamp* 를 가진 여러 TransactionEventRequest(eventType = Updated) 메시지로 분할할 수 있습니다.
	E04.FR.10	`SampledDataSignReadings`는 *true* 입니다.	충전기는 서명된 미터 값을 검색하여 sampledValues의 *signedMeterValue* 필드에 넣어야 합니다.

ID	전제 조건	요구 사항 정의	참고
	E04.FR.11	TransactionEventRequest(eventType = Started)에서 미터 데이터를 보내도록 구성된 경우 다음을 참조하세요. Meter Values - Configuration AND EVSE는 트랜잭션 시작 시 알려져있지 않음	충전기는 충전이 시작될 때 발생하는 TransactionEventRequest(eventType = Updated)에서 *context* = `Transaction.Begin` 인 선택적 meterValue 필드에 *eventType* = `Started` 에 대한 측정값을 추가해야 합니다.

E05 - Start Transaction - Id not Accepted

Table 104. E05 - Start Transaction - Id not Accepted

번호	유형	설명
1	**이름**	트랜잭션 시작 - ID가 허용되지 않음
2	**ID**	E05
	기능 블록	E. 트랜잭션
3	**목표**	충전기가 IdToken에 충전을 허용하지 않는 AuthorizationStatus가 있는 경우 트랜잭션을 일시 중단할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSMS에서 IdToken을 수락하지 않는 동안 충전기가 트랜잭션을 시작하려는 방법을 다룹니다. 식별자가 오래된 정보를 사용하여 충전기에서 로컬로 인증되었을 수 있으므로 CSMS는 IdTokenType이 포함된 모든 TransactionEventRequest 메시지에서 IdTokenType을 검증해야 합니다. **idTokenInfo** 필드 **status** 가 **Accepted** 가 아닌 TransactionEventResponse 메시지를 수신하면 충전기는 EV에 대한 에너지 공급을 중단해야 합니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 충전기는 EV 운전자가 제공한 IdToken이 포함된 TransactionEventRequest(eventType = Started)를 보냅니다. **2.** CSMS는 충전을 허용하지 않는 AuthorizationStatus와 함께 TransactionEventResponse로 응답합니다. **3.** 충전기는 에너지 제공을 중단합니다. (고려: `MaxEnergyOnInvalidId`, 지원되는 경우) **4.** 충전기는 trigger = *Deauthorized* 와 충전 상태 *SuspendedEVSE* 와 함께 TransactionEventRequest (eventType = Updated)를 보내고 CSMS에서 TransactionEventResponse를 수신합니다.
5	**필수 조건**	EV 운전자가 충전기에서 오프라인/로컬로 인증되었습니다. CSMS에서 IdToken을 충전할 수 없습니다.
6	**사후 조건**	**성공한 사후 조건:** 트랜잭션은 계속 진행되고 케이블은 잠긴 상태로 유지되지만 에너지는 전달되지 않습니다. **실패 사후 조건:** n/a

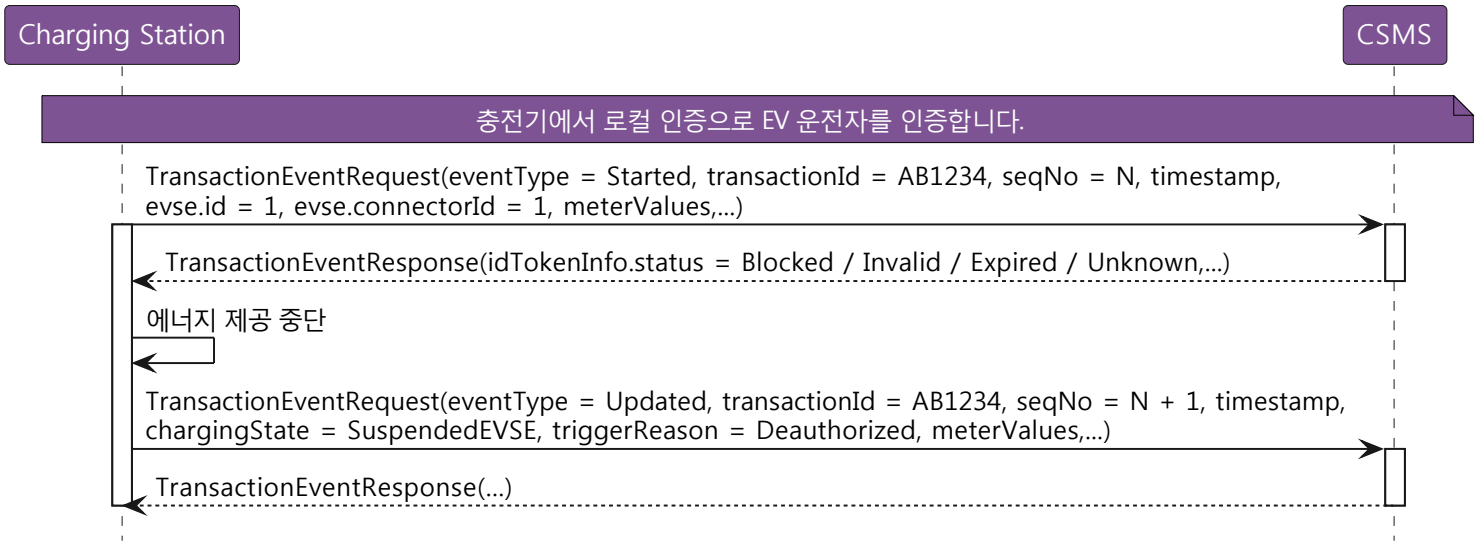

Figure 48. 시퀀스 다이어그램: 시작 트랜잭션 - ID가 허용되지 않음

7	**오류 처리**	n/a
8	**참고**	위의 시나리오 설명과 시퀀스 다이어그램은 시작 및 중지 트랜잭션에 대한 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStartPoint`: Authorized, DataSigned, PowerPathClosed, EnergyTransfer `TxStopPoint`: ParkingBayOccupancy, EVConnected 이 유즈케이스는 다른 구성에서도 유효하지만, 그 경우 트랜잭션이 다른 시점에서 시작/종료될 수 있으며, 이는 메시지가 전송되는 순서에 영향을 미칠 수 있습니다. 자세한 내용은 사용 사례를 참조하세요: E01 - Start Transaction options 및 E06 - Stop Transaction options.

E05 - Start Transaction - Id not Accepted - Requirements

Table 105. E05 - Requirements

ID	전제 조건	요구 사항 정의	참고
E05.FR.01		CSMS는 TransactionEventRequest 메시지의 식별자 유효성을 확인해야 합니다.	식별자는 오래된 정보를 사용하여 충전기에서 로컬로 인증되었을 수 있습니다. 예를 들어 식별자는 충전기의 인증 캐시에 추가된 이후 차단되었을 수 있습니다.
E05.FR.02	E05.FR.01 그리고 TransactionEventResponse의 인증 상태가 *Accepted* 가 아닙니다. 그리고 트랜잭션이 아직 진행 중입니다. 그리고 `StopTxOnInvalidId`가 *false* 로 설정되었습니다. 그리고 `MaxEnergyOnInvalidId`가 구현되지 않았거나 초과되었습니다. `TxStopPoint`에는 다음이 포함되지 않습니다. EnergyTransfer	충전기는 EV로의 에너지 공급을 즉시 중단하고 TransactionEventRequest(eventType = Updated)를 보내야 하며, *triggerReason* 은 *ChargingStateChanged* 로, chargingState는 *SuspendedEVSE* 로 설정해야 합니다.	트랜잭션은 deauthorized되지 않지만, `MaxEnergyOnInvalid` 가 초과되었거나 설정되지 않았기 때문에 에너지 전송이 중단됩니다. `TxStopPoint` 에 `EnergyTransfer` 가 포함되어 있으면 트랜잭션이 종료됩니다.
E05.FR.03	E05.FR.01 AND TransactionEventResponse의 인증 상태가 *Accepted* 가 아닙니다 AND 트랜잭션이 아직 진행 중입니다 AND `StopTxOnInvalidId`가 *false* 로 설정되었습니다 AND `MaxEnergyOnInvalidId`가 설정되었고 초과되지 않았습니다.	EV에 대한 에너지 전달은 `MaxEnergyOnInvalidId`에 지정된 에너지 양에 도달할 때까지 허용되어야 합니다.	
E05.FR.04	TransactionEventRequest를 생성해야 하는 경우	충전기는 Sequence Number Generation에 지정된 대로 메시지의 **seqNo** 필드를 설정해야 합니다.	이렇게 하면 CSMS가 트랜잭션 정보의 완전성을 추적할 수 있습니다.
E05.FR.05	TransactionEventRequest(eventType = Started)에서 미터 데이터를 보내도록 구성된 경우 다음을 참조하세요. Meter Values - Configuration AND EVSE는 트랜잭션 시작 시 알려져 있습니다.	충전기는 구성된 측정값을 CSMS로 전송하여 트랜잭션 중에 더 자세한 정보를 제공하기 위해 TransactionEventRequest(eventType = Started)에서 *context* = `Transaction.Begin` 인 선택적 meterValue 필드에 추가해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
E05.FR.06	`SampledDataSignReadings`는 *true* 입니다.	충전기는 서명된 미터 값을 검색하여 sampledValues의 *signedMeterValue* 필드에 넣어야 합니다.	
E05.FR.08	TransactionEventRequest(eventType = Started)에서 미터 데이터를 보내도록 구성된 경우 다음을 참조하세요. Meter Values - Configuration AND EVSE는 트랜잭션 시작 시 알 수 없음	충전기는 충전이 시작될 때 발생하는 TransactionEventRequest(eventType = Updated)에서 *context* = `Transaction.Begin` 인 선택적 meterValue 필드에 *eventType* = `Started` 에 대한 측정량을 추가해야 합니다.	
E05.FR.09	E05.FR.01 AND TransactionEventResponse의 인증 상태가 *Accepted* 가 아닙니다 AND 트랜잭션이 아직 진행 중입니다 AND `StopTxOnInvalidId`가 *true* 입니다 AND `TxStopPoint`에는 다음이 포함되지 않습니다. (Authorized 또는 PowerPathClosed 또는 EnergyTransfer)	충전기는 에너지 전송을 중지하고 TransactionEventRequest(eventType = Updated) *triggerReason* 을 *Deauthorized* 로 설정하고 동일 또는 다음 TransactionEventRequest 보고서에서 *chargingState* 를 바람직하게는 *EVConnected* 또는 대안적으로 *SuspendedEVSE* 로 설정합니다.	충전기에서 충전 상태의 물리적 변경이 트리거 *Deauthorized* 보다 몇 초 또는 밀리초 늦게 발생하는 경우 *chargingState* 변경은 *triggerReason* = *ChargingStateChanged* 로 별도로 보고될 수 있습니다. 이 상황에서 *EVConnected* 가 뒤따르지 않는 충전 상태 *SuspendedEVSE* 를 사용하는 것은 다음 OCPP 릴리스에서 더 이상 사용되지 않습니다.
E05.FR.10	E05.FR.01 AND TransactionEventResponse의 인증 상태가 *Accepted* 가 아닙니다 AND 트랜잭션이 아직 진행 중입니다 AND `StopTxOnInvalidId`가 *true* 입니다 AND `TxStopPoint`에는 다음이 포함됩니다. (Authorized 또는 PowerPathClosed 또는 EnergyTransfer)	충전기는 트랜잭션을 중지하고 TransactionEventRequest (eventType = Ended)를 다음과 함께 전송해야 합니다. *triggerReason* 을 *Deauthorized* 로 설정하고 stoppedReason을 *DeAuthorized* 로 설정합니다.	
E05.FR.11	E05.FR.10 AND 충전기에서 충전 케이블을 잠글 수 있는 경우	충전기는 소유자가 식별자를 제시할 때까지 충전 케이블을 잠근 상태로 유지해야 합니다.	

E06 - Stop Transaction options

Table 106. E06 - 트랜잭션 중지

번호	유형	설명
1	**이름**	트랜잭션 중지 옵션
2	**ID**	E06
	기능 블록	E. 트랜잭션
3	**목표**	충전기에서 트랜잭션이 중지되었음을 CSMS에 알립니다.
4	**설명**	이 사용 사례는 충전기의 구성에 따라 충전기가 트랜잭션을 중지할 수 있는 다양한 순간(TransactionEventRequest (eventType = Ended) 전송)을 설명합니다.
5	*Actors*	충전기, CSMS, EV 운전자
S1	*시나리오 목표*	주차장 점유 감지기가 더 이상 EV를 감지하지 못할 때 트랜잭션을 중지합니다.
	시나리오 설명	**1.** 충전기 주차장 점유 감지기가 EV 감지를 멈춥니다. **2.** 충전기는 TransactionEventRequest(eventType = Ended)를 보내 CSMS에 종료된 트랜잭션에 대해 알립니다. **3.** CSMS는 TransactionEventResponse로 응답하여 TransactionEventRequest가 수신되었음을 확인합니다.
	필수 조건	트랜잭션이 진행 중입니다. 구성 변수: `TxStopPoint`에는 다음이 포함됩니다. ParkingBayOccupancy
	사후 조건	**성공한 사후 조건:** 트랜잭션이 종료되고 CSMS에 *성공적으로* 알립니다. **실패 사후 조건:** 트랜잭션이 아직 진행 중입니다. *또는* CSMS에 정보가 *제공되지 않았습니다*.

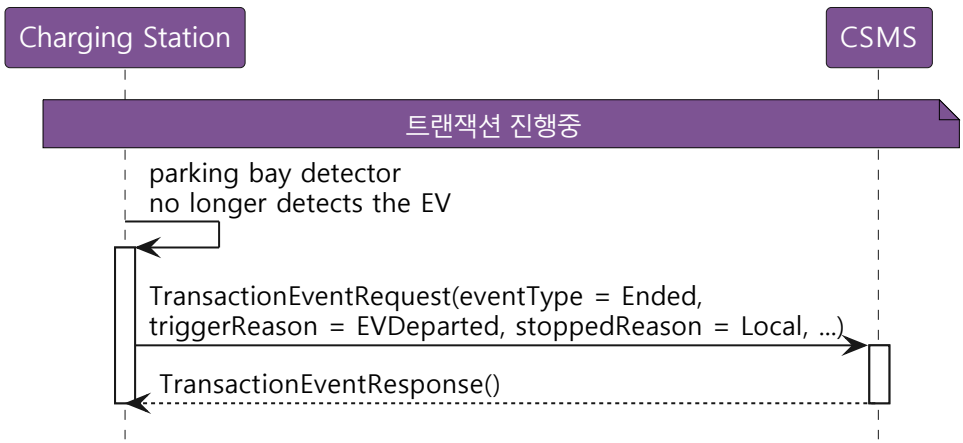

Figure 49. Sequence Diagram: Stop Transaction options - ParkingBayOccupancy

S2	*시나리오 목표*	**충전기와 EV 간 통신이 끊어졌을 때 트랜잭션을 중단합니다. (예: 케이블이 뽑힘)**
	시나리오 설명	**1.** 충전기와 EV 간 통신이 끊어졌습니다(충전 케이블이 뽑힘). **2.** 충전기 측에서 충전 케이블이 뽑힌 경우: StatusNotificationRequest를 CSMS로 보내 *Available*이 된 커넥터에 대해 알립니다. **3.** 충전기는 TransactionEventRequest(eventType = Ended)를 보내 CSMS에 종료된 트랜잭션을 알립니다. **4.** CSMS는 TransactionEventResponse로 응답하여 TransactionEventRequest가 수신되었음을 확인합니다.
	필수 조건	트랜잭션이 진행 중입니다. 구성 변수: `TxStopPoint`에는 다음이 포함됩니다. EVConnected
	사후 조건	**성공한 사후 조건:** 트랜잭션이 종료되고 CSMS에 *성공적으로* 알립니다. **실패한 사후 조건:** 트랜잭션이 여전히 진행 중입니다. *또는* CSMS에 *알리지 않습니다.*

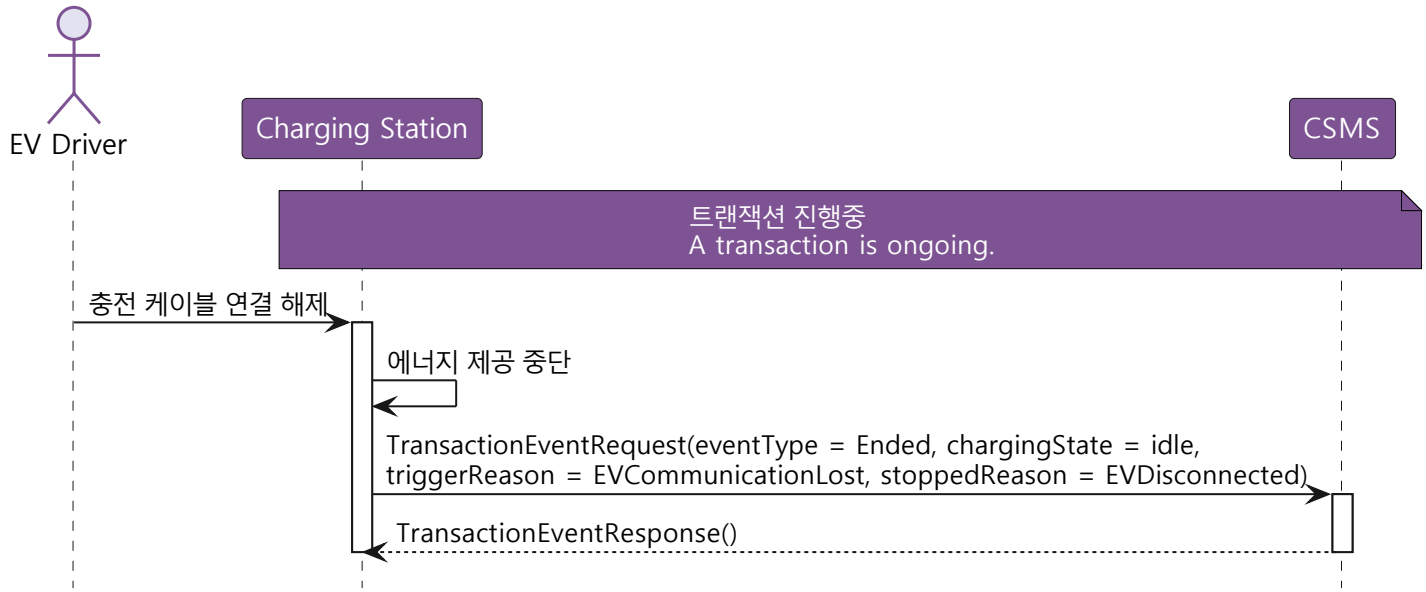

Figure 50. 시퀀스 다이어그램: 트랜잭션 중지 옵션 - EVConnected

S3	*시나리오 목표*	**운전자가 더 이상 권한이 없을 때 트랜잭션을 중지합니다.**
	시나리오 설명	**1.** 충전기에서 TransactionEventRequest를 CSMS로 보냅니다. **2.** TransactionEventResponse에서 잘못된 IdToken이 수신되었습니다. **3.** 충전기는 TransactionEventRequest(eventType = Ended)를 보내 CSMS에 종료된 트랜잭션에 대해 알립니다. **4.** CSMS는 TransactionEventResponse로 응답하여 TransactionEventRequest가 수신되었음을 확인합니다.
	필수 조건	트랜잭션이 진행 중입니다. 구성 변수: `TxStopPoint`에는 다음이 포함됩니다. Authorized
	사후 조건	**성공한 사후 조건:** 트랜잭션이 종료되고 CSMS에 *성공적으로* 알립니다. **실패 사후 조건:** 트랜잭션이 아직 진행 중입니다. *또는* CSMS에 정보가 *제공되지 않았습니다.*

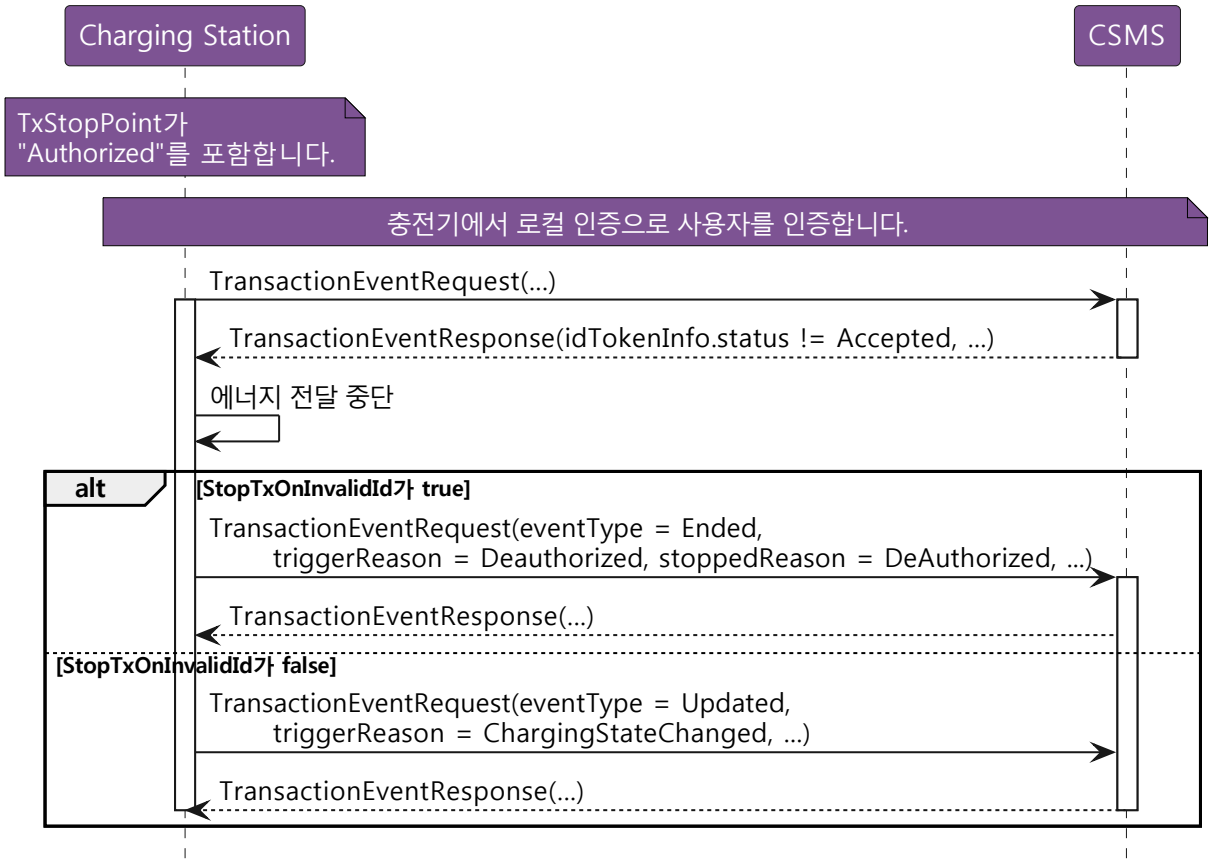

Figure 51. Sequence Diagram: Stop Transaction options - Deauthorized

S5	*시나리오 목표*	**EV 운전자가 더 이상 인증이 유효하지 않거나 EV가 연결 해제되면 트랜잭션을 중지합니다.**
	시나리오 설명	**1.** 충전기가 EV에서 연결 해제되거나 EV 운전자의 인증이 더 이상 유효하지 않습니다. **2.** 충전기가 TransactionEventRequest(eventType = Ended)를 보내 CSMS에 종료된 트랜잭션을 알립니다. **3.** CSMS가 TransactionEventResponse로 응답하여 TransactionEventRequest가 수신되었음을 확인합니다.
	필수 조건	트랜잭션이 진행 중입니다. 구성 변수: `TxStopPoint`에는 다음이 포함됩니다. PowerPathClosed
	사후 조건	**성공한 사후 조건:** 트랜잭션이 종료되고 CSMS에 *성공적으로* 알립니다. **실패한 사후 조건:** 트랜잭션이 아직 진행 중입니다. *또는* CSMS에 *알리지 않습니다.*

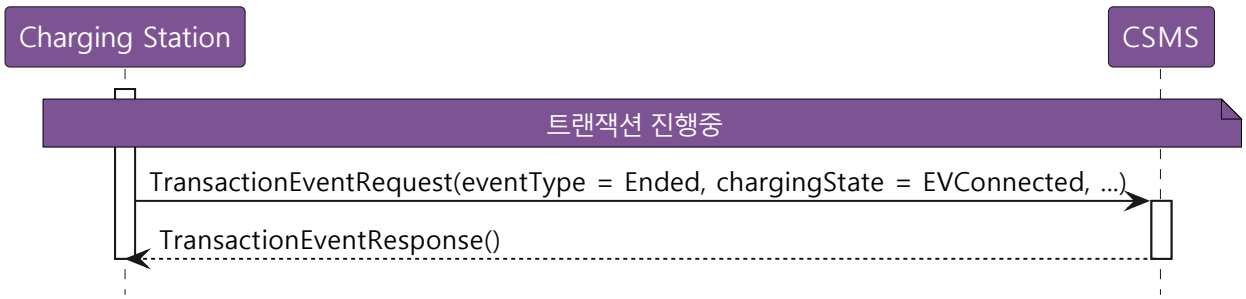

Figure 52. 시퀀스 다이어그램: 트랜잭션 중지 옵션 - PowerPathClosed

S6	*시나리오 목표*	**에너지 전송이 중지되면 트랜잭션을 중지합니다. 이는 EV가 에너지를 받지 않을 때(예: 배터리가 너무 뜨거울 때) 트랜잭션이 중지됨을 의미합니다.**
	시나리오 설명	**1.** EV와 충전기 간의 에너지 전송이 중지됩니다(예: EV 충전이 중지됨). **2.** 충전기는 TransactionEventRequest(eventType = Ended)를 보내 CSMS에 종료된 트랜잭션에 대해 알립니다. **3.** CSMS는 TransactionEventResponse로 응답하여 TransactionEventRequest가 수신되었음을 확인합니다.
	필수 조건	트랜잭션이 진행 중입니다. 구성 변수: `TxStopPoint`에는 다음이 포함됩니다. EnergyTransfer
	사후 조건	**성공한 사후 조건:** 트랜잭션이 종료되고 CSMS에 *성공적으로* 알립니다. **실패 사후 조건:** 트랜잭션이 아직 진행 중입니다. *또는* CSMS에 정보가 *제공되지 않았습니다.*

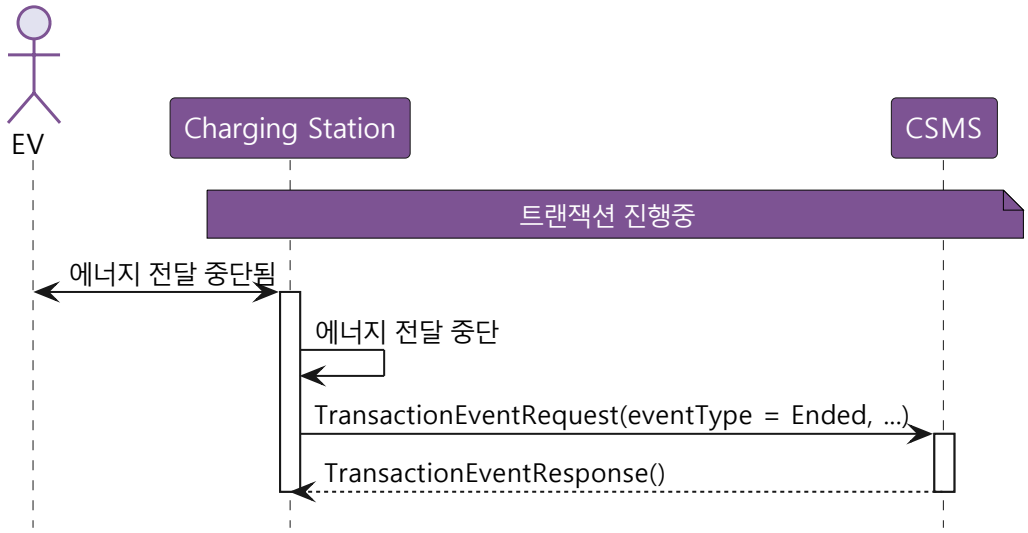

Figure 53. 시퀀스 다이어그램: 트랜잭션 중지 옵션 - EnergyTransfer

S7	*시나리오 목표*	**EV 운전자가 인증을 종료하면 트랜잭션 중지**
	시나리오 설명	**1.** EV 운전자가 충전을 종료하기 위해 IdToken을 제시합니다. **2.** 충전기는 TransactionEventRequest (eventType = Ended)를 보내 CSMS에 종료된 트랜잭션에 대해 알립니다. **3.** CSMS는 TransactionEventResponse로 응답하여 TransactionEventRequest가 수신되었음을 확인합니다.
	필수 조건	트랜잭션이 진행 중입니다. 구성 변수: `TxStopPoint`에는 Authorized (또는 PowerPathClosed)가 포함됩니다.
	사후 조건	**성공적인 사후 조건:** 트랜잭션이 종료되고 CSMS에 *성공적으로* 통보됩니다. **실패 사후 조건:** 트랜잭션이 아직 진행 중입니다. *또는* CSMS에 *통보되지* 않습니다.

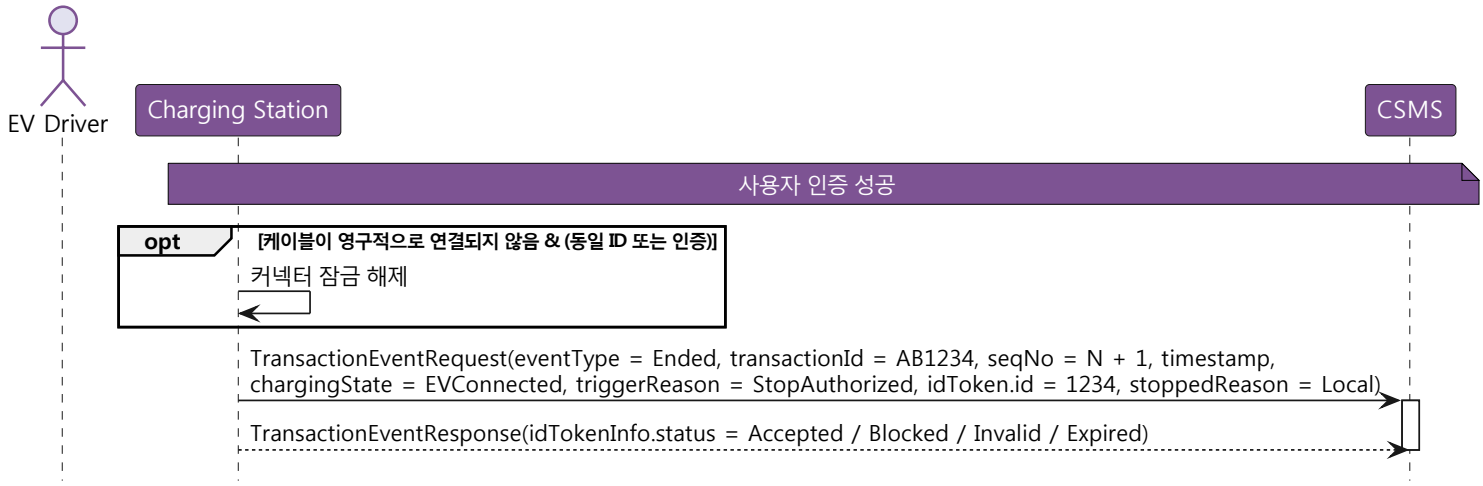

Figure 54. Sequence Diagram: Stop Transaction options - Authorized

7	**오류 처리**	n/a
8	**참고**	n/a

E06 - Stop Transaction options - Requirements

Table 107. E01 - 요구 사항

ID	전제 조건	요구 사항 정의
E06.FR.01	TxStopPoint에는 다음이 포함됩니다. ParkingBayOccupancy AND 주차 베이 감지기가 더 이상 "EV"를 감지하지 못함	충전기는 트랜잭션을 중지하고 TransactionEventRequest를 보내야 합니다. (eventType = Ended)를 CSMS로 전송합니다.
E06.FR.02	TxStopPoint에는 다음이 포함됩니다. EVConnected AND 충전기와 EV 간의 연결이 끊어졌습니다.	충전기는 트랜잭션을 중지하고 TransactionEventRequest (eventType = Ended)를 CSMS로 전송해야 합니다.
E06.FR.03	TxStopPoint에는 다음이 포함됩니다. Authorized AND EV 운전자는 트랜잭션을 중지할 권한이 있습니다.	충전기는 트랜잭션을 중지하고 TransactionEventRequest (eventType = Ended)를 CSMS로 전송해야 합니다.
E06.FR.04	TxStopPoint에는 다음이 포함됩니다. Authorized AND CSMS는 TransactionEventResponse에서 유효하지 않은 idTokenInfo를 반환합니다.	충전기는 트랜잭션을 중지하고 TransactionEventRequest (eventType = Ended)를 CSMS로 전송해야 합니다.
E06.FR.05	TxStopPoint에는 다음이 포함됩니다. DataSigned AND 충전기는 더 이상 서명된 미터 값을 얻을 수 없습니다.	충전기는 트랜잭션을 중지하고 TransactionEventRequest (eventType = Ended)를 CSMS로 보내야 합니다.
E06.FR.06	TxStopPoint에는 다음이 포함됩니다. PowerPathClosed AND (충전기와 EV 간의 연결이 끊어짐 OR 인증이 종료되었거나 idToken이 인증 해제됨)	충전기는 트랜잭션을 중지하고 TransactionEventRequest (eventType = Ended)를 CSMS로 보내야 합니다.

ID	전제 조건	요구 사항 정의
E06.FR.07	`TxStopPoint`에는 다음이 포함됩니다. EnergyTransfer AND 에너지 전송이 중지됨	충전기는 트랜잭션을 중지하고 TransactionEventRequest(eventType = Ended)를 CSMS로 전송해야 합니다.
E06.FR.08	충전기에서 EV 운전자가 트랜잭션을 종료하지 않은 경우	충전기는 TransactionEventRequest(eventType = Ended)에 stoppedReason 요소를 포함해야 합니다. 사용할 이유는 reasonEnumType의 설명에 설명되어 있습니다.
E06.FR.09	충전기에서 EV 운전자가 트랜잭션을 종료한 경우(예: EV 운전자가 트랜잭션을 중단하기 위해 IdToken을 제시한 경우)	충전기는 TransactionEventRequest(eventType = Ended)에서 stoppedReason 요소를 생략할 수 있습니다(따라서 CSMS는 생략 시 이유를 local로 해석할 수 있음).
E06.FR.10	정상적인 트랜잭션 종료의 일부로.	충전기는 케이블을 잠금 해제해야 합니다(영구적으로 연결되지 않은 경우).
E06.FR.11	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Ended), 다음을 참조하세요: Meter Values - Configuration	충전기는 구성된 측정값을 선택적 *meterValue* 필드에 *context* = `Transaction.End` 로 추가하여 TransactionEventRequest (eventType = Ended)에서 CSMS로 전송하여 트랜잭션 사용에 대한 자세한 정보를 제공해야 합니다.
E06.FR.12	E06.FR.11 AND 충전기의 메모리가 부족합니다.	충전기는 TransactionEventRequest(eventType = Ended) 메시지에서 미터 데이터를 삭제할 수 있습니다.
E06.FR.13	E06.FR.12	미터 데이터를 삭제할 때 충전기는 중간 값을 먼저 삭제해야 하며(첫 번째 값, 세 번째 값, 다섯 번째 등), 목록의 시작 부분에서 값을 삭제하거나 목록에 값을 추가하는 것을 중단해서는 안 됩니다.
E06.FR.14	TransactionEventRequest를 만들어야 하는 경우	충전기는 Sequence Number Generation에 지정된 대로 메시지의 **seqNo** 필드를 설정해야 합니다.
E06.FR.15	TransactionEventRequest를 보낼 때	충전기는 triggerReason을 설정하여 CSMS에 이벤트를 트리거한 원인을 알려야 합니다. 사용할 이유는 TriggerReasonEnumType의 설명에 설명되어 있습니다.
E06.FR.16	트랜잭션이 비정상 오류 또는 오류 조건으로 인해 중단되었습니다.	충전기는 TransactionEventRequest(eventType = Ended, triggerReason=AbnormalCondition)를 CSMS로 보내야 합니다.

E07 - Transaction locally stopped by IdToken

Table 108. E07 - IdToken에 의해 로컬에서 중단된 트랜잭션

번호	유형	설명
1	**이름**	IdToken에 의해 로컬에서 중단된 트랜잭션
2	**ID**	E07
	기능 블록	E. 트랜잭션
3	**목표**	EV 운전자는 로컬에서 자신의 IdToken을 제시하여 진행 중인 트랜잭션을 중단하려고 합니다.
4	**설명**	이 사용 사례는 EV 운전자가 충전기를 떠나려고 할 때 트랜잭션을 중단하는 방법을 다룹니다.
	Actors	충전기, CSMS, EV 운전자
	시나리오 설명 TxStopPoint = `Authorized` (또는 `PowerPathClosed`)	**1.** EV 운전자가 충전을 종료하기 위해 IdToken을 두 번째로 제시합니다. **2.** 충전기가 에너지 전송을 중지하고 케이블이 영구적으로 연결되지 않은 경우 충전기가 케이블의 잠금을 해제합니다. **3.** 충전기가 *triggerReason* = `StopAuthorized` 및 *stoppedReason* = `Local` 인 TransactionEventRequest(eventType = Ended)를 보냅니다. **4.** CSMS가 TransactionEventResponse로 응답합니다.
	대체 시나리오 TxStopPoint = `Authorized` (또는 `PowerPathClosed`)	*충전을 중지할 때 트랜잭션이 triggerReason=ChargingStateChanged로 종료됨:* **1.** EV 운전자가 충전을 종료하기 위해 두 번째로 IdToken을 제시합니다. **2.** 충전기가 TransactionEventRequest(eventType = Updated)를 *triggerReason* = `StopAuthorized` 로 전송합니다. **3.** CSMS가 TransactionEventResponse로 응답합니다. **4.** 충전기가 에너지 전송을 중지하고 케이블이 영구적으로 연결되지 않은 경우 충전기가 케이블의 잠금을 해제합니다. **5.** 충전기는 TransactionEventRequest (eventType = Ended)를 *triggerReason* = `ChargingStateChanged`, *transactionInfo.chargingState* = `EVConnected` 로 전송합니다. **6.** CSMS는 TransactionEventResponse로 응답합니다.
5	**필수 조건**	트랜잭션이 진행 중입니다.
6	**사후 조건**	**성공 사후 조건:** CSMS는 트랜잭션에 대한 모든 관련 정보를 수신했으며 충전기는 *Idle* 상태입니다. **실패 사후 조건:** 트랜잭션이 아직 진행 중이거나 충전기가 Idle 상태이며 CSMS에 전달해야 하는 트랜잭션에 대한 정보를 여전히 보유하고 있습니다.

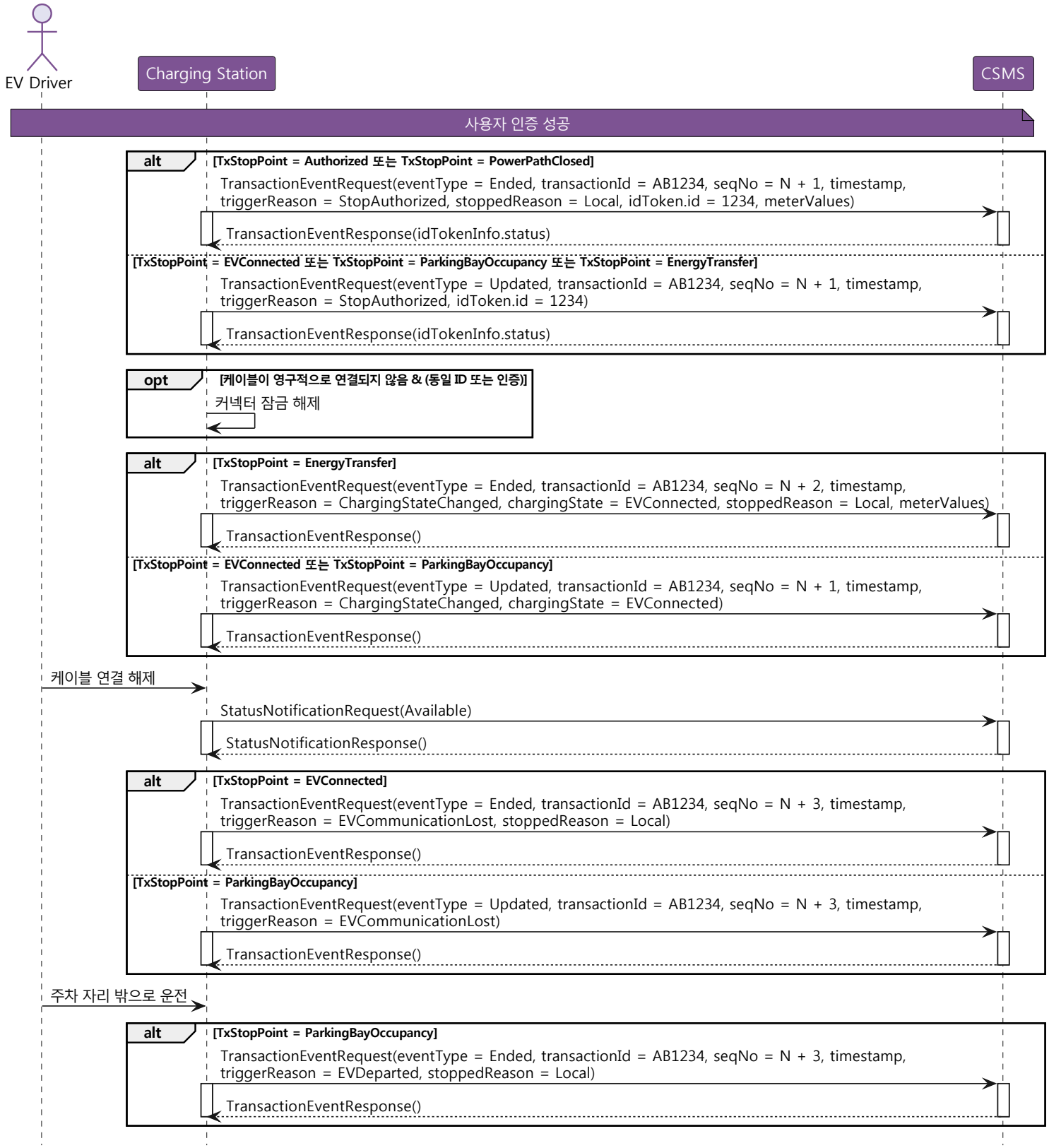

Figure 55. 시퀀스 다이어그램: 트랜잭션이 IdToken에 의해 로컬로 중단되었고 TransactionEventRequest는 TxStopPoint 구성에 의해 엄격하게 보고됨

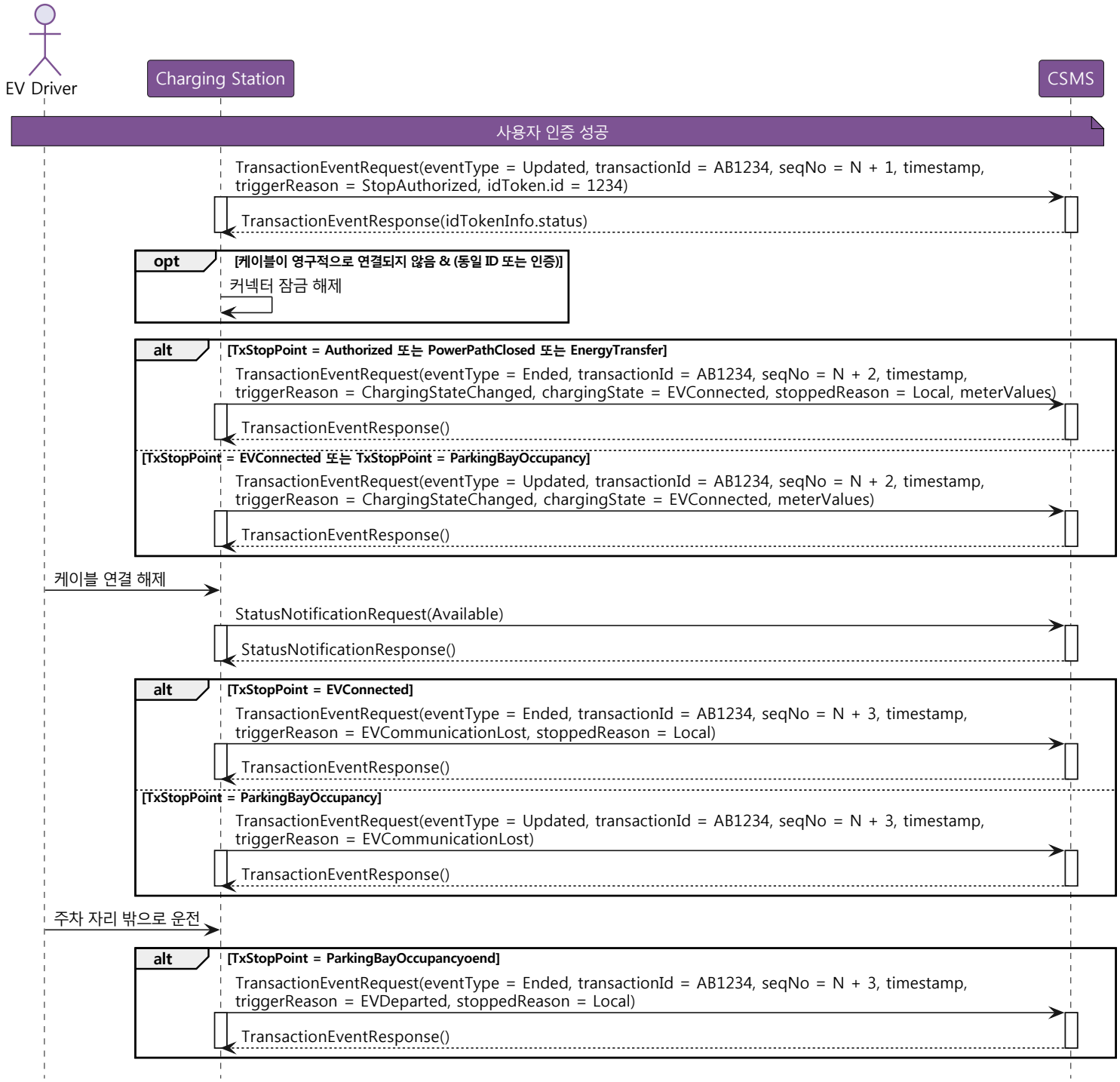

Figure 56. 시퀀스 다이어그램: 트랜잭션이 IdToken에 의해 로컬로 중단됨, 지연된 TransactionEventRequest eventType = Ended for TxStopPoint = Authorized 또는 PowerPathClosed

7	**오류 처리**	n/a
8	**참고**	시나리오 설명은 Authorized 또는 PowerPathClosed를 포함하는 TxStopPoint를 기반으로 합니다. 시퀀스 다이어그램은 또한 alt-블록의 다른 TxStopPoint 값에 대한 동작을 보여줍니다. CSMS는 트랜잭션이 중지되는 것을 막을 수 없습니다.

E07 - Transaction locally stopped by IdToken - Requirements

Table 109. E07 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
E07.FR.01	인증된 트랜잭션 중에 idToken이 제시될 때 AND (a) 제시된 idToken이 인증을 시작한 idToken과 동일할 때 또는 (b) 제시된 idToken이 로컬 인증 목록 또는 인증 캐시에 있고 인증이 유효하고 인증을 시작한 IdToken과 동일한 GroupIdToken을 가질 때	충전기는 AuthorizeRequest를 먼저 보내지 않고 트랜잭션 인증을 종료해야 합니다.	인증을 시작한 idToken은 항상 인증을 종료하는 데 사용할 수 있습니다. 인증을 종료하면 에너지 공급이 종료됩니다. TxStopPoint에 따라 인증을 종료하면 트랜잭션도 종료될 수 있습니다. (C01.FR.03 참조)
E07.FR.02	E07.FR.01	충전기는 *triggerReason* = `StopAuthorized` 인 TransactionEventRequest를 보내야 하며, 인증을 중지하는 데 사용된 *idToken* 을 포함해야 합니다.	중지 *idToken* 은 동일한 GroupId를 공유할 때 시작 *idToken* 과 다를 수 있습니다.
E07.FR.04	충전기에서 EV 운전자의 요청에 따라 트랜잭션이 중지된 경우.	충전기는 *eventType* = `Ended` 인 최종 TransactionEventRequest에서 stoppedReason 요소를 생략할 수 있습니다.(MAY)	예: EV 운전자가 트랜잭션을 중지하기 위해 IdToken을 제시하거나 (비상 정지가 아닌) "중지" 버튼을 눌렀습니다. 원격으로 중지하는 경우 사용 사례 F03을 참조하세요.
E07.FR.05	충전기에서 EV 운전자의 요청에 따라 트랜잭션이 중단된 경우.	충전기는 *eventType* = `Ended` 인 최종 TransactionEventRequest에서 stoppedReason = `Local` 을 사용해야 합니다.(SHOULD)	예: EV 운전자가 트랜잭션을 중단하기 위해 IdToken을 제시하거나 (비상 정지가 아닌) "중지" 버튼을 눌렀습니다. 원격으로 중단하는 경우 사용 사례 F03을 참조하세요.
E07.FR.06	충전기에서 EV 운전자의 요청에 따라 중단되지 않고 트랜잭션이 중단된 경우.	충전기는 *eventType* = `Ended` 인 최종 TransactionEventRequest에서 *stoppedReason* 에 대해 ReasonEnumType에서 가장 적절한 값을 사용해야 합니다.	원격으로 중단(`Remote`), CSMS가 권한을 취소(`DeAuthorized`)하거나 EV를 분리(`EVDisconnected`)하는 것 외에, 대부분의 다른 이유는 기술적 오류나 에너지 제한과 관련이 있습니다.
E07.FR.07	정상적인 트랜잭션 종료의 일부로.	충전기는 케이블(영구적으로 연결되지 않은 경우)을 잠금 해제해야 합니다.	
E07.FR.08	TransactionEventRequest (eventType = Ended)에서 미터 데이터를 보내도록 구성된 경우, 다음을 참조하세요: Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송하여 TransactionEventRequest(eventType = Ended)의 *context* = `Transaction.End` 인 선택적 *meterValue* 필드에 추가하여 트랜잭션 사용에 대한 자세한 정보를 제공해야 합니다.	
E07.FR.09	E07.FR.08 AND 충전기의 메모리가 부족합니다.	충전기는 TransactionEventRequest(eventType = Ended) 메시지에서 미터 데이터를 삭제할 수 있습니다.	

ID	전제 조건	요구 사항 정의	참고
E07.FR.10	E07.FR.09	미터 데이터를 삭제할 때 충전기는 먼저 중간 값(첫 번째 값, 세 번째 값, 다섯 번째 등)을 삭제해야 하며, 목록의 시작 부분에서 값을 삭제하거나 목록에 값을 추가하는 것을 중지해서는 안 됩니다.	
E07.FR.11	TransactionEventRequest를 생성해야 하는 경우	충전기는 Sequence Number Generation에 지정된 대로 메시지의 **seqNo** 필드를 설정해야 합니다.	이렇게 하면 CSMS가 트랜잭션 정보의 완전성을 추적할 수 있습니다.
E07.FR.12	`SampledDataSignReadings`는 *true* 입니다.	충전기는 서명된 미터 값을 검색하여 sampledValues의 *signedMeterValue* 필드에 넣어야 합니다.	

E08 - Transaction stopped while Charging Station is offline

Table 110. E08 - Transaction stopped while Charging Station is offline

번호	유형	설명
1	**이름**	충전기가 오프라인인 동안 트랜잭션이 중단됨
2	**ID**	E08
	기능 블록	E. 트랜잭션
	부모 사용 사례	E07 - Local Stop Transaction
3	**목표**	충전기가 *offline* 인 동안 EV 운전자가 트랜잭션을 중단할 수 있도록 합니다.
4	**설명**	이 사용 사례는 충전기가 *offline* 인 동안 EV 운전자가 트랜잭션을 중단하는 방법을 설명합니다. 트랜잭션이 진행 중이고 충전기가 *offline* 인 동안 EV 운전자가 자신의 IdToken을 제시하면 충전기가 로컬에서(CSMS에 묻지 않고) 이 IdToken이 트랜잭션을 중단할 수 있음을 알고 있으면 진행 중인 트랜잭션이 중단됩니다. 충전기가 CSMS와 연결을 복구하면 이 *offline* 중지 트랜잭션에 대한 정보를 CSMS로 보내야 합니다.
	Actors	충전기, CSMS, EV 운전자
	시나리오 설명	**1.** EV 운전자가 트랜잭션을 중지하기 위해 IdToken을 제시합니다. **2.** 이 IdToken이 트랜잭션을 시작하는 데 사용된 것과 동일하거나 로컬 인증 목록 및/또는 인증 캐시를 통해 GroupId를 검증할 수 있습니다. 트랜잭션이 중지됩니다. **3.** 충전기가 에너지 제공을 중지합니다. **4.** TransactionEventRequest (eventType = Ended)가 충전기에 저장/대기됩니다. **5.** 충전기와 CSMS 간의 연결이 복구됩니다. **6.** 충전기가 대기 중인 메시지를 보내기 시작합니다 **7.** 저장된 TransactionEventRequest가 전송되어 CSMS에 중지된 트랜잭션에 대해 알립니다.
5	**필수 조건**	트랜잭션이 진행 중이고 연결이 끊어졌습니다.
6	**사후 조건**	충전기가 *idle* 상태입니다.

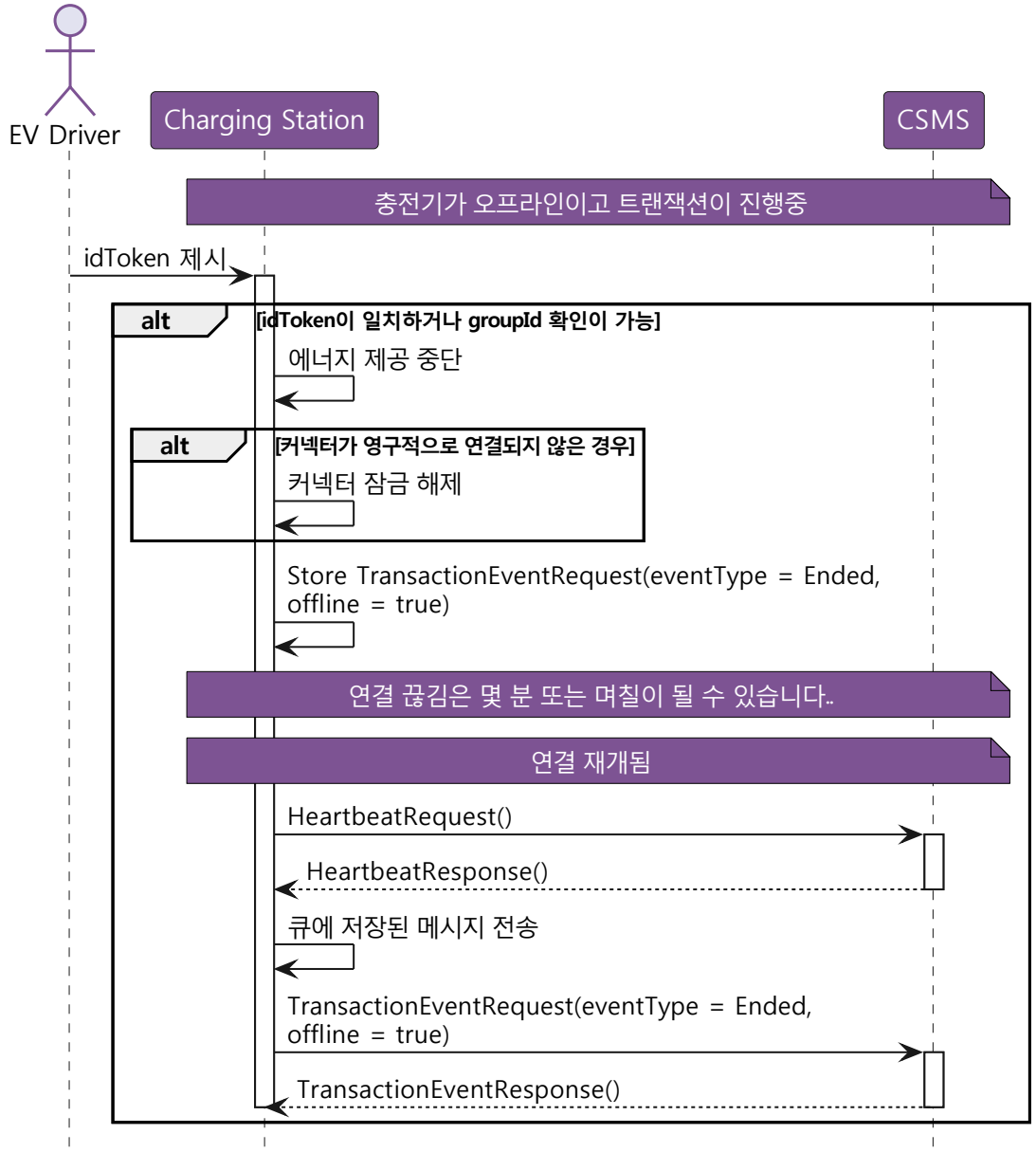

Figure 57. 시퀀스 다이어그램: 충전기가 오프라인인 동안 트랜잭션이 중지되었습니다

7	**오류 처리**	n/a
8	**참고**	groupId 검사가 가능할경우 로컬 인증 목록 및/또는 인증 캐시에서 이를 수행해야 합니다. 위의 시나리오 설명과 시퀀스 다이어그램은 중지 트랜잭션을 위한 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStopPoint`: ParkingBayOccupancy, EVConnected, Authorized 이 유즈케이스는 다른 구성에서도 유효하지만, 그 경우 트랜잭션이 다른 시점에서 시작/종료될 수 있으며, 이는 메시지가 전송되는 순서에 영향을 미칠 수 있습니다. 자세한 내용은 사용 사례를 참조하세요: E06 - Stop Transaction options

E08 - Transaction stopped while Charging Station is offline - Requirements

Table 111. E08 - Requirements

ID	전제 조건	요구 사항 정의	참고
E08.FR.01	제시된 IdToken이 트랜잭션을 시작하는 데 사용된 IdToken과 동일한 경우.	충전기는 에너지 제공을 중단해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
E08.FR.02	제시된 IdToken이 트랜잭션을 시작하는 데 사용된 IdToken과 동일한 GroupId를 갖는 경우.	충전기는 에너지 제공을 중단해야 합니다.	
E08.FR.03	(E08.FR.01 또는 E08.FR.02) AND 케이블이 영구적으로 연결되지 않음	충전기는 커넥터를 잠금 해제해야 합니다.	
E08.FR.04	(E08.FR.01 또는 E08.FR.02)	충전기는 TransactionEventRequest(eventType = Ended)를 "생성"해야 합니다.	
E08.FR.05	*offline* 일 때.	충전기는 TransactionEventRequest 메시지를 큐에 넣어야 합니다.	
E08.FR.06	연결이 복구된 후.	충전기는 큐에 넣은 TransactionEventRequest 메시지를 보내야 합니다.	
E08.FR.07		플래그: *offline* 은 충전기가 오프라인 상태일 때 발생한 모든 TransactionEventRequest에 대해 TRUE로 설정되어야 합니다.	
E08.FR.08	TransactionEventRequest를 생성해야 하는 경우	충전기는 Sequence Number Generation에 지정된 대로 메시지의 **seqNo** 필드를 설정해야 합니다.	이렇게 하면 CSMS가 트랜잭션 정보의 완전성을 추적할 수 있습니다.
E08.FR.09	TransactionEventRequest(eventType = Ended)에서 미터 데이터를 보내도록 구성된 경우 다음을 참조하세요. Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Ended)의 선택적 meterValue 필드에 추가하여 트랜잭션 사용에 대한 자세한 정보를 제공해야 합니다.	
E08.FR.10	E08.FR.09 그리고 충전기의 메모리가 부족합니다.	충전기는 TransactionEventRequest(eventType = Ended) 메시지에서 미터 데이터를 삭제할 수 있습니다.	
E08.FR.11	E08.FR.10	미터 데이터를 삭제할 때 충전기는 중간 값을 먼저 삭제해야 합니다(첫 번째 값, 세 번째 값, 다섯 번째 등). 목록의 시작 부분에서 값을 삭제하거나 목록에 값을 추가하는 것을 중단해서는 안 됩니다.	
E08.FR.12	`SampledDataSignReadings`는 *true* 입니다.	충전기는 서명된 미터 값을 검색하여 sampledValues의 *signedMeterValue* 필드에 넣어야 합니다.	

E09 - When cable disconnected on EV-side: Stop Transaction

Table 112. E09 - EV 측에서 케이블이 분리된 경우: 트랜잭션 중지

번호	유형	설명
1	**이름**	EV 측에서 케이블이 분리된 경우: 트랜잭션 중지
2	**ID**	E09
	기능 블록	E. 트랜잭션
	부모 사용 사례	E07 - Local Stop Transaction
3	**목표**	EV 측에서 충전 케이블이 분리되면 진행 중인 트랜잭션을 중지합니다.
4	**설명**	이 사용 사례는 EV 운전자가 EV 측에서 케이블을 분리하면 트랜잭션이 중지되는 방법을 다룹니다. 이 사용 사례에서 구성 변수는 다음과 같습니다: `StopTxOnEVSideDisconnect` = true. EV 측에서 충전 케이블이 분리됩니다. 이는 충전기에서 감지합니다. 충전기는 트랜잭션을 중지하고 CSMS에 TransactionEventRequest를 보냅니다. 충전 케이블이 잠겨 있고 `UnlockOnEVSideDisconnect` = false인 경우 EV 운전자가 돌아와서 자신의 IdToken을 제시할 때까지 충전기에서 잠긴 상태로 유지됩니다. 그렇지 않으면 케이블이 잠금 해제됩니다.
	Actors	충전기, CSMS, EV 운전자
	시나리오 설명	**1.** EV에서 케이블이 분리되었습니다. **2.** 에너지 제공이 중단되었습니다. **3.** 충전기에서 TransactionEventRequest(eventType = Ended, stoppedReason = EVDisconnected)를 CSMS로 전송합니다. **4.** CSMS에서 TransactionEventResponse로 응답합니다. **5.** EV 운전자가 인증을 하고 케이블을 분리합니다. **6.** 충전기에서 StatusNotificationRequest를 CSMS로 *Available* 상태로 전송합니다. **7.** CSMS는 StatusNotificationResponse로 응답합니다.
	대체 시나리오	E10 - When cable disconnected on EV-side: Suspend Transaction
5	**필수 조건**	구성 변수: `StopTxOnEVSideDisconnect` = true 트랜잭션이 진행 중입니다.
6	**사후 조건**	**성공한 사후 조건:** 충전기가 *idle* 상태입니다. **실패 사후 조건:** n/a

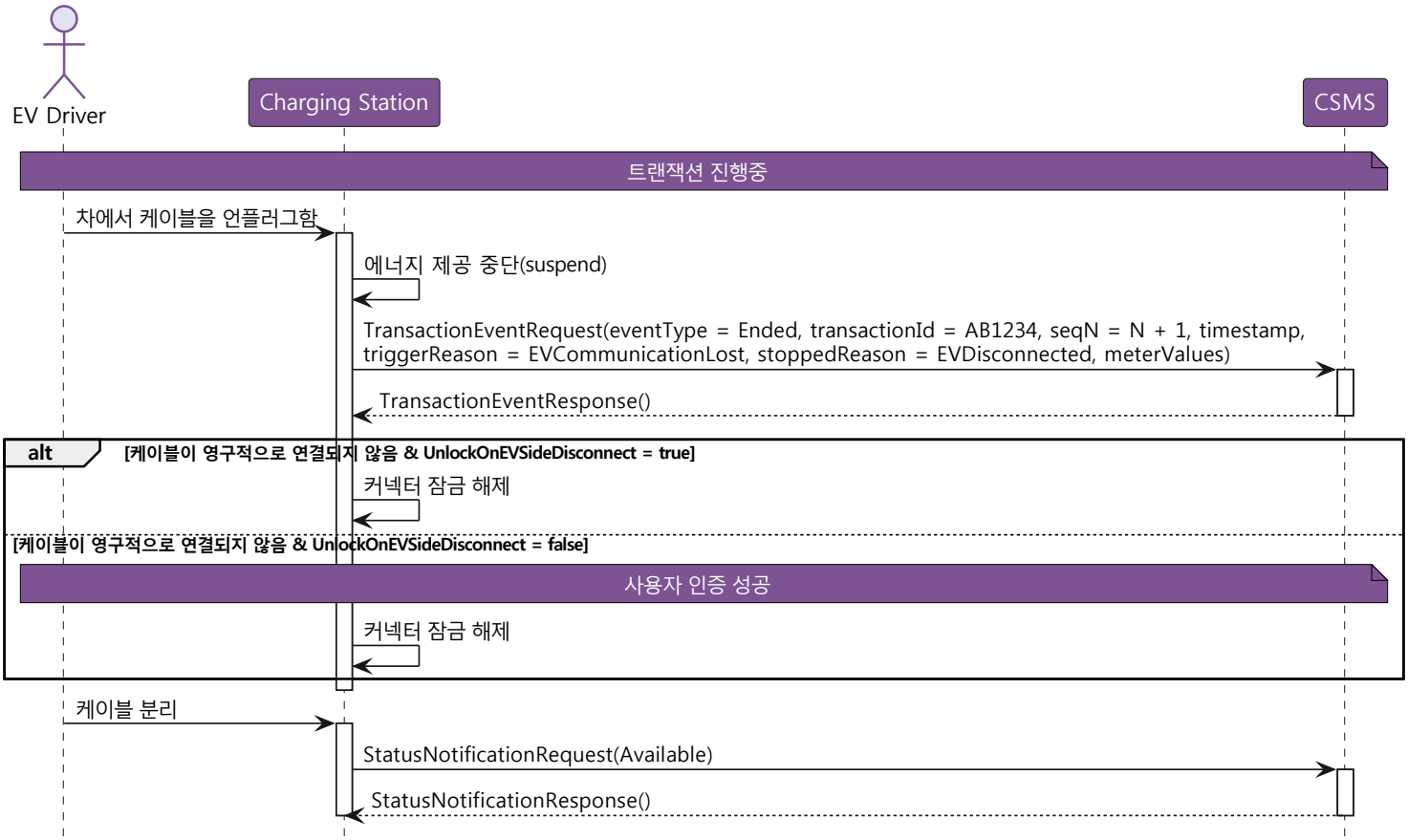

Figure 58. 시퀀스 다이어그램: EV 측에서 케이블이 분리된 경우: 트랜잭션 중지

7	**오류 처리**	n/a
8	**참고**	충전 케이블을 다시 연결하면 충전이 재개/계속되지 않습니다. 위의 시나리오 설명과 시퀀스 다이어그램은 트랜잭션 중지를 위한 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStopPoint`: Authorized 이 유즈케이스는 다른 구성에서도 유효하지만, 그 경우 트랜잭션이 다른 시점에서 시작/종료될 수 있으며, 이는 메시지가 전송되는 순서에 영향을 미칠 수 있습니다. 자세한 내용은 사용 사례를 참조하세요. E06 - Stop Transaction options

E09 - When cable disconnected on EV-side: Stop Transaction - Requirements

Table 113. E09 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
E09.FR.01	`StopTxOnEVSideDisconnect` = *true* 이면.	케이블이 EV에서 분리되면 트랜잭션이 인증 취소되어야 합니다. EV가 다시 연결된 경우 트랜잭션이 다시 인증될 때까지 에너지 전송이 허용되지 않습니다.	`StopTxOnEVSideDisconnect`를 *true* 로 설정하면 EV 측에서 잠기지 않은 케이블을 분리할 때 방해 행위를 방지할 수 있습니다.
E09.FR.02	E09.FR.01 AND 케이블이 영구적으로 연결되지 않은 경우 AND `UnlockOnEVSideDisconnect` = true.	충전기는 충전 케이블을 잠금 해제해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
E09.FR.03	E09.FR.01 AND 케이블이 영구적으로 연결되지 않은 경우 AND `UnlockOnEVSideDisconnect` = false.	충전기는 EV 운전자의 인증 후에만 충전 케이블을 잠금 해제해야 합니다.	
E09.FR.04	TransactionEventRequest를 생성해야 하는 경우	충전기는 Sequence Number Generation에 지정된 대로 메시지의 **seqNo** 필드를 설정해야 합니다.	이렇게 하면 CSMS가 트랜잭션 정보의 완전성을 추적할 수 있습니다.
E09.FR.05	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Ended), 다음을 참조하세요: Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Ended)의 선택적 meterValue 필드에 추가하여 트랜잭션 사용에 대한 자세한 정보를 제공해야 합니다.	
E09.FR.06	E09.FR.05 AND 충전기의 메모리가 부족합니다.	충전기는 TransactionEventRequest(eventType = Ended) 메시지에서 미터 데이터를 삭제할 수 있습니다.	
E09.FR.07	E09.FR.06	미터 데이터를 삭제할 때 충전기는 먼저 중간 값(첫 번째 값, 세 번째 값, 다섯 번째 등)을 삭제해야 하며, 목록의 시작 부분에서 값을 삭제하거나 목록에 값을 추가하는 것을 중지해서는 안 됩니다.	
E09.FR.08	`SampledDataSignReadings`는 *true* 입니다.	충전기는 서명된 미터 값을 검색하여 sampledValues의 *signedMeterValue* 필드에 넣어야 합니다.	

E10 - When cable disconnected on EV-side: Suspend Transaction

Table 114. E10 - EV 측에서 케이블이 분리된 경우: 트랜잭션 중단

번호	유형	설명
1	**이름**	EV 측에서 케이블이 분리된 경우: 트랜잭션 일시 중단
2	**ID**	E10
	기능 블록	E. 트랜잭션
	부모 사용 사례	E07 - Local Stop Transaction
3	**목표**	EV 측에서 충전 케이블이 분리되면 진행 중인 트랜잭션을 일시 중단합니다.
4	**설명**	이 사용 사례는 EV 운전자가 EV 측에서 케이블을 분리하면 트랜잭션이 일시 중단되는 방식을 다룹니다. 이 사용 사례에서 구성 변수는 다음과 같습니다: `StopTxOnEVSideDisconnect` = false. EV 측에서 충전 케이블이 분리되었습니다. 이는 충전기에서 감지합니다. 충전기는 에너지 제공(안전을 위해)을 중단하지만 트랜잭션은 중단하지 않습니다. 충전 케이블이 잠긴 경우 EV 운전자가 돌아와서 ID 토큰을 제시할 때까지 충전기에서 잠긴 상태로 유지됩니다.
	Actors	충전기, CSMS, EV 운전자
	시나리오 설명	**1.** EV 드라이버가 트랜잭션이 진행되는 동안 EV에서 케이블을 분리합니다. **2.** 에너지 제공이 중단됩니다. ***EV 드라이버가 케이블을 다시 연결하면 트랜잭션이 재개됩니다.*** **A1.** 충전기에서 TransactionEventRequest (eventType = Updated, trigger = CablePluggedIn)를 보냅니다. **A2.** CSMS에서 TransactionEventResponse로 응답합니다. ***케이블이 영구적으로 연결되지 않은 경우*** **B1.** EV 드라이버는 충전기 및/또는 CSMS에서 충전 케이블 잠금을 해제하도록 인증되었습니다. **B2.** 케이블이 잠금 해제되었습니다. **B3.** 충전기에서 TransactionEventRequest(eventType = Ended, trigger = StopAuthorized)를 전송합니다. **B4.** EV 운전자가 충전 케이블을 제거합니다. **B5.** 충전기에서 StatusNotificationRequest를 CSMS에 전송하고 상태는 *Available*입니다. **B6.** CSMS에서 StatusNotificationResponse로 응답합니다. ***케이블이 영구적으로 연결된 경우*** **C1.** 케이블이 시간 초과 내에 연결되지 않았습니다. **C2.** 충전기는 TransactionEventRequest(eventType = Ended, trigger = EVCommunicationLost, stoppedReason = EVDisconnected)를 전송합니다. **C3.** 충전기는 StatusNotificationRequest를 CSMS에 상태 *Available*로 전송합니다. **C4.** CSMS는 StatusNotificationResponse로 응답합니다.
	대체 시나리오	E09 - When cable disconnected on EV-side: Stop Transaction
5	**필수 조건**	구성 변수: `StopTxOnEVSideDisconnect` = false 트랜잭션이 진행 중입니다.
6	**사후 조건**	**성공적인 사후 조건:** 충전기가 *idle* 상태입니다. 일반 트랜잭션이 재개됩니다. **실패 사후 조건:** n/a

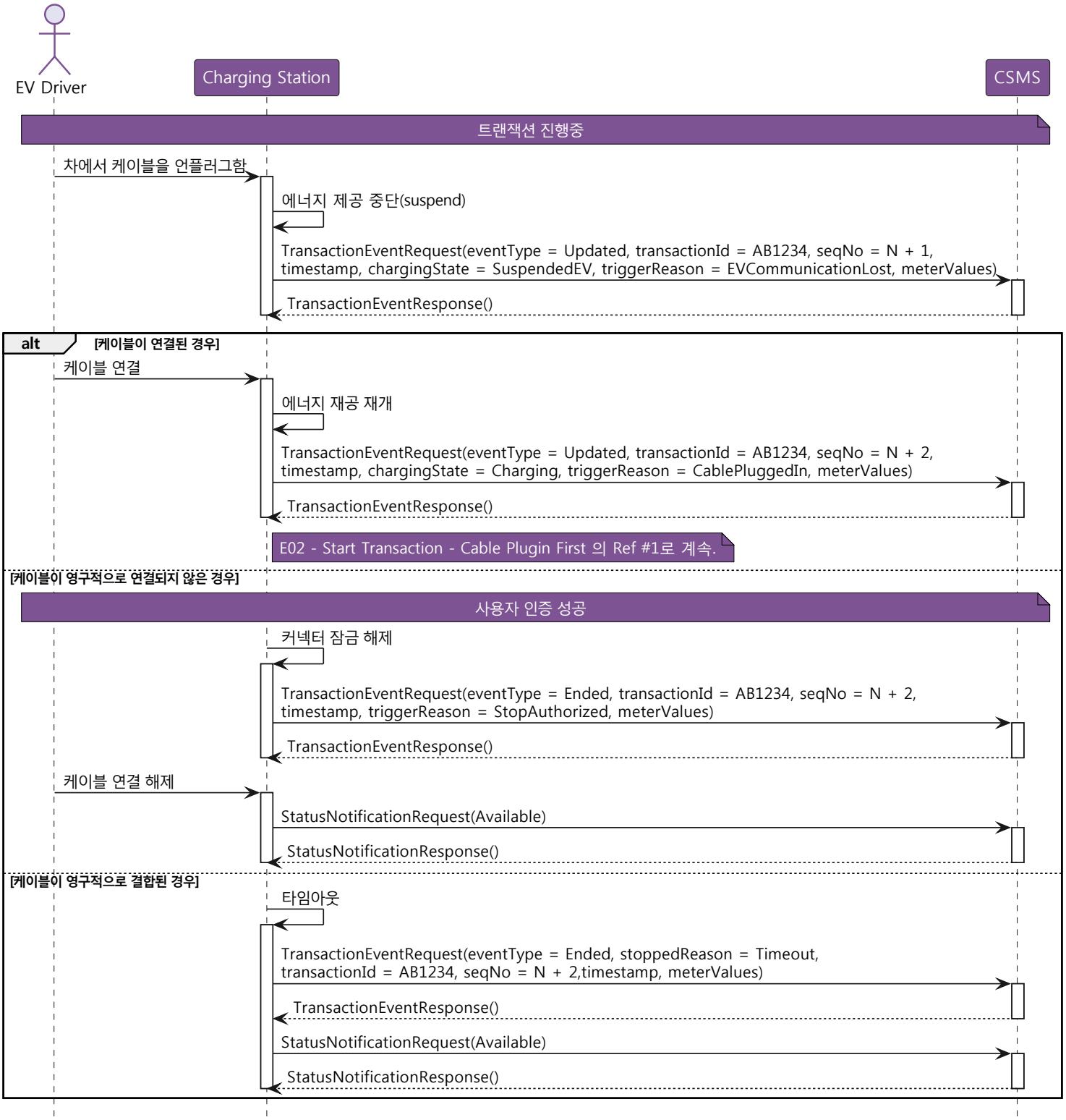

Figure 59. 시퀀스 다이어그램: EV 측에서 케이블이 분리된 경우: 트랜잭션 중단

7	**오류 처리**	n/a

8	참고 사항	충전 케이블을 다시 연결하면 충전이 재개됩니다. 케이블이 영구적으로 연결되어 있고 특정 시간 초과 내에 케이블이 연결되지 않으면 충전기가 트랜잭션을 중단합니다. 이 시간 초과는 OCPP에서 정의하지 않으며 충전기 구현자에게 맡겨집니다. 위의 시나리오 설명과 시퀀스 다이어그램은 트랜잭션 중단을 위한 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStopPoint`: ParkingBayOccupancy, Authorized 이 유즈케이스는 다른 구성에서도 유효하지만, 그 경우 트랜잭션이 다른 시점에서 시작/종료될 수 있으며, 이는 메시지가 전송되는 순서에 영향을 미칠 수 있습니다. 자세한 내용은 사용 사례를 참조하세요: E06 - Stop Transaction options

E10 - When cable disconnected on EV-side: Suspend Transaction - Requirements

Table 115. E10 - Requirements

ID	전제 조건	요구 사항 정의	참고
E10.FR.01	케이블이 영구적으로 연결되지 않음	커넥터는 EV 운전자가 IdToken을 제시할 때까지 충전기에서 잠긴 상태로 유지되어야 합니다.	
E10.FR.02	케이블이 영구적으로 연결됨 AND 케이블이 시간 초과 내에 연결되지 않음	충전기는 트랜잭션을 인증 취소해야 합니다.	
E10.FR.03	TransactionEventRequest를 만들어야 하는 경우	충전기는 Sequence Number Generation에 지정된 대로 메시지의 **seqNo** 필드를 설정해야 합니다.	이렇게 하면 CSMS가 트랜잭션 정보의 완전성을 추적할 수 있습니다.
E10.FR.04	TransactionEventRequest(eventType = Ended)에서 미터 데이터를 보내도록 구성된 경우 다음을 참조하세요. Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Ended)의 선택적 meterValue 필드에 추가하여 트랜잭션 사용에 대한 자세한 정보를 제공해야 합니다.	
E10.FR.05	E10.FR.04 AND 충전기의 메모리가 부족합니다.	충전기는 TransactionEventRequest(eventType = Ended) 메시지에서 미터 데이터를 삭제할 수 있습니다.	
E10.FR.06	E10.FR.05	미터 데이터를 삭제할 때 충전기는 먼저 중간 값(첫 번째 값, 세 번째 값, 다섯 번째 등)을 삭제해야 하며, 목록의 시작 부분에서 값을 삭제하거나 목록에 값을 추가하는 것을 중지해서는 안 됩니다.	
E10.FR.07	`SampledDataSignReadings`는 *true* 입니다.	충전기는 서명된 미터 값을 검색하여 sampledValues의 *signedMeterValue* 필드에 넣어야 합니다.	

E11 - Connection Loss During Transaction

Table 116. E11 - 트랜잭션 중 연결 손실

번호	유형	설명
1	**이름**	트랜잭션 중 연결 손실
2	**ID**	E11
	기능 블록	E. 트랜잭션
3	**목표**	충전기가 연결이 끊어진 동안에도 충전기가 트랜잭션을 이어갈 수 있도록 함
4	**설명**	이 사용 사례는 충전기가 CSMS와의 연결을 끊고 다시 연결하는 동안에도 진행 중인 트랜잭션을 계속할 수 있는 방법을 설명합니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 트랜잭션이 진행 중인 동안 충전기의 연결이 끊어졌습니다. **2.** 충전기의 트랜잭션 이벤트가 저장됩니다. **3.** CSMS와의 연결이 복구됩니다. **4.** 충전기는 TransactionEventRequest(offline = TRUE)를 사용하여 저장된 트랜잭션 이벤트를 CSMS로 보냅니다. **5.** 충전기가 정기적인 통신을 재개합니다.
	대체 시나리오	E04 - Offline Start Transaction
5	**필수 조건**	트랜잭션이 진행 중이고 연결이 끊어짐.
6	**사후 조건**	**성공적인 사후 조건:** 충전기가 정상적인 통신을 재개합니다. **실패 사후 조건:** n/a

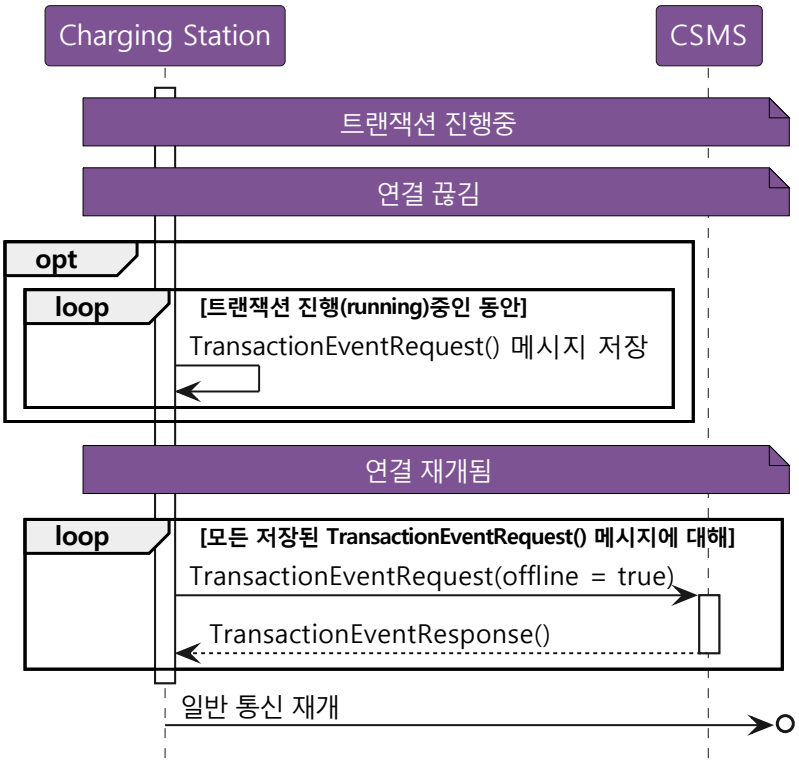

Figure 60. 시퀀스 다이어그램: 트랜잭션 중 연결 끊김

7	**오류 처리**	n/a
8	**참고**	n/a

E11 - Connection Loss During Transaction - Requirements

Table 117. E11 - Requirements

ID	전제 조건	요구 사항 정의
E11.FR.01	*offline* 일 때	충전기는 충전기가 온라인 상태였다면 CSMS에 전송했을 모든 TransactionEventRequest 메시지를 큐에 넣어야 합니다.
E11.FR.02	연결이 복구된 후.	충전기는 플래그 *offline* 을 TRUE로 설정하여 큐에 넣은 TransactionEventRequest 메시지를 전송해야 합니다.
E11.FR.03	TransactionEventRequest(eventType = Updated)에서 미터 데이터를 보내도록 구성된 경우 다음을 참조하세요. Meter Value - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Updated)의 선택적 MeterValue 필드에 추가하여 트랜잭션 중에 더 자세한 정보를 제공해야 합니다.
E11.FR.04	E11.FR.03 AND *offline* AND 충전기의 메모리가 부족합니다.	충전기는 TransactionEventRequest(eventType = Updated) 메시지를 삭제할 수 있습니다.
E11.FR.05	E11.FR.04	TransactionEventRequest(eventType = Updated) 메시지를 삭제할 때 충전기는 중간 메시지를 먼저 삭제해야 합니다(첫 번째 메시지, 세 번째 메시지, 다섯 번째 메시지 등). 처음부터 메시지를 삭제하거나 큐에 메시지를 추가하는 것을 중단해서는 안 됩니다.

ID	전제 조건	요구 사항 정의
E11.FR.06	E11.FR.03 AND 1개의 TransactionEventRequest(eventType = Updated)에 대한 미터 데이터 양이 너무 많은 경우.	충전기는 동일한 *timestamp* 를 가진 여러 개의 TransactionEventRequest(eventType = Updated) 메시지로 미터 데이터를 분할할 수 있습니다.
E11.FR.07		충전기가 오프라인이 되면 대기열에 남아 있는 모든 메시지는 *offline* 으로 설정해야 합니다.
E11.FR.08	`SampledDataSignReadings`는 *true* 입니다.	충전기는 서명된 미터 값을 검색하여 sampledValues의 *signedMeterValue* 필드에 넣어야 합니다.

E12 - Inform CSMS of an Offline Occurred Transaction

Table 118. E12 - CSMS에 오프라인 발생 트랜잭션 알림

번호	유형	설명
1	**이름**	CSMS에 오프라인 발생 트랜잭션 알림
2	**ID**	E12
	기능 블록	E. 트랜잭션
3	**목표**	충전기가 *offline* 상태일 때 트랜잭션이 발생했음을 CSMS에 알릴 수 있도록 합니다.
4	**설명**	이 사용 사례는 충전기가 연결 손실 이후 트랜잭션을 시작하고 중지하는 방법을 다룹니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS와의 연결이 복구됩니다. **2.** 충전기가 CSMS에 Heartbeat 메시지를 보냅니다. **3.** 충전기가 TransactionEventRequest (eventType = Started, offline = TRUE)를 CSMS에 보냅니다. **4.** CSMS가 TransactionEventResponse로 응답하여 트랜잭션을 수락합니다. **5.** 충전기에서 TransactionEventRequest (eventType = Updated, offline = TRUE)를 전송합니다. **6.** CSMS에서 TransactionEventResponse로 응답합니다. **7.** 충전기에서 TransactionEventRequest (eventType = Ended, offline = TRUE)를 전송합니다. **8.** CSMS에서 TransactionEventResponse로 응답합니다.
5	**필수 조건**	최소 한 개의 *offline* 트랜잭션이 발생했습니다.
6	**사후 조건**	**성공한 사후 조건:** CSMS에서 *offline* 으로 발생한 모든 트랜잭션을 처리했습니다. **실패 사후 조건:** 없음

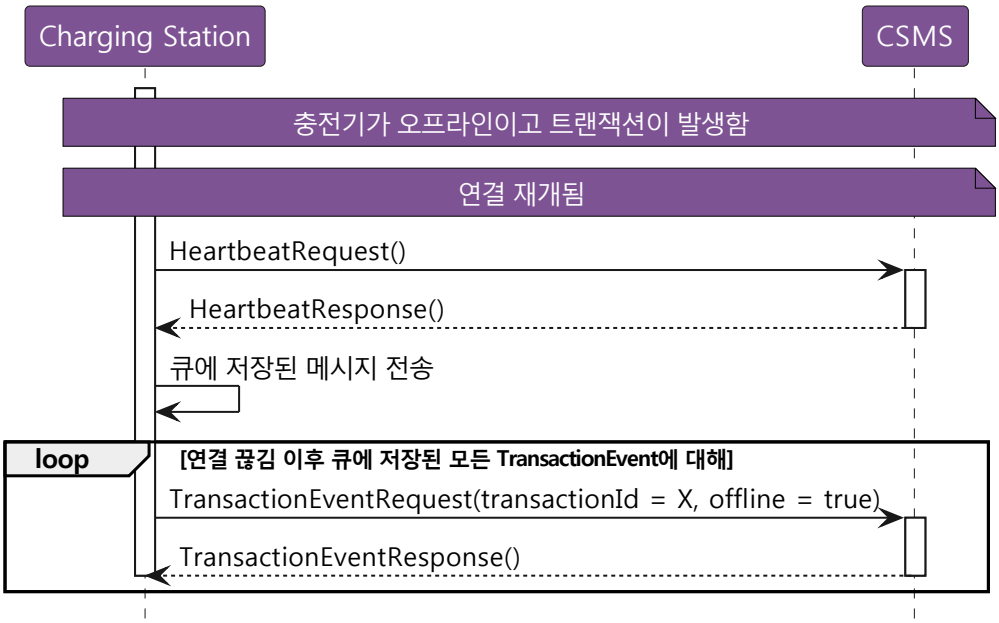

Figure 61. Sequence Diagram: Inform CSMS of an Offline Occurred Transaction

7	**오류 처리**	n/a

8	**참고**	n/a

E12 - Inform CSMS of an Offline Occurred Transaction - Requirements

Table 119. E12 - 요구 사항

ID	전제 조건	요구 사항 정의
E12.FR.01	*offline* 일 때	충전기는 충전기가 온라인 상태였다면 CSMS에 전송했을 모든 TransactionEventRequest 메시지를 큐에 넣어야 합니다.
E12.FR.02	연결이 복구된 후.	충전기는 플래그 *offline* 을 TRUE로 설정하여 큐에 넣은 TransactionEventRequest 메시지를 전송해야 합니다.
E12.FR.03	TransactionEventRequest(eventType = Updated)에서 미터 데이터를 보내도록 구성된 경우 다음을 참조하세요. Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Updated)의 선택적 MeterValue 필드에 추가하여 트랜잭션 중에 더 자세한 정보를 제공해야 합니다.
E12.FR.04	E12.FR.03 및 *Offline* 및 충전기의 메모리가 부족합니다.	충전기는 TransactionEventRequest(eventType = Updated) 메시지를 삭제할 수 있습니다.
E12.FR.05	E12.FR.04	TransactionEventRequest(eventType = Updated) 메시지를 삭제할 때 충전기는 중간 메시지를 먼저 삭제해야 합니다(첫 번째 메시지, 세 번째 메시지, 다섯 번째 메시지 등). 처음부터 메시지를 삭제하거나 큐에 메시지를 추가하는 것을 중단해서는 안 됩니다.
E12.FR.06	E12.FR.03 및 1개의 TransactionEventRequest(eventType = Updated)에 대한 미터 데이터 양이 너무 많은 경우.	충전기는 동일한 *timestamp* 를 가진 여러 개의 TransactionEventRequest(eventType = Updated) 메시지로 미터 데이터를 분할할 수 있습니다.
E12.FR.07	TransactionEventRequest(eventType = Ended)에서 미터 데이터를 보내도록 구성된 경우 다음을 참조하세요. Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Ended)의 선택적 meterValue 필드에 추가하여 트랜잭션 사용에 대한 자세한 정보를 제공해야 합니다.
E12.FR.08	E12.FR.07 AND 충전기의 메모리가 부족합니다.	충전기는 TransactionEventRequest(eventType = Ended) 메시지에서 미터 데이터를 삭제할 수 있습니다.
E12.FR.09	E12.FR.08	미터 데이터를 삭제할 때 충전기는 중간 값을 먼저 삭제해야 합니다(첫 번째 값, 세 번째 값, 다섯 번째 등). 목록의 시작 부분에서 값을 삭제하거나 목록에 값을 추가하는 것을 중단해서는 안 됩니다.
E12.FR.10	`SampledDataSignReadings`는 *true* 입니다.	충전기는 서명된 미터 값을 검색하여 sampledValues의 *signedMeterValue* 필드에 넣어야 합니다.

E13 - Transaction-related message not accepted by CSMS

Table 120. E13 - Transaction-related message not accepted by CSMS

번호	유형	설명
1	**이름**	CSMS에서 수락하지 않은 트랜잭션 관련 메시지
2	**ID**	E13
	기능 블록	E. 트랜잭션
3	**목표**	충전기에서 수락하지 않은 메시지를 처리하는 방법을 정의합니다.
4	**설명**	CSMS에서 트랜잭션 관련 메시지를 수락하지 않거나 MessageTimeout 내에 응답하지 않는 상황/문제가 있습니다. 대부분은 오류 시나리오입니다. 이와 같은 일이 발생하면 충전기는 메시지를 몇 번 다시 시도해야 합니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 충전기에서 CSMS에 트랜잭션 관련 메시지를 보냅니다. **2.** 메시지가 수락되지 않고 `MessageAttemptsTransactionEvent`에 도달하지 못했습니다. **3.** 충전기는 동일한 메시지의 이전 전송 횟수를 `MessageAttemptIntervalTransactionEvent`와 곱한 초 동안 기다립니다. **4.** 충전기는 트랜잭션 관련 메시지를 CSMS로 다시 보냅니다.
5	**필수 조건**	n/a
6	**사후 조건**	**성공 사후 조건:** MessageAttemptsTransactionEvent에 도달하지 않고 트랜잭션 관련 메시지가 수락됩니다. MessageAttemptsTransactionEvent에 도달하고 트랜잭션 관련 메시지가 삭제됩니다. **실패 사후 조건:** MessageAttemptsTransactionEvent에 도달하지 않고 트랜잭션 관련 메시지가 삭제됩니다. MessageAttemptsTransactionEvent에 도달하고 트랜잭션 관련 메시지가 수락됩니다.

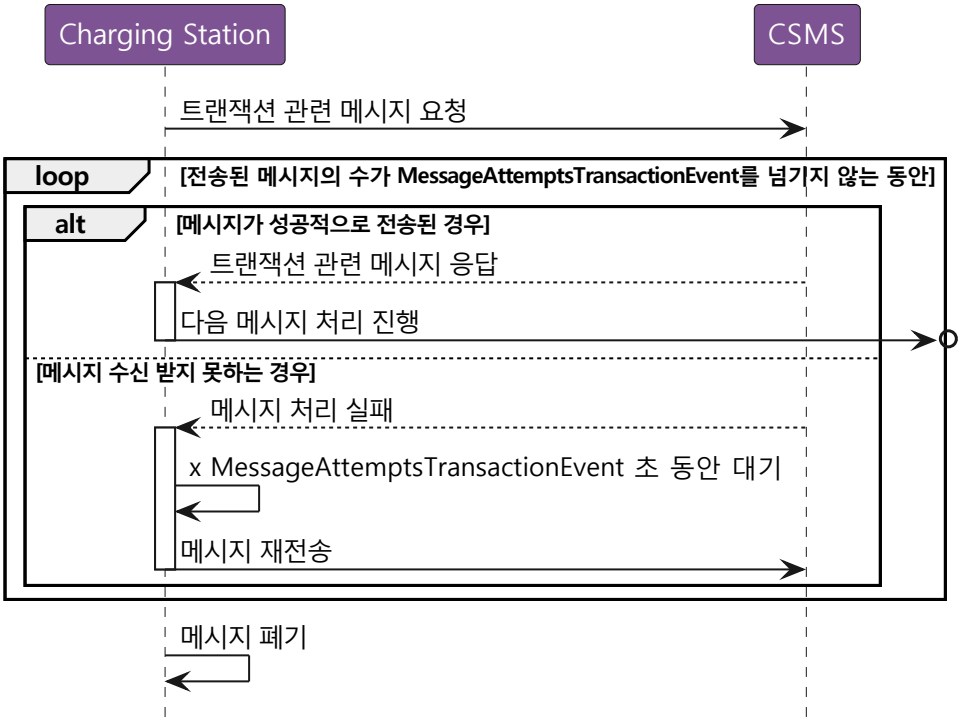

Figure 62. 시퀀스 다이어그램: CSMS에서 수락되지 않은 트랜잭션 관련 메시지

7	**오류 처리**	n/a

8	**참고**	이 사용 사례는 CSMS가 메시지를 수락하지 않거나 메시지 시간 초과 내에 응답하지 않을 때 예상되는 동작을 설명합니다. 이는 충전기와 CSMS 간의 통신이 *offline* 인 상황과 다릅니다.

E13 - Transaction-related message not accepted by CSMS - Requirements

Table 121. E13 - 요구 사항

ID	전제 조건	요구 사항 정의
E13.FR.01		충전기가 실패한 트랜잭션 관련 메시지를 재시도해야 하는 횟수와 간격은 `MessageAttemptsTransactionEvent` 및 `MessageAttemptIntervalTransactionEvent` 구성 변수를 사용하여 구성할 수 있습니다.
E13.FR.02	충전기가 특정 트랜잭션 관련 메시지를 전달하는 데 처음으로 실패할 때.	충전기는 이 메시지를 처리하는 데 실패하는 일이 계속 발생하고 `MessageAttemptsTransactionEvent` 구성 변수에 지정된 만큼의 이 메시지 처리 실패가 아직 발생하지 않은 한 이 메시지를 다시 보내야 합니다.
E13.FR.03	CSMS는 트랜잭션 관련 메시지를 수락하지 않습니다.	충전기는 `MessageAttemptIntervalTransactionEvent` 키에 지정된 시간(초)에 이 동일한 메시지의 이전 전송 횟수를 곱한 값을 대기해야 합니다.
E13.FR.04	최종 시도가 실패하는 경우.	충전기는 메시지를 삭제하고 다음 트랜잭션 관련 메시지가 있는 경우 계속 진행해야 합니다.

E13 - Transaction-related message not accepted by CSMS - Example

예를 들어, `MessageAttemptsTransactionEvent` 구성 변수에 "3" 값이 있고 `MessageAttemptIntervalTransactionEvent` 구성 변수에 "60" 값이 있는 충전기를 생각해 보세요. 이 충전기는 TransactionEventRequest 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 메시지(있는 경우)를 계속 진행합니다.

E14 - Check Transaction status

Table 122. E14 - 트랜잭션 상태 확인

번호	유형	설명
1	**이름**	트랜잭션 상태 확인
2	**ID**	E14
	기능 블록	E. 트랜잭션
3	**목표**	CSMS가 트랜잭션 상태를 요청하고 대기 중인 트랜잭션 관련 메시지가 있는지 확인할 수 있도록 합니다.
4	**설명**	CSMS가 전달해야 하는 트랜잭션에 대한 메시지가 아직 있는지 알아야 하는 시나리오가 있습니다. 예를 들어: CSMS가 TransactionEventRequest(eventType = Ended)를 수신하고, 이 트랜잭션에 대한 청구 프로세스를 시작하려고 하지만 여전히 일부 중간 메시지가 누락되어 있음을 감지합니다(메시지의 시퀀스 번호를 통해 확인할 수 있음). GetTransactionStatusRequest에서 transactionId를 지정하여 충전기에 이 트랜잭션에 대한 대기열에 메시지가 있는지 물어볼 수 있습니다. 결과에 따라 CSMS는 예를 들어 메시지가 전달될 때까지 기다리거나 정보 없이 청구 프로세스를 시작할 수 있습니다. 또한 트랜잭션이 아직 진행 중인지 알아야 할 수도 있습니다. CSMS가 대기열에 트랜잭션 관련 메시지가 있는지(특정 트랜잭션에 대한 메시지가 아닌) 알고 싶은 경우 transactionId 없이 GetTransactionStatusRequest를 보낼 수 있습니다.
	Actors	CSMS, 충전기
	시나리오 설명	**1.** CSMS는 GetTransactionStatusRequest를 충전기로 보냅니다. 이때 transactionId가 있거나 없어도 됩니다. **2.** 충전기는 GetTransactionStatusResponse로 응답합니다.
5	**필수 조건**	CSMS는 상태를 알고자 하는 트랜잭션의 transactionId를 알고 있습니다.
6	**사후 조건**	**성공 사후 조건:** CSMS는 요청된 트랜잭션의 상태를 알고 있습니다. **실패 사후 조건:** CSMS는 요청된 트랜잭션의 상태를 알지 못합니다.

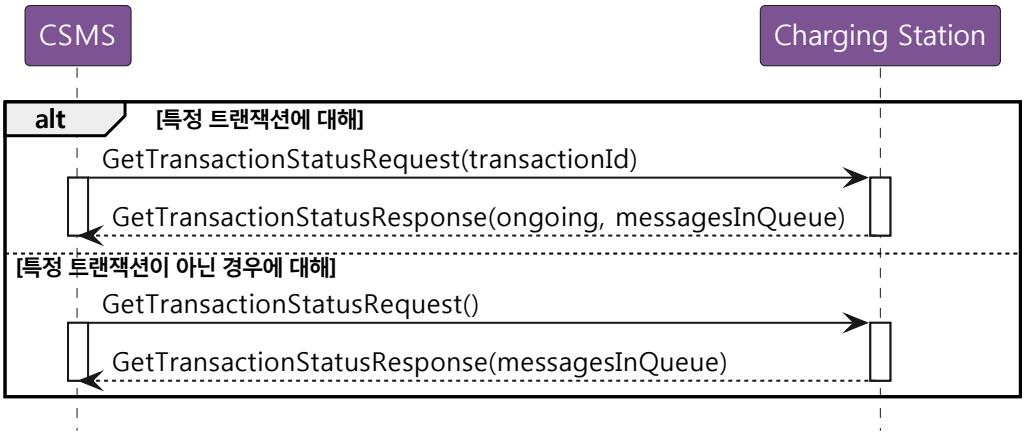

Figure 63. 시퀀스 다이어그램: 트랜잭션 상태 확인

7	**오류 처리**	n/a

8	**참고**	CSMS가 두 필드(*ongoingIndicator* 및 *messagesInQueue*)가 모두 false로 설정된 GetTransactionStatusResponse를 수신하는 경우, 이는 트랜잭션이 완료되었고 이 트랜잭션에 대한 대기열에 더 이상 메시지가 없거나 충전기가 이 트랜잭션에 대해 아무것도 알지 못한다는 것을 의미할 수 있습니다(더 이상).

E14 - Check transaction status - Requirements

Table 123. E14 - Requirements

ID	전제 조건	요구 사항 정의
E14.FR.01	충전기는 transactionId와 함께 GetTransactionStatusRequest를 수신하고 해당 transactionId로 트랜잭션을 수행하지 않았습니다.	충전기는 *ongoingIndicator* = false AND *messagesInQueue* = false로 응답해야 합니다.
E14.FR.02	충전기는 transactionId와 함께 GetTransactionStatusRequest를 수신하고 해당 transactionId로 트랜잭션이 아직 중단되지 않았습니다.	충전기의 응답은 *ongoingIndicator* = true여야 합니다.
E14.FR.03	충전기는 transactionId가 있는 GetTransactionStatusRequest를 수신하고 해당 transactionId가 있는 트랜잭션이 중단되었습니다.	충전기의 응답은 *ongoingIndicator* = false여야 합니다.
E14.FR.04	충전기는 transactionId가 있는 GetTransactionStatusRequest를 수신하고 해당 transactionId가 있는 트랜잭션에 대해 전달할 트랜잭션 관련 메시지가 있습니다.	충전기의 응답은 *messagesInQueue* = true여야 합니다.
E14.FR.05	충전기는 transactionId가 있는 GetTransactionStatusRequest를 수신하고 해당 transactionId가 있는 트랜잭션에 대해 전달할 트랜잭션 관련 메시지가 없습니다.	충전기의 응답은 *messagesInQueue* = false여야 합니다.
E14.FR.06	충전기는 transactionId 없이 GetTransactionStatusRequest를 수신합니다.	충전기의 응답은 *ongoingIndicator* = 설정되어서는 안 됩니다.
E14.FR.07	충전기는 transactionId 없이 GetTransactionStatusRequest를 수신하고 전달할 트랜잭션 관련 메시지가 있습니다.	충전기의 응답은 *messagesInQueue* = true여야 합니다.
E14.FR.08	충전기는 transactionId 없이 GetTransactionStatusRequest를 수신하고 전달할 트랜잭션 관련 메시지가 없습니다.	충전기의 응답은 *messagesInQueue* = false여야 합니다.

2.2. Interrupting and Stopping ISO 15118 Charging

E15 - End of charging process

Table 124. E15 - 충전 프로세스 종료

번호	유형	설명
1	**이름**	충전 프로세스 종료.
2	**ID**	E15
	기능 블록	E. 트랜잭션
	참조	ISO15118-1 H1 - 충전 프로세스 종료
3	**목표**	ISO15118-1 참조, 사용 사례 목표 H1, 44페이지.
4	**설명**	ISO15118-1 참조, 사용 사례 설명 H1, 44페이지.
	Actors	EV, EVSE, EV 드라이버
	시나리오 설명	ISO15118-1 참조, 사용 사례 설명 H1, 기본 초보 사용 사례 설명, 처음 5개의 글머리 기호와 마지막 2개의 설명, 44페이지. **1.** EV 드라이버가 EV에서 케이블을 분리합니다 **2.** 충전기는 TransactionEventRequest를 eventType eventType = Ended로 CSMS로 보냅니다.
7	**필수 조건**	ISO15118-1, 사용 사례 필수 조건 H1, 44페이지를 참조하세요.
8	**사후 조건**	CSMS가 트랜잭션에 대한 모든 관련 정보를 수신했습니다. ISO15118-1, 사용 사례 종료 조건 H1, 44페이지를 참조하세요.

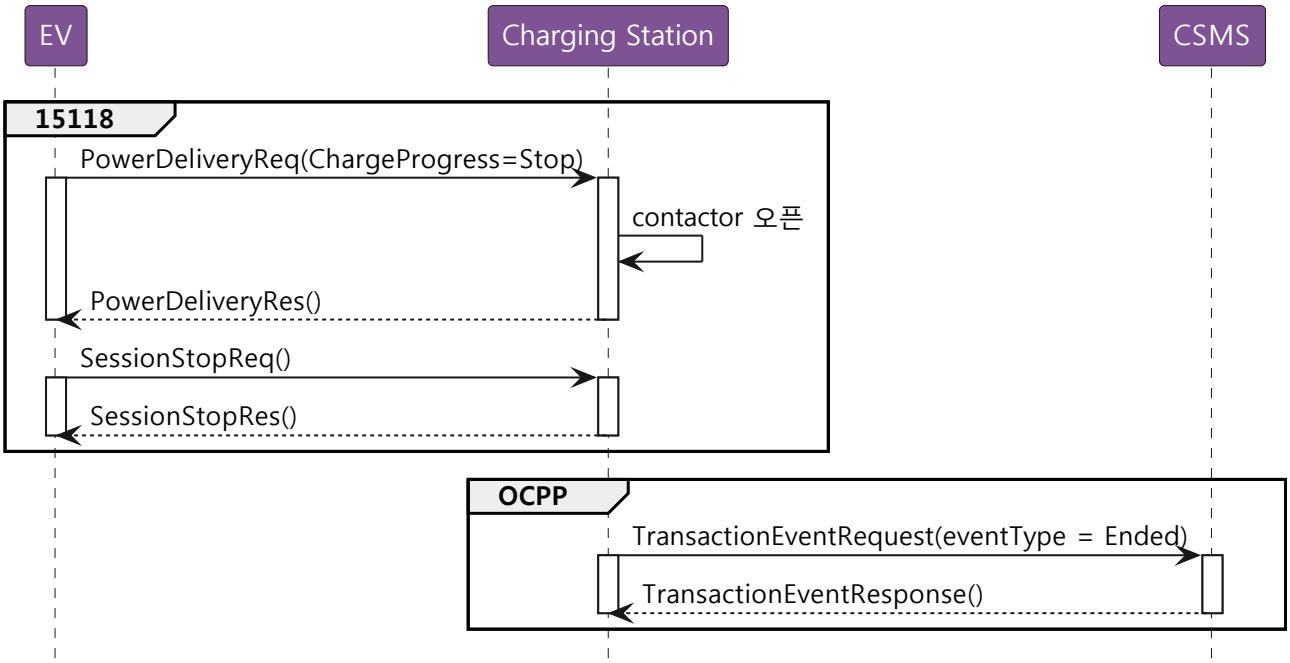

Figure 64. End of charging process

9	**오류 처리**	n/a
10	**참고**	트리거에 대한 ISO15118-1, 사용 사례 요구 사항 H1, 44페이지를 참조하세요. 위의 시나리오 설명과 시퀀스 다이어그램은 중지 트랜잭션에 대한 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStopPoint`: ParkingBayOccupancy, EVConnected, Authorized, DataSigned, PowerPathClosed 이 유즈케이스는 다른 구성에서도 유효하지만, 그 경우 트랜잭션이 다른 시점에서 시작/종료될 수 있으며, 이는 메시지가 전송되는 순서에 영향을 미칠 수 있습니다. 자세한 내용은 사용 사례를 참조하세요. E06 - Stop Transaction options

출처: ISO15118-1

E15 - End of charging process - Requirements

Table 125. E15 - 요구 사항

ID	전제 조건	요구 사항 정의
E15.FR.01	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Ended), 다음을 참조하세요. Meter Values - Configuration	충전기는 구성된 측정값을 선택 사항에 추가해야 합니다. TransactionEventRequest의 meterValue 필드(eventType = Ended)는 CSMS로 전송되어 트랜잭션 사용에 대한 자세한 정보를 제공합니다.
E15.FR.02	E15.FR.01 AND 충전기의 메모리가 부족합니다.	충전기는 TransactionEventRequest(eventType = Ended) 메시지에서 미터 데이터를 삭제할 수 있습니다.
E15.FR.03	E15.FR.02	미터 데이터를 삭제할 때 충전기는 중간 값(첫 번째 값, 세 번째 값, 다섯 번째 등)을 먼저 삭제해야 하며, 목록의 시작 부분에서 값을 삭제하거나 목록에 값을 추가하는 것을 중지해서는 안 됩니다.
E15.FR.04	`TxStopPoint`에 "Authorized" 또는 "PowerPathClosed" 또는 "EnergyTransfer"가 포함되어 있고 충전기가 EV에서 ISO 15118 SessionStopReq(Terminate) 메시지를 수신할 때 *triggerReason* = `StopAuthorized` 인 TransactionEventRequest를 아직 보내지 않은 경우	충전기는 eventType = Ended 및 *triggerReason* = `StopAuthorized` 및 *stoppedReason* = `StoppedByEV` 인 TransactionEventRequest 메시지를 보내서 CSMS에 충전 트랜잭션이 (EV에 의해) 중단되었음을 알려야 합니다.
E15.FR.05	`TxStopPoint`에 "Authorized" 또는 "PowerPathClosed" 또는 "EnergyTransfer"가 포함되어 있지 않고 충전기가 EV에서 ISO 15118 SessionStopReq(Terminate) 메시지를 수신했을 때 *triggerReason* = `StopAuthorized` 인 TransactionEventRequest를 아직 보내지 않은 경우	충전기는 eventType = Updated 및 *triggerReason* = `StopAuthorized` 인 TransactionEventRequest 메시지를 보내 CSMS에 인증이 종료되었음을 알려야 합니다.

F. RemoteControl

1. Introduction

이 기능 블록은 CSMS의 원격 제어 관리를 위한 세 가지 유형의 사용 사례를 설명합니다.

1. 원격 트랜잭션 제어. 이러한 사용 사례는 CSO(또는 간접적으로 제3자)가 원격 명령으로 트랜잭션을 시작/중지할 수 있도록 하는 기능을 설명합니다.
2. 커넥터 잠금 해제. 이러한 사용 사례는 CSO(또는 간접적으로 제3자)가 원격 명령으로 커넥터를 잠금 해제할 수 있도록 하는 기능을 설명합니다. 예를 들어, 고객이 케이블을 뽑는 데 문제가 있을 때 이를 지원하는 데 사용할 수 있습니다.
3. 원격 트리거. 이러한 사용 사례는 OCPP의 모든 원격 트리거 기능을 설명합니다. 이 기능은 메시지의 원격 트리거를 가능하게 합니다. 예를 들어, 메시지를 다시 보내거나 충전기에서 진행 중인 일부 프로세스의 현재 상태를 요청합니다.

2. Use cases & Requirements

2.1. Remote Transaction Control

F01 - Remote Start Transaction - Cable Plugin First

Table 126. F01 - 원격 시작 트랜잭션 - 케이블 플러그인 먼저

번호	유형	설명
1	**이름**	원격 시작 트랜잭션 - 케이블 플러그인 먼저
2	**ID**	F01
	기능 블록	F. 원격 제어
3	**목표**	**1.** CSMS로 트랜잭션을 원격으로 시작합니다. **2.** CSO가 트랜잭션을 시작하는 데 문제가 있는 EV 운전자를 도울 수 있도록 합니다. **3.** 타사(예: 모바일 앱)가 CSMS를 통해 충전 트랜잭션을 제어할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSMS가 RequestStartTransactionRequest를 보내 충전기에 트랜잭션을 시작하도록 원격으로 요청하는 방법을 설명합니다. 수신 시 충전기는 RequestStartTransactionResponse와 트랜잭션을 시작할 수 있는지 여부를 나타내는 상태로 응답합니다.
	Actors	충전기, CSMS, CSO
	시나리오 설명	**1.** EV 운전자가 충전기에 케이블을 꽂습니다. **2.** 충전기는 StatusNotificationRequest를 CSMS에 보내 *Occupied*가 된 커넥터에 대해 알립니다. **3.** CSMS는 StatusNotificationResponse로 응답하여 StatusNotificationRequest가 수신되었음을 확인합니다. **4.** 충전기는 TransactionEventRequest (eventType = Started)를 보내 CSMS에 시작된 트랜잭션에 대해 알립니다(운전자가 아직 알려지지 않은 경우에도 해당). **5.** CSMS는 TransactionEventResponse로 응답하여 TransactionEventRequest가 수신되었음을 확인합니다. **6.** 외부 트리거(이 사용 사례의 예: EV Driver)가 원격 시작을 트리거합니다. **7.** CSMS는 RequestStartTransactionRequest를 충전기로 보냅니다. **8** 충전기는 이미 시작된 트랜잭션의 *transactionId* 를 CSMS에 포함하여 RequestStartTransactionResponse로 응답합니다. **9.** 선택적으로: EV 운전자는 CSMS에서 인증을 받습니다. **10.** 충전기는 TransactionEventRequest (eventType = Updated, chargingState = Charging) 메시지를 보내 CSMS에 충전이 시작되었음을 알립니다.
	대체 시나리오	원격 시작 트랜잭션 - 원격 시작 먼저 F02 - Remote Start Transaction - Remote Start First
5	**필수 조건**	충전 케이블을 먼저 꽂습니다.
6	**사후 조건**	충전기에서 에너지를 제공합니다. `AuthorizeRemoteStart`의 값이 *true* 이면 충전기는 Local Authorization List, Authorization Cache 및/또는 AuthorizeRequest를 사용하여 IdToken을 성공적으로 인증했을 때만 에너지를 제공합니다.

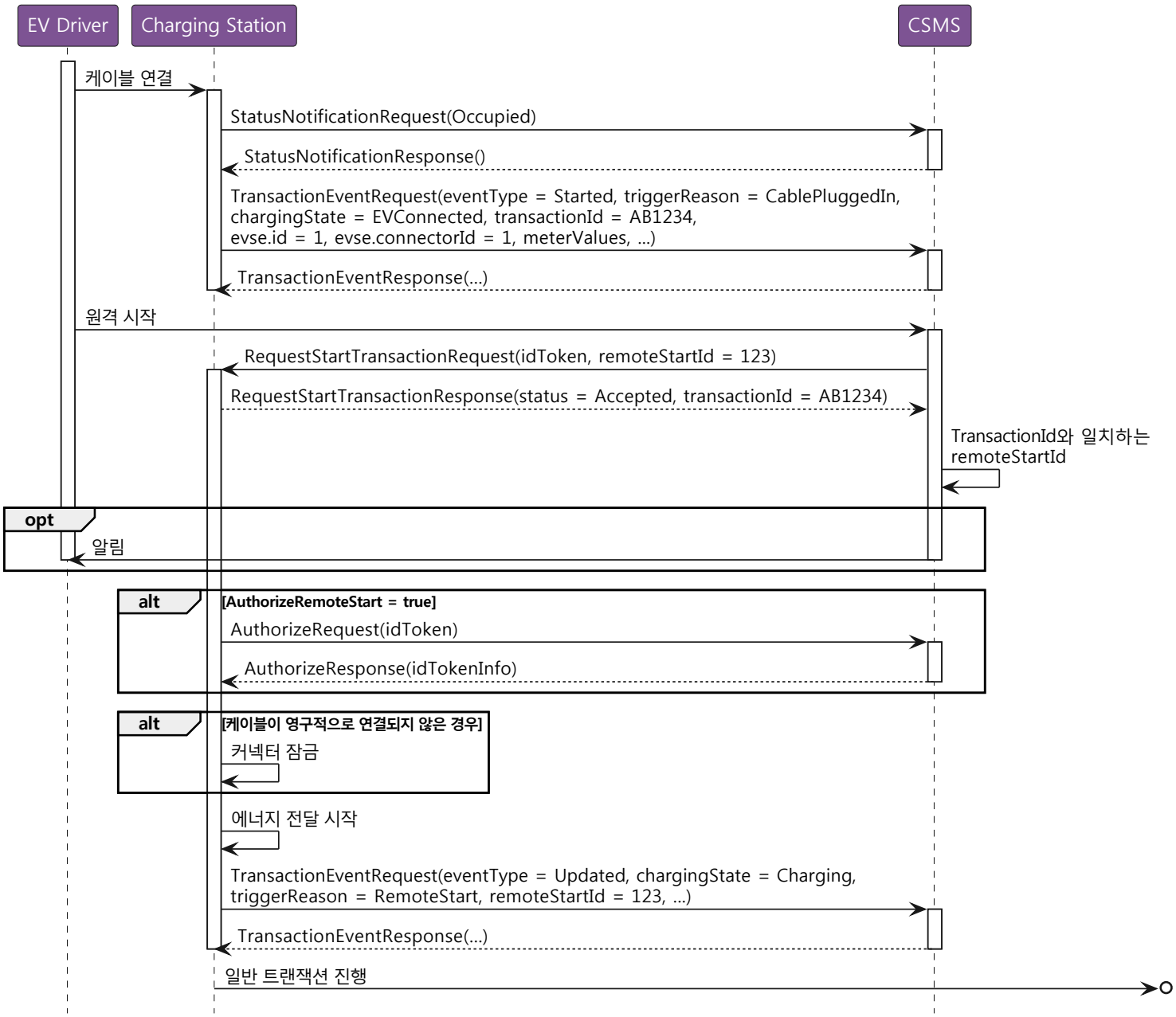

Figure 65. Sequence Diagram: Remote Start Transaction - Cable Plugged in First

7	**오류 처리**	n/a
8	**참고**	외부 트리거는 예를 들어 충전기 운영자 또는 EV 운전자 앱일 수 있습니다. RequestStartTransactionResponse에는 충전기가 요청을 수락했는지 여부를 나타내는 상태가 포함되어 있으며 트랜잭션을 시작하려고 시도합니다. CSMS는 NoAuthorization유형의 IdTokenType의 RequestStartTransactionRequest를 보낼 수 있습니다. 운영자는 충전기가 로컬 트랜잭션 중지을 지원하는 경우 누구나 이 트랜잭션을 중지할 수 있다는 점을 알고 있어야 합니다. 위의 시나리오 설명과 시퀀스 다이어그램은 시작 트랜잭션을 위한 구성 변수가 다음과 같이 구성되는 것을 기반으로 합니다. `TxStartPoint`: EVConnected, Authorized, DataSigned, PowerPathClosed, EnergyTransfer 이 사용 사례는 다른 구성에서도 유효하지만, 그 경우 트랜잭션이 다른 시점에서 시작/종료될 수 있으며, 이는 메시지가 전송되는 순서에 영향을 미칠 수 있습니다. 자세한 내용은 다음 사용 사례를 참조하세요. E01 - Start Transaction options.

F01 - Remote Start Transaction - Cable Plugin First - Requirements

Table 127. F01 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
F01.FR.01	`AuthorizeRemoteStart`의 값이 true이고 충전기가 RequestStartTransactionRequest를 수신하는 경우	충전기는 RequestStartTransactionRequest 메시지에서 제공된 IdToken의 성공적인 인증 후 에너지 전송을 허용하기 위해 충전기에서 로컬 작업에 대한 응답으로 동작해야 합니다.	충전기는 먼저 요청에 응답한 다음 로컬 인증 목록, 인증 캐시 및/또는 AuthorizeRequest를 사용하여 IdToken을 인증하려고 시도합니다. 에너지 전송은 인증을 받은 후에만 허용됩니다.
F01.FR.02	`AuthorizeRemoteStart`의 값이 false이고 충전기가 RequestStartTransactionRequest를 수신하는 경우	충전기는 인증을 확인하지 않고 RequestStartTransactionRequest 메시지에 제공된 IdToken에 대한 에너지 전송을 허용해야 합니다.	충전기는 먼저 요청에 응답하고 idToken과 함께 TransactionEventRequest를 CSMS로 보낼 때 CSMS는 이 TransactionEventRequest를 처리할 때 IdToken의 권한 상태를 확인합니다.
F01.FR.03	F01.FR.01 또는 F01.FR.02	충전기는 TransactionEventRequest를 CSMS로 보내야 하며 CSMS는 이 TransactionEventRequest를 처리할 때 IdToken의 인증 상태를 확인합니다.	CSMS가 `Accepted` 가 아닌 인증 상태를 반환하는 경우 충전기는 사용 사례 E05에 따라 에너지 전송을 중지해야 합니다.
F01.FR.04		RequestStartTransactionRequest에는 IdToken이 포함되어야 하며, 충전기는 CSMS로 전송된 TransactionEventRequest에서 트랜잭션을 시작할 수 있는 경우 이 토큰을 사용해야 합니다.	
F01.FR.05		트랜잭션은 E02 - Start Transaction - Cable Plugin First에 설명된 것과 같은 방식으로 시작해야 합니다.	
F01.FR.06		RequestStartTransactionRequest에는 특정 EVSE에서 트랜잭션을 시작해야 하는 경우 evseId가 포함될 수 있습니다.	evseId가 제공되지 않으면 충전기가 EVSE 선택을 제어합니다.
F01.FR.07	RequestStartTransactionRequest에 evseId가 없는 경우.	충전기는 RequestStartTransactionRequest를 거부할 수 있습니다.	
F01.FR.08		CSMS는 RequestStartTransactionRequest에 ChargingProfile을 포함할 수 있습니다.	
F01.FR.09	F01.FR.08	이 ChargingProfile의 목적은 TxProfile로 설정되어야 합니다.	
F01.FR.10	F01.FR.08	충전기는 이 ChargingProfile을 이 RequestStartTransaction에 의해 시작된 트랜잭션에 사용해야 합니다.	
F01.FR.11	F01.FR.08	ChargingProfile의 transactionId는 설정되지 않아야 합니다.	

ID	전제 조건	요구 사항 정의	참고
F01.FR.12	스마트 충전을 지원하지 않는 충전기가 RequestStartTransactionRequest를 ChargingProfile과 함께 수신하는 경우.	충전기는 지정된 ChargingProfile을 무시해야 합니다.	장치 모델 변수 SmartChargingCtrlr.Enabled는 CSMS에 스마트 충전이 지원되는지 여부를 알려줍니다.
F01.FR.13	충전기에서 트랜잭션이 생성되었지만 인증되지 않은 경우. AND RequestStartTransactionRequest가 수신되는 경우.	충전기는 RequestStartTransactionResponse에서 *transactionId* 를 반환해야 합니다.	
F01.FR.14	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Started), 다음을 참조하세요: Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Started)의 선택적 MeterValue 필드에 추가하여 트랜잭션 중에 더 자세한 정보를 제공해야 합니다.	
F01.FR.15	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Updated), 다음을 참조하세요: Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Updated)의 선택적 MeterValue 필드에 추가하여 트랜잭션 중에 더 자세한 정보를 제공해야 합니다.	
F01.FR.16	F01.FR.15 AND 1 TransactionEventRequest에 대한 미터 데이터 양이 너무 많습니다(eventType = Updated)	충전기는 미터 데이터를 동일한 *timestamp* 를 사용하여 여러 TransactionEventRequest (eventType = Updated) 메시지로 분할할 수 있습니다.	
F01.FR.17	TransactionEventRequest를 보낼 때	충전기는 triggerReason을 설정하여 CSMS에 이벤트를 트리거한 원인을 알려야 합니다. 어떤 사유를 사용해야하는지는 TriggerReasonEnumType에 설명되어 있습니다.	
F01.FR.18	트랜잭션이 시작된 후	충전기는 트리거 이벤트가 발생할 때 트랜잭션 중에 추가 TransactionEventRequest(eventType = Updated) 메시지를 보낼 수 있습니다.	
F01.FR.19	RequestStartTransactionRequest가 수신될 때.	다음 TransactionEventRequest에는 *triggerReason* : RemoteStart이 포함되어야 합니다.	
F01.FR.20	RequestStartTransactionRequest에 *evseId* 가 포함되어 있지 않고 충전기에서 EVSE를 선택할 수 있는 경우	충전기는 작동에 대한 *evseId* 값으로 사용할 EVSE를 선택해야 함	충전기에서 임의의 EVSE에서 시작하는 것을 지원하지 않는 경우 F01.FR.07도 참조하세요.
F01.FR.21	RequestStartTransactionRequest에 대한 *evseId* 가 요청의 *idToken* 과 다른 *idToken* 에 예약되어 있고 *groupIdToken* 에 대한 예약이 없는 경우	충전기는 RequestStartTransactionResponse와 *status* = `Rejected` 로 응답해야 합니다.	
F01.FR.22	RequestStartTransactionRequest의 *evseId* 가 요청의 *idToken* 과 다른 *idToken* 에 대해 `Reserved` 이고 요청의 *groupIdToken* 과 다른 *groupIdToken* 에 대해 예약된 경우	충전기는 RequestStartTransactionResponse와 *status* = `Rejected` 로 응답해야 합니다.	*idToken* 과 *idGroupToken* 이 예약과 일치하지 않으면 EV가 스테이션을 사용할 수 없습니다.

ID	전제 조건	요구 사항 정의	참고
F01.FR.23	RequestStartTransactionRequest에 대한 *evse* 가 `Unavailable` 또는 `Faulted` 인 경우	충전기는 *status* = `Rejected` 인 RequestStartTransactionResponse로 응답해야 합니다.	
F01.FR.24	RequestStartTransactionRequest에 대한 *evseId* 가 `Occupied` 이고 이 *evseId* 에 인증된 트랜잭션이 있는 경우	충전기는 *status* = `Rejected` 인 RequestStartTransactionResponse로 응답해야 합니다.	트랜잭션이 없거나 아직 인증되지 않은 트랜잭션이 있는 EVSE만 RequestStartTransactionRequest와 일치될 수 있습니다.
F01.FR.25	F01.FR.13	충전기는 연관된 트랜잭션에 대해 다음 보내는 TransactionEventRequest에 *remoteStartId* 를 넣어야 합니다.	
F01.FR.26	스마트 충전을 **지원하는** 충전기가 잘못된 ChargingProfile이 있는 RequestStartTransactionRequest를 수신하는 경우.	충전기는 *status* = `Rejected` 및 선택적으로 *reasonCode* = "InvalidProfile" 또는 "InvalidSchedule"인 RequestStartTransactionResponse로 응답해야 합니다.	장치 모델 변수 SmartChargingCtrlr.Enabled는 CSMS에 스마트 충전이 지원되는지 여부를 알려줍니다.

F02 - Remote Start Transaction - Remote Start First

Table 128. F02 - 원격 시작 트랜잭션 - 원격 시작 먼저

번호	유형	설명
1	**이름**	원격 시작 트랜잭션 - 원격 시작 먼저
2	**ID**	F02
	기능 블록	F. 원격 제어
	부모 사용 사례	F01 - Remote Start Transaction - Cable Plugin First
3	**목표**	충전기와 EV 간의 연결이 설정되기 전에 RequestStartTransactionRequest가 먼저 전송되는 동안 CSMS가 원격으로 트랜잭션을 시작할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSMS가 사용자를 위해 원격으로 트랜잭션을 시작하는 방법을 다룹니다.
	Actors	충전기, CSMS, 외부 트리거
	시나리오 설명	**1.** 외부 트리거가 원격 시작을 트리거합니다. **2.** CSMS가 충전기로 RequestStartTransactionRequest를 보냅니다. **3.** 충전기는 RequestStartTransactionResponse로 CSMS에 응답합니다. **4.** EV 운전자는 구성 변수 설정에 따라 CSMS에서 인증됩니다. **5.** 충전기는 TransactionEventRequest(eventType = Started)를 보내서 CSMS에 시작된 트랜잭션에 대해 알립니다. **6.** 케이블이 연결됩니다. **6a.** 충전기는 StatusNotificationRequest를 *Occupied* 로 보냅니다. **6b.** CSMS는 StatusNotificationResponse를 충전기로 보냅니다. **7.** 에너지 제공이 시작됩니다. **8.** 충전기는 TransactionEventRequest (eventType = Updated, chargingState = Charging) 메시지를 보내서 CSMS에 충전이 시작되었음을 알립니다. **9.** CSMS는 TransactionEventResponse를 충전기로 보냅니다.
5	**필수 조건**	충전 케이블이 연결되지 않았습니다. 원격으로 먼저 시작합니다. 모바일 앱에서 CSMS를 통해 충전 트랜잭션을 제어할 수 있도록 합니다.
6	**사후 조건**	**성공 사후 조건:** 시작 요청이 있는 트랜잭션이 시작되었고 EV가 충전 중입니다. **실패한 사후 조건:** 시작 요청이 있는 트랜잭션이 시작되지 않았거나 EV가 충전 중이 아닙니다.

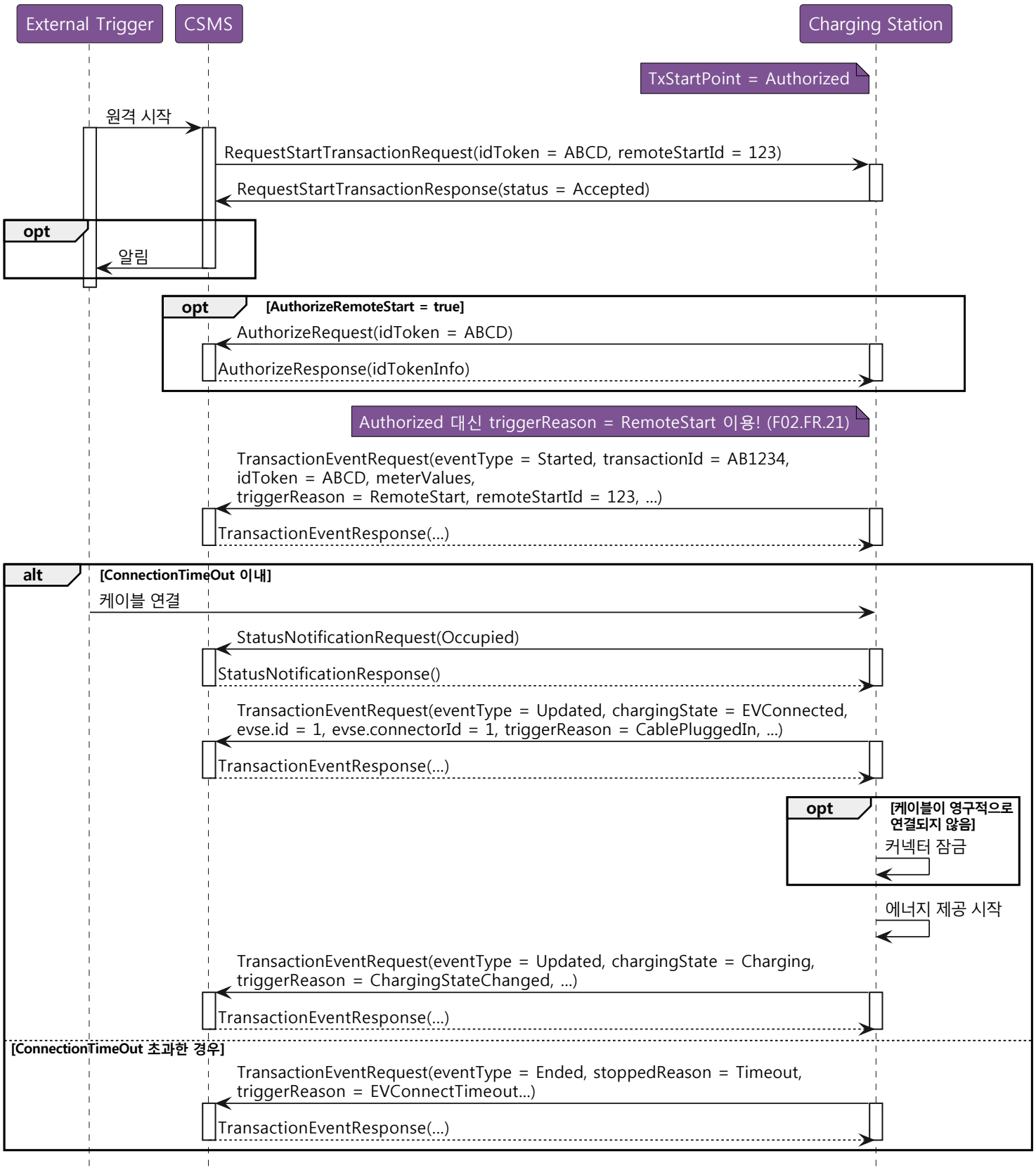

Figure 66. Sequence Diagram: Remote Start Transaction - Remote Start First with TxStartPoint=Authorized

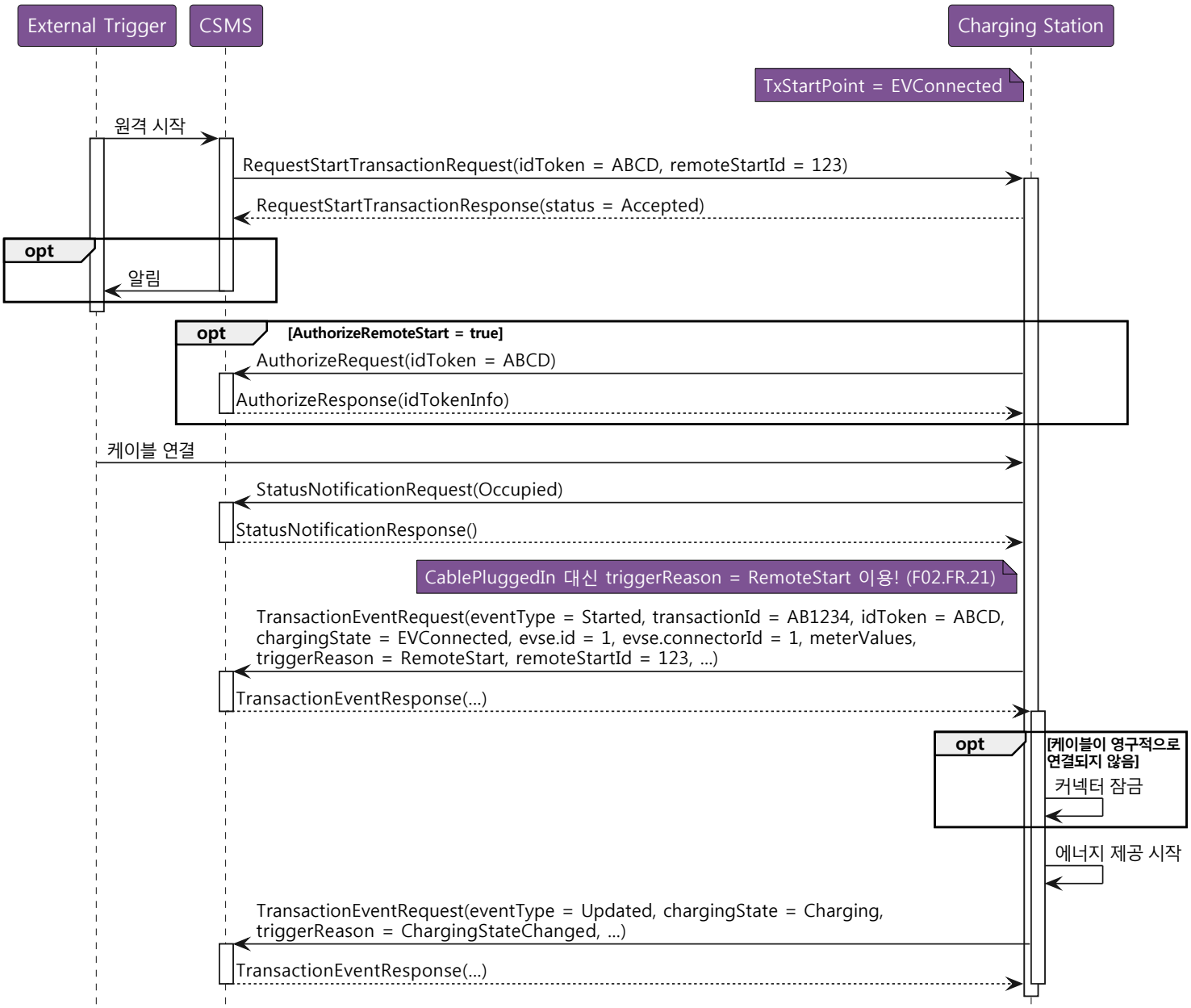

Figure 67. Sequence Diagram: Remote Start Transaction - Remote Start First with TxStartPoint=EVConnected

7	**오류 처리**	n/a
8	**참고**	외부 트리거는 예를 들어 충전기 운영자 또는 EV 운전자 앱일 수 있습니다. 어떤 EVSE를 이용하여 충전을 시작할지 불확실하기 때문에 evseId 없이 원격으로 트랜잭션을 시작하지 않는 것이 좋습니다. 많은 EVSE가 있는 로직 컨트롤러의 경우 EV 운전자가 활성화된 EVSE 앞에 없을 수 있습니다. CSMS는 NoAuthorization 유형의 IdTokenType을 사용하여 RequestStartTransactionRequest를 보낼 수 있습니다. 운영자는 충전기가 로컬 트랜잭션 중지를 지원하는 경우 누구나 이 트랜잭션을 중지할 수 있다는 점을 알고 있어야 합니다. 위의 시나리오 설명과 시퀀스 다이어그램은 시작 트랜잭션을 위한 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStartPoint`: EVConnected, Authorized, DataSigned, PowerPathClosed, EnergyTransfer 이 사용 사례는 다른 구성에서도 유효하지만, 그 경우 트랜잭션이 다른 시점에서 시작/종료될 수 있으며, 이는 메시지가 전송되는 순서에 영향을 미칠 수 있습니다. 자세한 내용은 사용 사례를 참조하세요. E01 - Start Transaction options.

F02 - Remote Start Transaction - Remote Start First - Requirements

Table 129. F02 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
F02.FR.01	RequestStartTransactionRequest의 결과로 트랜잭션이 시작될 때.	충전기는 이 새로운 트랜잭션을 위해 보내는 첫 번째 TransactionEventRequest에 *remoteStartId* 를 넣어야 합니다.	
F02.FR.02	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Started), 다음을 참조하세요: Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Started)의 선택적 MeterValue 필드에 추가하여 트랜잭션 중에 더 자세한 정보를 제공해야 합니다.	
F02.FR.03	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Updated), 다음을 참조하세요: Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Updated)의 선택적 MeterValue 필드에 추가하여 트랜잭션 중에 더 자세한 정보를 제공해야 합니다.	
F02.FR.04	F02.FR.03 AND 1 TransactionEventRequest에 대한 미터 데이터 양이 너무 많습니다(eventType = Updated)	충전기는 미터 데이터를 동일한 *timestamp* 를 사용하여 여러 TransactionEventRequest (eventType = Updated) 메시지로 분할할 수 있습니다.	
F02.FR.05	IdToken 정보가 알려진 경우.	다음 TransactionEventRequest에는 IdTokenType 정보가 포함되어야 합니다.	
F02.FR.06	이 트랜잭션은 특정 IdToken에 대한 예약을 종료합니다.	다음 TransactionEventRequest에는 reservationId가 포함되어야 합니다.	H. Reservation을 참조하세요.
F02.FR.07	EV 운전자가 `EVConnectionTimeOut` 구성 변수에서 설정한 시간 초과 전에 충전 케이블을 연결하지 않은 경우 AND TxStopPoint에 `ParkingBayOccupancy` 가 포함되지 않은 경우	충전기는 트랜잭션을 종료하고 TransactionEventRequest(eventType = Ended, ceaseReason = Timeout, *triggerReason* = `EVConnectionTimeout`)를 CSMS로 전송해야 합니다.	그렇지 않으면 TxStopPoint에 Authorized가 포함되지 않은 경우 트랜잭션이 종료되지 않습니다.
F02.FR.08	EV 운전자가 `EVConnectionTimeOut` 구성 변수에서 설정한 시간 초과 전에 충전 케이블을 연결하지 않는 경우 AND TxStopPoint에 `ParkingBayOccupancy` 가 포함되어 있음	충전기는 트랜잭션을 인증 취소하고 TransactionEventRequest (*triggerReason* = `EVConnectionTimeout`)를 CSMS로 전송해야 합니다.	운전자가 주차 공간을 떠나면 트랜잭션이 정상적으로 종료됩니다.
F02.FR.09	`AuthorizeRemoteStart`의 값이 true이고 충전기가 RequestStartTransactionRequest를 수신합니다.	충전기는 RequestStartTransactionRequest 메시지에서 제공된 IdToken의 성공적인 인증 후 트랜잭션을 시작하기 위해 충전기에서 로컬 작업에 대한 응답으로 동작해야 합니다.	충전기는 먼저 요청에 응답한 다음 로컬 인증 목록, 인증 캐시 및/또는 AuthorizeRequest를 사용하여 IdToken을 인증하려고 시도합니다. 인증이 획득된 후에만 트랜잭션이 시작됩니다.

ID	전제 조건	요구 사항 정의	참고
F02.FR.10	AuthorizeRemoteStart의 값이 false AND 충전기가 RequestStartTransactionRequest를 수신합니다.	충전기는 권한 확인 없이 RequestStartTransactionRequest 메시지에 제공된 IdToken에 대한 트랜잭션을 시작해야 합니다.	트랜잭션이 시작된 후 충전기는 idToken과 함께 TransactionEventRequest를 CSMS로 보내고 CSMS는 이 TransactionEventRequest를 처리할 때 IdToken의 인증 상태를 확인합니다.
F02.FR.11	F02.FR.09 OR F02.FR.10	충전기는 CSMS에 TransactionEventRequest를 보내야 하며, CSMS는 이 TransactionEventRequest를 처리할 때 IdToken의 인증 상태를 확인합니다.	
F02.FR.12		RequestStartTransactionRequest에는 IdToken이 포함되어야 하며, 충전기는 CSMS에 전송된 TransactionEventRequest에서 트랜잭션을 시작할 수 있는 경우 이를 사용해야 합니다.	
F02.FR.13		트랜잭션은 E03 - Start Transaction - Id Token First에 설명된 것과 동일한 방식으로 시작되어야 합니다.	
F02.FR.14		RequestStartTransactionRequest는 특정 EVSE에서 트랜잭션을 시작해야 하는 경우 evseId를 포함할 수 있습니다.	evseId가 제공되지 않으면 충전기가 EVSE 선택을 제어합니다.
F02.FR.15	RequestStartTransactionRequest에 evseId가 포함되어 있지 않은 경우.	충전기는 RequestStartTransactionRequest를 거부할 수 있습니다.	
F02.FR.16		CSMS는 RequestStartTransactionRequest에 ChargingProfile을 포함할 수 있습니다.	
F02.FR.17	F02.FR.16	이 ChargingProfile의 목적은 TxProfile로 설정되어야 합니다.	
F02.FR.18	F02.FR.16	충전기는 이 ChargingProfile을 이 RequestStartTransaction에 의해 시작된 트랜잭션에 사용해야 합니다.	
F02.FR.19	F02.FR.16	ChargingProfile의 transactionId는 설정되지 않아야 합니다.	
F02.FR.20	스마트 충전을 지원하지 않는 충전기가 ChargingProfile이 포함된 RequestStartTransactionRequest를 수신하는 경우.	충전기는 지정된 ChargingProfile을 무시해야 합니다.	장치 모델 변수 SmartChargingCtrlr.Enabled는 CSMS에 스마트 충전이 지원되는지 여부를 알려줍니다.

ID	전제 조건	요구 사항 정의	참고
F02.FR.21	RequestStartTransactionRequest가 수신될 때.	다음 TransactionEventRequest에는 *triggerReason*: `RemoteStart` 와 RequestStartTransactionRequest의 *remoteStartId* 가 포함되어야 합니다.	이는 CSMS에 이것이 RequestStartTransaction의 결과임을 알리기 위한 것입니다. TxStartPoint=`EVConnected` 인 경우 케이블 연결 시 트랜잭션이 시작되지만 *triggerReason* = `RemoteStart` 를 계속 보내야 합니다. 연결 이벤트는 *chargingState* = `EVConnected` 라는 사실로 보고됩니다.
F02.FR.22	RequestStartTransactionRequest에 *evseId* 가 없고 충전기가 EVSE를 선택할 수 있는 경우	충전기는 작업에 대한 *evseId* 값으로 사용할 EVSE를 선택해야 합니다.	충전기가 임의의 EVSE에서 시작하는 것을 지원하지 않는 경우 F02.FR.15도 참조하세요.
F02.FR.23	RequestStartTransactionRequest에 대한 *evseId* 가 요청의 *idToken* 과 다른 *idToken* 에 대해 `Reserved` 되고 *groupIdToken* 에 대한 예약이 없는 경우	충전기는 *status* = `Rejected` 인 RequestStartTransactionResponse로 응답해야 합니다.	
F02.FR.24	RequestStartTransactionRequest에 대한 *evseId* 가 요청의 *idToken* 과 다른 *idToken* 에 대해 `Reserved` 되고 요청의 *groupIdToken* 과 다른 *groupIdToken* 에 대해 `Reserved` 되는 경우	충전기는 *status* = `Rejected` 인 RequestStartTransactionResponse로 응답해야 합니다.	*idToken* 과 *idGroupToken* 이 예약과 일치하지 않으면 EV가 스테이션을 사용할 수 없습니다.
F02.FR.25	RequestStartTransactionRequest에 대한 *evseId* 가 `Unavailable` 또는 `Faulted` 인 경우	충전기는 RequestStartTransactionResponse로 *status* = `Rejected` 로 응답해야 합니다.	
F02.FR.26	RequestStartTransactionRequest에 대한 *evseId* 가 `Occupied` 이고 이 `evseId` 에 인증된 트랜잭션이 있는 경우	충전기는 RequestStartTransactionResponse로 `status` = `Rejected` 로 응답해야 합니다.	트랜잭션이 없거나 아직 인증되지 않은 트랜잭션이 있는 EVSE만 RequestStartTransactionRequest와 일치할 수 있습니다.
F02.FR.27	스마트 충전을 **지원하는** 충전기가 잘못된 ChargingProfile이 포함된 RequestStartTransactionRequest를 수신하는 경우.	충전기는 RequestStartTransactionResponse와 *status* = `Rejected` 및 선택적으로 *reasonCode* = "InvalidProfile" 또는 "InvalidSchedule"로 응답해야 합니다.	장치 모델 변수 SmartChargingCtrlr.Enabled는 CSMS에 스마트 충전이 지원되는지 여부를 알려줍니다.

NOTE 이전 사용 사례의 요구 사항: F01 - Remote Start Transaction - Cable Plugin First는 F02 - Remote Start Transaction - Remote Start First에도 관련이 있는 것으로 간주됩니다.

F03 - Remote Stop Transaction

Table 130. F03 - 원격 중지 트랜잭션

번호.	유형	설명
1	**이름**	원격 정지 트랜잭션
2	**ID**	F03
	기능 블록	F. 원격 제어
3	**목표**	**1.** CSO가 트랜잭션을 정지하는 데 문제가 있는 EV 운전자를 도울 수 있도록 합니다. ***또는*** **2.** 모바일 앱이 CSMS를 통해 트랜잭션을 제어할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSMS가 충전기에 트랜잭션을 정지하도록 요청하는 방법을 설명합니다.
	Actors	충전기, CSMS, CSO, EV 운전자
	시나리오 설명	**1.** 외부 트리거가 원격 정지를 트리거합니다. **2.** CSMS가 충전기에 RequestStopTransactionRequest를 트랜잭션의 transactionId와 함께 충전기로 보내 트랜잭션을 정지하도록 요청합니다. **3.** 충전기는 RequestStopTransactionResponse와 요청을 수락했는지 여부와 주어진 transactionId를 가진 트랜잭션이 진행 중이며 중지될 것인지를 나타내는 상태로 응답합니다. **4.** 충전이 중지되면 충전기는 TransactionEventRequest(eventType = Updated)를 전송하고 해당되는 경우 커넥터를 잠금 해제합니다. **5.** EV 운전자가 케이블을 분리한 후 충전기는 StatusNotificationRequest를 *Available* 상태로 전송합니다. **6.** 충전기는 트랜잭션을 종료하고 TransactionEventRequest (eventType = *Ended*, stoppedReason = *Remote*) 메시지를 CSMS로 보냅니다.
5	**필수 조건**	트랜잭션이 진행 중입니다.
6	**사후 조건**	**성공 사후 조건:** 중지가 요청된 트랜잭션이 종료되었습니다. **실패한 사후 조건:** 중지가 요청된 트랜잭션이 아직 진행 중입니다.

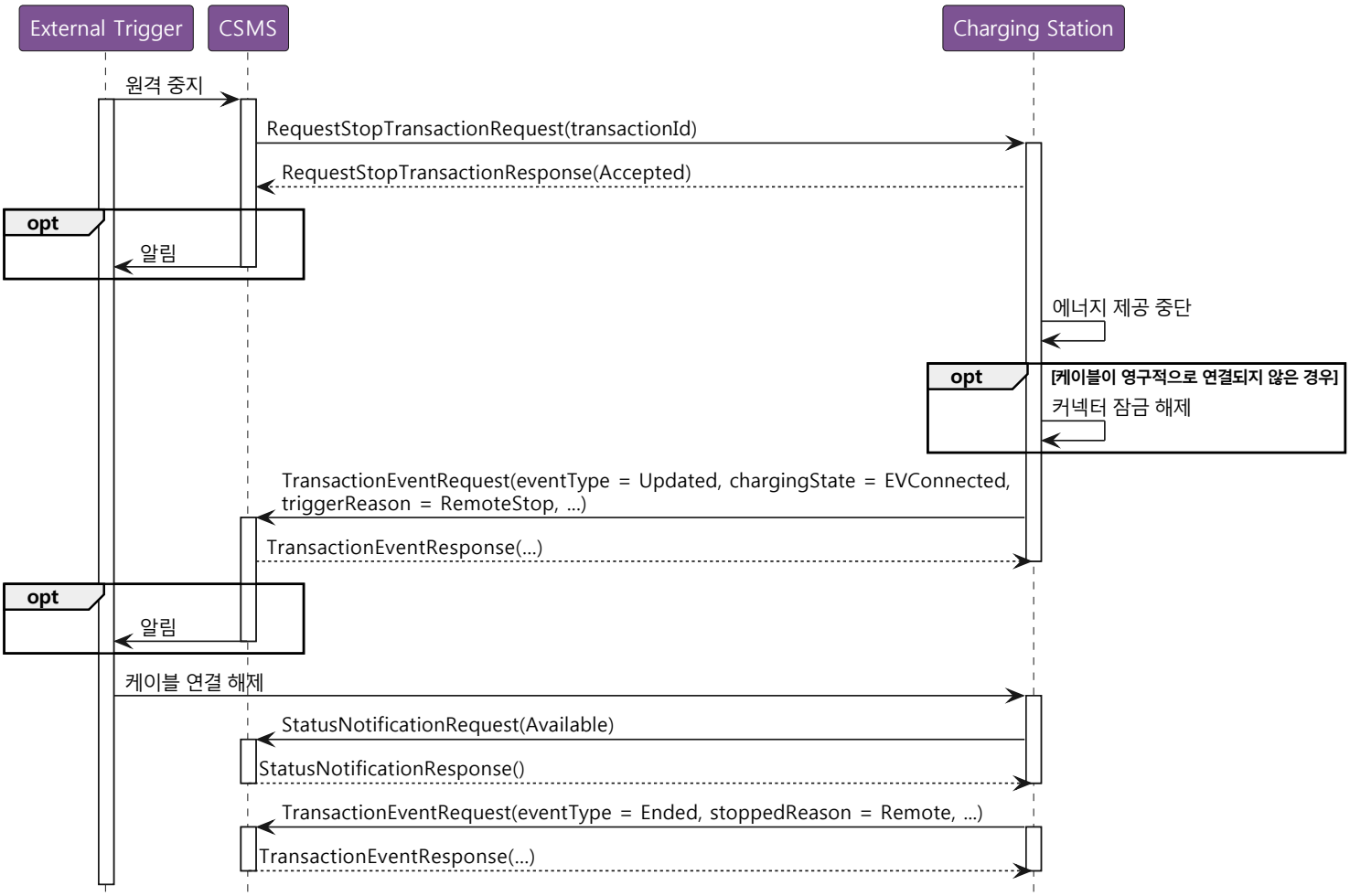

Figure 68. Sequence Diagram: Remote Stop Transaction

7	**참고**	이 원격 트랜잭션 중지 요청은 로컬 트랜잭션 중지 작업과 동일합니다. 위의 시나리오 설명과 시퀀스 다이어그램은 트랜잭션 중지 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStopPoint`: ParkingBayOccupancy, EVConnected 이 사용 사례는 다른 구성에서도 유효하지만, 그 경우 트랜잭션이 다른 시점에서 중지될 수 있으며, 이는 메시지가 전송되는 순서에 영향을 미칠 수 있습니다. 자세한 내용은 사용 사례를 참조하세요. E06 - Stop Transaction options

F03 - Remote Stop Transaction - Requirements

Table 131. F03 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
F03.FR.01	CSMS가 원격 Stop Transaction 트리거를 수신하는 경우(예: 스마트폰 앱을 사용하여 종료하는 경우, (비지역) 선불 크레딧 초과)	CSMS는 트랜잭션의 transactionId와 함께 RequestStopTransactionRequest를 충전기로 전송해야 합니다.	
F03.FR.02	F03.FR.01 AND TxStopPoint 구성으로 인해 트랜잭션이 종료되지 않음(예: TxStopPoint가 `Authorized` 또는 `PowerPathClosed` 가 아님)	충전기는 에너지 제공을 중단하고 CSMS에 TransactionEventRequest(*eventType* = `Updated`, *triggerReason* = `RemoteStop`)를 보내야 합니다.	예를 들어 TxStopPoint = `EVConnected` 인 경우 EV가 연결 해제될 때까지 트랜잭션이 종료되지 않습니다.

ID	전제 조건	요구 사항 정의	참고
F03.FR.03	F03.FR.01 AND TxStopPoint 구성으로 인해 트랜잭션이 종료됩니다(예: TxStopPoint가 `Authorized` 또는 `PowerPathClosed`)	충전기는 TransactionEventRequest (*eventType* = `Ended`, *triggerReason* = `RemoteStop`, *stoppedReason* = `Remote`)를 CSMS로 보내야 합니다.	
F03.FR.04	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Ended), 다음을 참조하세요: Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Ended)의 선택적 meterValue 필드에 추가하여 트랜잭션 사용에 대한 자세한 정보를 제공해야 합니다.	
F03.FR.05	F03.FR.04 AND 충전기의 메모리가 부족합니다.	충전기에서 미터 데이터가 삭제될 수 있습니다.	
F03.FR.06	F03.FR.05	미터 데이터를 삭제할 때 충전기는 중간 값을 먼저 삭제해야 하며(첫 번째 값, 세 번째 값, 다섯 번째 등), 목록의 시작 부분에서 값을 삭제하거나 목록에 값을 추가하는 것을 중단해서는 안 됩니다.	
F03.FR.07	충전기가 RequestStopTransactionRequest를 수신하는 경우	그리고 TransactionId가 활성 트랜잭션와 일치할 수 있는 경우; 충전기는 RequestStopTransactionResponse로 응답해야 하며, 상태는 *Accepted* 로 설정해야 합니다.	
F03.FR.08	충전기가 RequestStopTransactionRequest를 수신하는 경우	그리고 TransactionId가 활성 트랜잭션와 일치하지 않는 경우; 충전기는 RequestStopTransactionResponse로 응답해야 하며, 상태는 *Rejected* 로 설정해야 합니다.	
F03.FR.09	TransactionEventRequest를 보낼 때	충전기는 triggerReason을 설정하여 CSMS에 이벤트를 트리거한 원인을 알려야 합니다. 사용해야할 이유는 TriggerReasonEnumType에 설명되어 있습니다.	

F04 - Remote Stop ISO 15118 Charging from CSMS

Table 132. F04 - CSMS에서 인터럽트가 있는 충전 루프

번호	유형	설명
1	**이름**	CSMS에서 원격 정지 ISO 15118 충전
2	**ID**	F04
	기능 블록	F. 원격 제어
	참조	ISO15118-1 F2 SECC에서 인터럽트가 있는 충전 루프
3	**목표**	ISO15118-1, 사용 사례 목표 F2, 38페이지를 참조하세요.
4	**설명**	ISO15118-1, 사용 사례 설명 F2, 38페이지를 참조하세요.
	액터	EV, EVSE, 충전기
6	**필수 조건**	- 기능 블록 C의 사용 사례에 따른 권한이 적용되는 경우, 성공적으로 완료되어야 합니다. ISO15118-1, 사용 사례 필수 조건 F2, 38페이지를 참조하세요.
7	**결합된 시나리오 설명**	**OCPP:** **1.** CSMS가 충전기에 RequestStopTransactionRequest를 보냅니다. **2.** 충전기는 RequestStopTransactionResponse로 응답합니다. **ISO 15118:** **3.** EV는 ChargingStatus(AC 충전의 경우) 또는 CurrentDemandReq(DC 충전의 경우) PDU를 충전기로 보냅니다. **4.** 충전기는 EVSENotification = StopCharging으로 응답합니다.
8	**사후 조건**	ISO15118-1 참조, 사용 사례 종료 조건 F2, 38페이지.

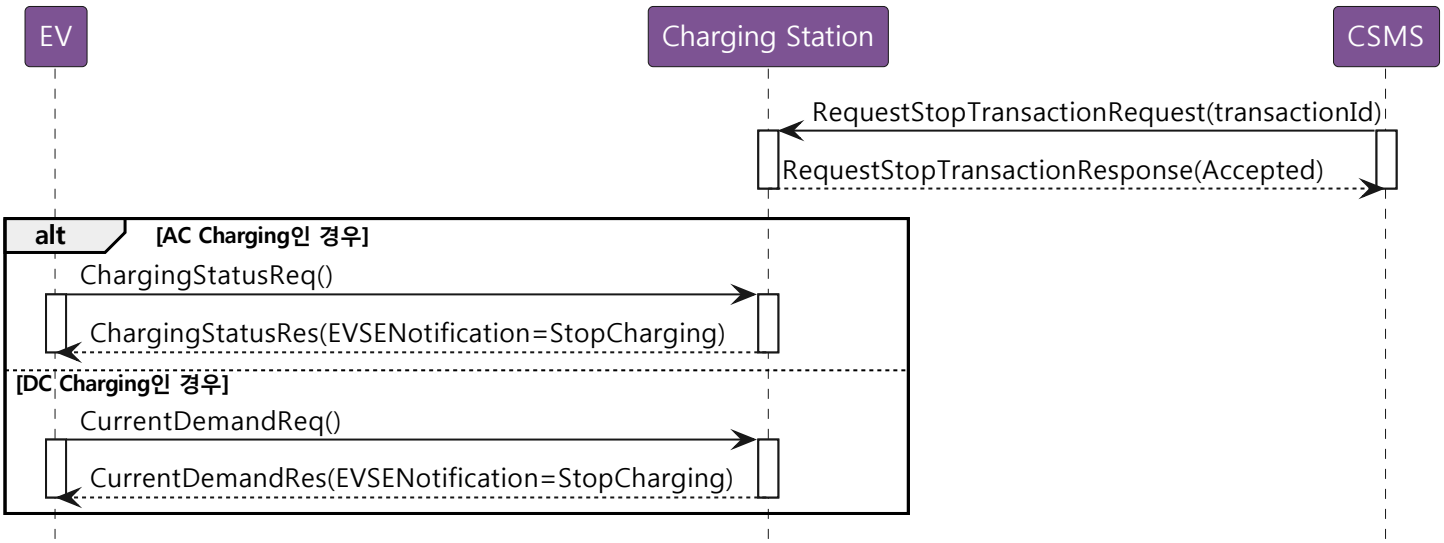

Figure 69. 충전기에서 인터럽트가 있는 충전 루프

9	**오류 처리**	n/a
10	**비고**	n/a

F04 - Remote Stop ISO 15118 Charging from CSMS - Requirements

이러한 요구 사항은 규범적입니다.

Table 133. F04 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
F04.FR.01	CSMS가 원격 중지 트랜잭션 트리거를 수신하는 경우(예: 스마트폰 앱을 사용하여 종료하는 경우, (비지역) 선불 크레딧 초과)	CSMS는 트랜잭션의 transactionId와 함께 RequestStopTransactionRequest를 충전기로 보내야 합니다.	
F04.FR.02	F04.FR.01	충전기는 에너지 제공을 중지하고, 케이블을 잠금 해제하고, TransactionEventRequest (eventType = Updated)를 CSMS로 보내야 합니다.	영구적으로 연결되지 않은 경우 케이블 잠금 해제됩니다.
F04.FR.03	F04.FR.02 AND EV 운전자가 케이블을 뽑을 때.	충전기는 TransactionEventRequest (eventType = *Ended*, stoppedReason = *Remote*)를 CSMS로 보내야 합니다.	
F04.FR.04	TransactionEventRequest에서 미터 데이터를 보내도록 구성된 경우(eventType = Ended), 다음을 참조하세요: Meter Values - Configuration	충전기는 구성된 측정값을 CSMS로 전송된 TransactionEventRequest(eventType = Ended)의 선택적 MeterValue 필드에 추가하여 트랜잭션 사용에 대한 자세한 정보를 제공해야 합니다.	
F04.FR.05	F04.FR.04 AND 충전기의 메모리가 부족합니다.	충전기에서 미터 데이터가 삭제될 수 있습니다.	
F04.FR.06	F04.FR.05	미터 데이터를 삭제할 때 충전기는 중간 값을 먼저 삭제해야 합니다(첫 번째 값, 세 번째 값, 다섯 번째 등). 목록의 시작 부분에서 값을 삭제하거나 목록에 값을 추가하는 것을 중단해서는 안 됩니다.	

2.2. Unlock Connector

F05 - Remotely Unlock Connector

Table 134. F05 - Remotely Unlock Connector

번호	유형	설명
1	**이름**	커넥터 원격 잠금 해제
2	**ID**	F05
	기능 블록	F. 원격 제어
3	**목표**	트랜잭션이 끝난 후 잠금이 실패하여 충전 케이블을 분리하는 데 문제가 있는 EV 운전자를 돕기 위해 CSO를 활성화합니다.
4	**설명**	때때로 충전기 소켓의 커넥터가 올바르게 잠금 해제되지 않는 경우가 있습니다. 이것은 충전 케이블에 장력이 있을 때 대부분 발생합니다. 즉, 운전자가 충전기에서 충전 케이블을 분리할 수 없습니다. 운전자를 돕기 위해 CSO는 충전기에 UnlockConnectorRequest를 보낼 수 있습니다. 그러면 충전기가 커넥터 잠금을 다시 시도합니다.
	Actors	충전기, CSMS, 외부 트리거
	시나리오 설명	**1.** 외부 트리거(아마도 CSO)가 충전기의 특정 커넥터 잠금 해제를 요청합니다. **2.** CSMS가 충전기에 UnlockConnectorRequest를 보냅니다. **3.** UnlockConnectorRequest를 수신하면 충전기는 UnlockConnectorResponse로 응답합니다. **4.** 응답 메시지는 충전기가 커넥터를 잠금 해제할 수 있었는지 여부를 나타냅니다.
5	**필수 조건**	지정된 커넥터에서 진행 중인 트랜잭션 없음 충전기의 커넥터가 잠긴 상태입니다.
6	**사후 조건**	충전기가 커넥터를 잠금 해제할 수 있었습니다.

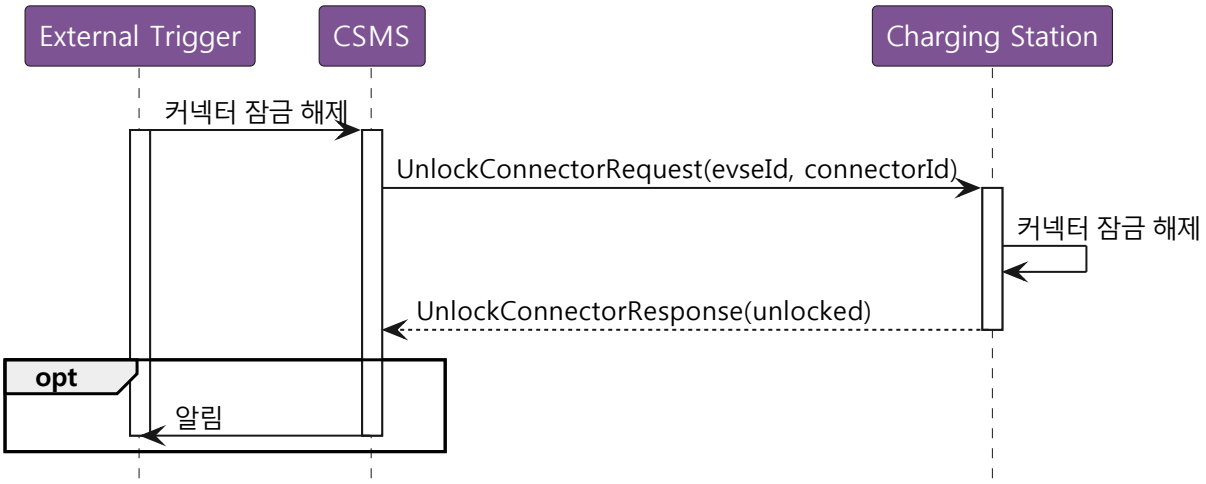

Figure 70. 시퀀스 다이어그램: 커넥터 잠금 해제

7	**오류 처리**	n/a

8	참고	잠금 해제 명령을 트리거하는 외부 트리거는 예를 들어 충전기 운영자 또는 EV 드라이버 앱일 수 있습니다. UnlockConnectorRequest는 커넥터 액세스 도어를 잠금 해제하는 것이 아니라 커넥터의 케이블 고정 잠금을 잠금 해제하기 위한 것입니다.

F05 - Remotely Unlock Connector - Requirements

Table 135. F05 - 요구 사항

ID	전제 조건	요구 사항 정의
F05.FR.01	UnlockConnectorRequest를 수신하면.	충전기는 UnlockConnectorResponse로 응답해야 합니다.
F05.FR.02	F05.FR.01 AND 지정된 커넥터에서 진행 중인 인증된 트랜잭션이 있습니다.	충전기는 커넥터 잠금을 해제하거나 트랜잭션을 중지하려고 시도해서는 안 되지만 UnlockConnectorResponse에서 `OngoingAuthorizedTransaction` 상태를 사용해야 합니다.
F05.FR.03	F05.FR.01 AND 지정된 커넥터를 알 수 없음.	충전기는 UnlockConnectorResponse에서 UnknownConnector 상태를 사용해야 합니다.
F05.FR.04	F05.FR.01 AND 충전기는 지정된 커넥터를 잠금 해제할 수 있었습니다.	충전기는 UnlockConnectorResponse에서 Unlocked 상태를 사용해야 합니다.
F05.FR.05	F05.FR.01 AND 충전기는 지정된 커넥터를 잠금 해제할 수 없었습니다.	충전기는 UnlockConnectorResponse에서 UnlockFailed 상태를 사용해야 합니다.
F05.FR.06	F05.FR.01 AND 커넥터에 케이블이 연결되지 않았습니다.	충전기는 케이블이 감지되지 않더라도 커넥터 잠금을 해제하려고 시도하고 잠금 해제 시도의 결과를 반환해야 합니다.

2.3. Remote Trigger

F06 - Trigger Message

Table 136. F06 - Trigger Message

번호	유형	설명
1	**이름**	트리거 메시지
2	**ID**	F06
	기능 블록	F. 원격 제어
3	**목표**	CSMS가 충전기에 충전기에서 시작한 메시지를 보내도록 요청할 수 있도록 합니다.
4	**설명**	이 사용 사례는 TriggerMessageRequest 메시지의 사용을 설명합니다. CSMS가 충전기에 충전기에서 시작한 메시지를 보내도록 요청하는 방법입니다. 요청에서 CSMS는 수신하고자 하는 메시지를 표시합니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS가 충전기에 TriggerMessageRequest를 보냅니다. **2.** 충전기는 TriggerMessageResponse로 응답하여 *Accepted, Rejected* 또는 *NotImplemented* 를 반환하여 보낼지 여부를 나타냅니다. **3.** CSMS에서 요청하였고 충전기가 *Accepted* 로 표시한 메시지가 전송됩니다.
5	**필수 조건**	기능 블록 *Remote Trigger* 가 설치되어 있습니다.
6	**사후 조건**	**성공 사후 조건:** **1.** CSMS가 TriggerMessageResponse 메시지를 ***성공적으로*** 수신했습니다. **2.** CSMS는 상태가 *Accepted* 인 TriggerMessageResponse 메시지를 ***성공적으로*** 수신했으며 요청된 메시지를 ***성공적으로*** 수신했습니다. **실패 사후 조건:** **1.** CSMS는 TriggerMessageResponse 메시지를 수신하지 않았습니다. **2.** CSMS는 상태가 *Accepted* 인 TriggerMessageResponse 메시지를 ***성공적으로*** 수신했으며 요청된 메시지를 수신하지 않았습니다.

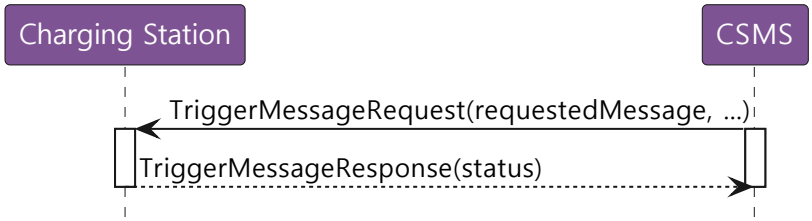

Figure 71. 시퀀스 다이어그램: 트리거 메시지

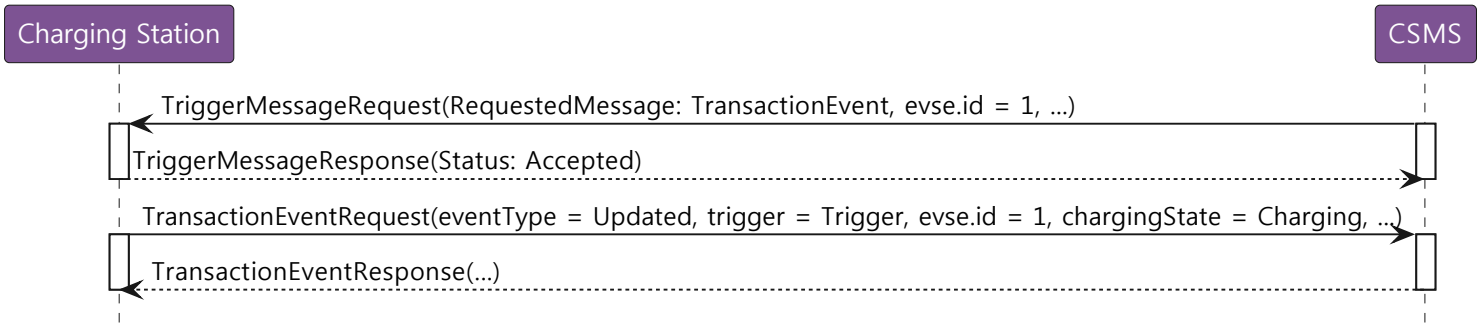

Figure 72. 시퀀스 다이어그램: 트리거 메시지 예

7	**오류 처리**	n/a
8	**참고**	TriggerMessage 메커니즘은 과거 데이터를 검색하도록 의도되지 않았습니다.

F06 - Trigger Message - Requirements

Table 137. F06 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
F06.FR.01		TriggerMessageRequest 메시지에서 CSMS는 수신하고자 하는 메시지를 표시해야 합니다.	
F06.FR.02	F06.FR.01. 이러한 모든 요청 메시지에 대해.	CSMS는 이 요청이 적용되는 EVSE를 표시할 수 있습니다.	
F06.FR.03	F06.FR.02	요청된 메시지가 선행되어야 합니다. 지정된 evseId가 메시지와 관련이 없는 경우 무시해야 합니다. 이러한 경우 요청된 메시지는 여전히 전송해야 합니다.	
F06.FR.04	충전기가 TriggerMessageRequest를 수신하는 경우.	충전기는 요청된 메시지를 보내기 전에 먼저 TriggerMessage 응답을 보내야 합니다.	
F06.FR.05	F06.FR.04	TriggerMessageResponse에서 충전기는 *Accepted* 또는 *Rejected* 를 반환하여 요청된 메시지를 보낼지 여부를 표시해야 합니다.	충전기가 전송 요청을 수락하거나 거부하는 것은 충전기에 달려 있습니다.
F06.FR.06	충전기가 *requestedMessage* 가 *MeterValues* 로 설정된 TriggerMessageRequest를 수락하는 경우	충전기는 구성 변수 `AlignedDataMeasurands`에 구성된 모든 측정량에 대한 최신 측정값과 함께 MeterValuesRequest를 CSMS로 전송해야 합니다.	
F06.FR.07	충전기가 *requestedMessage* 가 *TransactionEvent* 로 설정된 TriggerMessageRequest를 수락하는 경우	충전기는 *triggerReason* = *Trigger*, *transactionInfo*(최소 *chargingState*, *meterValue*)와 구성 변수 `SampledDataTxUpdatedMeasurands`에 구성된 모든 측정량에 대한 최신 측정값을 포함하는 TransactionEventRequest를 CSMS로 보내야 합니다.	

ID	전제 조건	요구 사항 정의	참고
F06.FR.08	충전기가 구현되지 않은 요청된 메시지를 포함하는 TriggerMessageRequest를 수신하는 경우	충전기는 TriggerMessageResponse로 *NotImplemented* 상태로 응답해야 합니다.	
F06.FR.09		트리거하는 메시지는 현재 정보만 제공해야 합니다.	
F06.FR.10		충전기가 *Accepted* 로 표시한 메시지는 전송해야 합니다.	예를 들어, 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.
F06.FR.11	필드 evse가 관련이 있지만 TriggerMessageRequest에 없는 경우.	충전기는 이를 "허용된 모든 evse 값에 대해"로 해석해야 합니다.	StatusNotifications는 특정 커넥터에 대해서만 요청할 수 있습니다. F06.FR.12/13 참조
F06.FR.12	충전기가 TriggerMessageRequest를 수신하고 *requestedMessage* 가 *StatusNotification* 으로 설정된 경우 AND (*evse* 가 생략되거나 *evse.connectorId* 가 생략됨)	충전기는 TriggerMessageResponse를 *Rejected* 상태로 응답할 수 있습니다.	StatusNotification 메시지는 커넥터 수준에서만 요청할 수 있습니다.
F06.FR.13	TriggerMessageRequest를 *requestedMessage* 를 *StatusNotification* 으로 설정하여 보낼 때	CSMS는 connectorId 필드를 설정해야 합니다.	StatusNotification 메시지는 커넥터 수준에서만 보낼 수 있습니다.
F06.FR.14	충전기가 TriggerMessageRequest를 수신하고 *requestedMessage* 를 *LogStatusNotification* 으로 설정하고 충전기가 로그 파일을 업로드하는 경우	충전기는 status *Uploading* 으로 LogStatusNotificationRequest를 CSMS로 보내야 합니다.	
F06.FR.15	충전기가 TriggerMessageRequest를 수신하고 *requestedMessage* 가 *LogStatusNotification* 로 설정된 경우 AND 충전기가 로그 파일을 업로드하지 않는 경우	충전기는 status *Idle* 로 LogStatusNotificationRequest를 CSMS로 보내야 합니다.	
F06.FR.16	충전기가 TriggerMessageRequest를 수신하고 *requestedMessage* 가 *FirmwareStatusNotification* 로 설정된 경우 AND 충전기가 펌웨어 업데이트 관련 작업을 수행하지 않는 경우	충전기는 status *Idle* 로 CSMS에 FirmwareStatusNotificationRequest를 보내야 합니다.	
F06.FR.17	충전기가 TriggerMessageRequest를 수신하고 *requestedMessage* 가 *BootNotification* 로 설정되고 CSMS에서 마지막 BootNotificationRequest에 대한 응답이 *Accepted* 인 경우	충전기는 TriggerMessageResponse를 status *Rejected* 로 응답해야 합니다.	충전기에 BootNotification을 보내도록 요청하는 트리거는 BootNotification이 아직 수락되지 않은 경우에만 사용하도록 의도되었습니다.

G. Availability

1. Introduction

이 기능 블록은 충전기가 CSMS에 새 트랜잭션을 시작하기 위한 현재 가용성을 알리는 방법을 지정합니다.

CSO의 경우 새 EV를 충전할 수 있는 충전기가 있는지 아는 것이 중요합니다. CSO는 이 정보를 확인하고 EV 운전자에게 충전기가 사용 가능한지 알려줍니다. 이를 알기 위해 충전기는 자체 또는 EVSE 중 하나의 상태 변경 사항을 CSMS로 전송해야 합니다. 예를 보려면 B04 - Offline Behavior Idle Charging Station을 참조하세요.

CSO의 경우 트랜잭션 상태를 아는 것이 매우 유용하므로 충전기는 자세한 상태를 CSMS로 전송할 수 있습니다. 이는 충전 중에 문제가 발생하는 EV 운전자를 도울 때 매우 유용할 수 있습니다.

충전기에서 오류가 감지되면 CSMS에 오류에 대한 메시지를 보낼 수 있습니다.

CSO가 충전기에서 더 이상 새 트랜잭션을 시작하지 않기를 원하는 경우 가용성을 변경할 수 있습니다. 예를 들어, 충전기에서 유지관리를 해야 하기 때문에 충전기를 사용하지 않으려고 합니다.

CSO는 하나 이상의 EVSE에 대한 가용성을 변경할 수도 있습니다. 예를 들어, 고객이 전화를 걸어 충전기의 EVSE가 고장났다고 불평합니다. 그런 다음 CSO는 커넥터를 사용할 수 없게 설정하여 EV 운전자가 해당 커넥터를 사용할 수 없게 할 수 있습니다.

물론 CSMS의 명령으로 충전기나 커넥터를 다시 사용할 수 있게 할 수도 있습니다.

NOTE 커넥터 상태 개요는 ConnectorStatusEnumType에서 확인할 수 있습니다.

2. Use cases & Requirements

G01 - Status Notification

Table 138. G01 - 상태 알림

번호	유형	설명
1	**이름**	상태 알림
2	**ID**	G01
	기능 블록	G. 가용성
3	**목표**	커넥터 상태 변경을 CSMS에 알리는 것.
4	**설명**	이 사용 사례는 충전기가 CSMS에 알림을 보내 커넥터 상태 변경을 CSMS에 알리는 기능을 다룹니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 커넥터 상태가 변경되어 충전기에서 StatusNotificationRequest를 CSMS로 보내 CSMS에 새 상태를 알립니다. **2.** CSMS에서 StatusNotificationResponse로 충전기에 응답합니다.
	대체 시나리오	**1.** StatusNotificationRequest 대신 충전기는 *component.name* "Connector"에 대한 *trigger* = `Delta` 와 *evse.id* 의 EVSE 번호, *evse.connectorId* 의 커넥터 번호, *variable* = "AvailabilityState"와 새 상태 값을 CSMS에 포함하는 NotifyEventRequest를 보낼 수 있습니다. **1a.** 선택적으로 충전기는 NotifyEventRequest에 *component* = "ChargingStation"과 *component* = "EVSE"의 상태를 포함할 수도 있습니다. **2.** CSMS는 충전기에 NotifyEventResponse로 응답합니다.
5	**필수 조건**	n/a
6	**사후 조건**	**성공한 사후 조건:** CSMS는 상태 변경에 대해 *성공적으로* 통보받았습니다. **실패한 사후 조건:** n/a

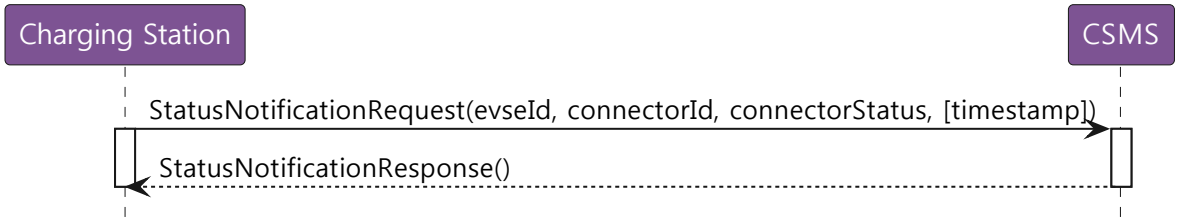

Figure 73. 시퀀스 다이어그램: 상태 알림

7	**오류 처리**	n/a

8	참고	충전기는 *Unavailable* 상태를 다른 목적으로 내부적으로 사용할 수 있습니다(예: 펌웨어 업데이트 중 또는 초기 *Accepted* RegistrationStatus를 기다리는 동안). EVSE의 커넥터 중 하나가 예약/점유 중이면 CSMS는 다른 시스템이나 사용자에게 가용성 정보를 제공할 때 다른 커넥터의 상태를 처리해야 합니다. CSMS는 어떤 커넥터가 동일한 EVSE에 속하는지 알고 있습니다. 충전기에서 CSMS로 커넥터 상태를 알리는 작업은 새로운 장치 관리 모니터링 기능이 담당하지만 이 메커니즘은 아직 현장에서 검증되지 않았습니다. 따라서 이전의 StatusNotificationRequest 메시지는 현재 사용할 수 있습니다.

G01 - Status Notification - State transition overview for connecting/disconnecting

초기	케이블 플러그인	케이블 언플러그
사용 가능	→ 점유	-
점유	-	→ 사용 가능 (→ 사용 불가능, 사용 불가능으로 예약된 경우)
예약됨	- (→ 점유, 예약된 IdToken에 대해 권한이 부여된 경우에만)	-
사용 불가능	-	-
오류 발생	-	-

G01 - Status Notification - Requirements

Table 139. G01 - 요구 사항

ID	전제 조건	요구 조건 정의
G01.FR.01		충전기 커넥터는 반드시 ConnectorStatus 열거형의 유효한 상태 중 하나를 가져야 합니다.
G01.FR.02	ChangeAvailabilityRequest 메시지에 의해 EVSE의 상태가 *Unavailable* 로 설정되었을 때	EVSE의 *Unavailable* 상태는 재부팅 후에도 지속되어야 합니다.
G01.FR.03	EV가 연결 중일 때 커넥터 상태는 `Available` 입니다.	충전기는 *connectorStatus* = `Occupied` 상태로 StatusNotificationRequest를 전송하거나, *component* = "Connector", *variable* = "AvailabilityState", *actualValue* = "Occupied" 및 *trigger* = "Delta"로 설정된 NotifyEventRequest를 전송해야 합니다.
G01.FR.04	EV가 연결을 끊고 있고 커넥터가 `Unavailable` 상태가 되도록 예약되지 않은 경우 (G03.FR.05)	EV가 연결을 끊었을 때 충전기는 *connectorStatus* = `Available` 상태로 StatusNotificationRequest를 전송하거나, *component* = "Connector", *variable* = "AvailabilityState", *actualValue* = "Available" 및 *trigger* = "Delta"로 설정된 NotifyEventRequest를 전송해야 합니다.

ID	전제 조건	요구 조건 정의
G01.FR.05	EV가 연결을 끊고 있으며 커넥터가 `Unavailable` 상태가 되도록 예약된 경우 (G03.FR.05)	EV가 연결을 끊었을 때 충전기는 *connectorStatus* = `Unavailable` 상태로 StatusNotificationRequest를 전송하거나, *component* = "Connector", *variable* = "AvailabilityState", *actualValue* = "Unavailable" 및 *trigger* = "Delta"로 설정된 NotifyEventRequest를 전송해야 합니다.
G01.FR.06	EV가 연결 중이고 EV 드라이버가 예약과 일치하는 IdToken을 제시한 경우	충전기는 *connectorStatus* = `Occupied` 상태로 StatusNotificationRequest를 전송하거나, *component* = "Connector", *variable* = "AvailabilityState", *actualValue* = "Occupied" 및 *trigger* = "Delta"로 설정된 NotifyEventRequest를 전송해야 합니다.
G01.FR.07	ChangeAvailabilityRequest에 의해 커넥터 상태가 변경되었을 때	충전기는 해당 *connectorStatus_로 StatusNotificationRequest를 전송하거나*, *_component* = "Connector", *variable* = "AvailabilityState", *trigger* = "Delta" 및 해당 _actualValue_로 설정된 NotifyEventRequest를 전송해야 합니다.
G01.FR.08	케이블이 EVSE의 커넥터에 연결되어 있고 EVSE에 다수의 커넥터가 있는 경우	충전기는 다른 커넥터가 더 이상 사용 불가능하더라도, 해당 커넥터에 대해 StatusNotificationRequest를 전송하지 않아야 합니다.

G02 - Heartbeat

Table 140. G02 - 하트비트

번호	유형	설명
1	**이름**	하트비트
2	**ID**	G02
	기능 블록	G. 가용성
3	**목표**	충전기가 여전히 연결되어 있음을 CSMS에 알리기 위해, 선택적으로 하트비트를 시간 동기화에 사용할 수 있습니다.
4	**설명**	이 사용 사례는 충전기가 여전히 연결되어 있음을 CSMS에 알리는 방법을 설명합니다. 충전기는 구성 가능한 시간 간격 후에 하트비트를 보냅니다. 구성에 따라 하트비트를 시간 동기화에 사용할 수 있습니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 일정 시간 동안 활동이 없으면 충전기는 HeartbeatRequest를 보내 CSMS가 충전기가 여전히 살아 있음을 알 수 있도록 합니다. **2.** HeartbeatRequest를 수신하면 CSMS는 HeartbeatResponse로 응답합니다. 응답 메시지에는 CSMS의 현재 시간이 포함되어 있으며, 충전기는 이를 사용하여 내부 시계를 동기화할 수 있습니다.
5	**필수 조건**	하트비트 간격이 설정됩니다.
6	**사후 조건**	**성공한 사후 조건:** CSMS는 충전기가 여전히 연결되어 있음을 알고 있습니다. **실패한 사후 조건:** CSMS는 충전기가 *Offline* 이라고 결론 내립니다.

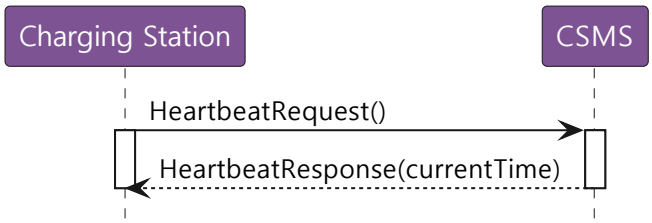

Figure 74. 시퀀스 다이어그램: 하트비트

7	**오류 처리**	n/a
8	**참고**	웹소켓을 통한 JSON의 경우, 하트비트를 보내는 것은 웹소켓을 유지하는 데 ***중요하지 않습니다***. 웹소켓은 이미 이를 위한 메커니즘을 제공하기 때문입니다. 그러나 충전기가 시간 동기화를 위해 하트비트를 사용하는 경우, 적어도 24시간에 한 번은 하트비트를 보내는 것이 좋습니다.

G02 - Heartbeat - Requirements

Table 141. G02 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
G02.FR.01	CSMS가 BootNotificationResponse로 응답하고 상태가 *Accepted* 인 경우.	충전기는 응답 메시지의 간격에 따라 하트비트 간격을 조정해야 합니다.	
G02.FR.02		충전기는 구성 가능한 시간 간격 후에 HeartbeatRequest를 보내야 합니다.	충전기가 여전히 온라인 상태인 것을 CSMS가 알 수 있도록 합니다.
G02.FR.03		HeartbeatResponse 메시지에는 CSMS의 현재 시간이 포함되어야 합니다.	
G02.FR.04	충전기에서 메시지를 수신할 때마다.	CSMS는 해당 충전기의 가용성을 가정해야 합니다.	
G02.FR.05		다른 메시지가 CSMS로 전송되면 충전기가 하트비트 간격 타이머를 재설정하는 것이 좋습니다.	
G02.FR.06	충전기가 HeartbeatResponse를 수신할 때.	충전기가 내부 시계를 동기화하기 위해 현재 시간을 사용하는 것이 좋습니다.	
G02.FR.07	메시지가 지속적으로 전송되어 하트비트 간격 타이머가 지속적으로 재설정되고 HeartbeatRequest가 시간 동기화에 사용되는 경우	충전기가 시계를 동기화하기 위해 최소 24시간마다 HeartbeatRequest를 보내는 것이 좋습니다.	

G03 - Change Availability EVSE/Connector

Table 142. G03 - 가용성 EVSE/커넥터 변경

번호	유형	설명
1	**이름**	가용성 EVSE/커넥터 변경
2	**ID**	G03
	기능 블록	G. 가용성
3	**목표**	CSMS가 EVSE 또는 커넥터의 가용성을 작동 가능 또는 _작동 불가능_으로 변경할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSMS가 충전기에 EVSE 또는 커넥터 중 하나의 가용성을 *Operative* 또는 *Inoperative* 으로 변경하도록 요청하는 방법을 다룹니다. EVSE/커넥터는 *Faulted* 및 *Unavailable* 이 아닌 모든 상태에서 *Operative* 로 간주됩니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS가 ChangeAvailabilityRequest를 보내 충전기에 EVSE 또는 커넥터의 가용성을 변경하도록 요청합니다. **2.** 충전기가 EVSE/커넥터의 가용성을 ChangeAvailabilityRequest에서 요청한 operationalStatus로 변경합니다. **3**. ChangeAvailabilityRequest를 수신하면 충전기가 ChangeAvailabilityResponse로 응답합니다. ChangeAvailabilityResponse에서 'Scheduled' 상태가 보고된 경우 트랜잭션이 실행 중이었으며 이 트랜잭션이 먼저 완료됩니다. **4**. 충전기는 StatusNotification을 사용하여 EVSE/커넥터의 상태를 보고합니다.
	대체 시나리오	G04 - Change Availability Charging Station
5	**필수 조건**	n/a
6	**사후 조건**	**성공한 사후 조건:** EVSE/커넥터의 가용성을 *Operative* 으로 변경할 때 EVSE의 상태가 *Available*, *Occupied* 또는 *Reserved* 로 변경되었습니다. EVSE/커넥터의 가용성을 *Inoperative* 로 변경할 때 EVSE의 상태가 *Unavailable* 로 변경되었습니다. **실패 사후 조건:** EVSE의 상태는 충전기가 ChangeAvailabilityRequest를 수신하기 직전의 상태이며 요청된 가용성 변경에 따른 것이 아닙니다.

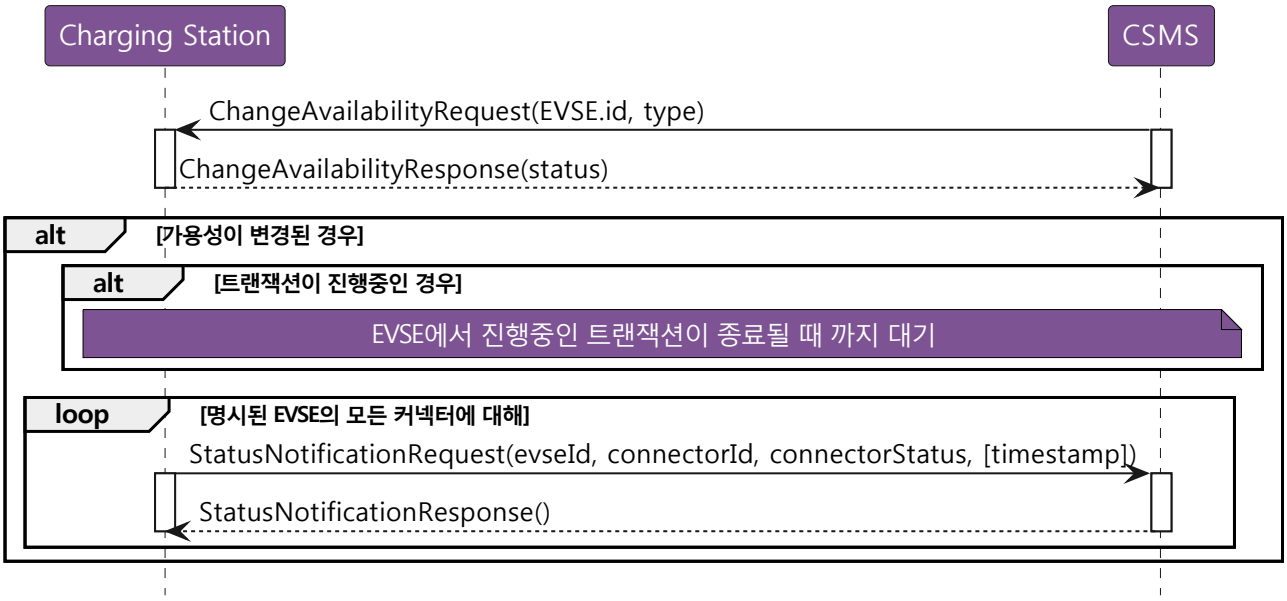

Figure 75. 시퀀스 다이어그램: 가용성 변경

7	**오류 처리**	n/a
8	**참고**	영구 상태, 예: EVSE가 *Available* 로 설정되면 재부팅 이후에도 지속되어야 합니다.

G03 - Change Availability EVSE - Requirements

Table 143. G03 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
G03.FR.01	ChangeAvailabilityRequest를 수신하면.	충전기는 ChangeAvailabilityResponse로 응답해야 합니다.	
G03.FR.02	G03.FR.01	이 응답 메시지는 충전기가 요청된 가용성으로 변경할 수 있는지 여부를 나타냅니다.	
G03.FR.03	CSMS가 충전기에 EVSE 또는 커넥터를 이미 있는 상태로 변경하도록 요청하는 경우.	충전기는 가용성 상태 *Accepted* 로 응답해야 합니다.	
G03.FR.04	ChangeAvailabilityRequest를 사용한 가용성 변경 요청이 커넥터 상태를 변경한 경우.	충전기는 StatusNotificationRequest를 사용하여 CSMS에 새 커넥터 가용성 상태를 알려야 합니다.	ChangeAvailabilityStatusEnumType에 설명된 대로
G03.FR.05	트랜잭션이 진행 중이고 G03.FR.03이 아닌 경우	충전기는 트랜잭션이 완료된 후에 발생하도록 예약되었음을 나타내기 위해 가용성 상태 *Scheduled* 로 응답해야 합니다.	
G03.FR.06	EVSE의 가용성이 Inoperative(*Unavailable*, *Faulted*)가 되는 경우	해당 EVSE의 모든 동작하는 커넥터(즉, *Faulted* 가 아닌)는 *Unavailable* 이 됩니 다.	
G03.FR.07	EVSE의 가용성이 Operative가 되는 경우	충전기는 해당 EVSE의 모든 커넥터 상태를 원래 상태로 되돌려야 합니다.	NOTE 1 참조.
G03.FR.08	ChangeAvailabilityRequest를 통해 EVSE 또는 커넥터의 가용성이 명시적으로 설정된 경우	설정된 가용성 상태는 재부팅/전원 손실 시에도 지속되어야 합니다.	

ID	전제 조건	요구 사항 정의	참고
G03.FR.09	EV가 연결되고 EV 운전자가 예약과 일치하는 IdToken을 제시하지 않으면 커넥터가 예약됩니다.	커넥터 상태는 변경되지 않아야 합니다.	예약과 일치하는 IdToken이 제시되거나 예약이 만료될 때까지 커넥터는 예약된 상태로 유지됩니다.

NOTE	1. 충전기, EVSE 및 커넥터는 별도의 개별 상태를 갖습니다. 즉, (예를 들어) 커넥터를 작동 불능으로 설정한 다음 연결된 EVSE를 작동 불능으로 설정한 다음 EVSE를 다시 작동으로 변경하면 커넥터는 작동 불능 상태로 유지됩니다.

NOTE	2. 커넥터의 상태 변경을 보고하는 데만 필요합니다. StatusNotificationRequest는 커넥터 상태 보고만 지원합니다.

G04 - Change Availability Charging Station

Table 144. G04 - 가용성 변경 충전기

번호	유형	설명
1	**이름**	가용성 변경 충전기
2	**ID**	G04
	기능 블록	G. 가용성
	부모 사용 사례	G03 - Change Availability EVSE/Connector
3	**목표**	CSMS가 충전기의 가용성을 변경할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSMS가 충전기에 가용성을 변경하도록 요청하는 방법을 설명합니다. 충전기는 충전 중이거나 충전할 준비가 되면 *Operative* 하는 것으로 간주됩니다. 충전기는 충전을 허용하지 *않을 때 Inoperative* 으로 간주됩니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS가 충전기의 가용성을 변경하도록 요청하는 ChangeAvailabilityRequest를 보냅니다. **2.** ChangeAvailabilityRequest를 수신하면 충전기는 ChangeAvailabilityResponse로 응답합니다.
5	**필수 조건**	n/a
6	**사후 조건**	**성공한 사후 조건:** CSMS가 충전기의 가용성을 변경할 수 있었습니다. 충전기의 가용성을 *Operative* 으로 변경할 때 충전기의 상태가 *Available* 으로 변경되었습니다. 충전기의 가용성을 *Inoperative* 으로 변경하면 충전기의 상태가 *Unavailable* 로 변경되었습니다. **실패 사후 조건:** CSMS가 요청된 충전기의 가용성을 변경할 수 *없었습니다*.

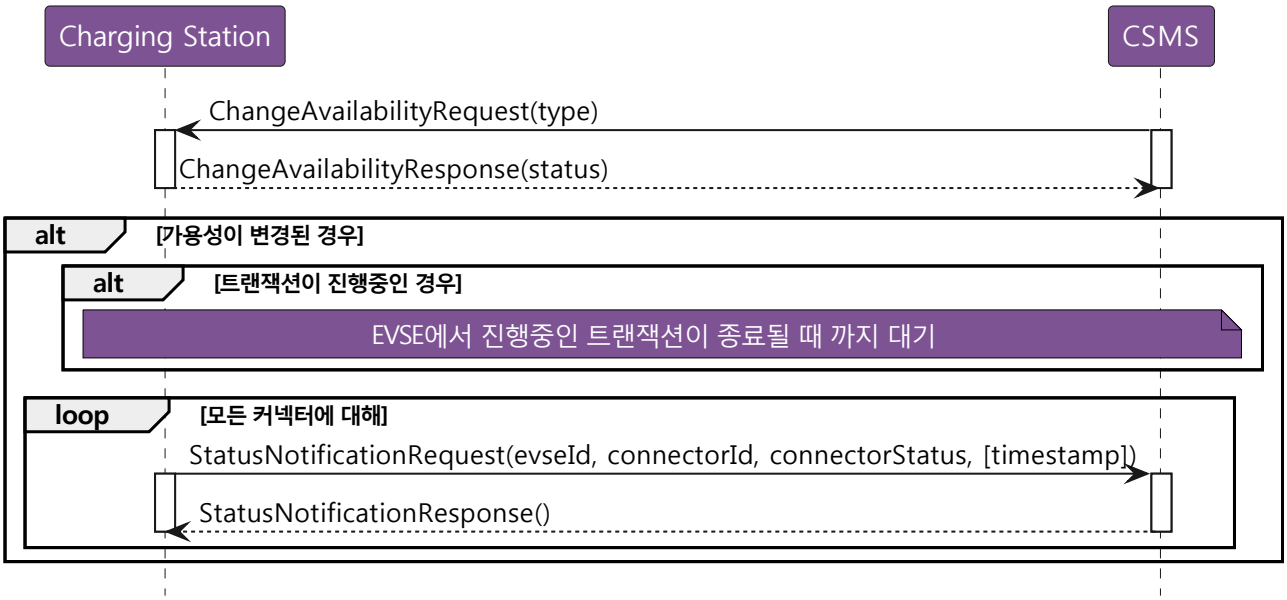

Figure 76. 시퀀스 다이어그램: 충전기 가용성 변경

7	**오류 처리**	n/a
8	**참고**	지속적인 상태: 예를 들어, 충전기가 *Unavailable* 로 설정된 경우 재부팅 이후에도 상태가 지속되어야 합니다.

G04 - Change Availability Charging Station - Requirements

Table 145. G04 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
G04.FR.01	ChangeAvailabilityRequest에서 evse 필드가 생략된 경우.	충전기 상태 변경은 전체 충전기에 적용되어야 합니다.	
G04.FR.02	ChangeAvailabilityRequest를 수신하면.	충전기는 ChangeAvailabilityResponse로 응답해야 합니다.	
G04.FR.03	G04.FR.02	이 응답 메시지는 충전기가 요청된 가용성으로 변경할 수 있는지 여부를 나타냅니다.	
G04.FR.04	CSMS가 충전기에 이미 있는 상태로 변경하도록 요청하는 경우.	충전기는 가용성 상태 *Accepted* 로 응답합니다.	
G04.FR.05	ChangeAvailabilityRequest를 사용하여 가용성 변경 요청이 발생한 경우.	충전기는 StatusNotificationRequest를 통해 변경된 각 커넥터의 상태를 전송하여 CSMS에 알립니다.	ConnectorStatusEnumType에 설명된 대로
G04.FR.06	트랜잭션이 진행 중인 경우.	충전기는 트랜잭션이 완료된 후 발생하도록 예약되었음을 나타내기 위해 가용성 상태 *Scheduled* 로 응답해야 합니다.	
G04.FR.07	충전기의 가용성이 Inoperative(*Unavailable*, *Faulted*)가 되는 경우	모든 작동 EVSE 및 커넥터(즉, *Faulted* 가 아닌)는 *Unavailable* 이 됩니다.	
G04.FR.08	충전기의 가용성이 Operative가 되는 경우	충전기는 모든 EVSE 및 커넥터의 상태를 원래 상태로 되돌려야 합니다.	NOTE 1을 참조하세요.
G04.FR.09	충전기의 가용성이 ChangeAvailabilityRequest를 통해 명시적으로 설정된 경우	설정된 가용성 상태는 재부팅/전원 손실 시에도 지속되어야 합니다.	

NOTE	1. 충전기, EVSE 및 커넥터는 별도의/개별 상태를 갖습니다. 즉, (예를 들어) 커넥터를 작동 불능으로 설정한 다음 연결된 EVSE를 작동 불능으로 설정한 다음 EVSE를 다시 작동으로 변경하면 커넥터는 작동 불능 상태로 유지됩니다.
NOTE	2. 커넥터의 상태 변경을 보고하는 데만 필요합니다. StatusNotificationRequest는 커넥터 상태 보고만 지원합니다.

G05 - Lock Failure

Table 146. G05 - 잠금 실패

번호	유형	설명
1	**이름**	잠금 실패
2	**ID**	G05
	기능 블록	G. 가용성
3	**목표**	커넥터가 제대로 잠기지 않은 상태에서 EV 드라이버가 충전되지 않도록 합니다.
4	**설명**	이 사용 사례는 커넥터가 제대로 잠기지 않은 상태에서 EV 드라이버가 충전기에서 충전 세션을 시작하지 못하도록 하는 방법을 설명합니다.
	액터	충전기, CSMS, EV 드라이버
	시나리오 설명	**1.** EV 드라이버가 충전기 및/또는 CSMS에서 승인되었습니다. **2.** 커넥터 잠금 시도가 실패합니다. **3.** ConnectorPlugRetentionLock 구성 요소에 대한 NotifyEventRequest, variable = Proble, value = *true*.
5	**필수 조건**	충전 케이블이 연결됨(status = *Occupied*) 충전기에는 해당 장치 모델에 정의된 ConnectorPlugRetentionLock 구성 요소가 있습니다. MonitoringLevel은 커넥터 잠금 이벤트 실패가 보고되는 수준으로 설정됩니다.
6	**사후 조건**	트랜잭션이 시작되지 않고 커넥터 잠금 이벤트 실패가 보고됩니다.

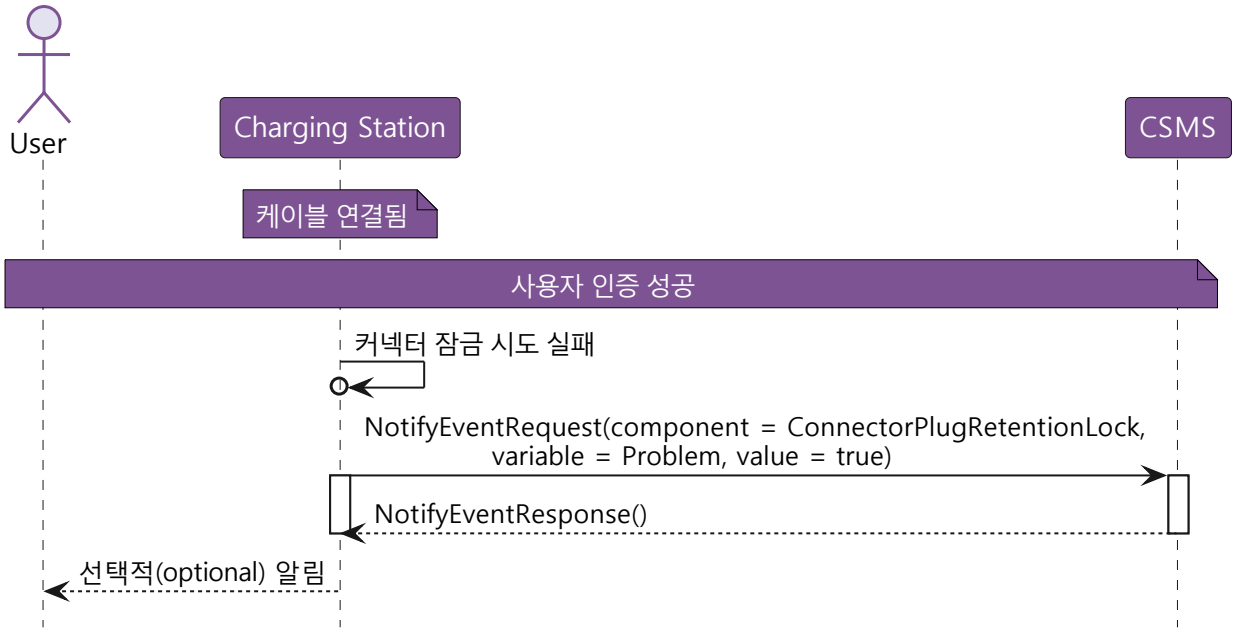

Figure 77. 시퀀스 다이어그램: 잠금 실패

7	**오류 처리**	n/a
8	**참고**	EV 드라이버에게 어떤 종류의 알림을 제공하는 것이 좋습니다("케이블을 잠글 수 없습니다").

G05 - Lock Failure - Requirements

Table 147. G05 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
G05.FR.01	커넥터 고정 잠금 장치의 잠금이 실패하는 경우.	충전기는 충전을 시작해서는 안 됩니다.	
G05.FR.02	G05.FR.01	충전기는 *ConnectorPlugRetentionLock* 구성 요소에 대한 NotifyEventRequest를 CSMS로 보내야 합니다. variable = Problem, Value = *True*.	
G05.FR.03	G05.FR.02	CSMS는 NotifyEventResponse로 응답해야 합니다.	
G05.FR.04	G05.FR.01	충전기는 EV 운전자에게 선택적 알림을 표시할 수 있습니다.	EV 운전자에게 잠금 장치 오류를 알리기 위해.

H. Reservation

1. Introduction

이 기능 블록은 OCPP의 예약 기능을 설명합니다. 예약 기능을 통해 EV 운전자는 충전기/EVSE를 예약하여 도착 시 충전기에서 사용 가능한 커넥터를 확보할 수 있습니다.

충전기가 풍부하게 제공되지 않고 EV의 주행 범위가 제한되어 EV 운전자는 충전기에서 충전기로 이동할 계획을 세웁니다. 그들은 가려고 하는 충전기를 사용할 수 있는지 확실히 알아야 합니다. 충전기로 이동하는 동안 다른 EV 운전자가 충전기를 사용하기 시작하면 그들은 좋아하지 않습니다.

EV 운전자에게는 특정 유형의 커넥터를 예약하거나 EV 운전자가 선호하지 않는 경우 충전기에서 지정되지 않은 EVSE를 예약할 수 있는 것이 유용합니다. 그러면 도착 시 충전기에서 충전할 수 있다는 것을 확실히 알 수 있습니다.

2. Use cases & Requirements

H01 - Reservation

Table 148. H01 - 예약

번호	유형	설명
1	**이름**	예약
2	**ID**	H01
	기능 블록	H. 예약
3	**목표**	EV 운전자가 충전기에서 EV를 충전할 수 있도록 하기 위해 EV 운전자는 특정 만료 시간까지 예약을 할 수 있습니다.
4	**설명**	이 사용 사례는 특정 IdTokenType에 대해 충전기를 예약하는 방법을 설명합니다.
5	*Actors*	충전기, CSMS, EV 운전자
S1	*시나리오 목표*	충전기에서 지정되지 않은 EVSE 예약
	시나리오 설명	**1.** EV 운전자가 CSMS에 충전기에서 지정되지 않은 EVSE를 예약해 달라고 요청합니다. **2.** CSMS가 *evseId* 없이 ReserveNowRequest를 충전기로 보냅니다. **3.** ReserveNowRequest를 수신하면 충전기는 ReserveNowResponse로 응답하고 상태는 *Accepted* 입니다.
	필수 조건	충전기에 사용 가능한 EVSE가 하나 이상 있음
	사후 조건	**성공 사후 조건:** 충전기에서 ReserveNowRequest를 수락했습니다. **실패 사후 조건:** 충전기에서 ReserveNowRequest를 거부했습니다.

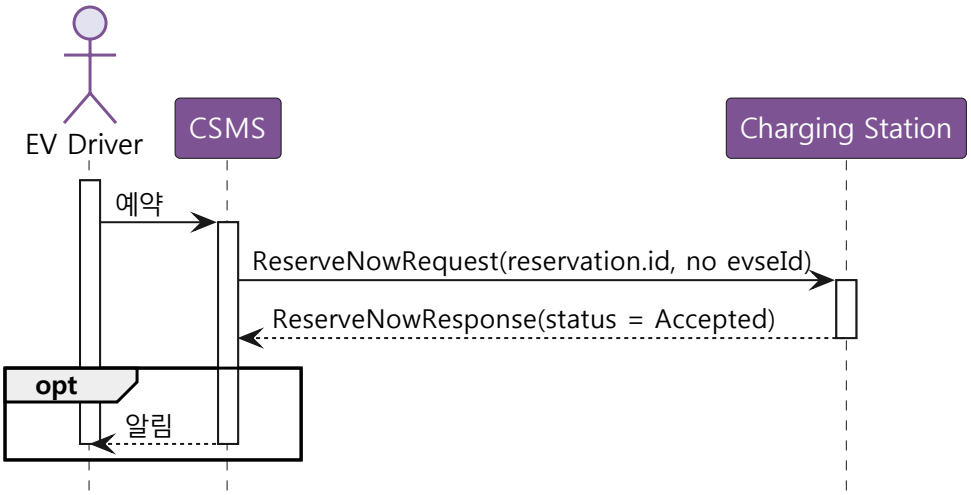

Figure 78. 시퀀스 다이어그램: S1 - 충전기에서 지정되지 않은 EVSE 예약

S2	*시나리오 목표*	충전기에서 특정 EVSE 예약

	시나리오 설명	**1.** EV 운전자가 CSMS에 충전기에서 특정 EVSE를 예약해 달라고 요청합니다. **2.** CSMS가 충전기에 EVSE와 함께 ReserveNowRequest를 보냅니다. **3.** ReserveNowRequest를 수신하면 충전기는 ReserveNowResponse를 *Accepted* 상태로 응답합니다. **4.** 충전기는 해당 EVSE의 모든 커넥터에 대해 StatusNotificationRequest를 *Reserved* 상태로 보냅니다. **5.** CSMS가 충전기에 StatusNotificationResponse를 응답합니다.
	대체 시나리오 설명	위의 1, 2, 3단계. **4.** StatusNotificationRequest 대신 충전기는 *component.name* = "Connector"에 대한 *trigger* = `Delta`, *evse.id* 의 EVSE 번호, *evse.connectorId* 의 커넥터 번호, *variable* = "AvailabilityState", *actualValue* = "Reserved"인 NotifyEventRequest를 보낼 수 있습니다. **5a.** 선택적으로 충전기는 *component* = "EVSE", *variable* = "AvailabilityState" 및 *actualValue* = "Reserved"에 대해 NotifyEventRequest를 보고할 수 있으며, 해당되는 경우 *component* = "ChargingStation"에 대해서도 이를 보고할 수 있습니다.
	Prerequisite(s)	충전기의 지정된 EVSE는 *Available* 상태입니다.
	Postcondition(s)	**성공 사후 조건:** 충전기는 ReserveNowRequest를 수락했습니다. AND StatusNotificationRequest를 *Reserved* 상태로 보냈습니다. **실패 사후 조건:** 충전기가 ReserveNowRequest를 거부했습니다. OR 충전기가 StatusNotificationRequest를 *Reserved* 상태로 보내지 않았습니다.

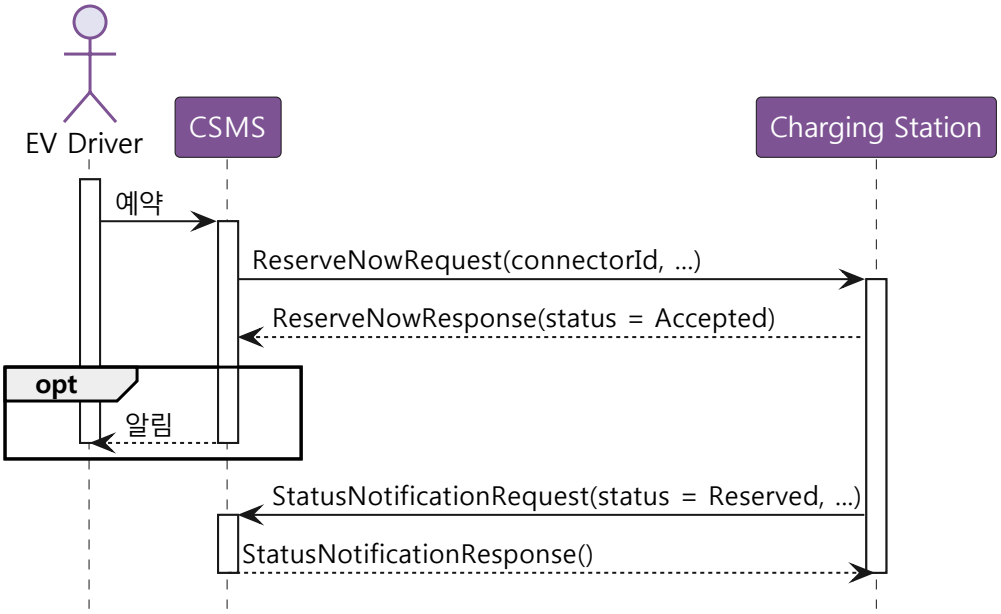

Figure 79. 시퀀스 다이어그램: S2 - 충전기에서 지정된 EVSE 예약

S3	*시나리오 목표*	충전기에서 커넥터 유형 예약
	시나리오 설명	**1.** EV 드라이버가 CSMS에 충전기에서 커넥터 유형을 예약하도록 요청합니다. **2.** CSMS가 커넥터 유형과 함께 ReserveNowRequest를 충전기로 보냅니다. **3.** ReserveNowRequest를 수신하면 충전기는 ReserveNowResponse를 *Accepted* 상태로 응답합니다.
	필수 조건	충전기에 지정된 커넥터 유형이 있는 사용 가능한 EVSE가 하나 이상 있음

사후 조건	**성공 사후 조건:** 충전기가 ReserveNowRequest를 수락했습니다. **실패 사후 조건:** 충전기가 ReserveNowRequest를 거부했습니다.

Figure 80. 시퀀스 다이어그램: S3 - 충전기에서 커넥터 유형 예약

6	**오류 처리**	n/a
7	**참고**	ReserveNowRequest를 수신한 후, 트랜잭션을 시작하기 전에 AuthorizeRequest로 식별자를 검증하는 것이 좋습니다.

H01 - Reservation - Requirements

Table 149. H01 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
H01.FR.01	충전기가 예약을 수락하지 않도록 구성된 경우.	충전기는 *Rejected* 를 반환해야 합니다.	
H01.FR.02	ReserveNowRequest의 *id* 가 충전기의 예약과 일치하는 경우.	충전기는 해당 예약을 요청의 새 예약으로 대체해야 합니다.	
H01.FR.03	ReserveNowRequest의 *id* 가 충전기의 어떤 예약과도 일치하지 않는 경우.	충전기는 EVSE 예약에 성공하면 상태 값 *Accepted* 를 반환해야 합니다.	
H01.FR.04	충전기가 *evseId* 없이 ReserveNowRequest를 수신하는 경우 AND 최소 하나의 EVSE가 *Available* 인 경우 AND H01.FR.18	충전기는 예약을 수락하고 ReserveNowResponse로 응답해야 합니다. 상태는 *Accepted* 입니다.	
H01.FR.06	충전기가 커넥터 유형이 있는 ReserveNowRequest를 수신 AND 지정된 커넥터 유형이 있는 EVSE가 하나 이상 *Available* AND H01.FR.18	충전기는 예약을 수락하고 상태가 *Accepted* 인 ReserveNowResponse로 응답해야 합니다.	
H01.FR.07	충전기가 *evseId* 없이 ReserveNowRequest를 *Accepted* 한 경우	충전기는 예약 유효 기간 동안 언제든지 예약된 IdTokenType에 대해 하나의 EVSE가 사용 가능한 상태로 유지되도록 해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
H01.FR.09	충전기가 커넥터 유형이 있는 ReserveNowRequest를 *Accepted* 한 경우	충전기는 예약의 유효 기간 동안 언제든지 지정된 유형의 커넥터 하나가 예약된 IdTokenType에 사용 가능한지 확인해야 합니다.	
H01.FR.11	ReserveNowRequest를 수신 AND (모든) 대상 EVSE가 *Reserved* 또는 *Occupied* 상태인 경우	충전기는 *Occupied* 를 반환해야 합니다.	
H01.FR.12	ReserveNowRequest를 수신하고 (모든) 대상 EVSE가 *Faulted* 상태인 경우	충전기는 *Faulted* 를 반환해야 합니다.	
H01.FR.14	ReserveNowRequest를 수신하고 (모든) 대상 EVSE가 *Unavailable* 상태인 경우	충전기는 *Unavailable* 을 반환해야 합니다.	
H01.FR.15	예약된 IdTokenType에 대한 트랜잭션이 시작된 경우.	충전기는 TransactionEventRequest에서 reservationId를 보내야 합니다.	CSMS에 예약이 종료되었음을 알리기 위해. E. Transactions를 참조하세요.
H01.FR.16	대상 EVSE의 상태가 *Faulted* 로 변경된 경우	충전기는 예약을 취소하고 ReservationStatusUpdate를 *Removed* 상태로 보내야 합니다.	
H01.FR.17	대상 EVSE의 상태가 *Faulted* 로 변경된 경우	충전기는 예약을 취소하고 ReservationStatusUpdate를 *Removed* 상태로 전송해야 합니다.	
H01.FR.18	구성 변수 `ReservationNonEvseSpecific`가 *true* 로 설정된 경우.	충전기는 지정되지 않은 EVSE에 대한 예약을 수락해야 합니다.	
H01.FR.19	구성 변수 `ReservationNonEvseSpecific`가 설정되지 않았거나 *false* 로 설정된 경우.	충전기는 지정되지 않은 EVSE에 대한 예약을 거부해야 합니다.	
H01.FR.20	H01.FR.04 AND 사용 가능한 EVSE의 수는 예약 수와 같음	충전기는 EVSE의 모든 커넥터에 대해 다음을 전송해야 합니다. - StatusNotificationRequest(*connectorStatus* = `Reserved`) 또는 - NotifyEventRequest(*component* = "Connector", *variable* = "AvailabilityState", *trigger* = "Delta", *actualValue* = "Reserved")	EVSE가 예약된 경우 모든 커넥터가 예약된 것으로 보고됩니다.
H01.FR.23	충전기가 *evseId* 에 대한 ReserveNowRequest를 수신하고 이 EVSE가 사용 가능한 경우	충전기는 ReserveNowResponse를 *Accepted* 상태로 응답하고 EVSE의 모든 커넥터에 대해 다음을 전송해야 합니다. - StatusNotificationRequest가 *connectorStatus* = `Reserved`이거나, - NotifyEventRequest가 *component* = "Connector", *variable* = "AvailabilityState", *trigger* = "Delta", *actualValue* = "Reserved"인 경우	EVSE가 예약된 경우 모든 커넥터가 예약된 것으로 보고됩니다.

ID	전제 조건	요구 사항 정의	참고
H01.FR.24	H01.FR.06 AND 특정 *connectorType* 에 대한 예약 수는 해당 특정 *connectorType* 을 가진 사용 가능한 EVSE의 수와 같습니다.	충전기는 특정 *connectorType* 을 가진 EVSE의 모든 커넥터에 대해 다음을 전송해야 합니다. - *connectorStatus* = `Reserved` 인 StatusNotificationRequest 또는 - *component* = "Connector", *variable* "AvailabilityState", *trigger* = "Delta", *actualValue* = "Reserved"인 NotifyEventRequest	EVSE가 특정 *connectorType* 에 대해 예약된 경우 EVSE의 모든 커넥터가 예약된 것으로 보고됩니다.

H02 - Cancel Reservation

Table 150. H02 - 예약 취소

번호	유형	설명
1	**이름**	예약 취소
2	**ID**	H02
	기능 블록	H. 예약
3	**목표**	충전기에서 예약을 취소합니다.
4	**설명**	이 사용 사례는 EV 운전자가 기존 예약을 취소하는 방법을 설명합니다. CSMS는 EV 운전자가 충전기에서 예약한 것을 취소할 수 있습니다.
	Actors	충전기, CSMS, EV 운전자
	시나리오 설명	**1.** EV 운전자가 CSMS에 예약 취소를 요청합니다. **2.** 예약을 취소하려면 CSMS가 CancelReservationRequest를 충전기로 보냅니다. **3.** 충전기에 요청 PDU의 reservationId와 일치하는 예약이 있는 경우 *Accepted* 상태를 반환합니다. **4.** 특정 EVSE가 이 예약에 예약된 경우 충전기는 해당 EVSE의 모든 커넥터에 대한 *Available* 상태와 함께 StatusNotificationRequest를 보냅니다. **5.** CSMS가 StatusNotificationResponse로 충전기에 응답합니다. **6.** 예약이 취소됩니다.
5	**필수 조건**	- 기능 블록 *Reservation* 이 설치되었습니다. - EV Driver가 충전기에서 예약을 했습니다.
6	**사후 조건**	**성공 사후 조건:** CSMS가 충전기에서 EV 운전자의 예약을 취소할 수 있었습니다. **실패 사후 조건:** n/a

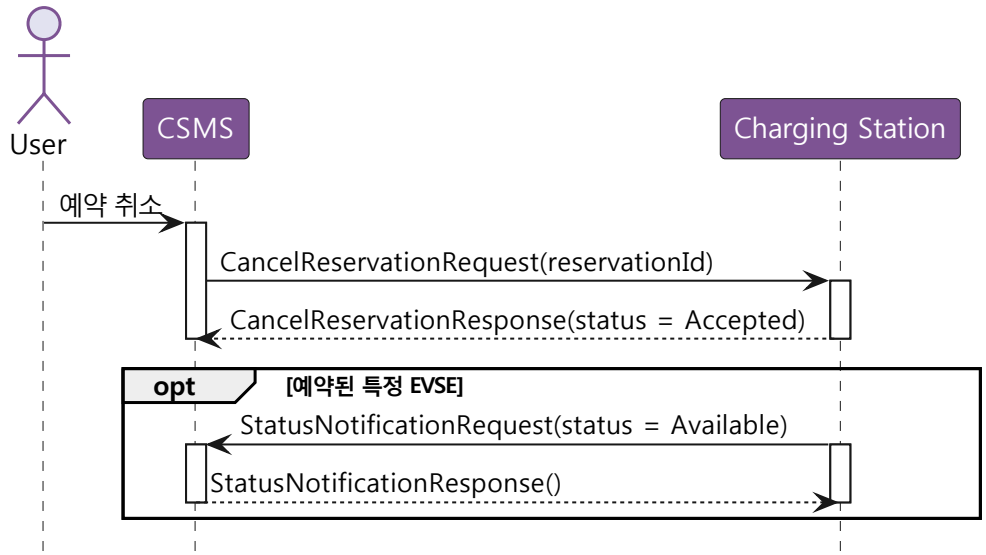

Figure 81. 시퀀스 다이어그램: 예약 취소

7	**오류 처리**	n/a

8	**참고**	충전기는 ReservationStatusUpdate를 보내지 않습니다. CSMS가 명시적으로 취소했기 때문에 이미 이벤트를 알고 있습니다.

H02 - Cancel Reservation - Requirements

Table 151. H02 - 요구 사항

ID	전제 조건	요구 사항 정의
H02.FR.01	충전기에서 CancelReservationRequest를 수신했지만 일치하는 reservationId가 없습니다.	충전기는 *Rejected* 를 반환해야 합니다.
H02.FR.02	충전기에서 유효하고 알려진 reservationId가 있는 CancelReservationRequest를 수신한 경우.	예약이 취소되어야 합니다.

H03 - Use a reserved EVSE

Table 152. H03 - 예약된 EVSE 사용

번호	유형	설명
1	**이름**	예약된 EVSE 사용
2	**ID**	H03
	기능 블록	H. 예약
3	**목표**	예약된 EVSE 사용
4	**설명**	이 사용 사례는 IdToken 및 GroupIdToken 정보를 기반으로 예약된 EVSE를 사용하는 방법을 다룹니다.
	Actors	충전기, CSMS, EV 운전자
S1	*시나리오 목표*	동일한 IdToken으로 예약된 EVSE 사용
	시나리오 설명	**1.** CSMS는 특정 IdTokenType에서 사용할 EVSE를 예약하기 위해 충전기에 ReserveNowRequest를 보냅니다. **2.** ReserveNowRequest를 수신하면 충전기는 ReserveNowResponse로 응답합니다. **3.** 특정 EVSE가 이 예약에 이용된 경우 충전기는 해당 EVSE의 모든 커넥터에 대해 상태가 *Reserved* 인 StatusNotificationRequest를 전송합니다. **4.** CSMS는 충전기에 StatusNotificationResponse로 응답합니다. **5.** EV 운전자가 충전기에 IdTokenType을 제시하고 IdTokenType이 예약의 IdTokenType과 동일하면 충전기는 IdTokenType을 인식하고 충전을 시작하고 E02 - Start Transaction - Cable Plugin First가 적용됩니다.
5	**필수 조건**	n/a
6	**사후 조건**	n/a

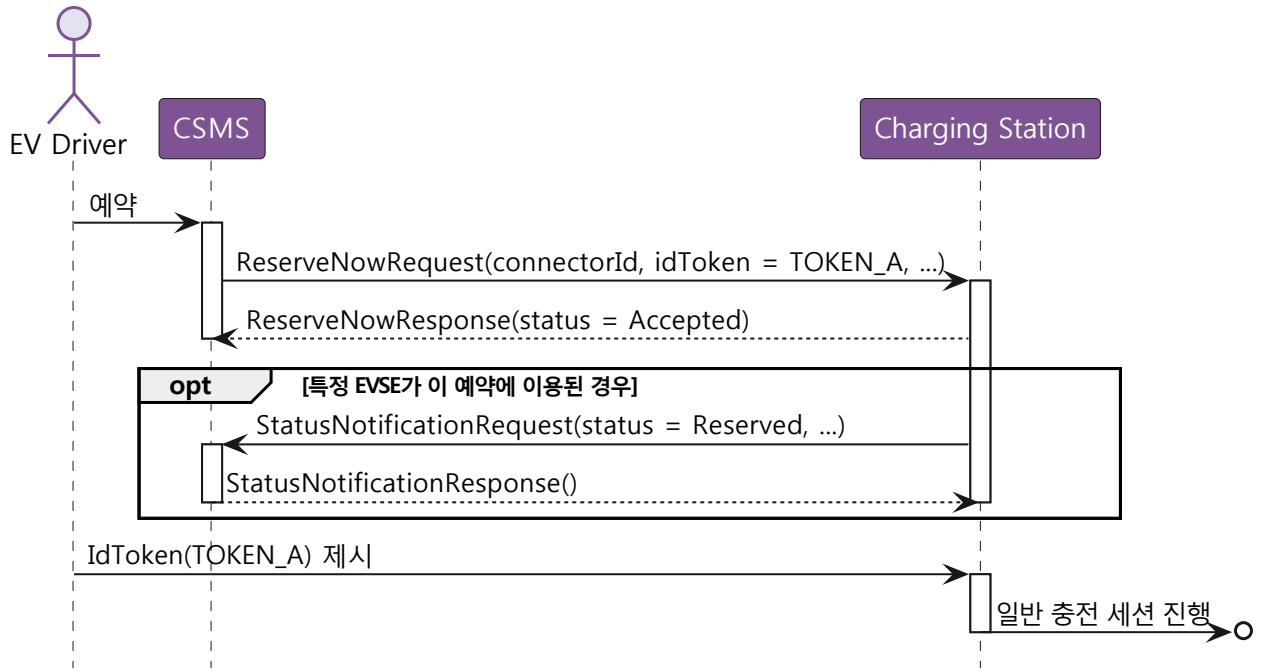

Figure 82. 시퀀스 다이어그램: IdToken이 있는 예약된 EVSE 사용

S2	*시나리오 목표*	동일한 GroupIdToken으로 예약된 EVSE 사용
	시나리오 설명	**1.** CSMS는 GroupId가 포함된 ReserveNowRequest를 충전기로 보내 특정 IdTokenType에서 사용할 EVSE를 예약합니다. **2.** ReserveNowRequest를 수신하면 충전기는 ReserveNowResponse로 응답합니다. **3.** 특정 EVSE가 이 예약에 예약된 경우 충전기는 해당 EVSE의 모든 커넥터에 대해 *Reserved* 상태가 포함된 StatusNotificationRequest를 보냅니다. **4.** CSMS는 충전기에 StatusNotificationResponse로 응답합니다. **5.** EV 운전자가 충전기에 IdTokenType을 제시하고, IdTokenType이 예약의 IdTokenType과 다르면, 충전기는 AuthorizeRequest를 CSMS로 보냅니다. **6.** CSMS는 AuthorizeResponse로 응답합니다. 이 응답 메시지에는 GroupId가 포함됩니다. **7.** 두 응답의 일치하는 GroupId 정보를 기반으로 충전기는 충전을 시작하고 E02 - Start Transaction - Cable Plugin First가 적용됩니다.
5	**필수 조건**	n/a
6	**사후 조건**	n/a

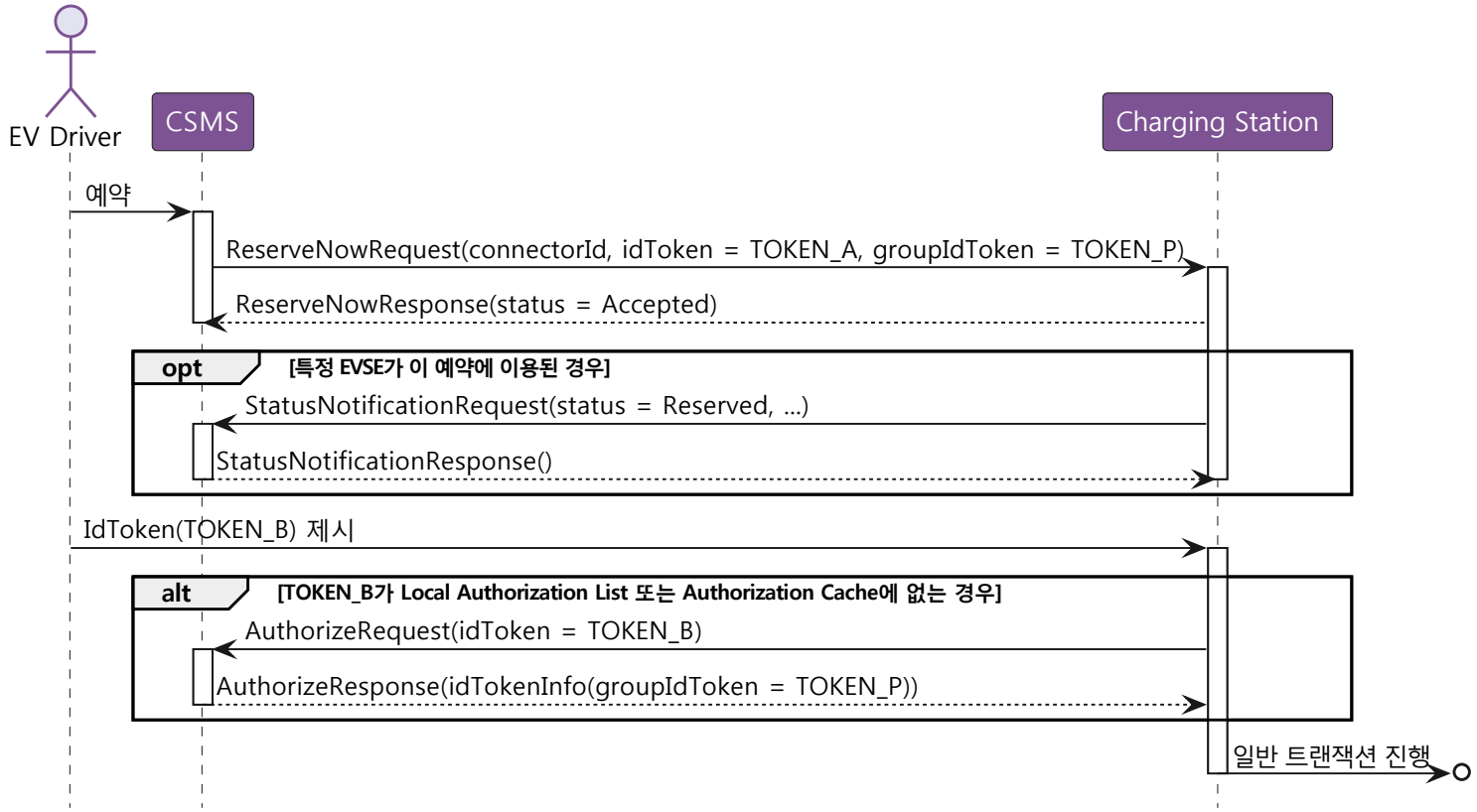

Figure 83. 시퀀스 다이어그램: GroupId가 있는 예약된 EVSE 사용

7	**오류 처리**	n/a
8	**참고**	ReserveNowRequest를 수신한 후, 트랜잭션을 시작하기 전에 AuthorizeRequest로 식별자를 검증하는 것이 좋습니다.

H03 - Use a reserved EVSE - Requirements

Table 153. H03 - 요구 사항

ID	전제 조건	요구 사항 정의
H03.FR.01	특정 *evseId* 에 대한 특정 *idToken* 에 대한 예약이 보류 중입니다.	충전기는 인증을 위해 제시된 IdToken이 예약의 특정 *idToken* 과 일치할 때 해당 EVSE에서 충전을 허용해야 합니다.
H03.FR.02	특정 *connectorType* 에 대한 특정 *idToken* 에 대한 예약이 보류 중입니다.	충전기는 인증을 위해 제시된 IdToken이 예약의 특정 *idToken* 과 일치할 때 *connectorType* 유형의 커넥터가 있는 EVSE에서 충전을 허용해야 합니다.
H03.FR.03	특정 *evseId* 또는 *connectorType* 없이 특정 *idToken* 에 대한 예약이 보류 중입니다.	충전기는 인증을 위해 제시된 IdToken이 예약의 특정 *idToken* 과 일치할 때 EVSE에서 충전을 허용해야 합니다.
H03.FR.04	H03.FR.01 AND 예약의 *groupIdToken* 속성에 값이 있습니다.	충전기는 인증을 위해 제시된 IdToken이 예약의 특정 *idToken* 과 일치하거나 연관된 *groupIdToken* 이 일치할 때 해당 EVSE에서 충전을 허용해야 합니다.
H03.FR.05	H03.FR.02 AND 예약의 *groupIdToken* 속성에 값이 있습니다.	충전기는 인증을 위해 제시된 IdToken이 예약의 특정 *idToken* 과 일치하거나 연관된 *groupIdToken* 이 일치할 때 *connectorType* 유형의 커넥터가 있는 EVSE에서 충전을 허용해야 합니다.
H03.FR.06	H03.FR.03 AND 예약의 *groupIdToken* 속성에 값이 있습니다.	충전기는 인증을 위해 제시된 IdToken이 예약의 특정 *idToken* 과 일치하거나 연관된 *groupIdToken* 이 일치할 때 모든 EVSE에서 충전을 허용해야 합니다.
H03.FR.07	예약의 *groupIdToken* 속성에 값이 있는 경우(선택 사항).	수신한 IdToken과 연관된 *groupIdToken* 을 확인하기 위해 충전기는 로컬 인증 목록 또는 인증 캐시를 조회할 수 있습니다.
H03.FR.08	H03.FR.07 및 수신한 IdToken이 로컬 인증 목록 또는 인증 캐시에서 발견되지 않는 경우.	충전기는 수신 IdToken에 대한 AuthorizeRequest를 CSMS로 보내 연관된 *groupIdToken* 을 가져와야 합니다. (참고: 이 AuthorizeRequest는 idToken이 승인을 위해 제시되었을 때 이미 수행되었을 수 있습니다.)
H03.FR.09	예약과 일치하는 *idToken* 또는 *groupIdToken* 이 제시될 때	충전기는 예약을 사용(소비)한 것으로 간주해야 합니다.
H03.FR.10	H03.FR.09 AND 예약과 관련된 커넥터는 `Reserved` 상태입니다.	충전기는 케이블이 연결되지 않은 경우 커넥터 상태를 `Available` 로 설정하고, 케이블이 이미 연결된 경우 `Occupied` 로 설정해야 합니다.

H04 - Reservation Ended, not used

Table 154. H04 - 예약 종료, 사용 안 함

번호	유형	설명
1	**이름**	예약 종료, 사용 안 함
2	**ID**	H04
	기능 블록	H. 예약
3	**목표**	충전기가 만료된 예약에 대해 CSMS에 알릴 수 있도록 합니다.
4	**설명**	이 사용 사례는 EV 운전자가 충전기를 사용하기 전에 충전기가 종료/시간 초과된 예약에 대해 CSMS에 알리는 방법을 다룹니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 충전기에 예약이 있습니다. **2.** 예약의 만료 날짜에 도달했습니다. **3.** 충전기가 예약을 제거합니다. **4.** 이 예약에 특정 EVSE가 예약된 경우 충전기는 EVSE를 다시 사용할 수 있게 만들고 해당 EVSE의 모든 커넥터에 대한 StatusNotificationRequest를 *Available* 상태로 전송하여 CSMS에 알립니다. **5.** CSMS가 StatusNotificationResponse로 응답합니다. **6.** 충전기가 ReservationStatusUpdateRequest를 *Expired* 상태로 CSMS에 전송합니다. **7.** CSMS가 ReservationStatusUpdateResponse로 응답합니다.
5	**필수 조건**	n/a
6	**사후 조건**	n/a

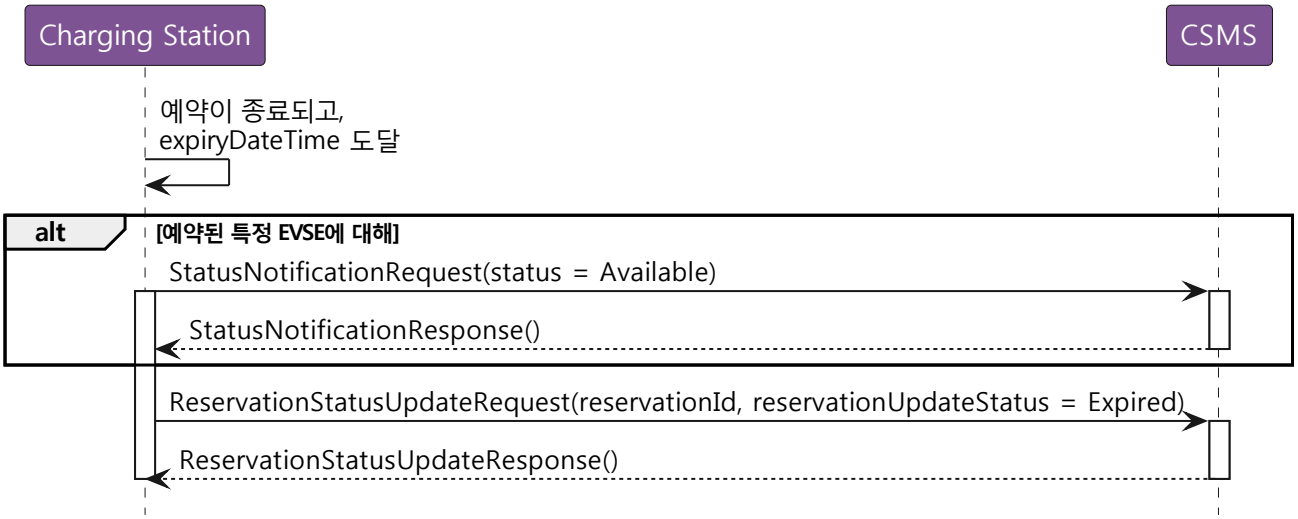

Figure 84. 시퀀스 다이어그램: 예약 종료, 미사용

7	**오류 처리**	n/a
8	**비고**	n/a

H04 - Reservation Ended, not used - Requirements

Table 155. H04 - 요구 사항

ID	전제 조건	요구 사항 정의
H04.FR.01	예약이 종료됨(*expiryDateTime* 에 도달)	충전기는 ReservationStatusUpdateRequest를 *Expired* 상태로 전송해야 합니다.
H04.FR.02	H04.FR.01 AND 이 예약에 특정 EVSE가 예약된 경우	충전기는 이 EVSE에서 다시 충전을 허용해야 합니다.
H04.FR.03	H04.FR.02	충전기는 CSMS에 StatusNotificationRequest를 *Available* 상태로 보내 이 EVSE의 모든 커넥터가 모든 EV 운전자에게 다시 사용 가능하다는 것을 CSMS에 알려야 합니다.

I. TariffAndCost

1. Introduction

이 기능 블록은 충전기에서 디스플레이에 요금 및 비용 정보를 표시할 수 있는 경우 EV 운전자에게 요금 및 비용 정보를 제공합니다.

운전자가 충전을 시작하기 전에 요금 정보를 제공받아 이 충전기에서 이 운전자에게 적용되는 요금 계획을 구성하는 모든 구성 요소에 대한 자세한 가격을 제공해야 합니다. 이것은 사람이 읽을 수 있는 텍스트 메시지이므로 개인적인 환영 메시지와 같은 다른 용도로도 사용할 수 있습니다.

일부 비즈니스 사례에서는 EV 운전자에게 충전 중 실행 총 비용을 정기적으로 적합한 간격으로 업데이트하여 표시해야 할 수 있습니다. EV 운전자가 충전을 중단하면 방금 중단된 트랜잭션의 총 비용을 표시해야 합니다.

모든 요금 및 비용은 구성 변수 Currency에서 구성된 통화로 표시됩니다.

1.1. Why no structured tariff information?

요금 구조가 매우 복잡해질 수 있으므로 충전기에서 이를 사람이 읽을 수 있는 텍스트로 변환하기 어려울 것입니다. CSO는 요금의 소유자이며 충전기에 사람이 읽을 수 있는 요금 텍스트를 제공할 수 있어야 합니다. CSO가 자체 요금에서 사람이 읽을 수 있는 텍스트를 생성할 수 없다면 충전기가 어떻게 이를 할 수 있을 것으로 기대할 수 있겠습니까? 그렇기 때문에 우리는 OCPP에서 요금의 복잡성을 제외했습니다.

2. Show EV Driver-specific Tariff Information

I01 - Show EV Driver-specific Tariff Information

Table 156. I01 - EV 운전자별 요금 정보 표시

번호	유형	설명
1	**이름**	EV 운전자별 요금 정보 표시
2	**ID**	I01
	기능 블록	I. 단가 및 비용
3	**목표**	트랜잭션 시작 전에 EV 운전자별 단가를 표시합니다.
4	**설명**	EV 운전자가 EV를 충전하려면 현재 충전기에서 얼마의 충전 비용이 드는지 알고 싶어합니다. EV 운전자는 (RFID) 토큰으로 인증됩니다. 충전기는 CSMS에 제시된 토큰에 대한 정보를 요청합니다. CSMS는 이 EV 운전자에게 적용되는 단가를 포함하여 토큰에 대한 정보를 반환합니다.
	Actors	충전기, CSMS, EV 운전자
	시나리오 설명	**1.** EV 운전자가 EV를 충전하려면 IdTokenType을 제시합니다. **2.** 충전기는 AuthorizeRequest를 CSMS로 보내 승인을 요청합니다. **3.** AuthorizeRequest를 수신하면 CSMS는 AuthorizeResponse로 응답합니다. 이 응답 메시지는 CSMS가 IdTokenType을 수락하는지 여부를 나타내며 personalMessage 필드에 EV 운전자별 요금을 표시합니다. **4.** 충전기는 EV 운전자에게 EV 운전자별 요금을 표시합니다.
	대체 시나리오	I04 - Show Fallback Tariff Information
5	**필수 조건**	충전기는 요금 정보를 지원합니다.
6	**사후 조건**	**성공한 사후 조건:** EV 운전자는 인증되고 자신에게 적용되는 요금을 알고 충전을 시작할 수 있습니다. **실패 사후 조건:** 인증 상태가 *Accepted* 이 아닌 경우 EV 운전자는 시작할 수 ***없으며*** 요금을 알지 못할 수 있습니다.

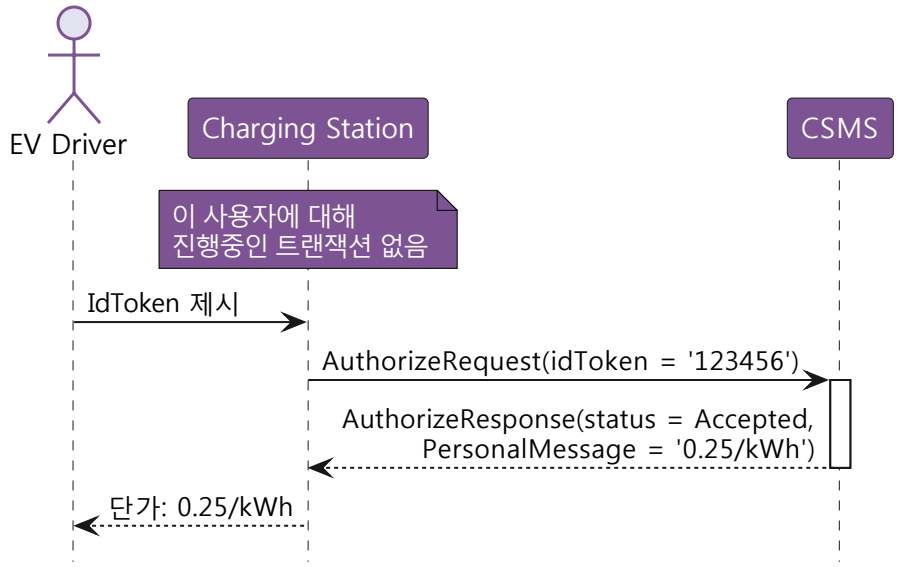

Figure 85. 시퀀스 다이어그램: EV 운전자별 요금 정보 표시

7	**오류 처리**	n/a
8	**참고**	이런 방식으로 제시된 요금 정보는 제시된 다른 토큰과 동일할 수 있습니다. EV 운전자가 이해할 수 있는 언어를 알고 있고 적용 가능한 경우, 해당 언어로 요금 정보를 표시하는 것을 추천합니다. 운전자가 요금에 동의하지 않을 때 트랜잭션을 취소할 수 있는 옵션을 제공하는 것을 추천합니다. 운전자에게 트랜잭션을 취소하는 방법이 명확하다면 이는 케이블을 연결하지 않은 것일 수도 있고, 사용자 인터페이스에 취소 버튼이 있는 것일 수도 있습니다.

I01 - Show EV Driver-specific Tariff Information - Requirements

Table 157. I01 - Show EV Driver-specific Tariff Information - Requirements

ID	전제 조건	요구 사항 정의
I01.FR.01		CSMS는 AuthorizeResponse 메시지의 PersonalMessage 필드에 EV 운전자별 요금 정보를 보낼 수 있습니다.
I01.FR.02		CSMS는 충전기가 요금 또는 DisplayMessage 기능을 지원하는 경우에만 요금 정보를 보내야 합니다.
I01.FR.03	I01.FR.01	충전기는 EV 운전자별 요금 정보를 EV 운전자에게 표시해야 합니다.

I02 - Show EV Driver Running Total Cost During Charging

Table 158. I02 - 충전 중 EV 운전자 총 비용 표시

번호	유형	설명
1	**이름**	충전 중 EV 운전자의 총 주행 비용 표시
2	**ID**	I02
	기능 블록	I. 단가 및 비용
3	**목표**	충전 중 EV 운전자에게 총 주행 비용을 표시
4	**설명**	트랜잭션이 진행되는 동안 운전자는 관련 간격으로 업데이트된 총 주행 비용을 알고 싶어합니다.
	Actors	충전기, CSMS, EV 운전자
	시나리오 설명	**1.** CSMS는 Y초마다 CostUpdatedRequest를 충전기로 보내 현재 총 비용을 업데이트합니다. **2.** CostUpdatedRequest를 수신하면 충전기는 CostUpdatedResponse로 응답합니다. **3.** 충전기는 EV 운전자에게 현재 총 비용을 보여줍니다.
	대체 시나리오	**1.** *eventType* = `Updated` 인 TransactionEventRequest를 수신하면 CSMS는 TransactionEventResponse의 있는 *timestamp* 와 *meterValue* 에 해당하는 running cost을 *totalCost* 필드에 넣어 반환합니다. **2.** 충전기는 EV 운전자에게 현재 총 비용을 보여줍니다.
5	**필수 조건**	충전기가 요금 정보를 지원함 진행 중인 트랜잭션

번호	유형	설명
6	사후 조건	**성공한 사후 조건:** EV 운전자는 충전 중 총 비용을 알고 있습니다. **실패한 사후 조건:** EV 운전자는 충전 중 총 비용을 알 수 없습니다.

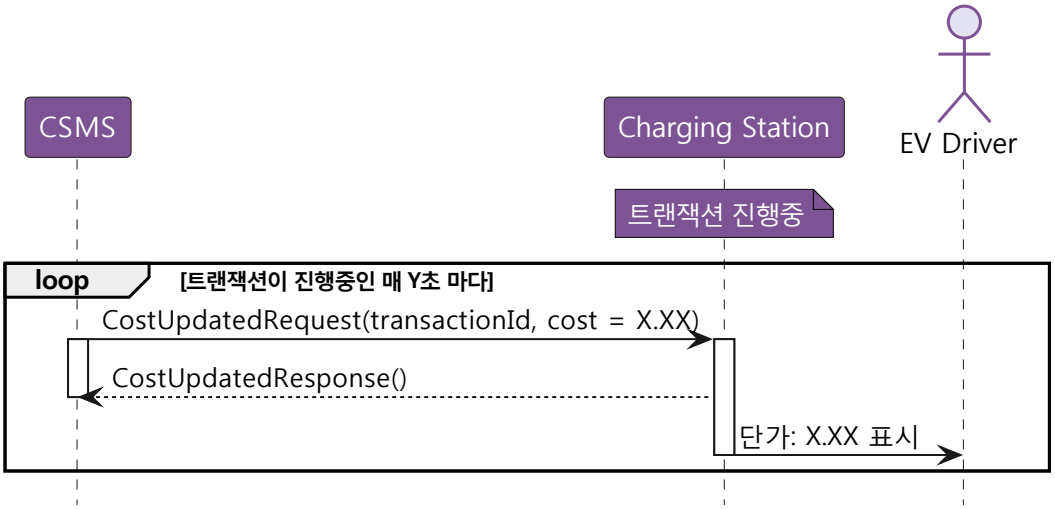

Figure 86. 시퀀스 다이어그램: 충전 중 EV 운전자의 총 비용 표시

7	오류 처리	해당 없음
8	참고	실행 비용을 매우 자주 업데이트하면 많은 메시지가 생성되어 모바일 데이터 비용이 많이 발생할 수 있습니다.

I02 - Show EV Driver Running Total Cost During Charging - Requirements

Table 159. I02 - 충전 중 EV 운전자의 총 주행 비용 표시 - 요구 사항

ID	전제 조건	요구 사항 정의
I02.FR.01		CSMS는 적절한 간격/순간에 CostUpdatedRequest를 보내거나 TransactionEventResponse에서 주행 비용을 반환해야 합니다. 이는 충전 속도, 주행 비용 등에 따라 달라질 수 있습니다.
I02.FR.02	CostUpdatedRequest 메시지를 수신하면	충전기는 CostUpdatedResponse 메시지로 응답해야 합니다.
I02.FR.03	I02.FR.02	충전기는 EV 운전자에게 현재 총 비용을 표시해야 합니다.
I02.FR.04	TransactionEventResponse에서 실행 비용이 보고되는 경우	충전기는 EV 운전자에게 현재 실행 비용을 표시해야 합니다.

I03 - Show EV Driver Final Total Cost After Charging

Table 160. I03 - Show EV Driver Final Total Cost After Charging

번호	유형	설명
1	이름	충전 후 EV 운전자에게 최종 비용 표시

번호	유형	설명
2	**ID**	I03
	기능 블록	I. 단가 및 비용
3	**목표**	트랜잭션이 완료된 후 EV 운전자에게 총 비용을 표시합니다.
4	**설명**	EV 운전자는 자신의 식별 토큰(예: RFID)을 제시하여 진행 중인 트랜잭션을 중단합니다. 트랜잭션이 중단되고 트랜잭션의 총 비용이 EV 운전자에게 표시됩니다.
	Actors	충전기, CSMS, EV 운전자
	시나리오 설명	**1.** EV 운전자는 트랜잭션을 중단하기 위해 IdTokenType을 제시합니다. **2.** 충전기는 TransactionEventRequest (eventType = Ended)를 전송합니다. **3.** CSMS는 TransactionEventResponse로 응답하며 여기에는 트랜잭션의 총 비용이 포함됩니다. **4.** 충전기는 EV 운전자에게 총 비용을 보여줍니다.
	대체 시나리오	I05 - Show Fallback Total Cost Message
5	**필수 조건**	충전기는 단가 정보를 지원합니다. 진행 중인 트랜잭션
6	**사후 조건**	**성공한 사후 조건:** EV 운전자는 트랜잭션의 총 비용을 알고 있습니다. **실패한 사후 조건:** EV 운전자는 트랜잭션의 총 비용을 알지 못합니다.

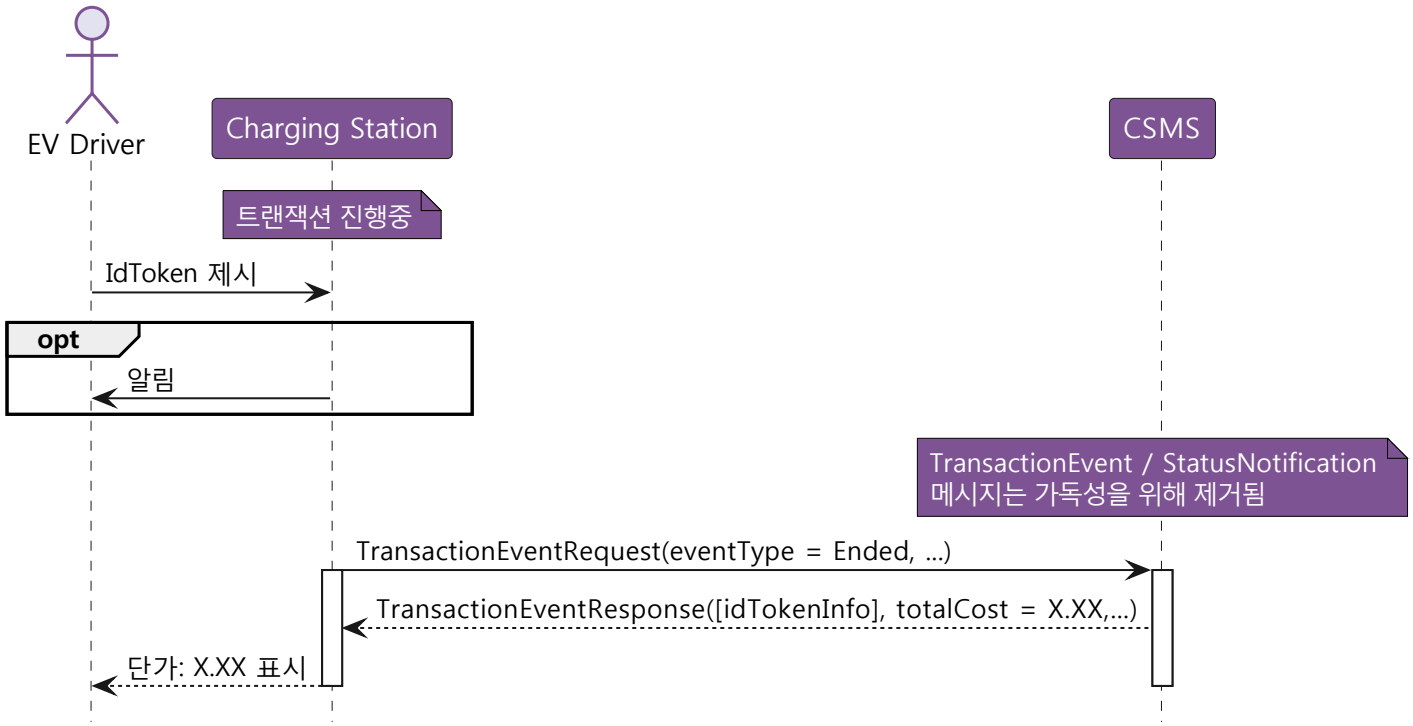

Figure 87. 시퀀스 다이어그램: 충전 후 EV 운전자 최종 총 비용 표시

7	**오류 처리**	n/a

		트랜잭션이 종료될 때 충전기가 오프라인 상태였고 그 후 얼마 지나지 않아 충전기가 다시 온라인 상태가 되었을 때 *totalCost* 가 포함된 TransactionEventResponse가 수신된 경우 비용을 표시할 필요가 없습니다. 사용자가 이미 떠났을 가능성이 높기 때문입니다. `TxStopPoint` 가 ParkingBayOccupancy로 정의된 경우에도 비슷한 상황이 적용되며, 이 경우 EV는 충전기를 떠나서 트랜잭션을 종료해야 합니다.
8	**참고**	위의 시나리오 설명과 시퀀스 다이어그램은 다음과 같이 구성된 중지 트랜잭션에 대한 구성 변수를 기반으로 합니다. `TxStopPoint`: ParkingBayOccupancy, EVConnected, Authorized 이 유즈케이스는 다른 구성에도 유효하지만 트랜잭션이 다른 순간에 중지될 수 있으며, 이때 메시지를 보내는 순서가 변경될 수 있습니다. 자세한 내용은 사용 사례를 참조하세요. E06 - Stop Transaction options

I03 - Show EV Driver Final Total Cost After Charging - Requirements

Table 161. I03 - 충전 후 EV 운전자 최종 총 비용 표시 - 요구 사항

ID	전제 조건	요구 사항 정의
I03.FR.01	트랜잭션이 중지된 경우	충전기는 TransactionEventRequest(eventType = Ended)를 CSMS로 보내야 합니다.
I03.FR.02	I03.FR.01 AND 총 비용이 CSMS에 알려진 경우	CSMS는 TransactionEventResponse 메시지의 totalCost 필드에 트랜잭션의 총 비용을 보내야 합니다.
I03.FR.03	I03.FR.02 AND 트랜잭션이 중단되었을 때 충전기가 온라인 상태였습니다.	충전기는 EV 운전자에게 총 비용을 표시해야 합니다.
I03.FR.04		무료 트랜잭션을 나타내기 위해 CSMS는 *totalCost* 를 0.00으로 설정해야 합니다. 따라서 *totalCost* 를 생략하는것은 트랜잭션이 무료였음을 의미하지 않습니다.
I03.FR.05	I02.FR.02 AND `TxStopPoint` 는 ParkingBayOccupancy로 정의됩니다.	충전기는 EV 운전자에게 총 비용을 표시해서는 안 됩니다. (운전자가 이미 떠났습니다.)

I04 - Show Fallback Tariff Information

Table 162. I04 - Show Fallback Tariff Information

번호	유형	설명
1	**이름**	대체 요금 정보 표시
2	**ID**	I04
	기능 블록	I. 요금 및 비용
3	**목표**	충전기에서 이 EV 운전자에 대한 요금 정보를 검색할 수 없는 경우 EV 운전자에게 일부 정보, 일반 요금, 메시지 등을 표시합니다.
4	**설명**	EV 운전자가 EV를 충전하고 싶을 때, 현재 충전기에서 충전 비용이 얼마인지 알고 싶어하지만 충전기에서 이 EV 운전자에 대한 특정 요금을 얻을 수 없는 경우(예: 충전기가 *Offline* 이거나 EV 운전자 전용 요금이 없는 경우). 이러한 시나리오의 경우 충전기에서 대체 요금 정보 메시지를 구성할 수 있습니다.

번호	유형	설명
	Actors	충전기, EV 운전자
	시나리오 설명	**1.** EV 운전자가 EV를 충전하고 싶어하며, IdTokenType을 제시합니다. **2.** 충전기가 EV 운전자를 인증 캐시에 대해 승인합니다. **3.** 충전기가 EV 운전자에게 TariffFallbackMessage를 보여줍니다.
	대체 시나리오	I01 - Show EV Driver-specific Tariff Information
5	**필수 조건**	충전기는 요금 정보를 지원합니다. 구성 변수: TariffFallbackMessage가 구성됩니다.
6	**사후 조건**	**성공한 사후 조건:** EV 운전자에게 대체 요금 정보 메시지가 표시되었습니다. **실패한 사후 조건:** EV 운전자는 이 충전기의 요금에 대한 정보가 없습니다.

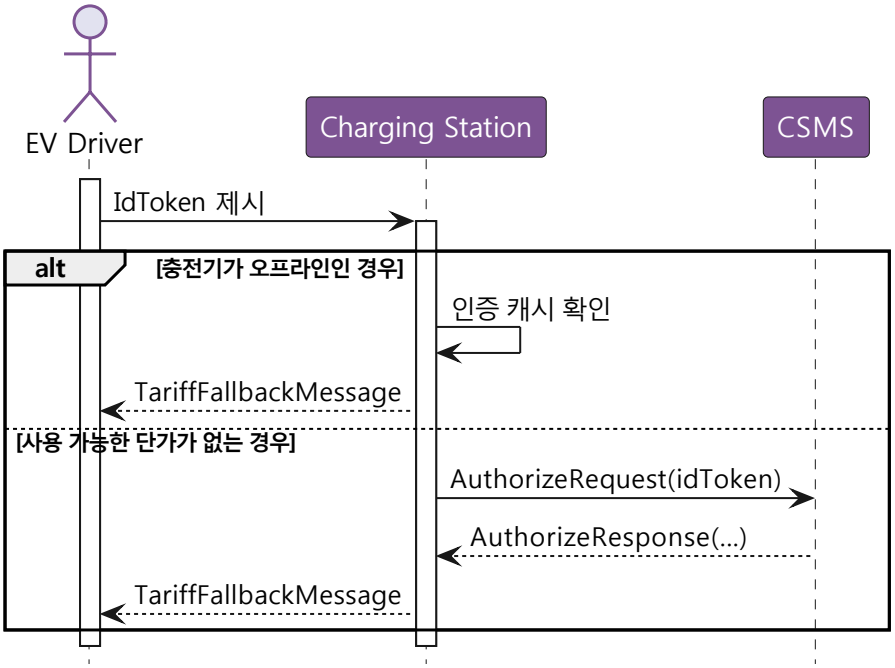

Figure 88. 시퀀스 다이어그램: 대체 단가 정보 표시

7	**오류 처리**	n/a
8	**참고**	n/a

I04 - Show Fallback Tariff Information - Requirements

Table 163. I04 - 대체 단가 정보 표시 - 요구 사항

ID	전제 조건	요구 사항 정의
I04.FR.01	충전기에서 EV 운전자에 대한 특정 단가를 얻을 수 없는 경우(예: 충전기가 오프라인이거나 EV 운전자 전용 요금이 제공되지 않음)	충전기는 EV 운전자에게 대체 요금 정보 메시지를 표시해야 하며, 이는 구성 변수 `TariffFallbackMessage`에서 구성됩니다.

ID	전제 조건	요구 사항 정의
I04.FR.02		CSMS는 구성 변수 `TariffFallbackMessage`를 통해 TariffFallbackMessage를 구성할 수 있습니다.

I05 - Show Fallback Total Cost Message

Table 164. I05 - Show Fallback Total Cost Message

번호	유형	설명
1	**이름**	폴백 총 비용 메시지 표시
2	**ID**	I05
	기능 블록	I. 단가 및 비용
3	**목표**	트랜잭션이 중단될 때 충전기가 *offline* 인 경우 EV 운전자에게 실제 총 비용 대신 메시지를 표시합니다.
4	**설명**	EV 운전자가 진행 중인 트랜잭션을 중단하려고 하지만 충전기가 *offline* 인 경우. 트랜잭션은 앞서 설명한 대로 중단됩니다. 충전기는 중단된 트랜잭션의 총 비용을 검색할 수 없습니다. EV 운전자에게 메시지를 제공해야 하며, 이 메시지는 구성 변수에서 구성할 수 있습니다: TotalCostFallbackMessage.
	Actors	충전기, EV 운전자
	시나리오 설명	**1.** EV 운전자가 IdTokenType을 제시하여 트랜잭션을 중단합니다. **2.** 충전기가 에너지 제공을 중단합니다. **3.** 충전기가 TotalCostFallbackMessage를 EV 운전자에게 보여줍니다.
	대체 시나리오	I03 - Show EV Driver Final Cost After Charging
5	**필수 조건**	충전기가 요금 정보를 지원합니다. 충전기가 *offline* 입니다. 구성 변수: `TotalCostFallbackMessage`가 구성되었습니다.
6	**사후 조건**	**성공한 사후 조건:** EV 드라이버가 사전 구성된 폴백 메시지를 수신했습니다. **실패 사후 조건:** EV 드라이버가 사전 구성된 폴백 메시지를 수신하지 못했습니다.

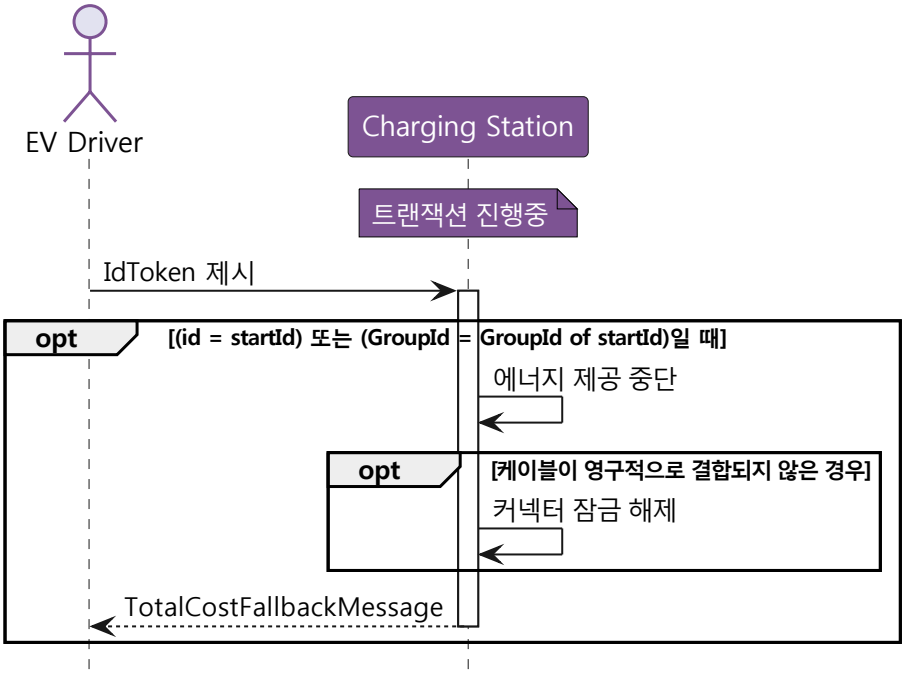

Figure 89. 시퀀스 다이어그램: 폴백 총 비용 메시지 표시

7	**오류 처리**	n/a
8	**참고**	n/a

I05 - Show Fallback Total Cost Message - Requirements

Table 165. I05 - 폴백 총 비용 메시지 표시 - 요구 사항

ID	사전 조건	요구 사항 정의
I05.FR.01		CSMS는 폴백을 구성할 수 있습니다. 구성 변수를 통한 총 비용 정보 메시지: `TotalCostFallbackMessage`.
I05.FR.02	충전기가 오프라인 상태이기 때문에 중단된 트랜잭션에 대한 총 비용을 검색할 수 없는 경우.	충전기는 EV 운전자에게 폴백 총 비용 정보 메시지를 표시해야 합니다.

I06 - Update Tariff Information During Transaction

Table 166. I06 - 트랜잭션 중 단가 정보 업데이트

번호	유형	설명
1	**이름**	트랜잭션 중 단가 정보 업데이트
2	**ID**	I06
	기능 블록	I. 단가 및 비용
3	**목표**	트랜잭션 중에 EV 운전자에게 최신 단가 정보를 표시합니다.

번호	유형	설명
4	**설명**	충전 중(특히 DC 고속 충전)에는 EV 운전자에게 최신 단가 정보가 제공되면 표시하는 것이 유용할 수 있습니다. 예: 단가에 대역폭이 있는 경우: *충전 비용은 현재 에너지 가격에 따라 0.25~0.40유로/kWh입니다. 현재 가격은 0.28유로/kWh입니다.* 그런 다음 가격이 변경되면 이 단가 정보를 업데이트해야 합니다. *충전 비용은 현재 에너지 가격에 따라 0.25~0.40유로/kWh입니다. 현재 가격은 0,32유로/kWh입니다.*
	시나리오 설명	**1.** 충전기는 트랜잭션 중에 TransactionEventRequest (eventType = Updated) 메시지를 보냅니다. **2.** CSMS가 TransactionEventRequest 메시지를 받으면 업데이트된 요금 정보가 있는지 확인합니다. **3.** CSMS는 TransactionEventResponse 메시지로 확인합니다. 여기에는 업데이트된 요금 정보가 포함되어 있습니다(있는 경우).
5	**필수 조건**	충전기는 요금 정보를 지원합니다. 진행 중인 트랜잭션이 있습니다.
6	**사후 조건**	**성공적인 사후 조건:** 업데이트된 요금 정보가 EV 운전자에게 표시됩니다. **실패 사후 조건:** EV 운전자에게 업데이트된 요금 정보가 표시되지 않았습니다.

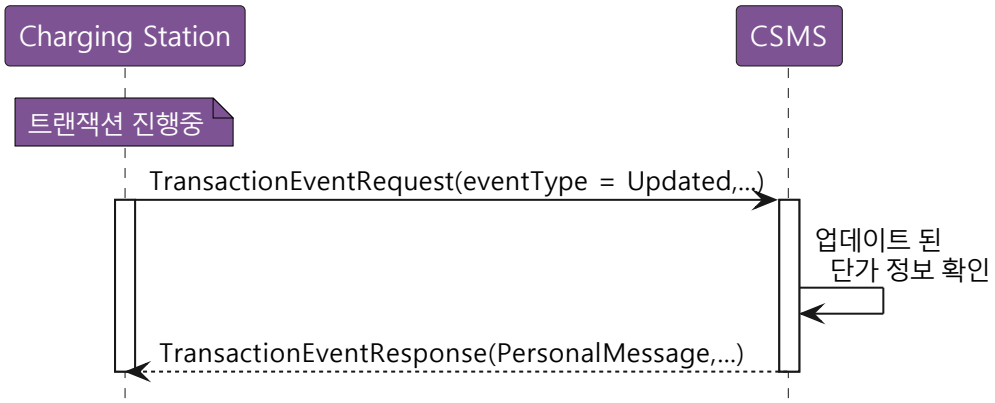

Figure 90. 시퀀스 다이어그램: 트랜잭션 중 단가 정보 업데이트

7	**오류 처리**	해당 없음
8	**참고**	트랜잭션 시작 시 전달된 단가를 전체 트랜잭션에 사용해야 하는 정책이나 법적 요구 사항이 있을 수 있으며, 이 경우 트랜잭션 중에 업데이트된 단가 정보를 보내서는 안 됩니다.

I06 - Update Tariff Information During Transaction - Requirements

Table 167. I06 - 트랜잭션 중 단가 정보 업데이트 - 요구 사항

ID	전제 조건	요구 사항 정의
I06.FR.01	CSMS가 충전기에서 TransactionEventRequest (eventType = Updated)를 수신하는 경우.	CSMS는 업데이트된 단가 정보가 있는지 확인해야 합니다.
I06.FR.02	I06.FR.01 및 업데이트된 요금 정보가 있는 경우.	CSMS는 PersonalMessage 필드에 업데이트된 요금 정보가 포함된 TransactionEventResponse 메시지로 충전기에 응답해야 합니다.
I06.FR.03	I06.FR.02	충전기는 EV 운전자에게 업데이트된 요금 정보를 표시해야 합니다.

J. MeterValues

1. Introduction

이 기능 블록은 충전기가 주기적이고 시계에 맞춰진(clock-aligned) MeterValues를 보낼 수 있도록 하는 기능을 설명합니다.

충전기에서 CSMS로 MeterValues를 전송하는 것은 새로운 장치 관리 모니터링 기능이 담당하지만 이 메커니즘은 아직 현장에서 검증되지 않았습니다. 따라서 이전의 MeterValuesRequest 메시지는 현재 사용할 수 있습니다.

트랜잭션과 관련된 광범위한 계량 데이터는 의도된 목적에 따라 다양한 방식으로 기록 및 전송할 수 있습니다. 두 가지 명백한 사용 사례가 있지만(하지만 계량기 값의 사용은 이 두 가지에 국한되지 않음):

- Transaction Meter Values
- Clock-Aligned Meter Values

두 가지 유형의 계량기 판독값은 TransactionEventRequest 메시지의 *meterValue* 요소에서 보고될 수 있습니다. Clock-Aligned Meter Values는 독립형 MeterValuesRequest 메시지에서 보고될 수 있습니다.

2. Configuration

이 섹션은 규범적입니다.

2.1. Transaction Meter Values

자주(예: 1-5분 간격) 미터 판독값을 측정하여 CSMS로 전송(일반적으로 "실시간")하여 웹, 앱, SMS 등을 통해 EV 사용자(일반적으로 충전기 주변에 없음)에게 트랜잭션 진행 상황에 대한 정보 업데이트를 제공할 수 있도록 합니다. OCPP에서는 이를 "샘플링된 미터 데이터"라고 합니다. 판독값의 정확한 빈도와 시간은 "충분히 자주" 측정되는 한 그다지 중요하지 않기 때문입니다. "샘플링된 미터 데이터"는 다음 구성 변수로 구성할 수 있습니다.

- `SampledDataTxStartedMeasurands`
- `SampledDataTxUpdatedMeasurands`
- `SampledDataTxUpdatedInterval`
- `SampledDataTxEndedMeasurands`
- `SampledDataTxEndedInterval`

`SampledDataTxUpdatedInterval`은 트랜잭션 중에 TransactionEventRequest (eventType = Updated) 메시지를 통해 전송되도록 의도된 미터링(또는 기타) 데이터 샘플링 사이의 시간(초)입니다. 관례에 따라 "0"(숫자 0) 값은 샘플링된 데이터가 전송되지 않아야 함을 의미하는 것으로 해석됩니다.

`SampledDataTxEndedInterval`은 TransactionEventRequest (eventType = Ended) 메시지에서 전송되도록 의도된 미터링(또는 기타) 데이터 샘플링 사이의 시간(초)입니다.

`SampledDataTxStartedMeasurands`는 TransactionEventRequest의 *meterValues* 필드에 포함될 측정량 세트를 규정하는 쉼표로 구분된 목록입니다(eventType = Started).

`SampledDataTxUpdatedMeasurands`는 TransactionEventRequest (eventType = Updated)의 *meterValues* 필드에 포함될 측정값 세트를 SampledDataTxUpdatedInterval초마다 규정하는 쉼표로 구분된 목록입니다.

`SampledDataTxEndedMeasurands`는 TransactionEventRequest의 *meterValues* 필드에 포함될 샘플링된 측정값을 규정하는 쉼표로 구분된 목록입니다(eventType = Ended). 이러한 측정값은 트랜잭션 시작부터 `SampledDataTxEndedInterval`초마다 가져와야 하며 TransactionEventRequest(eventType = Ended)에서만 전송됩니다.

트랜잭션이 끝날 때 예상되는 측정량 양이 하나의 TransactionEventRequest (eventType = Ended) 메시지에 맞는지 확인해야 합니다. `SampledDataTxEndedMeasurands`의 측정량 수를 최소한으로 유지하고 `SampledDataTxEndedInterval`에서 큰 간격을 구성하여 샘플 수를 작게 유지합니다.

NOTE *트랜잭션 관련* MeterValues는 MeterValuesRequest에서 전송되지 않습니다.

2.2. Clock-Aligned Meter Values

그리드 운영자는 재정적으로 인증된 에너지 미터에서 특정 시계 정렬 시간(일반적으로 15분 또는 30분)에 미터 판독값을 가져오도록 요구할 수 있습니다.

"클럭 정렬 미터 값"은 다음 구성 변수로 구성할 수 있습니다.

- `AlignedDataMeasurands`
- `AlignedDataInterval`

- `AlignedDataTxEndedMeasurands`
- `AlignedDataTxEndedInterval`
- `AlignedDataSendDuringIdle`

`AlignedDataInterval`는 클록 정렬 데이터 간격의 크기(초)입니다. 이는 00:00:00(자정)부터 시작하여 하루에 균등하게 간격을 둔 미터 데이터 집계 간격 세트를 정의하며, 이 시간에 충전기는 측정을 수행하여 MeterValuesRequest 메시지로 CSMS에 전송해야 합니다. 관례에 따라 "0"(숫자 0) 값은 클록 정렬된 데이터가 전송되지 않아야 함을 의미하는 것으로 해석됩니다.

`AlignedDataTxEndedInterval`은 클록 정렬된 데이터 간격의 크기(초)입니다. 이는 00:00:00(자정)부터 시작하여 하루에 균등하게 간격을 둔 미터 데이터 집계 간격 세트를 정의하며, TransactionEventRequest(eventType = Ended) 메시지로 전송되도록 의도되었습니다.

예를 들어, 900(15분)의 값은 0:15, 0:30, 0:45, 1:00, 1:15 등 매일 0:00에서 시작하여 15분 간격으로 96개로 나누어야 함을 나타냅니다.

`AlignedDataMeasurands`는 MeterValuesRequest PDU에 `AlignedDataInterval`초마다 포함될 측정량 세트를 규정하는 쉼표로 구분된 목록입니다.

`AlignedDataTxEndedMeasurands`는 TransactionEventRequest (eventType = Ended) PDU의 *meterValue* 요소에 포함될 클록 정렬 주기적 측정값 집합을 트랜잭션의 모든 `AlignedDataTxEndedInterval`에 대해 규정하는 쉼표로 구분된 목록입니다.

`AlignedDataSendDuringIdle`는 진행 중인 트랜잭션이 없는 경우에만 클록 정렬 미터 값을 전송하는 데 사용할 수 있습니다.

NOTE 트랜잭션에 관련된 EVSE의 시계 정렬 미터 값은 MeterValuesRequests가 아닌 *context* = `Sample.Clock` 인 TransactionEventRequests에서 전송될 수 있습니다.

2.3. Multiple Locations/Phases

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

예: *Inlet*(모든 3상)(그리드 연결) 및 *Outlet*(두 EVSE 모두에서 EVSE당 3상)에서 *Current.Import* 를 측정할 수 있는 충전소를 예로 들어봅니다. *Current.Import* 는 `AlignedDataMeasurands`에 설정되어 있습니다. `AlignedDataInterval`은 900(초)으로 설정되어 있습니다. 그런 다음 충전소는 다음을 보내야 합니다. (15분마다)

- MeterValuesRequest: evseId = 0; 3개의 SampledValue 요소, location = Inlet인 위상당 하나씩.
- MeterValuesRequest: evseId = 1; 3개의 SampledValue 요소, location = Outlet인 위상당 하나씩.
- MeterValuesRequest: evseId = 2; 3개의 SampledValue 요소, location = Outlet인 위상당 하나씩.

NOTE 구성 변수 `SampledDataRegisterValuesWithoutPhases`의 값이 *true* 인 경우 측정 대상 `Energy.Active.Import.Register` 의 미터 값은 개별 위상 값을 보고하지 않고 모든 위상에 대한 총 에너지만 보고합니다.

2.4. Signed Meter Values

OCPP 2.0.1은 서명된 미터 값을 지원합니다. 충전기가 서명된 미터 값을 지원하는 경우 구성 변수 `AlignedDataSignReadings` 및 `SampledDataSignReadings`를 사용하여 이를 보고할 수 있습니다. 그런 다음 CSMS는 이 동일한 변수를 사용하여 서명된 미터 값의 사용을 *켜거나 끌* 수 있습니다.

활성화된 경우 충전기는 서명된 미터 값을 SampledValue의 *SignedMeterValue* 필드에 넣습니다.

2.5. Configuration Examples

다음은 트랜잭션 관련 측정항목에 대한 구성의 몇 가지 예입니다.

트랜잭션 종료 시 시작/중지에 대한 샘플링된 에너지 레지스터 값만

- SampledDataCtrlr.TxStartedMeasurands 및 TxUpdatedMeasurands는 비어 있습니다.
- SampledDataCtrlr.TxEndedMeasurands = "Energy.Active.Import.Register"
- SampledDataCtrlr.TxEndedInterval = 0

트랜잭션 시작, 트랜잭션 중, 트랜잭션 종료 시 에너지 레지스터 값

- SampledDataCtrlr.TxStartedMeasurands = "Energy.Active.Import.Register"
- SampledDataCtrlr.TxUpdatedMeasurands = "Energy.Active.Import.Register"
- SampledDataCtrlr.TxUpdatedInterval = 300(5분마다)
- SampledDataCtrlr.TxEndedMeasurands = "Energy.Active.Import.Register"
- SampledDataCtrlr.TxEndedInterval = 0

트랜잭션 중, 트랜잭션 종료 시 시작/중지되는 클록 정렬 레지스터 값만

- SampledDataCtrlr.TxStartedMeasurands 및 TxUpdatedMeasurands는 비어 있습니다.
- SampledDataCtrlr.TxEndedMeasurands = "Energy.Active.Import.Register"
- SampledDataCtrlr.TxEndedInterval = 0
- AlignedDataCtrlr.Measurands = "Energy.Active.Import.Register"
- AlignedDataCtrlr.Interval = 300(5분마다)

3. Use cases & Requirements

3.1. MeterValues

J01 - Sending Meter Values not related to a transaction

Table 168. J01 - 트랜잭션과 관련 없는 미터 값 전송

번호	유형	설명
1	**이름**	트랜잭션과 관련 없는 미터 값 전송
2	**ID**	J01
	기능 블록	J. 미터 값
3	**목표**	충전기 미터 값에 대한 정보를 제공하기 위해 전기 미터 또는 기타 센서/트랜스듀서 하드웨어를 샘플링합니다.
4	**설명**	충전기는 미터 값에 대한 정보를 제공하기 위해 전기 미터 또는 기타 센서/트랜스듀서 하드웨어를 샘플링합니다. 구성 설정에 따라 충전기는 미터 값을 전송합니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 충전기는 미터 값을 CSMS로 오프로드하기 위해 MeterValuesRequest 메시지를 보냅니다. **2.** MeterValuesRequest 메시지를 수신하면 CSMS는 MeterValuesResponse 메시지로 응답합니다.
5	**필수 조건**	충전기는 XX초마다 미터 값을 보내도록 구성되어 있습니다. 실행 중인 트랜잭션이 없습니다.
6	**사후 조건**	**성공 사후 조건:** n/a **실패 사후 조건:** n/a

Figure 91. 시퀀스 다이어그램: 미터 값 전송

7	**오류 처리**	n/a

8	참고 사항	위상 필드는 모든 Measurands에 적용되지 않습니다. 그리드 연결에 대한 커넥터의 위상 회전은 그리드 연결에서 커넥터까지 체인의 모든 구성 요소의 PhaseRotation 구성 변수를 쿼리하여 파생할 수 있습니다. 각 sampledValue의 특성은 선택 사항인 Measurand, context, location, unit 및 phase 필드에 의해 결정됩니다. 선택 사항인 SignedMeterValue 필드에는 디지털 서명된 이진 미터 값 데이터가 포함될 수 있습니다.

J01 - Sending Meter Values not related to a transaction - Requirements

Table 169. J01 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
J01.FR.01		충전기는 에너지 미터(또는 다른 센서/트랜스듀서 하드웨어)를 샘플링하여 미터 값에 대한 추가 정보를 제공할 수 있습니다.	충전기가 미터 값을 언제 보낼지는 충전기에 달려 있습니다. 이는 SetVariablesRequest 메시지를 사용하여 데이터 수집 간격을 구성하고 수집 및 보고할 데이터를 지정할 수 있습니다.
J01.FR.02		MeterValuesRequest 메시지에는 샘플을 가져온 EVSE의 ID가 포함되어야 합니다.	
J01.FR.03	J01.FR.02 AND evseId는 0입니다.	MeterValuesRequest 메시지는 전체 충전기와 연결되어야 합니다.	
J01.FR.04	J01.FR.03 AND Measurand는 에너지와 관련이 있습니다.	샘플은 주 에너지 미터에서 가져와야 합니다.	
J01.FR.05	모두 동일한 시점에 캡처된 경우.	각 MeterValue 요소에는 타임스탬프가 포함되어야 합니다.	
J01.FR.06	모두 동일한 시점에 캡처된 경우.	각 MeterValue(s) 요소에는 하나 이상의 개별 SampledValue 요소 집합이 포함되어야 합니다.	
J01.FR.07		선택적 측정량 필드는 측정/보고되는 값의 유형을 지정해야 합니다.	
J01.FR.08		선택적 컨텍스트 필드는 판독을 트리거하는 이유/이벤트를 지정해야 합니다.	
J01.FR.09		선택적 위치 필드는 측정이 수행되는 위치를 지정해야 합니다.	(예: 입구, 출구).
J01.FR.10		선택적 위상 필드는 값이 적용되는 전기 설비의 위상 또는 위상을 지정해야 합니다.	
J01.FR.11		충전기는 전기 미터(또는 없는 경우 그리드 연결) 관점에서 모든 위상 번호 종속 값을 보고해야 합니다.	
J01.FR.13	구성 요소의 위상 회전을 보고할 때	충전기는 그리드 연결을 기준으로 위상 회전을 보고해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
J01.FR.14	AlignedDataCtrlr.Interval > 0 AND 측정량이 전송되는 EVSE가 트랜잭션에 관련되지 않은 경우	충전기는 모든 *evseId*, 위치 및 구성된 측정량이 지원되는 위상에 대해 모든 AlignedDataCtrlr.Interval에서 AlignedDataCtrlr.Measurands의 측정량에 대한 MeterValuesRequest 메시지를 CSMS에 전송해야 합니다.	특정 측정량이 모든 위치에서 사용 가능하지 않을 수 있습니다. 예를 들어, *evseId* = 0(그리드 미터)에는 "Current.Offered" 또는 "SoC" 측정값이 없습니다. J01.FR.22도 참조하세요.
J01.FR.15	J01.FR.14 AND 측정량 양이 1 MeterValuesRequest에 비해 너무 많습니다.	충전기는 모든 측정량을 보내기 위해 여러 MeterValuesRequest 메시지를 사용할 수 있습니다.	
J01.FR.17		MeterValue의 타임스탬프는 모든 SampledValues에 적용되어야 합니다.	
J01.FR.18	CSMS가 MeterValuesRequest를 수신하면	CSMS는 MeterValuesResponse로 응답해야 합니다.	MeterValuesResponse로 응답하지 못하면 충전기에서 동일한 메시지를 다시 시도할 수 있습니다.
J01.FR.19	EVSE에 대해 `AlignedDataSendDuringIdle` 이 true로 설정되고 지정된 EVSE에 진행 중인 트랜잭션이 있는 경우.	충전기는 이 EVSE에 대해 클록 정렬 미터 값 전송을 중지해야 합니다.	
J01.FR.20	충전기에 대해 `AlignedDataSendDuringIdle` 이 true로 설정되고 충전기에 진행 중인 트랜잭션이 있는 경우.	충전기는 모든 EVSE와 주 전력 미터에 대해 클록 정렬 미터 값 전송을 중지해야 합니다.	
J01.FR.21	`AlignedDataSignReadings`는 *true* 입니다.	충전기는 데이터 서명을 지원하는 구성 요소에서 서명된 미터 값을 검색하여 *signedMeterValue* 필드에 넣어야 합니다.	
J01.FR.22	AlignedDataCtrlr.Interval > 0 AND 측정량이 전송된 EVSE가 트랜잭션에 관여하는 경우	충전기는 다음 중 하나를 전송해야 합니다. - MeterValuesRequest 메시지 또는 - TransactionEventRequest(*triggerReason* = `Sample.Clock`) AlignedDataCtrlr.Measurands의 측정량에 대한 CSMS에 AlignedDataCtrlr.Interval마다 전송합니다.	J01.FR.14도 참조하세요.

J02 - Sending transaction related Meter Values

Table 170. J02 - 트랜잭션 관련 미터 값 전송

번호	유형	설명
1	**이름**	트랜잭션 관련 미터 값 전송
2	**ID**	J02
	기능 블록	J. 미터 값
3	**목표**	에너지 미터 또는 기타 센서/트랜스듀서 하드웨어를 샘플링하여 충전기의 트랜잭션 관련 미터 값에 대한 정보를 제공합니다.
4	**설명**	충전기는 에너지 미터 또는 기타 센서/트랜스듀서 하드웨어를 샘플링하여 트랜잭션 관련 미터 값에 대한 정보를 제공합니다. 구성 설정에 따라 충전기는 트랜잭션 중에 미터 값을 전송합니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 충전기는 TransactionEventRequest (eventType = Updated) 메시지를 보내 미터 값을 CSMS로 오프로드합니다. **2.** TransactionEventRequest 메시지를 수신하면 CSMS는 TransactionEventResponse 메시지로 응답합니다.
5	**필수 조건**	충전기는 XX초마다 미터 값을 보내도록 구성되어 있습니다. 트랜잭션이 실행 중입니다.
6	**사후 조건**	**성공 사후 조건:** n/a **실패 사후 조건:** n/a

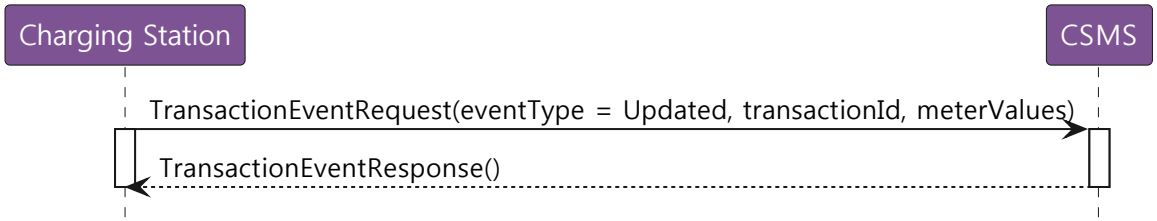

Figure 92. 시퀀스 다이어그램: 트랜잭션 관련 미터 값 전송

7	**오류 처리**	*Offline* 일 때, 충전기는 충전기가 온라인 상태였다면 CSMS에 전송했을 트랜잭션 관련 메시지(트랜잭션에 속하는 미터 값)를 큐에 넣어야 합니다.
8	**주의 사항**	위상 필드는 모든 Measurands에 적용되지 않습니다. 그리드 연결에 대한 Connector의 위상 회전은 그리드 연결에서 Connector까지 체인의 모든 구성 요소의 `PhaseRotation` 구성 변수를 쿼리하여 얻을 수 있습니다. 각 sampledValue의 특성은 선택적 Measurand, context, location, unit 및 phase 필드에 의해 결정됩니다. 선택적 SignedMeterValue 필드에는 디지털 서명된 이진 미터 값 데이터가 포함될 수 있습니다.

J02 - Sending transaction related Meter Values - Requirements

Table 171. J02 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
J02.FR.01		충전기는 에너지 미터(또는 다른 센서/트랜스듀서 하드웨어)를 샘플링하여 미터 값에 대한 추가 정보를 제공할 수 있습니다.	충전기가 미터 값을 보내는 시점은 충전기에 달려 있습니다. 이는 SetVariablesRequest 메시지를 사용하여 데이터 수집 간격을 구성하고 수집 및 보고할 데이터를 지정할 수 있습니다.
J02.FR.02	모두 동일한 시점에 캡처된 경우.	각 MeterValue 요소에는 하나 이상의 개별 SampledValue 요소 집합이 포함되어야 합니다.	
J02.FR.03		선택적 측정량 필드는 측정/보고되는 값의 유형을 지정해야 합니다.	
J02.FR.04		선택적 컨텍스트 필드는 판독을 트리거하는 이유/이벤트를 지정해야 합니다.	
J02.FR.05		선택적 위치 필드는 측정이 수행되는 위치를 지정해야 합니다.	(예: 입구, 출구).
J02.FR.06		선택적 위상 필드는 값이 적용되는 전기 설비의 위상 또는 위상을 지정해야 합니다.	
J02.FR.07		충전기는 전력계(또는 없는 경우 그리드 연결) 관점에서 모든 위상 번호 종속 값을 보고해야 합니다.	
J02.FR.09	구성 요소의 위상 회전을 보고할 때	충전기는 그리드 연결을 기준으로 위상 회전을 보고해야 합니다.	
J02.FR.10	TransactionEventRequest 메시지에 eventType = Started 또는 eventType = Update가 하나 이상의 *sampledValue* 요소가 있는 하나의 *meterValue* 가 아닌 여러 *meterValue* 요소가 포함된 경우	모든 *meterValue* 요소에는 현재 샘플링 간격 내에 있는 타임스탬프가 있어야 합니다. 즉, (트랜잭션 이벤트 타임스탬프 - `SampledDataTxUpdatedInterval`) < *meterValue.timestamp* ⇐ 트랜잭션 이벤트 타임스탬프	*eventType* = Ended 인 경우에만 TransactionEventRequest에 여러 간격에 대한 미터 값이 있을 수 있습니다.
J02.FR.11	`SampledDataTxUpdatedInterval` > 0일 때	충전기는 TransactionEventRequest(eventType = Updated)를 *triggerReason* = `MeterValuePeriodic` 로 전송해야 하며, 측정값은 SampledDataCtrlr.TxUpdatedInterval마다 MeterValue 필드에 SampledDataCtrlr.TxUpdatedMeasurands에 구성되어야 합니다.	SampledDataCtrlr.TxStartedMeasurands를 전송하려면 E01을 참조하고, SampledDataCtrlr.TxEndedMeasurands를 전송하려면 E06을 참조하세요.
J02.FR.12	J02.FR.11 AND *Offline* AND 충전기의 메모리가 부족합니다.	충전기는 TransactionEventRequest(eventType = Updated) 메시지를 삭제할 수 있습니다.	

ID	전제 조건	요구 사항 정의	참고
J02.FR.13	J02.FR.12	TransactionEventRequest(eventType = Updated) 메시지를 삭제할 때 충전기는 중간 메시지를 먼저 삭제해야 하며(첫 번째 메시지, 세 번째 메시지, 다섯 번째 메시지 등), 처음부터 메시지를 삭제하거나 큐에 메시지를 추가하는 것을 중단해서는 안 됩니다.	
J02.FR.14	J02.FR.11 AND 1 TransactionEventRequest(eventType = Updated)에 대한 미터 데이터 양이 너무 많습니다.	충전기는 모든 측정값을 보내기 위해 동일한 *timestamp* 를 사용하여 여러 TransactionEventRequest(eventType = Updated) 메시지를 사용할 수 있습니다.	
J02.FR.16		단일 충전 트랜잭션 또는 비 트랜잭션 소비자(예: 충전기 내부 전원 공급, 전체 공급)와 관련된 모든 "등록" 값은 시간에 따라 단조롭게 증가해야 합니다.	미터 교체의 경우는 제외합니다. MeasurandEnumType을 참조하세요.
J02.FR.17		추적성을 향상시키기 위해, ".Register" 값은 전력 계량 하드웨어의 비휘발성 레지스터에서 직접 읽어온 그대로 보고되어야 하며, 트랜잭션 시작 시 0으로 재설정(re-base)되어서는 안 됩니다.	이렇게 하면 하드웨어 오류, 미터 교체, 오배선, 사기 등으로 인해 연속 트랜잭션 간에 "누락된 에너지"를 식별할 수 있으며, CSMS가 모든 트랜잭션의 시작 레지스터 값이 동일한 커넥터에서 이전 트랜잭션의 완료 레지스터 값과 동일함을 확인할 수 있습니다.
J02.FR.18		MeterValue의 타임스탬프는 모든 SampledValues에 적용해야 합니다.	
J02.FR.19	CSMS가 TransactionEventRequest를 수신하는 경우	CSMS는 TransactionEventResponse로 응답해야 합니다.	TransactionEventResponse로 응답하지 못하면 충전기가 동일한 메시지를 다시 시도할 수 있습니다.
J02.FR.20	TransactionEventRequest(eventType = Ended)에 미터 데이터를 보내도록 구성 한 경우 AND 미터 데이터 양이 하나의 TransactionEventRequest(eventType = Ended) 메시지에 포함되기에 양이 너무 많습니다.	충전기는 메시지에 맞을 때까지 샘플을 제거할 수 있습니다. 샘플을 제거할 때 충전기는 중간 샘플을 먼저 제거해야 합니다(예: 2번째 샘플, 4번째 샘플, 6번째 샘플 등).	샘플은 청구에 영향을 미치지 않는 방식으로 제거해야 합니다. E06.FR.12도 참조하세요.
J02.FR.21	`SampledDataSignReadings`는 *true* 입니다.	충전기는 데이터 서명을 지원하는 구성 요소에서 서명된 미터 값을 검색하여 *signedMeterValue* 필드에 입력해야 합니다.	
J02.FR.22		TransactionEventRequest 메시지에서 보고된 미터 값은 모두 트랜잭션이 발생하는 EVSE와 관련이 있어야 합니다.	

3.2. ISO 15118 MeterValue signing

J03 - Charging Loop with metering information exchange

Table 172. J03 - 계량 정보 교환이 있는 충전 루프

번호	유형	설명
1	**이름**	계량 정보 교환이 있는 충전 루프
2	**ID**	J03
	기능 블록	J. 미터 값
	참조	ISO15118-1 F1
3	**목표**	ISO15118-1 참조, 사용 사례 목표 F1, 37페이지.
4	**설명**	ISO15118-1 참조, 사용 사례 설명 F1, 37페이지.
5	**필수 조건**	- 기능 블록 C의 사용 사례에 따른 권한이 적용되는 경우 성공적으로 완료되어야 합니다. ISO15118-1, 사용 사례 전제 조건 F1, 37페이지를 참조하세요.
6	*Actors*	EV, EVSE, 충전기
7	**통합 시나리오 설명**	**15118** **1a.** EV가 충전기에 ChargingStatusReq(AC 충전의 경우) 메시지를 보내고, EVSE가 재정 미터의 미터 값이 포함된 ChargingStatusRes를 반환합니다. **1b.** EV가 충전기에 CurrentDemandReq(DC 충전의 경우) 메시지를 보내고, EVSE가 재정 미터의 미터 값이 포함된 CurrentDemandRes를 반환합니다. **2.** EV가 충전기에 MeteringReceiptReq를 보내 미터 값 수신을 확인합니다.
8	**사후 조건**	ISO15118-1 참조, 사용 사례 종료 조건 F1, 37페이지.

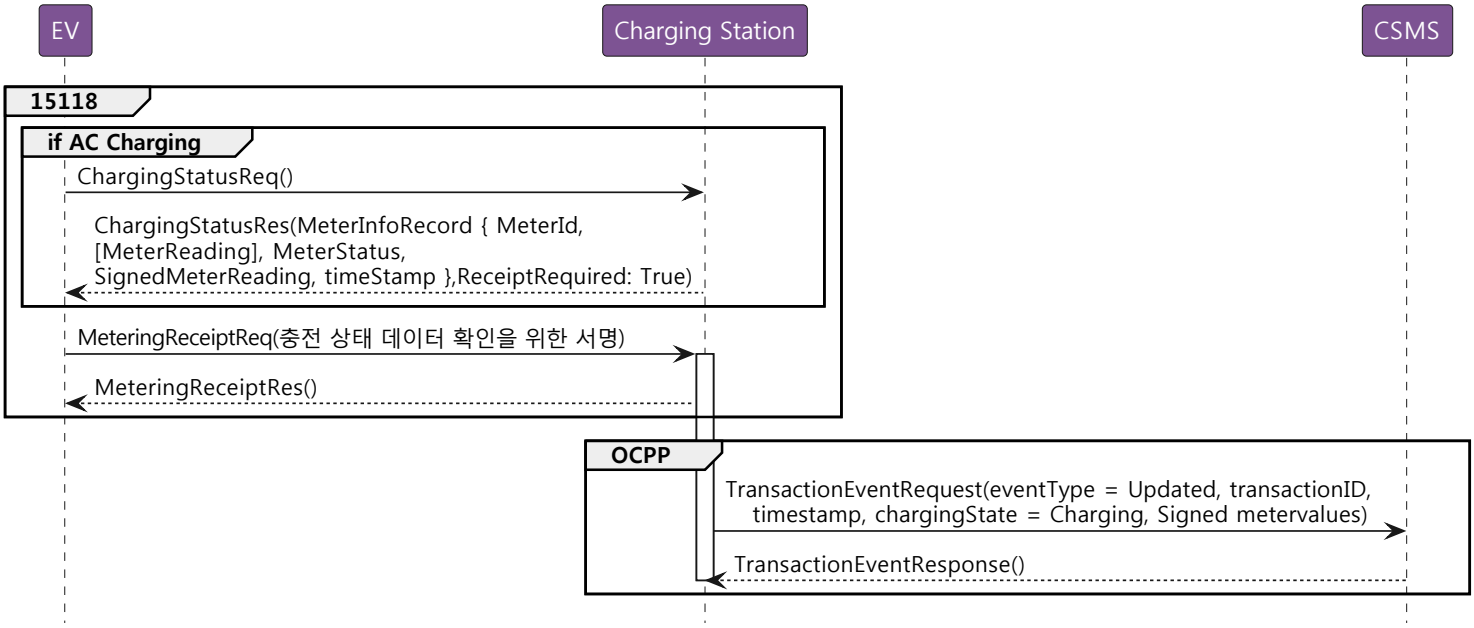

Figure 93. 측정 정보 교환이 있는 충전 루프

9	**오류 처리**	n/a

10	**비고**	n/a

J03 - Charging Loop with metering information exchange - Requirements

Table 173. J03 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
J03.FR.04	충전기가 EV로부터 ISO 15118로 서명된 MeteringReceiptReq 메시지를 수신할 때	충전기는 MeteringReceiptReq 메시지의 미터 값을 TransactionEventRequest (eventType = Updated) 메시지로 CSMS에 전달해서는 안 됩니다. 대신 충전기는 사용 사례 J02에 설명된 대로 트랜잭션 관련 미터 값을 보냅니다.	이는 충전기가 EV에 MeteringReceiptReq 메시지를 보내도록 요구할 수 없다는 것을 의미하지 않습니다. 충전기에서의 구현은 재정 미터(fiscal meter)에서 CSMS로 전송된 모든 미터 값(사용 사례 J02에 따라)이 먼저 EV의 MeterReceiptReq에 의해 확인되어야 하는 방식일 수 있습니다.

K. SmartCharging

1. Introduction

이 기능 블록은 CSO(또는 제3자)가 트랜잭션 중에 전송되는 충전 전류/전력에 영향을 미치거나 충전기가 그리드에서 끌어올 수 있는 전류/전력량에 대한 제한을 설정할 수 있는 모든 기능을 설명합니다.

일반적으로 스마트 충전에는 두 가지 이상의 정의가 있습니다. 이는 소비자가 대규모 소비자 그룹이 동시에 '충전'하기를 원하더라도 언제든지 배터리를 완전히 충전할 수 있는 방식으로 그리드 용량을 사용한다는 것을 의미할 수 있습니다. 스마트는 충전 시 에너지 가격을 고려할 수 있다는 것을 의미할 수도 있고, 또는 태양광 패널에서 지속 가능한 에너지를 지역적으로 공급하는 것으로 간주될 수 있습니다. 그리고 전기 자동차(EV) 운전자가 솔루션의 일부가 되기를 원할 때 더욱 '스마트'한것 일겁니다. OCPP 내에서 스마트 충전은 CSMS가 특정 EV의 (충전) 전력 또는 전류의 충전(방전)에 영향을 미치거나, 전체 충전기/충전기 그룹의 총 허용 에너지 소비에 영향을 미칠 수 있는 기능을 얻는 것을 의미합니다. 다양한 설정을 사용할 수 있습니다. 다음의 OCPP를 사용한 스마트 충전의 네 가지 일반적인 스마트 충전 동작을 설명하는 유형입니다.

- 내부 부하 분산
- 중앙 스마트 충전
- 로컬 스마트 충전
- 외부 스마트 충전 제어 신호

이러한 유형은 Types of Smart Charging에서 설명합니다. 물론, 위의 두 가지 이상의 사용 사례가 하나의 더 복잡한 시스템으로 결합되는 더 복잡한 사용 사례도 가능합니다.

NOTE ISO 15118 및 OCPP 용어 매핑은 ISO 15118 and OCPP terminology mapping에 제공됩니다.

2. Types of Smart Charging

이 섹션은 정보 제공을 목적으로 합니다.

2.1. Internal Load Balancing

가장 간단한 형태의 스마트 충전은 부하 분산 사용 사례입니다. 이는 충전기 내에서 내부 부하 분산과 관련이 있으며, 충전기는 EVSE당 전류/전력을 제어합니다. 충전기는 고정된 한도(예: 그리드 연결의 최대 전류)로 구성됩니다. 이 경우 충전기는 모든 EVSE에 대한 충전을 최적화할 책임이 있습니다. 충전기가 그리드에 직접 연결되지 않은 경우, 고객의 에너지 시스템이 전력 공급을 담당합니다.
이 설정은 일반적으로 알려진 물리적 한계로 인해 필요한 한계를 설정하는 데 사용됩니다.

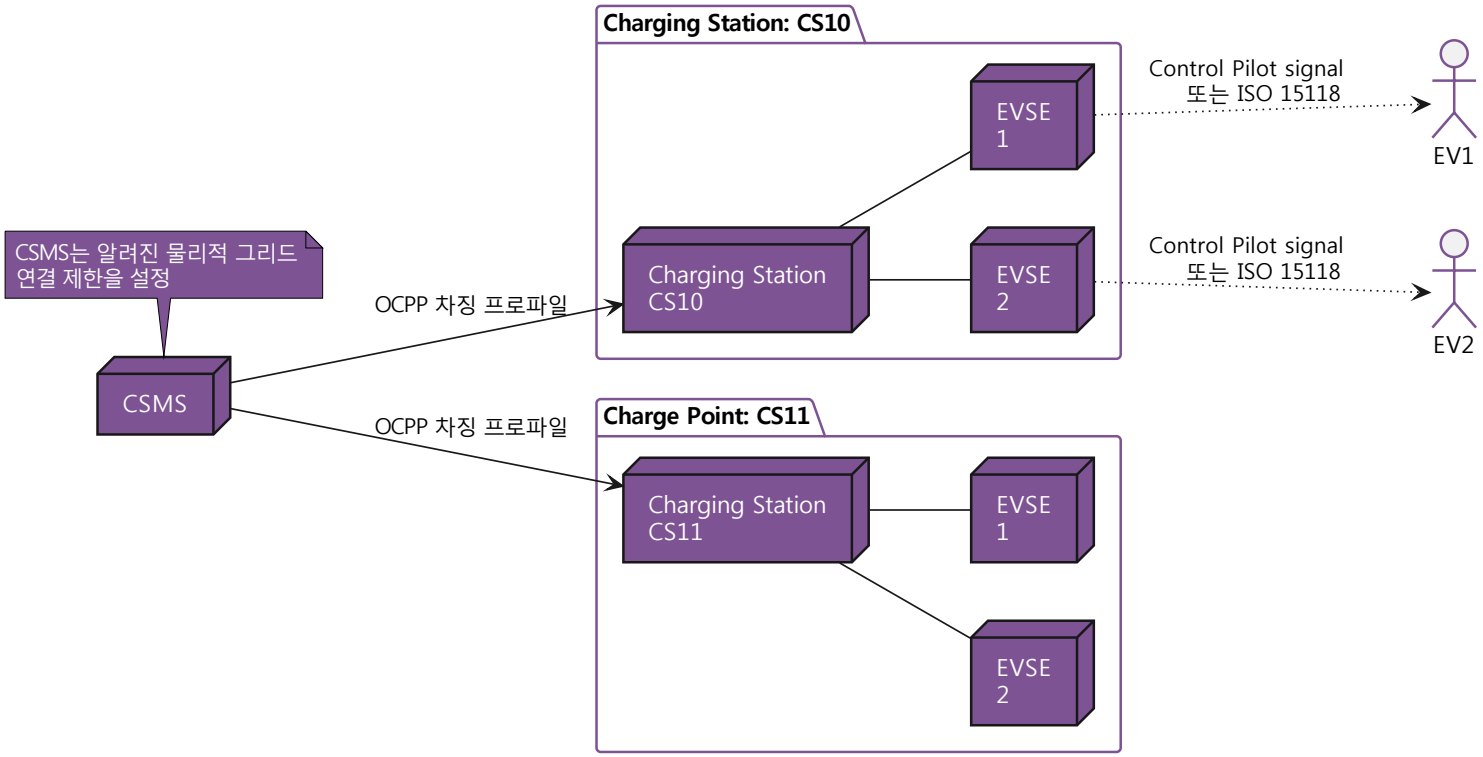

Figure 94. 내부 부하 분산 스마트 충전 토폴로지

2.2. Central Smart Charging

스마트 충전의 다음 단계는 CSMS가 특정 EV의 충전 전력 또는 전류, 전체 충전기 또는 충전기 그룹의 총 허용 에너지 소비에 영향을 미칠 수 있는 기능입니다. Central Smart Charging은 충전 한도가 CSMS에 의해 제어된다고 가정합니다. 예를 들어, 이는 그리드 연결, 그리드의 에너지 가용성(예: 그리드 운영자(DSO)의 용량 예측) 또는 건물의 배선을 기반으로 할 수 있습니다. 이 설정에서 CSMS는 하나의 충전기뿐만 아니라 한 단계 "위"에서 충전을 최적화할 수 있습니다. 즉, 연결을 공유하는 두 개 이상의 충전기를 최적화하여 충전을 위한 보다 효율적인 일정을 계산할 수 있습니다.

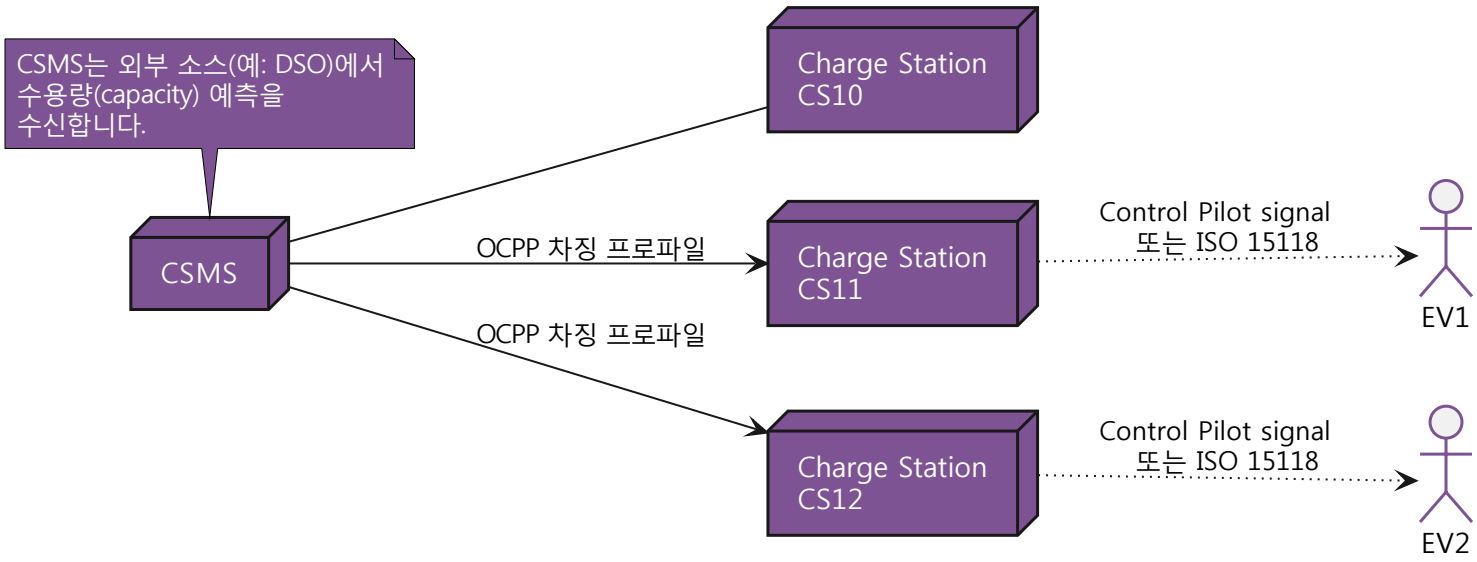

Figure 95. 중앙 스마트 충전 토폴로지

중앙 스마트 충전은 제어 파일럿 신호로 수행할 수 있지만, EV가 제어 파일럿 신호를 통해 충전 요구 사항을 전달할 수 없기 때문에 몇 가지 제한이 있습니다. Local Smart Charging 사용 사례와 유사하게 EVSE는 제어 파일럿 신호로 충전 일정을 실행할 수 있습니다.

2.3. Local Smart Charging

로컬 스마트 충전은 스마트 충전이 가능한 충전기가 CSMS가 아닌 로컬 컨트롤러에 의해 로컬에서 제어되는 충전 한도를 갖는 사용 사례를 설명합니다. 이 유형의 스마트 충전은 충전기 그룹을 제어하는 논리적 구성 요소인 로컬 컨트롤러가 있다고 가정합니다. 일반적인 사용 사례는 그리드 연결 정격이 충전기 정격의 합보다 작은 주차장에 있는 여러 충전기입니다. 또 다른 응용 분야는 로컬 컨트롤러가 DSO 또는 로컬 스마트 그리드 노드에서 전력 가용성에 대한 정보를 수신하는 것입니다.

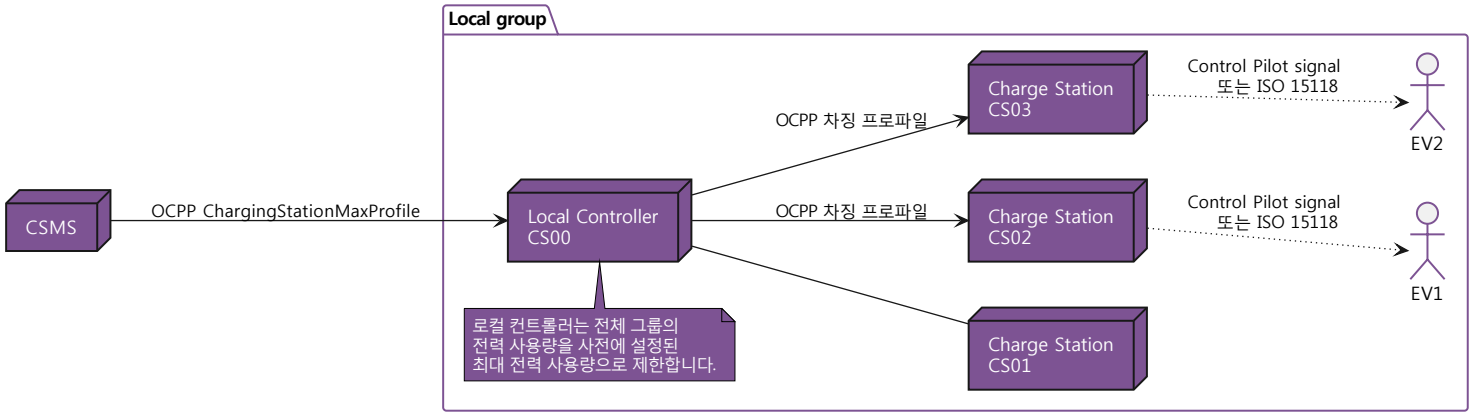

Figure 96. 로컬 스마트 충전 토폴로지

2.4. External Smart Charging Control Signals

OCPP 프로토콜은 원래 CSMS와 하나 이상의 충전기 간의 통신을 위해 개발되었습니다. 위에서 설명한 대로 이는 충전기 운영자(CSO) CSMS가 충전기를 제어하고 EV와 충전기의 충전 한도에 따라 CSO가 EV가 얼마나 빨리 충전되는지 결정한다는 것을 의미합니다. 그러나 OCPP 지원 충전기의 일부 상황/애플리케이션에서는 충전 속도를 결정하는 두 가지 요소가 이것들뿐이 아닙니다. 충전 속도를 결정하는 다른 입력은 DSO 신호(예: IEC 61850 [IEC61850-7-420], IEC 60870 [IEC60870-5-104], DNP3 [DNP3] 또는 OpenADR [OPENADR]) 또는 건물/가정 에너지 관리 시스템의 신호일 수 있습니다. 이러한 신호는 OCPP의 범위를 벗어나지만, OCPP 관점에서 보면 CSMS가 외부 신호에 의한 충전 변경 사항을 통보받아야 한다는 것은 분명해 보입니다. 그러나 이는 충돌하는 신호를 처리하는 방법과 같은 여러 가지 의문으로 이어집니다. 아래 그림은 에너지 관리 시스템을 사용한 예시 설정을 보여줍니다. 여기서 외부 신호는 충전기와 직접 통신하는 설정과 로컬 컨트롤러를 사용하는 여러 충전기 설정 모두에서 시각화됩니다.

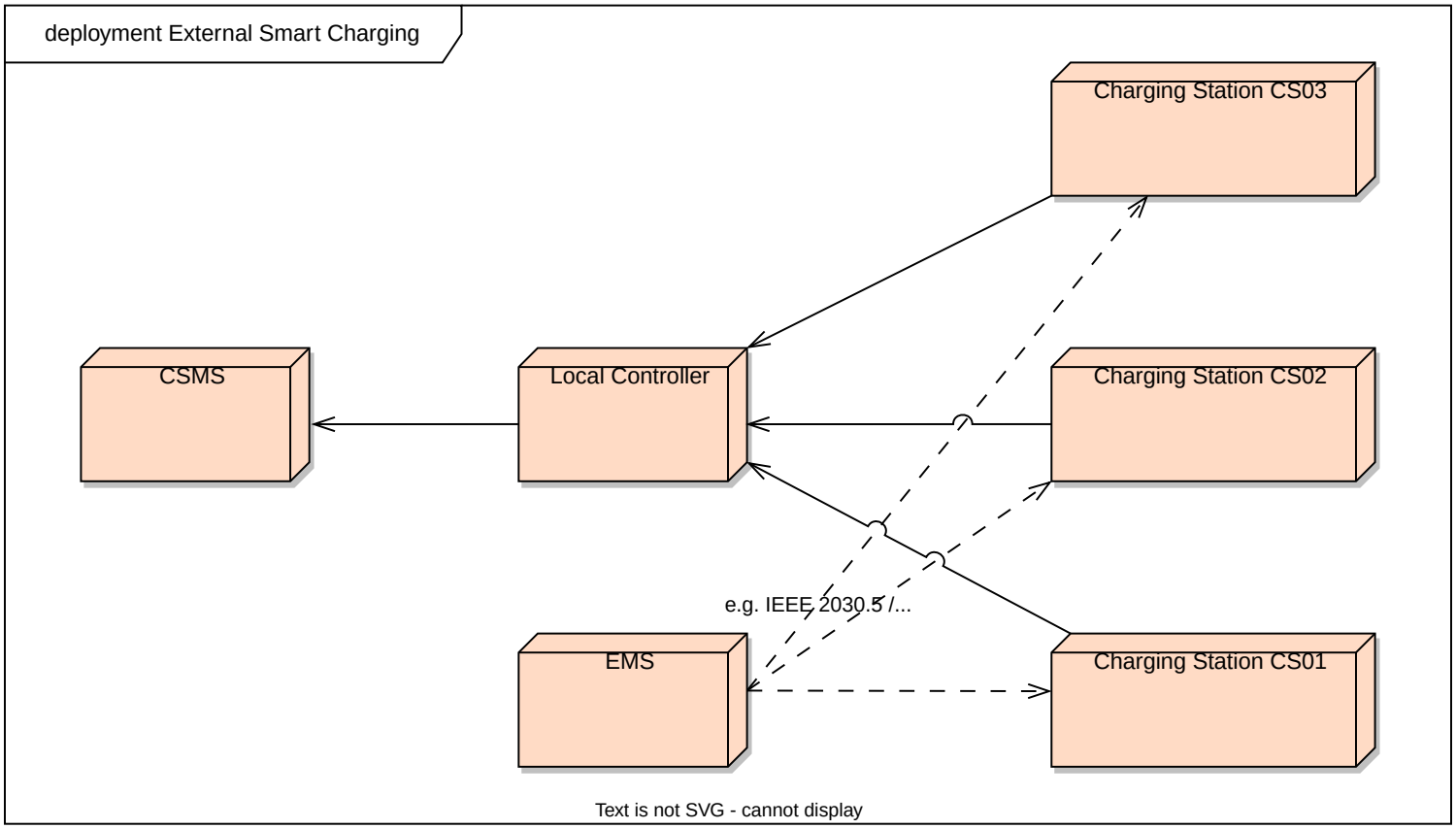

Figure 97. 외부 스마트 충전

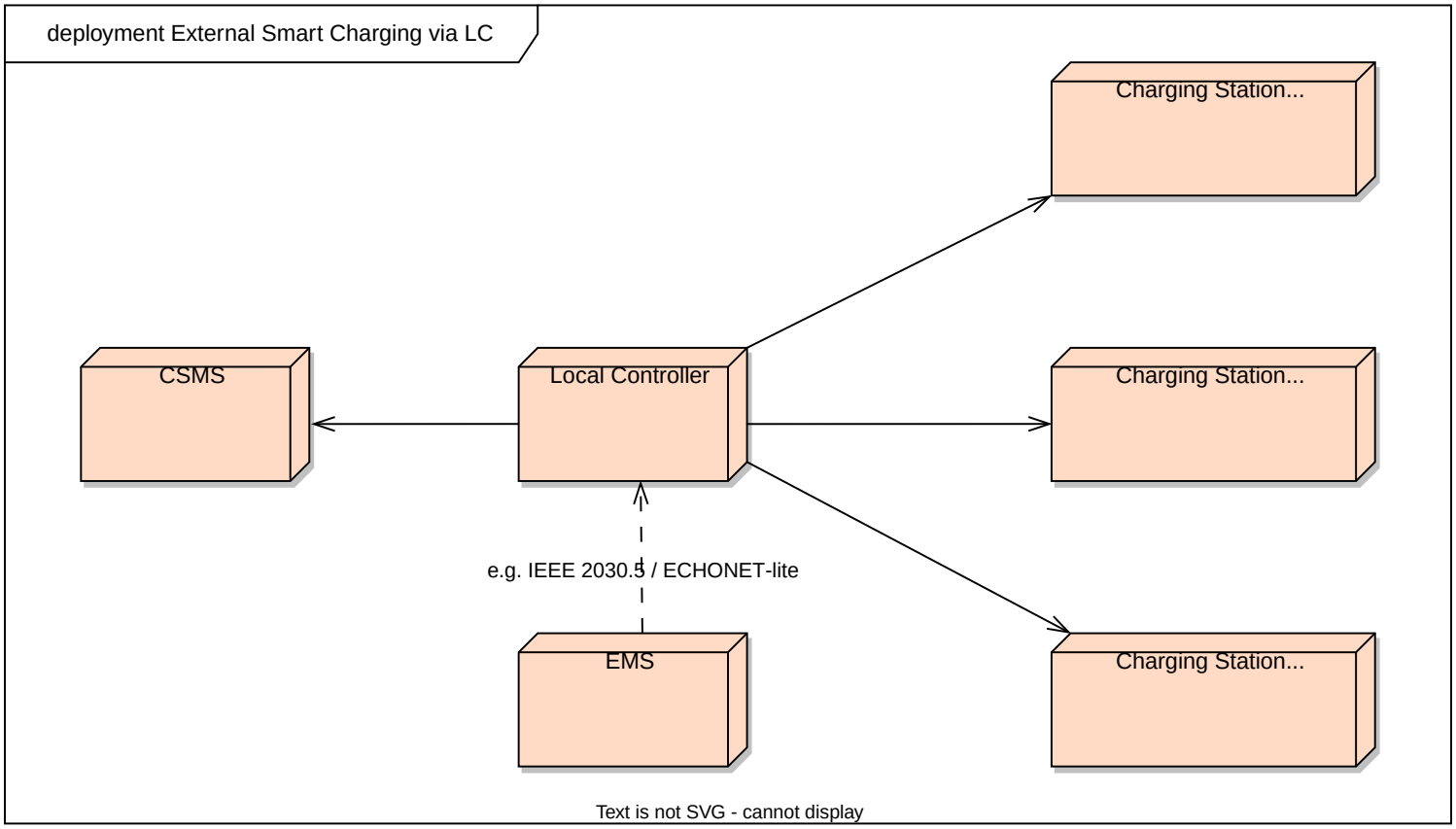

Figure 98. LC를 통한 외부 스마트 충전

충전기가 외부 세계와 에너지 관리 시스템(EMS)에 모두 연결되어 있는 경우, EMS가 CSMS에서 수신한 충전 일정에도 불구하고 어떤 이유로든 충전이 적절하지 않다고 판단하는 상황이 발생할 수 있습니다. 이는 충전기가 CSMS에서 예상한 대로 작동하지 않음을 의미합니다. 이를 방지하기 위해 충전기는 EMS에서 명령을 받았다는 것을 CSMS에 알릴 수 있어야 합니다. 예를 들어 가정 사용자가 EV를 충전하는 대신 에어컨 시스템에 우선권/우선순위를 부여하는 경우가 있습니다(이 경우 에어컨과 EV 충전을 동시에 사용할 수 없다고 가정). 이 EMS는 연결의 최대 한도를 관리하기 위해 배치될 수 있지만 외부에서 제어할 수도 있습니다.

3. Charging Profiles

3.1. Introduction

충전 전력 또는 전류에 영향을 미치는 것은 특정 시점에 충전기로 에너지 전송 한도를 보내는 것에 기반합니다. 이러한 한도는 ChargingProfile에 결합되어 있습니다. ChargingProfile은 충전 전력 또는 전류 한도 블록을 정의하고 시작 시간과 기간을 포함할 수 있는 ChargingSchedule을 가지고 있습니다. 이는 충전기와 충전기의 EVSE에 적용될 수 있습니다. Example ChargingProfile에는 충전 프로파일을 사용하는 방법을 설명하기 위해 ChargingProfile의 예시가 주어집니다.

CSMS는 다음과 같은 상황에서 메시지 SetChargingProfileRequest를 사용하여 충전기에 충전 프로파일을 보낼 수 있습니다.

- 트랜잭션 시작 시 트랜잭션에 대한 충전 프로파일을 설정
- 충전기에 RequestStartTransaction 요청이 전송 될때
- 트랜잭션 중 트랜잭션에 대한 활성 프로파일을 변경
- 트랜잭션 컨텍스트 외에 별도의 메시지로 로컬 컨트롤러, 충전기 또는 EVSE에 대한 기본 충전 프로파일을 설정

3.2. Charging profile purposes

이 섹션에서는 OCPP에서 지원되는 여러 유형의 충전 프로파일을 설명합니다. 충전 프로파일에는 ***목적*** 에 따라 네 가지 유형이 있습니다.

ChargingProfile 목적	**설명**
ChargingStationMaxProfile	내부 부하 분산 시나리오에서 충전기에는 충전기의 모든 EVSE가 공유할 전력 또는 전류를 제한하는 하나 이상의 로컬 충전 프로파일이 있습니다. CSMS는 이러한 프로파일을 ChargingProfilePurpose를 "*ChargingStationMaxProfile*"로 설정하여 구성해야 합니다. *ChargingStationMaxProfile* 은 충전기 evseId 0에서만 설정할 수 있습니다.
TxProfile	TxProfile이라는 목적이 있는 트랜잭션별 프로파일은 어느 쪽이 먼저 발생하는지에 따라 현재 트랜잭션 기간 동안만 TxDefaultProfile을 무시하거나 TxProfile이 만료될 때까지 무시합니다.
TxDefaultProfile	충전 정책을 부과하는 데 사용될 수 있는 새 트랜잭션에 대한 기본 일정입니다. 예를 들어 낮 동안 충전을 하지 않는 정책이 있습니다.
ChargingStationExternalConstraints	CSMS가 아닌 외부 시스템이 충전 한도 또는 일정을 설정하면 충전기는 이 목적을 사용하여 해당 한도/일정을 보고합니다.

3.3. Charging profile recurrency

이 섹션에서는 속성 *chargingProfileKind* 의 값에 따라 정의된 충전 프로파일에서 사용할 수 있는 다양한 종류의 충전 일정을 설명합니다.

ChargingProfile Kind	**설명**
Absolute	충전 일정 기간은 일정에 정의된 절대 시점을 기준으로 합니다. 이를 위해서는 *startSchedule* 이 시작 시점으로 설정되어야 합니다. 예를 들어 충전 세션이 시작된 시점과 관계없이 17:00h와 21:00h 사이에 충전을 줄이는 일정을 사용할 때 사용할 수 있습니다.
Recurring	충전 일정은 첫 번째 일정 기간에 주기적으로 다시 시작됩니다. 가장 유용하려면 *startSchedule* 이 시작 시점으로 설정되어야 합니다. *recurrencyKind* = `Daily` 와 함께 사용하세요. 예를 들어, 충전 세션이 시작된 시점과 관계없이 매일 17:00h와 21:00h 사이에 충전을 줄이는 일정을 정의할 때 사용됩니다.

ChargingProfile Kind	설명
Relative	충전 일정 기간은 EVSE가 에너지를 공급할 준비가 되면 시작해야 합니다. 즉, EV 운전자가 인증되고 EV가 연결될 때입니다. 이미 충전 중인 트랜잭션에 대한 ChargingProfile을 수신하면 충전 일정 기간은 PowerPathClosed 순간까지 relative하게 유지되어야 합니다. startSchedule에 대한 값을 제공해서는 안 됩니다.

3.4. Stacking charging profiles

복잡한 달력을 설명하기 위해 동일한 ChargingProfile 목적의 충전 프로파일을 스태킹할 수 있습니다. 예를 들어, TxDefaultProfile의 목적인 ChargingProfile을 정의하여 주중 23:00h에서 06:00h까지, 그리고 주말 00:00h에서 24:00h까지 최대 전력으로 충전을 허용하고 다른 시간에는 전력 또는 전류 충전을 줄이는 한 주의 기간과 반복을 지정할 수 있습니다. 그 위에 다른 TxDefaultProfiles를 정의하여 휴일과 같이 이 규칙에 대한 예외를 정의할 수 있습니다.

ChargingProfile은 특정 시간 간격에 대한 제한을 정의하는 ChargingSchedule을 보유합니다. ChargingSchedule의 우선순위는 ChargingProfile의 *stackLevel* 에 따라 결정됩니다. 동일한 chargingProfilePurpose를 갖는 두 개 이상의 ChargingProfile이 유효한 경우, 스택 수준이 더 높은 ChargingProfile의 ChargingSchedule이 스택 수준이 더 낮은 ChargingProfile의 ChargingSchedule을 무효화합니다.

충돌을 피하기 위해 동일한 *stackLevel* 과 동일한 *chargingProfilePurpose* 를 가진 여러 개의 충전 프로파일이 주어진 시간에 동일한 EVSE에서 유효하도록 설정하는것을 허용하지 않습니다. EVSE #0의 충전 프로파일은 모든 EVSE에서 활성화된 것으로 간주됩니다!

3.5. Combining Charging Profile Purposes

충전 수준을 안내하는 복합 일정은 다양한 chargingProfilePurposes의 기존 충전 프로파일과 스택 수준의 조합입니다.

앞서 언급했듯이 각 충전 프로파일 목적에 대해 언제든지 해당 목적에 대한 선행(leading) 충전 일정은 해당 시간에 정의된 일정 기간이 있고 *validFrom* 및 *validTo* 매개변수에 따라 결정되는 해당 시간에 유효한 가장 높은 스택 수준을 가진 충전 프로파일에 속하는 충전 일정입니다. 복합 일정은 각 시간 간격에 대한 다양한 목적의 주요 프로파일 중에서 가장 낮은 충전 한도(다른 chargingRateUnits 고려)를 고려하여 계산됩니다.

유일한 예외는 TxDefaultProfiles와 TxProfile이 모두 유효한 경우입니다. 이 경우 TxProfile이 항상 TxDefaultProfiles보다 우선하므로 복합 일정은 이 특정 상황에서 주요 목적 프로파일 TxDefaultProfiles을 고려하지 않습니다. 시간 간격은 고정된 길이일 필요가 없으며 모든 ChargingProfile 목적에 대해 동일할 필요도 없습니다. 즉, 결과적으로 생성된 복합 일정에는 길이가 다른 간격이 포함될 수 있습니다.

충전기에 두 개 이상의 EVSE가 장착된 경우 ChargingStationMaxProfile의 한계 값은 모든 EVSE를 합친 한계입니다.

아래 두 그림은 서로 다른 stackLevels 및 Purposes를 가진 여러 충전 프로파일을 결합하는 예를 제공하는 데 사용됩니다.

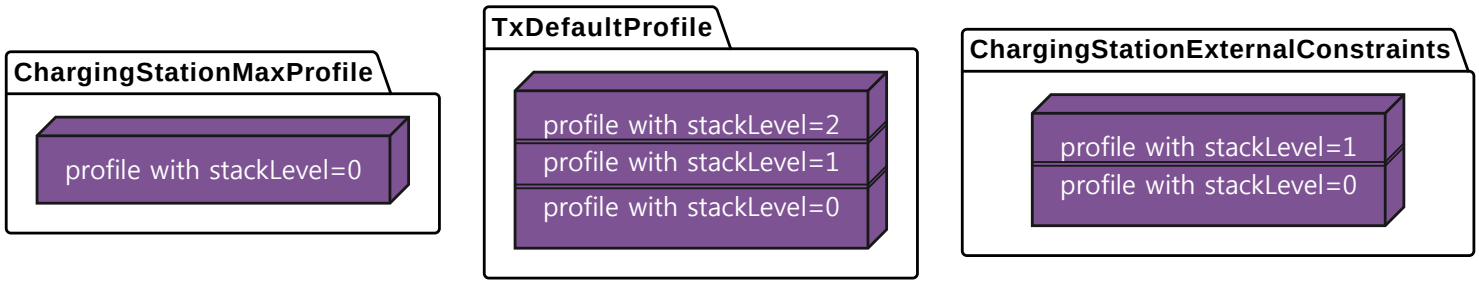

Figure 99. 다수의 유효한 충전 프로파일 - 상황 1

특정 시간 간격에서 유효한 충전 프로파일이 위의 그림(상황 1)과 같다고 가정합니다. 이 시간 간격의 복합 일정은 stackLevel 0인 ChargingStationMaxProfile, stackLevel 2인 TxDefaultProfile, stackLevel 1인 ChargingStationExternalConstraints 프로파일에 주어진 충전 한도 중 가장 낮은 것이 됩니다.

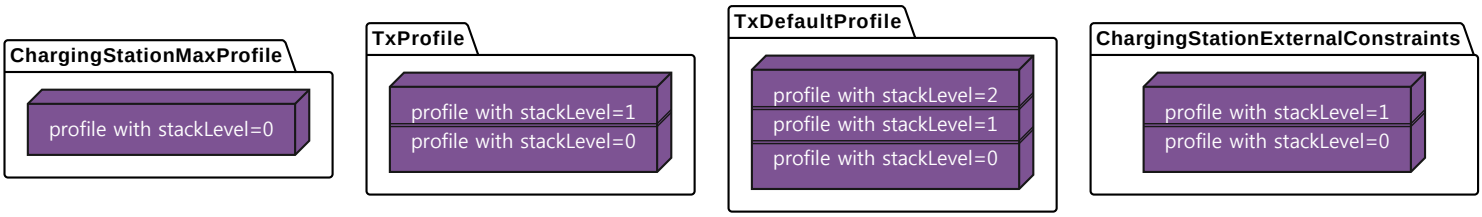

Figure 100. 다수의 유효한 충전 프로파일 - 상황 2

반면에, 특정 시간 간격에 대해 유효한 충전 프로파일이 위 그림과 같은 상황(상황 2)인 상황을 고려해 보겠습니다. 이 시간 간격의 복합 일정은 StackLevel 0인 ChargingStationMaxProfile, StackLevel 1인 TxProfile, StackLevel 1인 ChargingStationExternalConstraints 프로파일에 제공된 충전 한도 중 가장 낮은 값이 됩니다. 이 상황에서 TxProfile이 TxDefaultProfiles보다 우선합니다.

3.6. Example Charging Profile

이 섹션은 정보 제공을 목적으로 합니다.

다음 데이터 구조는 08:00h~20:00h 사이에 전력을 6kW로 제한하고 00:00h~08:00h와 20:00h~00:00h 사이에 전력을 11kW로 제한하는 일일 기본 프로파일을 설명합니다.

ChargingProfile			
chargingProfileId	**100**		
stackLevel	**0**		
chargingProfilePurpose	**TxDefaultProfile**		
chargingProfileKind	**Recurring**		
recurrencyKind	**Daily**		
chargingSchedule	*(1개의 ChargingSchedule 요소 목록)*		
	ChargingSchedule		
	duration	**86400 (= 24시간)**	
	startSchedule	**2013-01-01T00:00Z**	
	chargingRateUnit	**W**	
	chargingSchedulePeriod	(3개의 ChargingSchedulePeriod 요소 목록)	
		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

IMPORTANT	사용되는 위상의 양 충전은 충전기, EV, CS와 EV 간 케이블의 기능에 따라 제한됩니다. 이 세 가지 중 하나라도 3상 충전이 불가능한 경우 EV는 세 가지 모두에서 지원하는 위상 수를 사용하여 충전됩니다.
IMPORTANT	일정이나 트랜잭션 중에 사용되는 위상 수를 전환하는 것은 신중하게 수행해야 합니다. 일부 EV는 이를 지원하지 않을 수 있으며 위상 수를 변경하면 물리적 손상이 발생할 수 있습니다. 구성 변수: Phases3to1 으로 충전기가 트랜잭션 중에 위상 수를 전환하는 것을 지원하는지 여부를 알 수 있습니다.
TIP	일광 절약 시간이 적용되거나 적용되지 않는 날에는 특수 프로파일이 필요할 수 있습니다(예: relativeprofiles의 경우).

3.6.1. Example Using Stacked Charging Profiles

CSO가 17:00h에서 20:00h까지의 피크 시간 동안 충전을 2kW로 제한하고자 합니다. 이 제한은 일요일에는 적용되지 않으며 크리스마스에도 적용되지 않습니다.

이것이 많은 수의 충전기에 적용되는 경우 매주 일요일에 충전 프로파일을 삭제한 다음 월요일에 다시 추가하는 것은 실용적이지 않습니다. 가능한 해결책은 기본 프로파일에 대한 예외에 대해 더 높은 스택 레벨이 있는 프로파일을 추가하는 것입니다. 스택 레벨 #2와 #3이 일요일과 크리스마스에 대한 예외를 정의하는 데 사용되는 다음 JSON 예를 참조하세요.

(1) TxDefaultProfile, 스택 #1: 시간대 제한을 2kW로, 매일 17:00h에서 20:00h까지 반복합니다.

```
"chargingProfile": {
  "id": 10,
  "stackLevel": 1,
  "chargingProfilePurpose": "TxDefaultProfile",
  "chargingProfileKind": "Recurring",
  "recurrencyKind": "Daily",
  "chargingSchedule": [ {
    "id": 1,
    "startSchedule": "2020-01-09T17:00:00",
    "duration": 1080,
    "chargingRateUnit": "W",
    "chargingSchedulePeriod": [ { "startPeriod": 0, "limit": 2000 } ]
  } ]
}
```

(2) TxDefaultProfile, 스택 #2: 일요일을 무제한으로 무효화하고 매주 반복 2020-01-05.

```
"chargingProfile": {
  "id": 11,
  "stackLevel": 2,
  "chargingProfilePurpose": "TxDefaultProfile",
  "chargingProfileKind": "Recurring",
  "recurrencyKind": "Weekly",
  "chargingSchedule": [ {
    "id": 1,
    "startSchedule": "2020-01-05T00:00:00",
    "duration": 86400,
    "chargingRateUnit": "W",
    "chargingSchedulePeriod": [ { "startPeriod": 0, "limit": 999999 } ]
  } ]
}
```

(3) TxDefaultProfile, 스택 #3: 2020년 크리스마스 데이를 무제한으로 변경하고, 날짜를 2020-12-25로 고정. 이 프로파일은 2020년에만 유효합니다.

```
"chargingProfile": {
  "id": 12,
  "stackLevel": 3,
  "chargingProfilePurpose": "TxDefaultProfile",
  "chargingProfileKind": "Absolute",
  "validFrom": "2020-01-01T00:00:00",
  "validTo": "2021-01-01T00:00:00",
  "chargingSchedule": [ {
    "id": 1,
    "startSchedule": "2020-12-25T00:00:00",
    "duration": 86400,
    "chargingRateUnit": "W",
    "chargingSchedulePeriod": [ { "startPeriod": 0, "  ": 999999 } ]
  } ]
}
```

NOTE	일반적으로 충전에 대한 제한이 필요하지 않은 경우 해당 시간에 대한 충전 일정 기간을 정의하지 않습니다(17:00h - 20:00h 이외의 시간에 대한 스택 수준 #1 참조). 그러나 더 높은 스택 수준이 있는 프로파일에서 충전 일정을 1만큼 무시하는 경우 제한이 없는 충전 일정 기간을 정의할 수 없습니다. 따라서 위의 예에서 스택 #2 및 #3의 충전 일정은 999999의 (임의의) 높은 값을 사용합니다.

4. Smart Charging Signals to a Charging Station from Multiple Actors

이 섹션은 규범적입니다.

OCPP 내에서 스마트 충전을 위한 여러 메커니즘이 지원됩니다. 즉, EV를 충전할 때 제한을 추가할 수 있는 여러 메커니즘을 사용할 수 있습니다.

1. CSMS는 충전기에 SetChargingProfile 메시지를 보내 충전에 영향을 미칠 수 있습니다. K01 - SetChargingProfile을 참조하세요.
2. EV는 PlugAndCharge 기능에 따라 충전에 영향을 미칠 수 있습니다. ISO 15118은 EV가 시작한 충전 제한을 활성화합니다. 5.3. ISO 15118 based Smart Charging 섹션을 참조하세요.
3. HEMS(Home Energy Management System) 또는 DSO와 같은 일부 로컬 입력은 외부 스마트 충전 제어 신호를 통해 충전에 영향을 미칠 수 있습니다. K11 - Set / Update External Charging Limit를 참조하세요.
4. 충전기는 2대 이상의 EV가 충전 중일 때 부하 분산을 수행할 때 충전을 제한할 수 있습니다.

EV 충전 제한을 설정하는 데 관여할 수 있는 모든 당사자가 위의 메커니즘 중 하나를 직접 또는 간접적으로 사용한다는 가정이 있습니다.

충전기가 여러 주체의 동시 스마트 충전 신호에 어떻게 대응해야 하는지 결정하려면 다음 규칙을 따라야 합니다.

Table 174. 여러 주체 상황에 대한 스마트 충전 규칙

ID	전제 조건	요구 사항 정의	참고
SC.01		어떤 시점에서든 외부 소스의 일정과 각 목적의 ChargingStationMaxProfile, ChargingStationExternalConstraints 및 TxDefaultProfile(또는 TxProfile)에서 가장 높은 stackLevel을 가진 OCPP 충전 프로파일을 병합한 결과인 충전 한도는 병합된 일정에서 사용 가능한 전력 또는 전류의 가장 낮은 값보다 작거나 같아야 합니다.	이는 안전을 위해서 입니다.
SC.02	충전 프로파일이 변경된 경우	충전기는 항상 CSMS에 알려야 합니다.	이에 사용되는 메시지는 이 섹션의 시작 부분에 언급된 메커니즘 중 어떤 것이 적용되는지에 따라 다릅니다. 1. 해당 없음 2. NotifyEVChargingScheduleRequest 3. NotifyChargingLimitRequest 4. TransactionEventRequest
SC.03		SC.02에 메커니즘 3 및 4에 설명된 충전 프로파일의 변경된 한도에 대한 CSMS 보고는 한도 변경이 구성 변수 `LimitChangeSignificance`에 정의된 백분율보다 작은 경우 건너뛸 수 있습니다.	이는 충전기가 작은 변동(예: 충전기의 HEMS/스마트 미터 입력으로 인해)에 대해 많은 메시지를 보내는 것을 방지하기 위한 것입니다.

ID	전제 조건	요구 사항 정의	참고
SC.04		GetCompositeScheduleResponse 메시지는 항상 예상 충전 일정, 즉 충전을 위한 가장 낮은 *limit* 를 보고해야 합니다. 즉, EV에 충전 한도 X가 있고 (예: ISO 15118 프로토콜을 사용하여) 제공된 양보다 적은 에너지 Y를 사용할 것이라고 표시하면 충전기는 한도 Y를 보고해야 합니다.	

5. Use cases & Requirements

5.1. General Smart Charging

K01 - SetChargingProfile

Table 175. K01 - Central Smart Charging

번호	유형	설명
1	**이름**	SetChargingProfile
2	**ID**	K01
	기능 블록	K. 스마트 충전
3	**목표**	CSMS가 특정 EVSE 또는 전체 충전기에서 일정 기간 동안 끌어오는 충전 전력 또는 전류에 영향을 미칠 수 있도록 합니다.
4	**설명**	CSMS는 SetChargingProfileRequest를 충전기로 보내 EV에서 끌어오는 전력 또는 전류에 영향을 미칩니다. CSMS는 ChargingSchedule을 계산하여 특정 한도 내에 머물도록 하는데, 이는 외부 시스템에서 부과할 수 있습니다.
	Actors	충전기, CSMS, EV
	시나리오 설명	**1.** CSMS는 SetChargingProfileRequest를 충전기로 보내 충전 한도를 설정합니다. **2.** 충전기는 SetChargingProfileResponse로 응답합니다.
5	**필수 조건**	해당 없음
6	**사후 조건**	**성공한 사후 조건:** 충전기는 CSMS가 보낸 SetChargingProfileRequest에 따라 특정 EV의 충전 전력 또는 전류에 ***성공적으로*** 영향을 미칩니다. **실패 사후 조건:** 충전기는 CSMS에서 보낸 SetChargingProfileRequest에 따라 특정 EV의 충전 전력 또는 전류에 영향을 미칠 수 없습니다.

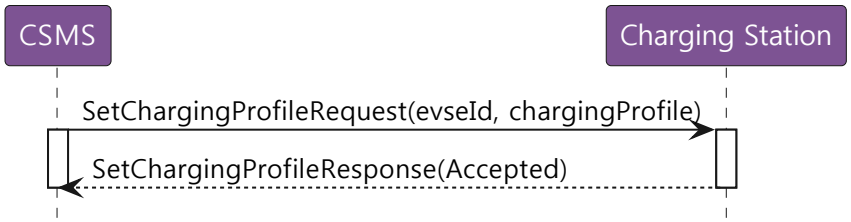

Figure 101. 시퀀스 다이어그램: SetChargingProfile

7	**오류 처리**	n/a
8	**비고**	n/a

K01 - SetChargingProfile - Requirements

Table 176. K01 - 요구 사항

ID	사전 조건	요구 사항 정의	참고
K01.FR.01		CSMS는 충전 한도를 설정할 수 있습니다. TxProfile을 사용하여 트랜잭션에 연결합니다.	
K01.FR.02		CSMS는 EV에 대한 제한 일정으로 사용해야 하는 EVSE에 대한 새로운 충전 프로파일을 보낼 수 있습니다.	
K01.FR.03		CSMS는 TxProfile을 설정할 때 SetChargingProfileRequest에 *transactionId* 를 포함해야 합니다.	transactionId는 프로파일을 특정 트랜잭션에 일치시키는 데 사용됩니다.
K01.FR.04	K01.FR.03 AND 주어진 *transactionId* 가 알려져 있는 경우	충전기는 지정된 *transactionId* 를 사용하여 보낸 TxProfile을 트랜잭션에 적용해야 합니다.	
K01.FR.05	이미 알려진 ChargingProfile.*id* 가 있는 SetChargingProfileRequest를 수신하고 기존 ChargingProfile에 chargingProfilePurpose = `ChargingStationExternalConstraints` 가 없는 경우	충전기는 기존 ChargingProfile을 지정된 것으로 대체해야 합니다.	ChargingStationExternalConstraints 프로파일은 대체할 수 없습니다.
K01.FR.06	*chargingProfilePurpose* 가 `TxProfile` 이 아닌 경우	CSMS는 충전기의 다른 ChargingProfile(다른 *id*)에 이미 존재하고 유효 기간이 겹치는 *stackLevel* - *chargingProfilePurpose* - *evseId* 조합을 갖는 ChargingProfile을 보내서는 안 됩니다.	이는 동일한 스택 수준과 용도를 가진 두 개의 충전 프로파일이 동시에 유효하지 않도록 하기 위한 것입니다.
K01.FR.07	충전기가 SetChargingProfileRequest를 *accepts* 하는 경우	충전기는 충전 프로파일 컬렉션을 다시 평가하여 어떤 ChargingProfile이 활성화될지 결정해야 합니다.	
K01.FR.08		CSMS는 기본 충전 프로파일로 사용할 충전 프로파일을 충전기로 보낼 수 있습니다.	
K01.FR.09	TxProfile이 있는 SetChargingProfileRequest가 수신되고 지정된 EVSE에서 활성화된 트랜잭션이 없는 경우	충전기는 `Rejected` 상태의 SetChargingProfileResponse를 보내야 합니다.	
K01.FR.10	ChargingProfile의 *validFrom* 및 *validTo* 가 설정되지 않은 경우	충전기는 ChargingProfile이 명시적으로 대체될 때까지 무기한 유효한 것으로 간주해야 합니다.	
K01.FR.11	ChargingSchedule에 *duration* 이 있고 ChargingSchedulePeriod.*startPeriod* >= ChargingSchedule.*duration* 인 경우	충전기는 ChargingSchedulePeriod를 실행해서는 안 됩니다. ChargingSchedule의 기간을 지났기 때문입니다.	
K01.FR.12		ChargingSchedulePeriod는 목록의 다음 ChargingSchedulePeriod가 시작되거나 ChargingSchedule.*duration* 이 경과할 때까지 활성 상태를 유지합니다.	
K01.FR.13	recurrencyKind가 recurrencyKind period보다 짧은 ChargingSchedule duration과 함께 사용되는 경우.	충전기는 ChargingSchedule duration이 끝나면 기본 동작으로 돌아갑니다.	

ID	사전 조건	요구 사항 정의	참고
K01.FR.14	TxDefaultProfile 및 evseId = 0인 SetChargingProfileRequest가 수신되고 동일한 stackLevel을 가진 다른 TxDefaultProfile이 특정 EVSE에 설치되지 않은 경우.	충전기는 이 프로파일을 모든 EVSE에 적용해야 하지만 복사해서는 안 됩니다.	EVSE #0의 TxDefaultProfile 충전 프로파일은 EVSE #0이 "소유"하지만 모든 EVSE에 적용됩니다.
K01.FR.15	TxDefaultProfile 및 evseId > 0을 갖는 SetChargingProfileRequest를 수신하고 동일한 stackLevel을 갖는 TxDefaultProfile이 EVSE #0에 설치되지 않은 경우.	충전기는 지정된 EVSE에만 이 프로파일을 적용해야 합니다.	
K01.FR.16		TxProfile은 evseId >0에서만 사용해야 합니다.	
K01.FR.17		같은 chargingProfilePurpose를 가진 두 개 이상의 ChargingProfile가 유효한 경우, *validFrom* 및 *validTo* 필드에 따라 결정되며, 더 높은 *stackLevel* 을 가진 ChargingProfile의 ChargingSchedule이 더 낮은 *stackLevel* 을 가진 ChargingProfile의 ChargingSchedule을 무효화합니다.	
K01.FR.19		CSMS는 numberPhases가 1이 아닌 경우 SetChargingProfileRequest에서 phaseToUse를 설정해서는 안 됩니다.	
K01.FR.20		CSMS는 EVSE에 ACPhaseSwitchingSupported가 정의되어 있지 않고 true로 설정되어 있지 않은 경우 SetChargingProfileRequest에서 phaseToUse를 설정해서는 안 됩니다.	
K01.FR.21		선택 사항인 ChargingSchedule 필드 minChargingRate는 충전기에서 EVSE 간 전력 분배를 최적화하는 데 사용할 수 있습니다.	이 매개변수는 로컬 컨트롤러에 minChargingRate 이하의 충전은 비효율적임을 알려 다른 밸런싱 전략을 선택할 수 있는 가능성을 제공합니다.
K01.FR.22		CSMS는 SetChargingProfileRequest에서 chargingProfilePurpose를 ChargingStationExternalConstraints로 설정해서는 안 됩니다.	이 목적은 외부 시스템에서 충전 한도/일정을 설정한 경우에만 사용됩니다.
K01.FR.26	SetChargingProfileRequest가 구성 변수 `ChargingScheduleChargingRateUnit` 에 구성되지 않은 값이 *chargingRateUnit* 에 대한 값으로 수신되는 경우.	충전기는 SetChargingProfileResponse로 `Rejected` 상태로 응답해야 합니다.	
K01.FR.27		SetChargingProfileRequest를 통해 설정된 ChargingProfiles는 재부팅/전원 주기를 거쳐도 지속되어야 합니다.	
K01.FR.28	존재하지 않는 evseId에 대해 SetChargingProfileRequest가 수신된 경우.	충전기는 SetChargingProfileResponse로 응답해야 하며 상태는 `Rejected` 입니다.	

ID	사전 조건	요구 사항 정의	참고
K01.FR.29	충전기가 스마트 충전을 지원하지 않는 경우.	충전기는 RPC 프레임워크 호출자 CALLERROR: NotSupported 또는 NotImplemented로 응답해야 합니다.	
K01.FR.30	*chargingProfile* 에는 *startSchedule* 이 미래 시간으로 설정된 *chargingSchedule* 이 있습니다.	충전기는 *startSchedule* 에서 설정한 시점부터 이 일정의 제한을 부과하기 시작해야 합니다.	
K01.FR.31		chargingSchedule의 첫 번째 chargingSchedulePeriod의 *startPeriod* 는 항상 0이어야 합니다.	
K01.FR.32	(K01.FR.14 또는 K01.FR.15)이고 지정된 EVSE에서 트랜잭션이 활성화되어 있습니다(evseId = 0은 모든 EVSE를 나타냄).	충전기는 지정된 EVSE에서 트랜잭션을 계속 진행해야 하지만 새로운/업데이트된 TxDefaultProfile을 사용합니다.	
K01.FR.33	K01.FR.03 AND 주어진 transactionId가 알려지지 않음	충전기는 SetChargingProfileRequest를 reject해야 합니다.	
K01.FR.34	CSMS는 현재 트랜잭션에 대한 NotifyEVChargingNeedsRequest를 수신하지 못했습니다. 즉, 충전 세션은 ISO 15118을 사용하지 않습니다.	SetChargingProfileRequest의 ChargingProfile에는 ChargingScheduleType이 하나만 포함되어야 합니다.	ISO 15118 스마트 충전에 대한 사용 사례 K15-K17을 참조하세요.
K01.FR.35		chargingSchedule의 ChargingSchedulePeriod 요소 목록은 ChargingSchedulePeriod *.startPeriod* 의 값 기준 오름차순으로 정렬해야 합니다.	이는 목록이 시간순으로 정렬됨을 의미합니다.
K01.FR.36	ChargingProfile의 *validFrom* 이 설정된 경우	충전기는 현재 시간 >= *validFrom* 일 때 ChargingProfile을 유효한 것으로 간주해야 합니다.	
K01.FR.37	ChargingProfile의 *validTo* 가 설정된 경우	충전기는 현재 시간 < *validTo* 일 때 ChargingProfile을 유효한 것으로 간주해야 합니다.	
K01.FR.38	*chargingProfilePurpose* = `ChargingStationMaxProfile` 인 경우	*chargingProfileKind* 가 `Relative` 이면 안 됩니다	
K01.FR.39	*chargingProfilePurpose* 가 `TxProfile` 인 경우	CSMS는 다른 ChargingProfile(다른 *id*)에 이미 존재하는 *stackLevel* - *transactionId* 조합을 가진 ChargingProfile을 `TxProfile` 목적으로 보내면 안 됩니다.	이는 동일한 스택 수준과 목적을 가진 두 개의 충전 프로파일이 동시에 유효하지 않도록 하기 위한 것입니다.
K01.FR.40	ChargingProfile의 *chargingProfileKind* 가 `Absolute` 또는 `Recurring` 인 경우	ChargingProfile의 ChargingSchedule에 *startSchedule* 값이 있어야 합니다.	이는 일정과 반복 시퀀스의 시작 날짜/시간을 결정합니다.
K01.FR.41	ChargingProfile의 *chargingProfileKind* 가 `Relative` 인 경우	ChargingProfile의 ChargingSchedule에는 *startSchedule* 필드가 없어야 합니다.	상대적 프로파일은 프로파일이 활성화된 때부터 시작됩니다.

ID	사전 조건	요구 사항 정의	참고
K01.FR.42	K01.FR.41	충전기가 에너지를 공급할 준비가 된 순간을 기준으로 ChargingSchedulePeriods를 relative로 만드는 것을 권장합니다. 즉, EV 운전자가 인증되고 EV가 연결된 시점입니다.	이것은 충전기가 에너지를 공급할 준비가 된 트랜잭션의 지점입니다. PowerPathClosed가 TxStartPoint인 경우 이는 트랜잭션의 시작과 일치합니다. 다음 OCPP 버전에서는 이것이 더 엄격한 요구 사항이 됩니다.
K01.FR.43	numberPhases 값이 있는 SetChargingProfileRequest를 수신하고 EVSE가 AC 유형이고 충전기가 수신된 numberPhases 이상을 사용하지 않도록 보장할 수 없는 경우	충전기는 status = `Rejected` 로 응답해야 합니다.	예를 들어 ChargingProfile이 3상을 정의하고 충전기가 3상으로 충전할 수 있는 경우에도 EV 또는 케이블이 3상으로 충전할 수 있다는 보장은 없습니다. 수신된 MeterValues를 기반으로 CSMS는 사용된 위상 수를 확인할 수 있습니다. 위상 한도를 올바르게 계산하려면 요구 사항 K01.FR.50 및 K01.FR.51을 참조하세요.
K01.FR.44	SetChargingProfileRequest에 *numberPhases* 또는 *phaseToUse* 값이 수신되고 EVSE가 DC 유형인 경우	충전기는 Rejected 대신 status = *Accepted* 로 응답하고 제공된 *numberPhases* 및 *phaseToUse* 값을 무시할 수 있습니다.(MAY)	
K01.FR.45	SetChargingProfileRequest에 *numberPhases* 값이 수신되고 EVSE가 AC 유형이고 수신된 *numberPhases* 가 충전기에서 지원되지 않거나 충전기에서 지원하는 *numberPhases* 보다 높은 경우	충전기는 *Rejected* 대신 status = *Accepted* 로 응답하고 더 낮은 *numberPhases* 로 제한을 부과할 수 있습니다.	단계당 제한을 올바르게 계산하려면 요구 사항 K01.FR.50 및 K01.FR.51을 참조하세요.
K01.FR.46	SetChargingProfileRequest가 *numberPhases = 1* 이고 *phaseToUse* 에 대한 값이 수신되고 EVSE가 AC 유형이고 EVSE가 EV에 연결된 위상을 전환할 수 있는 경우(ACPhaseSwitchingSupported가 *true* 로 정의됨) 또는 EVSE가 이미 수신된 *phaseToUse* 를 사용하려고 하는 경우	충전기는 수신된 *phaseToUse* 가 나타내는 위상을 사용하여 EV에 연결해야 합니다.	
K01.FR.47	SetChargingProfileRequest가 *numberPhases = 1* 이고 *phaseToUse* 가 생략된 경우 수신되고 EVSE가 AC 유형인 경우	충전기는 스스로 위상을 선택해야 합니다.	
K01.FR.48	SetChargingProfileRequest에 *phaseToUse* 값이 수신되고 EVSE가 EV에 연결된 위상을 전환할 수 없는 경우, 이는 ACPhaseSwitchingSupported가 구현되지 않았거나 *false* 로 정의되지 않은 것으로 표시되며 EVSE가 수신된 *phaseToUse* 를 사용하지 않을 경우	충전기는 status = *Rejected* 로 응답해야 합니다.	

ID	사전 조건	요구 사항 정의	참고
K01.FR.49	SetChargingProfileRequest에 *numberPhases* 값이 없는 경우 수신되고 EVSE가 AC 유형인 경우	충전기는 기본값으로 numberPhases = 3을 가정해야 합니다.	
K01.FR.50	SetChargingProfileRequest와 chargingRateUnit = *W* 가 수신되고 ChargingSchedule은 AC 충전에 사용됩니다.	충전기는 다음을 통해 위상 전류 제한을 계산해야 합니다. 위상당 전류 = 전력 / (라인 전압 * 위상 수).	계산에 사용된 "라인 전압"은 측정된 전압이 아니라 해당 영역에 설정된 전압(예: 230 또는 110V)입니다. "위상 수"는 ChargingSchedulePeriod의 numberPhases입니다. 일반적으로 AC 충전의 경우 chargingRateUnit = *A* 를 사용하는 것이 더 편리합니다.
K01.FR.51	chargingRateUnit = *A* 인 SetChargingProfileRequest를 수신할 때	충전기는 제공된 제한을 사용하여 모든 단계의 합이 아닌 단계당 암페어 양을 제한해야 합니다.	
K01.FR.52	TxDefaultProfile 및 *evseId* = 0인 SetChargingProfileRequest가 수신되고 동일한 stackLevel을 가진 TxDefaultProfile이 특정 EVSE에 설치되고 해당 *chargingProfile.id* 가 수신된 *chargingProfile.id* 와 같지 않은 경우	충전기는 `Rejected` 상태와 선택적으로 *reasonCode* = `DuplicateProfile` 인 SetChargingProfileResponse로 응답해야 합니다.	
K01.FR.53	TxDefaultProfile 및 *evseId* > 0인 SetChargingProfileRequest를 수신하고 동일한 *stackLevel* 을 가진 TxDefaultProfile이 EVSE #0에 설치되고 해당 *chargingProfile.id* 가 수신된 *chargingProfile.id* 와 같지 않은 경우	충전기는 `Rejected` 상태와 선택적으로 *reasonCode* = `DuplicateProfile` 인 SetChargingProfileResponse로 응답해야 합니다.	

K02 - Central Smart Charging

Table 177. K02 - Central Smart Charging

번호	유형	설명
1	**이름**	Central Smart Charging
2	**ID**	K02
	기능 블록	K. Smart Charging
3	**목표**	CSMS가 특정 EVSE 또는 전체 충전기에서 일정 기간 동안 끌어오는 충전 전력 또는 전류에 영향을 미칠 수 있도록 합니다.
4	**설명**	CSMS가 충전기에 SetChargingProfileRequest를 보내 EV에서 끌어오는 전력 또는 전류에 영향을 미칩니다. CSMS는 외부 시스템에서 부과할 수 있는 한도 내에 머물기 위해 ChargingSchedule을 계산합니다. 참조: Central Smart Charging
	Actors	충전기, CSMS, EV, EV 운전자

번호	유형	설명
	시나리오 설명	**1.** 인증 후 충전기는 제어 파일럿 신호를 통해 EV가 사용할 수 있는 최대 전류를 설정합니다. 이 한도는 충전기가 이전에 CSMS에서 수신한 (기본) ChargingProfiles를 기반으로 합니다. **2.** EV가 충전을 시작하고 TransactionEventRequest가 CSMS로 전송됩니다. **3.** CSMS가 TransactionEventResponse로 응답합니다. **4.** TransactionEventRequest에 대한 응답으로 CSMS는 SetChargingProfileRequest를 사용하여 트랜잭션에 대한 충전 한도를 설정할 수 있습니다. **5.** 충전기는 SetChargingProfileResponse로 응답합니다. **6.** 충전이 진행되는 동안 EVSE는 설치된 ChargingProfiles에 따라 최대 전류 또는 전력을 지속적으로 조정합니다.
	대체 시나리오	K03 - Local Smart Charging K04 - Internal Load Balancing
5	**필수 조건**	기능 블록 *Smart Charging* 이 설치되어 있습니다.
6	**사후 조건**	**성공한 사후 조건:** 충전기는 CSMS에서 보낸 SetChargingProfileRequest에 따라 특정 EV의 충전 전력 또는 전류에 *성공적으로* 영향을 미칩니다. **실패한 사후 조건:** 충전기는 CSMS에서 보낸 SetChargingProfileRequest에 따라 특정 EV의 충전 전력 또는 전류에 영향을 미칠 수 없습니다.

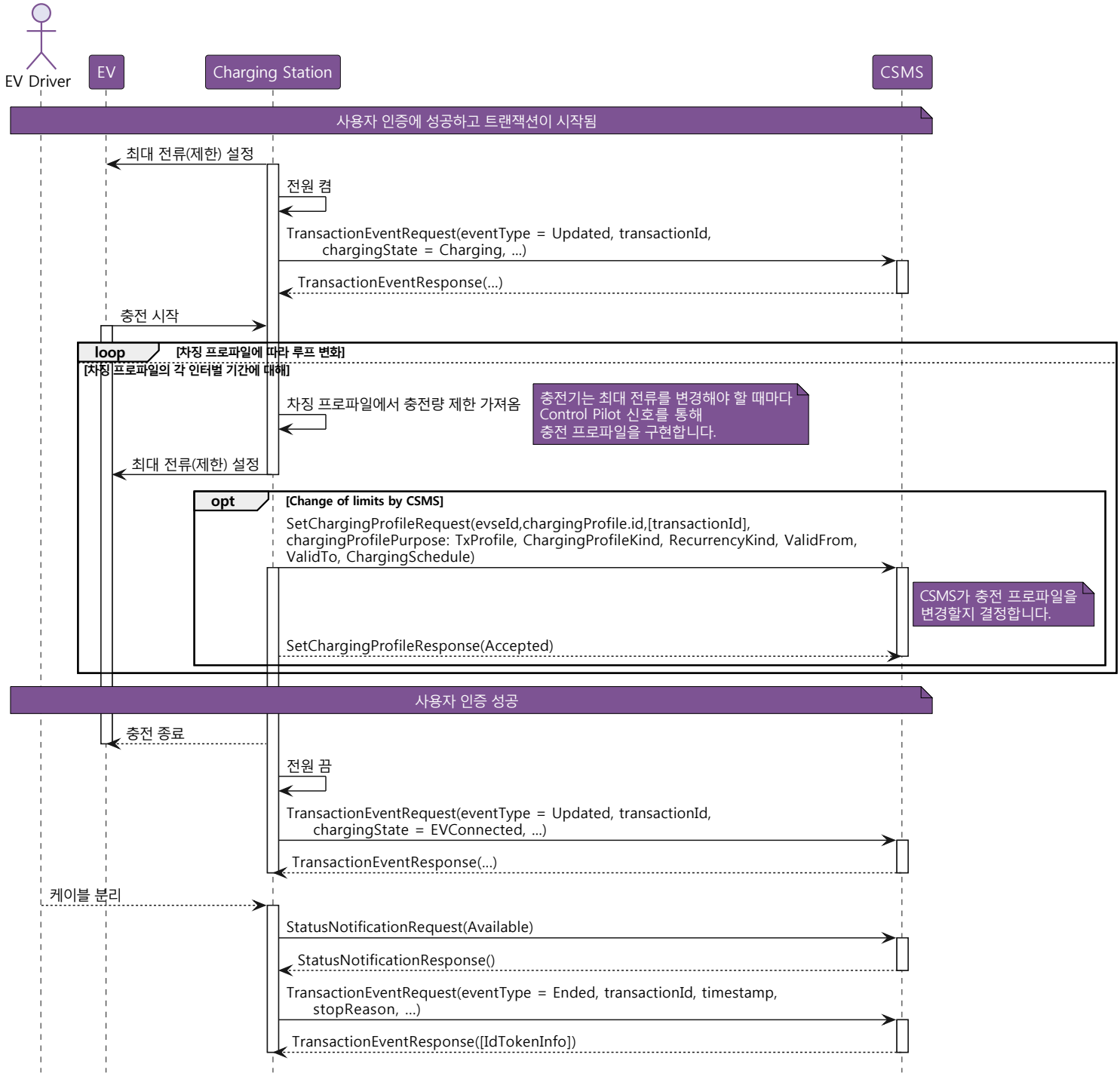

Figure 102. Sequence Diagram: Central Smart Charging

위 그림에 대한 설명:

- 인증 후 EVSE는 Control Pilot 신호를 통해 사용할 최대 전류를 설정합니다. 이 제한은 EVSE가 이전에 CSMS에서 수신한 (기본) 충전 프로파일을 기반으로 합니다. EV가 충전을 시작하고 TransactionEventRequest가 CSMS로 전송됩니다.
- 충전이 진행되는 동안 EVSE는 충전 프로파일에 따라 최대 전류 또는 전력을 지속적으로 조정합니다. 선택적으로 CSMS는 언제든지 EVSE에 대한 새 충전 프로파일을 보낼 수 있습니다. 그러면 충전기는 새 복합 일정을 계산할 때 이 새 일정도 고려합니다. 이런 방식으로 CSMS는 진행 중인 트랜잭션의 충전에 영향을 미칠 수 있습니다.

7	**오류 처리**	n/a

8	참고	CSMS는 트랜잭션당 ChargingSchedule에 대한 제약 조건을 결정합니다. CSMS는 EVSE에 충전 한도를 부과합니다. TransactionEventRequest에 대한 응답으로 CSMS는 TxProfile을 사용하여 트랜잭션에 대한 충전 한도를 설정하도록 선택할 수 있습니다. 충전 프로파일을 보내기 전에 TransactionEventRequest에서 `offline` 플래그를 확인하여 트랜잭션이 계속 진행될 가능성이 있는지 확인하는 것이 좋습니다. TransactionEventRequest는 *Offline* 기간 동안 캐시되었을 수 있습니다. 트랜잭션에 적용되는 최종 일정 제약은 목적이 *ChargingStationMaxProfile* 인 프로파일을 목적이 *TxProfile* 인 프로파일과 병합하거나 목적이 *TxProfile* 인 프로파일이 제공되지 않는 경우 TxDefaultProfile 프로파일을 병합하여 결정됩니다. 0개 이상의 *ChargingStationMaxProfile* 또는 TxDefaultProfile 과 같은 ChargingProfile이 이전에 CSMS에서 수신되었을 수 있습니다. TxProfile에서 ChargingSchedule의 기간 필드를 생략하여 자동으로 트랜잭션이 끝날 때까지 지속되도록 하는 것을 권장합니다. 트랜잭션이 끝나기 전에 TxProfile이 만료되면 활성 TxDefaultProfile과 ChargingStationMaxProfile의 가장 낮은 한도로 돌아갑니다. 다른 활성 프로파일이 없으면 충전기의 로컬 한도로 돌아갑니다. 위의 시나리오 설명과 시퀀스 다이어그램은 시작 트랜잭션을 위한 구성 변수가 다음과 같이 구성되는 것을 기반으로 합니다. `TxStartPoint`: Authorized, DataSigned, PowerPathClosed, EnergyTransfer 이 유즈케이스는 다른 구성에서도 유효하지만, 그 경우 트랜잭션이 다른 시점에서 시작/종료될 수 있으며, 이는 메시지가 전송되는 순서에 영향을 미칠 수 있습니다. 자세한 내용은 사용 사례를 참조하세요. E01 - Start Transaction options.

K02 - Central Smart Charging - Requirements

Table 178. K02 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
K02.FR.01		CSMS는 외부 시스템에서 부과한 제한 내에 머물기 위해 충전 프로파일을 사용해야 합니다.	
K02.FR.02	인증 후.	EVSE는 Control Pilot 신호를 통해 사용할 최대 전류를 설정합니다.	이 요구 사항은 61851을 사용하는 AC 충전기에만 적용됩니다. 제한은 EVSE가 이전에 CSMS에서 수신한 (기본) 충전 프로파일을 기반으로 할 수 있습니다.
K02.FR.03		업데이트된 ChargingProfile이 현재 트랜잭션에만 적용되도록 하기 위해 CSMS는 ChargingProfile의 chargingProfilePurpose를 *TxProfile* 로 설정해야 합니다.	업데이트된 충전 프로파일은 CSMS에서 동일한 chargingProfileId로 ChargingProfile을 보내면 보낼 수 있습니다.
K02.FR.04	*TxProfile* 목적이 있는 트랜잭션별 프로파일이 있는 경우.	TxProfile은 현재 트랜잭션 기간 동안만 목적이 TxDefaultProfile인 기본 청구 프로파일을 무시해야 합니다.	
K02.FR.05	K02.FR.04 트랜잭션이 중단된 후	TxProfile은 삭제되어야 합니다.	
K02.FR.06		선택 사항인 ChargingSchedule 필드 minChargingRate는 충전기에서 EVSE 간 전력 분배를 최적화하는 데 사용할 수 있습니다.	이 매개변수는 로컬 컨트롤러에 minChargingRate 이하의 충전은 비효율적임을 알려 다른 밸런싱 전략을 선택할 수 있는 가능성을 제공합니다.

ID	전제 조건	요구 사항 정의	참고
K02.FR.07		CSMS는 SetChargingProfileRequest에서 chargingProfilePurpose를 ChargingStationExternalConstraints로 설정해서는 안 됩니다.	이 목적은 외부 시스템이 충전 한도/일정을 설정한 경우에만 사용됩니다.
K02.FR.08	K02.FR.04 AND TxProfile의 충전 일정은 설정된 기간 또는 validTo 기간이 만료되었기 때문에 트랜잭션이 끝나기 전에 종료됩니다.	충전기는 활성 TxDefaultProfile 및 ChargingStationMaxProfile의 가장 낮은 한도를 사용하도록 돌아갑니다. 다른 활성 프로파일이 없으면 충전기의 로컬 한계로 돌아갑니다	

K03 - Local Smart Charging

Table 179. K03 - 로컬 스마트 충전

번호	유형	설명
1	**이름**	로컬 스마트 충전
2	**ID**	K03
	기능 블록	K. 스마트 충전
3	**목표**	로컬 컨트롤러가 충전기에서 충전 한계를 설정할 수 있도록 합니다.
4	**설명**	로컬 스마트 충전은 스마트 충전이 활성화된 충전기의 충전 한계가 CSMS가 아닌 로컬 컨트롤러가 로컬로 제어하는 사용 사례를 설명합니다. 충전 한도는 어떤 방식으로든 로컬 컨트롤러에서 미리 구성하거나 CSMS에서 설정할 수 있습니다. 로컬 컨트롤러는 필요에 따라 한도를 조정하여 연결된 EVSE에 이 용량을 분배하는 로직을 포함해야 합니다. 로컬 스마트 충전에 대한 이 사용 사례는 충전기 그룹에서 사용할 수 있는 전력량을 특정 최대치로 제한하는 것입니다. Figure Local Smart Charging Topology 참조
	Actors	충전기, CSMS, EV, 로컬 컨트롤러, EV 드라이버
	시나리오 설명	**1.** 충전기는 인증 후 제어 파일럿 신호를 통해 EV가 사용할 수 있는 최대 전류를 설정합니다. 이 한도는 충전기가 이전에 CSMS에서 수신한 TxDefaultProfile을 기반으로 합니다. **2.** EV가 충전을 시작하고 충전기가 TransactionEventRequest를 보냅니다. **3.** 로컬 컨트롤러를 통해 TransactionEventRequest가 CSMS로 전송되어 로컬 컨트롤러가 트랜잭션이 시작되었음을 알 수 있습니다. **4.** 트랜잭션 중에 로컬 컨트롤러가 SetChargingProfileRequest를 보내 충전 전류/전력에 영향을 미칩니다. **5.** 충전기는 설치된 ChargingProfiles에 따라 충전 한도를 계산합니다. **6.** 로컬 컨트롤러는 충전기와 CSMS 간에 메시지를 전달하기만 하므로 CSMS가 모든 로컬 스마트 충전 그룹 구성원을 개별적으로 처리할 수 있습니다. **7.** 충전이 진행되는 동안 EVSE는 설치된 ChargingProfiles에 따라 최대 전류를 지속적으로 조정합니다.
5	**필수 조건**	기능 블록 *Smart Charging* 이 설치되었습니다.
6	**사후 조건**	**성공적인 사후 조건:** 로컬 컨트롤러가 제어 파일럿 신호를 통해 최대 충전 한도를 ***성공적으로*** 제어합니다. **실패 사후 조건:** 해당 없음

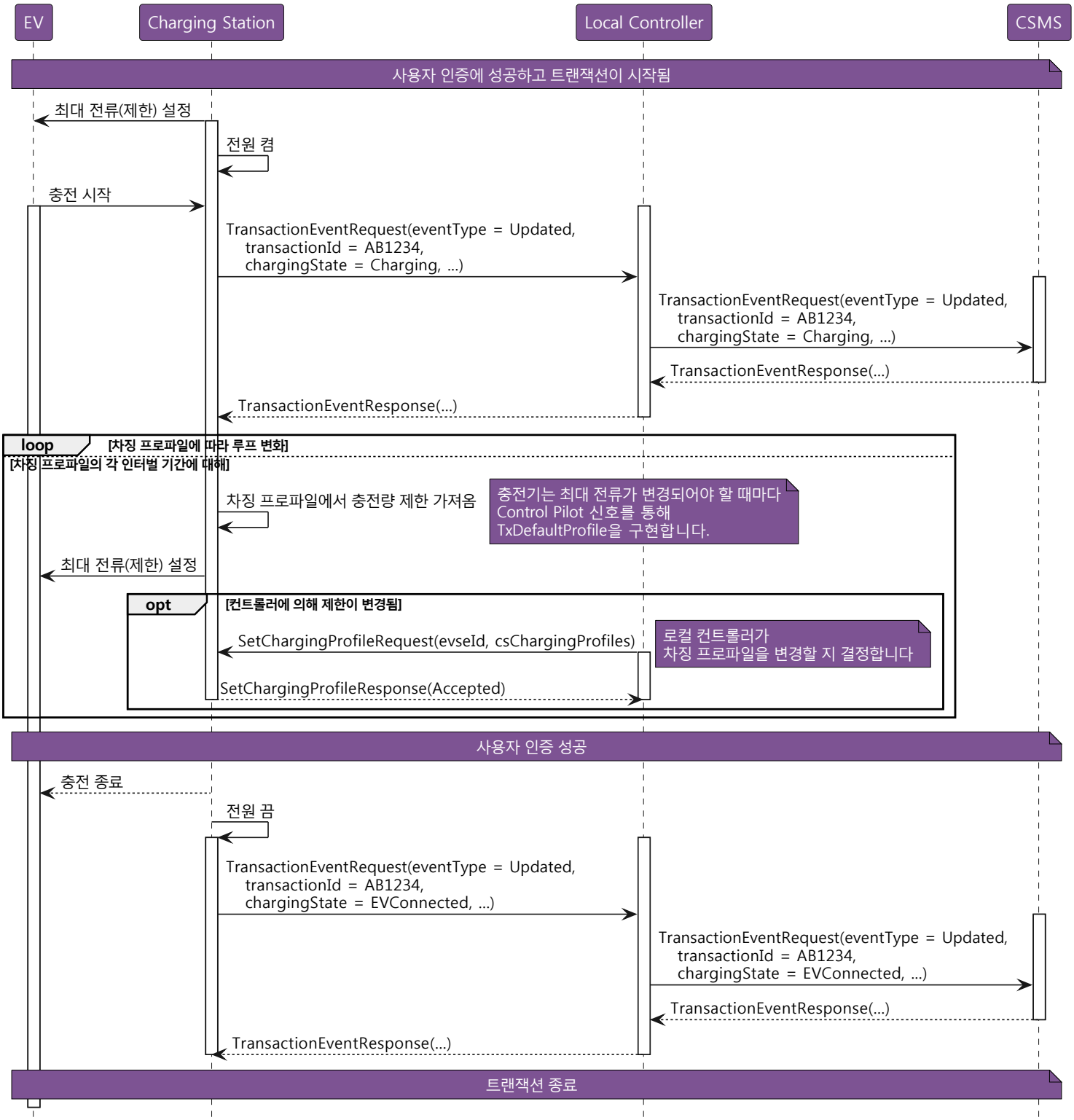

Figure 103. Sequence Diagram: Local Smart Charging

7	**오류 처리**	해당 없음
		로컬 스마트 충전을 위한 로컬 컨트롤러는 다음과 같이 다양한 방식으로 구현할 수 있습니다. 별도의 물리적 구성 요소 또는 여러 다른 충전기를 제어하는 '마스터' 충전기의 일부로 구현할 수 있습니다.
8	**참고**	로컬 컨트롤러는 자체 EVSE를 가질 수도 있고 가질 수 없을 수도 있습니다. 로컬 스마트 충전 그룹의 충전기에 대한 제한은 로컬 컨트롤러에서 어떤 방식으로든 미리 구성하거나 CSMS에서 설정할 수 있습니다. 로컬 컨트롤러에는 필요에 따라 제한을 조정하여 연결된 EVSE 간에 이용량을 분배하는 로직이 포함되어 있습니다.

K03 - Local Smart Charging - Requirements

Table 180. K03 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
K03.FR.01		로컬 컨트롤러는 충전기에 충전 제한을 부과할 수 있습니다.	
K03.FR.02	K03.FR.01	이러한 제한은 충전기 그룹의 전력 소비를 그룹 제한 내로 유지하기 위해 충전 프로세스 중에 동적으로 변경될 수 있습니다.	
K03.FR.03	로컬 컨트롤러가 EVSE에 새로운 ChargingProfile을 보내는 경우	충전기는 EV를 충전하는 데 사용할 새로운 복합 일정을 계산할 때 이 새로운 ChargingProfile을 고려해야 합니다.	
K03.FR.04		다른 트랜잭션보다 충전 우선 순위가 높은 트랜잭션은 다른 트랜잭션을 중단해야 하는 경우에도 가능한 한 오랫동안 이행되어야 합니다.	
K03.FR.05	TransactionEventResponse에 제공된 충전 우선 순위가 IdTokenInfo의 충전 우선 순위와 다른 경우	TransactionEventResponse의 chargingPriority는 이 트랜잭션에만 사용해야 합니다.	따라서 인증 부여 캐시 등에 저장해서는 안 됩니다.
K03.FR.06	chargingPriority를 알 수 없는 경우.	트랜잭션 또는 IdToken은 chargingPriority가 0인 것으로 가정해야 합니다.	
K03.FR.07		선택 사항인 ChargingSchedule 필드 minChargingRate는 충전기에서 EVSE 간 전력 분배를 최적화하는 데 사용할 수 있습니다.	매개변수는 로컬 컨트롤러에 minChargingRate 이하의 충전은 비효율적임을 알려 다른 밸런싱 전략을 선택할 수 있는 가능성을 제공합니다.
K03.FR.08		로컬 컨트롤러는 SetChargingProfileRequest에서 chargingProfilePurpose를 ChargingStationExternalConstraints로 설정해서는 안 됩니다.	이 목적은 외부 시스템이 충전 한도/일정을 설정한 경우에만 사용됩니다.

K04 - Internal Load Balancing

Table 181. K04 - 내부 부하 분산

번호	유형	설명
1	**이름**	내부 부하 분산
2	**ID**	K04
	기능 블록	K. 스마트 충전
3	**목표**	충전기 내부와 EVSE 간의 내부 부하 분산을 활성화합니다.
4	**설명**	부하 분산 사용 사례는 충전기 내부에서 내부 부하 분산을 수행하는 것으로, 충전기가 EVSE당 전류/전력을 제어합니다. 충전기는 고정된 한도(예: 그리드 연결의 최대 전류)로 구성됩니다. K01 - Set Charging Profile 참조
	Actors	충전기, CSMS, EVSE
	시나리오 설명	**1.** CSMS는 ChargingProfile을 전송하여 알려진 물리적 그리드 연결 한계를 설정합니다. **2.** 충전기는 EVSE당 전류/전력을 제어합니다. **3.** EVSE는 EV에 제어 파일럿 신호를 보냅니다.
5	**필수 조건**	기능 블록 *Smart Charging* 이 설치되었습니다.
6	**사후 조건**	**성공적인 사후 조건:** 충전기는 CSMS가 전송하는 내용을 기반으로 다양한 EVSE 간의 전류/전력을 ***성공적으로*** 균형 조정합니다. **실패 사후 조건:** ChargingProfile은 ***허용되지 않습니다***. 충전은 가능하지만 충전기는 ChargingProfile을 ***허용하지 않습니다***.
7	**오류 처리**	해당 없음
8	**비고**	해당 없음

K04 - Internal Load Balancing - Requirements

Table 182. K04 - 요구 사항

ID	사전 조건	요구 사항 정의	참고
K04.FR.01		충전기는 EVSE당 ChargingSchedule을 제어해야 합니다.	
K04.FR.02		충전기는 고정된 한도로 구성되어야 합니다.	예: 그리드 연결의 최대 전류.
K04.FR.03		ChargingStationMaxProfile의 목적을 가진 ChargingProfile은 ID가 0인 충전기 EVSE에서만 설정할 수 있습니다.	

ID	사전 조건	요구 사항 정의	참고
K04.FR.04		선택 사항인 ChargingSchedule 필드 minChargingRate는 충전기에서 EVSE 간의 전력 분배를 최적화하는 데 사용할 수 있습니다.	매개변수는 로컬 컨트롤러에 minChargingRate 이하의 충전은 비효율적임을 알려 다른 밸런싱 전략을 선택할 수 있는 가능성을 제공합니다.
K04.FR.05		모든 EVSE(및 충전기 하드웨어 자체)의 결합된 에너지 흐름은 *ChargingStationMaxProfile* 에서 설정한 제한보다 커서는 안 됩니다.	

K05 - Remote Start Transaction with Charging Profile

Table 183. K05 - 충전 프로파일이 있는 원격 시작 트랜잭션

번호	유형	설명
1	**이름**	충전 프로파일을 사용한 원격 시작 트랜잭션
2	**ID**	K05
	기능 블록	K. 스마트 충전
3	**목표**	CSMS가 ChargingProfile을 직접 포함하여 트랜잭션을 원격으로 시작할 수 있도록 하여 트랜잭션이 올바른 ChargingProfile을 사용하도록 합니다.
4	**설명**	이 사용 사례는 CSMS가 TxProfile 목적으로 트랜잭션을 원격으로 시작하는 방법을 다룹니다. 이렇게 하면 올바른 TxProfile이 사용됩니다. 또한 충전기가 RequestStartTransactionRequest를 수신한 후 *offline* 이 되는 경우도 있습니다. 3상에서 1상로의 충전 전환이 항상 가능한 것은 아니며 트랜잭션은 올바른 단계에서 시작해야 하므로 이것도 필요합니다.
	Actors	충전기, CSMS, 외부 트리거
	시나리오 설명	**1.** CSMS는 충전기에 RequestStartTransactionRequest를 ChargingProfile 목적과 함께 TxProfile로 전송하여 원격으로 트랜잭션을 시작하도록 요청합니다. **2.** 충전기는 RequestStartTransactionResponse로 응답하여 트랜잭션을 시작할 수 있으며 ChargingProfile을 사용할 것임을 나타냅니다. **3.** 충전기는 TransactionEventRequest (eventType = Started) 메시지를 전송하여 CSMS에 트랜잭션이 시작되었음을 알립니다. **4.** 트랜잭션은 E. Transaction에서 설명한 것과 같은 방식으로 시작됩니다. **5.** 충전기는 TransactionEventRequest(eventType = Updated)를 보내 CSMS에 충전 중임을 알립니다. **6.** 충전기는 설정된 ChargingProfiles에 따라 일반 스마트 충전 세션을 계속합니다.
5	**필수 조건**	기능 블록 *Smart Charging* 이 설치되었습니다.
6	**사후 조건**	**성공한 사후 조건:** 충전기가 제공된 ChargingProfile을 고려하여 성공적으로 충전합니다. **실패한 사후 조건:** 트랜잭션이 *시작되지* 않습니다. 충전기가 제공된 ChargingProfile을 고려하여 성공적으로 충전하지 않습니다.

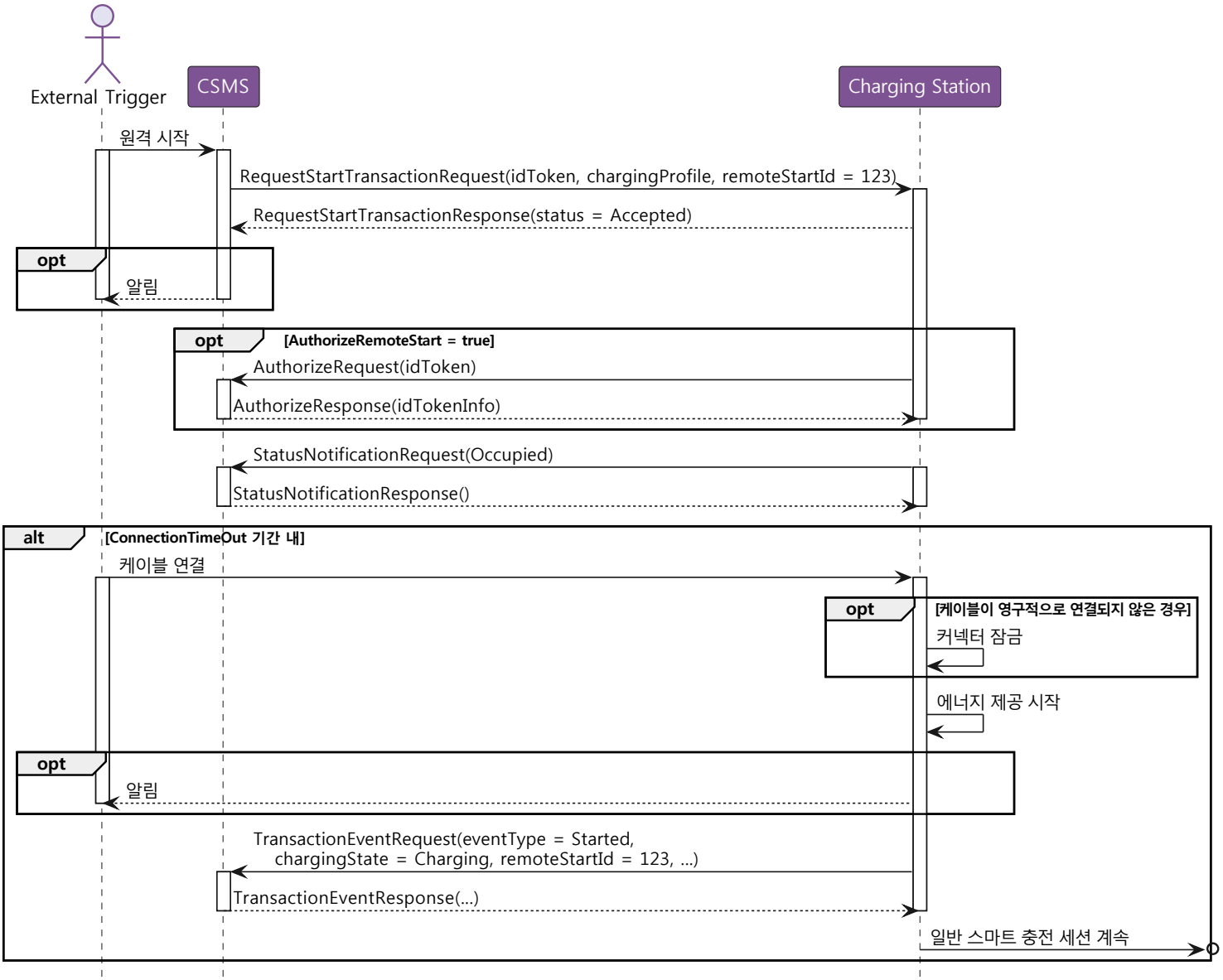

Figure 104. 시퀀스 다이어그램: 충전 프로파일이 있는 원격 시작 트랜잭션

7	**오류 처리**	n/a
8	**참고**	위의 시나리오 설명과 시퀀스 다이어그램은 시작 트랜잭션에 대한 구성 변수가 다음과 같이 구성된 것을 기반으로 합니다. `TxStartPoint`: EVConnected, Authorized, DataSigned, PowerPathClosed, EnergyTransfer 이 유즈케이스는 다른 구성에서도 유효하지만, 그 경우 트랜잭션이 다른 시점에서 시작/종료될 수 있으며, 이는 메시지가 전송되는 순서에 영향을 미칠 수 있습니다. 자세한 내용은 사용 사례 E01 - Start Transaction options를 참조하세요. 목적이 있는 ChargingProfile이 RequestStartTransactionRequest의 일부로 제공되는 경우, 당시에는 transactionId를 알 수 없기 때문에 ChargingProfile에 제공할 수 없습니다.

K05 - Remote Start Transaction with Charging Profile - Requirements

Table 184. K05 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
K05.FR.01		CSMS는 RequestStartTransactionRequest에 ChargingProfile을 포함할 수 있습니다.	

ID	전제 조건	요구 사항 정의	참고
K05.FR.02	K05.FR.01	ChargingProfile의 목적은 항상 TxProfile이어야 합니다.	
K05.FR.03	K05.FR.01 AND NOT K05.FR.04	충전기는 제공된 프로파일을 사용하여 복합 일정을 계산해야 합니다.	
K05.FR.04	스마트 충전을 지원하지 않는 충전기가 ChargingProfile이 포함된 RequestStartTransactionRequest를 수신하는 경우.	충전기는 지정된 ChargingProfile을 무시해야 합니다.	장치 모델 변수 SmartChargingCtrlr.Enabled는 CSMS에 스마트 충전이 지원되는지 여부를 알려줍니다.
K05.FR.05	스마트 충전을 지원하는 충전기가 잘못된 ChargingProfile이 포함된 RequestStartTransactionRequest를 수신하는 경우.	충전기는 RequestStartTransactionResponse로 *status* = `Rejected` 및 선택적으로 *reasonCode* = "InvalidProfile" 또는 "InvalidSchedule"로 응답해야 합니다.	기기 모델 변수 SmartChargingCtrlr.Enabled는 CSMS에 스마트 충전이 지원되는지 여부를 알려줍니다.

K06 - Offline Behavior Smart Charging During Transaction

Table 185. K06 - Offline Behavior Smart Charging During Transaction

번호	유형	설명
1	**이름**	트랜잭션 중 오프라인 스마트 충전 동작
2	**ID**	K06
	기능 블록	K. 스마트 충전
3	**목표**	충전기가 *offline* 인 동안 트랜잭션 기간 동안 현재 ChargingProfile을 계속 사용할 수 있도록 합니다.
4	**설명**	충전기가 목적이 TxProfile인 트랜잭션별 ChargingProfile을 수신한 후 *offline* 이 되면 트랜잭션 기간 동안 이 프로파일을 계속 사용합니다.
	Actors	충전기, CSMS, EV
	시나리오 설명	**1.** CSMS가 TxProfile과 함께 SetChargingProfileRequest를 충전기로 보냅니다. **2.** 충전기가 SetChargingProfileResponse로 응답합니다. **3.** 충전이 진행되는 동안 EVSE는 설치된 ChargingProfiles에 따라 최대 전류 또는 전력을 지속적으로 조정합니다. **4.** 충전기가 *offline* 상태이고 단독으로 작동합니다. **5.** 충전이 진행되는 동안 EVSE는 이미 설치된 ChargingProfiles에 따라 최대 전류 또는 전력을 지속적으로 조정합니다.
5	**필수 조건**	트랜잭션이 진행 중입니다. 기능 블록 *Smart Charging* 이 설치되었습니다.
6	**사후 조건**	**성공적인 사후 조건:** 충전 스테이션은 사용 가능한 충전 프로파일을 계속 사용합니다. **실패 사후 조건:** 해당 없음

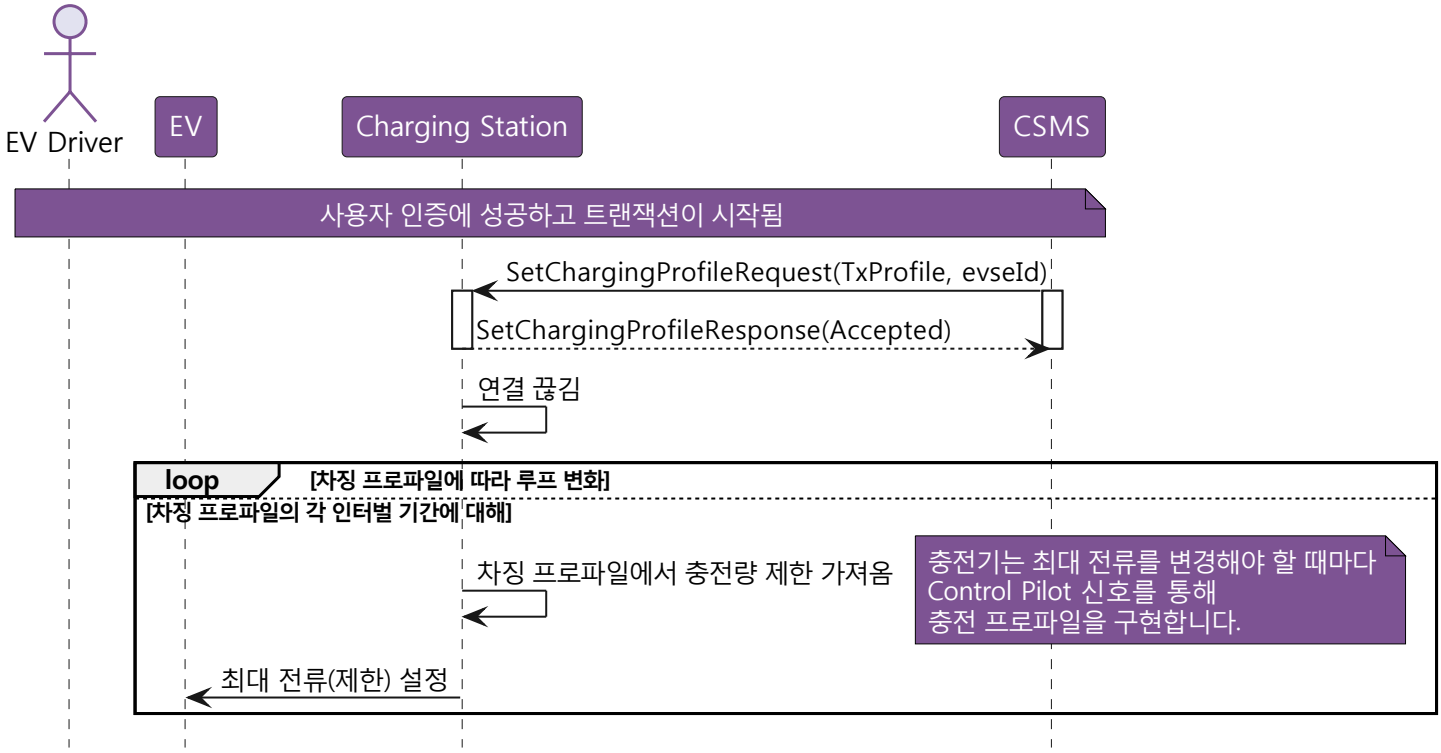

Figure 105. 시퀀스 다이어그램: 오프라인 동작 스마트 충전

7	**오류 처리**	해당 없음
8	**비고**	해당 없음

K06 - Offline Behavior Smart Charging During Transaction - Requirements

Table 186. K06 - 요구 사항

ID	사전 조건	요구 사항 정의
K06.FR.01	충전기가 목적이 TxProfile인 트랜잭션별 ChargingProfile을 수신한 후 *offline* 으로 전환됩니다.	충전기는 트랜잭션 기간 동안 이 프로파일을 계속 사용해야 합니다.
K06.FR.02	충전기가 충전 프로파일 없이 *offline* 으로 전환되는 경우.	충전기는 제약 조건이 적용되지 않는 것처럼 트랜잭션을 실행해야 합니다.

K07 - Offline Behavior Smart Charging at Start of Transaction

Table 187. K07 - 트랜잭션 시작 시 오프라인 동작 스마트 충전

번호	유형	설명
1	**이름**	트랜잭션 시작 시 오프라인 스마트 충전 동작
2	**ID**	K07
	기능 블록	K. 스마트 충전
3	**목표**	충전기가 *offline* 으로 시작된 트랜잭션에 대해 ChargingProfile을 계속 사용할 수 있도록 합니다.

번호	유형	설명
4	**설명**	충전기에 TxDefaultProfile을 설정하면 CSMS는 CSMS와의 통신이 *offline* 인 동안 시작된 모든 트랜잭션이 이 프로파일을 사용하도록 할 수 있습니다.
	Actors	충전기, CSMS, EV, EV 운전자
	시나리오 설명	**1.** CSMS가 TxDefaultProfile과 함께 SetChargingProfileRequest를 충전기로 보냅니다. **2.** 충전기가 SetChargingProfileResponse로 응답합니다. **3.** 충전기가 *offline* 이 되고 단독으로 작동합니다. **4.** 충전기는 다음 중 하나를 통해 제공된 모든 IdToken의 자동 인증을 허용합니다. *a.* 로컬 인증 목록. CSMS와 동기화할 수 있는 식별자 목록입니다. *b.* 인증 캐시 항목. CSMS에서 성공적으로 인증한 이전에 제공된 식별자의 기록을 자율적으로 유지합니다. (성공적인 의미: IdToken이 포함된 메시지에 대한 응답 수신). *c.* 구성 변수: OfflineTxForUnknownIdEnabled = TRUE **5.** 트랜잭션은 E. Transactions에서 설명한 것과 같은 방식으로 시작됩니다. **6.** 충전이 진행되는 동안 EVSE는 이미 설치된 ChargingProfiles에 따라 최대 전류 또는 전력을 지속적으로 조정합니다.
5	**필수 조건**	충전기가 *offline* 입니다. 기능 블록 *Smart Charging* 이 설치되었습니다. 로컬 인증 부여 목록에서 IdToken이 알려져 있거나, 인증 부여 캐시에서 IdToken이 알려져 있거나, 알 수 없는 오프라인 인증이 활성화되었습니다.
6	**사후 조건**	**성공한 사후 조건:** 충전기는 *offline* 시작된 트랜잭션에 사용할 수 있는 설치된 TxDefaultProfile을 사용합니다. **실패 사후 조건:** 해당 없음

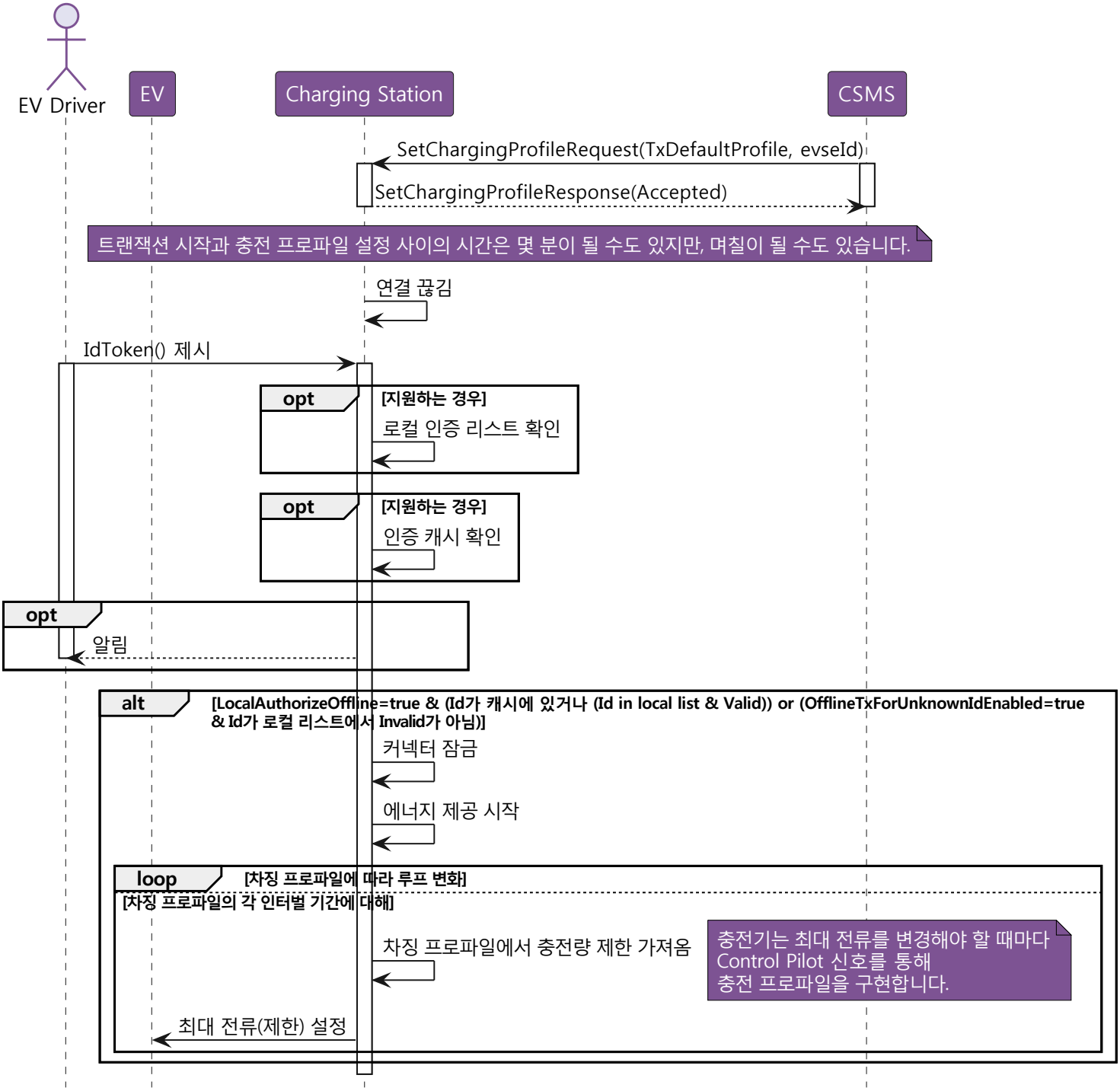

Figure 106. 시퀀스 다이어그램: 오프라인 동작 스마트 충전

7	**오류 처리**	해당 없음
8	**참고**	다른 충전 프로파일 용도를 결합하는 방법에 대한 설명은 섹션 Combining Charging Profile Purposes를 참조하세요.

K07 - Offline Behavior Smart Charging at Start of Transaction - Requirement

Table 188. K07 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
K07.FR.01	트랜잭션이 시작되기 전이나 목적이 TxProfile인 트랜잭션의 ChargingProfile을 수신하기 전에 충전기가 *offline* 이 되는 경우.	충전기는 사용 가능한 충전 프로파일을 사용해야 합니다.	TxDefaultProfile을 현재 트랜잭션 기간 동안만 적용되어야 합니다.

K08 - Get Composite Schedule

Table 189. K08 - 복합 일정 가져오기

번호	유형	설명
1	**이름**	복합 일정 가져오기
2	**ID**	K08
	기능 블록	K. 스마트 충전
3	**목표**	충전기에 복합 충전 일정을 보고하도록 요청합니다.
4	**설명**	이 사용 사례는 CSMS가 충전기에 GetCompositeScheduleRequest를 보내 충전기에서 계산한 복합 충전 일정을 보고하도록 요청하는 방법을 설명합니다. CompositeSchedule은 충전기에 있는 모든 활성 일정과 가능한 로컬 제한을 계산한 결과입니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS는 GetCompositeScheduleRequest를 보내 충전기에 복합 충전 일정을 보고하도록 요청합니다. **2.** 충전기는 일정을 계산합니다. **3.** 충전기는 GetCompositeScheduleResponse로 상태와 ChargingSchedule을 응답합니다.
5	**필수 조건**	기능 블록 *Smart Charging* 이 설치되었습니다.
6	**사후 조건**	**성공한 사후 조건:** CSMS가 충전기에서 복합 일정을 *성공적으로* 수신했습니다. **실패한 사후 조건:** CSMS가 충전기에서 복합 일정을 *수신하지 못했습니다*.

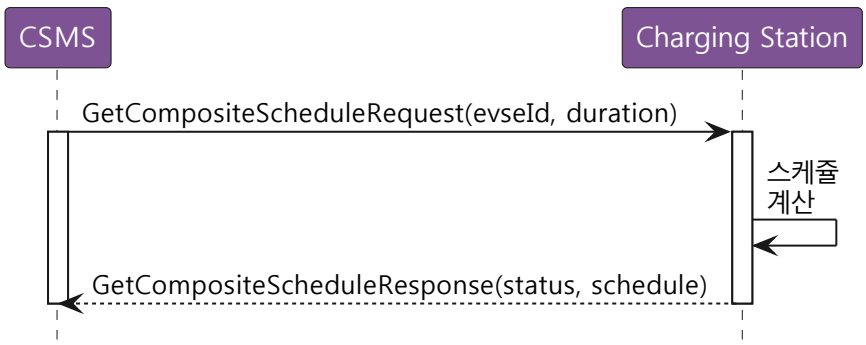

Figure 107. Sequence Diagram: Get Composite Schedule

7	**오류 처리**	해당 없음

8	참고	충전기에서 보낸 충전 일정은 해당 시점의 것만 표시된다는 점에 유의하세요. 이 일정은 외부 원인(예: 그리드 연결 용량을 기반으로 하는 로컬 밸런싱이 활성화되고 하나의 EVSE가 사용할 수 있게 됨)으로 인해 시간이 지남에 따라 변경될 수 있습니다. 충전 수준을 안내하는 복합 일정은 다양한 chargingProfilePurposes의 기존 충전 프로파일을 조합한 것입니다. 이 복합 일정은 각 시간 간격의 최소값을 취하여 계산합니다(참조: Smart Charging signals to a Charging Station from multiple actors). 시간 간격은 고정된 길이일 필요가 없으며 모든 충전 프로파일 목적에 대해 동일할 필요도 없습니다. 즉, 결과 복합 일정에는 길이가 다른 간격이 포함될 수 있습니다. GetCompositeScheduleResponse에서 보고된 일정은 충전기에 있는 모든 활성 일정과 가능한 지역 제한을 계산한 결과입니다. 복합 일정은 요청된 EVSE에 대해 요청된 기간 동안 충전기가 그리드에서 소비할 것으로 예상하는 예상 전력 또는 전류를 보고합니다. evseid=0으로 요청하면 충전기는 그리드 연결에 대한 총 예상 소비량을 계산합니다.

K08 - Get Composite Schedule - Requirements

Table 190. K08 - 요구 사항

ID	전제 조건	요구 사항 정의
K08.FR.01		CSMS는 GetCompositeScheduleRequest를 보내 충전기가 CompositeSchedule을 보고하도록 요청할 수 있습니다.
K08.FR.02	GetCompositeScheduleRequest를 수신하면.	충전기는 메시지 수신 순간부터 Duration(초)까지 예약된 시간 간격을 계산하여 CSMS에 보내야 합니다.
K08.FR.03	GetCompositeScheduleRequest의 evseId가 '0'으로 설정된 경우	충전기는 요청된 기간 동안 그리드에서 소비할 것으로 예상되는 총 예상 전력 또는 전류를 보고해야 합니다.
K08.FR.04		어느 시점에서든 충전 프로파일 ChargingStationMaxProfile, ChargingStationExternalConstraints 및 TxDefaultProfile(또는 TxProfile)의 일정을 병합한 결과인 CompositeSchedule은 사용 가능한 전력 또는 전류가 여러 일정들의 가장 낮은 값보다 작거나 같아야 합니다.
K08.FR.05	충전기가 요청된 일정을 보고할 수 없는 경우(예: *evseId* 가 알려지지 않은 경우)	충전기는 `Rejected` 상태로 응답해야 합니다.
K08.FR.06	K08.FR.02 AND EVSE에서 활성화된 트랜잭션이 없는 경우	충전기는 TxDefaultProfile을 사용하는 EVSE에서 진행 중인 트랜잭션이 있는 것처럼 CompositeSchedule을 계산해야 합니다(이 프로파일 목적이 설정된 경우)
K08.FR.07	구성 변수 `ChargingScheduleChargingRateUnit` 에 구성되지 않은 chargingRateUnit이 포함된 GetCompositeScheduleRequest를 수신하는 경우	충전기는 `Rejected` 상태의 GetCompositeScheduleResponse로 응답해야 합니다.

K09 - Get Charging Profiles

Table 191. K09 - Get Charging Profiles

번호	유형	설명
1	이름	충전 프로파일 가져오기

번호	유형	설명
2	**ID**	K09
	기능 블록	K. 스마트 충전
3	**목표**	CSMS가 충전기에 설치된 충전 일정/제한을 볼 수 있도록 하려면 CSMS 또는 다른 소스에서 이를 설치할 수 있습니다.
4	**설명**	CSMS는 GetChargingProfilesRequest 메시지를 사용하여 충전기에 다른 가능한 소스에서 설치된 모든 충전 프로파일 또는 모든 설치된 충전 프로파일의 하위 집합을 보고하도록 요청할 수 있습니다. 이는 일부 자동 스마트 충전 제어 시스템에 사용하거나 CSO에서 디버그 목적으로 사용할 수 있습니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS는 GetChargingProfilesRequest 메시지를 보내 설치된 충전 프로파일을 충전기에 요청합니다. **2.** 충전기는 GetChargingProfilesResponse 메시지를 보내 충전 일정을 보고할 수 있는지 여부를 나타내며 응답합니다. **3.** 충전기는 여러 개의 ReportChargingProfilesRequest 메시지를 CSMS에 보냅니다. **4.** CSMS는 모든 ReportChargingProfilesRequest에 대해 충전기에 ReportChargingProfilesResponse를 보내 보고서 수신을 확인합니다.
5	**필수 조건**	해당 없음
6	**사후 조건**	CSMS는 요청된 매개변수와 일치하는 충전기에 설치된 충전 프로파일을 알고 있습니다.

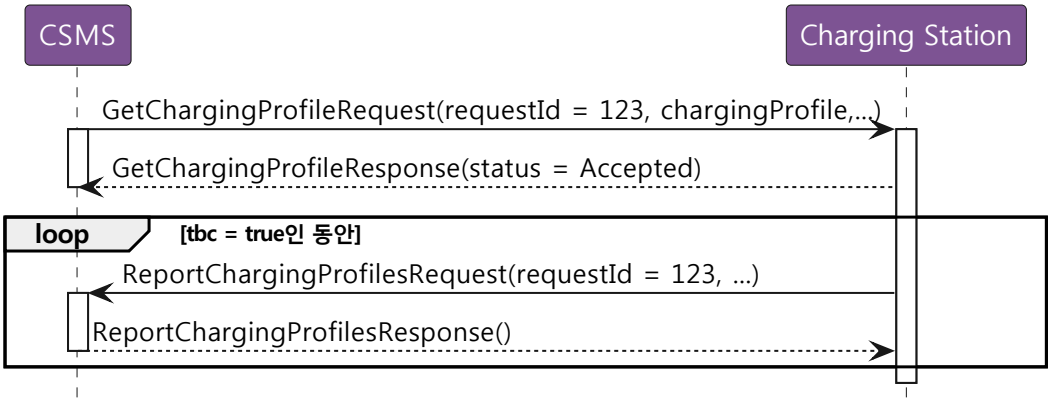

Figure 108. 사용 사례 "충전 프로파일 가져오기"의 시퀀스 다이어그램

7	**오류 처리**	충전기에 GetChargingProfilesRequest의 매개변수와 일치하는 충전 프로파일이 없는 경우 충전기는 NoProfiles로 응답해야 합니다.
8	**비고**	충전 프로파일 보고서는 여러 ReportChargingProfilesRequest 메시지로 분할될 수 있습니다. 이는 다양한 충전 소스에 대한 충전 프로파일을 보고해야 하거나 하나의 메시지에 대한 데이터가 너무 많기 때문일 수 있습니다. 더 많은 보고서가 뒤따를 것임을 나타내려면 **tbc** 플래그를 사용할 수 있습니다.

K09 - Get Charging Profiles - Requirements

Table 192. K09 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
K09.FR.01	GetChargingProfilesRequest에서 requestId가 설정된 경우	충전기는 이 GetChargingProfilesRequest의 결과로 전송된 모든 ReportChargingProfilesRequest에서 requestId를 설정해야 합니다.	
K09.FR.02	충전 프로파일이 두 개 이상의 ReportChargingProfilesRequest에서 보고되는 경우	충전기는 마지막 메시지를 제외한 모든 ReportChargingProfilesRequest 메시지에 대해 *tbc* 플래그를 **true** 로 설정해야 합니다.	
K09.FR.03		CSMS는 GetChargingProfilesRequest에서 *chargingProfile* 기준을 지정해야 합니다. - *chargingProfileId(s)* 의 (목록) 또는 - *stackLevel*, *chargingLimitSource*, *chargingProfilePurpose* 필드 중 하나 이상.	이러한 필드는 동등한 중요성을 가진 필터 값이지만 *chargingProfileId* 는 충전 프로파일을 고유하게 식별하기 때문에 *chargingProfileIds* 를 사용하는 경우 다른 필드는 필요하지 않습니다.
K09.FR.04	GetChargingProfilesRequest에서 *evseId* 가 0보다 큰 값으로 설정된 경우	충전기는 *chargingProfile* 의 모든 필드와 일치하는 지정된 EVSE에 대한 설치된 충전 프로파일을 보고해야 합니다.	
K09.FR.05	GetChargingProfilesRequest에서 *evseId* 가 0으로 설정된 경우	충전기는 *chargingProfile* 의 모든 필드와 일치하는 충전기 자체(그리드 연결)에 설치된 충전 프로파일만 보고해야 합니다.	EVSE #0에는 `ChargingStationMaxProfile`, `ChargingStationExternalConstraints` 또는 `TxDefaultProfile` 이 있을 수 있습니다. `TxDefaultProfile` 은 EVSE #0에 적용되지 않고 모든 개별 EVSE에 적용됩니다(K01.FR.14 참조).
K09.FR.06	GetChargingProfilesRequest에 *evseId* 가 설정되지 않은 경우	충전기는 *chargingProfile* 의 모든 필드와 일치하는 모든 설치된 충전 프로파일을 보고해야 합니다.	

K10 - Clear Charging Profile

Table 193. K10 - Clear Charging Profile

번호	유형	설명
1	**이름**	충전 프로파일 지우기
2	**ID**	K10
	기능 블록	K. 스마트 충전
3	**목표**	충전 프로파일 중 일부 또는 전부를 지웁니다.

번호	유형	설명
4	**설명**	CSMS가 이전에 충전기로 전송된 충전 프로파일 중 일부 또는 전부를 지우고자 하는 경우 CSMS는 ClearChargingProfileRequest를 충전기로 전송합니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS가 ClearChargingProfileRequest를 충전기로 전송합니다. **2.** 충전기는 ClearChargingProfileResponse로 응답하여 해당 상태에서 요청을 처리할 수 있었는지 여부를 지정합니다.
5	**필수 조건**	하나 이상의 ChargingProfile이 설치되었습니다.
6	**사후 조건**	**성공적인 사후 조건:** 요청된 충전 프로파일이 *성공적으로* 지워졌습니다. **실패 사후 조건:** ChargingProfile이 없으므로 요청된 충전 프로파일이 지워지지 *않습니다*.

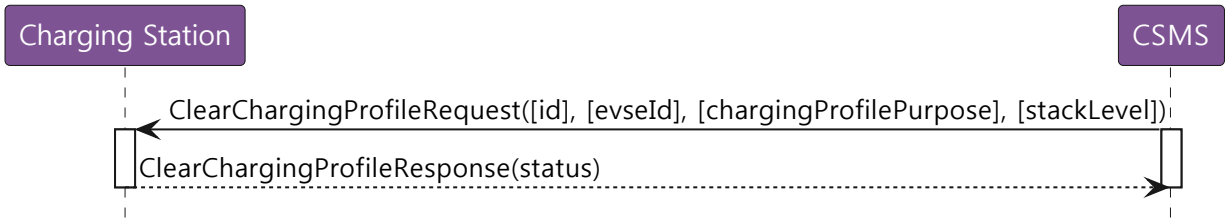

Figure 109. 사용 사례 "충전 프로파일 지우기"의 시퀀스 다이어그램

7	**오류 처리**	n/a
8	**비고**	n/a

K10 - Clear Charging Profile - Requirements

Table 194. K10 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
K10.FR.01	충전기에 일치하는 ChargingProfile이 없는 경우.	ClearChargingProfileRequest를 수신하면 충전기는 *Unknown* 상태로 응답해야 합니다.	
K10.FR.02		CSMS는 chargingProfile.id를 지정하거나 ClearChargingProfileRequest에 하나 이상의 필드 stackLevel, evseId 및 chargingProfilePurpose를 포함하여 지워야 할 충전 프로파일을 지정해야 합니다.	
K10.FR.03	지정된 *chargingProfileId* 가 있는 ClearChargingProfileRequest를 수신하고 참조된 ChargingProfile의 chargingProfilePurpose가 `ChargingStationExternalConstraints` 가 아닌 경우	충전기는 일치하는 ID로 충전 프로파일을 지우고 *status* = `Accepted` 인 ClearChargingProfileResponse 메시지로 응답해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
K10.FR.04	K10.FR.03 AND가 아님 K10.FR.08 AND가 아님 ClearChargingProfileRequest를 수신하면 *evseId*, *chargingProfilePurpose*, *stackLevel* 에 대한 선택적 값이 있습니다.	충전기는 요청의 값과 일치하는(논리적 AND로) ChargingProfile(s)를 지워야 합니다. 단, ChargingProfile = `ChargingStationExternalConstraints` 인 경우는 예외입니다. 그리고 *status* = `Accepted` 인 ClearChargingProfileResponse 메시지로 응답해야 합니다.	
K10.FR.05	하나 이상의 충전 프로파일을 지운 후.	충전기는 복합 일정을 다시 계산하고 그 결과 최대 전력/전류 값을 모든 진행 중인 트랜잭션에 설정해야 합니다.	
K10.FR.06		CSMS는 ClearChargingProfileRequest에서 chargingProfilePurpose를 ChargingStationExternalConstraints로 설정해서는 안 됩니다.	
K10.FR.07	K10.FR.05 그리고 삭제된 프로파일의 chargingProfilePurpose = TxDefaultProfile	충전기는 TxDefaultProfile로 시작된 모든 활성 트랜잭션을 TxDefaultProfile 없이 시작된 것처럼 계속해야 합니다.	
K10.FR.08	ClearChargingProfileRequest를 수신하면 *evseId*, *chargingProfilePurpose*, *stackLevel* AND 에 대한 선택적 값이 있는 경우 일치하는 ChargingProfile(s)는 모두 ChargingProfile = `ChargingStationExternalConstraints` 일 때	충전기는 *status* = `Unknown` 인 ClearChargingProfileResponse 메시지로 응답해야 합니다.	외부 제약에 대한 충전 프로파일은 ClearChargingProfile 메시지에서 무시됩니다.
K10.FR.09	지정된 *chargingProfileId* 와 참조된 ChargingProfile = `ChargingStationExternalConstraints` 의 chargingProfilePurpose를 수신하면	충전기는 *status* = `Unknown` 인 ClearChargingProfileResponse 메시지로 응답해야 합니다.	외부 제약에 대한 충전 프로파일은 ClearChargingProfile 메시지에서 무시됩니다.

5.2. External Charging Limit based Smart Charging

K11 - Set / Update External Charging Limit With Ongoing Transaction

Table 195. K11 - 진행 중인 트랜잭션로 외부 충전 한도 설정/업데이트

번호	유형	설명
1	**이름**	진행 중인 트랜잭션로 외부 충전 한도 설정/업데이트
2	**ID**	K11
	기능 블록	K. 스마트 충전
3	**목표**	진행 중인 트랜잭션이 있는 충전기에서 외부 제어 시스템이 부과하는 충전 일정 또는 충전 한도를 CSMS에 알리는 것.
4	**설명**	외부 제어 시스템이 충전기에 충전 한도/일정을 보냅니다. 이 한도는 CSMS로 전송됩니다.
	Actors	외부 제어 시스템, 충전기, CSMS
	시나리오 설명	**1.** 외부 제어 시스템이 충전기에 충전 한도/일정을 보냅니다. **2.** 선택 사항: 충전기가 새로운 충전 일정을 계산합니다. **3.** 충전기가 진행 중인 트랜잭션의 충전 속도를 조정합니다. **4.** 충전 한도가 `LimitChangeSignificance` 이상 변경된 경우 충전기는 NotifyChargingLimitRequest 메시지를 CSMS로 전송하며, 선택적으로 설정된 충전 한도/일정을 함께 보냅니다. **5.** CSMS는 충전기에 NotifyChargingLimitResponse로 응답합니다. **6.** 충전 속도가 `LimitChangeSignificance` 이상 변경되면 충전기는 TransactionEventRequest 메시지를 보내 CSMS에 알립니다. **7.** CSMS는 충전기에 TransactionEventResponse로 응답합니다.
5	**필수 조건**	충전기가 오류 상태가 아닙니다. 외부 시스템은 OCPP가 아닌 다른 연결을 통해 충전기에서 충전 한도/일정을 설정/지울 수 있습니다.
6	**사후 조건**	진행 중인 트랜잭션은 외부 시스템에서 수신한 충전 한도에 따라 제한됩니다. CSMS는 외부 시스템에서 부과한 새로운 한도/일정을 통보받습니다.

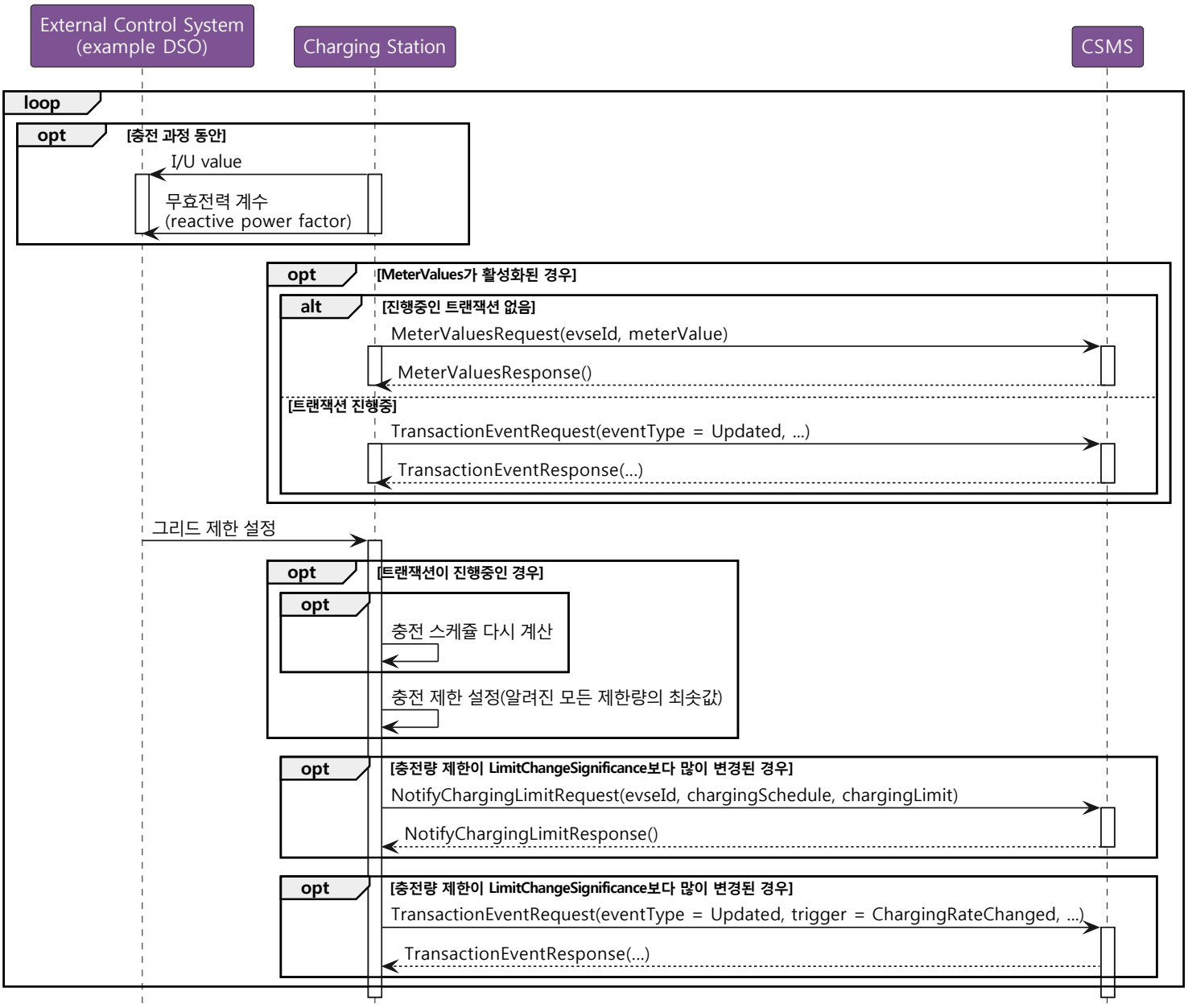

Figure 110. Sequence diagram of the use case "Setting / Updating External Charging Limit with Ongoing Transaction"

7	**오류 처리**	해당 없음
8	**비고**	예를 들어 외부 시스템은 IEC 61850 [IEC61850-7-420] 또는 OpenADR [OPENADR]을 사용하여 그리드 한계를 충전기에 전달할 수 있지만 이는 어떠한 프로토콜일 수 있습니다. 또한, 외부 시스템의 예로는 전력망 문제 발생 시 외부 충전 한도를 설정할 수 있는 DSO가 제시되지만, 이는 충전 한도를 설정할 수 있는 다른 외부 시스템이나 이유일 수도 있습니다.

K11 - Set / Update External Charging Limit With Ongoing Transaction - Requirements

Table 196. K11 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
K11.FR.01	진행 중인 트랜잭션 중에 외부 충전 한도/일정을 수신하는 경우	충전기는 이 주어진 한도/일정보다 더 빠르게 진행 중인 트랜잭션을 충전해서는 안 됩니다.	

ID	전제 조건	요구 사항 정의	참고
K11.FR.02	K11.FR.01 AND 충전 한도가 `LimitChangeSignificance` 이상 변경됨	충전기는 NotifyChargingLimitRequest를 보내 외부 시스템에서 부과한 새로운 충전 한도/일정을 CSMS에 알려야 합니다.	
K11.FR.03	K11.FR.02 AND EnableNotifyChargingLimitWithSchedules가 참입니다.	NotifyChargingLimitRequest에는 외부 시스템에서 설정한 충전 한도/일정이 포함되어야 합니다.	
K11.FR.04	K11.FR.01 AND 충전 속도가 `LimitChangeSignificance` 이상 변경됨	충전기는 TransactionEventRequest 메시지를 CSMS로 보내야 합니다. 이때 trigger = ChargingRateChanged가 있어야 합니다.	
K11.FR.05	K11.FR.02	충전기는 NotifyChargingLimitRequest에서 chargingLimitSource를 CSO로 설정해서는 안 됩니다.	
K11.FR.06	외부 충전 한도/일정을 수신하는 경우	충전기는 이 한도에 대해 보고할 때 목적 ChargingStationExternalConstraints를 사용해야 합니다(예: ReportChargingProfilesRequest).	CSMS가 (향후) 충전 프로파일에 사용할 수 있는 *id*와 충돌할 위험을 최소화하기 위해 `ChargingStationExternalConstraints` 프로파일의 *id* 에 대해 음수 값을 사용하는 것을 권장합니다.

K12 - Set / Update External Charging Limit Without Ongoing Transaction

Table 197. K12 - 진행중인 트랜잭션 없이 외부 충전 한도 설정/업데이트

번호	유형	설명
1	**이름**	지속적인 트랜잭션 없이 외부 충전 한도 설정/업데이트
2	**ID**	K12
{nbso}	*기능 블록*	K. 스마트 충전
3	**목표**	새로운 트랜잭션이 그리드 연결에 대해 충전기에서 외부 시스템이 부과한 충전 일정이나 충전 한도를 CSMS에 알리는 것.
4	**설명**	외부 제어 시스템이 충전기에 충전 한도를 전송합니다. 이 한도는 CSMS로 전송됩니다.
	Actors	외부 제어 시스템, 충전기, CSMS

번호	유형	설명
	시나리오 설명	**1.** 외부 제어 시스템이 충전기에 충전 한도를 전송합니다(트랜잭션 중이 아님). **2.** 선택 사항: 충전기가 새로운 충전 일정을 계산합니다. **3.** 충전기가 충전 속도를 조정합니다. **4.** 충전 한도가 `LimitChangeSignificance` 이상 변경된 경우 충전기는 NotifyChargingLimitRequest 메시지를 CSMS에 전송하고 선택적으로 설정된 충전 한도/일정을 전송합니다. **5.** CSMS가 충전기에 NotifyChargingLimitResponse로 응답합니다.
5	**필수 조건**	충전기가 오류 상태가 아닙니다. OCPP가 아닌 다른 연결을 통해 충전기에서 충전 한도/일정을 설정/지울 수 있는 외부 시스템.
6	**사후 조건**	새로운 트랜잭션은 외부 시스템에서 수신한 충전 한도에 따라 제한됩니다. CSMS는 외부 시스템에서 부과한 새로운 한도/일정을 통보받습니다.

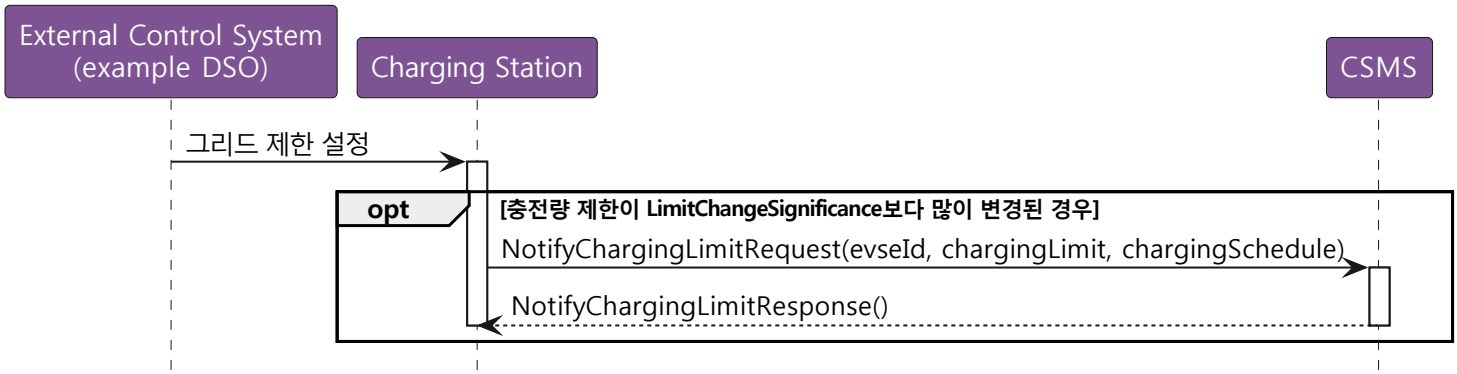

Figure 111. Sequence diagram of the use case "Set / Update External Charging Limit Without Ongoing Transaction"

7	**오류 처리**	해당 없음
8	**비고**	예를 들어 외부 시스템은 IEC 61850 [IEC61850-7-420] 또는 OpenADR [OPENADR]을 사용하여 그리드 한계를 충전기에 전달할 수 있지만 이는 모든 프로토콜일 수 있습니다. 또한, 외부 시스템의 예로는 전력망 문제 발생 시 외부 충전 한도를 설정할 수 있는 DSO가 제시되지만, 이는 충전 한도를 설정할 수 있는 다른 외부 시스템이나 이유일 수도 있습니다.

K12 - Set / Update External Charging Limit Without Ongoing Transaction - Requirements

Table 198. K12 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
K12.FR.01	트랜잭션이 진행 중이 아닌 상태에서 외부 충전 한도/일정을 수신한 경우	모든 EVSE의 총 부하는 이 주어진 한도를 초과해서는 안 됩니다.	
K12.FR.02	K12.FR.01 AND 충전 한도가 `LimitChangeSignificance` 이상 변경됨	충전기는 NotifyChargingLimitRequest를 보내 외부 시스템에서 부과한 새로운 충전 한도/일정을 CSMS에 알려야 합니다.	
K12.FR.03	K12.FR.02 AND `EnableNotifyChargingLimitWithSchedules`가 참입니다.	NotifyChargingLimitRequest에는 외부 시스템에서 설정한 충전 한도/일정이 포함되어야 합니다.	

ID	전제 조건	요구 사항 정의	참고
K12.FR.04	K12.FR.02	충전기는 NotifyChargingLimitRequest에서 chargingLimitSource를 CSO로 설정해서는 안 됩니다.	
K12.FR.05	외부 충전 한도/일정을 수신한 경우	충전기는 이 한도에 대해 보고할 때(예: ReportChargingProfilesRequest에서) 목적 ChargingStationExternalConstraints를 사용해야 합니다.	CSMS가 (향후) 충전 프로파일에 사용할 수 있는 *id*와 충돌할 위험을 최소화하기 위해 `ChargingStationExternalConstraints` 프로파일의 *id* 에 음수 값을 사용하는 것이 좋습니다.

K13 - Reset / Release External Charging Limit

Table 199. K13 - 외부 충전 한도 초기화/해제

번호	유형	설명
1	**이름**	외부 충전 한도 초기화/해제
2	**ID**	K13
	기능 블록	K. 스마트 충전
3	**목표**	이전에 부과된 충전 한도를 해제합니다.
4	**설명**	외부 제어 시스템은 이전에 부과된 충전 한도를 해제하라는 신호를 충전기로 보냅니다. 충전기는 CSMS에 이를 알립니다.
	Actors	외부 제어 시스템, 충전기, CSMS
	시나리오 설명	**1.** 외부 제어 시스템이 충전기에서 충전 한도/일정을 해제/제거합니다. **2.** 트랜잭션이 진행 중일 때 충전기는 새로운 충전 일정을 계산하고 충전 속도를 조정합니다. **3.** 충전기는 ClearedChargingLimitRequest를 보내 CSMS에 알립니다. **4.** CSMS는 ClearedChargingLimitResponse로 충전기에 확인합니다. **5.** 변경 사항이 충전 진행 중인 트랜잭션에 영향을 미치고 다음 `LimitChangeSignificance`보다 큰 경우, 충전기는 TransactionEventRequest를 보내 CSMS에 알립니다. **6.** CSMS는 TransactionEventResponse로 충전기에 확인합니다.
5	**필수 조건**	이전에 고려 중인 충전기에 충전 한도가 전송되었습니다. OCPP가 아닌 다른 연결을 통해 충전기에서 충전 한도/일정을 설정/지울 수 있는 외부 시스템.
6	**사후 조건**	이전에 수신된 충전 한도가 더 이상 충전을 제한하지 않습니다.

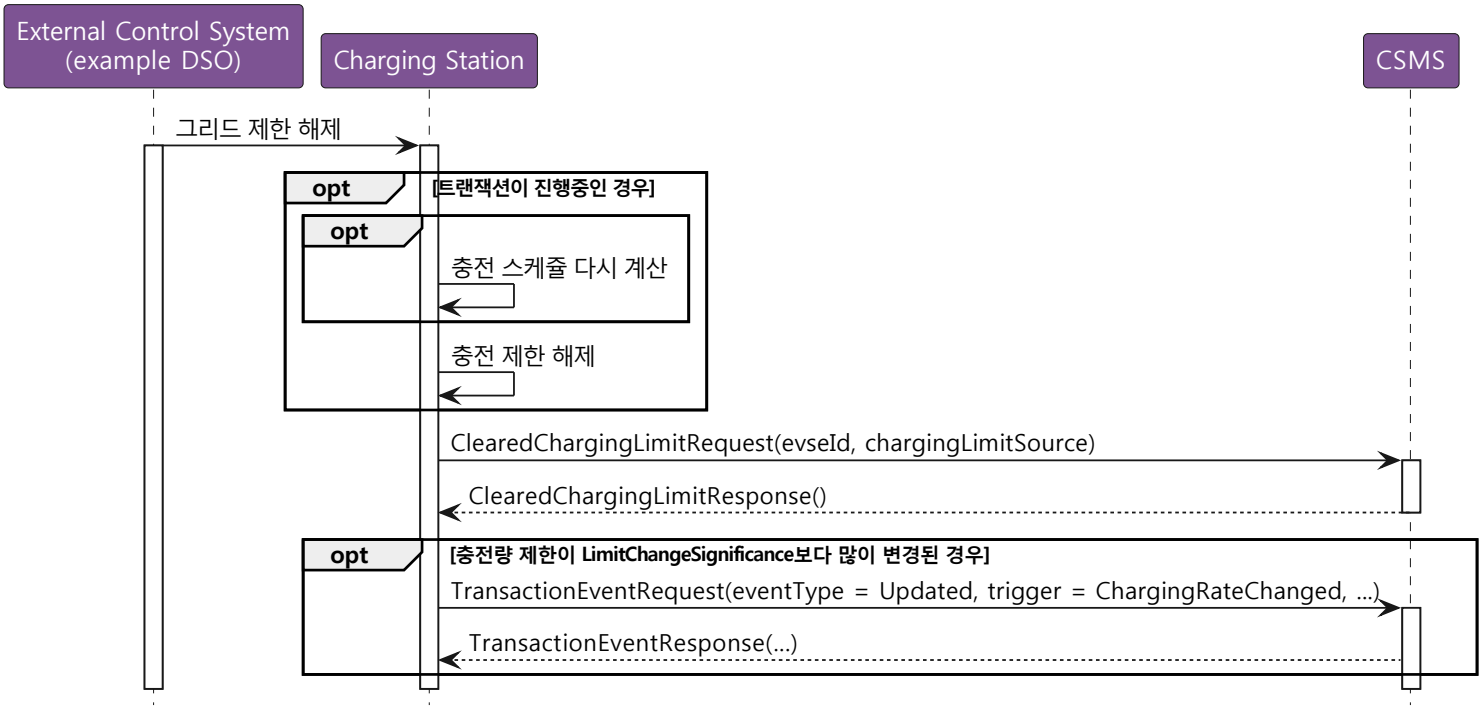

Figure 112. "외부 충전 한도 리셋 / 릴리즈" 유즈케이스 시퀀스 다이어그램

7	**오류 처리**	해당 없음
8	**설명**	외부 시스템은 예를 들어 IEC 61850 [IEC61850-7-420] 또는 OpenADR [OPENADR]을 사용하여 그리드 제한을 해제할 수 있지만 이는 모든 프로토콜일 수 있습니다. 또한, 외부 시스템의 예로는 전력망 문제 발생 시 외부 충전 한도를 설정할 수 있는 DSO가 제시되지만, 이는 충전 한도를 설정할 수 있는 다른 외부 시스템이나 이유일 수도 있습니다.

K13 - Reset / Release External Charging Limit - Requirements

Table 200. K13 - 요구 사항

ID	전제 조건	요구 사항 정의
K13.FR.01	트랜잭션이 진행 중 AND 외부 충전 한도가 해제/제거됨	충전기는 이전에 수신한 한도에 따라 더 이상 충전을 제한해서는 안 됩니다.
K13.FR.02	K13.FR.01	충전기는 ClearedChargingLimitRequest 메시지를 보내 CSMS에 알려야 합니다.
K13.FR.03	K13.FR.01 AND 충전 속도가 `LimitChangeSignificance` 이상 변경됨	충전기는 TransactionEventRequest 메시지를 trigger = ChargingRateChanged로 CSMS에 보내야 합니다.

K14 - External Charging Limit with Local Controller

Table 201. K14 - 로컬 컨트롤러를 사용한 외부 충전 한도

번호	유형	설명
1	**이름**	로컬 컨트롤러로 외부 충전 한도 처리

번호	유형	설명
2	**ID**	K14
	기능 블록	K. 스마트 충전
3	**목표**	외부 제어 시스템 요구 사항에 따라 충전 한도를 조정합니다.
4	**설명**	외부 제어 시스템이 로컬 컨트롤러에 충전 한도를 보냅니다. 로컬 컨트롤러는 CSMS에 알리고, 새로운 충전 일정을 계산하고, 충전 프로파일이 변경된 모든 충전기에 SetChargingProfileRequest 메시지를 보냅니다.
	Actors	외부 제어 시스템, 로컬 컨트롤러, 충전기, CSMS
	시나리오 설명	**1.** 외부 제어 시스템이 로컬 컨트롤러에 충전 한도/일정을 보냅니다. **2.** 로컬 컨트롤러가 NotifyChargingLimitRequest 메시지를 CSMS로 전송합니다. **3.** 로컬 컨트롤러가 연결된 모든 충전기에 대한 새로운 충전 프로파일을 계산합니다. **4.** 로컬 컨트롤러가 충전 프로파일이 변경된 모든 충전기에 SetChargingProfileRequest 메시지를 전송합니다. **5.** 외부 제어 시스템이 충전 한도/일정을 로컬 컨트롤러로 전송합니다. **6.** 로컬 컨트롤러가 ClearedChargingLimitRequest 메시지를 CSMS로 전송합니다. **7.** 로컬 컨트롤러가 연결된 모든 충전기에 대한 새로운 충전 프로파일을 계산합니다. **8.** 로컬 컨트롤러가 영향을 받는 모든 충전기에 ClearChargingProfileRequest 메시지를 전송합니다.
5	**필수 조건**	진행 중인 트랜잭션. OCPP가 아닌 다른 연결을 통해 로컬 컨트롤러에서 충전 한도/일정을 설정/지울 수 있는 외부 시스템.
6	**사후 조건**	**성공한 사후 조건:** 진행 중인 트랜잭션은 외부 시스템에서 수신한 충전 한도에 의해 제한됩니다. CSMS는 외부 시스템에서 부과한 새로운 한도/일정에 대해 알림을 받습니다. **실패한 사후 조건:** CSMS는 변경된 충전 한도에 대해 알림을 받지 못합니다. 외부 제어 시스템은 충전 한도를 변경할 수 없습니다.

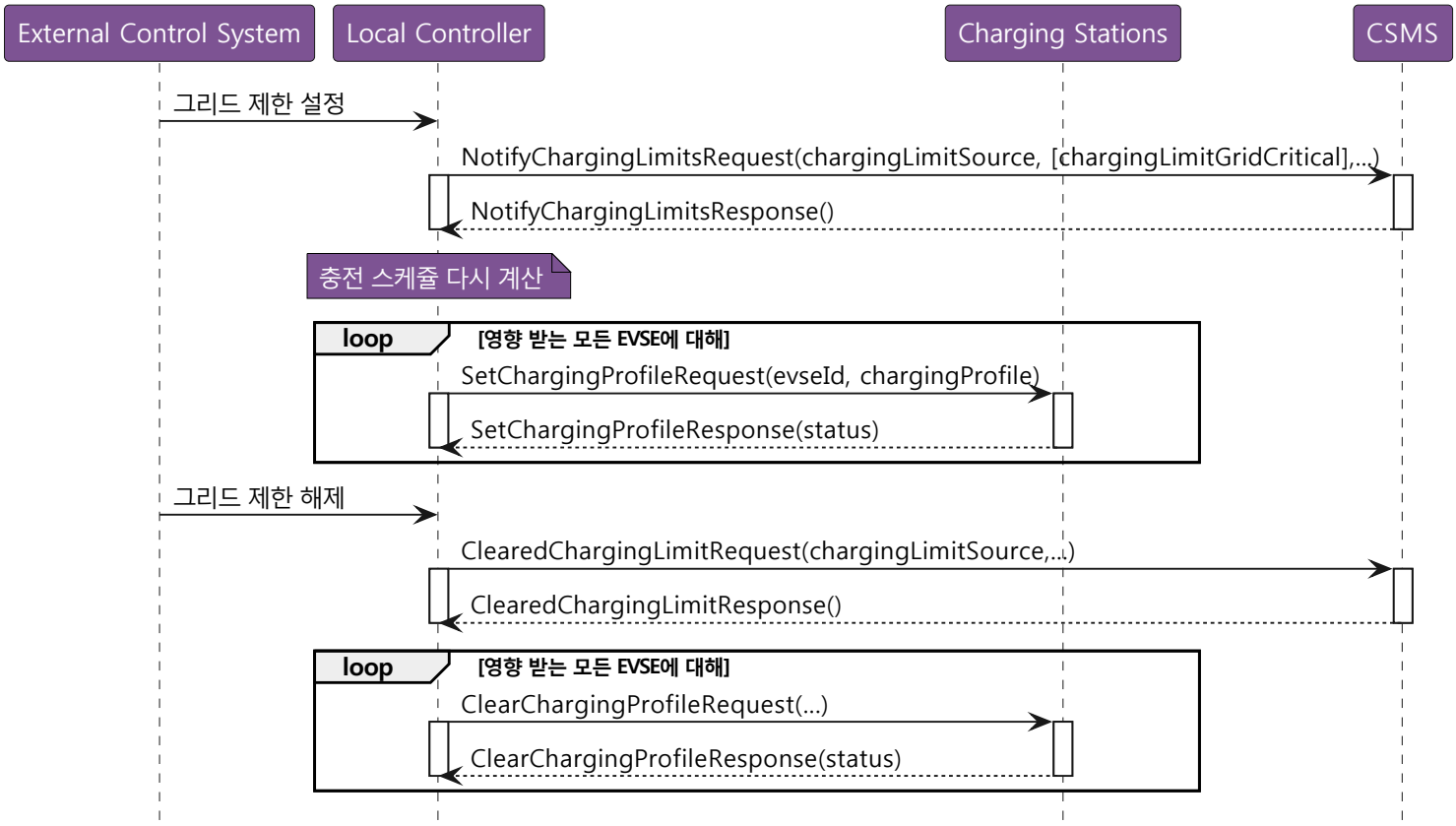

Figure 113. 시퀀스 다이어그램: 로컬 컨트롤러가 있는 외부 충전 한도

7	**오류 처리**	n/a
8	**참고**	n/a

K14 - External Charging Limit with Local Controller - Requirements

Table 202. K14 - Requirements

ID	Precondition	Requirement definition
K14.FR.01	외부 충전 한도/일정이 수신된 경우	모든 충전기의 총 부하는 이 주어진 한도를 초과해서는 안 됩니다.
K14.FR.02	K14.FR.01 AND 충전 한도가 `LimitChangeSignificance` 이상 변경된 경우	로컬 컨트롤러는 NotifyChargingLimitRequest를 보내 외부 시스템에서 부과한 새로운 충전 한도/일정을 CSMS에 알려야 합니다.
K14.FR.03	외부 충전 한도/일정이 해제된 경우	로컬 컨트롤러는 ClearedChargingLimitRequest를 보내 CSMS에 알려야 합니다.
K14.FR.04	K14.FR.03	로컬 컨트롤러는 충전기에 ClearChargingProfileRequest 메시지를 보내 충전기의 hard 한도를 지워야 합니다.
K14.FR.05	로컬 컨트롤러가 외부 충전 한도/일정을 수신하면	충전 프로파일이 변경된 모든 충전기에 SetChargingProfileRequest를 보내야 합니다.
K14.FR.06	K14.FR.05	로컬 컨트롤러는 chargingProfilePurpose를 ChargingStationExternalConstraints로 설정해서는 안 됩니다.

5.3. ISO 15118 based Smart Charging

K15 - Charging with load leveling based on High Level Communication

Table 203. K15 - 고수준 통신을 기반으로 한 부하 평준화로 충전

번호	유형	설명
1	**이름**	고수준 통신을 기반으로 한 부하 평준화로 충전.
2	**ID**	K15
	기능 블록	K. 스마트 충전
	참조	ISO15118-1 E1 AC 충전(고수준 통신 기반 부하 레벨링) 및 E4 DC 충전(고수준 통신 기반 부하 레벨링).
3	**목표**	ISO15118-1, 사용 사례 목표 E1, 29페이지를 참조하세요.
4	**설명**	ISO15118-1, 사용 사례 설명 E1, 29페이지를 참조하세요.
	Actors	EV, 충전기, CSMS.
5	**통합 시나리오 설명**	**1.** EV가 충전기에 ChargeParameterDiscoveryReq 메시지를 보냅니다. **2**. 충전기는 NotifyEVChargingNeedsRequest 메시지를 CSMS로 보냅니다. **3**. CSMS는 NotifyEVChargingNeedsResponse 메시지를 충전기로 보냅니다. **4.** CSMS는 SetChargingProfileRequest 메시지를 충전기로 보냅니다. **5.** 충전기는 SetChargingProfileResponse 메시지를 CSMS로 보냅니다. **6.** 충전기는 EV에 ChargeParameterDiscoveryRes 메시지를 보내 EV에 응답합니다. **7.** EV는 ChargeProgress=Start로 PowerDeliveryReq 메시지를 충전기로 보냅니다. 이는 EVSE가 출력 전원 콘센트에 전압을 공급하고 EV가 배터리를 재충전하기 시작할 수 있는 시점을 표시합니다. **8.** 접촉기가 닫힙니다. **9.** 트랜잭션은 TransactionEventRequest 메시지로 업데이트됩니다. **10.** PowerdeliveryRes 메시지가 EV로 전송됩니다. **11.** 선택적으로 충전기는 NotifyEVChargingScheduleRequest 메시지를 CSMS로 전송합니다.
6	**필수 조건**	충전기와 EV는 모두 ISO 15118을 지원합니다. 구성된 TxStartPoint에는 ParkingBayOccupied, EVConnected, Authorized 또는 PowerPathClosed 중 하나 이상이 포함되어야 하므로 EV에서 ChargeParameterDiscoverReq를 보내기 전에 OCPP 트랜잭션이 시작되어 CSMS가 TxProfile 충전 프로파일을 보낼 수 있습니다.
7	**사후 조건**	ISO15118-1 참조, 사용 사례 종료 조건 E1, 29페이지.

Figure 114. 시퀀스 다이어그램: 고수준 통신을 기반으로 한 부하 평준화 충전

8	**오류 처리**	충전기는 SetChargingProfileRequest 메시지의 정보를 사용하여 EV에 대한 ISO 15118 ChargeParameterDiscoveryReq에 대한 응답을 생성해야 합니다. 이 메시지의 시간 초과는 60초입니다. 즉, NotifyEVChargingNeedsRequest를 수신한 후 60초 이내에 SetChargingProfileRequest를 보내야 합니다. 충전기가 SetChargingProfileRequest를 제때 수신하지 못하거나 NotifyEVChargingNeedsResponse가 *status* = `Processing` 인 경우 충전기는 EVSE에 맞는 일정을 ChargeParameterDiscoverRes에 반환합니다. CSMS가 나중에 SetChargingProfileRequest를 보내면 사용 사례 K16 - Renegotiation started by CSMS에 따라 재협상이 트리거됩니다.
9	**참고**	Signed SalesTariffs는 현재 지원되지 않습니다. 필요한 경우 P01 - Data Transfer to the Charging Station을 사용하여 충전기로 보내주세요.

K15 - Charging with load leveling based on High Level Communication - Requirements

Table 204. K15 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
K15.FR.01	충전기가 EV에서 충전 요구 사항을 수신하는 경우	충전기는 CSMS에 NotifyEVChargingNeedsRequest를 보내야 합니다.	
K15.FR.02	K15.FR.01	NotifyEVChargingNeedsRequest에 대한 응답으로 CSMS는 NotifyEVChargingNeedsResponse를 보내야 합니다.	

ID	전제 조건	요구 사항 정의	참고
K15.FR.03	K15.FR.02	CSMS가 충전 일정을 제공할 수 있는 경우 NotifyEVChargingNeedsResponse의 상태 필드를 'Accepted'으로 설정하여 이를 표시해야 합니다.	
K15.FR.04	K15.FR.02	CSMS가 충전 일정을 제공할 수 없는 경우, NotifyEVChargingNeedsResponse의 상태 필드를 'Rejected'으로 설정하여 이를 표시해야 합니다.	
K15.FR.05	K15.FR.02	CSMS가 충전 일정을 제공할 수 있지만 처리 시간이 필요한 경우, NotifyEVChargingNeedsResponse의 상태 필드를 'Processing'으로 설정하여 이를 표시해야 합니다.	충전기는 SetChargingProfileRequest를 기다릴 필요가 없습니다. CSMS는 나중에 이를 보내고 사용 사례 K16에 따라 재협상을 트리거합니다.
K15.FR.06		NotifyEVChargingNeedsRequest에는 ACChargingParameters 또는 DCChargingParameters가 포함되어야 합니다.	
K15.FR.07	K15.FR.03 또는 K15.FR.05	CSMS는 SetChargingProfileRequest를 *chargingProfilePurpose* = `TxProfile`, *transactionId* 및 최대 3개의 *chargingSchedule* 및 선택적 *salesTariff* 요소와 함께 보내야 하며, 각각은 NotifyEVChargingNeedsRequest의 *maxScheduleTuples* 및 장치 모델 변수 `SmartChargingCtrlr.PeriodsPerSchedule` 로 지정된 기간보다 더 많은 기간을 포함하지 않아야 합니다.	충전기는 EVSE에 대한 복합 일정을 계산하고(`ChargingStationMaxProfile` 또는 `ChargingStationExternalConstraints` 가 있는 경우 고려) 이를 ISO 15118의 SAScheduleList 형식으로 변환합니다.
K15.FR.08	K15.FR.01	CSMS는 60초 이내에 SetChargingProfileRequest를 충전기로 보내야 합니다.	이는 ISO 15118 ChargeParameterDiscoveryReq 시간 초과를 충족하기 위한 것입니다.
K15.FR.09	K15.FR.07 AND EV가 충전 프로파일을 반환합니다.	충전기는 제공된 충전 프로파일이 CSMS의 ChargingSchedule 경계 내에 있는지 확인해야 합니다.	ISO 15118에서 EV는 PowerDeliveryReq의 일부로 충전 프로파일을 보낼 수 있습니다.
K15.FR.10	K15.FR.09	충전기는 EV 충전 프로파일을 NotifyEVChargingScheduleRequest 메시지로 CSMS에 전송해야 합니다.	
K15.FR.11	K15.FR.10 그리고 EV 충전 프로파일이 CSMS 충전 일정의 한도 내에 있습니다.	CSMS는 NotifyEVChargingScheduleResponse로 충전기에 *status* `Accepted` 로 응답합니다.	참고: 충전기에서 이미 확인했지만 CSMS가 직접 확인합니다.
K15.FR.12	K15.FR.10 AND EV 충전 프로파일이 CSMS 충전 일정의 한도 내에 있지 않음	CSMS가 NotifyEVChargingScheduleResponse로 충전기에 *status* `Rejected` 로 응답합니다.	

ID	전제 조건	요구 사항 정의	참고
K15.FR.13	K15.FR.12	CSMS가 사용 사례 K16에 따라 새로운 재협상을 시작합니다.	
K15.FR.14	K15.FR.11	충전기는 실제 복합 일정을 계산할 때 NotifyEVChargingScheduleRequest의 일정을 고려해야 합니다.	
K15.FR.15	K15.FR.01 AND 충전기가 오프라인입니다.	충전기는 TxDefaultProfile(있는 경우)을 사용하고 복합 일정의 한계 내에서 충전 일정을 생성해야 합니다.	
K15.FR.16	K15.FR.07	충전기를 구성하여 NotifyEVChargingNeedsRequest 메시지 이전에 idToken이 포함된 TransactionEvent를 보내도록 하는 것이 좋습니다. 이렇게 하면 CSMS가 충전 일정을 생성할 때 사용자를 고려할 수 있습니다.	
K15.FR.17	충전기가 트랜잭션이 시작된 직후 SetChargingProfileRequest를 수신하고 NotifyEVChargingNeedsRequest를 CSMS로 보내기 전에 수신하는 경우	충전기는 SetChargingProfileResponse를 *status* = `Rejected` 로 응답하고 *statusInfo* 를 *reasonCode* = `InvalidMessageSeq` 로 응답해야 합니다.	CSMS가 프로파일을 너무 일찍 보냈습니다. CSMS가 NotifyEVChargingNeedsRequest를 수신할 때 다시 충전 프로파일을 보내기만 하면 충전기가 충전 프로파일을 거부하는 대신 수락해도 문제가 없습니다.
K15.FR.18	K15.FR.03 또는 K15.FR.05	CSMS는 SetChargingProfileRequest에서 하나의 *chargingSchedule* 만 사용하는 것이 좋습니다.	이렇게 하면 NotifyEVChargingScheduleRequest를 수신하지 않더라도 EV가 어떤 일정을 따를지 의심할 여지가 없습니다.
K15.FR.19	K15.FR.07 그리고 EV가 충전 프로파일을 반환하지 않음	충전기는 EV가 선택한 일정과 일치하는 CSMS에 NotifyEVChargingScheduleRequest 메시지의 chargingSchedule로 EV 충전 프로파일을 반환하는 것이 좋습니다(예: chargingSchedule.id = SAScheduleTupleId)	ISO 15118-2에서 EV 충전 프로파일과 선택한 일정은 PowerDeliveryReq에서 *ChargingProfile* 과 *SAScheduleTupleId* 로 반환됩니다.

K16 - Renegotiation initiated by CSMS

Table 205. K16 - Renegotiation initiated by CSMS

번호	유형	설명
1	**이름**	CSMS에서 재협상 시작
2	**ID**	K16
	기능 블록	K. 스마트 충전
3	**목표**	충전기의 충전 전력 또는 전류 제어

번호	유형	설명
4	**설명**	CSMS는 EV에서 사용하는 전력 또는 전류에 영향을 미치기 위해 충전기에 SetChargingProfileRequest를 보냅니다. CSMS는 외부 시스템에서 부과할 수 있는 한도 내에 있도록 ChargingSchedule을 계산합니다. **참고:** EV와 충전기 간의 동작에 대한 설명은 정보 제공용일 뿐 OCPP에서 의무화한 것이 아닙니다.
	Actors	EV, 충전기, CSMS
	시나리오 설명	**1** CSMS는 충전기에 SetChargingProfileRequest를 보냅니다. **2** 충전기는 SetChargingProfileResponse로 CSMS에 응답합니다. **3** EV가 다음 CurrentDemandReq(DC의 경우) 또는 ChargingStatusReq(AC의 경우)를 보내면 충전기는 *evseNotification* = `ReNegotiation` 으로 응답합니다. **4** EV는 *chargeProgress* = `ReNegotiate` 로 PowerDeliveryReq를 보내 이를 확인합니다. **5** 충전기는 PowerDeliveryRes로 응답합니다. **6** EV는 ChargeParameterDiscoveryReq를 보냅니다. **7** 충전기는 SetChargingProfileRequest의 ChargingSchedule 데이터가 포함된 SAScheduleList로 ChargeParameterDiscoveryRes로 응답합니다. **8** EV는 *chargeProgress* = `Start`(선택적 충전 프로파일 포함)로 PowerDeliveryReq를 보내 이를 확인합니다. **9** 충전기는 PowerDeliveryRes로 응답하고, 재협상 시작 시 충전이 중단된 경우 전력 공급을 재개합니다. **10** EV가 이전 단계에서 충전 프로파일을 제공한 경우 충전기는 NotifyEVChargingScheduleRequest를 CSMS로 보냅니다.
5	**필수 조건**	사용 사례 K15에 따라 충전 세션이 시작되었습니다.
6	**사후 조건**	충전 세션이 새 충전 프로파일을 사용합니다.

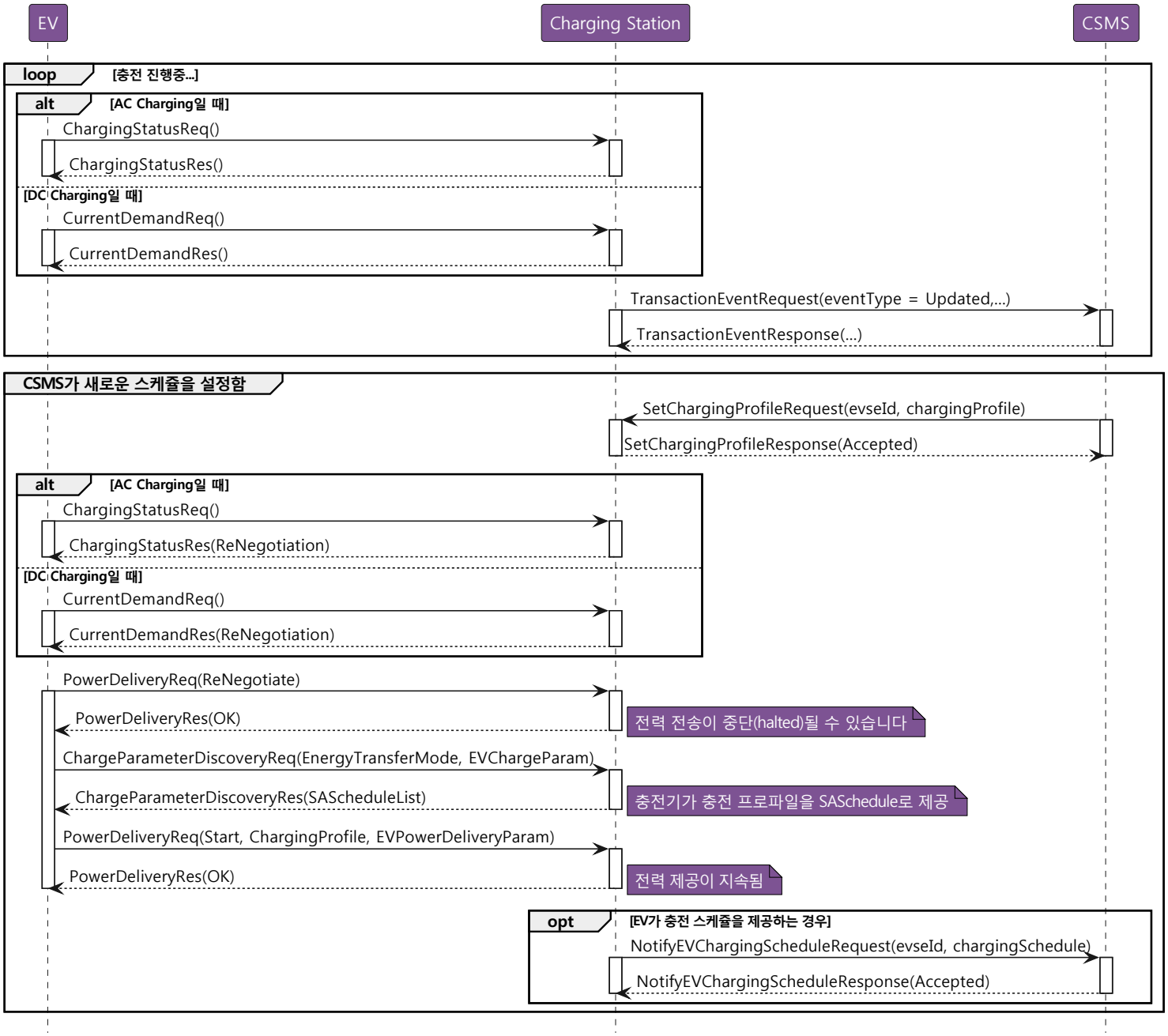

Figure 115. CSMS에서 재협상 시작

7	**비고**	Signed SalesTariffs는 현재 지원되지 않습니다. 필요한 경우 P01 - Data Transfer to the Charging Station을 사용하여 충전기로 보내주세요.

K16 - Renegotiation initiated by CSMS - Requirements

ID	전제 조건	요구 사항 정의	참고
K16.FR.01	CSMS에서 새 SetChargingProfileRequest를 보냅니다.	충전기는 SetChargingProfileResponse로 *status* = `Accepted` 로 응답해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
K16.FR.02	K16.FR.01	충전기는 EV와 일정 재협상을 시작해야 합니다.	ISO 15118에서는 CurrentDemandReq(DC의 경우) 또는 ChargingStatusReq(AC의 경우) 메시지에 EVSENotification=`ReNegotiation` 으로 응답하여 이를 수행합니다.
K16.FR.03	K16.FR.02	충전기는 EV에 ChargingSchedule 데이터를 제공해야 합니다.	ISO 15118에서는 ChargeParameterDiscoverRes 메시지에서 이를 수행합니다.
K16.FR.04	EV가 충전 프로파일을 반환합니다.	충전기는 제공된 충전 프로파일이 CSMS의 ChargingSchedule 경계 내에 있는지 확인해야 합니다.	ISO 15118에서는 EV가 PowerDeliveryReq 메시지의 일부로 이를 제공할 수 있습니다.
K16.FR.05	K16.FR.04	충전기는 EV 충전 프로파일을 NotifyEVChargingScheduleRequest 메시지로 CSMS에 보내야 합니다.	
K16.FR.06	K16.FR.05 AND EV 충전 프로파일이 CSMS ChargingSchedule의 한도 내에 있습니다.	CSMS가 NotifyEVChargingScheduleResponse로 충전기에 *status* `Accepted` 로 응답합니다.	참고: 충전기에서 이미 확인했지만 CSMS가 자체적으로 확인합니다.
K16.FR.07	K16.FR.05 AND EV 충전 프로파일이 CSMS 충전 일정의 한도 내에 있지 않음	CSMS가 NotifyEVChargingScheduleResponse로 응답하고 *status* `Rejected` 로 충전기에 응답합니다.	
K16.FR.08	K16.FR.07	CSMS가 사용 사례 K16에 따라 새로운 재협상을 시작합니다.	
K16.FR.09	충전기가 EV로부터 충전 요구를 수신하는 경우	충전기는 CSMS에 NotifyEVChargingNeedsRequest를 보내면 안 됩니다.	CSMS가 재협상을 시작했고 EV의 초기 충전 요구 사항, EV가 이미 소비한 에너지, CSMS가 충전 프로파일을 업데이트하게 된 모든 정보를 기반으로 새로운 충전 프로파일을 방금 보냈습니다. ISO 15118에서 충전 요구 사항은 ChargeParameter-DiscoveryReq를 통해 전송됩니다.
K16.FR.10	K16.FR.04	충전기는 실제 복합 일정을 계산할 때 NotifyEVChargingScheduleRequest의 일정을 고려해야 합니다.	
K16.FR.11	K16.FR.02 그리고 새로운 충전 일정의 전류 또는 전력이 실제 전류 또는 전력보다 낮음	충전기는 가능한 한 빨리 EV에 새로운 충전 일정과 일치하는 값으로 전류 또는 전력을 낮추도록 요청해야 합니다.	ISO 15118에서 이는 CurrentDemandRes(DC의 경우) 또는 ChargingStatusRes(AC의 경우)에서 전달할 수 있습니다.

ID	전제 조건	요구 사항 정의	참고
K16.FR.12	K16.FR.09 AND 충전기에서 NotifyEVChargingNeedsRequest를 전송합니다.	CSMS는 SetChargingProfileRequest를 전송해야 합니다.	이 상황은 바람직하지 않습니다. 충전 프로파일이 K16.FR.01과 동일할 가능성이 높지만, 충전기가 K16.FR.09를 준수하지 않을 때 견고성을 위해 추가되었습니다.
K16.FR.13	EV가 충전 프로파일을 반환하지 않음	충전기는 EV가 선택한 일정과 일치하는(즉, chargingSchedule.id = SAScheduleTupleId) CSMS에 대한 NotifyEVChargingScheduleRequest 메시지의 chargingSchedule로 EV 충전 프로파일을 반환하는 것이 좋습니다.	ISO 15118-2에서 EV 충전 프로파일과 선택한 일정은 PowerDeliveryReq에서 *ChargingProfile* 과 *SAScheduleTupleId* 로 반환됩니다.

K17 - Renegotiation initiated by EV

Table 206. K17 - EV가 시작한 재협상

번호	유형	설명
1	**이름**	EV가 시작한 재협상.
2	**ID**	K16
	기능 블록	K. 스마트 충전
3	**목표**	EV가 새로운 충전 일정을 요청하도록 합니다.
4	**설명**	EV가 충전기에 재협상을 원한다는 신호를 보내고 새로운 충전 요구 사항을 제공하며, 충전기는 이를 CSMS로 보냅니다. 이 매개변수와 다른 매개변수를 기반으로 CSMS는 새로운 충전 일정을 계산하고 SetChargingProfileRequest를 통해 충전기로 전송하며, 충전기는 이를 EV에 전달합니다. **참고:** EV와 충전기 간의 동작에 대한 설명은 정보 제공용일 뿐 OCPP에서 의무화한 것이 아닙니다.
	Actors	EV, 충전기, CSMS
	시나리오 설명	**1** EV가 충전 요구 사항 매개변수와 함께 ChargeParameterDiscoveryReq를 보내면 충전기는 이 정보를 NotifyEVChargingNeedsRequest에 담아 CSMS로 보냅니다. **2** CSMS가 NotifyEVChargingNeedsResponse로 충전기에 응답합니다. **3** CSMS가 새로운 충전 일정을 계산하여 EV 충전 요구 사항을 수용하려고 하고 다른 매개변수에서 부과한 일정 경계 내에 여전히 들어맞습니다. **4** CSMS가 새로운 일정과 함께 SetChargingProfileRequest를 충전기로 보냅니다. **5** 충전기가 `Accepted` 상태로 SetChargingProfileResponse로 응답합니다. **6** 충전기가 ChargeParameterDiscoveryRes 메시지로 새로운 충전 일정을 EV로 보냅니다. **7** EV는 *chargeProgress* = `Start`(선택적 충전 프로파일 포함)로 PowerDeliveryReq를 보내 이를 확인합니다. **8** 충전기는 PowerDeliveryRes로 응답하고, 재협상 시작 시 충전이 중단된 경우 전력 공급을 재개합니다. **9** EV가 이전 단계에서 충전 프로파일을 제공한 경우 충전기는 NotifyEVChargingScheduleRequest를 CSMS로 보냅니다.
5	**필수 조건**	사용 사례 K15에 따라 충전 세션이 시작되었습니다.
6	**사후 조건**	충전 세션이 새 충전 프로파일을 사용합니다.

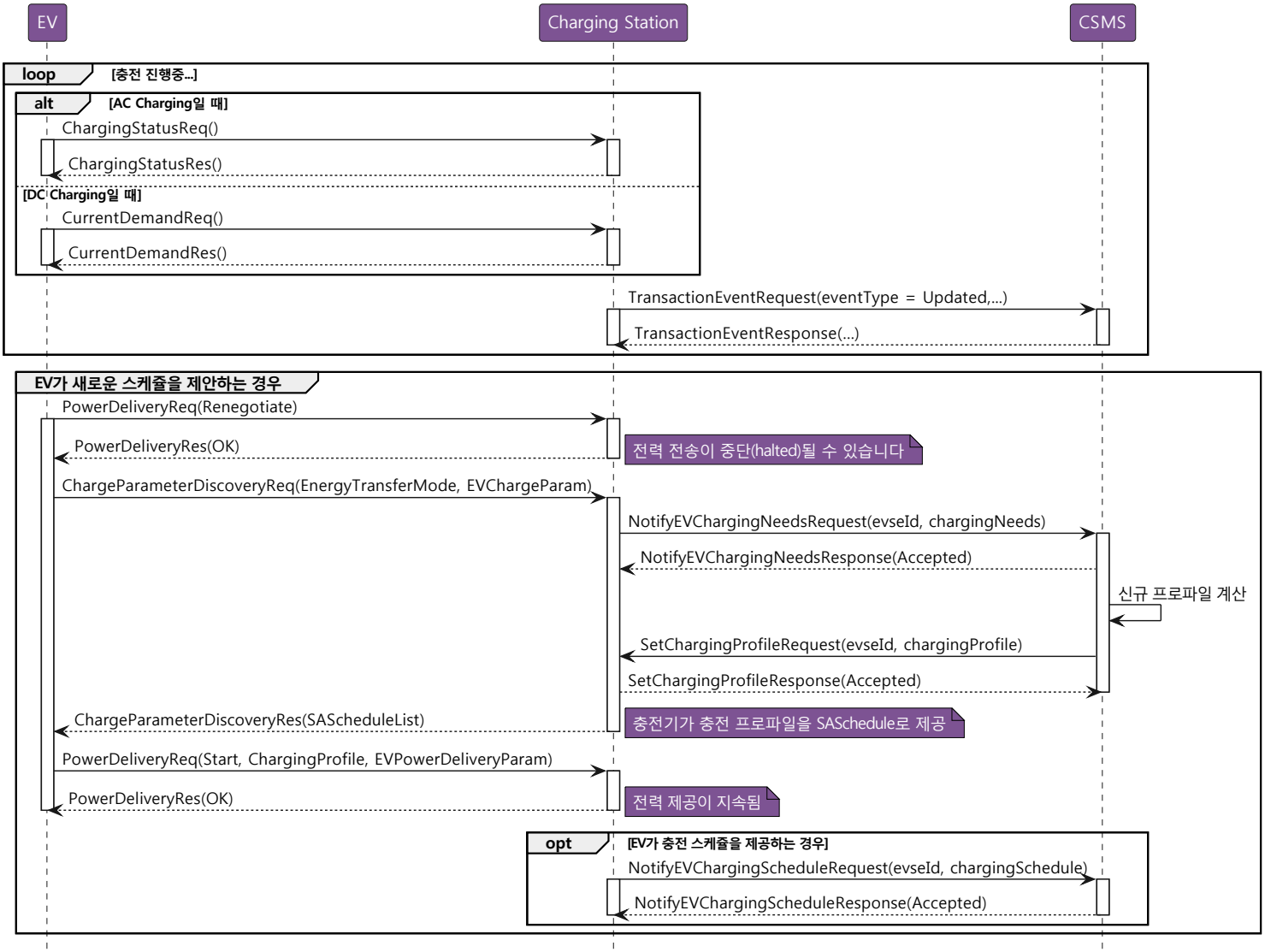

Figure 116. EV가 시작한 재협상

7	**참고**	Signed SalesTariffs는 현재 지원되지 않습니다. 필요한 경우 P01 - Data Transfer to the Charging Station을 사용하여 충전기로 보내주세요.

K17 - Renegotiation initiated by EV - Requirements

Table 207. K17 - 요구 사항

ID	사전 조건	요구 사항 정의	참고
K17.FR.01	EV에서 재협상을 트리거하고 새로운 충전 요구 사항을 전송	충전기는 CSMS에 NotifyEVChargingNeedsRequest를 전송해야 합니다.	
K17.FR.02	K17.FR.01	NotifyEVChargingNeedsRequest에 대한 응답으로 CSMS는 NotifyEVChargingNeedsResponse를 보내야 합니다.	
K17.FR.03	K17.FR.02	CSMS가 충전 일정을 제공할 수 있는 경우 NotifyEVChargingNeedsResponse의 *status* 필드를 'Accepted'로 설정하여 이를 표시해야 합니다.	

ID	사전 조건	요구 사항 정의	참고
K17.FR.04	K17.FR.02	CSMS가 충전 일정을 제공할 수 없는 경우, NotifyEVChargingNeedsResponse의 *status* 필드를 'Rejected'으로 설정하여 이를 표시해야 합니다.	
K17.FR.05	K17.FR.02	CSMS가 충전 일정을 제공할 수 있지만 처리 시간이 필요한 경우, NotifyEVChargingNeedsResponse의 *status* 필드를 'Processing'으로 설정하여 이를 표시해야 합니다.	
K17.FR.06		NotifyEVChargingNeedsRequest에는 ACChargingParameters 또는 DCChargingParameters가 포함되어야 합니다.	
K17.FR.07	K17.FR.03 또는 K17.FR.05	CSMS는 SetChargingProfileRequest를 *chargingProfilePurpose* = `TxProfile` 로 보내고 최대 3개의 *chargingSchedule* 및 선택적 salesTariff 요소를 포함해야 하며, 각각 NotifyEVChargingNeedsRequest의 *maxScheduleTuples* 및 장치 모델 변수 SmartChargingCtrlr.PeriodsPerSchedule로 지정된 기간보다 더 많은 기간을 포함하지 않아야 합니다.	
K17.FR.08	K17.FR.01	CSMS는 60초 이내에 충전기에 SetChargingProfileRequest를 보내야 합니다.	이는 ISO 15118 ChargeParameterDiscoveryReq 시간 초과를 충족하기 위한 것입니다.
K17.FR.09	K17.FR.07 AND EV가 충전 프로파일을 반환합니다.	충전기는 제공된 충전 프로파일이 CSMS의 ChargingSchedule 경계 내에 있는지 확인해야 합니다.	ISO 15118에서 EV는 PowerDeliveryReq의 일부로 충전 프로파일을 보낼 수 있습니다.
K17.FR.10	K17.FR.09	충전기는 EV 충전 프로파일을 NotifyEVChargingScheduleRequest 메시지로 CSMS에 보내야 합니다.	
K17.FR.11	K17.FR.10 AND EV 충전 프로파일이 CSMS ChargingSchedule의 한도 내에 있습니다.	CSMS가 NotifyEVChargingScheduleResponse로 *status* `Accepted` 로 충전기에 응답합니다.	참고: 충전기에서 이미 확인했지만 CSMS가 직접 확인합니다.
K17.FR.12	K17.FR.10 AND EV 충전 프로파일이 CSMS ChargingSchedule의 한도 내에 없습니다.	CSMS가 NotifyEVChargingScheduleResponse로 *status* `Rejected` 로 충전기에 응답합니다.	
K17.FR.13	K17.FR.12	CSMS가 사용 사례 K16에 따라 새로운 재협상을 시작합니다.	
K17.FR.14	K17.FR.11	충전기는 실제 복합 일정을 계산할 때 NotifyEVChargingScheduleRequest의 일정을 고려해야 합니다.	

ID	사전 조건	요구 사항 정의	참고
K17.FR.15	K17.FR.01 그리고 충전기가 오프라인입니다.	충전기는 TxDefaultProfile(있는 경우)을 사용하고 복합 일정의 한계 내에서 충전 일정을 생성해야 합니다.	
K17.FR.16	K17.FR.07 EV가 충전 프로파일을 반환하지 않음	충전기는 EV가 선택한 일정과 일치하는 CSMS에 대한 NotifyEVChargingScheduleRequest 메시지의 chargingSchedule로 EV 충전 프로파일을 반환하는 것이 좋습니다(즉, chargingSchedule.id = SAScheduleTupleId)	ISO 15118-2에서 EV 충전 프로파일과 선택한 일정은 PowerDeliveryReq에서 *ChargingProfile* 과 *SAScheduleTupleId* 로 반환됩니다.

L. FirmwareManagement

1. Introduction

이 기능 블록은 CSO가 충전기의 펌웨어를 업데이트할 수 있는 기능을 설명합니다.

충전기를 새 펌웨어로 업데이트해야 하는 경우 CSMS는 충전기가 새 펌웨어를 다운로드할 수 있는 시간을 충전기에 알립니다. 충전기는 새 펌웨어를 다운로드하고 설치할 때마다 각 단계마다 CSMS에 알려야 합니다.

2. Use cases & Requirements

L01 - Secure Firmware Update

Table 208. L01 - 보안 펌웨어 업데이트

번호	유형	설명
1	**이름**	보안 펌웨어 업데이트
2	**ID**	L01
	기능 블록	L. 펌웨어 관리
3	**목표**	보안 펌웨어 업데이트를 다운로드하고 설치합니다.
4	**설명**	충전기가 보안 펌웨어 업데이트를 처리하는 방법을 설명합니다.
	Actors	CSMS, 충전기, 충전기 제조업체
	시나리오 설명	**1.** CSMS는 펌웨어 위치, 펌웨어를 검색해야 하는 시간, 충전기가 펌웨어 다운로드를 다시 시도해야 하는 횟수에 대한 정보가 포함된 UpdateFirmwareRequest 메시지를 보냅니다. **2.** 충전기는 제조업체 루트 인증서와 인증서의 유효성을 확인합니다. **3.** 인증서가 유효한 경우 충전기는 펌웨어 다운로드를 시작하고 FirmwareStatusNotificationRequest를 Downloading 상태로 보냅니다. 인증서가 유효하지 않거나 확인할 수 없는 경우 충전기는 펌웨어 업데이트 프로세스를 중단하고 상태가 InvalidCertificate인 UpdateFirmwareResponse와 보안 이벤트가 InvalidFirmwareSigningCertificate인 SecurityEventNotificationRequest를 보냅니다(보안 이벤트 전체 목록은 Part 2 - Appendices 참조). **4.** 펌웨어가 성공적으로 다운로드되면 충전소는 FirmwareStatusNotificationRequest를 상태 Downloaded로 전송합니다. 그렇지 않으면 FirmwareStatusNotificationRequest를 상태 DownloadFailed로 전송합니다. **5.** 검증이 성공하면 충전기는 FirmwareStatusNotificationRequest를 상태 Installing으로 전송합니다. 펌웨어 검증이 실패하거나 서명이 전혀 없는 경우 충전기는 FirmwareStatusNotificationRequest를 상태 InvalidSignature로 전송하고 SecurityEventNotificationRequest를 보안 이벤트 InvalidFirmwareSignature로 전송합니다(보안 이벤트 전체 목록은 Part 2 - Appendices 참조). **6.** 설치가 성공하면 충전기는 FirmwareStatusNotificationRequest를 상태 Installed로 전송합니다. 그렇지 않으면 FirmwareStatusNotificationRequest를 상태 InstallationFailed로 보냅니다.
	대체 시나리오	L02 - Non-Secure Firmware Update
5	**필수 조건**	충전기 제조업체에서 펌웨어 업데이트를 제공했습니다.
6	**사후 조건**	**성공 사후 조건:** 펌웨어가 업데이트되고 충전기가 *Installed* 상태입니다. **실패 사후 조건:** 인증서가 유효하지 않거나 확인할 수 없으며 충전기가 *InvalidCertificate* 상태입니다. 펌웨어 다운로드가 실패하고 충전기가 *DownloadFailed* 상태입니다. 펌웨어 디지털 서명 검증에 실패했고 충전기가 *InvalidSignature* 상태입니다. 펌웨어 설치에 성공하지 못했고 충전기가 *InstallationFailed* 상태입니다.

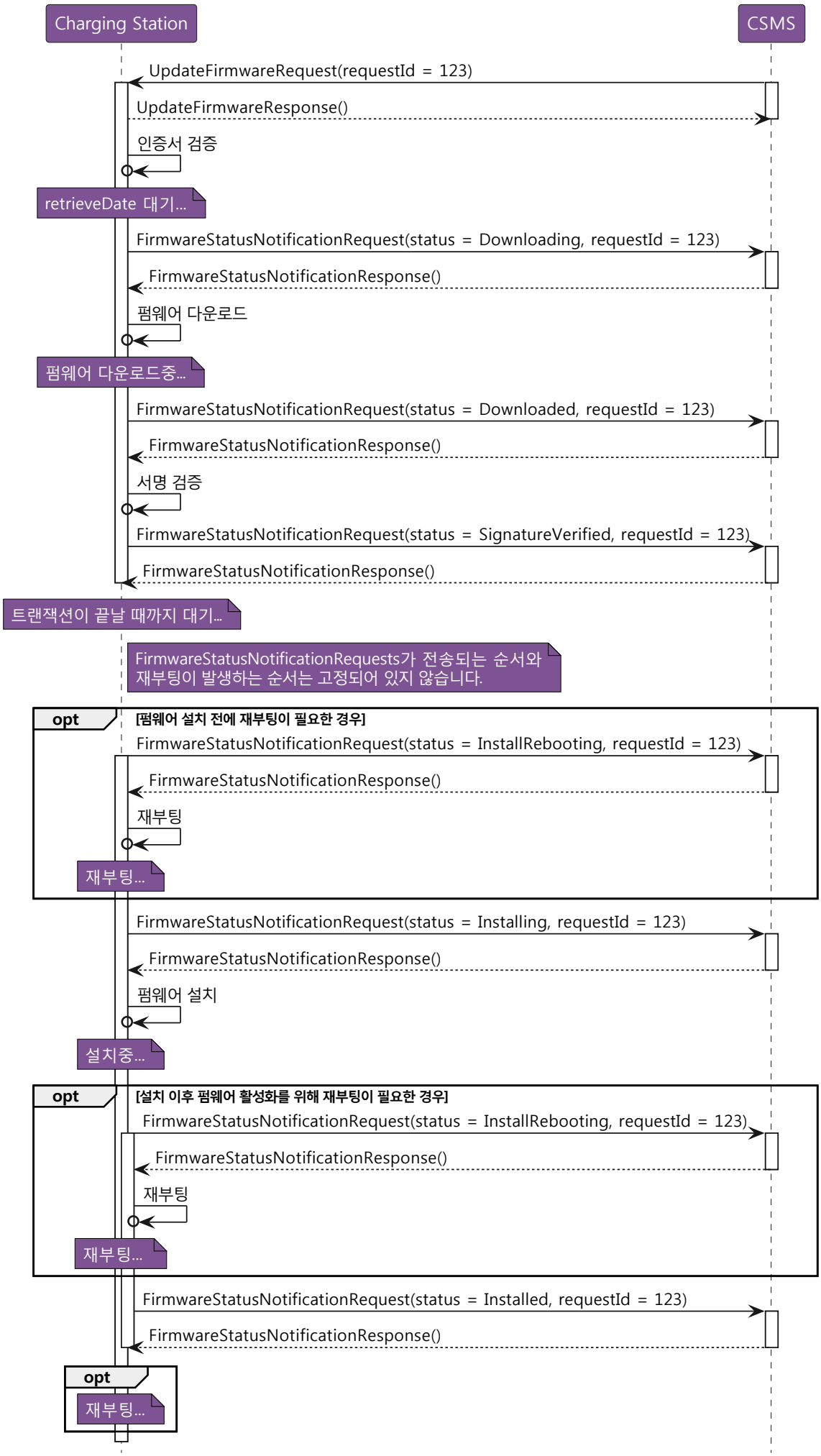

Figure 117. 시퀀스 다이어그램 보안 펌웨어 업그레이드(행복한 흐름)

7	오류 처리	n/a
		이 사용 사례의 예로 requestId = 123이지만 이는 모든 값일 수 있습니다. 펌웨어가 서버나 워크스테이션에 저장될 때 펌웨어를 보호하기 위한 조치를 취해야 합니다. 충전기에는 지원하는 파일 전송 프로토콜을 보고하는 필수 구성 변수가 있습니다: `FileTransferProtocols`
8	참고	새 버전의 OCPP로 마이그레이션할 때 새 구성으로 대체 NetworkConnectionProfile을 설치하는 것이 좋습니다. 펌웨어 서명 인증서에 대한 요구 사항은 Certificate Properties 섹션에 설명되어 있습니다. 제조업체는 충전기의 펌웨어 서명 인증서에 중간 인증서를 사용해서는 안 됩니다. FTP는 가능한 다양한 네트워크 토폴로지를 통과하기 위해 Passive FTP를 사용할 수 있어야 합니다.

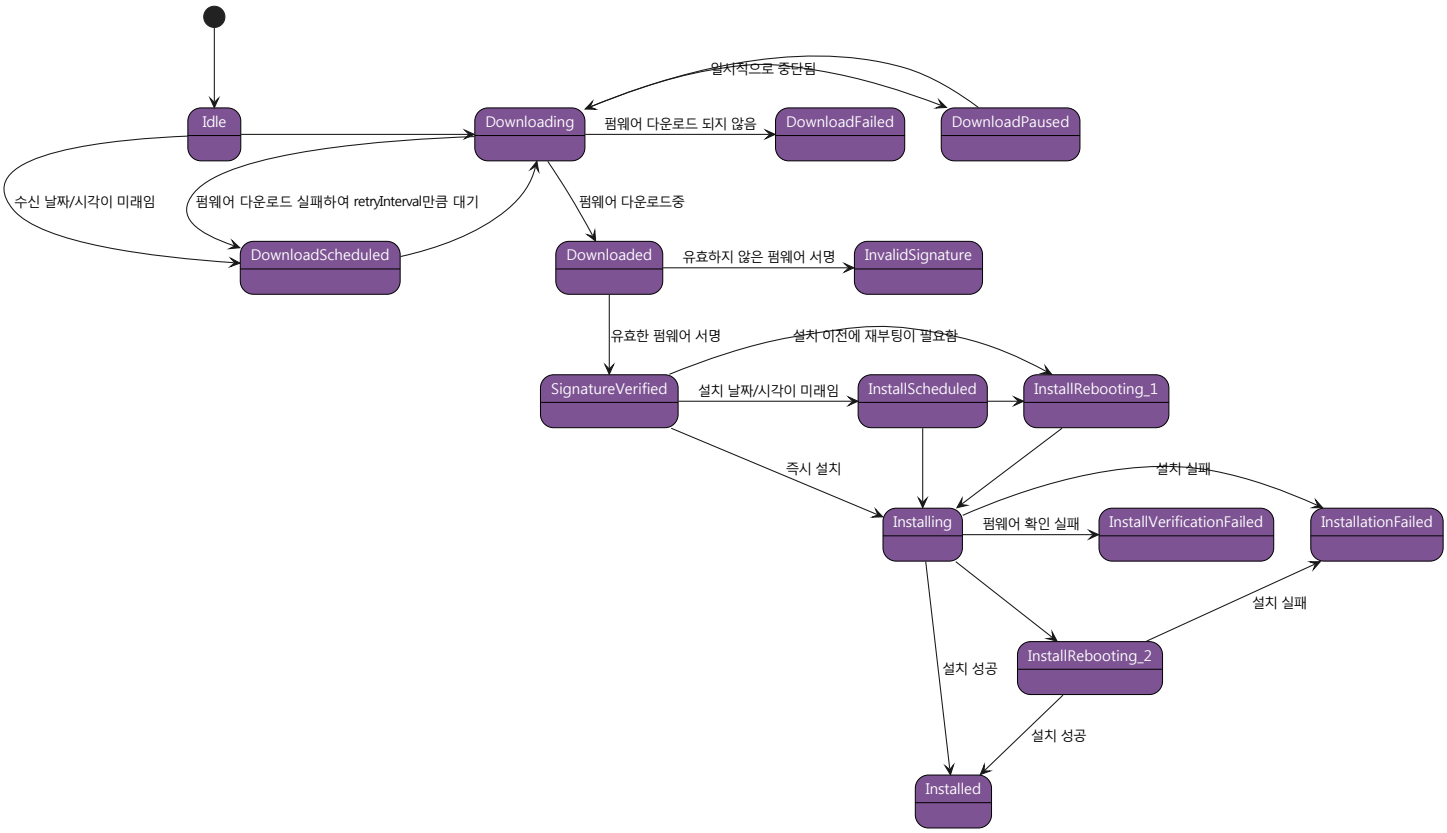

Figure 118. 펌웨어 상태 전환

L01 - Secure Firmware Update - Requirements

Table 209. L01 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
L01.FR.01	충전기가 펌웨어 업데이트 프로세스에서 새로운 상태로 전환될 때마다.	충전기는 이 새로운 상태와 함께 FirmwareStatusNotificationRequest 메시지를 CSMS로 보내야 합니다. 사용 이유는 FirmwareStatusEnumType 설명에 설명되어 있습니다.	

ID	전제 조건	요구 사항 정의	참고
L01.FR.02	충전기가 펌웨어 프로세스에서 잘못된 인증서 상태로 전환될 때.	충전기는 보안 이벤트 InvalidFirmwareSigningCertificate와 함께 CSMS에 SecurityEventNotificationRequest 메시지를 보내야 합니다(보안 이벤트 전체 목록은 Part 2 - Appendices 참조).	
L01.FR.03	충전기가 Invalid Signature 상태로 전환될 때.	충전기는 보안 이벤트 InvalidFirmwareSignature와 함께 CSMS에 SecurityEventNotificationRequest 메시지를 보내야 합니다(보안 이벤트 전체 목록은 Part 2 - Appendices 참조).	
L01.FR.04	충전기가 새 펌웨어를 성공적으로 다운로드할 때	서명은 서명을 위해 RSA-PSS 또는 ECSchnorr 알고리즘을 사용하고 해시 값을 계산하기 위해 SHA256 알고리즘을 사용하여 전체 펌웨어 파일에 대한 서명을 계산하여 검증해야 합니다.	
L01.FR.05	L01.FR.04 AND (*installDateTime* 이 설정되지 않음 또는 현재 시간 >= *installDateTime*)	충전기는 가능한 한 빨리 새 펌웨어를 설치해야 합니다.	
L01.FR.06	L01.FR.05 AND 충전기에 진행 중인 트랜잭션이 있음 AND 트랜잭션이 진행 중인 동안 펌웨어 설치를 시작할 수 없는 경우	충전기는 설치를 시작하기 전에 모든 트랜잭션이 끝날 때까지 기다려야 합니다.	
L01.FR.07	L01.FR.06 또는 L01.FR.33 AND 구성 변수 `AllowNewSessionsPendingFirmwareUpdate`가 *false* 이거나 존재하지 않습니다.	충전기는 진행 중인 트랜잭션이 끝날 때까지 충전기가 사용하지 않는 모든 EVSE를 UNAVAILABLE로 설정해야 합니다. 펌웨어가 설치될 때까지 사용 가능해지는 모든 EVSE는 UNAVAILABLE로 설정해야 합니다.	
L01.FR.08		펌웨어는 충전기로 암호화하여 보내는 것이 좋습니다. 이는 보안 프로토콜(HTTPS, SFTP 또는 FTPS 등)을 사용하여 펌웨어를 보내거나 펌웨어 자체를 암호화한 다음 보내면 됩니다.	
L01.FR.09		펌웨어 업데이트는 진위성을 보장하고 출처 증명을 제공하기 위해 디지털로 보호되어야 합니다.	이 보호는 펌웨어 이미지의 해시 값에 디지털 서명을 적용하여 달성됩니다. 이상적으로는 이 서명이 제조업체에서 이미 계산되어 있어야 합니다. 이렇게 하면 펌웨어 이미지의 출처 증명을 펌웨어의 원래 작성자에게 추적할 수 있습니다.
L01.FR.10		펌웨어 업데이트를 위해 전송된 모든 FirmwareStatusNotificationRequest에는 이 펌웨어 업데이트를 시작한 UpdateFirmwareRequest와 동일한 requestId가 포함되어야 합니다.	

ID	전제 조건	요구 사항 정의	참고
L01.FR.11		보안상의 이유로 CSMS는 펌웨어 서명 인증서(Keys used in OCPP 참조)를 UpdateFirmwareRequest에 포함해야 합니다.	
L01.FR.12		인증서를 검증하려면(Certificate Hierarchy 참조) X.509 인증서 규칙 [19]을 사용해야 합니다. 충전기는 펌웨어 서명 인증서를 사용하여 파일의 디지털 서명을 검증해야 합니다.	
L01.FR.13	충전 중이거나 *retrieveDateTime* 이 미래이기 때문에 충전기가 펌웨어 다운로드를 시작하지 않는 경우	충전기는 FirmwareStatusNotificationRequest를 DownloadScheduled 상태로 전송해야 합니다.	
L01.FR.14	충전기가 다운로드 일시 중지 상태로 전환될 때.	충전기는 FirmwareStatusNotificationRequest를 상태 DownloadPaused로 전송해야 합니다.	예를 들어, 충전기에 우선순위가 더 높은 작업이 있는 경우.
L01.FR.15	다운로드한 펌웨어를 설치하기 전에 충전기을 재부팅해야 하는 경우.	충전기는 재부팅하기 전에 FirmwareStatusNotificationRequest를 상태 InstallRebooting로 전송해야 합니다.	
L01.FR.16	L01.FR.04 AND *installDateTime* 이 미래의 시간으로 설정된 경우	충전기는 FirmwareStatusNotificationRequest를 상태 InstallScheduled로 전송하고 지정된 설치 시간에 펌웨어를 설치해야 합니다.	
L01.FR.20		FirmwareStatusNotificationRequest의 필드 *requestId* 는 *status* = Idle이 아닌 한 필수입니다.	
L01.FR.21	충전기가 UpdateFirmwareRequest를 수신하는 경우	충전기는 메시지를 수락하기 전에 인증서를 검증해야 합니다.	
L01.FR.22	L01.FR.21 AND 인증서가 유효하지 않습니다.	충전기는 UpdateFirmwareResponse로 응답해야 하며 상태는 `InvalidCertificate` 입니다.	
L01.FR.23	펌웨어 업데이트 중에 충전기가 재부팅해야 하고 부트로더가 OCPP 메시지를 보낼 수 없는 경우	충전기는 FirmwareStatusNotificationRequest 메시지를 생략할 수 있으며 상태는 Installing입니다.	
L01.FR.24	충전기에서 새 펌웨어를 설치 중이거나 새 펌웨어를 설치하려고 하지만 나중에 설치하라는 UpdateFirmware 명령을 받았고 충전기에서 새 UpdateFirmwareRequest를 수신한 경우	충전기는 진행 중인 펌웨어 업데이트를 취소하고 `AcceptedCanceled` 상태로 응답해야 합니다.	충전기는 먼저 새 펌웨어 파일이 있는지 확인하지 않아야 합니다. 이렇게 하면 CSMS가 새 펌웨어 업데이트를 시작하지 않고도 진행 중인 펌웨어 업데이트를 취소할 수 있습니다. 충전기는 현재 취소된 펌웨어 업데이트에 대해 *status* `DownloadFailed` 또는 `InstallationFailed` 로 FirmwareStatusNotificationRequest를 보낼 수 있습니다.

ID	전제 조건	요구 사항 정의	참고
L01.FR.25	충전기는 `FirmwareStatusNotification` 에 대한 TriggerMessageRequest를 수신하고 마지막으로 보낸 FirmwareStatusNotificationRequest는 *status* = Installed을 가졌습니다.	충전기는 *status* = Idle인 FirmwareStatusNotificationRequest를 반환해야 합니다.	
L01.FR.26	충전기가 `FirmwareStatusNotification` 에 대한 TriggerMessageRequest를 수신합니다. AND 마지막으로 전송된 FirmwareStatusNotificationRequest는 *status* = Installed가 아니었습니다.	충전기는 마지막으로 전송된 *status* 와 함께 FirmwareStatusNotificationRequest를 반환해야 합니다.	
L01.FR.27	L01.FR.24 AND 충전기가 펌웨어 설치를 취소할 수 없습니다.	충전기는 *status* = `Rejected` 로 응답할 수 있습니다.	
L01.FR.28	충전기가 새 펌웨어를 성공적으로 설치한 경우	충전기는 FirmwareStatusNotificationRequest를 상태 Installed 로 보내야 합니다. AND 충전기는 이미 새 펌웨어를 활성화했거나 즉시 활성화해야 합니다.	새 펌웨어를 활성화하려면 자동 재부팅이 필요할 수 있지만 반드시 그럴 필요는 없습니다.
L01.FR.29	새 펌웨어 검증(예: 체크섬 또는 다른 수단 사용)이 실패한 경우	충전기는 FirmwareStatusNotificationRequest를 상태 `InstallVerificationFailed` 로 보내야 합니다.	
L01.FR.30	충전기가 실패한 경우 펌웨어 다운로드를 위한 모든 재시도.	충전기는 FirmwareStatusNotificationRequest를 상태 DownloadFailed로 보내야 합니다.	충전기는 재시도할 때마다 새로운 Downloading 상태를 보낼 수 있습니다.
L01.FR.31	L01.FR.28	충전기는 SecurityEventNotificationRequest 메시지를 *type* = "FirmwareUpdated"로 보내야 합니다.	
L01.FR.32	충전기가 새 펌웨어를 성공적으로 설치하고 새 펌웨어를 활성화하기 전에 충전기를 재부팅해야 하는 경우	충전기는 다음 중 하나를 수행해야 합니다. (a) 재부팅하기 전에 *status* = `InstallRebooting` 인 선택적 FirmwareStatusNotificationRequest를 보내고 새로 활성화된 펌웨어에서 *status* = `Installed` 인 필수 FirmwareStatusNotificationRequest를 보내거나 (b) 재부팅 및 새 펌웨어 활성화를 보고하지 않고 상태가 `Installed` 로 설정된 FirmwareStatusNotificationRequest만 보냅니다.	옵션(a)가 선호됩니다. 이는 CSMS에 충전기의 다가올 재부팅을 알리고, 최종 *status* = `Installed` 이 새 펌웨어 이미지에서 전송되므로 CSMS가 새 펌웨어가 활성화되어 있는지 확인할 수 있기 때문입니다. 새 펌웨어 재부팅이 실패할 경우 옵션(b)에서는 이를 보장하지 않습니다.
L01.FR.33	L01.FR.05 AND 충전기에 진행 중인 트랜잭션이 있음 AND 설치된 펌웨어를 활성화하려면 재부팅이 필요함	충전기는 설치된 펌웨어를 활성화하기 전에 모든 트랜잭션이 끝날 때까지 기다려야 합니다.	예: A/B 펌웨어 업데이트의 경우
L01.FR.34	L01.FR.04 AND *installDateTime* 이 설정되지 않음 AND 충전기가 트랜잭션이 완료되기를 기다리고 있음	충전기는 FirmwareStatusNotificationRequest를 상태 InstallScheduled로 보낼 수 있음.	*installDateTime* 이 설정된 경우는 L01.FR.16에 해당합니다.

L02 - Non-Secure Firmware Update

Table 210. L02 - 비보안 펌웨어 업데이트

번호	유형	설명
1	**이름**	비보안 펌웨어 업데이트
2	**ID**	L02
	기능 블록	L. 펌웨어 관리
3	**목표**	비보안 펌웨어 업데이트를 다운로드하고 설치합니다.
4	**설명**	충전기가 비보안 펌웨어 업데이트를 처리하는 방법을 설명합니다.
	Actors	CSMS, 충전기
	시나리오 설명	**1.** CSMS는 펌웨어 위치, 펌웨어를 검색해야 하는 시간, 충전기가 펌웨어 다운로드를 재시도해야 하는 횟수에 대한 정보가 포함된 UpdateFirmwareRequest 메시지를 보냅니다. **2.** 충전기는 UpdateFirmwareResponse로 응답합니다. **3.** 충전기는 FirmwareStatusNotificationRequest를 *Downloading* 상태로 보냅니다. **4.** CSMS는 FirmwareStatusNotificationResponse로 응답합니다. **5.** 충전기는 FirmwareStatusNotificationRequest를 상태 *Downloaded* 로 전송합니다. **6.** CSMS는 FirmwareStatusNotificationResponse로 응답합니다. **7.** 충전기는 FirmwareStatusNotificationRequest를 상태 *Installing* 으로 전송합니다. **8.** CSMS는 FirmwareStatusNotificationResponse로 응답합니다. **9.** 충전기는 FirmwareStatusNotificationRequest를 상태 *Installed* 로 전송합니다. **10.** CSMS는 FirmwareStatusNotificationResponse로 응답합니다.
	대체 시나리오	L01 - Secure Firmware Update
5	**필수 조건**	충전기 제조업체에서 펌웨어 업데이트를 제공했습니다.
6	**사후 조건**	**성공 사후 조건:** 펌웨어 업데이트가 성공적으로 설치되었습니다. **실패 사후 조건:** 펌웨어 업데이트가 실패했습니다.

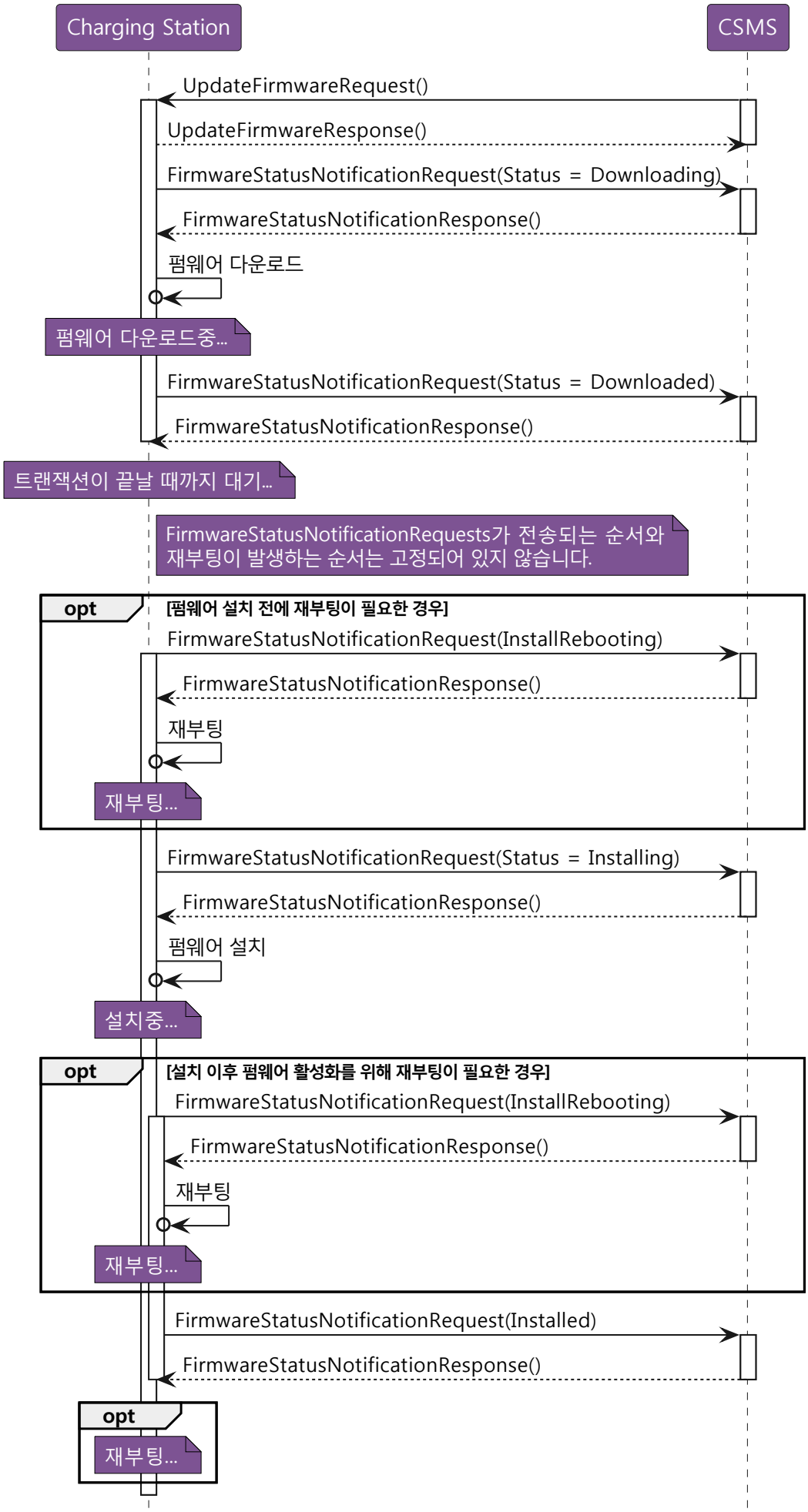

Figure 119. 시퀀스 다이어그램 비보안 펌웨어 업그레이드

7	**오류 처리**	n/a
8	**참고**	서버나 워크스테이션에 저장된 펌웨어를 보호하기 위한 조치를 취해야 합니다. 새로운 버전의 OCPP로 마이그레이션할 때는 새로운 구성으로 대체 NetworkConnectionProfile을 설치하는 것이 좋습니다. FTP는 가능한 다양한 네트워크 토폴로지를 통과하기 위해 Passive FTP를 사용할 수 있어야 합니다.

L02 - Non-Secure Firmware Update - Requirements

Table 211. L02 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
L02.FR.01	충전기가 펌웨어 업데이트 프로세스에서 새 상태로 전환될 때마다.	충전기는 이 새 상태와 함께 FirmwareStatusNotificationRequest 메시지를 CSMS로 보내야 합니다.	
L02.FR.02	충전기가 새 펌웨어를 성공적으로 다운로드하고 (*installDateTime_이 설정되지 않았거나 현재 시간 >= _installDateTime*)	충전기는 가능한 한 빨리 새 펌웨어를 설치해야 합니다.	
L02.FR.03	L02.FR.02 AND 충전기에 진행 중인 트랜잭션이 있는 경우 AND 트랜잭션이 진행되는 동안 펌웨어 설치를 시작할 수 없는 경우	충전기는 설치를 시작하기 전에 모든 트랜잭션이 끝날 때까지 기다려야 합니다.	
L02.FR.04	L02.FR.03 또는 L02.FR.22 AND 구성 변수 `AllowNewSessionsPendingFirmwareUpdate`가 *false* 이거나 존재하지 않습니다.	충전기는 진행 중인 트랜잭션이 끝날 때까지 사용하지 않는 모든 EVSE를 UNAVAILABLE로 설정해야 합니다. 펌웨어가 설치될 때까지 사용 가능해지는 모든 EVSE는 UNAVAILABLE로 설정해야 합니다.	
L02.FR.05		펌웨어는 충전기로 암호화하여 보내는 것이 좋습니다. 이는 보안 프로토콜(HTTPS, SFTP 또는 FTPS 등)을 사용하여 펌웨어를 보내거나 펌웨어 자체를 암호화한 다음 보내는 방식으로 수행할 수 있습니다.	
L02.FR.06		펌웨어 업데이트를 위해 보낸 모든 FirmwareStatusNotificationRequest에는 이 펌웨어 업데이트를 시작한 UpdateFirmwareRequest와 동일한 requestId가 포함되어야 합니다.	
L02.FR.07	충전 중이거나 *retrieveDateTime* 이 미래이기 때문에 충전기가 펌웨어 다운로드를 시작하지 않는 경우	충전기는 FirmwareStatusNotificationRequest를 상태 DownloadScheduled로 전송해야 합니다.	
L02.FR.08	충전기가 다운로드 일시 중지 상태로 전환되는 경우.	충전기는 FirmwareStatusNotificationRequest를 상태 DownloadPaused로 전송해야 합니다.	예를 들어, 충전기에 우선순위가 높은 작업이 있는 경우.

ID	전제 조건	요구 사항 정의	참고
L02.FR.09	다운로드한 펌웨어를 설치하기 전에 충전기을 재부팅해야 하는 경우.	충전기는 재부팅하기 전에 FirmwareStatusNotificationRequest를 상태 InstallRebooting로 전송해야 합니다.	
L02.FR.10	충전기가 새 펌웨어를 성공적으로 다운로드하고 *installDateTime* 이 미래의 시간으로 설정된 경우	충전기는 FirmwareStatusNotificationRequest를 상태 InstallScheduled로 전송하고 지정된 설치 시간에 펌웨어를 설치해야 합니다.	
L02.FR.14		FirmwareStatusNotificationRequest의 필드 *requestId_는 _status* = `Idle` 이 아닌 한 필수입니다.	
L02.FR.15	충전기에서 새 펌웨어를 설치 중이거나 새 펌웨어를 설치하려고 하지만 나중에 설치하라는 UpdateFirmware 명령을 받음 AND 충전기에서 새 UpdateFirmwareRequest를 수신하는 경우	충전기는 진행 중인 펌웨어 업데이트를 취소해야 하며 *AcceptedCanceled* 상태로 응답해야 합니다.	충전기는 새 펌웨어 파일이 있는지 먼저 확인하지 않아야 합니다. 이렇게 하면 CSMS가 새 펌웨어 업데이트를 시작하지 않고도 진행 중인 펌웨어 업데이트를 취소할 수 있습니다.
L02.FR.16	충전기는 `FirmwareStatusNotification` 에 대한 TriggerMessageRequest를 수신하고 마지막으로 보낸 FirmwareStatusNotificationRequest는 *status* = `Installed` 입니다	충전기는 *status* = `Idle` 인 FirmwareStatusNotificationRequest를 반환해야 합니다.	
L02.FR.17	충전기가 `FirmwareStatusNotification` 에 대한 TriggerMessageRequest를 수신하고 마지막으로 전송된 FirmwareStatusNotificationRequest는 *status* = `Installed` 가 아니었습니다.	충전기는 마지막으로 전송된 *status* 와 함께 FirmwareStatusNotificationRequest를 반환해야 합니다.	
L02.FR.18	L02.FR.15 AND 충전기가 펌웨어 설치를 취소할 수 없습니다.	충전기는 *status* = `Rejected` 로 응답할 수 있습니다.	
L02.FR.19	충전기가 실패하면 펌웨어 다운로드를 위한 모든 재시도가 실패합니다.	충전기는 FirmwareStatusNotificationRequest를 상태 DownloadFailed로 전송해야 합니다.	충전기는 재시도할 때마다 새로운 Downloading 상태를 전송할 수 있습니다.
L02.FR.20	충전기가 새 펌웨어를 성공적으로 설치하고 활성화한 경우	충전기는 FirmwareStatusNotificationRequest를 상태 Installed로 전송해야 합니다.	새 펌웨어 활성화에는 재부팅이 필요할 수 있습니다.
L02.FR.21	충전기가 새 펌웨어를 성공적으로 설치하고 새 펌웨어를 활성화하기 전에 충전기를 재부팅해야 하는 경우	충전기는 다음 중 하나를 수행해야 합니다. (a) 재부팅하기 전에 *status* = `InstallRebooting` 인 선택적 FirmwareStatusNotificationRequest를 보내고 새로 활성화된 펌웨어에서 *status* = `Installed` 인 필수 FirmwareStatusNotificationRequest를 보내거나, (b) 재부팅 및 새 펌웨어 활성화를 보고하지 않고 상태가 `Installed` 로 설정된 FirmwareStatusNotificationRequest만 보냅니다.	옵션(a)가 선호됩니다. 이는 CSMS에 충전기의 다가올 재부팅을 알리고, 최종 *status* = `Installed` 가 새 펌웨어 이미지에서 전송되므로 CSMS가 새 펌웨어가 활성화되어 있는지 확인할 수 있기 때문입니다. 새 펌웨어 재부팅이 실패할 경우 옵션(b)에서는 이를 보장하지 않습니다.

ID	전제 조건	요구 사항 정의	참고
L02.FR.22	L02.FR.02 AND 충전기에 진행 중인 트랜잭션이 있음 AND 설치된 펌웨어를 활성화하려면 재부팅이 필요함	충전기는 설치된 펌웨어를 활성화하기 전에 모든 트랜잭션이 끝날 때까지 기다려야 합니다.	예: A/B 펌웨어 업데이트의 경우
L02.FR.23	충전기가 펌웨어를 성공적으로 다운로드하고 *installDateTime* 이 설정되지 않고 충전기가 트랜잭션이 완료되기를 기다리는 경우	충전기는 FirmwareStatusNotificationRequest를 상태 InstallScheduled로 보낼 수 있습니다.	installDateTime이 설정된 경우는 L02.FR.10에 포함됩니다.

L03 - Publish Firmware file on Local Controller

Table 212. L03 - 로컬 컨트롤러에 펌웨어 파일 게시

번호	유형	설명
1	**이름**	로컬 컨트롤러에 펌웨어 파일 게시.
2	**ID**	L03
	기능 블록	L. 펌웨어 관리
3	**목표**	충전기가 로컬 컨트롤러에서 직접 펌웨어 업데이트를 다운로드할 수 있도록 합니다.
4	**설명**	로컬 컨트롤러가 지정된 URL에서 펌웨어 업데이트를 다운로드하고 게시합니다. 이를 통해 CSMS는 로컬 컨트롤러를 가리키는 URI와 함께 UpdateFirmwareRequests를 로컬 컨트롤러에 연결된 모든 충전기로 보낼 수 있습니다. 이를 통해 사이트에서 WAN 인터페이스의 대역폭과 데이터를 절약할 수 있습니다.
	Actors	로컬 컨트롤러, CSMS
	시나리오 설명	**1.** CSMS가 PublishFirmwareRequest를 보내 로컬 컨트롤러에 펌웨어 파일의 MD5 체크섬을 포함하여 펌웨어를 다운로드하고 게시하도록 지시합니다. **2.** PublishFirmwareRequest를 수신하면 로컬 컨트롤러가 PublishFirmwareResponse로 응답합니다. **3.** 로컬 컨트롤러가 펌웨어 다운로드를 시작합니다. **4.** 로컬 컨트롤러가 MD5 체크섬을 확인합니다. **5.** 로컬 컨트롤러가 PublishFirmwareStatusNotificationRequest에 명시된 URI에서 펌웨어 파일을 게시합니다. **6.** CSMS는 사용 사례 L01 - Secure Firmware Update에 설명된 대로 충전기에 펌웨어를 업데이트하도록 지시합니다.
5	**필수 조건**	n/a
6	**사후 조건**	**성공 사후 조건:** 펌웨어가 로컬 컨트롤러에 의해 성공적으로 게시되었습니다. **실패 사후 조건:** 로컬 컨트롤러가 펌웨어 파일을 다운로드할 수 없었고 *DownloadFailed* 상태를 보냈습니다. 로컬 컨트롤러가 MD5 체크섬을 확인할 수 없었고 *InvalidChecksum* 상태를 보냈습니다. 로컬 컨트롤러가 펌웨어 파일을 게시할 수 없었고 *PublishFailed* 상태를 보냈습니다.

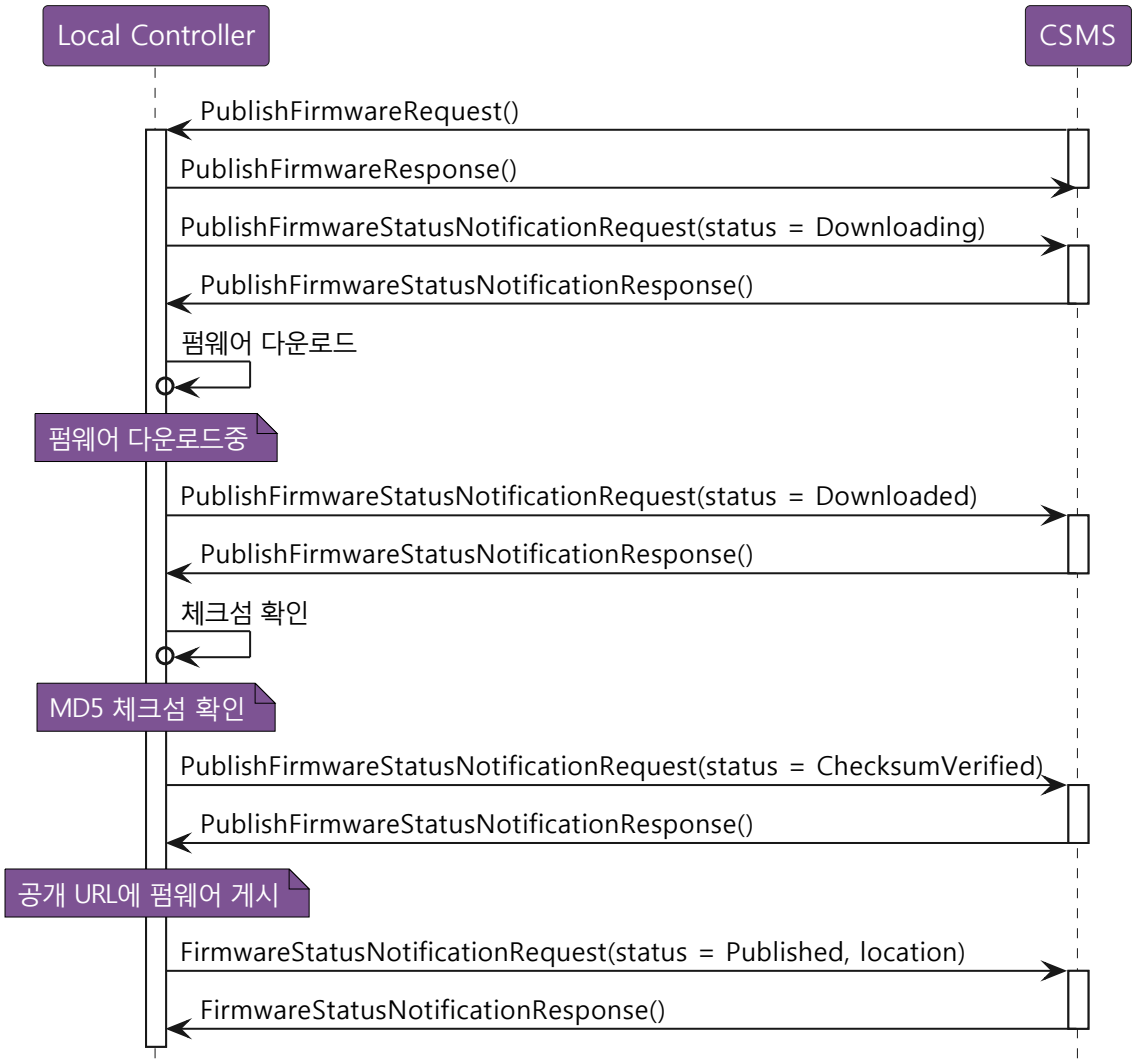

Figure 120. 시퀀스 다이어그램: 펌웨어 게시(행복한 흐름)

7	**오류 처리**	n/a
8	**참고**	MD5 체크섬에 대한 정보는 RFC-1321 [RFC1321]을 참조하세요.

L03 - Publish Firmware file on Local Controller - Requirements

Table 213. L03 - 요구 사항

ID	전제 조건	요구 사항 정의
L03.FR.01		로컬 컨트롤러가 게시 프로세스에서 새 상태를 입력할 때마다 CSMS에 PublishFirmwareStatusNotificationRequest 메시지를 보내야 합니다.
L03.FR.02		MD5 체크섬은 전체 펌웨어 파일에 대해 계산해야 합니다.
L03.FR.03		로컬 컨트롤러는 지원되는 모든 프로토콜(예: HTTP, HTTPS 및 FTP)을 사용하여 펌웨어 파일을 게시해야 합니다.
L03.FR.04		로컬 컨트롤러는 지원되는 모든 프로토콜(예: HTTP, HTTPS 및 FTP)에 대한 URI를 PublishFirmwareStatusNotificationRequest 메시지의 *location* 필드에 *Published* 상태로 설정해야 합니다.

ID	전제 조건	요구 사항 정의
L03.FR.05	PublishFirmwareRequest 메시지를 수신하는 경우	로컬 컨트롤러는 PublishFirmwareResponse 메시지로 응답하여 요청을 수락했는지 여부를 표시해야 합니다.
L03.FR.06	로컬 컨트롤러가 펌웨어 파일을 다운로드할 수 없는 경우.	로컬 컨트롤러는 PublishFirmwareStatusNotificationRequest를 상태 *DownloadFailed* 로 전송해야 합니다.
L03.FR.07	로컬 컨트롤러가 MD5 체크섬을 확인할 수 없는 경우.	로컬 컨트롤러는 PublishFirmwareStatusNotificationRequest를 상태 *InvalidChecksum* 로 전송해야 합니다.
L03.FR.08	로컬 컨트롤러가 펌웨어 파일을 게시할 수 없는 경우.	로컬 컨트롤러는 PublishFirmwareStatusNotificationRequest를 상태 *PublishFailed* 로 전송해야 합니다.
L03.FR.09	펌웨어 파일을 성공적으로 게시한 후.	로컬 컨트롤러는 PublishFirmwareStatusNotificationRequest를 상태 *Published* 로 전송해야 합니다.
L03.FR.10	충전기는 `PublishFirmwareStatusNotification` 에 대한 TriggerMessageRequest를 수신합니다. AND 마지막으로 보낸 PublishFirmwareStatusNotificationRequest는 *status* = `Published` 였습니다.	충전기는 *status* = `Idle` 인 PublishFirmwareStatusNotificationRequest를 반환해야 합니다.
L03.FR.11	충전기는 `PublishFirmwareStatusNotification` 에 대한 TriggerMessageRequest를 수신합니다. AND 마지막으로 전송된 PublishFirmwareStatusNotificationRequest는 *status* `Published` 가 아닙니다.	충전기는 마지막으로 전송된 *status* 와 함께 PublishFirmwareStatusNotificationRequest를 반환해야 합니다.

L04 - Unpublish Firmware file on Local Controller

Table 214. L04 - 로컬 컨트롤러에서 펌웨어 파일 게시 취소

번호	유형	설명
1	**이름**	로컬 컨트롤러에서 펌웨어 파일 게시 취소
2	**ID**	L04
	기능 블록	L. 펌웨어 관리
3	**목표**	로컬 컨트롤러가 충전기에 펌웨어 업데이트를 게시하는 것을 중지합니다.
4	**설명**	연결된 충전기에 펌웨어 업데이트를 제공하는 것을 중지합니다.
	Actors	로컬 컨트롤러, CSMS
	시나리오 설명	**1.** CSMS가 UnpublishFirmwareRequest를 보내 로컬 컨트롤러에 펌웨어를 게시 취소하도록 지시합니다. **2.** 로컬 컨트롤러가 펌웨어를 게시 취소합니다. **3.** 로컬 컨트롤러가 UnpublishFirmwareResponse로 응답합니다.

번호	유형	설명
5	필수 조건	로컬 컨트롤러가 성공적으로 게시한 펌웨어.
6	사후 조건	**성공 사후 조건:** 펌웨어 파일이 더 이상 게시되지 않습니다. **실패 사후 조건:** n/a

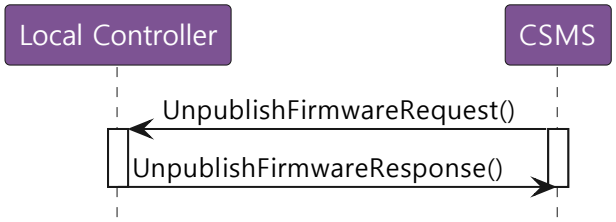

Figure 121. 시퀀스 다이어그램: 펌웨어 파일 게시 취소

7	오류 처리	n/a
8	참고	CSMS는 전체 펌웨어 파일에 대한 MD5 체크섬을 고유 식별자로 사용하여 어떤 펌웨어 파일을 게시 취소해야 하는지 나타냅니다.

L04 - Unpublish Firmware file on Local Controller - Requirements

Table 215. L04 - 요구 사항

ID	전제 조건	요구 사항 정의
L04.FR.01	로컬 컨트롤러가 UnpublishFirmwareRequest 메시지를 수신하고 진행 중인 다운로드가 없는 경우.	펌웨어 파일은 게시 취소되어야 합니다.
L04.FR.02	펌웨어 파일을 성공적으로 게시 취소한 후.	로컬 컨트롤러는 UnpublishFirmwareResponse 메시지를 *Unpublished* 상태로 전송해야 합니다.
L04.FR.03	로컬 컨트롤러가 UnpublishFirmwareRequest 메시지를 수신하고 게시된 파일이 없는 경우.	로컬 컨트롤러는 UnpublishFirmwareResponse 메시지를 *NoFirmware* 상태로 보내야 합니다.
L04.FR.04	로컬 컨트롤러가 UnpublishFirmwareRequest 메시지를 수신하고 충전기가 펌웨어 파일을 다운로드하는 경우.	로컬 컨트롤러는 *Downloading* 상태로 응답하고 펌웨어 파일을 게시 취소하지 **않아야** 합니다.

M. ISO 15118 CertificateManagement

1. Introduction

ISO/IEC JWG 15118 for the Vehicle to Grid Communication Interface(V2G CI)는 2009년에 설립되었으며, 인터넷 프로토콜을 기반으로 양방향 디지털 통신을 제공하는 IEC 61851-1 [IEC61851-1]에 대한 보완적인 국제 표준이 필요하다는 필요성을 충족하기 위해 만들어졌습니다. 15118의 주요 목적은 EV와 충전 인프라 간에 보다 진보적이고 자율적으로 작동하는 충전 제어 메커니즘을 구축하는 것입니다. 이 표준은 현재 개발 중이며 궁극적으로 다양한 인증 방식(예: 플러그 충전 대 RFID 카드와 같은 외부 식별 수단), 충전 서비스의 자동 처리 및 (독점적인) 부가가치 서비스, 충전 일정 및 사전 계획 등을 위한 수단을 제공할 것입니다.

15118 표준은 충전 일정을 교환하고 EV가 충전기에서 소모할 수 있는 전력량을 제어할 수 있기 때문에 Open Charge Alliance의 관심을 끌고 있으며, 여기에는 차량과 그리드 간 통신의 어떤 형태가 필요합니다. 특히, TCP/IP 기반 전송 및 네트워크 계층을 통해 통신 파트너(애플리케이션 계층), 연관된 데이터 및 데이터 유형(프레젠테이션 계층) 간에 교환할 메시지를 지정하는 두 번째 부분은 이 사양에서 인식하는 것이 중요합니다. 충전에 대한 승인은 RFID 카드와 같은 외부 식별 수단(EIM) 또는 EV에 저장된 계약 인증서를 사용하는 플러그 앤 충전(PnC) 메커니즘을 통해 제공되며, 사용 사례 요소 "C"의 인증서 처리 프로세스에서 처리되므로 다른 승인 수단이 필요하지 않습니다.

이 15118 OCPP 기능 블록은 여러 정렬 목표를 충족하도록 설계되었습니다.

- EV(BEV 또는 PHEV)와 EVSE 간의 통신을 허용합니다.
- 충전기에서 인증서 기반 인증 및 승인(즉, 플러그 앤 충전)을 지원합니다.

설명을 위해 아래 그림은 승인 및 스케줄링이 포함된 전체 시퀀스를 보여줍니다.

NOTE 아래 그림에: 이 시퀀스는 AC 충전에만 적용되지만, 인증서 처리(이 섹션에서 중점적으로 다루는 부분)는 AC나 DC에서 다르지 않습니다.

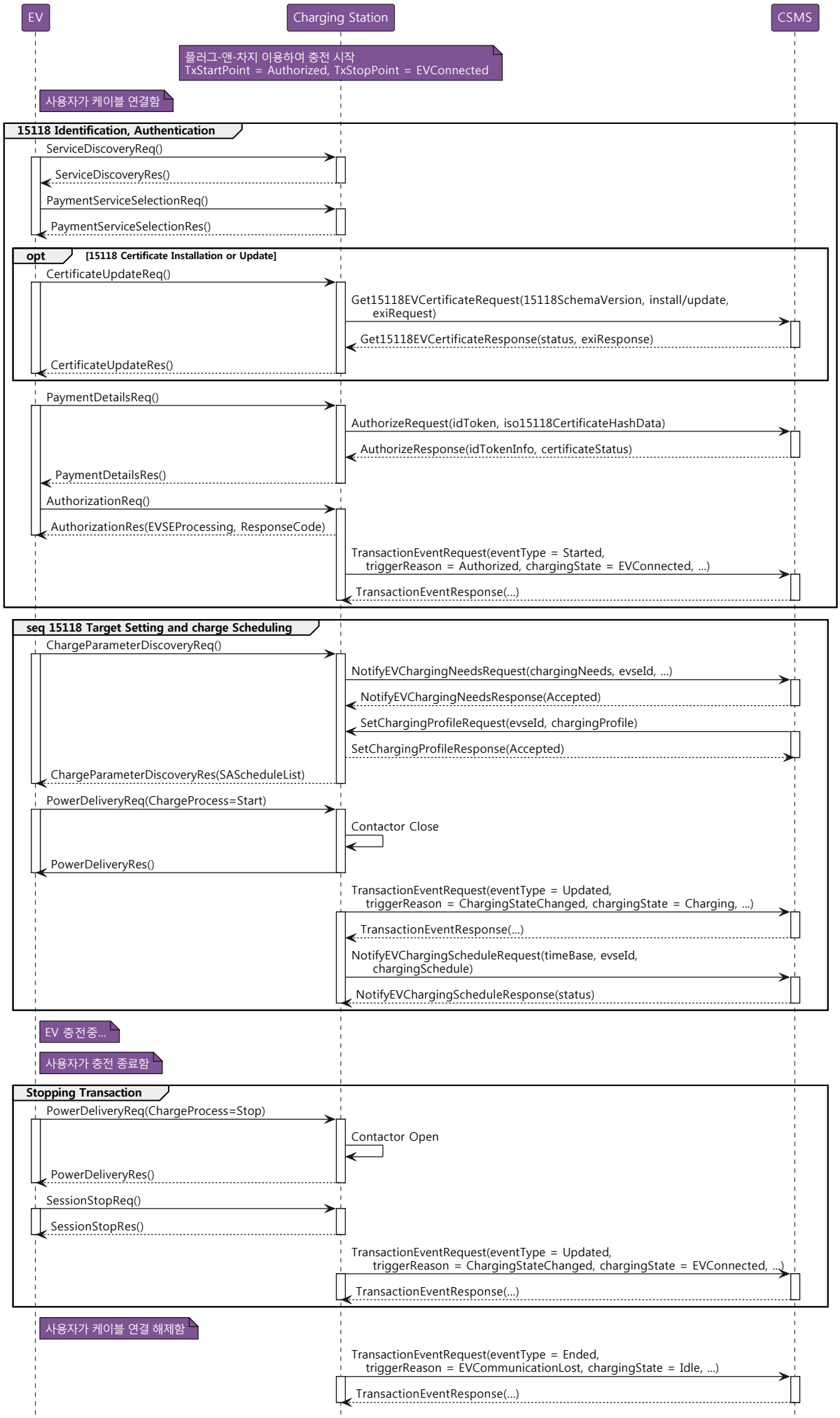

Figure 122. 인증 및 스케줄링이 있는 시퀀스

NOTE	ChargeParameterDiscoveryReq의 시간 초과는 2초이지만, CSMS에서 제공하는 충전 프로필을 기다리기 위해 최대 60초까지 연장할 수 있습니다. ISO 15118-2 [ISO15118-2]를 참조하세요.

NOTE	기능 블록A의 TLS 기반 보안 프로필 중 하나를 사용하는 것이 좋습니다. 이렇게 하지 않으면 ISO 15118 보안이 "파괴"될 수 있습니다.

EV가 충전기에서 소모할 수 있는 전력량을 제어하려면 어떤 형태의 차량-그리드 통신이 필요합니다. OCPP는 EV와 충전기(EVSE) 간 통신을 위한 ISO 15118 표준을 지원하도록 설계되었습니다. 그러나 앞으로 몇 년 동안 대부분의 EV는 제어 파일럿 PWM 신호 IEC61851만 지원할 것으로 예상되므로 이를 통해 스마트 충전을 지원하는 것 또한 주의를 기울였습니다.

NOTE	ISO 15118 및 OCPP 용어 매핑은 ISO 15118 and OCPP terminology mapping에 제공되고 ISO 15118에서 사용되는 약어는 ISO 15118 Abbreviations에 나열되어 있습니다.

2. ISO 15118 Certificates

2.1. ISO 15118 Certificate structure

ISO 15118 표준은 Plug & Charge 메커니즘을 제공합니다. 이는 고객이 전기 자동차를 EVSE에 연결하기만 하면 모든 인증, 승인, 부하 제어 및 청구 측면이 추가 사용자 상호 작용 없이 자동으로 처리되는 식별 및 승인 모드입니다. 이는 디지털 서명을 적용하고 공개 키 인프라(PKI) 모델에 바인딩된 X.509 인증서를 교환하여 용이하게 합니다.

ISO 15118에서 정의한 PKI 구조는 아래 그림에 나와 있습니다. 일반적으로 4개의 PKI가 있어야 합니다.

- 충전기 운영자(CSO)용 PKI
- 인증서 제공 서비스(CPS)용 PKI
- 모빌리티 운영자(MO)용 PKI
- 자동차 제조업체(OEM)용 PKI

CSO 및 CPS용 트러스트 앵커(루트 CA)는 소위 V2G 루트 CA입니다. 반면, 각 OEM과 MO는 자체 루트 CA를 운영하거나 V2G 루트 CA에서 인증서를 파생해야 합니다(각각 V2G 루트와 MO 하위 CA 1 및 OEM 하위 CA 1 사이의 점선으로 표시).

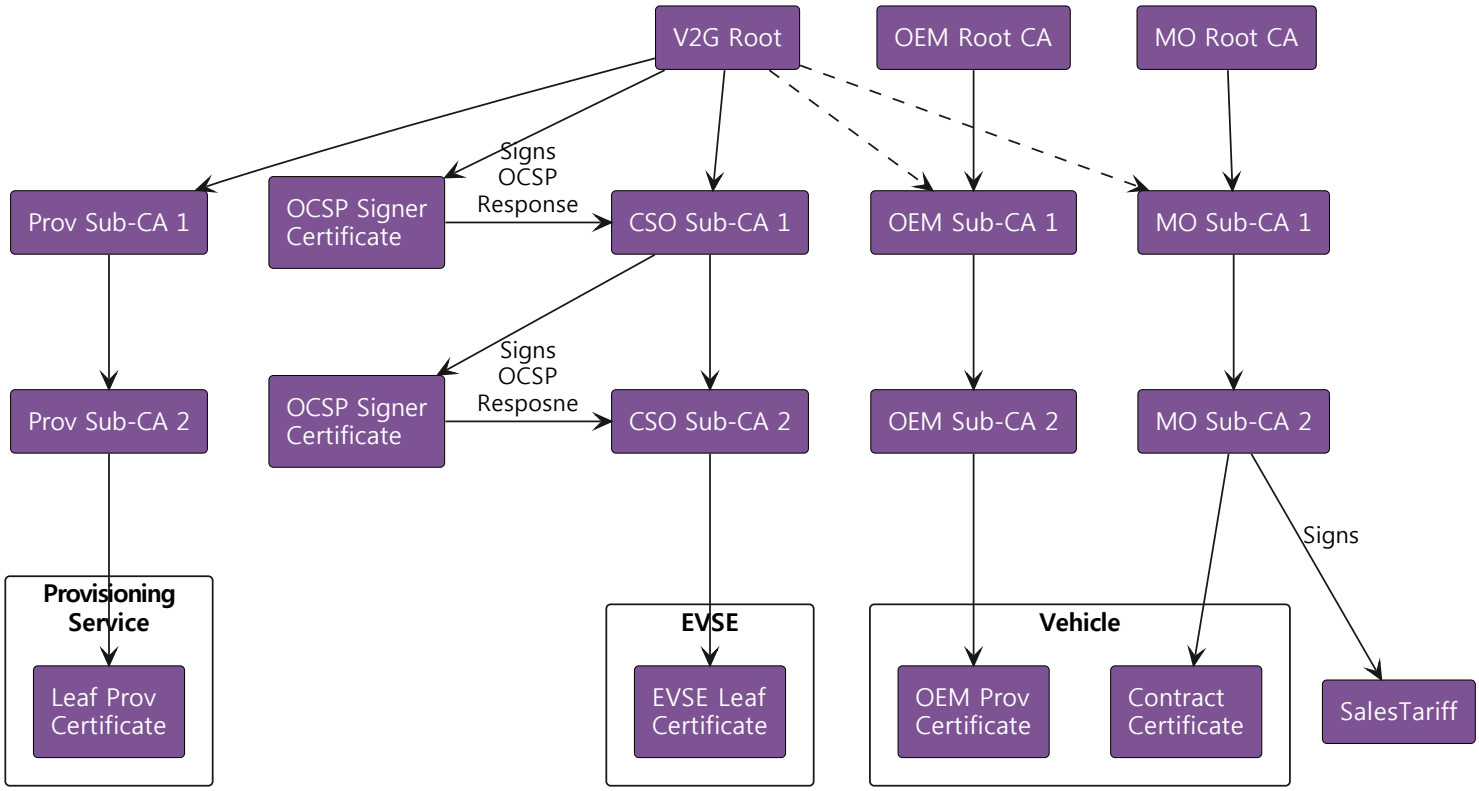

Figure 123. Plug & Charge 식별 모드에 적용된 PKI

하위 CA 계층이 하나만 사용되는 경우, 즉 루트 CA가 서명한 하위 CA가 리프 인증서에 직접 서명하는 경우 하위 CA 2의 프로필이 해당 하위 CA에 적용됩니다(출처: ISO15118-2)

OCPP는 계약 제공을 용이하게 하기 위해 EV, 충전기 및 백엔드 IT 인프라 간에 필요한 정보를 교환할 수 있는지 확인해야 합니다. 계약 프로비저닝(Contract provisioning)은 ISO 15118에 정의된 프로세스로, EV가 통신 세션 중에 유효한 계약 인증서를 검색하여 충전 프로세스에 대한 인증 및 권한을 부여하는 방법을 설명합니다.

위 그림의 PKI 구조를 고려할 때 OCPP는 다음 인증서를 전송할 수 있는 메시지를 제공해야 합니다.

- **CPS 인증서 체인**
  Prov Sub-CA 1, Prov Sub-CA 2 및 리프 프로비저닝 인증서로 구성됨. CertificateInstallationRes 및 CertificateUpdateRes 메시지와 함께 전송됨.

- **MO 인증서 체인**
  MO Sub-CA 1, MO Sub-CA 2 및 계약 인증서로 구성됨. CertificateInstallationRes, CertificateUpdateReq 및 CertificateUpdateRes 메시지와 함께 전송됨.
- **OEM 프로비저닝 인증서**
  CertificateInstallationReq 메시지와 함께 전송됨.

또한 일부 ISO 15118 메시지에는 디지털 XML 기반 서명이 필요합니다. 이러한 서명은 해당 인증서 체인을 사용하여 수신 당사자가 검증하고 해당 신뢰 앵커(V2G 루트, MO 루트 또는 OEM 루트)까지 서명 체인을 검증해야 합니다. ISO15118-2의 45페이지에 있는 표 13은 ISO 15118에서 적용된 XML 기반 서명에 대한 개요를 제공합니다. 거기에서 볼 수 있듯이 충전기(EVSE는 충전기의 일부)는 다음 메시지의 서명을 검증해야 합니다.

- **AuthorizationReq**
  서명을 검증하는 데 필요한 인증서 체인은 PaymentDetailsReq와 함께 제공됩니다.
- **MeteringReceiptReq**
  서명을 검증하는 데 필요한 인증서 체인은 PaymentDetailsReq와 함께 제공됩니다.
- **CertificateUpdateReq**
  서명을 검증하는 데 필요한 인증서 체인은 이 메시지와 함께 제공됩니다.

EV에서 제공하는 각 인증서의 유효성 검사와 서명 검증은 오프라인에서 수행할 수 있습니다. 이 세 가지 메시지는 EV에 설치된 계약 인증서의 공개 키에 속하는 개인 키로 서명됩니다. CSO는 해당 MO 루트 CA 인증서(MO 신뢰 앵커)가 충전기에 설치되어 오프라인에서 서명 검증이 가능하도록 해야 합니다(계약 인증서와 하위 CA 인증서 체인은 EV에서 PaymentDetailsReq 메시지에서 이미 충족되므로 MO 루트 CA만 필요합니다).

PaymentDetailsReq 메시지는 AuthorizationReq 및 MeteringReceiptReq 메시지보다 먼저 전송됩니다. 따라서 충전기는 EV에서 생성한 서명을 검증하기 위해 현재 트랜잭션이 활성화된 동안 PaymentDetailsReq 메시지와 함께 제공된 인증서 체인을 일시적으로 저장해야 합니다. 트랜잭션이 종료된 후에는 충전기 측에서 일시적으로 저장된 인증서 체인을 삭제해야 합니다.
충전기는 계약 인증서와 루트 CA 인증서를 제외한 가능한 하위 CA 인증서가 포함된 ContractSignatureCertChain을 전달하는 EV에서 PaymentDetailsReq 메시지를 수신할 때만 계약 인증서를 확인하면 됩니다. *그러나* EV에_ 전달될 때 계약 인증서를 설치하거나 업데이트할 때는 계약 인증서를 확인할 필요가 없습니다.

반대로 **CertificateInstallationReq**와 함께 제공된 서명은 CSO 백엔드와 통신하는 시장 이해 관계자인 소위 보조 행위자(secondary actor)가 확인해야 합니다. 즉, OCPP는 전체 CertificateInstallationReq 메시지를 전송하는 수단을 제공해야 합니다.

CertificateUpdateRes 및 CertificateInstallationRes는 CSO 백엔드에서 충전기로 Base64 인코딩된 바이너리 데이터로 전송해야 합니다. 충전기는 Base64 인코딩을 제거하고 EV로 바이너리 EXI 메시지로 전송합니다.

마지막으로, 충전기 인증서(그림 1에서 EVSE Leaf 인증서로 표시됨)와 개인 키는 TLS를 통해 EV와 EVSE 간에 보안 연결을 설정하는 데 사용됩니다. ISO 15118에 따르면 이 인증서는 2~3개월 동안만 유효해야 합니다. 충전기 인증서를 설치하거나 업데이트하려면 Certificate installation Charging Station을 참조하세요.

충전기는 MO 계약 인증서 체인의 각 인증서의 서명과 유효 기간을 오프라인에서 확인할 수 있지만, 충전기는 오프라인에서 확인할 수 없는 두 가지 사항이 있습니다.+

1. **EMAID의 승인 상태**
   EMAID는 MO가 계약 인증서와 함께 발급한 고유 식별자입니다. 따라서 MO만이 이 EMAID를 기반으로 사용자가 충전을 승인받았는지 여부에 대한 정보를 제공할 수 있습니다. 충전기는 계약 인증서 체인의 각 인증서의 서명이 유효한지 확인한 후 EMAID를 CSO에 전달해야 합니다. 이러한 단계 순서는 계약 인증서가 발급자의 디지털 서명을 통한 조작으로부터 EMAID를 보호하기 때문에 필요합니다. 충전기는 로컬에 캐시된 EMAID의 허용 목록으로 작업할 수도 있습니다. 그러나 사용되는 승인 정보가 오래되지 않도록 허용 목록을 자주 업데이트해야 합니다.+
2. **각 인증서의 해지 상태**
   인증서를 해지하는 이유는 다음과 같습니다. 인증서의 공개 키에 속하는 개인 키가 손상되었거나 서명을 만드는 데 사용된 알고리즘이 더 이상 안전하지 않은 것으로 간주된다는 것입니다. 해지 상태는 X.509 인증서의 속성 값으로 주소가 제공된 OCSP 응답자를 사용하여 확인됩니다.

2.2. Using ISO 15118 Certificates in OCPP

OCPP 관점에서 위의 문단을 기준으로 충전기는 다음 인증서 유형 중 하나 이상을 가져야 합니다.

유형	설명
V2GChargingStation 인증서	충전기 인증서. 15118에서는 이를 *SECC 인증서*(또는 *EVSE Leaf 인증서*)라고 합니다. 이 인증서는 충전기와 EV 간의 TLS 연결을 설정하는 동안 사용됩니다.
V2GRootCertificate	ISO15118 V2G 루트 인증서. V2G 충전기 인증서는 반드시 이 루트에서 파생되어야 합니다.
MORootCertificate	eMobility 서비스 공급자의 인증서. V2G 루트에서 인증서를 파생하지 않은 서비스 공급자의 계약으로 PnC 충전을 지원합니다.

NOTE	V2G 충전기 인증서는 충전기와 CSMS 간 연결을 보호하는 데 사용되는 인증서와 동일할 수 있습니다. 이를 작동시키려면 이 인증서가 V2G 루트에서 파생되어야 합니다.

계약 인증서는 V2G 루트 또는 eMobility 루트에서 파생될 수 있습니다. 즉, 충전기는 계약 인증서와 관련 인증서 체인을 통해 운전자를 인증할 수 있도록 해당 루트 인증서를 소유해야 합니다.

NOTE	충전기가 온라인인 경우 반드시 그럴 필요는 없습니다. CSMS에서 검증할 계약 인증서와 함께 AuthorizeRequest 메시지를 보낼 수 있기 때문입니다.

V2G 충전기 인증서는 V2G 루트에서 파생되어야 합니다. 이 루트가 EV에서 알려지지 않은 경우 15118을 통한 연결이 불가능하므로 15118에서 제어하는 충전은 불가능합니다. 충전기에서 두 개 이상의 V2G 루트를 지원해야 하는 경우 여러 개의 V2G 충전기 인증서가 필요합니다.

2.3. 15118 communication set-up

15118 통신 세션이 시작되면 EV는 TLS 연결을 시작합니다. 이 요청에서 자동차는 알려진 V2G 루트 인증서를 제시합니다.

TLS 핸드셰이크 중에 EVCC는 IETF RFC 6961에 정의된 OCSP 스테이플링을 사용하여 충전기의 OCSP 상태와 중간 인증서를 요청할 수 있습니다. 충전기는 GetCertificateStatusRequest를 CSMS로 보내 이 정보를 검색할 수 있습니다. 사용 사례 M06 - Get Charging Station Certificate status를 참조하세요.

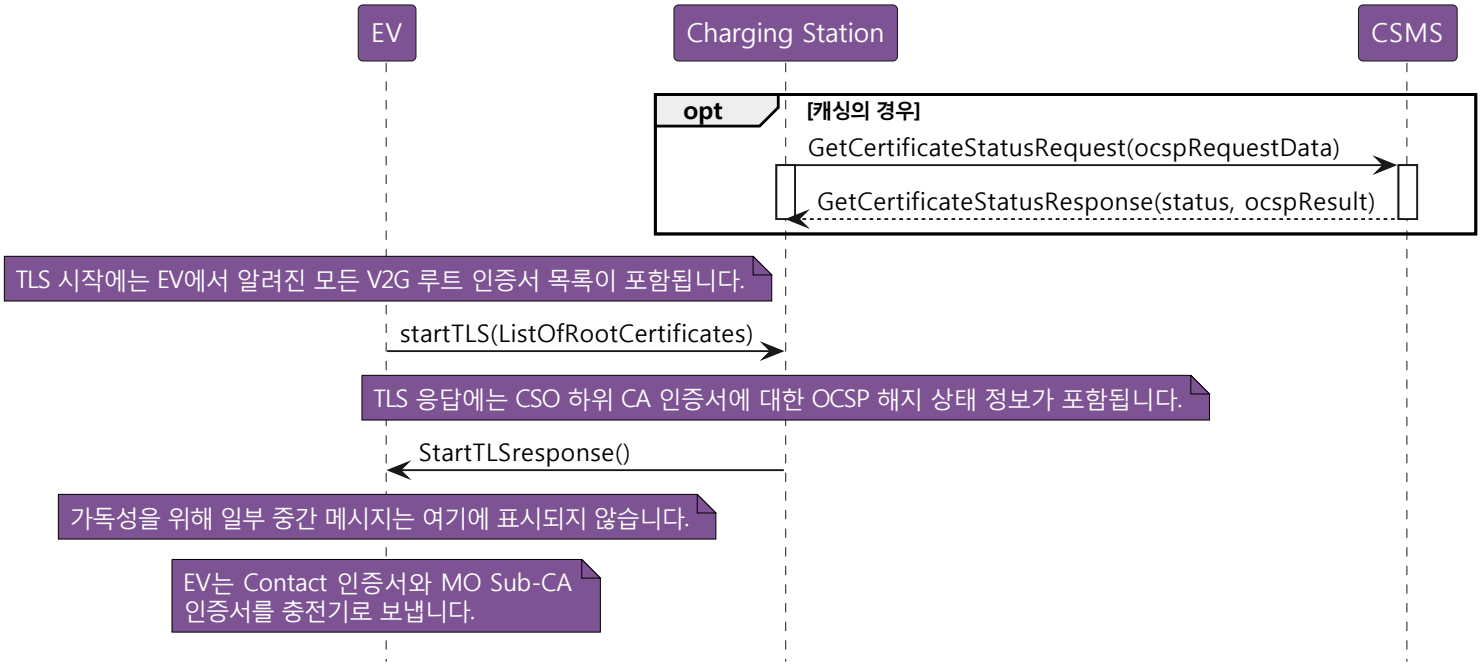

Figure 124. 통신 설정

2.4. Certificate - Use Case mapping

다음 표에는 OCPP에서 ISO 15118 요금 청구에 필요한 인증서를 관리하는 데 사용할 수 있는 사용 사례가 나와 있습니다.

Table 216. 15118에 관련된 인증서

인증서	사용 대상	사용 사례	비고
ChargingStationCertificate	충전기 - CSMS 연결	A02 및 A03	일반적으로 OCPP 보안에 사용됨. 인증서 체인도 사용할 수 있어야 하며 인증서를 설치할 때 충전기에서 검색할 수 있어야 합니다.
CPS 인증서 체인	플러그 앤 충전 인증	M03, M04 및 M05	
EVContractCertificate	플러그 앤 충전 인증	M01 및 M02	단기 인증서(플러그 앤 충전용)
MORootCertificate	플러그 앤 충전 인증	M03, M04 및 M05	
MO Certificate Chain	플러그 앤 충전 인증	N.a.	플러그 앤 충전 인증을 위해 MO 루트 인증서만 설치하면 되고, 다른 중간 인증서는 EV에서 제공합니다.
OEMProvisioningCertificate	EV에 인증서 설치	M01 및 M02	OEM에서 EV에 설치한 장기 인증서
V2GChargingStationCertificate	EV - 충전기 TLS 연결	A02 및 A03	인증서 체인도 사용할 수 있어야 하며 인증서를 설치할 때 충전기에서 검색할 수 있어야 합니다.
V2GRootCertificate	EV - 충전기 TLS 연결	M03, M04 및 M05	플러그 앤 차지 인증을 위해 V2G 루트 인증서만 설치하면 됩니다.
V2GIntermediateCertificate	플러그 앤 차지 인증	A02, A03, M03 및 M04	*V2GChargingStationCertificate* 와 *V2GRootCertificate* 사이의 중간 인증서입니다. EV와 충전기 사이의 TLS 설정 중에 사용할 수 있습니다.

2.5. Use cases from ISO 15118 relevant for OCPP

모든 기본 사용 사례 목록은 ISO15118-1 17페이지를 참조하세요. **굵은 글씨**로 표시된 사용 사례 구성 요소는 ISO15118-1에 따른 OCPP 통신의 영향으로 식별됩니다.

Table 217. 15118 OCPP 관련 사용 사례(출처 원래 표: ISO15118-1)

번호	사용 사례 요소 이름/그룹화
A1	강제 고수준 통신(High Level Communication)을 통한 충전 프로세스 시작
A2	동시 IEC61851-1 및 고수준 통신을 통한 충전 프로세스 시작
B1	**EV/충전기 통신 설정**
C1	**인증서 업데이트**
C2	**인증서 설치**
D1	EVSE에서 수행된 계약 인증서를 사용한 인증
D2	**SA의 도움으로 수행된 계약 인증서를 사용한 인증**
D3	EVSE에서 수행된 외부 자격 증명을 사용한 EVSE 인증
D4	**SA의 도움으로 수행된 외부 자격 증명을 사용한 EVSE 인증**
E1	**고수준 통신을 기반으로 부하 레벨링을 사용한 AC 충전**
E2	**보조 행위자(Secondary Actor)에 대한 스케줄링을 사용한 최적화된 충전**
E3	**EV에서 스케줄링을 사용한 최적화된 충전**
E4	**고수준 통신을 기반으로 부하 레벨링을 사용한 DC 충전**
E5	허가된 충전 일정으로 재개
F0	충전 루프
F1	계량 정보 교환이 있는 충전 루프
F2	**충전기에서 인터럽트가 있는 충전 루프**
F3	**EV 또는 사용자에서 인터럽트가 있는 충전 루프**
F4	무효 전력 보상
F5	**V2G 지원**
G1	부가가치 서비스
G2	충전 세부 정보
H1	**충전 프로세스 종료**

NOTE	이 기능 블록에서 모든 15118 관련 OCPP 사용 사례가 설명되는 것은 아닙니다. 이 기능 블록은 EV 및 CA 인증서 처리에서 인증서 설치 및 업데이트(15118과 관련되지 않은 목적에도 해당)를 설명합니다. 인증 사용 사례는 ISO 15118 Authorization을 참조하세요. 스마트 충전 관련 사용 사례는 Smart Charging 장에서 설명합니다.

3. Use cases & Requirements

M01 - Certificate installation EV

Table 218. M01 - 인증서 설치

번호	유형	설명
1	**이름**	인증서 설치
2	**ID**	M01
	기능 블록	M. ISO 15118 인증서 관리
	참조	ISO15118-1 C2
3	**목표**	EV에 CSMS의 새 인증서를 설치합니다.
4	**설명**	EV가 새 인증서 설치를 시작합니다. 충전기가 새 인증서에 대한 요청을 CSMS로 전달합니다. 추가 정보는 ISO15118-1, 사용 사례 설명 C2, 22페이지를 참조하세요.
	Actors	EV, 충전기, CSMS
	시나리오 설명	**15118** : ISO15118-1, 사용 사례 설명 C2, 시나리오 설명, 처음 3개 항목, 22페이지를 참조하세요. **OCPP** : - 충전기는 Get15118EVCertificateRequest 메시지를 ***action*** = `Install`과 함께 CSMS로 보냅니다. - CSMS는 Get15118EVCertificateResponse로 충전기에 응답합니다.
	대체 시나리오	n/a
5	**필수 조건**	- EV와 EVSE 간의 통신은 성공적으로 설정되어야 합니다. - 충전기와 CSMS 간의 온라인 연결이 가능해야 합니다. - CSMS는 CertificateInstallationRequest를 처리할 수 있는 제3자(예: 계약 인증서 풀)와 통신할 수 있어야 합니다.
6	**사후 조건**	ISO15118-1 참조, 사용 사례 종료 조건 C2, 23페이지.

Figure 125. 인증서 설치

7	**오류 처리**	CSMS가 지정된 시간 내에 응답할 수 없는 경우 충전기는 EV에 실패를 표시해야 합니다.
8	**참고**	ISO15118-2에서 CertificateInstallationReq에 대한 메시지 시간 초과는 5초입니다. 인증서 설치를 위한 대체 통신 경로가 있을 수 있습니다. 그러나 이는 이 표준의 범위를 벗어납니다.

출처: ISO15118-1

M01 - Certificate installation - Requirements

Table 219. M01 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
M01.FR.01	15118 CertificateInstallationReq를 수신하면	충전기는 Get15118EVCertificateRequest 메시지에 ***action*** = **`Install`**를 사용하여 요청을 CSMS에 전달해야 합니다.	CSMS는 CertificateUpdateRequest를 처리할 보조 행위자에게 이를 전달해야 합니다. 이는 애플리케이션 가이드 VDE-AR-2802-100-1에 설명된 계약 인증서 풀일 수 있습니다.

M02 - Certificate Update EV

Table 220. M02 - 인증서 업데이트

번호	유형	설명
1	**이름**	인증서 업데이트
2	**ID**	M02
	기능 블록	M. ISO 15118 인증서 관리
	참조	ISO15118-1 C1
3	**목표**	ISO15118-1, 사용 사례 목표 C1, 20페이지를 참조하세요.
4	**설명**	ISO15118-1, 사용 사례 설명 C1, 21페이지에서 세 번째 "참고"까지 참조하세요.
	Actors	EV, 충전기
	시나리오 설명	**15118**: ISO15118-1, 사용 사례 목표 C1, 시나리오 설명, 처음 3개 항목, 21페이지를 참조하세요. **OCPP**: - 충전기는 Get15118EVCertificateRequest 메시지를 ***action*** = **`Update`**를 사용하여 CSMS로 보냅니다. - CSMS는 Get15118EVCertificateResponse로 충전기에 응답합니다. **15118**: ISO15118-1, 사용 사례 설명 C1, 시나리오 설명, 마지막 2개 항목, 21페이지를 참조하세요.
5	**필수 조건**	- EV와 EVSE 간의 통신은 성공적으로 설정되어야 합니다. - 충전기와 CSMS 간의 온라인 연결이 가능해야 합니다. - CSMS는 CertificateInstallationRequest를 처리할 수 있는 제3자(예: 계약 인증서 풀)와 통신할 수 있어야 합니다.
6	**사후 조건**	ISO15118-1, 사용 사례 목표 C1 및 C2, 20/22페이지를 참조하세요.

Figure 126. 인증서 업데이트

7	**오류 처리**	CSMS가 지정된 시간 내에 응답할 수 없는 경우 충전기는 EV에 실패를 표시해야 합니다.
8	**참고 사항**	ISO15118-1, 사용 사례 요구 사항 C1, 트리거, 21페이지를 참조하세요. ISO15118-2에서 CertificateUpdateReq에 대한 메시지 시간 초과는 5초입니다.

출처: ISO15118-1

M02 - Certificate Update - Requirements

Table 221. M02 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
M02.FR.01		CertificateUpdateReq를 수신하면 충전기는 Get15118EVCertificateRequest 메시지를 사용하여 ***action*** = **`Update`**로 설정된 요청을 CSMS에 전달해야 합니다.	CSMS는 CertificateUpdateRequest를 처리할 보조 행위자에게 이를 전달해야 합니다. 이는 애플리케이션 가이드 VDE-AR-E 2802-100-1에 설명된 대로 계약 인증서 풀일 수 있습니다.

M03 - Retrieve list of available certificates from a Charging Station

Table 222. M03 - 충전기에서 사용 가능한 인증서 목록 검색

번호	유형	설명
1	**이름**	충전기에서 사용 가능한 인증서 목록 검색
2	**ID**	M03
	기능 블록	M. ISO 15118 인증서 관리
3	**목표**	CSMS가 충전기에서 사용 가능한 인증서 목록을 검색할 수 있도록 합니다.
4	**설명**	충전기에 설치된 인증서 관리를 용이하게 하기 위해 설치된 인증서를 검색하는 방법이 제공됩니다. CSMS가 충전기에 설치된 인증서 목록을 보내도록 요청
	Actors	충전기, CSMS

번호	유형	설명
	시나리오 설명	**1.** CSMS가 충전기에 GetInstalledCertificateIdsRequest 를 보내 설치된 인증서 목록을 보내도록 요청합니다. **2.** 충전기가 GetInstalledCertificateIdsResponse로 응답
5	**필수 조건**	해당 없음
6	**사후 조건**	CSMS가 설치된 인증서 목록을 수신

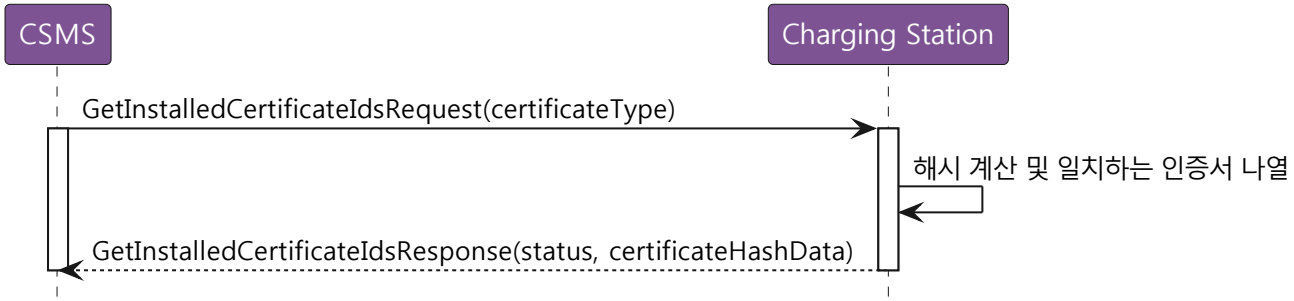

Figure 127. 충전기에서 사용 가능한 인증서 목록 검색

7	**오류 처리**	n/a
8	**참고**	(V2G) 충전기 인증서를 설치하는 경우 사용 사례 A02 - Update Charging Station Certificate by request of CSMS 및 A03 - Update Charging Station Certificate initiated by the Charging Station를 참조하세요. V2G 인증서 체인에는 V2GRootCertificate가 포함되어서는 안 됩니다. 이는 사용 사례 M05 - Install CA certificate in a Charging Station를 사용하여 설치해야 합니다.

M03 - Retrieve list of available certificates from a Charging Station - Requirements

Table 223. M03 - 요구 사항

ID	전제 조건	요구 사항 정의
M03.FR.01	GetInstalledCertificateIdsRequest를 수신한 후	충전기는 GetInstalledCertificateIdsResponse로 응답해야 합니다.
M03.FR.02	M03.FR.01 AND *certificateType* 과 일치하는 인증서를 찾을 수 없습니다.	충전기는 GetInstalledCertificateIdsResponse의 *status* 를 *NotFound* 로 설정하여 이를 표시해야 합니다.
M03.FR.03	M03.FR.01 AND *certificateType* 과 일치하는 인증서가 발견되었습니다.	충전기는 GetInstalledCertificateIdsResponse에서 *status* 를 *Accepted* 로 설정하여 이를 표시해야 합니다.
M03.FR.04	M03.FR.03	충전기는 GetInstalledCertificateIdsResponse에서 일치하는 각 설치된 인증서에 대한 해시 데이터를 포함해야 합니다.
M03.FR.05	충전기가 GetInstalledCertificateIdsRequest와 certificateType V2GCertificateChain을 수신할 때	충전기는 V2G 인증서 체인에 속하는 각 설치된 인증서에 대한 해시 데이터를 포함해야 합니다. 하위 CA 인증서는 V2G 충전기 인증서 아래에 childCertificate로 배치해야 합니다.

M04 - Delete a specific certificate from a Charging Station

Table 224. M04 - 충전기에서 특정 인증서 삭제

번호	유형	설명
1	**이름**	충전기에서 특정 인증서 삭제
2	**ID**	M04
	기능 블록	M. ISO 15118 인증서 관리
3	**목표**	CSMS가 충전기에 설치된 인증서를 삭제하도록 요청할 수 있도록 합니다.
4	**설명**	충전기의 설치된 인증서 관리를 용이하게 하기 위해 설치된 인증서를 삭제하는 방법이 제공됩니다. CSMS는 충전기에 특정 인증서를 삭제하도록 요청합니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS는 DeleteCertificateRequest를 보내서 충전기에 설치된 인증서를 삭제하도록 요청합니다. **2.** 충전기는 DeleteCertificateResponse로 응답합니다.
5	**필수 조건**	해당 없음
6	**사후 조건**	요청한 인증서가 충전기에서 삭제되었습니다.

Figure 128. 설치된 인증서 삭제

7	**오류 처리**	해당 없음
8	**참고**	(V2G) 충전기 인증서를 설치하는 경우 사용 사례 A02 - Update Charging Station Certificate by request of CSMS 및 A03 - Update Charging Station Certificate initiated by the Charging Station를 참조하세요. V2G 인증서 체인에는 V2GRootCertificate가 포함되어서는 안 됩니다. 이것은 사용 사례 M05 - Install CA certificate in a Charging Station를 사용하여 설치해야 합니다. 마지막으로 설치된(모든) CSMSRootCertificates를 삭제할 수 있습니다. 모든 CSMSRootCertificate가 삭제되면 충전기는 CSMS 인증서를 검증할 수 없으므로 CSMS에 연결할 수 없습니다. CSMS가 마지막/모든 CSMSRootCertificates를 삭제하는 DeleteCertificateRequest를 보내기 전에 CSMS는 이것이 실제로 원하는 것인지 확인하는 것이 좋습니다. 모든 ManufacturerRootCertificate가 삭제되면 마지막으로 설치된(모든) ManufacturerRootCertificates를 삭제할 수 있으며, 충전기에 "서명된 펌웨어"를 설치할 수 없습니다.

M04 - Delete a specific certificate from a Charging Station - Requirements

Table 225. M04 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
M04.FR.01	DeleteCertificateRequest를 수신한 후	충전기는 DeleteCertificateResponse로 응답해야 합니다.	
M04.FR.02	M04.FR.01 AND 요청한 인증서가 발견되었습니다.	충전기는 인증서를 삭제하려고 시도하고 DeleteCertificateResponse에서 상태를 `Accepted` 로 설정하여 성공을 표시해야 합니다.	
M04.FR.03	M04.FR.01 AND (삭제 실패 또는 충전기가 지정된 인증서 삭제 요청을 거부합니다.)	충전기는 DeleteCertificateResponse에서 *status* 를 `Failed` 로 설정하여 실패를 표시해야 합니다.	충전기는 인증서 유형에서 마지막 인증서인 경우 인증서 삭제를 방지하기 위한 요청을 거부할 수 있습니다.
M04.FR.04	M04.FR.01 AND 요청한 인증서를 찾을 수 없습니다.	충전기는 DeleteCertificateResponse에서 *status* 를 `NotFound` 로 설정하여 실패를 표시해야 합니다.	
M04.FR.06	M04.FR.01 AND *certificateHashData* 가 _충전기 인증서_를 참조하는 경우(사용 사례 A 참조)	충전기는 DeleteCertificateResponse와 *status* = `Failed` 로 응답해야 합니다.	DeleteCertificateRequest를 통해 *충전기 인증서(Charging Station Certificate* 를 삭제할 수 없습니다.
M04.FR.07	인증서를 삭제할 때	CSMS는 충전기가 GetInstalledCertificateIdsResponse에서 인증서에 대한 certificateHashData를 보고하는 데 사용하는 것과 동일한 *hashAlgorithm* 을 사용해야 합니다.	이렇게 하면 CSMS가 충전기에서 지원하는 *hashAlgorithm* 을 사용합니다.
M04.FR.08	M04.FR.02 AND 삭제할 인증서가 하위 CA 또는 루트 인증서입니다.	충전기는 모든 자식 인증서도 삭제할 수 있습니다.	그렇지 않으면 이러한 자식 인증서는 더 이상 삭제할 수 없는 사용할 수 없는 고아 인증서(orphan certificates)로 남습니다.

M05 - Install CA certificate in a Charging Station

Table 226. M05 - 충전기에 CA 인증서 설치

번호	유형	설명
1	**이름**	충전기에 CA 인증서 설치
2	**ID**	M05
	기능 블록	M. ISO 15118 인증서 관리
3	**목표**	충전기에 설치된 인증서 관리를 용이하게 하기 위해 새 CA 인증서를 설치하는 방법.
4	**설명**	CSMS는 충전기에 새 CSMS 루트 인증서, eMobility Operator 루트 인증서, Manufacturer 루트 인증서 또는 V2G 루트 인증서를 설치하도록 요청합니다.
	Actors	충전기, CSMS

번호	유형	설명
	시나리오 설명	**1.** CSMS는 InstallCertificateRequest를 보내 충전기에 새 인증서를 설치하도록 요청합니다. **2.** 충전기는 InstallCertificateResponse로 응답합니다.
5	**필수 조건**	해당 없음
6	**사후 조건**	새 인증서가 충전기 트러스트 스토어(trust store)에 설치되었습니다.

Figure 129. 충전기에 CA 인증서 설치

7	**오류 처리**	해당 없음
8	**참고**	메시지 CertificateSignedRequest(사용 사례 A02 - Update Charging Station Certificate by request of CSMS 및 A03 - Update Charging Station Certificate initiated by the Charging Station 참조) 및 InstallCertificateRequest(사용 사례 M05)는 둘 다 인증서를 보내는 데 사용되지만 그 용도는 다릅니다. CertificateSignedRequest는 충전기 자체의 공개 인증서와 인증 기관에서 서명한 V2G 인증서를 반환하는 데 사용됩니다. InstallCertificateRequest는 루트 인증서를 설치하는 데 사용됩니다. (V2G) 충전기 인증서를 설치하는 경우 사용 사례 A02 - Update Charging Station Certificate by request of CSMS 및 A03 - Update Charging Station Certificate initiated by the Charging Station를 참조하세요. V2G 인증서 체인에는 V2GRootCertificate가 포함되어서는 안 됩니다. 이것은 이 사용 사례를 사용하여 설치되어야 합니다. 동일한 유형의 여러 인증서를 설치하는 것이 허용됩니다.

M05 - Install CA certificate in a Charging Station - Requirements

Table 227. M05 - 요구 사항

ID	전제 조건	요구 사항 정의
M05.FR.01	InstallCertificateRequest를 수신한 후	충전기는 인증서를 설치하고 InstallCertificateResponse로 응답해야 합니다.
M05.FR.02	M05.FR.01 AND 설치가 성공했습니다.	충전기는 InstallCertificateResponse에서 'status'를 'Accepted'로 설정하여 성공을 표시해야 합니다.
M05.FR.03	M05.FR.01 AND 설치가 실패했습니다.	충전기는 InstallCertificateResponse에서 'status'를 'Failed'로 설정하여 실패를 표시해야 합니다.
M05.FR.06	새 인증서가 설치되고 CertificateEntries.maxLimit이 초과될 경우	충전기는 status를 *Rejected* 로 응답해야 합니다.

ID	전제 조건	요구 사항 정의
M05.FR.07	M05.FR.01 AND 인증서가 유효하지 않습니다.	충전기는 InstallCertificateResponse에서 'status'를 'Rejected'으로 설정하여 거부를 표시해야 합니다.
M05.FR.09	`AdditionalRootCertificateCheck`가 참일 때	certificateType CSMSRootCertificate의 인증서 하나만(임시 대체 인증서 포함) 설치할 수 있습니다.
M05.FR.10	`AdditionalRootCertificateCheck`가 참이고 certificateType이 CSMSRootCertificate인 새 인증서를 설치하는 경우	새 CSMS 루트 인증서는 이전 CSMS 루트 인증서를 대체해야 하며 새 루트 인증서는 대체하는 이전 루트 인증서로 서명되어야 합니다.
M05.FR.11	M05.FR.10 AND 새 CSMS 루트 인증서는 이전 CSMS 루트 인증서로 서명되지 않아야 합니다.	충전기는 새 CSMS 루트 인증서를 설치해서는 안 되며 *Rejected* 상태로 응답해야 합니다.
M05.FR.12	M05.FR.10 AND 새 CSMS 루트 인증서가 이전 CSMS 루트 인증서에 의해 서명됨	충전기는 새 CSMS 루트 인증서를 설치하고 이전 CSMS 루트 인증서를 대체 인증서로 임시 보관하고 *Accepted* 상태로 응답해야 함
M05.FR.13	M05.FR.12 AND 충전기가 새 CSMS 루트 인증서를 사용하여 CSMS에 성공적으로 연결됨	충전기는 이전 CSMS 루트(대체) 인증서를 제거해야 함
M05.FR.14	M05.FR.12 AND 충전기는 CSMS에 다시 연결하려고 시도하지만(사용 사례 B10 - Migrate to new CSMS를 사용하여 다른 CSMS로 마이그레이션하지 않음), CSMS에서 제공한 서버 인증서가 새 CSMS 루트 인증서를 사용하여 검증할 때 유효하지 않은 것으로 판단합니다.	충전기는 이전 CSMS 루트(폴백) 인증서를 사용하여 서버 인증서를 검증해야 합니다.
M05.FR.15	M05.FR.12 AND 충전기가 사용 사례 B10 - Migrate to new CSMS를 사용하여 다른 CSMS로 마이그레이션 중이지만, 새 CSMS 루트 인증서를 사용하여 확인할 때 CSMS에서 제공한 서버 인증서가 유효하지 않다고 판단하는 경우	충전기는 사용 사례 B10 - Migrate to new CSMS에서 설명한 대로 `NetworkProfileConnectionAttempts` 메커니즘을 사용해야 합니다.
M05.FR.16	M05.FR.15 AND 시도 횟수 이후에 연결이 실패하고 AND 이전 NetworkConnectionProfile로 돌아가는 경우(*B10.FR.03 참고*)	충전기는 이전 CSMS 루트(폴백) 인증서를 사용하여 서버 인증서를 확인해야 합니다.
M05.FR.17	NOT M05.FR.10 AND 충전기의 인증서 신뢰 저장소에 이미 있는 인증서에 대한 InstallCertificateRequest를 수신한 후	충전기는 인증서를 교체하고 *status* = `Accepted` 인 InstallCertificateResponse로 응답해야 합니다.

M06 - Get V2G Charging Station Certificate status

Table 228. M06 - V2G 충전기 인증서 상태 가져오기

번호	유형	설명
1	**이름**	V2G 충전기 인증서 상태 가져오기
2	**ID**	M06

번호	유형	설명
	기능 블록	M. ISO 15118 인증서 관리
3	**목표**	충전기가 EV와 충전기 간 TLS 핸드셰이크에 필요한 OCSP 인증서 상태를 캐시할 수 있도록 합니다.
4	**설명**	케이블이 연결되고 ISO 15118 지원 EV가 충전기에 연결되면 EV는 충전기에 OCSPResponse를 통해 (SubCA) 인증서의 유효성을 증명하도록 요청합니다. SubCA별로 요청을 보내야 합니다. ISO 15118의 시간 제한 제약 조건이 외부 서버에 호출하기에는 너무 엄격하기 때문에 OCPP는 인증서의 OCSP 인증서 상태를 미리 캐시해야 합니다. 충전기는 일주일에 한 번 캐시된 OCSP 데이터를 새로 고쳐야 합니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 충전기는 GetCertificateStatusRequest를 보내 CSMS에 OCSP 인증서 상태를 제공해 달라고 요청합니다. **2.** CSMS는 GetCertificateStatusResponse로 응답합니다.
5	**필수 조건**	해당 없음
6	**사후 조건**	**성공 사후 조건:** 충전기에서 요청한 인증서에 대한 OCSP 인증서 상태를 수신했습니다. **실패한 사후 조건:** CSMS에서 OCSP 인증서 상태를 검색하는 데 실패했습니다.

Figure 130. V2G 충전기 인증서 상태 가져오기

7	**오류 처리**	해당 없음
8	**비고**	GetCertificateStatusResponse의 상태 표시기는 CSMS가 인증서 상태를 검색하는 데 성공했는지 여부를 나타냅니다. 인증서의 유효성을 나타내지 않습니다. (V2G) 충전기 인증서를 설치하려면 사용 사례 A02 - Update Charging Station Certificate by request of CSMS 및 A03 - Update Charging Station Certificate initiated by the Charging Station를 참조하세요. V2G 인증서 체인에는 V2GRootCertificate가 포함되어서는 안 됩니다. 이는 사용 사례 M05 - Install CA certificate in a Charging Station을 사용하여 설치해야 합니다. OCPP는 GetCertificateStatusRequest당 하나의 인증서만 허용합니다. GetCertificateStatusRequest에 대한 여러 답변이 예상되면 요청과 상태를 처리하는 것이 더 복잡해지기 때문입니다. 따라서 GetCertificateStatusRequest는 SubCA당 전송되어야 합니다. *responderURL* 은 OCPP에서 필수이지만 ISO 15118에서는 선택 사항입니다. 인증서에 *responderURL* 이 없으면 작동할 수 없으므로 GetCertificateStatusRequest가 예상되는 모든 인증서에는 responderURL이 필요합니다.

M06 - Get V2G Charging Station Certificate status - Requirements

Table 229. M06 - 요구 사항

ID	전제 조건	요구 사항 정의
M06.FR.01	GetCertificateStatusRequest를 수신한 후	CSMS는 GetCertificateStatusResponse로 응답해야 합니다.
M06.FR.02	M06.FR.01 AND CSMS는 OCSP 인증서 상태를 검색하는 데 성공했습니다.	CSMS는 GetCertificateStatusResponse에서 'status'를 'Accepted'로 설정하여 성공을 표시해야 합니다.
M06.FR.03	M06.FR.02	CSMS는 GetCertificateStatusResponse의 OCSPResult 필드에 OCSP 응답 데이터를 포함해야 합니다.
M06.FR.04	M06.FR.01 AND CSMS는 OCSP 인증서 상태를 검색하는 데 성공하지 못했습니다.	CSMS는 GetCertificateStatusResponse에서 *status* 를 `Failed` 로 설정하여 성공하지 못했음을 표시해야 합니다.
M06.FR.06		충전기는 V2G 인증서에 대한 OCSP 상태를 요청하고 캐시해야 합니다.
M06.FR.07		충전기 인증서가 업데이트된 후, 충전기는 새 인증서와 중간 인증서에 대해 GetCertificateStatusRequest를 보내 캐시된 OCSP 데이터를 새로 고쳐야 합니다.
M06.FR.08		CSMS는 IETF RFC 6960에 정의된 OCSPResponse에 따라 응답 데이터를 포맷해야 하며, ASN.1 [X.680]에 따라 포맷해야 합니다.
M06.FR.09		OCSPResponse 데이터는 DER로 인코딩해야 합니다.
M06.FR.10		충전기는 주 1회 이상 캐시된 OCSP 데이터를 새로 고쳐야 합니다.

N. Diagnostics

1. Introduction

이 기능 블록은 OCPP의 진단 기능을 설명합니다. 이 기능은 충전기의 문제에 대한 원격 진단을 가능하게 합니다. 충전기는 진단 정보가 포함된 파일을 업로드하도록 요청할 수 있습니다(지정된 간격으로 제한할 수 있음).

2. Use cases & Requirements

2.1. Logging

N01 - Retrieve Log Information

Table 230. N01 - 로그 정보 검색

번호	유형	설명
1	**이름**	로그 검색
2	**ID**	N01
	기능 블록	N. 진단
3	**목표**	CSMS가 충전기에서 로그 정보를 검색할 수 있도록 합니다.
4	**설명**	이 사용 사례는 충전기에서 로그 정보를 가져오는 기능을 다룹니다. CSMS는 충전기에 로그 정보가 포함된 파일을 지정된 위치(URL)에 업로드하도록 요청할 수 있습니다. 이 로그 파일의 형식은 규정되지 않았습니다. 충전기는 로그 파일을 업로드하고 CSMS에 상태 알림을 보내 업로드 상태에 대한 정보를 제공합니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS가 충전기에 GetLogRequest를 보냅니다. **2.** 충전기가 GetLogResponse로 응답합니다. **3.** 충전기가 LogStatusNotificationRequest를 status `Uploading` 으로 보냅니다. **4.** CSMS가 LogStatusNotificationResponse로 응답하여 상태 업데이트 요청을 확인합니다. **5.** 진단 파일 업로드가 진행됩니다. **6.** 충전기가 LogStatusNotificationRequest를 status `Uploaded` 로 보냅니다. **7.** CSMS는 LogStatusNotificationResponse로 응답하여 상태 업데이트 요청을 확인합니다.
5	**필수 조건(들)**	- 진단 정보를 업로드할 수 있습니다. - 파일을 업로드할 URL에 도달할 수 있고 존재합니다.
6	**사후 조건(들)**	**성공 사후 조건:** 로그 파일이 성공적으로 업로드되었습니다. **실패 사후 조건:** 로그 파일이 성공적으로 업로드되지 않아 실패했습니다.

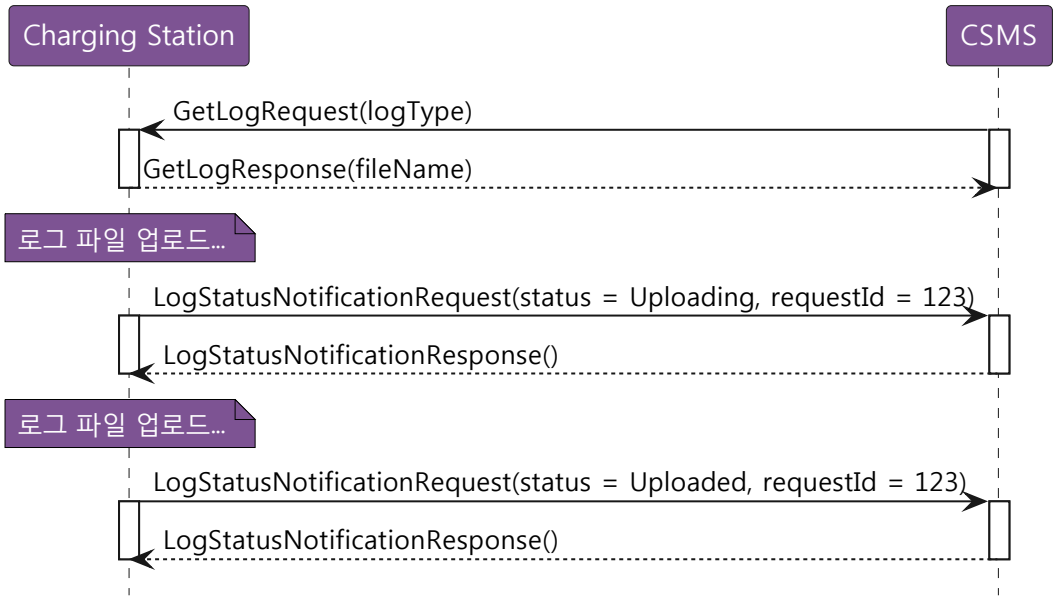

Figure 131. 시퀀스 다이어그램: 진단 받기

7	**오류 처리**	업로드가 실패하고 전송 프로토콜이 "재개"를 지원하는 경우 충전기는 업로드를 중단하기 전에 재개를 시도하는 것이 좋습니다.
8	**참고**	이 사용 사례의 예로 requestId = 123이지만 이는 어떤 값이든 될 수 있습니다. 충전기가 로그 파일을 업로드하라는 요청을 받으면 CSMS는 충전기가 파일을 업로드해야 하는 URL을 요청에 제공합니다. URL에는 파일을 업로드하는 데 사용해야 하는 프로토콜도 포함됩니다. FTP 또는 FTPS를 통해 로그 파일을 업로드하는 것이 좋습니다. FTP(S)는 HTTP보다 대용량 바이너리 데이터에 더 잘 최적화되어 있습니다. 또한 FTP(S)는 업로드를 재개할 수 있는 기능이 있습니다. 업로드가 중단된 경우 충전기는 이미 업로드한 부분 다음에 업로드를 재개할 수 있습니다. FTP URL은 *ftp://User:password@host:port/path* 형식이며, 여기서 *User:password@*, *:password* 또는 *:port* 부분은 제외할 수 있습니다. 충전기에는 지원하는 파일 전송 프로토콜을 보고하는 필수 구성 변수가 있습니다: `FileTransferProtocols` 로그 파일의 형식은 규정되지 않았습니다. FTP는 가능한 한 많은 다른 유형을 횡단할 수 있도록 수동 FTP를 사용할 수 있어야 합니다.

N01 - Retrieve Log Information - Requirements

Table 231. N01 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
N01.FR.01	GetLogRequest를 수신하고 요청된 로그 정보를 사용할 수 있는 경우	충전기는 파일 이름과 상태 *Accepted* 를 명시하는 GetLogResponse로 응답해야 합니다.	
N01.FR.02	N01.FR.01	충전기는 지정된 위치에 단일 로그 파일을 업로드하기 시작해야 합니다.	
N01.FR.03	N01.FR.02 AND GetLogRequest에는 logType *SecurityLog* 가 포함되어 있습니다.	충전기는 보안 로그를 업로드해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
N01.FR.04	N01.FR.02 AND GetLogRequest에는 logType *DiagnosticsLog* 가 포함되어 있습니다.	충전기는 진단을 업로드해야 합니다.	
N01.FR.05	GetLogRequest를 수신하고 요청된 로그 정보를 사용할 수 없는 경우	충전기는 GetLogResponse를 *Rejected* 상태로 응답해야 합니다.	
N01.FR.07		로그 업로드를 위해 전송된 모든 LogStatusNotificationRequest는 이 로그 업로드를 시작한 GetLogRequest와 동일한 requestId를 포함해야 합니다.	
N01.FR.08	로그 문서 업로드가 시작되면	충전기는 LogStatusNotificationRequest를 *Uploading* 상태로 전송해야 합니다.	
N01.FR.09	로그 문서가 성공적으로 업로드된 경우	충전기는 LogStatusNotificationRequest를 상태 *Uploaded* 로 전송해야 합니다.	
N01.FR.10	로그 문서 업로드가 실패한 경우	충전기는 LogStatusNotificationRequest를 상태 `UploadFailure`, `BadMessage`, `PermissionDenied` 또는 `NotSupportedOperation` 로 전송해야 합니다.	모든 재시도가 실패한 후에만 상태를 전송하는 것이 좋습니다. 충전기는 각 재시도마다 새로운 `Uploading` 상태를 전송할 수 있습니다.
N01.FR.12	충전기가 로그 파일을 조립하거나 업로드하고 충전기가 새로운 GetLogRequest를 수신하는 경우	충전기는 진행 중인 로그 파일 업로드를 취소하고 *AcceptedCanceled* 상태로 응답해야 합니다.	
N01.FR.13		LogStatusNotificationRequest의 필드 requestId는 필수입니다. 단, 메시지가 TriggerMessageRequest에 의해 트리거되었고 진행 중인 로그 업로드가 없는 경우는 예외입니다.	
N01.FR.14		충전기와 CSMS는 로그 파일 업로드를 위한 전송 메커니즘으로 최소한 HTTP(s)를 지원하는 것이 좋습니다.	HTTP 전송은 OCPP 메시징에도 사용되므로 지원될 가능성이 가장 높습니다.
N01.FR.15		충전기는 보안 전송을 위해 최소한 CSMS 신뢰 체인을 지원해야 합니다	
N01.FR.16		충전기는 운영 체제에서 제공하는 일반적인 CA를 지원하는 것이 좋습니다	CSMS의 로그 파일 저장소는 CSMS 자체와 별도로 운영되는 클라우드 서비스일 수 있으며 CSMS 신뢰 체인의 일부가 아닐 수 있습니다.

ID	전제 조건	요구 사항 정의	참고
N01.FR.17	CSMS가 업로드에 대한 기본 인증을 요구하는 경우	CSMS는 OCPP 연결에 사용된 것과 다른 기본 인증 비밀번호를 업로드에 요구하는 것이 좋습니다.	이는 로그 파일 저장소가 다른 시스템인 경우 OCPP 비밀번호가 제3자에게 유출되는 것을 방지하기 위한 것입니다. 기본 권한은 다음과 같이 URL에 추가할 수 있습니다. *http://username:password@*csms.org/logs
N01.FR.18		CSMS가 충전기에서 업로드하는 PUT 및 POST 요청을 모두 수락하는 것이 좋습니다.	
N01.FR.19	충전기에서 HTTP(s) POST 요청을 사용하여 로그 파일을 업로드하는 경우	충전기는 최소한 다음 속성을 제공해야 합니다. `Content-Type:`(예: application/octet-stream) 및 파일 이름을 명시한 `Content-Disposition:`	예: Content-Type: application/octet-stream Content-Disposition: form-data; name="uploadedfile"; filename="logfile_20210420.zip"
N01.FR.20	N01.FR.12 AND 충전기에서 로그 파일 업로드 취소하였습니다.	충전기에서 LogStatusNotificationRequest를 *status* = `AcceptedCanceled` 로 보내야 합니다.	N01.FR.12는 "SHOULD" 요구 사항입니다. 요구 사항이 실행될 때만 상태 알림을 보내야 합니다.

2.2. Configure Monitoring

NOTE	충전기 모니터링을 관리하려면 장치 모델 개념에 대한 기본적인 이해가 필수적입니다. 이러한 개념은 "OCPP 2.0.1: Part 1 - Architecture & Topology", 4장에서 설명합니다.

N02 - Get Monitoring report

Table 232. N02 - 모니터링 보고서 가져오기

번호	유형	설명
1	**이름**	모니터링 보고서 가져오기
2	**ID**	N02
	기능 블록	N. 진단
3	**목표**	CSMS가 구성 요소 및 변수별로 구성된 모니터링 설정에 대한 보고서를 검색할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSMS가 충전기에 구성 요소 및 변수별로 구성된 모니터링 설정에 대한 보고서를 보내도록 요청하는 방법을 설명합니다. 선택적으로 이 목록은 monitoringCriteria 및 componentVariables에서 필터링할 수 있습니다.
	Actors	충전기, CSMS, CSO

번호	유형	설명
	시나리오 설명	**1.** CSO가 CSMS를 트리거하여 충전기에 모니터링 보고서를 요청합니다. **2.** CSMS가 충전기에 GetMonitoringReportRequest를 보냅니다. **3.** 충전기가 GetMonitoringReportResponse로 응답합니다. **4.** 충전기가 NotifyMonitoringReportRequest를 CSMS로 보냅니다. **5.** CSMS가 NotifyMonitoringReportResponse로 응답합니다. **6.** 모니터링 보고서의 모든 데이터가 전송될 때까지 4단계와 5단계를 반복합니다.
5	**필수 조건**	충전기가 모니터링을 지원합니다.
6	**사후 조건**	CSMS가 구성된 모니터링 설정에 대한 보고서를 받았습니다.

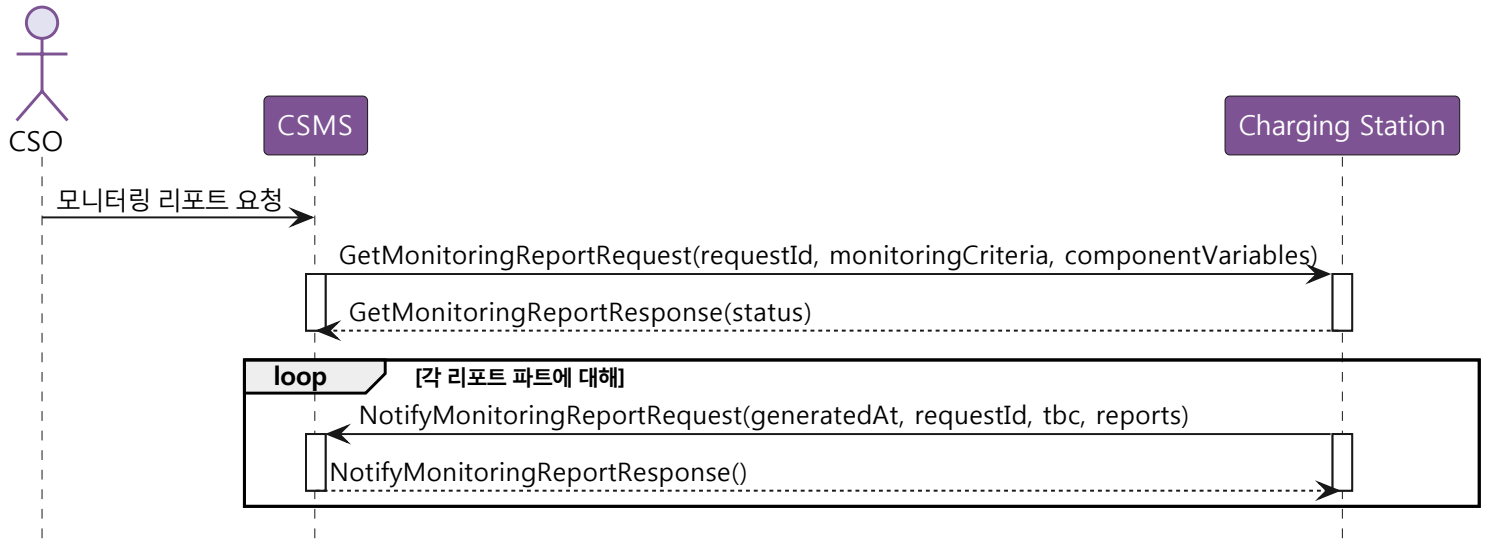

Figure 132. 시퀀스 다이어그램: 모니터링 보고서 가져오기

7	**오류 처리**	n/a
8	**설명**	n/a

N02 - Get Monitoring Report - Requirements

Table 233. N02 - 요구 사항

ID	전제 조건	요구 사항 정의
N02.FR.01	NOT N02.FR.10 AND 충전기가 지원되는 *monitoringCriteria* 또는 *monitoringCriteria* 없이 GetMonitoringReportRequest를 수신하는 경우	충전기는 Accepted와 함께 GetMonitoringReportResponse를 보내야 합니다.
N02.FR.02	충전기가 지원되지 않는 *monitoringCriteria* 에 대한 GetMonitoringReportRequest를 수신하는 경우	충전기는 NotSupported와 함께 GetMonitoringReportResponse를 보내야 합니다.
N02.FR.03	N02.FR.01	충전기는 하나 이상의 NotifyMonitoringReportRequest 메시지를 통해 요청된 정보를 CSMS로 보내야 합니다.
N02.FR.04	N02.FR.01 및 GetMonitoringReportRequest에는 *requestId* 가 포함되어 있습니다.	이 GetMonitoringReportRequest에 대해 전송된 모든 NotifyMonitoringReportRequest에는 동일한 *requestId* 가 포함되어야 합니다.

ID	전제 조건	요구 사항 정의
N02.FR.05	N02.FR.01 및 *monitoringCriteria* 및 *componentVariables* 가 모두 비어 있지 않습니다.	하나 이상의 NotifyMonitoringReportRequest 메시지에서 보고된 모니터 세트는 *monitoringCriteria* 및 *componentVariables* 로 정의된 세트로 제한됩니다.
N02.FR.06	N02.FR.01 AND *monitoringCriteria* 가 비어 있지 않음 AND *componentVariables* 가 비어 있음.	하나 이상의 NotifyMonitoringReportRequest 메시지에서 보고된 모니터 세트는 *monitoringCriteria* 에서 정의한 세트로 제한됩니다.
N02.FR.07		하나의 GetMonitoringReportRequest 메시지에서 componentVariables의 최대 수는 `ItemsPerMessageGetReport` 구성 변수에서 제공됩니다.
N02.FR.08	N02.FR.01 AND *monitoringCriteria* 가 없음 AND *componentVariables* 가 비어 있지 않음.	하나 이상의 NotifyMonitoringReportRequest 메시지에서 보고된 모니터 세트는 *componentVariables* 에서 정의한 세트로 제한됩니다.
N02.FR.09		NotifyMonitoringReportRequest의 seqNo 필드에 포함된 시퀀스 번호는 보고서당 증가합니다. 따라서 첫 번째 보고서 부분을 포함하는 NotifyMonitoringReportRequest 메시지는 값이 *0* 인 seqNo를 가져야 합니다.
N02.FR.10	충전기가 기준 조합을 포함하는 GetMonitoringReportRequest를 수신하여 빈 결과 집합이 생성되는 경우	충전기는 GetMonitoringReportResponse(*status* =`EmptyResultSet`)로 응답해야 합니다.
N02.FR.11	N02.FR.01 AND *monitoringCriteria* 가 비어 있고 *comComponentVariables* 가 비어 있습니다.	모든 기존 모니터 세트는 하나 이상 NotifyMonitoringReportRequest메시지로 보고됩니다.
N02.FR.12	*monitoringCriteria* 에 `ThresholdMonitoring` 이 포함된 경우	*type* = `UpperThreshold` 또는 *type* = `LowerThreshold` 인 모든 모니터가 보고됩니다.
N02.FR.13	*monitoringCriteria* 에 `DeltaMonitoring` 이 포함된 경우	*type* = `Delta` 인 모든 모니터가 보고됩니다.
N02.FR.14	*monitoringCriteria* 에 `PeriodicMonitoring` 이 포함된 경우	*type* = `Periodic` 또는 *type* = `PeriodicClockAligned` 인 모든 모니터가 보고됩니다.
N02.FR.16	충전기에서 GetMonitoringReportRequest를 수신하는 경우 *componentVariable* 요소 중 *variable* 이 누락된 경우	충전기는 *componentVariable* 의 *component* 의 모든 *variable* 에 대해 보고해야 합니다.
N02.FR.17	충전기가 *componentVariable* 요소가 있는 GetMonitoringReportRequest를 수신하고 *variable* 이 있지만 *instance* 가 없는 경우	충전기는 *componentVariable* 내 *component* 의 *variable* 의 모든 인스턴스에 대해 보고해야 합니다.
N02.FR.18	N02.FR.11 AND 충전기가 *component.evse.id* 가 있는 *componentVariable* 요소에 *component* 가 있는 GetMonitoringReportRequest를 수신하지만 *component.evse.connector* 가 없는 경우	충전기는 N02.FR.20을 고려하면서 모든 *component.evse.connector* 에 대해 이 *component.name*, *component.instance* 및 *component.evse.id* 를 가진 구성 요소를 보고해야 합니다.

ID	전제 조건	요구 사항 정의
N02.FR.19	N02.FR.11 AND 충전기가 *component.evse.id* 가 없는 *componentVariable* 요소에 *component* 가 포함된 GetMonitoringReportRequest를 수신하는 경우	충전기는 N02.FR.20을 고려하여 모든 *component.evse* 필드(*component.evse* 가 없는 최상위 구성 요소 포함)에 대해 이 *component.name*, *component.instance* 를 갖는 구성 요소를 보고해야 합니다.
N02.FR.20	N02.FR.11 AND 충전기가 *component.instance* 에 대한 값을 갖는 *componentVariable* 요소에 *component* 가 포함된 GetMonitoringReportRequest를 수신하는 경우	충전기는 N02.FR.18, N02.FR.19를 고려하여 모든 *component.instance* 필드에 대해 이 *component.name* 을 가진 구성 요소를 보고해야 합니다.
N02.FR.21	N02.FR.11 AND 충전기가 *component.instance* 필드가 없는 *componentVariable* 요소에 *component* 가 포함된 GetMonitoringReportRequest를 수신하는 경우	충전기는 N02.FR.18, N02.FR.19를 고려하여 모든 *component.instance* 필드에 대해 이 *component.name* 을 가진 구성 요소 또는 *component.instance* 필드가 없는 구성 요소를 보고해야 합니다.

N03 - Set Monitoring Base

Table 234. N03 - 모니터링 기반 설정

번호	유형	설명
1	**이름**	모니터링 기반 설정
2	**ID**	N03
	기능 블록	N. 진단
3	**목표**	CSMS가 충전기에 미리 구성된 모니터링 설정 세트를 활성화하도록 요청할 수 있는 기능을 제공하며, 이는 **MonitoringBase** 값으로 표시됩니다.
4	**설명**	이 사용 사례는 CSMS가 충전기에 미리 구성된 모니터링 설정 세트를 활성화하도록 요청하는 방법을 설명하며, 이는 **MonitoringBase** 값으로 표시됩니다. 충전기 제조업체가 All, FactoryDefault 및 HardWiredOnly로 활성화되는 모니터링 설정을 정의합니다.
	Actors	충전기, CSMS, CSO
	시나리오 설명	**1.** CSO가 CSMS를 트리거하여 충전기에 모니터링 기반을 설정하도록 요청합니다. **2.** CSMS가 충전기에 SetMonitoringBaseRequest를 보냅니다. **3.** 충전기가 SetMonitoringBaseResponse로 응답합니다.
5	**필수 조건**	충전기가 모니터링을 지원합니다.
6	**사후 조건**	충전기가 **MonitoringBase** 값으로 표시된 대로 모니터링 설정 세트를 활성화했습니다.

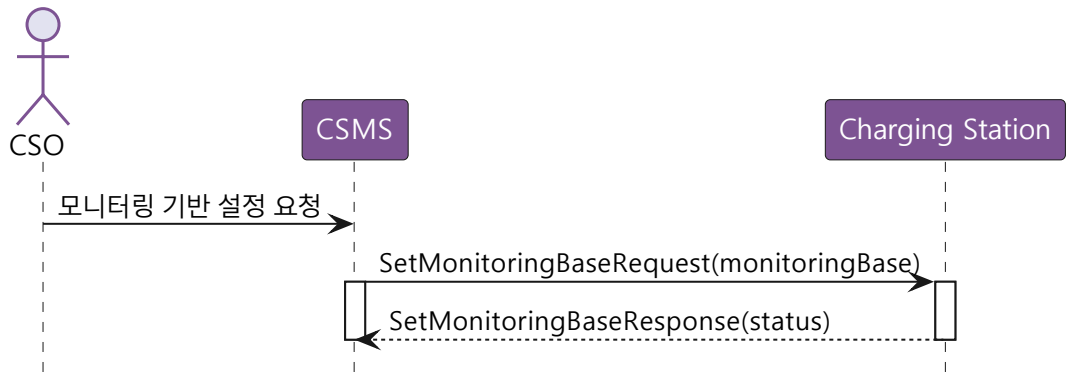

Figure 133. 시퀀스 다이어그램: 모니터링 기반 설정

7	**오류 처리**	n/a
8	**참고**	`HardWiredOnly` 또는 `FactoryDefault` 에 대한 SetMonitoringBaseRequest를 수신하면 충전기는 이전에 구성된 사용자 지정 모니터를 모두 삭제하고 지정된 MonitoringBase와 관련된 모니터링 설정을 활성화합니다. MonitoringBase = `All` 인 경우 충전기는 미리 구성된 모든 모니터를 활성화하고 이전에 구성된 사용자 지정 모니터는 그대로 둡니다. 여기에는 기존 미리 구성된 모니터를 변경할 때 생성된 사용자 지정 모니터가 포함됩니다. `All` 과 `FactoryDefault` 에 대한 사전 구성된 모니터 세트가 동일할 때, 두 가지의 차이점은 `FactoryDefault` 의 경우 모든 사용자 정의 모니터가 공장 기본 사전 구성된 모니터가 복원되기 전에 삭제된다는 것입니다.

N03 - Set Monitoring Base - Requirements

Table 235. N03 - 요구 사항

ID	전제 조건	요구 사항 정의
N03.FR.01	충전기가 SetMonitoringBaseRequest를 수락하면	그러면 충전기는 Accepted와 함께 SetMonitoringBaseResponse를 보내야 합니다.
N03.FR.02	충전기가 지원되지 않는 *monitoringBase* 에 대한 SetMonitoringBaseRequest를 수신하는 경우	충전기는 NotSupported와 함께 SetMonitoringBaseResponse를 보내야 합니다.
N03.FR.03	N03.FR.01 AND 충전기가 SetMonitoringBaseRequest와 함께 *monitoringBase* All를 수신하는 경우	충전기는 설치된 모든 사용자 지정 모니터(변경된 사전 구성된 모니터 포함)를 그대로 두고 모든 사전 구성된 모니터링을 활성화해야 합니다.
N03.FR.04	N03.FR.01 AND 충전기가 SetMonitoringBaseRequest와 *monitoringBase* FactoryDefault를 수신한 경우	충전기는 모든 사용자 지정 모니터(무시된 사전 구성된 모니터 포함)를 삭제하고 제조업체에서 권장하는 대로 기본 모니터링 설정을 활성화해야 합니다.
N03.FR.05	N03.FR.01 AND 충전기가 SetMonitoringBaseRequest와 *monitoringBase* HardWiredOnly를 수신한 경우	충전기는 모든 사용자 지정을 지우고 모든 사전 구성된 모니터를 비활성화해야 합니다. 하드와이어 모니터만 활성 상태로 유지됩니다.

N04 - Set Variable Monitoring

Table 236. N04 - 변수 모니터링 설정

번호	유형	설명
1	**이름**	변수 모니터링 설정
2	**ID**	N04
	기능 블록	N. 진단
3	**목표**	CSMS가 충전기에 변수에 대한 모니터링 트리거를 설정하도록 요청할 수 있는 기능을 제공합니다.
4	**설명**	이 사용 사례는 CSMS가 충전기에 변수에 대한 모니터링 트리거를 설정하도록 요청하는 방법을 설명합니다. 상위 또는 하위 임계값, 델타 변경 또는 주기적 보고에 대해 여러 트리거를 설정할 수 있습니다.
	Actors	충전기, CSMS, CSO
	시나리오 설명	**1.** CSO가 CSMS를 트리거하여 충전기에 변수 모니터링 설정을 요청합니다. **2.** CSMS가 충전기에 SetVariableMonitoringRequest를 보냅니다. **3.** 충전기가 SetVariableMonitoringResponse로 응답합니다.
5	**필수 조건**	충전기가 모니터링을 지원합니다. 특정 변수가 모니터링을 지원합니다.
6	**사후 조건**	충전기가 변수에서 모니터링 트리거 세트를 활성화했습니다.

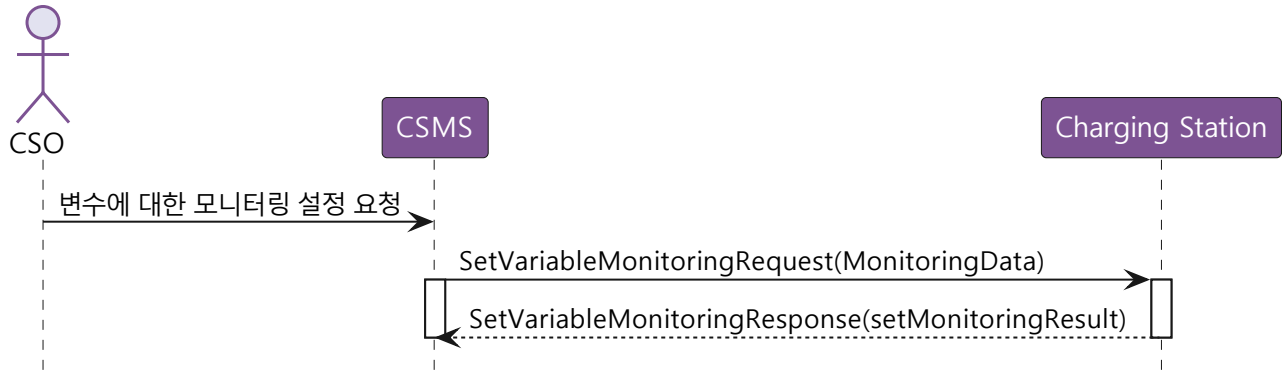

Figure 134. 시퀀스 다이어그램: 변수 모니터링 설정

7	**오류 처리**	n/a
8	**참고**	모든 variableMonitoring 설정은 재부팅 후에도 지속됩니다. variableMonitoring 설정은 모니터링되는 변수가 여전히 존재하고 여전히 모니터링 가능한 경우 펌웨어 업데이트 후에도 지속됩니다. 그렇지 않으면 variableMonitoring 설정이 제거됩니다.

N04 - Set Variable Monitoring - Requirements

Table 237. N04 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
N04.FR.01	충전기가 X개의 SetMonitoringData 요소가 포함된 SetVariableMonitoringRequest를 수신하면	충전 스테이션은 SetVariableMonitoringResponse로 응답해야 합니다. 이때 SetMonitoringResult 요소는 SetVariableMonitoringRequest의 모든 SetMonitoringData 요소마다 하나씩 동일한 수(X)로 지정되어야 합니다.	
N04.FR.02	N04.FR.01	SetVariableMonitoringResponse의 모든 SetMonitoringResult 요소에는 SetVariableMonitoringRequest의 SetVariableMonitoringRequest 요소 중 하나와 동일한 *component* 및 *variable* 조합이 포함되어야 합니다.	
N04.FR.03	충전기가 SetVariableMonitoringRequest를 수신하고 SetMonitoringData에 알 수 없는 Component가 있는 경우	충전기는 해당 SetMonitoringResult의 *attributeStatus* 필드를 UnknownComponent로 설정해야 합니다.	
N04.FR.04	충전기가 SetVariableMonitoringRequest를 수신하고 Component에 대해 알려지지 않은 SetMonitoringData를 수신하는 경우	충전기는 해당 SetMonitoringResult의 *attributeStatus* 필드를 UnknownVariable로 설정해야 합니다.	
N04.FR.05	충전기가 특정 변수에서 지원하지 않는 MonitorType을 포함하는 SetVariableMonitoringRequest를 수신하는 경우	충전기는 해당 SetMonitoringResult의 attributeStatus 필드를 UnsupportedMonitorType으로 설정해야 합니다.	
N04.FR.06	충전기가 모니터 유형이 UpperThreshold 또는 LowerThreshold인 SetVariableMonitoringRequest를 수신 AND *monitorValue* 가 주어진 Variable의 범위보다 낮거나 높을 때	충전기는 해당 SetMonitoringResult의 *attributeStatus* 필드를 Rejected로 설정해야 합니다.	선택 사항인 *statusInfo* 요소에서 자세한 정보를 제공할 수 있습니다.
N04.FR.07	충전기가 안전 요구 사항과 충돌하는 모니터에 대한 SetVariableMonitoringRequest를 수신하는 경우.	충전기는 해당 SetMonitoringResult의 *attributeStatus* 필드를 Rejected로 설정할 수 있습니다.	예를 들어, 요청된 모니터링이 공장 설정 보안 모니터링을 재정의하는 경우.
N04.FR.08	충전기가 SetMonitoringData에서 제공된 *monitorValue* 를 설정할 수 있는 경우	충전기는 해당 SetMonitoringResult의 *attributeStatus* 필드를 Accepted로 설정해야 합니다.	다양한 monitor types를 처리하는 방법에 대해서는 사용 사례 N07 - Alert Event를 참조하세요.
N04.FR.09		한 SetVariableMonitoringRequest 메시지에서 *monitoringData* 의 최대 크기와 항목 수는 `ItemsPerMessageSetVariableMonitoring` 및 `BytesPerMessageSetVariableMonitoring` 구성 변수에 의해 결정됩니다.	

ID	전제 조건	요구 사항 정의	참고
N04.FR.10	충전기가 동일한 *type* 및 *severity* 를 가진 모니터가 다른 *id* 로 이미 존재하는 *component/variable* 조합에 대한 SetVariableMonitoringRequest를 수신하는 경우.	충전기는 해당 SetMonitoringResult의 *attributeStatus* 필드를 Duplicate로 설정해야 합니다.	동일한 변수에 동일한 심각도를 가진 동일한 유형의 모니터가 두 개 있을 수 없습니다. 예를 들어, 구성 요소/변수에 UpperThreshold 값이 "67"이고 심각도가 "4-Error"인 모니터가 있는 경우 동일한 심각도 "4-Error"가 정의된 다른 UpperThreshold 값이 "78"일 수 없습니다.
N04.FR.11	충전기가 ID 없이 SetVariableMonitoringRequest를 수신 AND N04.FR.08	충전기는 ID를 생성하여 SetVariableMonitoringResponse에 반환합니다.	
N04.FR.12	충전기가 ID와 함께 SetVariableMonitoringRequest를 수신 AND 제공된 ID와 일치하는 모니터가 존재 AND 제공된 구성 요소/변수 조합이 기존 VariableMonitor와 일치합니다.	충전기는 모니터를 대체해야 합니다.	
N04.FR.13	충전기가 ID와 함께 SetVariableMonitoringRequest를 수신 AND 제공된 ID와 일치하는 모니터가 없습니다.	충전기는 해당 SetMonitoringResult의 *attributeStatus* 필드를 Rejected로 설정해야 합니다.	
N04.FR.14	충전기가 Delta 유형의 SetVariableMonitoringRequest를 수신하고 값에 음수 값이 포함된 경우.	충전기는 해당 SetMonitoringResult의 *attributeStatus* 필드를 Rejected로 설정해야 합니다.	선택 사항인 *statusInfo* 요소에서 자세한 정보를 제공할 수 있습니다.
N04.FR.15	N04.FR.12 AND 교체된 VariableMonitor는 'PreconfiguredMonitors'에 속했습니다.	새로운 VariableMonitor는 SetMonitoringBaseRequest로 재설정될 때까지 'CustomMonitor'로 분류됩니다.	
N04.FR.16	충전기가 ID가 있는 SetVariableMonitoringRequest를 수신 AND 제공된 ID와 일치하는 모니터가 존재 AND 제공된 구성 요소/변수 조합이 기존 VariableMonitor와 일치하지 않는 경우.	충전기는 *Rejected* 로 응답해야 하며 VariableMonitor를 대체해서는 안 됩니다.	모니터의 변수나 구성 요소를 변경할 수 없습니다.
N04.FR.17	CSMS가 숫자형이 아닌 변수에 대해 Delta 유형의 SetVariableMonitoringRequest를 보낼 때	1의 *monitorValue* 를 사용하는 것이 좋습니다.	*monitorValue* 는 숫자형이 아닌 유형(예: 10진수 또는 정수를 제외한 모든 유형)에는 무관합니다. 모니터는 변수가 변경될 때마다 트리거되기 때문입니다.
N04.FR.18	N04.FR.12 AND SetVariableMonitoringRequest의 *id* 는 `HardWiredMonitor` 를 참조합니다.	충전기는 *Rejected* 로 응답해야 하며 VariableMonitor를 대체해서는 안 됩니다.	하드와이어 모니터(hardwired monitor)는 변경할 수 없습니다.
N04.FR.19	충전기가 재부팅되었습니다.	CSMS는 GetMonitoringReportRequest 메시지를 보내 새 모니터 목록을 가져오는 것이 좋습니다.	사용자 지정 모니터는 재부팅 또는 펌웨어 업데이트 후에도 지속되지만 ID가 변경되었을 수 있습니다.

N05 - Set Monitoring Level

Table 238. N05 - 모니터링 수준 설정

번호	유형	설명
1	**이름**	모니터링 수준 설정
2	**ID**	N05
	기능 블록	N. 진단
3	**목표**	CSMS가 NotifyEventRequest를 통해 모니터링 이벤트 보고를 특정 심각도 이하인 모니터로만 제한하도록 충전기에 요청할 수 있는 기능을 제공합니다.
4	**설명**	모니터링 이벤트 보고를 심각도 이하인 모니터로만 제한하는 것이 바람직할 수 있습니다. 예를 들어, 충전기와 CSMS 간의 데이터 트래픽을 어떤 이유로 제한해야 하는 경우입니다. CSMS는 SetMonitoringLevelRequest 메시지를 통해 충전기에서 어떤 이벤트를 알릴지 제어할 수 있습니다.
	Actors	충전기, CSMS, CSO
	시나리오 설명	**1.** CSO는 심각도 수준 제한을 설정하여 충전기에 모니터링 이벤트 보고를 제한하도록 요청하도록 CSMS를 트리거합니다. **2.** CSMS는 충전기에 SetMonitoringLevelRequest를 보냅니다. **3.** 충전기는 SetMonitoringLevelResponse로 응답합니다.
5	**필수 조건**	충전기에서 모니터링 지원
6	**사후 조건**	충전기에서 NotifyEventRequest에 의한 모니터링 이벤트 보고를 사용자가 원하는 것으로만 제한했습니다.

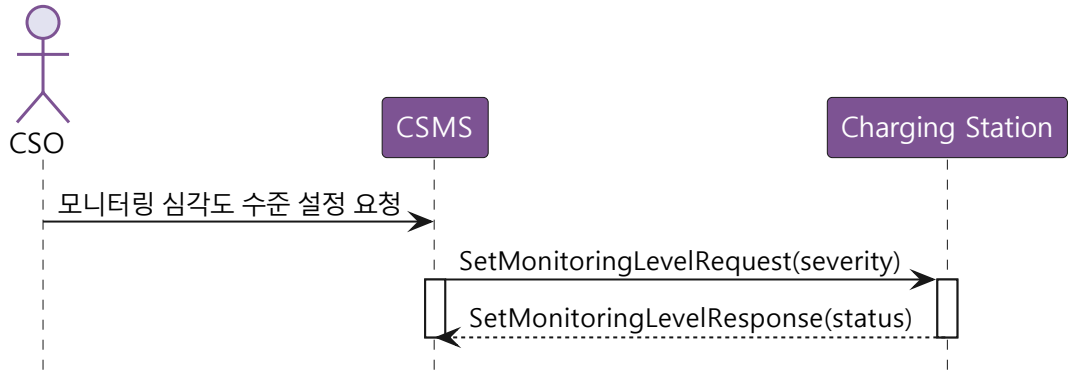

Figure 135. 시퀀스 다이어그램: 모니터링 수준 설정

7	**오류 처리**	n/a
8	**참고**	n/a

N05 - Set Monitoring Level - Requirements

Table 239. N05 - 요구 사항

ID	전제 조건	요구 사항 정의
N05.FR.01	충전기가 SetMonitoringLevelRequest를 수락할 때	충전기는 Accepted와 함께 SetMonitoringLevelResponse를 보내야 합니다.
N05.FR.02	충전기가 범위를 벗어난 *severity* 에 대한 SetMonitoringLevelRequest를 수신하는 경우	충전기는 Rejected와 함께 SetMonitoringLevelResponse를 보내야 합니다.
N05.FR.03	N05.FR.01	충전기는 NotifyEventRequest에 의한 모니터링 이벤트 보고를 지정된 심각도보다 낮거나 같은 심각도 번호를 가진 모니터로만 제한해야 합니다.

N06 - Clear / Remove Monitoring

Table 240. N06 - 모니터링 지우기/제거

번호	유형	설명
1	**이름**	모니터링 지우기/제거
2	**ID**	N06
	기능 블록	N. 진단
3	**목표**	CSMS에 모니터링 설정을 지우거나 제거할 수 있는 기능을 제공합니다.
4	**설명**	모니터링 설정은 모니터링 설정의 ID와 함께 ClearVariableMonitoringRequest를 보내서 지울 수 있습니다(제거할 수 있습니다).
	Actors	충전기, CSMS, CSO
	시나리오 설명	**1.** CSO가 CSMS를 트리거하여 충전기에서 하나 이상의 변수를 지우거나 제거하도록 요청합니다. **2.** CSMS가 ClearVariableMonitoringRequest를 충전기로 보냅니다. **3.** 충전기가 ClearVariableMonitoringResponse로 응답합니다.
5	**필수 조건**	충전기가 모니터링을 지원합니다.
6	**사후 조건**	충전기가 요청된 모니터링 설정을 지우거나 제거했습니다.

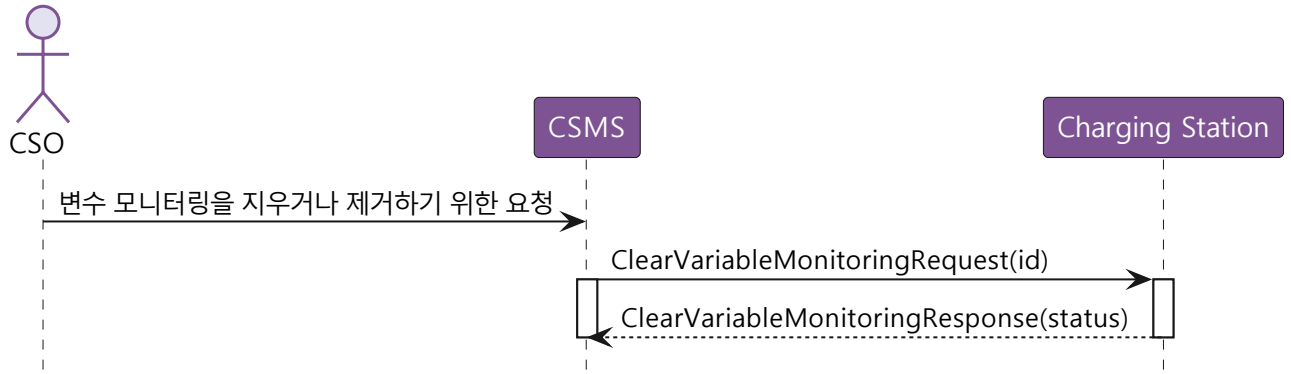

Figure 136. 시퀀스 다이어그램: 모니터링 지우기/제거

7	**오류 처리**	n/a

8	**참고**	n/a

N06 - Clear / Remove Monitoring - Requirements

Table 241. N06 - 요구 사항

ID	전제 조건	요구 사항 정의
N06.FR.01	충전기가 ClearVariableMonitoringRequest를 수락할 때	충전기는 Accepted와 함께 ClearVariableMonitoringResponse를 보내야 합니다.
N06.FR.02	충전기가 존재하지 않는 *id* 를 가진 ClearVariableMonitoringRequest를 수신하는 경우	충전기는 ClearVariableMonitoringResponse를 NotFound와 함께 전송해야 합니다.
N06.FR.03	충전기가 지울 수 없는 모니터를 참조하는 *id* 를 가진 ClearVariableMonitoringRequest를 수신하는 경우(예: 하드코딩된 경우).	충전기는 ClearVariableMonitoringResponse를 Rejected와 함께 전송해야 합니다.
N06.FR.04		CSMS는 충전기에서 `ItemsPerMessageClearVariableMonitoring` 및 `BytesPerMessageClearVariableMonitoring`을 통해 보고한 것보다 더 많은 *id* 요소를 ClearVariableMonitoringRequest에 넣지 않아야 합니다.
N06.FR.05		ClearVariableMonitoringRequest의 *모든 id* 에 대해 충전기는 CSMS로 전송된 ClearVariableMonitoringResponse에 *clearMonitoringResult* 요소를 추가해야 합니다.
N06.FR.06	충전기가 `ItemsPerMessageClearVariableMonitoring`에서 허용하는 것보다 더 많은 *id* 요소가 포함된 ClearVariableMonitoringRequest를 수신합니다.	충전기가 CALLERROR(OccurenceConstraintViolation)로 응답할 수 있습니다.
N06.FR.07	충전기가 `BytesPerMessageClearVariableMonitoring`에서 허용하는 것보다 더 많은 바이트 길이의 ClearVariableMonitoringRequest를 수신합니다.	충전기가 CALLERROR(FormatViolation)로 응답할 수 있습니다.

2.3. Monitoring Events

N07 - Alert Event

Table 242. N07 - 경고 이벤트

번호	유형	설명
1	**이름**	경고 이벤트
2	**ID**	N07
	기능 블록	N. 진단
3	**목표**	충전기에 모니터링 이벤트에 대해 CSMS에 알릴 수 있는 기능을 제공합니다.

번호	유형	설명
4	**설명**	NotifyEventRequest는 VariableMonitoring 설정이 트리거된 모든 구성 요소/변수를 보고합니다. 이벤트를 트리거하는 데 책임이 있는 VariableMonitoring 설정만 포함됩니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 임계값 또는 델타 값이 초과되면 충전기는 NotifyEventRequest를 CSMS로 보냅니다. **2.** CSMS는 NotifyEventResponse로 응답합니다.
5	**필수 조건**	충전기에 활성 모니터링 설정이 있습니다. 모니터링 설정은 SetVariableMonitoring 메시지를 통해 명시적으로 구성되었거나 충전기 펌웨어에 "하드 와이어링(hard-wired)"되어 있을 수 있습니다.
6	**사후 조건**	충전기가 CSMS에 모니터링 이벤트에 대해 알렸습니다.

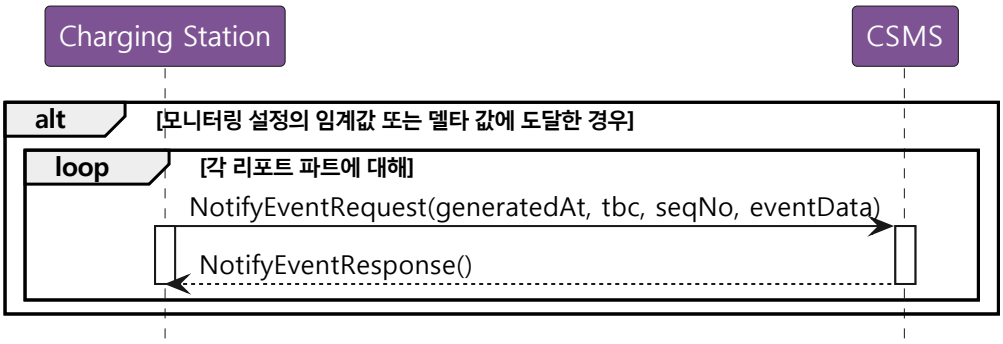

Figure 137. 시퀀스 다이어그램: 경고 이벤트

7	**오류 처리**	n/a
8	**참고**	요구 사항 N07.FR.04는 OfflineMonitoringEventQueuingSeverity보다 심각도가 낮거나 같은 이벤트는 충전기가 오프라인인 동안 대기열에 넣고 온라인이 되면 전달해야 한다고 명시합니다. 즉, OfflineMonitoringEventQueuingSeverity보다 심각도가 높은 이벤트는 CSMS로 전송되지 않습니다. 그 결과, 충전기가 다시 온라인이 되면 논리적 이벤트 체인이 끊어질 수 있습니다. 예를 들어, 임계값을 초과하는 변수에 대한 모니터링 이벤트가 오프라인 중에 발생하여 전송되지 않았습니다. 온라인 상태로 돌아오면 어느 시점에서 모니터링 이벤트가 보고되고 변수 *cleared* 가 true로 설정되지만 CSMS는 임계값을 초과했다는 사실조차 알지 못합니다. CSMS는 이를 처리할 수 있어야 합니다. 이러한 모니터링 이벤트를 단순히 삭제하는 것이 아니라 OfflineMonitoringEventQueuingSeverity를 초과하는 모니터의 평가를 충전기가 다시 온라인 상태가 될 때까지 지연함으로써 사양을 준수하는 동시에 이 문제를 방지할 수 있습니다. 그 결과 충전기가 다시 온라인 상태가 되면 CSMS는 현재 상황에 적용되는 모니터링 이벤트를 가져오고 모니터에 대한 최신 정보를 모두 얻습니다. 오프라인 기간 동안 트리거되고 지워진 모니터링 이벤트만 CSMS에 표시되지 않습니다.

N07 - Alert Event - Requirements

Table 243. N07 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
N07.FR.02	모니터링된 값이 설정된 *UpperThreshold* 또는 *LowerThreshold* 내로 돌아올 때	충전기는 *cleared* 속성이 참인 eventData와 함께 NotifyEventRequest를 보내야 합니다.	

ID	전제 조건	요구 사항 정의	참고
N07.FR.03	CSMS가 NotifyEventRequest를 수신할 때	CSMS는 빈 NotifyEventResponse로 응답해야 합니다.	
N07.FR.04	모니터가 트리거되고 AND 모니터의 심각도 번호가 구성 변수 `OfflineMonitoringEventQueuingSeverity`에 설정된 심각도 번호와 같거나 낮음 AND 충전기가 *오프라인* 상태임	충전기는 이 NotifyEventRequest를 큐에 넣고 다시 온라인 상태가 되면 전달해야 합니다.	
N07.FR.05	모니터가 트리거되고 AND 다른 이벤트가 이 이벤트를 발생시킨 경우	충전기는 NotifyEventRequest 메시지의 eventData 요소의 *cause* 필드에 다른 이벤트의 *eventId* 를 포함할 수 있습니다.	
N07.FR.06	모니터가 트리거될 때	NotifyEventRequest의 eventData 요소에는 이벤트를 발생시킨 Component, Variable 및 *variableMonitoringId* 가 포함되어야 합니다.	
N07.FR.07	모니터가 트리거될 때	충전기는 이 이벤트에 대해 전송된 첫 번째 NotifyEventRequest의 *seqNo* 를 0으로 설정해야 합니다.	
N07.FR.10	모니터가 트리거될 때 AND 쓰기 전용 변수에 variableMonitoring 설정이 설정되었습니다.	NotifyEventRequest의 actualField는 비어 있어야 합니다.	
N07.FR.11	설정된 *UpperThreshold* 또는 *LowerThreshold* VariableMonitor를 수정할 때.	충전기는 새 임계값이 이전 임계값을 지우는지 또는 새 임계값이 모니터링된 값에 의해 초과되는지 확인해야 합니다.	
N07.FR.12	설정된 *UpperThreshold* 또는 *LowerThreshold* VariableMonitor를 제거하고 임계값이 초과된 경우.	충전기는 eventData와 함께 NotifyEventRequest를 보내서는 안 됩니다. 속성 *cleared* 가 true입니다.	
N07.FR.13		VariableMonitoring은 재부팅 시에도 지속적으로 저장되어야 합니다.	
N07.FR.14	*UpperThreshold* 또는 *LowerThreshold* 유형의 variableMonitoring 설정이 트리거되고	재부팅이 발생한 후 모니터링된 값이 구성된 임계값 내에서 반환되었습니다.	충전기는 NotifyEventRequest를 eventData와 함께 보내야 하며, 속성 *cleared* 가 true입니다.
N07.FR.15	모니터가 트리거되고 모니터의 *severity* 가 CSMS에서 SetMonitoringLevelRequest에 설정된 모니터링 심각도 수준보다 높은 경우(사용 사례 N05 - Set Monitoring Level 참조)	충전기는 트리거된 모니터에 대해 NotifyEventRequest를 보내서는 안 됩니다.	
N07.FR.16	구성 요소/변수 조합에 UpperThreshold 유형의 모니터가 있음 AND 변수의 실제 값(attributeType Actual)이 *monitorValue* 를 초과하는 경우	충전기는 트리거된 모니터에 대해 *trigger* Alerting과 함께 NotifyEventRequest를 보내야 합니다.	알림은 임계값을 초과할 때 전송되며 임계값에서 전송되지 않습니다.

ID	전제 조건	요구 사항 정의	참고
N07.FR.17	구성 요소/변수 조합에 LowerThreshold 유형의 모니터가 있음 AND 변수의 실제 값(attributeType Actual)이 *monitorValue* 아래로 떨어지면	충전기는 트리거된 모니터에 대해 *trigger* Alerting과 함께 NotifyEventRequest를 보내야 합니다.	알림은 임계값 아래로 떨어지면 전송되고 임계값에서 전송되지는 않습니다.
N07.FR.18	구성 요소/변수 조합에 Delta 유형의 모니터가 있음 AND 변수가 숫자 유형 AND 변수의 실제 값(attributeType Actual)이 이 모니터가 설정된 시간 또는 이 이벤트 알림이 마지막으로 전송된 시간(마지막 시간) 이후로 플러스 또는 마이너스 *monitorValue* 보다 많이 변경된 경우	충전기는 트리거된 모니터에 대해 트리거 Delta와 함께 NotifyEventRequest를 전송해야 합니다.	
N07.FR.19	구성 요소/변수 조합에 Delta 유형의 모니터가 있음 AND 변수가 숫자 유형이 아님 AND 변수의 실제 값(attributeType Actual)이 이 모니터가 설정된 시간 이후 또는 이 이벤트 알림이 마지막으로 전송된 시간 이후로 변경된 경우(참고: 부울, 문자열 또는 열거형과 같이 숫자가 아닌 변수의 경우 Delta 유형의 모니터는 *monitorValue* 값에 관계없이 변수가 변경될 때마다 이벤트 알림을 트리거합니다.)	충전기는 트리거된 모니터에 대해 Delta 트리거와 함께 NotifyEventRequest를 전송해야 합니다.	

N08 - Periodic Event

Table 244. N08 - 주기적 이벤트

번호	유형	설명
1	**이름**	주기적 이벤트
2	**ID**	N08
	기능 블록	N. 진단
3	**목표**	충전기가 모니터링 이벤트에 대해 CSMS에 주기적으로 알릴 수 있는 기능을 제공합니다.
4	**설명**	NotifyEventRequest는 VariableMonitoring 설정이 트리거된 모든 구성 요소/변수를 보고합니다. 이벤트를 트리거하는 데 책임이 있는 VariableMonitoring 설정만 포함됩니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 주기적 값이 초과되면 충전기는 NotifyEventRequest를 트리거 *periodic* 과 함께 CSMS로 보냅니다. **2.** CSMS는 NotifyEventResponse로 응답합니다.
5	**필수 조건**	충전기에 활성 모니터링 설정이 있습니다. 모니터링 설정은 SetVariableMonitoring 메시지를 통해 명시적으로 구성되었거나 충전기 펌웨어에 "하드 와이어링(hard-wired)"되어 있을 수 있습니다.
6	**사후 조건**	충전기가 CSMS에 모니터링 이벤트에 대해 알렸습니다.

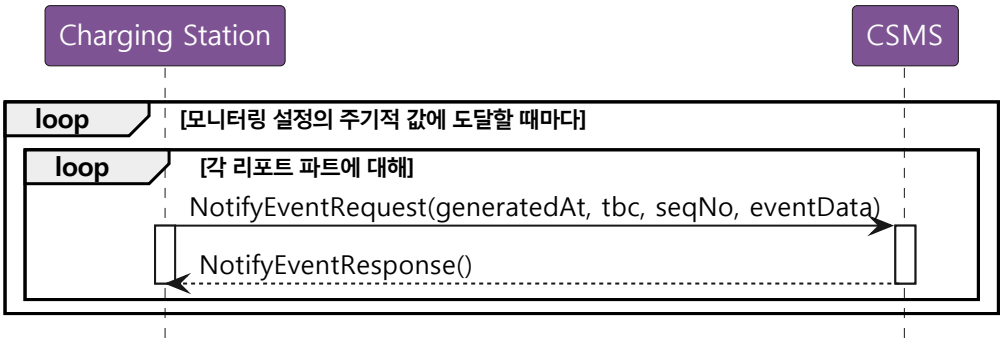

Figure 138. 시퀀스 다이어그램: 주기적 이벤트

7	**오류 처리**	n/a
8	**참고**	n/a

N08 - Periodic Event - Requirements

Table 245. N08 - 요구 사항

ID	전제 조건	요구 사항 정의
N08.FR.02	CSMS가 NotifyEventRequest를 수신하면	CSMS는 빈 NotifyEventResponse로 응답해야 합니다.
N08.FR.03	N08.FR.06 OR N08.FR.07 AND 모니터의 심각도 번호는 구성 변수 `OfflineMonitoringEventQueuingSeverity`에 설정된 심각도 번호와 같거나 낮음 AND 충전기가 *오프라인* 상태입니다.	충전기는 이 NotifyEventRequest를 대기열에 넣고 다시 온라인 상태가 되면 전달해야 합니다.
N08.FR.04	N08.FR.06 또는 N08.FR.07 AND 이 NotifyEventRequest는 첫 번째 또는 유일한 보고 부분입니다.	충전기는 *seqNo* 를 0으로 설정해야 합니다.
N08.FR.05	N08.FR.06 OR N08.FR.07 AND 이벤트를 트리거한 variableMonitoring 설정이 Periodic 또는 PeriodicClockAligned 유형인 경우	충전기는 *trigger* 를 Periodic로 설정해야 합니다.
N08.FR.06	구성 요소/변수 조합에 Periodic 유형의 모니터가 있음 AND *monitorValue* 에 지정된 초 수가 지난 경우(이 모니터가 설정되거나 트리거된 시간부터 시작)	충전기는 트리거된 모니터에 대해 *trigger* Periodic와 함께 NotifyEventRequest를 보내야 합니다.
N08.FR.07	구성 요소/변수 조합에 PeriodicClockAligned 유형의 모니터가 있음 AND 이 모니터가 설정된 후 가장 가까운 클록 정렬 간격(clock-aligned interval)에서 시작하여 *monitorValue* 로 지정된 초 수가 지난 경우(예를 들어, *monitorValue* 가 900이면 매 시간 정각 0, 15, 30 및 45분에 이벤트 알림이 트리거됨)	충전기는 트리거된 모니터에 대해 *trigger* Periodic와 함께 NotifyEventRequest를 보내야 합니다.

2.4. Customer Information

N09 - Get Customer Information

Table 246. N09 - 고객 정보 가져오기

번호	유형	설명
1	**이름**	고객 정보 가져오기
2	**ID**	N09
	기능 블록	N. 진단
3	**목표**	CSMS가 충전기에서 고객 정보 원본(raw customer data)를 검색할 수 있도록 합니다.
4	**설명**	CSMS는 충전기에 메시지를 보내 고객 정보 원본을 검색합니다. 예를 들어, 지역 개인 정보 보호법을 준수하기 위해서입니다. 충전기는 하나 이상의 보고서를 보내 CSMS에 알립니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS는 고객 참조(idToken, customerCertificate 또는 customerIdentifier)와 함께 CustomerInformationRequest를 충전기로 보냅니다. **2.** 충전기는 CustomerInformationResponse로 응답하여 보낼지 여부를 나타냅니다. **3.** 충전기는 하나 이상의 NotifyCustomerInformationRequest 메시지를 CSMS로 보냅니다. **4.** CSMS는 하나 이상의 NotifyCustomerInformationResponse 메시지로 충전기에 응답합니다.
5	**필수 조건**	n/a
6	**사후 조건**	CSMS는 상태가 *Accepted* 인 CustomerInformationResponse 메시지를 *성공적으로* 수신했으며 요청된 데이터를 *성공적으로* 수신했습니다.

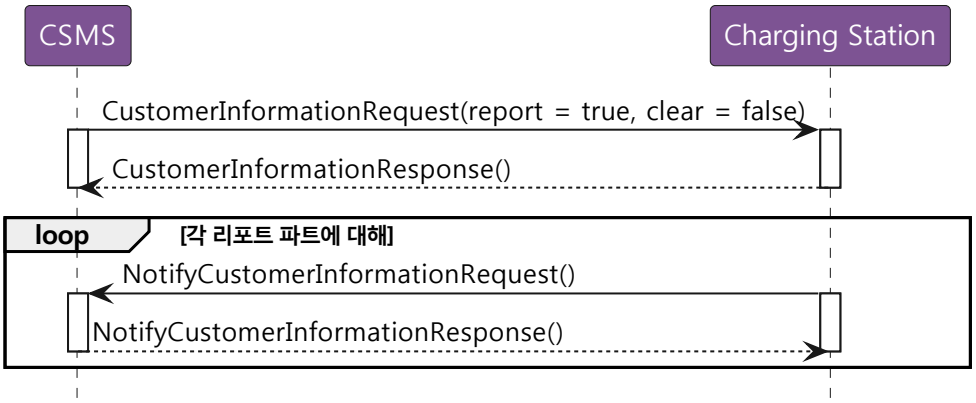

Figure 139. 시퀀스 다이어그램: 고객 정보 가져오기

7	**오류 처리**	n/a
8	**참고**	n/a

N09 - Get Customer Information - Requirements

Table 247. N09 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
N09.FR.01	CSMS가 충전기에서 CustomerInformation을 검색하려는 경우	CustomerInformationRequest의 report 플래그는 *true* 로 설정해야 합니다.	
N09.FR.02	충전기가 CustomerInformationRequest를 수신 AND 이 요청을 처리할 수 있는 상태인 경우	충전기는 CustomerInformationResponse 메시지로 *Accepted* 상태로 응답해야 합니다.	
N09.FR.03	충전기가 이 요청을 처리할 수 없는 상태일 때	CustomerInformationRequest를 수신하면 충전기는 CustomerInformationResponse를 *Rejected* 상태로 응답해야 합니다.	
N09.FR.04		CSMS는 CustomerInformationRequest에 idToken, customerCertificate 또는 customerIdentifier를 포함하여 고객에 대한 참조를 포함해야 합니다.	
N09.FR.05	N09.FR.02 AND 충전기에는 고객 식별자가 참조하는 고객에 대한 정보가 저장되어 있습니다.	충전기는 요청된 정보를 하나 이상의 NotifyCustomerInformationRequest 메시지를 통해 CSMS로 전송해야 합니다.	
N09.FR.06	N09.FR.02 AND 충전기에는 고객 식별자가 참조하는 고객에 대한 정보가 저장되어 있지 않습니다.	충전기는 데이터를 찾을 수 없음을 나타내는 NotifyCustomerInformationRequest 메시지를 CSMS로 전송해야 합니다.	
N09.FR.07	CustomerInformationRequest를 수신할 때 report 플래그와 clear 플래그가 모두 *false* 로 설정되어 있는 경우	CustomerInformationResponse 메시지를 상태 *Rejected* 로 응답하는 것이 좋습니다.	
N09.FR.08	*customerCertificate* 에 따라 사용자 정보를 요청할 때	CSMS는 인증서를 설치하는 데 사용된 *hashAlgorithm* 을 사용해야 합니다.	새 펌웨어를 설치할 때 CSMS는 사용된 *hashAlgorithm* 을 확인하기 위해 먼저 GetInstalledCertificateIdsRequest를 사용하여 인증서를 요청하는 것이 좋습니다.
N09.FR.09	CustomerInformationRequest에 *idToken*, *customerCertificate* 또는 *customerIdentifier* 가 하나도 없거나 CustomerInformationRequest에 *idToken*, *customerCertificate* 또는 *customerIdentifier* 가 두 개 이상 포함된 경우	충전기는 *status* = `Invalid` 로 응답해야 합니다.	*idToken*, *customerCertificate* 또는 *customerIdentifier* 에 대한 값은 하나만 제공될 수 있습니다. 충전기의 N09.FR.04에 해당하는 요구사항입니다.

N10 - Clear Customer Information

Table 248. N10 - 고객 정보 지우기

번호	유형	설명
1	**이름**	고객 정보 지우기
2	**ID**	N10
	기능 블록	N. 진단

번호	유형	설명
3	**목표**	CSMS가 충전기에서 고객 정보 원본을 지우고(검색)할 수 있도록 합니다.
4	**설명**	CSMS는 충전기에 메시지를 보내 고객 정보 원본을 지우고(검색)합니다. 예를 들어, 지역 개인 정보 보호법을 준수하기 위해서입니다. 충전기는 하나 이상의 보고서를 보내 CSMS에 알립니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS는 CustomerInformationRequest를 clear 플래그가 true로 설정된 상태로 충전기로 전송하고 고객(idToken, customerCertificate 또는 customerIdentifier)에 대한 참조를 포함합니다. **2.** 충전기는 CustomerInformationResponse로 응답하여 전송할지 여부를 표시합니다. **3.** report 플래그가 *true* 로 설정된 경우 충전기는 하나 이상의 NotifyCustomerInformationRequest 메시지를 CSMS로 전송합니다. **4.** CSMS는 하나 이상의 NotifyCustomerInformationResponse 메시지로 충전기에 응답합니다.
5	**필수 조건**	n/a
6	**사후 조건**	CSMS는 상태가 *Accepted* 인 CustomerInformationResponse 메시지를 ***성공적으로*** 수신했으며, 충전기는 요청한 대로 고객 정보를 제거했고(report 플래그가 *true* 로 설정된 경우) CSMS는 제거된 데이터를 ***성공적으로*** 수신했습니다.

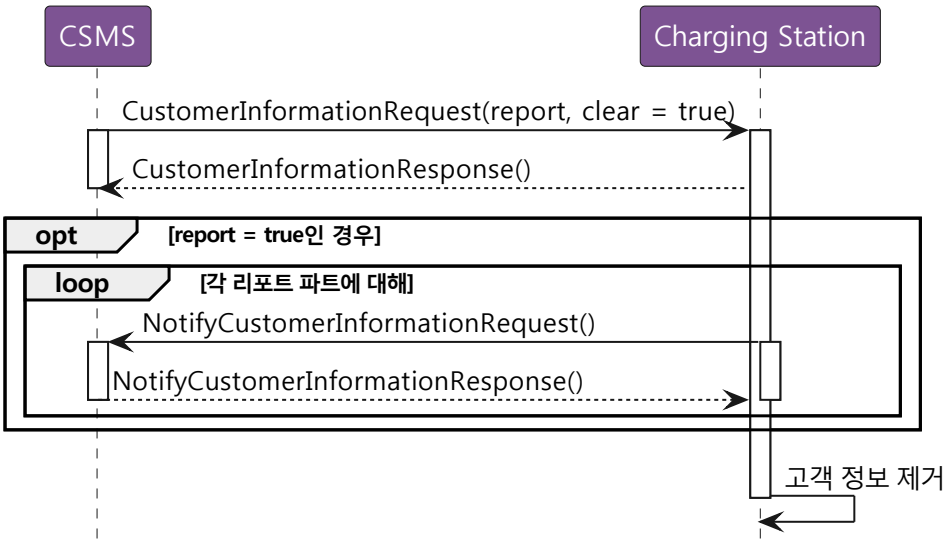

Figure 140. 시퀀스 다이어그램: 고객 정보 지우기

7	**오류 처리**	n/a
8	**참고**	n/a

N10 - Clear Customer Information - Requirements

Table 249. N10 - 요구 사항

ID	전제 조건	요구 사항 정의	참고
N10.FR.01	충전기가 CustomerInformationRequest를 수신 AND 이 요청을 처리할 수 있는 상태에 있는 경우	충전기는 CustomerInformationResponse 메시지로 *Accepted* 상태로 응답해야 합니다.	

ID	전제 조건	요구 사항 정의	참고
N10.FR.02	고객 식별자가 참조하는 고객이 충전기의 Local Authorization List에 있는 경우	CSMS는 SendLocalListRequest를 사용하여 Local Authorization List를 업데이트해야 합니다(D01 - Send Local Authorization List 참조).	Local Authorization List 버전과 관련된 문제를 방지하기 위함입니다
N10.FR.03	N10.FR.01 AND clear 플래그가 *true* 로 설정되고 report 플래그가 *true* 로 설정된 CustomerInformationRequest를 수신 AND 충전기에 고객 식별자가 참조하는 고객에 대한 정보가 저장되어 있는 경우	충전기는 고객 식별자가 참조하는 고객에 대한 모든 고객 관련 데이터를 충전기에서 제거해야 하며, LocalList는 제외해야 하며 충전기는 삭제된 정보를 하나 이상의 NotifyCustomerInformationRequest 메시지를 통해 CSMS로 전송해야 합니다.	LocalList 버전에서 발생하는 문제를 방지하기 위해 CSMS만 LocalList의 내용을 변경할 수 있습니다.
N10.FR.04	N10.FR.01 AND clear 플래그가 *true* 로 설정되고 report 플래그가 *true* 로 설정된 CustomerInformationRequest를 수신 AND 충전기에는 고객 식별자가 참조하는 고객에 대한 정보가 저장되어 있지 않는 경우	충전기는 CSMS에 하나의 NotifyCustomerInformationRequest 메시지를 전송하여 데이터가 발견되지 않았음을 표시해야 합니다.	
N10.FR.05	충전기가 CustomerInformationRequest를 수신하고 이 요청을 처리할 수 없는 상태인 경우	충전기는 CustomerInformationResponse로 응답해야 하며 상태는 *Rejected* 입니다.	
N10.FR.06	N10.FR.01 AND clear 플래그가 *true* 로, report 플래그가 *false* 로 설정된 CustomerInformationRequest를 수신해야 합니다.	충전기는 고객 식별자가 참조하는 고객에 대한 모든 고객 관련 데이터를 충전기에서 제거해야 하며, LocalList는 제외해야 하며 충전기는 데이터가 지워졌음을 나타내는 NotifyCustomerInformationRequest 메시지를 CSMS에 보내야 합니다.	LocalList 버전에서 발생하는 문제를 방지하기 위해 CSMS만 LocalList의 내용을 변경할 수 있습니다.
N10.FR.07	CustomerInformationRequest를 수신할 때 report 플래그와 clear 플래그가 모두 *false* 로 설정되어 있는 경우	CustomerInformationResponse 메시지를 *Rejected* 상태로 응답하는 것이 좋습니다.	
N10.FR.08		CSMS는 idToken, customerCertificate 또는 customerIdentifier를 CustomerInformationRequest에 포함하여 고객에 대한 참조를 포함해야 합니다.	
N10.FR.09	*customerCertificate* 에 따라 사용자 정보를 지울 때	CSMS는 인증서를 설치하는 데 사용된 *hashAlgorithm* 을 사용해야 합니다.	새 펌웨어가 설치되면 CSMS가 사용된 *hashAlgorithm* 을 확인하기 위해 GetInstalledCertificateIdsRequest를 사용하여 먼저 인증서를 요청하는 것이 좋습니다.

O. DisplayMessage

1. Introduction

DisplayMessage 기능을 사용하면 OCPP를 통해 CSO가 충전기 펌웨어에 포함되지 않은 메시지나 메시지 주기를 충전기에 표시할 수 있습니다. CSO는 이러한 메시지를 설정, 검색(가져오기), 변경하거나 지울 수 있습니다.

모든 메시지는 다양한 언어와 다양한 메시지 형식으로 구성할 수 있습니다. DisplayMessageSupportedFormats를 참조하세요. 따라서 충전기는 사용자에게 메시지를 표시해야 할 때 올바른 형식/언어를 선택할 수 있습니다. CSO가 충전기에 보내는 모든 메시지에는 메시지를 표시할 시기와 방법을 제어하는 몇 가지 매개변수가 있습니다. 우선순위, 상태, 시작/종료 시간 등입니다. DisplayMessageSupportedPriorities를 참조하세요.

NOTE	SetDisplayMessageRequest를 통해 구성되지 않은 메시지는 검색/수정할 수 없습니다. (즉, 충전기 펌웨어에 코딩된 메시지는 수정할 수 없습니다.)

2. Use cases & Requirements

O01 - Set DisplayMessage

Table 250. O01 - 디스플레이 메시지 설정

번호	유형	설명
1	**이름**	디스플레이 메시지 설정
2	**ID**	O01
	기능 블록	O. 디스플레이 메시지
3	**목표**	CSO가 펌웨어에 포함되지 않은 추가 메시지를 충전기에 표시할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSO가 충전기에 표시할 메시지를 설정하는 방법을 설명합니다. 주어진 매개변수에 따라 메시지는 특정 방식과 특정 순간에 충전기에 표시됩니다.
	Actors	CSO, CSMS, 충전기
	시나리오 설명	**1.** CSO는 CSMS를 구성하여 새 메시지를 설정하라는 요청을 보냅니다. **2.** CSMS는 SetDisplayMessageRequest 메시지를 충전기로 보냅니다. **3.** 충전기는 SetDisplayMessageResponse 메시지를 CSMS로 보내 요청을 수락합니다. **4.** 충전기는 구성된 순간에 디스플레이에 새 메시지를 표시합니다.
	대체 시나리오	O02 - Set DisplayMessage for Transaction O06 - Replace DisplayMessage
5	**필수 조건**	동일한 ID로 구성된 메시지가 없습니다.
6	**사후 조건**	새 메시지가 충전기에 표시됩니다(시간, 기간 및 위치는 구성에 따라 다름)

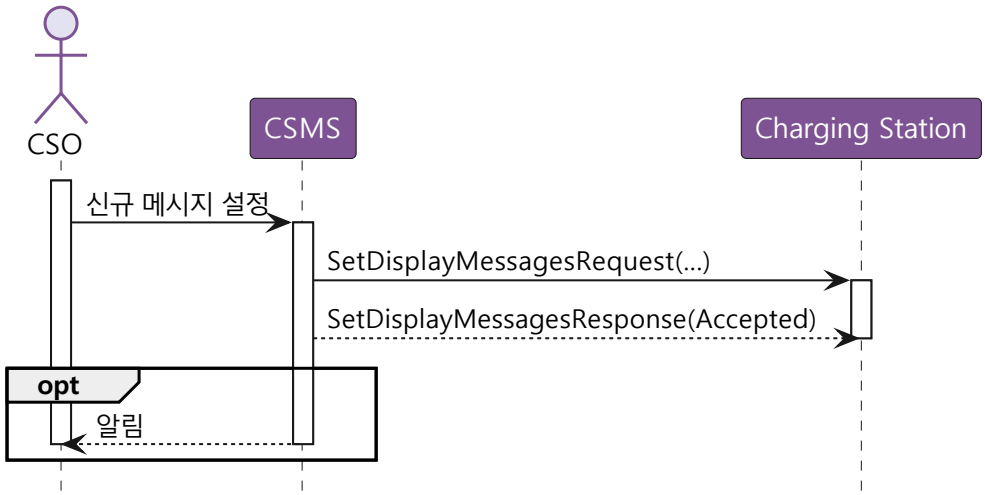

Figure 141. Set DisplayMessage sequence diagram

7	**오류 처리**	n/a
8	**참고**	충전기에 저장할 수 있는 최대 메시지 수는 CSMS에서 구성 변수 `NumberOfDisplayMessages`.maxLimit에서 읽을 수 있습니다.

O01 - Set DisplayMessage - Requirements

Table 251. O01 - DisplayMessage 설정 - 요구 사항

ID	전제 조건	요구 사항 정의
O01.FR.01	충전기가 SetDisplayMessageRequest를 통해 MessageInfo 객체를 수신하고 메시지의 우선순위가 충전기에서 지원되지 않는 경우	충전기는 SetDisplayMessageResponse에 **NotSupportedPriority** status로 전송해야 합니다.
O01.FR.02	충전기가 SetDisplayMessageRequest를 통해 MessageInfo 객체를 수신하고 메시지의 state가 충전기에서 지원되지 않는 경우	충전기는 SetDisplayMessageResponse에 **NotSupportedState** status로 전송해야 합니다.
O01.FR.03	충전기가 SetDisplayMessageRequest를 통해 MessageInfo 객체를 수신하고 메시지의 format이 충전기에서 지원되지 않는 경우	충전기는 SetDisplayMessageResponse에 **NotSupportedMessageFormat** status로 전송해야 합니다.
O01.FR.04		CSMS가 트랜잭션에 속하지 않는 충전기로 메시지를 보내는 경우 Message 필드의 **transactionId** 필드는 생략해야 합니다.
O01.FR.05		CSMS는 메시지를 설정할 때 startTime 및 endTime을 포함할 수 있습니다.
O01.FR.06	O01.FR.05	충전기는 startTime 전에 DisplayMessage 메시지를 표시해서는 안 됩니다.
O01.FR.07	O01.FR.05	충전기는 endTime 뒤에 있는 DisplayMessage 메시지를 제거해야 합니다.
O01.FR.08	충전기가 EV 운전자의 언어 기본 설정을 알고 있는 경우	충전기는 사용 가능한 경우 기본 언어로 DisplayMessage 메시지를 표시해야 합니다.
O01.FR.09	O01.FR.08	일치하는 언어를 사용할 수 없는 경우, 가능한 경우 대체 수단으로 영어로 DisplayMessage 메시지를 표시하는 것을 권장합니다.
O01.FR.10		충전기는 전원 주기/재부팅을 견뎌낼 수 있도록 메시지를 영구 저장소에 저장해야 합니다.
O01.FR.11	충전기가 SetDisplayMessageRequest를 수신하고 이 요청을 처리한 후의 총 메시지 수가 `NumberOfDisplayMessages`.maxLimit을 초과하는 경우.	충전기는 Rejected 상태로 응답해야 합니다.
O01.FR.12	충전기가 SetDisplayMessageRequest를 수신하고 메시지의 priority가 *NormalCycle* 인 경우	충전기는 NormalCycle 메시지중의 구성된 순간에 이 메시지를 표시해야 합니다.
O01.FR.13	충전기가 SetDisplayMessageRequest를 수신하고 메시지의 priority가 *InFront* 인 경우	충전기는 NormalCycle 메시지와 관계없이 구성된 순간에 이 메시지를 표시해야 합니다.
O01.FR.14	priority *InFront* 인 여러 메시지가 동시에 표시되도록 구성된 경우	충전기는 이러한 메시지를 순환해야 합니다.
O01.FR.15	충전기가 SetDisplayMessageRequest를 수신하고 메시지의 priority가 *AlwaysFront* 인 경우	충전기는 다른 설치된 메시지와 관계없이 구성된 순간에 이 메시지를 표시해야 합니다. 따라서 다른 메시지와 순환하지 않으며 충전기 자체의 메시지는 이 메시지를 재정의하지 않습니다.

ID	전제 조건	요구 사항 정의
O01.FR.16	O01.FR.15 AND priority *AlwaysFront* 가 있는 다른 메시지가 이미 설정되어 있음	충전기는 이전 메시지를 새로 설정된 메시지로 대체해야 합니다.
O01.FR.17		언어는 RFC-5646 태그로 지정해야 합니다. 참조: [RFC5646], 예: 미국 영어는 "en-US"입니다.

O02 - Set DisplayMessage for Transaction

Table 252. O02 - Set DisplayMessage for Transaction

번호	유형	설명
1	**이름**	트랜잭션에 대한 DisplayMessage 설정
2	**ID**	O02
	기능 블록	O. DisplayMessage
	부모 사용 사례	O01 - Set DisplayMessage
3	**목표**	CSO가 충전기에서 진행 중인 트랜잭션 중에 펌웨어에 내장되지 않은 메시지를 표시할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSO가 특정 트랜잭션에 대해 충전기에 표시할 메시지를 설정하는 방법을 설명합니다. 제공된 매개변수에 따라 메시지는 충전기에 특정 방식으로 표시됩니다.
	Actors	CSO, CSMS, 충전기
	시나리오 설명	**1.** CSO는 CSMS를 구성하여 주어진 트랜잭션 중에 새 메시지를 표시하라는 요청을 보냅니다. **2.** CSMS는 SetDisplayMessageRequest 메시지를 충전기로 전송합니다. **3.** 충전기는 SetDisplayMessageResponse 메시지를 CSMS로 전송하여 요청을 수락합니다. **4.** 충전기는 트랜잭션이 진행되는 동안 디스플레이에 새 메시지를 표시합니다.
	대체 시나리오	O01 - Set DisplayMessage O06 - Replace DisplayMessage
5	**필수 조건**	동일한 ID로 구성된 메시지가 없습니다.
6	**사후 조건**	트랜잭션이 진행되는 동안 충전기에 새 메시지가 표시됩니다(시간, 기간 및 위치는 구성에 따라 다름)

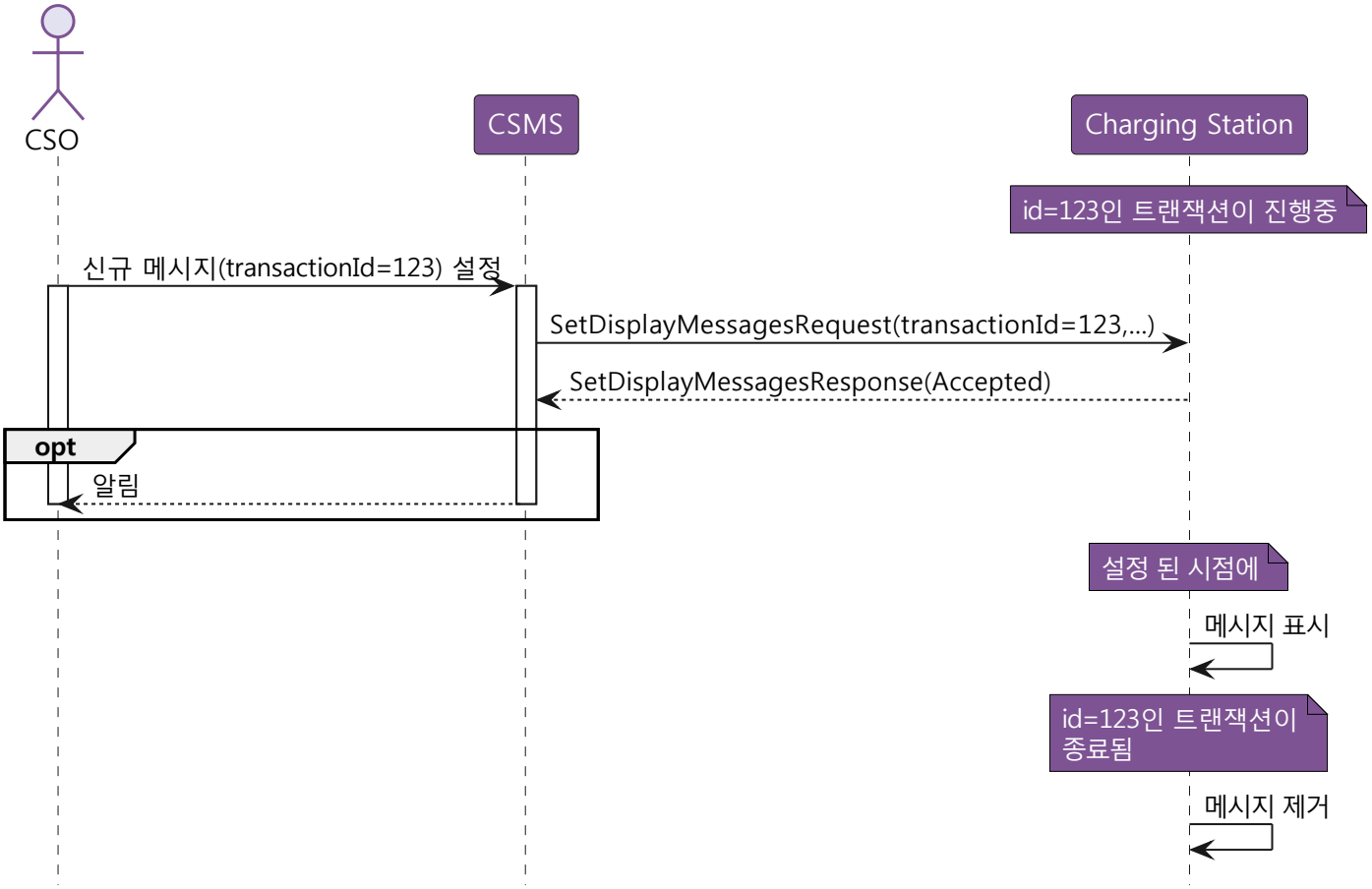

Figure 142. 트랜잭션 순서도에 대한 DisplayMessage 설정

7	**오류 처리**	n/a
8	**참고**	충전기에 저장할 수 있는 최대 메시지 수는 구성 변수 `NumberOfDisplayMessages`.maxLimit에서 CSMS가 읽을 수 있습니다.

O02 - Set DisplayMessage for Transaction - Requirements

Table 253. O02 - Set DisplayMessage for Transaction - Requirements

ID	전제 조건	요구 사항 정의
O02.FR.01	충전기가 SetDisplayMessageRequest를 통해 Message 객체를 수신하고 충전기에서 메시지의 **transactionId** 를 알 수 없는 경우	충전기는 **UnknownTransaction** status와 함께 SetDisplayMessageResponse를 보내야 합니다.
O02.FR.02	지정된 **transactionId** 가 있는 트랜잭션이 종료되는 경우	충전기는 메시지 목록에서 메시지를 제거해야 합니다.
O02.FR.03	충전기가 SetDisplayMessageRequest를 통해 MessageInfo 객체를 수신하고 메시지의 priority가 충전기에서 지원되지 않는 경우	충전기는 **NotSupportedPriority** status와 함께 SetDisplayMessageResponse를 보내야 합니다.
O02.FR.04	충전기가 SetDisplayMessageRequest를 통해 MessageInfo 객체를 수신하고 메시지의 state가 충전기에서 지원되지 않는 경우	충전기는 **NotSupportedState** status와 함께 SetDisplayMessageResponse를 보내야 합니다.

ID	전제 조건	요구 사항 정의
O02.FR.05	충전기가 SetDisplayMessageRequest를 통해 MessageInfo 객체를 수신하고 메시지의 format이 충전기에서 지원되지 않는 경우	충전기는 **NotSupportedMessageFormat** status와 함께 SetDisplayMessageResponse를 보내야 합니다.
O02.FR.06		충전기는 startTime 전에 DisplayMessage 메시지를 표시해서는 안 됩니다.
O02.FR.07		충전기는 endTime 후에 DisplayMessage 메시지를 제거해야 합니다.
O02.FR.08	충전기가 EV 운전자의 언어 기본 설정을 알고 있는 경우	충전기는 가능한 경우 기본 언어로 DisplayMessage 메시지를 표시해야 합니다.
O02.FR.09	O02.FR.08	일치하는 언어를 사용할 수 없는 경우, 가능한 경우 대체 수단으로 영어로 DisplayMessage 메시지를 표시하는 것이 좋습니다.
O02.FR.10		충전기는 메시지를 영구 저장소에 저장하여 충전기의 전원 주기/재부팅 이후에도 유지되어야 합니다.
O02.FR.11	충전기가 SetDisplayMessageRequest를 수신하고 이 요청을 처리한 후의 총 메시지 수가 `NumberOfDisplayMessages`.maxLimit을 초과하는 경우	충전기는 Rejected 상태로 응답해야 합니다.
O02.FR.12		언어는 RFC-5646 태그로 지정해야 합니다. 참조: [RFC5646], 예: 미국 영어는 "en-US"입니다.
O02.FR.14	충전기가 SetDisplayMessageRequest를 수신하고 메시지의 priority가 *NormalCycle* 인 경우	충전기는 NormalCycle 메시지중의 구성된 순간에 이 메시지를 표시해야 합니다.
O02.FR.15	충전기가 SetDisplayMessageRequest를 수신하고 메시지의 priority가 *InFront* 인 경우	충전기는 NormalCycle 메시지와 관계없이 구성된 순간에 이 메시지를 표시해야 합니다.
O02.FR.16	priority가 *InFront* 인 여러 메시지가 동시에 표시되도록 구성된 경우	충전기는 이러한 메시지를 순환해야 합니다.
O02.FR.17	충전기가 SetDisplayMessageRequest를 수신하고 메시지의 priority가 *AlwaysFront* 인 경우	충전기는 다른 설치된 메시지와 관계없이 구성된 순간에 이 메시지를 표시해야 합니다. 따라서 다른 메시지와 함께 순환되지 않으며 충전기 자체의 메시지는 이 메시지를 재정의하지 않습니다.
O02.FR.18	O02.FR.17 AND priority *AlwaysFront* 가 있는 다른 메시지가 이미 설정되어 있습니다.	충전기는 이전 메시지를 새로 설정된 메시지로 대체해야 합니다.

O03 - Get All DisplayMessages

Table 254. O03 - Get All DisplayMessage IDs

번호	유형	설명
1	**이름**	모든 DisplayMessage 가져오기
2	**ID**	O03

번호	유형	설명
	기능 블록	O. DisplayMessage
3	**목표**	CSO가 충전기에 현재 구성된 모든 메시지를 검색할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSO가 충전기에서 OCPP를 통해 구성된 모든 설치된 DisplayMessage를 요청하는 방법을 설명합니다. 충전기는 메시지가 오래되었거나 트랜잭션이 종료되면 메시지를 제거할 수 있습니다. CSO가 현재 메시지 목록을 볼 수 있으면 매우 유용할 수 있으므로 CSO는 어떤 메시지가 (여전히) 구성되어 있는지 알 수 있습니다.
	Actors	CSO, CSMS, 충전기
	시나리오 설명	**1.** CSO가 CSMS에 모든 메시지를 검색하도록 요청합니다. **2.** CSMS가 충전기에 GetDisplayMessagesRequest 메시지를 보냅니다. **3.** 충전기가 GetDisplayMessagesResponse *Accepted* 로 응답하여 메시지를 구성했으며 이를 보낼 것임을 나타냅니다. **4.** 충전기가 CSMS에 하나 이상의 NotifyDisplayMessagesRequest 메시지를 보냅니다(보낼 메시지 양에 따라 다름). **5.** CSMS가 모든 알림에 NotifyDisplayMessagesResponse 메시지로 응답합니다.
5	**필수 조건**	충전기에 구성된 메시지가 하나 이상 있음
6	**사후 조건**	n/a

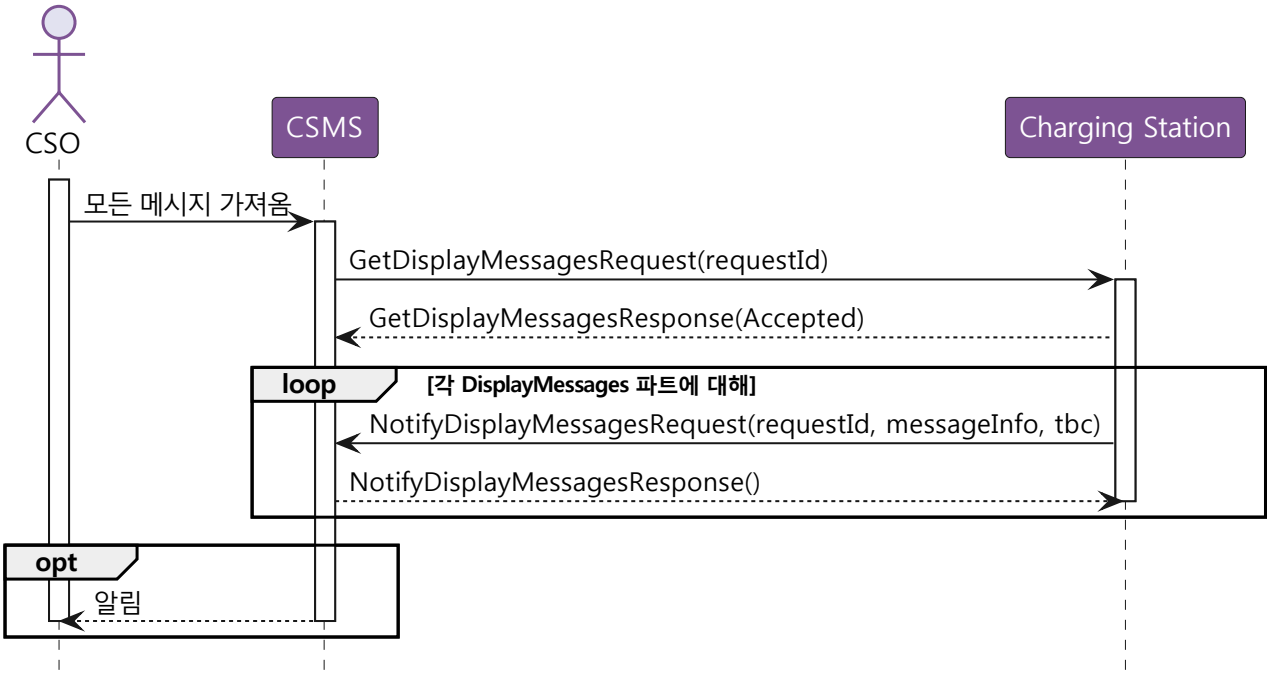

Figure 143. Get All DisplayMessages 시퀀스 다이어그램

7	**오류 처리**	n/a
8	**참고**	OCPP를 통해 구성된 메시지만 GetDisplayMessagesRequest를 통해 검색할 수 있습니다.

O03 - Get All DisplayMessage - Requirements

Table 255. O03 - 모든 DisplayMessage ID 가져오기 - 요구 사항

ID	전제 조건	요구 사항 정의
O03.FR.01	GetDisplayMessagesRequest에서 *requestId* 를 제외한 모든 필드가 생략되고 적어도 하나의 디스플레이 메시지가 구성된 경우.	충전기는 *Accepted* 로 응답해야 합니다.
O03.FR.02	O03.FR.01	충전기는 NotifyDisplayMessagesRequest를 통해 구성된 모든 디스플레이 메시지를 전송해야 합니다.
O03.FR.03	O03.FR.02 AND 충전기가 1개의 NotifyDisplayMessagesRequest로 전송할 수 있는 것보다 많은 디스플레이 메시지가 있는 경우	충전기는 디스플레이 메시지를 여러 개의 NotifyDisplayMessagesRequest 메시지로 분할해야 합니다.
O03.FR.04	O03.FR.03	충전기는 모든 NotifyDisplayMessagesRequest 메시지에서 *tbc* 필드를 *true* 로 설정해야 합니다(마지막 메시지 제외).
O03.FR.05	O03.FR.04	충전기는 *requestId* 필드를 GetDisplayMessagesRequest의 *requestId* 와 동일한 값으로 설정해야 합니다.
O03.FR.06	DisplayMessage가 구성되지 않은 경우	충전기는 *Unknown* 으로 응답해야 합니다.

O04 - Get Specific DisplayMessages

Table 256. O04 - 특정 DisplayMessage 가져오기

번호	유형	설명
1	**이름**	특정 DisplayMessage 가져오기
2	**ID**	O04
	기능 블록	O. DisplayMessage
3	**목표**	CSO가 현재 충전기에 구성된 하나 이상의 특정 DisplayMessage를 검색할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSO가 충전기에서 OCPP를 통해 구성된 (특정) DisplayMessage를 요청/쿼리하는 방법을 설명합니다. 충전기는 메시지가 오래되었거나 트랜잭션이 종료된 경우 메시지를 제거할 수 있습니다. CSO는 (아직) 구성된 메시지를 알 수 있도록 설치된 DisplayMessages에 대해 충전기를 쿼리할 수 있으면 CSO에게 매우 유용할 수 있습니다.
	Actors	CSO, CSMS, 충전기
	시나리오 설명	**1.** CSO가 CSMS에 DisplayMessages를 쿼리하도록 요청합니다. **2.** CSMS가 쿼리 매개변수가 포함된 GetDisplayMessagesRequest 메시지를 충전기로 보냅니다. **3.** 충전기에 요청된 매개변수와 일치하는 DisplayMessages가 있으면 GetDisplayMessagesResponse *Accepted* 로 응답합니다. **4.** 충전기가 하나 이상의 NotifyDisplayMessagesRequest 메시지를 CSMS로 보냅니다(보낼 메시지 양에 따라 다름). **5.** CSMS 응답은 NotifyDisplayMessagesResponse 메시지로 모든 알림을 보냅니다.
5	**필수 조건**	충전기에 구성된 주어진 ID가 있는 메시지가 있습니다.
6	**사후 조건**	n/a

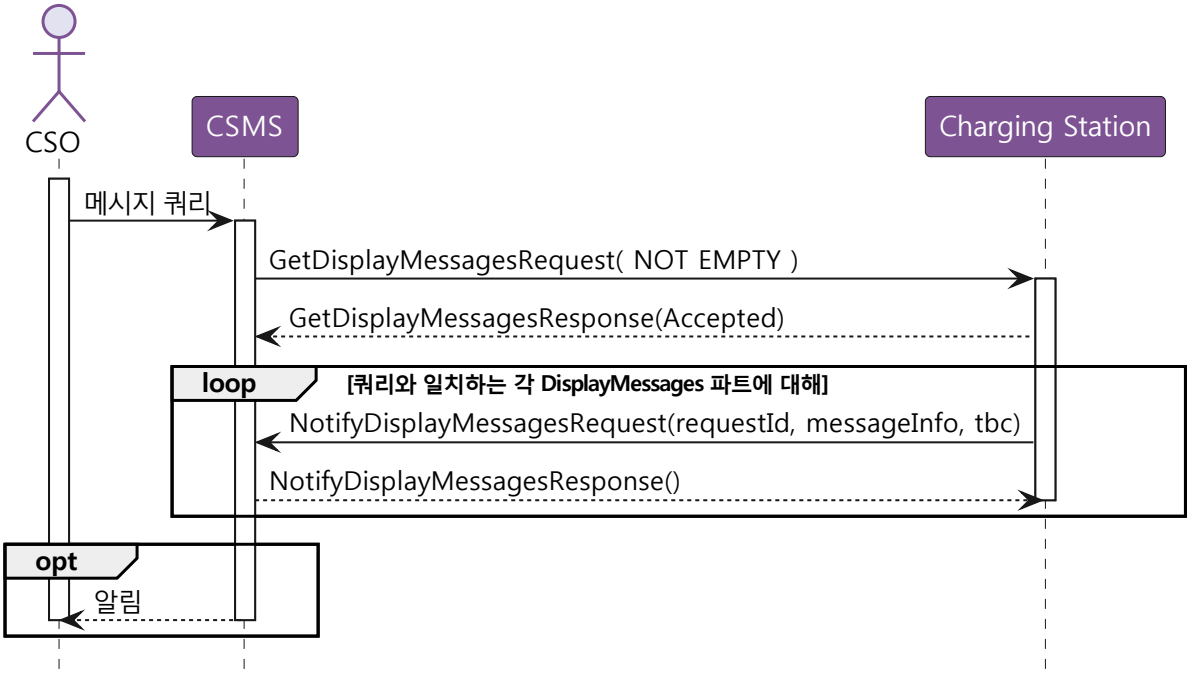

Figure 144. Get a specific DisplayMessages sequence diagram

7	**오류 처리**	n/a

8	참고	OCPP를 통해 구성된 메시지만 GetDisplayMessagesRequest를 통해 검색할 수 있습니다.

O04 - Get Specific DisplayMessages - Requirements

Table 257. O04 - 특정 DisplayMessage 가져오기 - 요구 사항

ID	전제 조건	요구 사항 정의
O04.FR.01	GetDisplayMessagesRequest의 필드 중 하나 이상이 사용되고 충전기에 요청의 매개변수와 일치하는 DisplayMessage가 구성되어 있어야 함	충전기는 *Accepted* 로 응답해야 함.
O04.FR.02	GetDisplayMessagesRequest의 필드 중 하나 이상이 사용되고 충전기에 요청의 매개변수와 일치하는 구성된 DisplayMessage가 없는 경우	충전기는 *Unknown* 으로 응답해야 합니다.
O04.FR.03	O04.FR.01	충전기는 구성된 모든 DisplayMessage를 NotifyDisplayMessagesRequest를 통해 보내야 합니다.
O04.FR.04	O04.FR.03 AND 충전기가 1개의 NotifyDisplayMessagesRequest에서 보낼 수 있는 것보다 많은 DisplayMessage가 있는 경우	충전기는 DisplayMessage를 여러 개의 NotifyDisplayMessagesRequest 메시지로 분할해야 합니다.
O04.FR.05	O04.FR.04	충전기는 마지막 메시지를 제외한 모든 NotifyDisplayMessagesRequest 메시지에서 *tbc* 필드를 *true* 로 설정해야 합니다.
O04.FR.06	O04.FR.05	충전기는 *requestId* 필드를 GetDisplayMessagesRequest의 *requestId* 와 동일한 값으로 설정해야 합니다.
O04.FR.07	DisplayMessage가 구성되지 않은 경우	충전기는 *Unknown* 으로 응답해야 합니다.

O05 - Clear DisplayMessage

Table 258. O05 - DisplayMessage 지우기

번호	유형	설명
1	**이름**	DisplayMessage 지우기
2	**ID**	O05
	기능 블록	O. DisplayMessage
3	**목표**	CSO가 현재 충전기에 구성된 특정 메시지를 제거할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSO가 충전기에서 OCPP를 통해 구성된 특정 메시지를 제거하는 방법을 설명합니다.
	Actors	CSO, CSMS, 충전기
	시나리오 설명	**1.** CSO가 CSMS에 특정 메시지를 제거하도록 요청합니다. **2.** CSMS가 특정 메시지의 ID가 포함된 ClearDisplayMessageRequest 메시지를 충전기로 보냅니다. **3.** 충전기가 메시지를 제거합니다. **4.** 충전기가 ClearDisplayMessageResponse 메시지를 CSMS로 보내 응답합니다.
5	**필수 조건**	충전기에 구성된 지정된 ID의 메시지가 있습니다.
6	**사후 조건**	지정된 ID의 메시지가 충전기에서 제거됩니다.

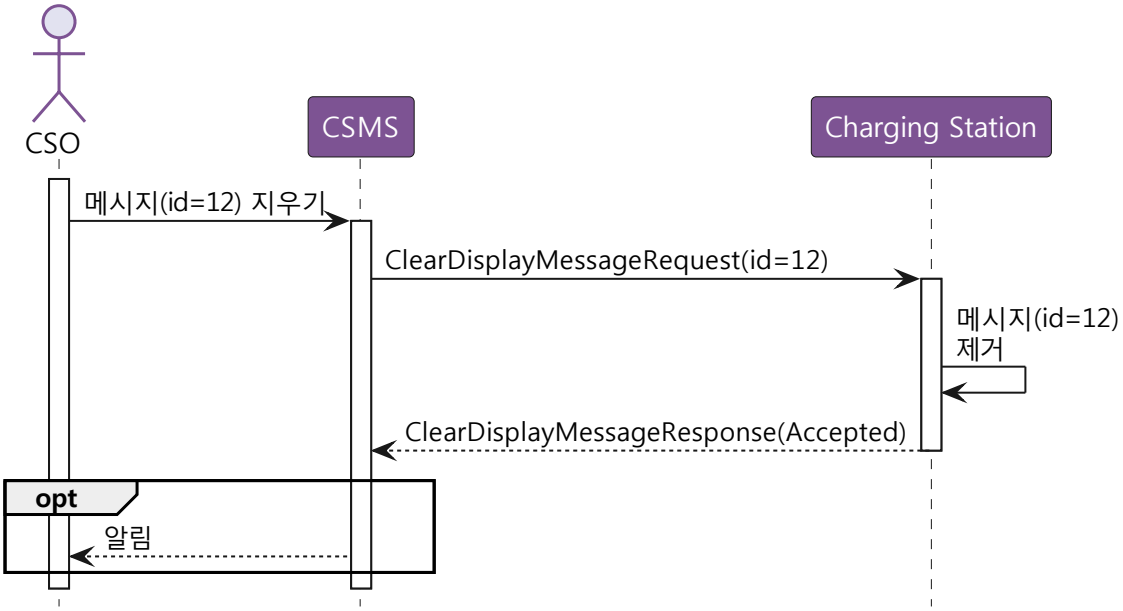

Figure 145. Clear a DisplayMessage sequence diagram

7	**오류 처리**	n/a
8	**참고**	OCPP를 통해 구성된 메시지만 ClearDisplayMessageRequest를 통해 지우거나 제거할 수 있습니다.

O05 - Clear DisplayMessage - Requirements

Table 259. O05 - DisplayMessage 지우기 - 요구 사항

ID	전제 조건	요구 사항 정의
O05.FR.01	충전기가 ClearDisplayMessageRequest를 수신하고 해당 ID로 충전기에 구성된 메시지가 있는 경우	충전기는 ClearDisplayMessageResponse 메시지로 응답해야 하며 상태는 *Accepted* 입니다.
O05.FR.02	충전기가 ClearDisplayMessageRequest를 수신하고 해당 ID로 구성된 메시지가 충전기에 없는 경우	충전기는 ClearDisplayMessageResponse 메시지로 응답해야 하며 상태는 *Unknown* 입니다.

O06 - Replace DisplayMessage

Table 260. O06 - DisplayMessage 교체

번호	유형	설명
1	**이름**	DisplayMessage 교체
2	**ID**	O06
	기능 블록	O. DisplayMessage
3	**목표**	CSO가 충전기에 이미 구성된 DisplayMessage를 대체할 수 있도록 합니다.
4	**설명**	이 사용 사례는 CSO가 충전기에 이전에 구성된 DisplayMessage를 대체하는 방법을 설명합니다. 메시지 내용과 제공된 모든 매개변수를 새 메시지로 대체합니다.
	Actors	CSO, CSMS, 충전기
	시나리오 설명	**1.** CSO가 CSMS에 기존 DisplayMessage를 대체하도록 요청합니다. **2.** CSMS가 충전기에 이미 구성된 것과 동일한 ID를 가진 DisplayMessage와 함께 SetDisplayMessageRequest 메시지를 충전기로 보냅니다. **3.** 충전기가 CSMS에 SetDisplayMessageResponse 메시지를 보내 요청을 수락합니다. **4.** 충전기는 구성된 순간에 디스플레이에 업데이트/대체된 메시지를 표시합니다.
	대체 시나리오	O01 - Set DisplayMessage 및 O02 - Set DisplayMessage for Transaction
5	**필수 조건**	충전기에 동일한 ID의 메시지가 있습니다.
6	**사후 조건**	디스플레이 메시지는 동일한 ID가 제공된 메시지로 대체됩니다.

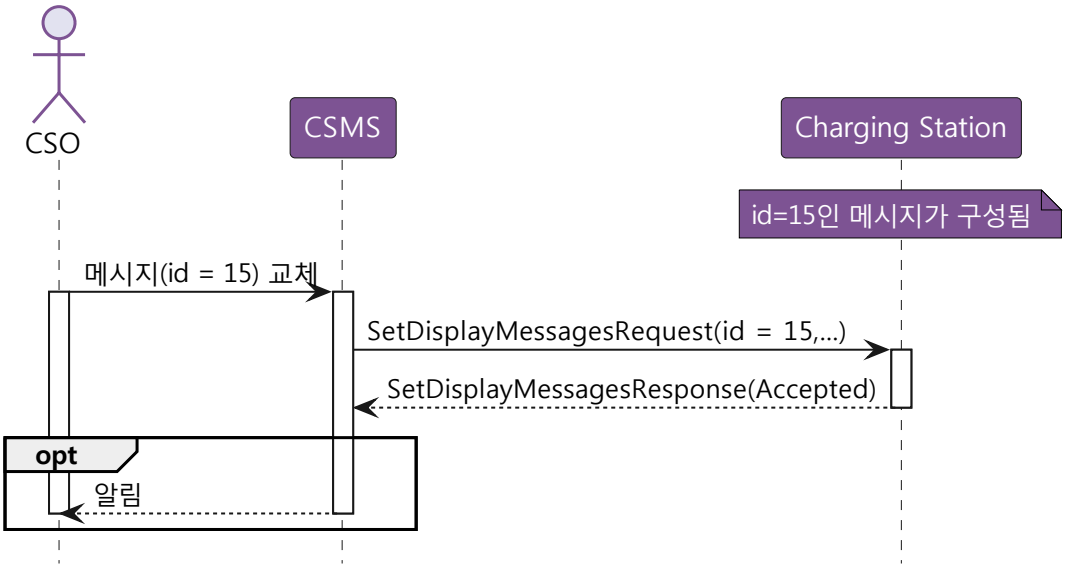

Figure 146. DisplayMessage 시퀀스 다이어그램 대체

7	**오류 처리**	n/a
8	**참고**	n/a

O06 - Replace DisplayMessage - Requirements

Table 261. O06 - DisplayMessage 대체 - 요구 사항

ID	전제 조건	요구 사항 정의
O06.FR.01	충전기에서 SetDisplayMessageRequest를 수신하고 충전기에 동일한 ID를 가진 메시지가 있습니다.	충전기는 기존 메시지를 새 메시지(모든 새 매개변수 포함)로 대체하고 SetDisplayMessageResponse 메시지로 응답해야 하며, 이 메시지에 대해 상태: *Accepted* 를 지정해야 합니다.

P. DataTransfer

1. Introduction

이 기능 블록은 당사자가 기존 명령을 사용자 정의 속성으로 확장하거나 OCPP에 새 사용자 정의 명령을 추가할 수 있는 기능을 설명합니다.

OCPP는 공급업체별 사용자 정의 확장을 만드는 두 가지 메커니즘을 제공합니다.

1. DataTransferRequest 메시지는 OCPP에서 표준화되지 않은 데이터 또는 메시지를 교환할 수 있도록 합니다. 따라서 OCPP 내에서 실험적 기능을 위한 프레임워크를 제공하며, 이는 향후 OCPP 버전에 도입될 수 있습니다. 새(호환되지 않을 수 있는) OCPP 변형을 만들지 않고도 실험할 수 있습니다. 둘째, 특정 CSMS 및 충전기 공급업체 간에 합의된 추가 기능을 구현할 수 있는 가능성을 제공합니다.
2. CustomData 요소는 모든 유형의 JSON 스키마에서 선택적 요소로 존재합니다. CustomData는 JSON 스키마 파일에서 추가 속성을 허용하는 유일한 클래스입니다. 따라서 모든 유형에 추가 사용자 정의 속성을 추가하는 데 사용할 수 있습니다. CustomData는 사양 문서에서 의도적으로 제외되었습니다. 많은 혼란을 야기하고 표준 구현에서 사용하도록 의도된 것이 아니기 때문입니다. [OCPP2.0-PART4]도 참조하세요.

DataTransferRequest/Response에는 길이 또는 유형 사양이 없는 필드가 포함되어 있습니다. 이 필드를 구조화된 JSON 콘텐츠로 사용하면 편리할 수 있습니다.

임베디드 JSON의 예

```
[2,
"<unique msg id>",
"DataTransfer",
{
  "vendorId": "com.mycompany.ice",
  "messageId": "iceParkedAtCs"
  "data": { "start_time": "2020-04-01T11:01:02" }
}
]
```

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

2. Use cases & Requirements

P01 - Data Transfer to the Charging Station

Table 262. P01 - 충전기로의 데이터 전송

번호	유형	설명
1	**이름**	충전기로의 데이터 전송
2	**ID**	P01
	기능 블록	P. 데이터 전송
3	**목표**	OCPP에서 지원하지 않는 기능에 대한 정보를 CSMS에서 충전기로 전송합니다.
4	**설명**	이 사용 사례는 CSMS에서 충전기로 DataTransfer 메시지를 전송하는 기능을 다룹니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** CSMS는 DataTransferRequest를 사용하여 OCPP에서 지원하지 않는 기능에 대한 정보를 충전기로 전송합니다. **2.** 충전기는 DataTransferResponse를 사용하여 CSMS에 응답합니다.
5	**필수 조건**	해당 없음
6	**사후 조건**	**성공한 사후 조건:** DataTransferRequest가 *성공적으로* 수신되고 *Accepted* 됨 **실패 사후 조건:** 메시지가 *Accepted* 되었지만 포함된 요청은 *Rejected* 됩니다. 다른 모든 경우 *Accepted* 또는 *Rejected* 상태와 데이터 요소의 사용은 관련 당사자 간의 공급업체별 계약의 일부입니다.

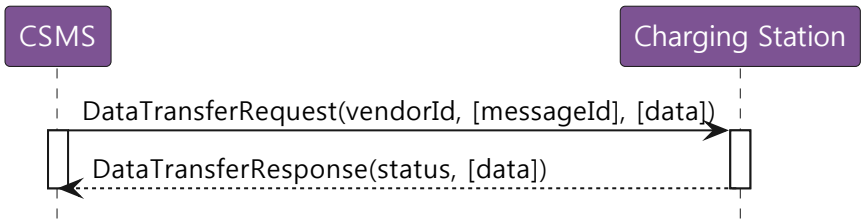

Figure 147. 시퀀스 다이어그램: 충전기로의 데이터 전송

7	**오류 처리**	해당 없음
8	**참고**	데이터 전송은 OCPP에서 함수에 대한 정보를 지원하지 않는 경우에 사용됩니다. 요청 및 응답 메시지의 데이터 길이는 정의되지 않으며 모든 관련 당사자가 이에 동의하는 것이 좋습니다.

P01 - Data Transfer to the Charging Station - Requirements

Table 263. P01 - 요구 사항

ID	전제 조건	요구 사항 정의
P01.FR.01		충전기는 OCPP에서 지원하지 않는 함수에 대해서만 DataTransferRequest를 사용해야 합니다.
P01.FR.02		vendorId는 역순 DNS 네임스페이스의 값이어야 하며, 이름의 최상위 계층은 역순으로 공급업체 조직의 공개적으로 등록된 기본 DNS 이름과 일치해야 합니다.
P01.FR.03		요청 메시지의 messageId는 특정 메시지 또는 구현을 나타내는 데 사용될 수 있습니다.
P01.FR.04		요청 및 응답 메시지의 데이터 길이는 정의되지 않았으며 관련 당사자 모두가 이에 동의하는 것이 좋습니다.
P01.FR.05	요청 수신자가 특정 vendorId에 대한 구현이 없는 경우.	수신자는 상태 *UnknownVendor* 를 반환해야 합니다.
P01.FR.06	DataTransferRequest를 수신하고 messageId가 일치하지 않는 경우(사용된 경우).	수신자는 상태 *UnknownMessageId* 를 반환해야 합니다.
P01.FR.07		상태 *Accepted* 또는 *Rejected* 및 데이터 요소의 사용은 관련 당사자 간의 공급업체별 계약의 일부가 되어야 합니다.

P02 - Data Transfer to the CSMS

Table 264. P02 - Data Transfer to the CSMS

번호	유형	설명
1	**이름**	CSMS로의 데이터 전송
2	**ID**	P02
	기능 블록	P. 데이터 전송
3	**목표**	OCPP에서 지원하지 않는 기능에 대한 정보를 충전기에서 CSMS로 전송합니다.
4	**설명**	이 사용 사례는 충전기에서 CSMS로 DataTransfer 메시지를 전송하는 기능을 다룹니다.
	Actors	충전기, CSMS
	시나리오 설명	**1.** 충전기는 DataTransferRequest를 사용하여 OCPP에서 지원하지 않는 기능에 대한 정보를 CSMS로 전송합니다. **2.** CSMS는 DataTransferResponse를 사용하여 충전기에 응답합니다.
5	**필수 조건**	해당 없음
6	**사후 조건**	**성공한 사후 조건:** DataTransferRequest가 *성공적으로* 수신되고 *Accepted* 됨 **실패 사후 조건:** 메시지가 수락되었지만 포함된 요청은 *Rejected* 됩니다. 다른 모든 경우 *Accepted* 또는 *Rejected* 상태와 데이터 요소의 사용은 관련 당사자 간의 공급업체별 계약의 일부입니다.

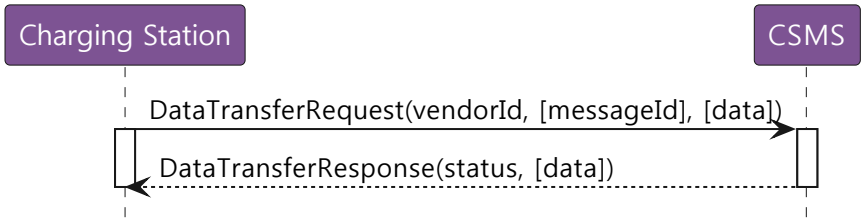

Figure 148. 시퀀스 다이어그램: CSMS로의 데이터 전송

7	**오류 처리**	해당 없음
8	**참고**	데이터 전송은 함수에 대한 정보가 OCPP에서 지원되지 *않는* 경우 사용됩니다. 요청 및 응답 메시지의 데이터 길이는 정의되지 않으며 관련된 모든 당사자가 합의해야 합니다.

P02 - Data Transfer to the CSMS - Requirements

Table 265. P02 - 요구 사항

ID	전제 조건	요구 사항 정의
P02.FR.01		요청 메시지의 vendorId는 충전기에서 알아야 하며 공급업체별 구현을 고유하게 식별해야 합니다.
P02.FR.02		충전기는 OCPP에서 지원하지 않는 함수에 대해서만 DataTransferRequest를 사용해야 합니다.
P02.FR.03		VendorId는 역순 DNS 네임스페이스의 값이어야 하며, 이름의 최상위 계층은 역순으로 변경되면 Vendor 조직의 공개적으로 등록된 기본 DNS 이름과 일치해야 합니다.
P02.FR.04		요청 메시지의 messageId는 특정 메시지 또는 구현을 나타내는 데 사용될 수 있습니다.
P02.FR.05		요청 및 응답 메시지의 데이터 길이는 정의되지 않으며 관련된 모든 당사자가 이에 동의하는 것이 좋습니다.
P02.FR.06	요청 수신자가 특정 vendorId에 대한 구현이 없는 경우.	수신자는 *UnknownVendor* 상태를 반환해야 합니다.
P02.FR.07	DataTransferRequest를 수신하고 messageId가 일치하지 않는 경우(사용된 경우).	수신자는 UnknownMessageId 상태를 반환해야 합니다.
P02.FR.08		*Accepted* 또는 *Rejected* 상태와 데이터 요소의 사용은 관련 당사자 간의 공급업체별 계약의 일부가 되어야 합니다.

Messages, Datatypes & Enumerations

1. Messages

1.1. Authorize

1.1.1. AuthorizeRequest

This contains the field definition of the AuthorizeRequest PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
certificate	string[0..5500]	0..1	Optional. The X.509 certificate chain presented by EV and encoded in PEM format. Order of certificates in chain is from leaf up to (but excluding) root certificate. Only needed in case of central contract validation when Charging Station cannot validate the contract certificate.
idToken	IdTokenType	1..1	Required. This contains the identifier that needs to be authorized.
iso15118CertificateHashData	OCSPRequestDataType	0..4	Optional. Contains the information needed to verify the EV Contract Certificate via OCSP. Not needed if certificate is provided.

1.1.2. AuthorizeResponse

This contains the field definition of the AuthorizeResponse PDU sent by the CSMS to the Charging Station in response to an AuthorizeRequest.

Class

Field Name	Field Type	Card.	Description
certificateStatus	AuthorizeCertificateStatusEnumType	0..1	Optional. Certificate status information. - if all certificates are valid: return 'Accepted'. - if one of the certificates was revoked, return 'CertificateRevoked'.
idTokenInfo	IdTokenInfoType	1..1	Required. This contains information about authorization status, expiry and group id.

1.2. BootNotification

1.2.1. BootNotificationRequest

This contains the field definition of the BootNotificationRequest PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
reason	BootReasonEnumType	1..1	Required. This contains the reason for sending this message to the CSMS.
chargingStation	ChargingStationType	1..1	Required. Identifies the Charging Station

1.2.2. BootNotificationResponse

This contains the field definition of the BootNotificationResponse PDU sent by the CSMS to the Charging Station in response to a BootNotificationRequest.

Class

Field Name	Field Type	Card.	Description
currentTime	dateTime	1..1	Required. This contains the CSMS's current time.
interval	integer	1..1	Required. When Status is Accepted, this contains the heartbeat interval in seconds. If the CSMS returns something other than Accepted, the value of the interval field indicates the minimum wait time before sending a next BootNotification request.
status	RegistrationStatusEnumType	1..1	Required. This contains whether the Charging Station has been registered within the CSMS.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.3. CancelReservation

1.3.1. CancelReservationRequest

This contains the field definition of the CancelReservationRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
reservationId	integer	1..1	Required. Id of the reservation to cancel.

1.3.2. CancelReservationResponse

This contains the field definition of the CancelReservationResponse PDU sent by the Charging Station to the CSMS in response to a CancelReservationRequest.

Class

Field Name	Field Type	Card.	Description
status	CancelReservationStatusEnumType	1..1	Required. This indicates the success or failure of the canceling of a reservation by CSMS.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.4. CertificateSigned

1.4.1. CertificateSignedRequest

This contains the field definition of the CertificateSignedRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
certificateChain	string[0..10000]	1..1	Required. The signed PEM encoded X.509 certificate. This SHALL also contain the necessary sub CA certificates, when applicable. The order of the bundle follows the certificate chain, starting from the leaf certificate. The Configuration Variable MaxCertificateChainSize can be used to limit the maximum size of this field.
certificateType	CertificateSigningUseEnumType	0..1	Optional. Indicates the type of the signed certificate that is returned. When omitted the certificate is used for both the 15118 connection (if implemented) and the Charging Station to CSMS connection. This field is required when a certificateType was included in the SignCertificateRequest that requested this certificate to be signed AND both the 15118 connection and the Charging Station connection are implemented.

1.4.2. CertificateSignedResponse

This contains the field definition of the CertificateSignedResponse PDU sent by the Charging Station to the CSMS in response to a CertificateSignedRequest.

Class

Field Name	Field Type	Card.	Description
status	CertificateSignedStatusEnumType	1..1	Required. Returns whether certificate signing has been accepted, otherwise rejected.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.5. ChangeAvailability

1.5.1. ChangeAvailabilityRequest

This contains the field definition of the ChangeAvailabilityRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
operationalStatus	OperationalStatusEnumType	1..1	Required. This contains the type of availability change that the Charging Station should perform.
evse	EVSEType	0..1	Optional. Contains Id's to designate a specific EVSE/connector by index numbers. When omitted, the message refers to the Charging Station as a whole.

1.5.2. ChangeAvailabilityResponse

This contains the field definition of the ChangeAvailabilityResponse PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
status	ChangeAvailabilityStatusEnumType	1..1	Required. This indicates whether the Charging Station is able to perform the availability change.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.6. ClearCache

1.6.1. ClearCacheRequest

This contains the field definition of the ClearCacheRequest PDU sent by the CSMS to the Charging Station. No fields are defined.

1.6.2. ClearCacheResponse

This contains the field definition of the ClearCacheResponse PDU sent by the Charging Station to the CSMS in response to a ClearCacheRequest.

Class

Field Name	Field Type	Card.	Description
status	ClearCacheStatusEnumType	1..1	Required. Accepted if the Charging Station has executed the request, otherwise rejected.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.7. ClearChargingProfile

1.7.1. ClearChargingProfileRequest

This contains the field definition of the ClearChargingProfileRequest PDU sent by the CSMS to the Charging Station. The CSMS can use this message to clear (remove) either a specific charging profile (denoted by id) or a selection of charging profiles that match with the values of the optional evse, stackLevel and ChargingProfilePurpose fields.

Class

Field Name	Field Type	Card.	Description
chargingProfileId	integer	0..1	Optional. The Id of the charging profile to clear.
chargingProfileCriteria	ClearChargingProfileType	0..1	Optional. Specifies the charging profile.

1.7.2. ClearChargingProfileResponse

This contains the field definition of the ClearChargingProfileResponse PDU sent by the Charging Station to the CSMS in response to a ClearChargingProfileRequest.

Class

Field Name	Field Type	Card.	Description
status	ClearChargingProfileStatusEnumType	1..1	Required. Indicates if the Charging Station was able to execute the request.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.8. ClearDisplayMessage

1.8.1. ClearDisplayMessageRequest

This contains the field definition of the ClearDisplayMessageRequest PDU sent by the CSMS to the Charging Station. The CSMS asks the Charging Station to clear a display message that has been configured in the Charging Station to be cleared/removed. See also O05 - Clear a Display Message.

Class

Field Name	Field Type	Card.	Description
id	integer	1..1	Required. Id of the message that SHALL be removed from the Charging Station.

1.8.2. ClearDisplayMessageResponse

This contains the field definition of the ClearDisplayMessageResponse PDU sent by the Charging Station to the CSMS in a response to a ClearDisplayMessageRequest. See also O05 - Clear a Display Message.

Class

Field Name	Field Type	Card.	Description
status	ClearMessageStatusEnumType	1..1	Required. Returns whether the Charging Station has been able to remove the message.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.9. ClearedChargingLimit

1.9.1. ClearedChargingLimitRequest

This contains the field definition of the ClearedChargingLimitRequest PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
chargingLimitSource	ChargingLimitSourceEnumType	1..1	Required. Source of the charging limit.
evseId	integer	0..1	Optional. EVSE Identifier.

1.9.2. ClearedChargingLimitResponse

This contains the field definition of the ClearedChargingLimitResponse PDU sent by the CSMS to the Charging Station. No fields are defined.

1.10. ClearVariableMonitoring

1.10.1. ClearVariableMonitoringRequest

This contains the field definition of the ClearVariableMonitoringRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
id	integer	1..*	Required. List of the monitors to be cleared, identified by there Id.

1.10.2. ClearVariableMonitoringResponse

This contains the field definition of the ClearVariableMonitoringResponse PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
clearMonitoringResult	ClearMonitoringResultType	1..*	Required. List of result statuses per monitor.

1.11. CostUpdated

1.11.1. CostUpdatedRequest

This contains the field definition of the CostUpdatedRequest PDU sent by the CSMS to the Charging Station. With this request the CSMS can send the current cost of a transaction to a Charging Station.

Class

Field Name	Field Type	Card.	Description
totalCost	decimal	1..1	Required. Current total cost, based on the information known by the CSMS, of the transaction including taxes. In the currency configured with the configuration Variable: [Currency]
transactionId	identifierString[0..36]	1..1	Required. Transaction Id of the transaction the current cost are asked for.

1.11.2. CostUpdatedResponse

This contains the field definition of the CostUpdatedResponse PDU sent by the Charging Station to the CSMS in response to CostUpdatedRequest. No fields are defined.

1.12. CustomerInformation

This contains the field definition of the CustomerInformationRequest PDU sent by the CSMS to the Charging Station.

1.12.1. CustomerInformationRequest

Class

Field Name	Field Type	Card.	Description
requestId	integer	1..1	Required. The Id of the request.
report	boolean	1..1	Required. Flag indicating whether the Charging Station should return NotifyCustomerInformationRequest messages containing information about the customer referred to.
clear	boolean	1..1	Required. Flag indicating whether the Charging Station should clear all information about the customer referred to.
customerIdentifier	string[0..64]	0..1	Optional. A (e.g. vendor specific) identifier of the customer this request refers to. This field contains a custom identifier other than IdToken and Certificate. One of the possible identifiers (customerIdentifier, customerIdToken or customerCertificate) should be in the request message.

Field Name	Field Type	Card.	Description
idToken	IdTokenType	0..1	Optional. The IdToken of the customer this request refers to. One of the possible identifiers (customerIdentifier, customerIdToken or customerCertificate) should be in the request message.
customerCertificate	CertificateHashDataType	0..1	Optional. The Certificate of the customer this request refers to. One of the possible identifiers (customerIdentifier, customerIdToken or customerCertificate) should be in the request message.

1.12.2. CustomerInformationResponse

Class

Field Name	Field Type	Card.	Description
status	CustomerInformationStatusEnumType	1..1	Required. Indicates whether the request was accepted.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.13. DataTransfer

1.13.1. DataTransferRequest

This contains the field definition of the DataTransferRequest PDU sent either by the CSMS to the Charging Station or vice versa.

Class

Field Name	Field Type	Card.	Description
messageId	string[0..50]	0..1	Optional. May be used to indicate a specific message or implementation.
data	anyType	0..1	Optional. Data without specified length or format. This needs to be decided by both parties (Open to implementation).
vendorId	string[0..255]	1..1	Required. This identifies the Vendor specific implementation

1.13.2. DataTransferResponse

This contains the field definition of the DataTransferResponse PDU sent by the Charging Station to the CSMS or vice versa in response to a DataTransferRequest.

Class

Field Name	Field Type	Card.	Description
status	DataTransferStatusEnumType	1..1	Required. This indicates the success or failure of the data transfer.
data	anyType	0..1	Optional. Data without specified length or format, in response to request.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.14. DeleteCertificate

1.14.1. DeleteCertificateRequest

Used by the CSMS to request deletion of an installed certificate on a Charging Station.

Class

Field Name	Field Type	Card.	Description
certificateHashData	CertificateHashDataType	1..1	Required. Indicates the certificate of which deletion is requested.

1.14.2. DeleteCertificateResponse

Response to a DeleteCertificateRequest.

Class

Field Name	Field Type	Card.	Description
status	DeleteCertificateStatusEnumType	1..1	Required. Charging Station indicates if it can process the request.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.15. FirmwareStatusNotification

1.15.1. FirmwareStatusNotificationRequest

This contains the field definition of the FirmwareStatusNotificationRequest PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
status	FirmwareStatusEnumType	1..1	Required. This contains the progress status of the firmware installation.

Field Name	Field Type	Card.	Description
requestId	integer	0..1	Optional. The request id that was provided in the UpdateFirmwareRequest that started this firmware update. This field is mandatory, unless the message was triggered by a TriggerMessageRequest AND there is no firmware update ongoing.

1.15.2. FirmwareStatusNotificationResponse

This contains the field definition of the FirmwareStatusNotificationResponse PDU sent by the CSMS to the Charging Station in response to a FirmwareStatusNotificationRequest. No fields are defined.

1.16. Get15118EVCertificate

1.16.1. Get15118EVCertificateRequest

This message is sent by the Charging Station to the CSMS if an ISO 15118 vehicle selects the service Certificate installation. NOTE: This message is based on CertificateInstallationReq Res from ISO 15118-2.

Class

Field Name	Field Type	Card.	Description
iso15118SchemaVersion	string[0..50]	1..1	Required. Schema version currently used for the 15118 session between EV and Charging Station. Needed for parsing of the EXI stream by the CSMS.
action	CertificateActionEnumType	1..1	Required. Defines whether certificate needs to be installed or updated.
exiRequest	string[0..5600]	1..1	Required. Raw CertificateInstallationReq request from EV, Base64 encoded.

1.16.2. Get15118EVCertificateResponse

Response message from CSMS to Charging Station containing the status and optionally new certificate. NOTE: This message is based on CertificateInstallationReq Res from ISO 15118-2.

Class

Field Name	Field Type	Card.	Description
status	Iso15118EVCertificateStatusEnumType	1..1	Required. Indicates whether the message was processed properly.
exiResponse	string[0..5600]	1..1	Required. Raw CertificateInstallationRes response for the EV, Base64 encoded. The Charging Station can let the CSMS know it supports a higher field size by reporting this using the device model as OCPPCommCtrlr.FieldLength["Get15118EVCertificateResponse.exiResponse"] = <New max length>

Field Name	Field Type	Card.	Description
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.17. GetBaseReport

1.17.1. GetBaseReportRequest

This contains the field definition of the GetBaseReportRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
requestId	integer	1..1	Required. The Id of the request.
reportBase	ReportBaseEnumType	1..1	Required. This field specifies the report base.

1.17.2. GetBaseReportResponse

This contains the field definition of the GetBaseReportResponse PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
status	GenericDeviceModelStatusEnumType	1..1	Required. This indicates whether the Charging Station is able to accept this request.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.18. GetCertificateStatus

1.18.1. GetCertificateStatusRequest

This contains the field definition of the GetCertificateStatusRequest PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
ocspRequestData	OCSPRequestDataType	1..1	Required. Indicates the certificate of which the status is requested.

1.18.2. GetCertificateStatusResponse

This contains the field definition of the GetCertificateStatusResponse PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
status	GetCertificateStatusEnumType	1..1	Required. This indicates whether the charging station was able to retrieve the OCSP certificate status.
ocspResult	string[0..5500]	0..1	Optional. OCSPResponse class as defined in IETF RFC 6960. DER encoded (as defined in IETF RFC 6960), and then base64 encoded. MAY only be omitted when status is not Accepted.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.19. GetChargingProfiles

1.19.1. GetChargingProfilesRequest

The message GetChargingProfilesRequest can be used by the CSMS to request installed charging profiles from the Charging Station. The charging profiles will then be reported by the Charging Station via ReportChargingProfilesRequest messages.

Class

Field Name	Field Type	Card.	Description
requestId	integer	1..1	Required. Reference identification that is to be used by the Charging Station in the ReportChargingProfilesRequest when provided.
evseId	integer	0..1	Optional. For which EVSE installed charging profiles SHALL be reported. If 0, only charging profiles installed on the Charging Station itself (the grid connection) SHALL be reported. If omitted, all installed charging profiles SHALL be reported. Reported charging profiles SHALL match the criteria in field *chargingProfile*.
chargingProfile	ChargingProfileCriterionType	1..1	Required. Specifies the charging profile.

1.19.2. GetChargingProfilesResponse

This contains the field definition of the GetChargingProfilesResponse PDU sent by the Charging Station to the CSMS in response to a GetChargingProfilesRequest.

Class

Field Name	Field Type	Card.	Description
status	GetChargingProfileStatusEnumType	1..1	Required. This indicates whether the Charging Station is able to process this request and will send ReportChargingProfilesRequest messages.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.20. GetCompositeSchedule

1.20.1. GetCompositeScheduleRequest

This contains the field definition of the GetCompositeScheduleRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
duration	integer	1..1	Required. Length of the requested schedule in seconds.
chargingRateUnit	ChargingRateUnitEnumType	0..1	Optional. Can be used to force a power or current profile.
evseId	integer	1..1	Required. The ID of the EVSE for which the schedule is requested. When evseid=0, the Charging Station will calculate the expected consumption for the grid connection.

1.20.2. GetCompositeScheduleResponse

This contains the field definition of the GetCompositeScheduleResponse PDU sent by the Charging Station to the CSMS in response to a GetCompositeScheduleRequest.

Class

Field Name	Field Type	Card.	Description
status	GenericStatusEnumType	1..1	Required. The Charging Station will indicate if it was able to process the request
schedule	CompositeScheduleType	0..1	Optional. This field contains the calculated composite schedule. It may only be omitted when this message contains status Rejected.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.21. GetDisplayMessages

1.21.1. GetDisplayMessagesRequest

Class

Field Name	Field Type	Card.	Description
id	integer	0..*	Optional. If provided the Charging Station shall return Display Messages of the given ids. This field SHALL NOT contain more ids than set in NumberOfDisplayMessages.maxLimit

Field Name	Field Type	Card.	Description
requestId	integer	1..1	Required. The Id of this request.
priority	MessagePriorityEnumType	0..1	Optional. If provided the Charging Station shall return Display Messages with the given priority only.
state	MessageStateEnumType	0..1	Optional. If provided the Charging Station shall return Display Messages with the given state only.

1.21.2. GetDisplayMessagesResponse

Class

Field Name	Field Type	Card.	Description
status	GetDisplayMessagesStatusEnumType	1..1	Required. Indicates if the Charging Station has Display Messages that match the request criteria in the GetDisplayMessagesRequest
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.22. GetInstalledCertificateIds

1.22.1. GetInstalledCertificateIdsRequest

Used by the CSMS to request an overview of the installed certificates on a Charging Station.

Class

Field Name	Field Type	Card.	Description
certificateType	GetCertificateIdUseEnumType	0..*	Optional. Indicates the type of certificates requested. When omitted, all certificate types are requested.

1.22.2. GetInstalledCertificateIdsResponse

Response to a GetInstalledCertificateIDsRequest.

Class

Field Name	Field Type	Card.	Description
status	GetInstalledCertificateStatusEnumType	1..1	Required. Charging Station indicates if it can process the request.
certificateHashDataChain	CertificateHashDataChainType	0..*	Optional. The Charging Station includes the Certificate information for each available certificate.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.23. GetLocalListVersion

1.23.1. GetLocalListVersionRequest

This contains the field definition of the GetLocalListVersionRequest PDU sent by the CSMS to the Charging Station. No fields are defined.

1.23.2. GetLocalListVersionResponse

This contains the field definition of the GetLocalListVersionResponse PDU sent by the Charging Station to CSMS in response to a GetLocalListVersionRequest.

Class

Field Name	Field Type	Card.	Description
versionNumber	integer	1..1	Required. This contains the current version number of the local authorization list in the Charging Station.

1.24. GetLog

1.24.1. GetLogRequest

This contains the field definition of the GetLogRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
logType	LogEnumType	1..1	Required. This contains the type of log file that the Charging Station should send.
requestId	integer	1..1	Required. The Id of this request
retries	integer	0..1	Optional. This specifies how many times the Charging Station must retry to upload the log before giving up. If this field is not present, it is left to Charging Station to decide how many times it wants to retry. If the value is 0, it means: no retries.
retryInterval	integer	0..1	Optional. The interval in seconds after which a retry may be attempted. If this field is not present, it is left to Charging Station to decide how long to wait between attempts.
log	LogParametersType	1..1	Required. This field specifies the requested log and the location to which the log should be sent.

1.24.2. GetLogResponse

This contains the field definition of the GetLogResponse PDU sent by the Charging Station to the CSMS in response to a GetLogRequest.

Class

Field Name	Field Type	Card.	Description
status	LogStatusEnumType	1..1	Required. This field indicates whether the Charging Station was able to accept the request.
filename	string[0..255]	0..1	Optional. This contains the name of the log file that will be uploaded. This field is not present when no logging information is available.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.25. GetMonitoringReport

1.25.1. GetMonitoringReportRequest

This contains the field definition of the GetMonitoringReportRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
requestId	integer	1..1	Required. The Id of the request.
monitoringCriteria	MonitoringCriterionEnumType	0..3	Optional. This field contains criteria for components for which a monitoring report is requested
componentVariable	ComponentVariableType	0..*	Optional. This field specifies the components and variables for which a monitoring report is requested.

1.25.2. GetMonitoringReportResponse

This contains the field definition of the GetMonitoringReportResponse PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
status	GenericDeviceModelStatusEnumType	1..1	Required. This field indicates whether the Charging Station was able to accept the request.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.26. GetReport

1.26.1. GetReportRequest

This contains the field definition of the GetReportRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
requestId	integer	1..1	Required. The Id of the request.
componentCriteria	ComponentCriterionEnumType	0..4	Optional. This field contains criteria for components for which a report is requested
componentVariable	ComponentVariableType	0..*	Optional. This field specifies the components and variables for which a report is requested.

1.26.2. GetReportResponse

The response to a GetReportRequest, sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
status	GenericDeviceModelStatusEnumType	1..1	Required. This field indicates whether the Charging Station was able to accept the request.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.27. GetTransactionStatus

1.27.1. GetTransactionStatusRequest

With this message, the CSMS can ask the Charging Station whether it has transaction-related messages waiting to be delivered to the CSMS. When a transactionId is provided, only messages for a specific transaction are asked for.

Class

Field Name	Field Type	Card.	Description
transactionId	identifierString[0..36]	0..1	Optional. The Id of the transaction for which the status is requested.

1.27.2. GetTransactionStatusResponse

The response to a GetTransactionStatusRequest, sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
ongoingIndicator	boolean	0..1	Optional. Whether the transaction is still ongoing.
messagesInQueue	boolean	1..1	Required. Whether there are still message to be delivered.

1.28. GetVariables

1.28.1. GetVariablesRequest

This contains the field definition of the GetVariablesRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
getVariableData	GetVariableDataType	1..*	Required. List of requested variables.

1.28.2. GetVariablesResponse

This contains the field definition of the GetVariablesResponse PDU sent by the CSMS to the Charging Station in response to GetVariablesRequest.

Class

Field Name	Field Type	Card.	Description
getVariableResult	GetVariableResultType	1..*	Required. List of requested variables and their values.

1.29. Heartbeat

1.29.1. HeartbeatRequest

This contains the field definition of the HeartbeatRequest PDU sent by the Charging Station to the CSMS. No fields are defined.

1.29.2. HeartbeatResponse

This contains the field definition of the HeartbeatResponse PDU sent by the CSMS to the Charging Station in response to a HeartbeatRequest.

Class

Field Name	Field Type	Card.	Description
currentTime	dateTime	1..1	Required. Contains the current time of the CSMS.

1.30. InstallCertificate

1.30.1. InstallCertificateRequest

Used by the CSMS to request installation of a certificate on a Charging Station. Note: This message is not for installing a TLS client certificate in a charging station. The CertificateSignedRequest mechanism is used for that.

Class

Field Name	Field Type	Card.	Description
certificateType	InstallCertificateUseEnumType	1..1	Required. Indicates the certificate type that is sent.
certificate	string[0..5500]	1..1	Required. A PEM encoded X.509 certificate.

1.30.2. InstallCertificateResponse

The response to a InstallCertificateRequest, sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
status	InstallCertificateStatusEnumType	1..1	Required. Charging Station indicates if installation was successful.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.31. LogStatusNotification

1.31.1. LogStatusNotificationRequest

This contains the field definition of the LogStatusNotificationRequest PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
status	UploadLogStatusEnumType	1..1	Required. This contains the status of the log upload.
requestId	integer	0..1	Optional. The request id that was provided in GetLogRequest that started this log upload. This field is mandatory, unless the message was triggered by a TriggerMessageRequest AND there is no log upload ongoing.

1.31.2. LogStatusNotificationResponse

This contains the field definition of the LogStatusNotificationResponse PDU sent by the CSMS to the Charging Station in response to LogStatusNotificationRequest. No fields are defined.

1.32. MeterValues

1.32.1. MeterValuesRequest

This contains the field definition of the MeterValuesRequest PDU sent by the Charging Station to the CSMS. This message might be removed in a future version of OCPP. It will be replaced by Device Management Monitoring events.

Class

Field Name	Field Type	Card.	Description
evseId	integer	1..1	Required. This contains a number (>0) designating an EVSE of the Charging Station. '0' (zero) is used to designate the main power meter.
meterValue	MeterValueType	1..*	Required. The sampled meter values with timestamps.

1.32.2. MeterValuesResponse

This contains the field definition of the MeterValuesResponse PDU sent by the CSMS to the Charging Station in response to a MeterValuesRequest PDU. This message might be removed in a future version of OCPP. It will be replaced by Device Management Monitoring events.

No fields are defined.

1.33. NotifyChargingLimit

1.33.1. NotifyChargingLimitRequest

The message NotifyChargingLimitRequest can be used to communicate a charging limit, set by an external system on the Charging Station (Not installed by the CSO via SetChargingProfileRequest), to the CSMS.

Class

Field Name	Field Type	Card.	Description
evseId	integer	0..1	Optional. The charging schedule contained in this notification applies to an EVSE. evseId must be > 0.
chargingLimit	ChargingLimitType	1..1	Required. This contains the source of the charging limit and whether it is grid critical.
chargingSchedule	ChargingScheduleType	0..*	Optional. Contains limits for the available power or current over time, as set by the external source.

1.33.2. NotifyChargingLimitResponse

The NotifyChargingLimitResponse message is sent by the CSMS to the Charging Station in response to a NotifyChargingLimitsRequest. No fields are defined.

1.34. NotifyCustomerInformation

This contains the field definition of the NotifyCustomerInformationRequest PDU sent by the Charging Station to the CSMS.

1.34.1. NotifyCustomerInformationRequest

Class

Field Name	Field Type	Card.	Description
data	string[0..512]	1..1	Required. (Part of) the requested data. No format specified in which the data is returned. Should be human readable.
tbc	boolean	0..1	Optional. "to be continued" indicator. Indicates whether another part of the monitoringData follows in an upcoming notifyMonitoringReportRequest message. Default value when omitted is false.
seqNo	integer	1..1	Required. Sequence number of this message. First message starts at 0.
generatedAt	dateTime	1..1	Required. Timestamp of the moment this message was generated at the Charging Station.
requestId	integer	1..1	Required. The Id of the request.

1.34.2. NotifyCustomerInformationResponse

1.35. NotifyDisplayMessages

1.35.1. NotifyDisplayMessagesRequest

This contains the field definition of the NotifyDisplayMessagesRequest PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
requestId	integer	1..1	Required. The id of the GetDisplayMessagesRequest that requested this message.
tbc	boolean	0..1	Optional. "to be continued" indicator. Indicates whether another part of the report follows in an upcoming NotifyDisplayMessagesRequest message. Default value when omitted is false.
messageInfo	MessageInfoType	0..*	Optional. The requested display message as configured in the Charging Station.

1.35.2. NotifyDisplayMessagesResponse

The NotifyDisplayMessagesResponse message is sent by the CSMS to the Charging Station in response to a NotifyDisplayMessagesRequest. No fields are defined.

1.36. NotifyEVChargingNeeds

1.36.1. NotifyEVChargingNeedsRequest

The Charging Station uses this message to communicate the charging needs as calculated by the EV to the CSMS.

Class

Field Name	Field Type	Card.	Description
maxScheduleTuples	integer	0..1	Optional. Contains the maximum schedule tuples the car supports per schedule.
evseId	integer	1..1	Required. Defines the EVSE and connector to which the EV is connected. EvseId may not be 0.
chargingNeeds	ChargingNeedsType	1..1	Required. The characteristics of the energy delivery required.

1.36.2. NotifyEVChargingNeedsResponse

Response to a NotifyEVChargingNeedsRequest.

Class

Field Name	Field Type	Card.	Description
status	NotifyEVChargingNeedsStatusEnumType	1..1	Required. Returns whether the CSMS has been able to process the message successfully. It does not imply that the evChargingNeeds can be met with the current charging profile.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.37. NotifyEVChargingSchedule

1.37.1. NotifyEVChargingScheduleRequest

The Charging Station uses this message to communicate the charging schedule as calculated by the EV to the CSMS.

Class

Field Name	Field Type	Card.	Description
timeBase	dateTime	1..1	Required. Periods contained in the charging profile are relative to this point in time.
evseId	integer	1..1	Required. The charging schedule contained in this notification applies to an EVSE. EvseId must be > 0.
chargingSchedule	ChargingScheduleType	1..1	Required. Planned energy consumption of the EV over time. Always relative to timeBase.

1.37.2. NotifyEVChargingScheduleResponse

Response to a NotifyEVChargingScheduleRequest message.

Class

Field Name	Field Type	Card.	Description
status	GenericStatusEnumType	1..1	Required. Returns whether the CSMS has been able to process the message successfully. It does not imply any approval of the charging schedule.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.38. NotifyEvent

1.38.1. NotifyEventRequest

This contains the field definition of the NotifyEventRequest PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
generatedAt	dateTime	1..1	Required. Timestamp of the moment this message was generated at the Charging Station.
tbc	boolean	0..1	Optional. "to be continued" indicator. Indicates whether another part of the report follows in an upcoming notifyEventRequest message. Default value when omitted is false.
seqNo	integer	1..1	Required. Sequence number of this message. First message starts at 0.
eventData	EventDataType	1..*	Required. List of EventData. An EventData element contains only the Component, Variable and VariableMonitoring data that caused the event. The list of EventData will usally contain one eventData element, but the Charging Station may decide to group multiple events in one notification. For example, when multiple events triggered at the same time.

1.38.2. NotifyEventResponse

Response to NotifyEventRequest. No fields are defined.

1.39. NotifyMonitoringReport

1.39.1. NotifyMonitoringReportRequest

This contains the field definition of the NotifyMonitoringRequest PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
requestId	integer	1..1	Required. The id of the GetMonitoringRequest that requested this report.
tbc	boolean	0..1	Optional. "to be continued" indicator. Indicates whether another part of the monitoringData follows in an upcoming notifyMonitoringReportRequest message. Default value when omitted is false.
seqNo	integer	1..1	Required. Sequence number of this message. First message starts at 0.
generatedAt	dateTime	1..1	Required. Timestamp of the moment this message was generated at the Charging Station.
monitor	MonitoringDataType	0..*	Optional. List of MonitoringData containing monitoring settings.

1.39.2. NotifyMonitoringReportResponse

Response to a NotifyMonitoringRequest message. No fields are defined.

1.40. NotifyReport

1.40.1. NotifyReportRequest

This contains the field definition of the NotifyReportRequest PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
requestId	integer	1..1	Required. The id of the GetReportRequest or GetBaseReportRequest that requested this report
generatedAt	dateTime	1..1	Required. Timestamp of the moment this message was generated at the Charging Station.
tbc	boolean	0..1	Optional. "to be continued" indicator. Indicates whether another part of the report follows in an upcoming notifyReportRequest message. Default value when omitted is false.
seqNo	integer	1..1	Required. Sequence number of this message. First message starts at 0.
reportData	ReportDataType	0..*	Optional. List of ReportData.

1.40.2. NotifyReportResponse

Response to a NotifyReportRequest message. No fields are defined.

1.41. PublishFirmware

1.41.1. PublishFirmwareRequest

This contains the field definition of the PublishFirmwareRequest PDU sent by the CSMS to the Local Controller.

Class

Field Name	Field Type	Card.	Description
location	string[0..512]	1..1	Required. This contains a string containing a URI pointing to a location from which to retrieve the firmware.
retries	integer	0..1	Optional. This specifies how many times the Charging Station must retry to download the firmware before giving up. If this field is not present, it is left to Charging Station to decide how many times it wants to retry. If the value is 0, it means: no retries.
checksum	identifierString[0..32]	1..1	Required. The MD5 checksum over the entire firmware file as a hexadecimal string of length 32.
requestId	integer	1..1	Required. The Id of the request.
retryInterval	integer	0..1	Optional. The interval in seconds after which a retry may be attempted. If this field is not present, it is left to Charging Station to decide how long to wait between attempts.

1.41.2. PublishFirmwareResponse

This contains the field definition of the PublishFirmwareResponse PDU sent by the Local Controller to the CSMS in response to a PublishFirmwareRequest.

Class

Field Name	Field Type	Card.	Description
status	GenericStatusEnumType	1..1	Required. Indicates whether the request was accepted.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.42. PublishFirmwareStatusNotification

1.42.1. PublishFirmwareStatusNotificationRequest

This contains the field definition of the PublishFirmwareStatusNotificationRequest PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
status	PublishFirmwareStatusEnumType	1..1	Required. This contains the progress status of the publishfirmware installation.
location	string[0..512]	0..*	Optional. Required if status is Published. Can be multiple URI's, if the Local Controller supports e.g. HTTP, HTTPS, and FTP.
requestId	integer	0..1	Optional. The request id that was provided in the PublishFirmwareRequest which triggered this action.

1.42.2. PublishFirmwareStatusNotificationResponse

This contains the field definition of the PublishFirmwareStatusNotificationResponse PDU sent by the CSMS to the Charging station in response to a PublishFirmwareStatusNotificationRequest.

1.43. ReportChargingProfiles

1.43.1. ReportChargingProfilesRequest

Reports charging profiles installed in the Charging Station, as requested via a GetChargingProfilesRequest message. The charging profile report can be split over multiple ReportChargingProfilesRequest messages, this can be because charging profiles for different charging sources need to be reported, or because there is just to much data for one message.

Class

Field Name	Field Type	Card.	Description
requestId	integer	1..1	Required. Id used to match the GetChargingProfilesRequest message with the resulting ReportChargingProfilesRequest messages. When the CSMS provided a requestId in the GetChargingProfilesRequest, this field SHALL contain the same value.
chargingLimitSource	ChargingLimitSourceEnumType	1..1	Required. Source that has installed this charging profile.
tbc	boolean	0..1	Optional. To Be Continued. Default value when omitted: false. false indicates that there are no further messages as part of this report.
evseId	integer	1..1	Required. The evse to which the charging profile applies. If evseId = 0, the message contains an overall limit for the Charging Station.
chargingProfile	ChargingProfileType	1..*	Required. The charging profile as configured in the Charging Station.

1.43.2. ReportChargingProfilesResponse

The ReportChargingProfilesResponse message is sent by the CSMS to the Charging Station in response to a ReportChargingProfilesRequest. No fields are defined.

1.44. RequestStartTransaction

1.44.1. RequestStartTransactionRequest

This contains the field definitions of the RequestStartTransactionRequest PDU sent to Charging Station by CSMS.

Class

Field Name	Field Type	Card.	Description
evseId	integer	0..1	Optional. Number of the EVSE on which to start the transaction. EvseId SHALL be > 0
remoteStartId	integer	1..1	Required. Id given by the server to this start request. The Charging Station will return this in the TransactionEventRequest, letting the server know which transaction was started for this request.
idToken	IdTokenType	1..1	Required. The identifier that the Charging Station must use to start a transaction.
chargingProfile	ChargingProfileType	0..1	Optional. Charging Profile to be used by the Charging Station for the requested transaction. ChargingProfilePurpose MUST be set to TxProfile
groupIdToken	IdTokenType	0..1	Optional. The groupIdToken is only relevant when the transaction is to be started on an EVSE for which a reservation for groupIdToken is active, and the configuration variable AuthorizeRemoteStart = false (otherwise the AuthorizeResponse could return the groupIdToken).

1.44.2. RequestStartTransactionResponse

This contains the field definitions of the RequestStartTransactionResponse PDU sent from Charging Station to CSMS.

Class

Field Name	Field Type	Card.	Description
status	RequestStartStopStatusEnumType	1..1	Required. Status indicating whether the Charging Station accepts the request to start a transaction.
transactionId	identifierString[0..36]	0..1	Optional. When the transaction was already started by the Charging Station before the RequestStartTransactionRequest was received, for example: cable plugged in first. This contains the transactionId of the already started transaction.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.45. RequestStopTransaction

1.45.1. RequestStopTransactionRequest

This contains the field definitions of the RequestStopTransactionRequest PDU sent to Charging Station by CSMS.

Class

Field Name	Field Type	Card.	Description
transactionId	identifierString[0..36]	1..1	Required. The identifier of the transaction which the Charging Station is requested to stop.

1.45.2. RequestStopTransactionResponse

This contains the field definitions of the RequestStopTransactionResponse PDU sent from Charging Station to CSMS.

Class

Field Name	Field Type	Card.	Description
status	RequestStartStopStatusEnumType	1..1	Required. Status indicating whether Charging Station accepts the request to stop a transaction.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.46. ReservationStatusUpdate

1.46.1. ReservationStatusUpdateRequest

This contains the field definition of the ReservationStatusUpdateRequest PDU sent by the Charging Station to the CSMS.

Class

Field Name	Field Type	Card.	Description
reservationId	integer	1..1	Required. The ID of the reservation.
reservationUpdateStatus	ReservationUpdateStatusEnumType	1..1	Required. The updated reservation status.

1.46.2. ReservationStatusUpdateResponse

This contains the field definition of the ReservationStatusUpdateResponse PDU sent by the CSMS to the Charging Station in response to a ReservationStatusUpdateRequest. No fields are defined.

1.47. ReserveNow

1.47.1. ReserveNowRequest

This contains the field definition of the ReserveNowRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
id	integer	1..1	Required. Id of reservation.
expiryDateTime	dateTime	1..1	Required. Date and time at which the reservation expires.
connectorType	ConnectorEnumType	0..1	Optional. This field specifies the connector type.
evseId	integer	0..1	Optional. This contains ID of the evse to be reserved.
idToken	IdTokenType	1..1	Required. The identifier for which the reservation is made.
groupIdToken	IdTokenType	0..1	Optional. The group identifier for which the reservation is made.

1.47.2. ReserveNowResponse

This contains the field definition of the ReserveNowResponse PDU sent by the Charging Station to the CSMS in response to ReserveNowRequest PDU.

Class

Field Name	Field Type	Card.	Description
status	ReserveNowStatusEnumType	1..1	Required. This indicates the success or failure of the reservation.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.48. Reset

1.48.1. ResetRequest

This contains the field definition of the ResetRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
type	ResetEnumType	1..1	Required. This contains the type of reset that the Charging Station or EVSE should perform.
evseId	integer	0..1	Optional. This contains the ID of a specific EVSE that needs to be reset, instead of the entire Charging Station.

1.48.2. ResetResponse

This contains the field definition of the ResetResponse PDU sent by the Charging Station to the CSMS in response to ResetRequest.

Class

Field Name	Field Type	Card.	Description
status	ResetStatusEnumType	1..1	Required. This indicates whether the Charging Station is able to perform the reset.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.49. SecurityEventNotification

1.49.1. SecurityEventNotificationRequest

Sent by the Charging Station to the CSMS in case of a security event.

Class

Field Name	Field Type	Card.	Description
type	string[0..50]	1..1	Required. Type of the security event. This value should be taken from the Security events list.
timestamp	dateTime	1..1	Required. Date and time at which the event occurred.
techInfo	string[0..255]	0..1	Optional. Additional information about the occurred security event.

1.49.2. SecurityEventNotificationResponse

Sent by the CSMS to the Charging Station to confirm the receipt of a SecurityEventNotificationRequest message. No fields are defined.

1.50. SendLocalList

1.50.1. SendLocalListRequest

This contains the field definition of the SendLocalListRequest PDU sent by the CSMS to the Charging Station. If no (empty) localAuthorizationList is given and the updateType is Full, all IdTokens are removed from the list. Requesting a Differential update without or with empty localAuthorizationList will have no effect on the list. All IdTokens in the localAuthorizationList MUST be unique, no duplicate values are allowed.

Class

Field Name	Field Type	Card.	Description
versionNumber	integer	1..1	Required. In case of a full update this is the version number of the full list. In case of a differential update it is the version number of the list after the update has been applied.
updateType	UpdateEnumType	1..1	Required. This contains the type of update (full or differential) of this request.

Field Name	Field Type	Card.	Description
localAuthorizationList	AuthorizationData	0..*	Optional. This contains the Local Authorization List entries.

1.50.2. SendLocalListResponse

This contains the field definition of the SendLocalListResponse PDU sent by the Charging Station to the CSMS in response to SendLocalListRequest PDU.

Class

Field Name	Field Type	Card.	Description
status	SendLocalListStatusEnumType	1..1	Required. This indicates whether the Charging Station has successfully received and applied the update of the Local Authorization List.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.51. SetChargingProfile

1.51.1. SetChargingProfileRequest

This contains the field definition of the SetChargingProfileRequest PDU sent by the CSMS to the Charging Station. The CSMS uses this message to send charging profiles to a Charging Station.

Class

Field Name	Field Type	Card.	Description
evseId	integer	1..1	Required. For TxDefaultProfile an evseId=0 applies the profile to each individual evse. For ChargingStationMaxProfile and ChargingStationExternalConstraints an evseId=0 contains an overal limit for the whole Charging Station.
chargingProfile	ChargingProfileType	1..1	Required. The charging profile to be set at the Charging Station.

1.51.2. SetChargingProfileResponse

This contains the field definition of the SetChargingProfileResponse PDU sent by the Charging Station to the CSMS in response to SetChargingProfileRequest PDU.

Class

Field Name	Field Type	Card.	Description
status	ChargingProfileStatusEnumType	1..1	Required. Returns whether the Charging Station has been able to process the message successfully. This does not guarantee the schedule will be followed to the letter. There might be other constraints the Charging Station may need to take into account.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.52. SetDisplayMessage

1.52.1. SetDisplayMessageRequest

This contains the field definition of the SetDisplayMessageRequest PDU sent by the CSMS to the Charging Station. The CSMS asks the Charging Station to configure a new display message that the Charging Station will display (in the future). See also O01 - Set Display Message, O02 - Set Display Message for Transaction and O06 - Replace Display Message

Class

Field Name	Field Type	Card.	Description
message	MessageInfoType	1..1	Required. Message to be configured in the Charging Station, to be displayed.

1.52.2. SetDisplayMessageResponse

This contains the field definition of the SetDisplayMessageResponse PDU sent by the Charging Station to the CSMS in a response to a SetDisplayMessageRequest. See also O01 - Set Display Message, O02 - Set Display Message for Transaction and O06 - Replace Display Message

Class

Field Name	Field Type	Card.	Description
status	DisplayMessageStatusEnumType	1..1	Required. This indicates whether the Charging Station is able to display the message.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.53. SetMonitoringBase

1.53.1. SetMonitoringBaseRequest

This contains the field definition of the SetMonitoringBaseRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
monitoringBase	MonitoringBaseEnumType	1..1	Required. Specify which monitoring base will be set

1.53.2. SetMonitoringBaseResponse

This contains the field definition of the SetMonitoringBaseResponse PDU sent by the Charging Station to the CSMS in response to a SetMonitoringBaseRequest.

Class

Field Name	Field Type	Card.	Description
status	GenericDeviceModelStatusEnumType	1..1	Required. Indicates whether the Charging Station was able to accept the request.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.54. SetMonitoringLevel

1.54.1. SetMonitoringLevelRequest

This contains the field definition of the SetMonitoringLevelRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
severity	integer	1..1	Required. The Charging Station SHALL only report events with a severity number lower than or equal to this severity. The severity range is 0-9, with 0 as the highest and 9 as the lowest severity level. The severity levels have the following meaning: **0-Danger** Indicates lives are potentially in danger. Urgent attention is needed and action should be taken immediately. **1-Hardware Failure** Indicates that the Charging Station is unable to continue regular operations due to Hardware issues. Action is required. **2-System Failure** Indicates that the Charging Station is unable to continue regular operations due to software or minor hardware issues. Action is required. **3-Critical** Indicates a critical error. Action is required. **4-Error** Indicates a non-urgent error. Action is required. **5-Alert** Indicates an alert event. Default severity for any type of monitoring event. **6-Warning** Indicates a warning event. Action may be required. **7-Notice** Indicates an unusual event. No immediate action is required. **8-Informational** Indicates a regular operational event. May be used for reporting, measuring throughput, etc. No action is required. **9-Debug** Indicates information useful to developers for debugging, not useful during operations.

1.54.2. SetMonitoringLevelResponse

This contains the field definition of the SetMonitoringLevelResponse PDU sent by the Charging Station to the CSMS in response to a SetMonitoringLevelRequest.

Class

Field Name	Field Type	Card.	Description
status	GenericStatusEnumType	1..1	Required. Indicates whether the Charging Station was able to accept the request.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.55. SetNetworkProfile

1.55.1. SetNetworkProfileRequest

With this message the CSMS gains the ability to configure the connection data (e.g. CSMS URL, OCPP version, APN, etc) on a Charging Station.

Class

Field Name	Field Type	Card.	Description
configurationSlot	integer	1..1	Required. Slot in which the configuration should be stored.
connectionData	NetworkConnectionProfileType	1..1	Required. Connection details.

1.55.2. SetNetworkProfileResponse

This contains the field definition of the SetNetworkProfileResponse PDU sent by the Charging Station to the CSMS in response to a SetNetworkProfileRequest.

Class

Field Name	Field Type	Card.	Description
status	SetNetworkProfileStatusEnumType	1..1	Required. Result of operation.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.56. SetVariableMonitoring

1.56.1. SetVariableMonitoringRequest

This contains the field definition of the SetVariableMonitoringRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
setMonitoringData	SetMonitoringDataType	1..*	Required. List of MonitoringData containing monitoring settings.

1.56.2. SetVariableMonitoringResponse

This contains the field definition of the SetVariableMonitoringResponse PDU sent by the Charging Station to the CSMS in response to a SetVariableMonitoringRequest.

Class

Field Name	Field Type	Card.	Description
setMonitoringResult	SetMonitoringResultType	1..*	Required. List of result statuses per monitor.

1.57. SetVariables

1.57.1. SetVariablesRequest

This contains the field definition of the SetVariablesRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
setVariableData	SetVariableDataType	1..*	Required. List of Component-Variable pairs and attribute values to set.

1.57.2. SetVariablesResponse

This contains the field definition of the SetVariablesResponse PDU sent by the Charging Station to the CSMS in response to a SetVariablesRequest.

Class

Field Name	Field Type	Card.	Description
setVariableResult	SetVariableResultType	1..*	Required. List of result statuses per Component-Variable.

1.58. SignCertificate

1.58.1. SignCertificateRequest

Sent by the Charging Station to the CSMS to request that the Certificate Authority signs the public key into a certificate.

Class

Field Name	Field Type	Card.	Description
csr	string[0..5500]	1..1	Required. The Charging Station SHALL send the public key in form of a Certificate Signing Request (CSR) as described in RFC 2986 [22] and then PEM encoded, using the SignCertificateRequest message.
certificateType	CertificateSigningUseEnumType	0..1	Optional. Indicates the type of certificate that is to be signed. When omitted the certificate is to be used for both the 15118 connection (if implemented) and the Charging Station to CSMS connection.

1.58.2. SignCertificateResponse

Sent by the CSMS to the Charging Station in response to the SignCertificateRequest message.

Class

Field Name	Field Type	Card.	Description
status	GenericStatusEnumType	1..1	Required. Specifies whether the CSMS can process the request.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.59. StatusNotification

1.59.1. StatusNotificationRequest

This contains the field definition of the StatusNotificationRequest PDU sent by the Charging Station to the CSMS. This message might be removed in a future version of OCPP. It will be replaced by Device Management Monitoring events.

Class

Field Name	Field Type	Card.	Description
timestamp	dateTime	1..1	Required. The time for which the status is reported.
connectorStatus	ConnectorStatusEnumType	1..1	Required. This contains the current status of the Connector.
evseId	integer	1..1	Required. The id of the EVSE to which the connector belongs for which the the status is reported.
connectorId	integer	1..1	Required. The id of the connector within the EVSE for which the status is reported.

1.59.2. StatusNotificationResponse

This contains the field definition of StatusNotificationResponse sent by the CSMS to the Charging Station in response to a StatusNotificationRequest. This message might be removed in a future version of OCPP. It will be replaced by Device Management Monitoring events.

No fields are defined.

1.60. TransactionEvent

1.60.1. TransactionEventRequest

This section contains the field definition of the TransactionEventRequest PDU sent by the Charging Station to the CSMS. For each of the eventTypes; Started, Updated and Ended, the corresponding cardinality is specified.

Class

Field Name	Field Type	Card.	Description
eventType	TransactionEventEnumType	1..1	Required. This contains the type of this event. The first TransactionEvent of a transaction SHALL contain: "Started" The last TransactionEvent of a transaction SHALL contain: "Ended" All others SHALL contain: "Updated"
timestamp	dateTime	1..1	Required. The date and time at which this transaction event occurred.
triggerReason	TriggerReasonEnumType	1..1	Required. Reason the Charging Station sends this message to the CSMS
seqNo	integer	1..1	Required. Incremental sequence number, helps with determining if all messages of a transaction have been received.
offline	boolean	0..1	Optional. Indication that this transaction event happened when the Charging Station was offline. Default = false, meaning: the event occurred when the Charging Station was online.
numberOfPhasesUsed	integer	0..1	Optional. If the Charging Station is able to report the number of phases used, then it SHALL provide it. When omitted the CSMS may be able to determine the number of phases used via device management.
cableMaxCurrent	integer	0..1	Optional. The maximum current of the connected cable in Ampere (A).
reservationId	integer	0..1	Optional. This contains the Id of the reservation that terminates as a result of this transaction.
transactionInfo	TransactionType	1..1	Required. Contains transaction specific information.
idToken	IdTokenType	0..1	Optional. This contains the identifier for which a transaction is (or will be) started or stopped. Is required when the EV Driver becomes authorized for this transaction and when the EV Driver ends authorization. The IdToken should only be sent once in a TransactionEventRequest for every authorization (for starting or for stopping) done for this transaction, so that CSMS can return the idTokenInfo in the TransactionEventResponse. idToken should not be present in the TransactionEventRequest when a transaction is ended by a RequestStopTransactionRequest or a ResetRequest.
evse	EVSEType	0..1	Optional. This identifies which evse (and connector) of the Charging Station is used.

Field Name	Field Type	Card.	Description
meterValue	MeterValueType	0..*	Optional. This contains the relevant meter values. Depending on the EventType of this TransactionEvent the following Configuration Variable is used to configure the content: Started: SampledDataTxStartedMeasurands Updated: SampledDataTxUpdatedMeasurands Ended: SampledDataTxEndedMeasurands & AlignedDataTxEndedMeasurands

1.60.2. TransactionEventResponse

This contains the field definition of the TransactionEventResponse PDU sent by the CSMS to the Charging Station in response to a TransactionEventRequest.

Class

Field Name	Field Type	Card.	Description
totalCost	decimal	0..1	Optional. When *eventType* of TransactionEventRequest is `Updated`, then this value contains the *running cost*. When *eventType* of TransactionEventRequest is `Ended`, then this contains the final *total cost* of this transaction, including taxes, in the currency configured with the Configuration Variable: `Currency`. Absence of this value does not imply that the transaction was free. To indicate a free transaction, the CSMS SHALL send a value of 0.00.
chargingPriority	integer	0..1	Optional. Priority from a business point of view. Default priority is 0, The range is from -9 to 9. Higher values indicate a higher priority. The chargingPriority in TransactionEventResponse is temporarily, so it may not be set in the IdTokenInfoType afterwards. Also the chargingPriority in TransactionEventResponse overrules the one in IdTokenInfoType.
idTokenInfo	IdTokenInfoType	0..1	Optional. This contains information about authorization status, expiry and group id. Is required when the TransactionEventRequest contained an idToken.
updatedPersonalMessage	MessageContentType	0..1	Optional. This can contain updated personal message that can be shown to the EV Driver. This can be used to provide updated tariff information.

1.61. TriggerMessage

1.61.1. TriggerMessageRequest

This contains the field definition of the TriggerMessageRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
requestedMessage	MessageTriggerEnumType	1..1	Required. Type of message to be triggered.
evse	EVSEType	0..1	Optional. Can be used to specifiy the EVSE and Connector if required for the message which needs to be sent.

1.61.2. TriggerMessageResponse

This contains the field definition of the TriggerMessageResponse PDU sent by the Charging Station to the CSMS in response to TriggerMessageRequest.

Class

Field Name	Field Type	Card.	Description
status	TriggerMessageStatusEnumType	1..1	Required. Indicates whether the Charging Station will send the requested notification or not.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.62. UnlockConnector

1.62.1. UnlockConnectorRequest

This contains the field definition of the UnlockConnectorRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
evseId	integer	1..1	Required. This contains the identifier of the EVSE for which a connector needs to be unlocked.
connectorId	integer	1..1	Required. This contains the identifier of the connector that needs to be unlocked.

1.62.2. UnlockConnectorResponse

This contains the field definition of the UnlockConnectorResponse PDU sent by the Charging Station to the CSMS in response to an UnlockConnectorRequest.

Class

Field Name	Field Type	Card.	Description
status	UnlockStatusEnumType	1..1	Required. This indicates whether the Charging Station has unlocked the connector.

Field Name	Field Type	Card.	Description
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

1.63. UnpublishFirmware

1.63.1. UnpublishFirmwareRequest

This contains the field definition of the UnpublishFirmwareRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
checksum	identifierString[0..32]	1..1	Required. The MD5 checksum over the entire firmware file as a hexadecimal string of length 32.

1.63.2. UnpublishFirmwareResponse

This contains the field definition of the UnpublishFirmwareResponse PDU sent by the Charging Station to the CSMS in response to a UnpublishFirmwareRequest.

Class

Field Name	Field Type	Card.	Description
status	UnpublishFirmwareStatusEnumType	1..1	Required. Indicates whether the Local Controller succeeded in unpublishing the firmware.

1.64. UpdateFirmware

1.64.1. UpdateFirmwareRequest

This contains the field definition of the UpdateFirmwareRequest PDU sent by the CSMS to the Charging Station.

Class

Field Name	Field Type	Card.	Description
retries	integer	0..1	Optional. This specifies how many times the Charging Station must retry to download the firmware before giving up. If this field is not present, it is left to Charging Station to decide how many times it wants to retry. If the value is 0, it means: no retries.
retryInterval	integer	0..1	Optional. The interval in seconds after which a retry may be attempted. If this field is not present, it is left to Charging Station to decide how long to wait between attempts.

Field Name	Field Type	Card.	Description
requestId	integer	1..1	Required. The Id of this request
firmware	FirmwareType	1..1	Required. Specifies the firmware to be updated on the Charging Station.

1.64.2. UpdateFirmwareResponse

This contains the field definition of the UpdateFirmwareResponse PDU sent by the Charging Station to the CSMS in response to an UpdateFirmwareRequest.

Class

Field Name	Field Type	Card.	Description
status	UpdateFirmwareStatusEnumType	1..1	Required. This field indicates whether the Charging Station was able to accept the request.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

2. Datatypes

2.1. ACChargingParametersType

Class

EV AC charging parameters.

ACChargingParametersType is used by: Common:ChargingNeedsType

Field Name	Field Type	Card.	Description
energyAmount	integer	1..1	Required. Amount of energy requested (in Wh). This includes energy required for preconditioning.
evMinCurrent	integer	1..1	Required. Minimum current (amps) supported by the electric vehicle (per phase).
evMaxCurrent	integer	1..1	Required. Maximum current (amps) supported by the electric vehicle (per phase). Includes cable capacity.
evMaxVoltage	integer	1..1	Required. Maximum voltage supported by the electric vehicle

2.2. AdditionalInfoType

Class

Contains a case insensitive identifier to use for the authorization and the type of authorization to support multiple forms of identifiers.

AdditionalInfoType is used by: Common:IdTokenType

Field Name	Field Type	Card.	Description
additionalIdToken	identifierString[0..36]	1..1	Required. This field specifies the additional IdToken.
type	string[0..50]	1..1	Required. This defines the type of the additionalIdToken. This is a custom type, so the implementation needs to be agreed upon by all involved parties.

2.3. APNType

Class

Collection of configuration data needed to make a data-connection over a cellular network.

NOTE	When asking a GSM modem to dial in, it is possible to specify which mobile operator should be used. This can be done with the mobile country code (MCC) in combination with a mobile network code (MNC). Example: If your preferred network is Vodafone Netherlands, the MCC=204 and the MNC=04 which means the key PreferredNetwork = 20404 Some modems allows to specify a preferred network, which means, if this network is not available, a different network is used. If you specify UseOnlyPreferredNetwork and this network is not available, the modem will not dial in.

APNType is used by: SetNetworkProfileRequest.NetworkConnectionProfileType

Field Name	Field Type	Card.	Description
apn	string[0..512]	1..1	Required. The Access Point Name as an URL.
apnUserName	string[0..20]	0..1	Optional. APN username.
apnPassword	string[0..20]	0..1	Optional. APN Password.
simPin	integer	0..1	Optional. SIM card pin code.
preferredNetwork	identifierString[0..6]	0..1	Optional. Preferred network, written as MCC and MNC concatenated. See note.
useOnlyPreferredNetwork	boolean	0..1	Optional. Default: false. Use only the preferred Network, do not dial in when not available. See Note.
apnAuthentication	APNAuthenticationEnumType	1..1	Required. Authentication method.

2.4. AuthorizationData

Class

Contains the identifier to use for authorization.

AuthorizationData is used by: SendLocalListRequest

Field Name	Field Type	Card.	Description
idTokenInfo	IdTokenInfoType	0..1	Optional. Required when UpdateType is Full. This contains information about authorization status, expiry and group id. For a Differential update the following applies: If this element is present, then this entry SHALL be added or updated in the Local Authorization List. If this element is absent, the entry for this IdToken in the Local Authorization List SHALL be deleted.
idToken	IdTokenType	1..1	Required. This contains the identifier which needs to be stored for authorization.

2.5. CertificateHashDataChainType

Class

CertificateHashDataChainType is used by: GetInstalledCertificateIdsResponse

Field Name	Field Type	Card.	Description
certificateType	GetCertificateIdUseEnumType	1..1	Required. Indicates the type of the requested certificate(s).

Field Name	Field Type	Card.	Description
certificateHashData	CertificateHashDataType	1..1	Required. Information to identify a certificate.
childCertificateHashData	CertificateHashDataType	0..4	Optional. Information to identify the child certificate(s).

2.6. CertificateHashDataType

Class

CertificateHashDataType is used by: Common:CertificateHashDataChainType, DeleteCertificateRequest, CustomerInformationRequest

Field Name	Field Type	Card.	Description
hashAlgorithm	HashAlgorithmEnumType	1..1	Required. Used algorithms for the hashes provided.
issuerNameHash	identifierString[0..128]	1..1	Required. The hash of the issuer's distinguished name (DN), that must be calculated over the DER encoding of the issuer's name field in the certificate being checked.
issuerKeyHash	string[0..128]	1..1	Required. The hash of the DER encoded public key: the value (excluding tag and length) of the subject public key field in the issuer's certificate.
serialNumber	identifierString[0..40]	1..1	Required. The string representation of the hexadecimal value of the serial number without the prefix "0x" and without leading zeroes.

2.7. ChargingLimitType

Class

ChargingLimitType is used by: NotifyChargingLimitRequest

Field Name	Field Type	Card.	Description
chargingLimitSource	ChargingLimitSourceEnumType	1..1	Required. Represents the source of the charging limit.
isGridCritical	boolean	0..1	Optional. Indicates whether the charging limit is critical for the grid.

2.8. ChargingNeedsType

Class

ChargingNeedsType is used by: NotifyEVChargingNeedsRequest

Field Name	Field Type	Card.	Description
requestedEnergyTransfer	EnergyTransferModeEnumType	1..1	Required. Mode of energy transfer requested by the EV.
departureTime	dateTime	0..1	Optional. Estimated departure time of the EV.
acChargingParameters	ACChargingParametersType	0..1	Optional. EV AC charging parameters.
dcChargingParameters	DCChargingParametersType	0..1	Optional. EV DC charging parameters

2.9. ChargingProfileCriterionType

Class

A ChargingProfile consists of ChargingSchedule, describing the amount of power or current that can be delivered per time interval.

ChargingProfileCriterionType is used by: GetChargingProfilesRequest

Field Name	Field Type	Card.	Description
chargingProfilePurpose	ChargingProfilePurposeEnumType	0..1	Optional. Defines the purpose of the schedule transferred by this profile
stackLevel	integer	0..1	Optional. Value determining level in hierarchy stack of profiles. Higher values have precedence over lower values. Lowest level is 0.
chargingProfileId	integer	0..*	Optional. List of all the chargingProfileIds requested. Any ChargingProfile that matches one of these profiles will be reported. If omitted, the Charging Station SHALL not filter on chargingProfileId. This field SHALL NOT contain more ids than set in ChargingProfileEntries.maxLimit
chargingLimitSource	ChargingLimitSourceEnumType	0..4	Optional. For which charging limit sources, charging profiles SHALL be reported. If omitted, the Charging Station SHALL not filter on chargingLimitSource.

2.10. ChargingProfileType

Class

A ChargingProfile consists of ChargingSchedule, describing the amount of power or current that can be delivered per time interval.

ChargingProfileType is used by: RequestStartTransactionRequest, SetChargingProfileRequest, ReportChargingProfilesRequest

Field Name	Field Type	Card.	Description
id	integer	1..1	Required. Id of ChargingProfile.

Field Name	Field Type	Card.	Description
stackLevel	integer	1..1	Required. Value determining level in hierarchy stack of profiles. Higher values have precedence over lower values. Lowest level is 0.
chargingProfilePurpose	ChargingProfilePurposeEnumType	1..1	Required. Defines the purpose of the schedule transferred by this profile
chargingProfileKind	ChargingProfileKindEnumType	1..1	Required. Indicates the kind of schedule.
recurrencyKind	RecurrencyKindEnumType	0..1	Optional. Indicates the start point of a recurrence.
validFrom	dateTime	0..1	Optional. Point in time at which the profile starts to be valid. If absent, the profile is valid as soon as it is received by the Charging Station.
validTo	dateTime	0..1	Optional. Point in time at which the profile stops to be valid. If absent, the profile is valid until it is replaced by another profile.
transactionId	identifierString[0..36]	0..1	Optional. SHALL only be included when ChargingProfilePurpose is set to TxProfile in a SetChargingProfileRequest. The transactionId is used to match the profile to a specific transaction.
chargingSchedule	ChargingScheduleType	1..3	Required. Schedule that contains limits for the available power or current over time. In order to support ISO 15118 schedule negotiation, it supports at most three schedules with associated tariff to choose from. Having multiple *chargingSchedules* is only allowed for charging profiles of purpose `TxProfile` in the context of an ISO 15118 charging session.

2.11. ChargingSchedulePeriodType

Class

Charging schedule period structure defines a time period in a charging schedule.

ChargingSchedulePeriodType is used by: Common:ChargingScheduleType, Common:CompositeScheduleType

Field Name	Field Type	Card.	Description
startPeriod	integer	1..1	Required. Start of the period, in seconds from the start of schedule. The value of StartPeriod also defines the stop time of the previous period.
limit	decimal	1..1	Required. Charging rate limit during the schedule period, in the applicable chargingRateUnit, for example in Amperes (A) or Watts (W). Accepts at most one digit fraction (e.g. 8.1).

Field Name	Field Type	Card.	Description
numberPhases	integer	0..1	Optional. The number of phases that can be used for charging. For a DC EVSE this field should be omitted. For an AC EVSE a default value of numberPhases = 3 will be assumed if the field is absent.
phaseToUse	integer	0..1	Optional. Values: 1..3, Used if numberPhases=1 and if the EVSE is capable of switching the phase connected to the EV, i.e. ACPhaseSwitchingSupported is defined and true. It's not allowed unless both conditions above are true. If both conditions are true, and phaseToUse is omitted, the Charging Station / EVSE will make the selection on its own.

2.11.1. ChargingScheduleType

Class

Charging schedule structure defines a list of charging periods, as used in: GetCompositeSchedule.conf and ChargingProfile.

ChargingScheduleType is used by: Common:ChargingProfileType, NotifyChargingLimitRequest, NotifyEVChargingScheduleRequest

Field Name	Field Type	Card.	Description
id	integer	1..1	Required. Identifies the ChargingSchedule.
startSchedule	dateTime	0..1	Optional. Starting point of an absolute or recurring schedule.
duration	integer	0..1	Optional. Duration of the charging schedule in seconds. If the duration is left empty, the last period will continue indefinitely or until end of the transaction if chargingProfilePurpose = TxProfile.
chargingRateUnit	ChargingRateUnitEnumType	1..1	Required. The unit of measure Limit is expressed in.
minChargingRate	decimal	0..1	Optional. Minimum charging rate supported by the EV. The unit of measure is defined by the chargingRateUnit. This parameter is intended to be used by a local smart charging algorithm to optimize the power allocation for in the case a charging process is inefficient at lower charging rates. Accepts at most one digit fraction (e.g. 8.1)
chargingSchedulePeriod	ChargingSchedulePeriodType	1..1024	Required. List of ChargingSchedulePeriod elements defining maximum power or current usage over time. The maximum number of periods, that is supported by the Charging Station, if less than 1024, is set by device model variable SmartChargingCtrlr.PeriodsPerSchedule.
salesTariff	SalesTariffType	0..1	Optional. Sales tariff associated with this charging schedule.

2.12. ChargingStationType

Class

The physical system where an Electrical Vehicle (EV) can be charged.

ChargingStationType is used by: BootNotificationRequest

Field Name	Field Type	Card.	Description
serialNumber	string[0..25]	0..1	Optional. Vendor-specific device identifier.
model	string[0..20]	1..1	Required. Defines the model of the device.
vendorName	string[0..50]	1..1	Required. Identifies the vendor (not necessarily in a unique manner).
firmwareVersion	string[0..50]	0..1	Optional. This contains the firmware version of the Charging Station.
modem	ModemType	0..1	Optional. Defines the functional parameters of a communication link.

2.13. ClearChargingProfileType

Class

A ChargingProfile consists of a ChargingSchedule, describing the amount of power or current that can be delivered per time interval.

ClearChargingProfileType is used by: ClearChargingProfileRequest

Field Name	Field Type	Card.	Description
evseId	integer	0..1	Optional. Specifies the id of the EVSE for which to clear charging profiles. An evseId of zero (0) specifies the charging profile for the overall Charging Station. Absence of this parameter means the clearing applies to all charging profiles that match the other criteria in the request.
chargingProfilePurpose	ChargingProfilePurposeEnumType	0..1	Optional. Specifies to purpose of the charging profiles that will be cleared, if they meet the other criteria in the request.
stackLevel	integer	0..1	Optional. Specifies the stackLevel for which charging profiles will be cleared, if they meet the other criteria in the request.

2.14. ClearMonitoringResultType

Class

ClearMonitoringResultType is used by: ClearVariableMonitoringResponse

Field Name	Field Type	Card.	Description
status	ClearMonitoringStatusEnumType	1..1	Required. Result of the clear request for this monitor, identified by its Id.
id	integer	1..1	Required. Id of the monitor of which a clear was requested.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

2.15. ComponentType

Class

A physical or logical component

ComponentType is used by: Common:ComponentVariableType, Common:MessageInfoType, GetVariablesRequest.GetVariableDataType, GetVariablesResponse.GetVariableResultType, NotifyMonitoringReportRequest.MonitoringDataType, NotifyReportRequest.ReportDataType, SetVariableMonitoringRequest.SetMonitoringDataType, SetVariableMonitoringResponse.SetMonitoringResultType, SetVariablesRequest.SetVariableDataType, SetVariablesResponse.SetVariableResultType, NotifyEventRequest.EventDataType

Field Name	Field Type	Card.	Description
name	identifierString[0..50]	1..1	Required. Name of the component. Name should be taken from the list of standardized component names whenever possible. Case Insensitive. strongly advised to use Camel Case.
instance	identifierString[0..50]	0..1	Optional. Name of instance in case the component exists as multiple instances. Case Insensitive. strongly advised to use Camel Case.
evse	EVSEType	0..1	Optional. Specifies the EVSE when component is located at EVSE level, also specifies the connector when component is located at Connector level.

2.16. ComponentVariableType

Class

Class to report components, variables and variable attributes and characteristics.

ComponentVariableType is used by: GetMonitoringReportRequest, GetReportRequest

Field Name	Field Type	Card.	Description
component	ComponentType	1..1	Required. Component for which a report of Variable is requested.
variable	VariableType	0..1	Optional. Variable(s) for which the report is requested.

2.17. CompositeScheduleType

Class

CompositeScheduleType is used by: GetCompositeScheduleResponse

Field Name	Field Type	Card.	Description
evseId	integer	1..1	Required. The ID of the EVSE for which the schedule is requested. When evseid=0, the Charging Station calculated the expected consumption for the grid connection.
duration	integer	1..1	Required. Duration of the schedule in seconds.
scheduleStart	dateTime	1..1	Required. Date and time at which the schedule becomes active. All time measurements within the schedule are relative to this timestamp.
chargingRateUnit	ChargingRateUnitEnumType	1..1	Required. The unit of measure Limit is expressed in.
chargingSchedulePeriod	ChargingSchedulePeriodType	1..*	Required. List of ChargingSchedulePeriod elements defining maximum power or current usage over time.

2.18. ConsumptionCostType

Class

ConsumptionCostType is used by: Common:SalesTariffEntryType

Field Name	Field Type	Card.	Description
startValue	decimal	1..1	Required. The lowest level of consumption that defines the starting point of this consumption block. The block interval extends to the start of the next interval.
cost	CostType	1..3	Required. This field contains the cost details.

2.19. CostType

Class

CostType is used by: Common:ConsumptionCostType

Field Name	Field Type	Card.	Description

|costKind |CostKindEnumType |1..1 |Required. The kind of cost referred to in the message element amount |amount |integer |1..1 |Required. The estimated or actual cost per kWh |amountMultiplier |integer |0..1 |Optional. Values: -3..3, The amountMultiplier defines the exponent to base 10 (dec). The final value is determined by: amount * 10 ^ amountMultiplier

2.20. DCChargingParametersType

Class

EV DC charging parameters

DCChargingParametersType is used by: Common:ChargingNeedsType

Field Name	Field Type	Card.	Description
evMaxCurrent	integer	1..1	Required. Maximum current (amps) supported by the electric vehicle. Includes cable capacity.
evMaxVoltage	integer	1..1	Required. Maximum voltage supported by the electric vehicle
energyAmount	integer	0..1	Optional. Amount of energy requested (in Wh). This inludes energy required for preconditioning.
evMaxPower	integer	0..1	Optional. Maximum power (in W) supported by the electric vehicle. Required for DC charging.
stateOfCharge	integer, 0 < = val < = 100	0..1	Optional. Energy available in the battery (in percent of the battery capacity)
evEnergyCapacity	integer	0..1	Optional. Capacity of the electric vehicle battery (in Wh)
fullSoC	integer, 0 < = val < = 100	0..1	Optional. Percentage of SoC at which the EV considers the battery fully charged. (possible values: 0 - 100)
bulkSoC	integer, 0 < = val < = 100	0..1	Optional. Percentage of SoC at which the EV considers a fast charging process to end. (possible values: 0 - 100)

2.21. EventDataType

Class

Class to report an event notification for a component-variable.

EventDataType is used by: NotifyEventRequest

Field Name	Field Type	Card.	Description
eventId	integer	1..1	Required. Identifies the event. This field can be referred to as a cause by other events.
timestamp	dateTime	1..1	Required. Timestamp of the moment the report was generated.
trigger	EventTriggerEnumType	1..1	Required. Type of trigger for this event, e.g. exceeding a threshold value.

Field Name	Field Type	Card.	Description
cause	integer	0..1	Optional. Refers to the Id of an event that is considered to be the cause for this event.
actualValue	string[0..2500]	1..1	Required. Actual value (*attributeType* Actual) of the variable. The Configuration Variable ReportingValueSize can be used to limit GetVariableResult.attributeValue, VariableAttribute.value and EventData.actualValue. The max size of these values will always remain equal.
techCode	string[0..50]	0..1	Optional. Technical (error) code as reported by component.
techInfo	string[0..500]	0..1	Optional. Technical detail information as reported by component.
cleared	boolean	0..1	Optional. Cleared is set to true to report the clearing of a monitored situation, i.e. a 'return to normal'.
transactionId	identifierString[0..36]	0..1	Optional. If an event notification is linked to a specific transaction, this field can be used to specify its transactionId.
variableMonitoringId	integer	0..1	Optional. Identifies the VariableMonitoring which triggered the event.
eventNotificationType	EventNotificationEnumType	1..1	Required. Specifies the event notification type of the message.
component	ComponentType	1..1	Required. Component for which event is notified.
variable	VariableType	1..1	Required. Variable for which event is notified.

2.22. EVSEType

Class

Electric Vehicle Supply Equipment

EVSEType is used by: Common:ComponentType, TriggerMessageRequest, ChangeAvailabilityRequest, TransactionEventRequest

Field Name	Field Type	Card.	Description
id	integer	1..1	Required. EVSE Identifier. This contains a number (> 0) designating an EVSE of the Charging Station.
connectorId	integer	0..1	Optional. An id to designate a specific connector (on an EVSE) by connector index number.

2.23. FirmwareType

Class

Represents a copy of the firmware that can be loaded/updated on the Charging Station.

FirmwareType is used by: UpdateFirmwareRequest

Field Name	Field Type	Card.	Description
location	string[0..512]	1..1	Required. URI defining the origin of the firmware.
retrieveDateTime	dateTime	1..1	Required. Date and time at which the firmware shall be retrieved.
installDateTime	dateTime	0..1	Optional. Date and time at which the firmware shall be installed.
signingCertificate	string[0..5500]	0..1	Optional. Certificate with which the firmware was signed. PEM encoded X.509 certificate.
signature	string[0..800]	0..1	Optional. Base64 encoded firmware signature.

2.24. GetVariableDataType

Class

Class to hold parameters for GetVariables request.

GetVariableDataType is used by: GetVariablesRequest

Field Name	Field Type	Card.	Description
attributeType	AttributeEnumType	0..1	Optional. Attribute type for which value is requested. When absent, default Actual is assumed.
component	ComponentType	1..1	Required. Component for which the Variable is requested.
variable	VariableType	1..1	Required. Variable for which the attribute value is requested.

2.25. GetVariableResultType

Class

Class to hold results of GetVariables request.

GetVariableResultType is used by: GetVariablesResponse

Field Name	Field Type	Card.	Description
attributeStatus	GetVariableStatusEnumType	1..1	Required. Result status of getting the variable.
attributeType	AttributeEnumType	0..1	Optional. Attribute type for which value is requested. When absent, default Actual is assumed.
attributeValue	string[0..2500]	0..1	Optional. Value of requested attribute type of component variable. This field can only be empty when the given status is NOT accepted. The Configuration Variable ReportingValueSize can be used to limit GetVariableResult.attributeValue, VariableAttribute.value and EventData.actualValue. The max size of these values will always remain equal.
component	ComponentType	1..1	Required. Component for which the Variable is requested.
variable	VariableType	1..1	Required. Variable for which the attribute value is requested.
attributeStatusInfo	StatusInfoType	0..1	Optional. Detailed attribute status information.

2.26. IdTokenInfoType

Class

Contains status information about an identifier. It is advised to not stop charging for a token that expires during charging, as ExpiryDate is only used for caching purposes. If ExpiryDate is not given, the status has no end date.

IdTokenInfoType is used by: Common:AuthorizationData, AuthorizeResponse, TransactionEventResponse

Field Name	Field Type	Card.	Description
status	AuthorizationStatusEnumType	1..1	Required. Current status of the ID Token.
cacheExpiryDateTime	dateTime	0..1	Optional. Date and Time after which the token must be considered invalid.
chargingPriority	integer	0..1	Optional. Priority from a business point of view. Default priority is 0, The range is from -9 to 9. Higher values indicate a higher priority. The chargingPriority in TransactionEventResponse overrules this one.
language1	string[0..8]	0..1	Optional. Preferred user interface language of identifier user. Contains a language code as defined in [RFC5646].
evseId	integer	0..*	Optional. Only used when the IdToken is only valid for one or more specific EVSEs, not for the entire Charging Station.

Field Name	Field Type	Card.	Description
language2	string[0..8]	0..1	Optional. Second preferred user interface language of identifier user. Don't use when language1 is omitted, has to be different from language1. Contains a language code as defined in [RFC5646].
groupIdToken	IdTokenType	0..1	Optional. This contains the group identifier.
personalMessage	MessageContentType	0..1	Optional. Personal message that can be shown to the EV Driver and can be used for tariff information, user greetings etc.

2.27. IdTokenType

Class

Contains a case insensitive identifier to use for the authorization and the type of authorization to support multiple forms of identifiers.

IdTokenType is used by: Common:AuthorizationData, Common:IdTokenInfoType, RequestStartTransactionRequest, AuthorizeRequest, TransactionEventRequest, ReserveNowRequest, CustomerInformationRequest

Field Name	Field Type	Card.	Description
idToken	identifierString[0..36]	1..1	Required. IdToken is case insensitive. Might hold the hidden id of an RFID tag, but can for example also contain a UUID.
type	IdTokenEnumType	1..1	Required. Enumeration of possible idToken types.
additionalInfo	AdditionalInfoType	0..*	Optional. AdditionalInfo can be used to send extra information which can be validated by the CSMS in addition to the regular authorization with *IdToken*. *AdditionalInfo* contains one or more custom types, which need to be agreed upon by all parties involved. When AdditionalInfo is NOT implemented or a not supported AdditionalInfo.type is used, the CSMS/Charging Station MAY ignore the AdditionalInfo.

2.28. LogParametersType

Class

Generic class for the configuration of logging entries.

LogParametersType is used by: GetLogRequest

Field Name	Field Type	Card.	Description
remoteLocation	string[0..512]	1..1	Required. The URL of the location at the remote system where the log should be stored.

Field Name	Field Type	Card.	Description
oldestTimestamp	dateTime	0..1	Optional. This contains the date and time of the oldest logging information to include in the diagnostics.
latestTimestamp	dateTime	0..1	Optional. This contains the date and time of the latest logging information to include in the diagnostics.

2.29. MessageContentType

Class

Contains message details, for a message to be displayed on a Charging Station.

MessageContentType is used by: Common:IdTokenInfoType, Common:MessageInfoType, TransactionEventResponse

Field Name	Field Type	Card.	Description
format	MessageFormatEnumType	1..1	Required. Format of the message.
language	string[0..8]	0..1	Optional. Message language identifier. Contains a language code as defined in [RFC5646].
content	string[0..512]	1..1	Required. Message contents.

2.30. MessageInfoType

Class

Contains message details, for a message to be displayed on a Charging Station.

MessageInfoType is used by: SetDisplayMessageRequest, NotifyDisplayMessagesRequest

Field Name	Field Type	Card.	Description
id	integer	1..1	Required. Unique id within an exchange context. It is defined within the OCPP context as a positive Integer value (greater or equal to zero).
priority	MessagePriorityEnumType	1..1	Required. With what priority should this message be shown
state	MessageStateEnumType	0..1	Optional. During what state should this message be shown. When omitted this message should be shown in any state of the Charging Station.
startDateTime	dateTime	0..1	Optional. From what date-time should this message be shown. If omitted: directly.
endDateTime	dateTime	0..1	Optional. Until what date-time should this message be shown, after this date/time this message SHALL be removed.

Field Name	Field Type	Card.	Description
transactionId	identifierString[0..36]	0..1	Optional. During which transaction shall this message be shown. Message SHALL be removed by the Charging Station after transaction has ended.
message	MessageContentType	1..1	Required. Contains message details for the message to be displayed on a Charging Station.
display	ComponentType	0..1	Optional. When a Charging Station has multiple Displays, this field can be used to define to which Display this message belongs.

2.31. MeterValueType

Class

Collection of one or more sampled values in MeterValuesRequest and TransactionEvent. All sampled values in a MeterValue are sampled at the same point in time.

MeterValueType is used by: MeterValuesRequest, TransactionEventRequest

Field Name	Field Type	Card.	Description
timestamp	dateTime	1..1	Required. Timestamp for measured value(s).
sampledValue	SampledValueType	1..*	Required. One or more measured values

2.32. ModemType

Class

Defines parameters required for initiating and maintaining wireless communication with other devices.

ModemType is used by: BootNotificationRequest.ChargingStationType

Field Name	Field Type	Card.	Description
iccid	identifierString[0..20]	0..1	Optional. This contains the ICCID of the modem's SIM card.
imsi	identifierString[0..20]	0..1	Optional. This contains the IMSI of the modem's SIM card.

2.33. MonitoringDataType

Class

Class to hold parameters of SetVariableMonitoring request.

MonitoringDataType is used by: NotifyMonitoringReportRequest

Field Name	Field Type	Card.	Description
component	ComponentType	1..1	Required. Component for which monitoring report was requested.
variable	VariableType	1..1	Required. Variable for which monitoring report was requested.
variableMonitoring	VariableMonitoringType	1..*	Required. List of monitors for this Component-Variable pair.

2.34. NetworkConnectionProfileType

Class

The NetworkConnectionProfile defines the functional and technical parameters of a communication link.

NetworkConnectionProfileType is used by: SetNetworkProfileRequest

Field Name	Field Type	Card.	Description
ocppVersion	OCPPVersionEnumType	1..1	Required. Defines the OCPP version used for this communication function. This field is ignored, since the OCPP version to use is determined during the websocket handshake.
ocppTransport	OCPPTransportEnumType	1..1	Required. Defines the transport protocol (e.g. SOAP or JSON). Note: SOAP is not supported in OCPP 2.0, but is supported by other versions of OCPP.
ocppCsmsUrl	string[0..512]	1..1	Required. URL of the CSMS(s) that this Charging Station communicates with, without the Charging Station identity part. The SecurityCtrlr.Identity field is appended to *ocppCsmsUrl* to provide the full websocket URL
messageTimeout	integer	1..1	Required. Duration in seconds before a message send by the Charging Station via this network connection timesout. The best setting depends on the underlying network and response times of the CSMS. If you are looking for a some guideline: use 30 seconds as a starting point.
securityProfile	integer	1..1	Required. This field specifies the security profile used when connecting to the CSMS with this NetworkConnectionProfile.
ocppInterface	OCPPInterfaceEnumType	1..1	Required. Applicable Network Interface. Charging Station is allowed to use a different network interface to connect if the given one does not work
vpn	VPNType	0..1	Optional. Settings to be used to set up the VPN connection

Field Name	Field Type	Card.	Description
apn	APNType	0..1	Optional. Collection of configuration data needed to make a data-connection over a cellular network.

2.35. OCSPRequestDataType

Class

OCSPRequestDataType is used by: AuthorizeRequest, GetCertificateStatusRequest

Field Name	Field Type	Card.	Description
hashAlgorithm	HashAlgorithmEnumType	1..1	Required. Used algorithms for the hashes provided.
issuerNameHash	identifierString[0..128]	1..1	Required. The hash of the issuer's distinguished name (DN), that must be calculated over the DER encoding of the issuer's name field in the certificate being checked.
issuerKeyHash	string[0..128]	1..1	Required. The hash of the DER encoded public key: the value (excluding tag and length) of the subject public key field in the issuer's certificate.
serialNumber	identifierString[0..40]	1..1	Required. The string representation of the hexadecimal value of the serial number without the prefix "0x" and without leading zeroes.
responderURL	string[0..512]	1..1	Required. This contains the responder URL (Case insensitive).

2.36. RelativeTimeIntervalType

Class

RelativeTimeIntervalType is used by: Common:SalesTariffEntryType

Field Name	Field Type	Card.	Description
start	integer	1..1	Required. Start of the interval, in seconds from NOW.
duration	integer	0..1	Optional. Duration of the interval, in seconds.

2.37. ReportDataType

Class

Class to report components, variables and variable attributes and characteristics.

ReportDataType is used by: NotifyReportRequest

Field Name	Field Type	Card.	Description
component	ComponentType	1..1	Required. Component for which a report of Variable is requested.
variable	VariableType	1..1	Required. Variable for which report is requested.
variableAttribute	VariableAttributeType	1..4	Required. Attribute data of a variable.
variableCharacteristics	VariableCharacteristicsType	0..1	Optional. Fixed read-only parameters of a variable.

2.38. SalesTariffEntryType

Class

SalesTariffEntryType is used by: Common:SalesTariffType

Field Name	Field Type	Card.	Description
ePriceLevel	integer, 0 < = val	0..1	Optional. Defines the price level of this SalesTariffEntry (referring to NumEPriceLevels). Small values for the EPriceLevel represent a cheaper TariffEntry. Large values for the EPriceLevel represent a more expensive TariffEntry.
relativeTimeInterval	RelativeTimeIntervalType	1..1	Required. Defines the time interval the SalesTariffEntry is valid for, based upon relative times.
consumptionCost	ConsumptionCostType	0..3	Optional. Defines additional means for further relative price information and/or alternative costs.

2.39. SalesTariffType

Class

NOTE This dataType is based on dataTypes from ISO 15118-2.

SalesTariffType is used by: Common:ChargingScheduleType

Field Name	Field Type	Card.	Description
id	integer	1..1	Required. SalesTariff identifier used to identify one sales tariff. An SAID remains a unique identifier for one schedule throughout a charging session.
salesTariffDescription	string[0..32]	0..1	Optional. A human readable title/short description of the sales tariff e.g. for HMI display purposes.
numEPriceLevels	integer	0..1	Optional. Defines the overall number of distinct price levels used across all provided SalesTariff elements.

Field Name	Field Type	Card.	Description
salesTariffEntry	SalesTariffEntryType	1..1024	Required. Encapsulating element describing all relevant details for one time interval of the SalesTariff. The number of SalesTariffEntry elements is limited by the parameter maxScheduleTuples.

2.40. SampledValueType

Class

Single sampled value in MeterValues. Each value can be accompanied by optional fields.

To save on mobile data usage, default values of all of the optional fields are such that. The value without any additional fields will be interpreted, as a register reading of active import energy in Wh (Watt-hour) units.

SampledValueType is used by: Common:MeterValueType

Field Name	Field Type	Card.	Description
value	decimal	1..1	Required. Indicates the measured value.
context	ReadingContextEnumType	0..1	Optional. Type of detail value: start, end or sample. Default = "Sample.Periodic"
measurand	MeasurandEnumType	0..1	Optional. Type of measurement. Default = "Energy.Active.Import.Register"
phase	PhaseEnumType	0..1	Optional. Indicates how the measured value is to be interpreted. For instance between L1 and neutral (L1-N) Please note that not all values of phase are applicable to all Measurands. When phase is absent, the measured value is interpreted as an overall value.
location	LocationEnumType	0..1	Optional. Indicates where the measured value has been sampled. Default = "Outlet"
signedMeterValue	SignedMeterValueType	0..1	Optional. Contains the MeterValueSignature with sign/encoding method information.
unitOfMeasure	UnitOfMeasureType	0..1	Optional. Represents a UnitOfMeasure including a multiplier

2.41. SetMonitoringDataType

Class

Class to hold parameters of SetVariableMonitoring request.

SetMonitoringDataType is used by: SetVariableMonitoringRequest

Field Name	Field Type	Card.	Description
id	integer	0..1	Optional. An id SHALL only be given to replace an existing monitor. The Charging Station handles the generation of id's for new monitors.
transaction	boolean	0..1	Optional. Monitor only active when a transaction is ongoing on a component relevant to this transaction. Default = false.
value	decimal	1..1	Required. Value for threshold or delta monitoring. For Periodic or PeriodicClockAligned this is the interval in seconds.
type	MonitorEnumType	1..1	Required. The type of this monitor, e.g. a threshold, delta or periodic monitor.
severity	integer	1..1	Required. The severity that will be assigned to an event that is triggered by this monitor. The severity range is 0-9, with 0 as the highest and 9 as the lowest severity level. The severity levels have the following meaning: **0-Danger** Indicates lives are potentially in danger. Urgent attention is needed and action should be taken immediately. **1-Hardware Failure** Indicates that the Charging Station is unable to continue regular operations due to Hardware issues. Action is required. **2-System Failure** Indicates that the Charging Station is unable to continue regular operations due to software or minor hardware issues. Action is required. **3-Critical** Indicates a critical error. Action is required. **4-Error** Indicates a non-urgent error. Action is required. **5-Alert** Indicates an alert event. Default severity for any type of monitoring event. **6-Warning** Indicates a warning event. Action may be required. **7-Notice** Indicates an unusual event. No immediate action is required. **8-Informational** Indicates a regular operational event. May be used for reporting, measuring throughput, etc. No action is required. **9-Debug** Indicates information useful to developers for debugging, not useful during operations.
component	ComponentType	1..1	Required. Component for which monitor is set.
variable	VariableType	1..1	Required. Variable for which monitor is set.

2.42. SetMonitoringResultType

Class

Class to hold result of SetVariableMonitoring request.

SetMonitoringResultType is used by: SetVariableMonitoringResponse

Field Name	Field Type	Card.	Description
id	integer	0..1	Optional. Id given to the VariableMonitor by the Charging Station. The Id is only returned when status is accepted. Installed VariableMonitors should have unique id's but the id's of removed Installed monitors should have unique id's but the id's of removed monitors MAY be reused.
status	SetMonitoringStatusEnumType	1..1	Required. Status is OK if a value could be returned. Otherwise this will indicate the reason why a value could not be returned.
type	MonitorEnumType	1..1	Required. The type of this monitor, e.g. a threshold, delta or periodic monitor.

Field Name	Field Type	Card.	Description
severity	integer	1..1	Required. The severity that will be assigned to an event that is triggered by this monitor. The severity range is 0-9, with 0 as the highest and 9 as the lowest severity level. The severity levels have the following meaning: **0-Danger** Indicates lives are potentially in danger. Urgent attention is needed and action should be taken immediately. **1-Hardware Failure** Indicates that the Charging Station is unable to continue regular operations due to Hardware issues. Action is required. **2-System Failure** Indicates that the Charging Station is unable to continue regular operations due to software or minor hardware issues. Action is required. **3-Critical** Indicates a critical error. Action is required. **4-Error** Indicates a non-urgent error. Action is required. **5-Alert** Indicates an alert event. Default severity for any type of monitoring event. **6-Warning** Indicates a warning event. Action may be required. **7-Notice** Indicates an unusual event. No immediate action is required. **8-Informational** Indicates a regular operational event. May be used for reporting, measuring throughput, etc. No action is required. **9-Debug** Indicates information useful to developers for debugging, not useful during operations.
component	ComponentType	1..1	Required. Component for which status is returned.
variable	VariableType	1..1	Required. Variable for which status is returned.
statusInfo	StatusInfoType	0..1	Optional. Detailed status information.

2.43. SetVariableDataType

Class

SetVariableDataType is used by: SetVariablesRequest

Field Name	Field Type	Card.	Description
attributeType	AttributeEnumType	0..1	Optional. Type of attribute: Actual, Target, MinSet, MaxSet. Default is Actual when omitted.

Field Name	Field Type	Card.	Description
attributeValue	string[0..1000]	1..1	Required. Value to be assigned to attribute of variable. The value is allowed to be an empty string (""). The Configuration Variable ConfigurationValueSize can be used to limit SetVariableData.attributeValue and VariableCharacteristics.valueList. The max size of these values will always remain equal.
component	ComponentType	1..1	Required. The component for which the variable data is set.
variable	VariableType	1..1	Required. Specifies the that needs to be set.

2.44. SetVariableResultType

Class

SetVariableResultType is used by: SetVariablesResponse

Field Name	Field Type	Card.	Description
attributeType	AttributeEnumType	0..1	Optional. Type of attribute: Actual, Target, MinSet, MaxSet. Default is Actual when omitted.
attributeStatus	SetVariableStatusEnumType	1..1	Required. Result status of setting the variable.
component	ComponentType	1..1	Required. The component for which result is returned.
variable	VariableType	1..1	Required. The variable for which the result is returned.
attributeStatusInfo	StatusInfoType	0..1	Optional. Detailed attribute status information.

2.45. SignedMeterValueType

Class

Represent a signed version of the meter value.

SignedMeterValueType is used by: Common:SampledValueType

Field Name	Field Type	Card.	Description
signedMeterData	string[0..2500]	1..1	Required. Base64 encoded, contains the signed data which might contain more then just the meter value. It can contain information like timestamps, reference to a customer etc.
signingMethod	string[0..50]	1..1	Required. Method used to create the digital signature.
encodingMethod	string[0..50]	1..1	Required. Method used to encode the meter values before applying the digital signature algorithm.

Field Name	Field Type	Card.	Description
publicKey	string[0..2500]	1..1	Required. Base64 encoded, sending depends on configuration variable *PublicKeyWithSignedMeterValue*.

2.46. StatusInfoType

Class

Element providing more information about the status.

StatusInfoType is used by: Common:ClearMonitoringResultType, BootNotificationResponse, CancelReservationResponse, TriggerMessageResponse, UnlockConnectorResponse, UpdateFirmwareResponse, ClearDisplayMessageResponse ,Get15118EVCertificateResponse, GetCompositeScheduleResponse, ChangeAvailabilityResponse, GetLogResponse, ClearChargingProfileResponse, NotifyEVChargingNeedsResponse, ClearCacheResponse, NotifyEVChargingScheduleResponse, RequestStartTransactionResponse, RequestStopTransactionResponse, SetChargingProfileResponse,SetDisplayMessageResponse, SetNetworkProfileResponse, SignCertificateResponse, DataTransferResponse, CertificateSignedResponse, DeleteCertificateResponse, GetChargingProfilesResponse, GetInstalledCertificateIdsResponse, InstallCertificateResponse, GetBaseReportResponse, GetMonitoringReportResponse, GetReportResponse, GetVariablesResponse.GetVariableResultType, ReserveNowResponse, SetMonitoringBaseResponse, SetMonitoringLevelResponse, SetVariableMonitoringResponse.SetMonitoringResultType, SetVariablesResponse.SetVariableResultType, PublishFirmwareResponse, GetCertificateStatusResponse, ResetResponse, GetDisplayMessagesResponse, CustomerInformationResponse, SendLocalListResponse

Field Name	Field Type	Card.	Description
reasonCode	string[0..20]	1..1	Required. A predefined code for the reason why the status is returned in this response. The string is case-insensitive.
additionalInfo	string[0..512]	0..1	Optional. Additional text to provide detailed information.

2.47. TransactionType

Class

TransactionType is used by: TransactionEventRequest

Field Name	Field Type	Card.	Description
transactionId	identifierString[0..36]	1..1	Required. This contains the Id of the transaction.
chargingState	ChargingStateEnumType	0..1	Optional. Current charging state, is required when state has changed.
timeSpentCharging	integer	0..1	Optional. Contains the total time that energy flowed from EVSE to EV during the transaction (in seconds). Note that timeSpentCharging is smaller or equal to the duration of the transaction.

Field Name	Field Type	Card.	Description
stoppedReason	ReasonEnumType	0..1	Optional. The *stoppedReason* is the reason/event that initiated the process of stopping the transaction. It will normally be the user stopping authorization via card (Local or MasterPass) or app (Remote), but it can also be CSMS revoking authorization (DeAuthorized), or disconnecting the EV when TxStopPoint = EVConnected (EVDisconnected). Most other reasons are related to technical faults or energy limitations. MAY only be omitted when *stoppedReason* is "Local"
remoteStartId	integer	0..1	Optional. The ID given to remote start request (RequestStartTransactionRequest). This enables to CSMS to match the started transaction to the given start request.

2.48. UnitOfMeasureType

Class

Represents a UnitOfMeasure with a multiplier

UnitOfMeasureType is used by: Common:SampledValueType

Field Name	Field Type	Card.	Description
unit	string[0..20]	0..1	Optional. Unit of the value. Default = "Wh" if the (default) measurand is an "Energy" type. This field SHALL use a value from the list Standardized Units of Measurements in Part 2 Appendices. If an applicable unit is available in that list, otherwise a "custom" unit might be used.
multiplier	integer	0..1	Optional. Multiplier, this value represents the exponent to base 10. I.e. multiplier 3 means 10 raised to the 3rd power. Default is 0.

2.49. VariableAttributeType

Class

Attribute data of a variable.

VariableAttributeType is used by: NotifyReportRequest.ReportDataType

Field Name	Field Type	Card.	Description
type	AttributeEnumType	0..1	Optional. Attribute: Actual, MinSet, MaxSet, etc. Defaults to Actual if absent.

Field Name	Field Type	Card.	Description
value	string[0..2500]	0..1	Optional. Value of the attribute. May only be omitted when mutability is set to 'WriteOnly'. The Configuration Variable ReportingValueSize can be used to limit GetVariableResult.attributeValue, VariableAttribute.value and EventData.actualValue. The max size of these values will always remain equal.
mutability	MutabilityEnumType	0..1	Optional. Defines the mutability of this attribute. Default is ReadWrite when omitted.
persistent	boolean	0..1	Optional. If true, value will be persistent across system reboots or power down. Default when omitted is false.
constant	boolean	0..1	Optional. If true, value that will never be changed by the Charging Station at runtime. Default when omitted is false.

2.50. VariableCharacteristicsType

Class

Fixed read-only parameters of a variable.

VariableCharacteristicsType is used by: NotifyReportRequest.ReportDataType

Field Name	Field Type	Card.	Description
unit	string[0..16]	0..1	Optional. Unit of the variable. When the transmitted value has a unit, this field SHALL be included.
dataType	DataEnumType	1..1	Required. Data type of this variable.
minLimit	decimal	0..1	Optional. Minimum possible value of this variable.
maxLimit	decimal	0..1	Optional. Maximum possible value of this variable. When the datatype of this Variable is String, OptionList, SequenceList or MemberList, this field defines the maximum length of the (CSV) string.

Field Name	Field Type	Card.	Description
valuesList	string[0..1000]	0..1	Optional. Mandatory when *dataType* = OptionList, MemberList or SequenceList. *valuesList* specifies the allowed values for the type. * OptionList: The (Actual) Variable value must be a single value from the reported (CSV) enumeration list. * MemberList: The (Actual) Variable value may be an (unordered) (sub-)set of the reported (CSV) valid values list. * SequenceList: The (Actual) Variable value may be an ordered (priority, etc) (sub-)set of the reported (CSV) valid values. This is a comma separated list. The Configuration Variable ConfigurationValueSize can be used to limit SetVariableData.attributeValue and VariableCharacteristics.valueList. The max size of these values will always remain equal.
supportsMonitoring	boolean	1..1	Required. Flag indicating if this variable supports monitoring.

2.51. VariableMonitoringType

Class

A monitoring setting for a variable.

VariableMonitoringType is used by: NotifyMonitoringReportRequest.MonitoringDataType

Field Name	Field Type	Card.	Description
id	integer	1..1	Required. Identifies the monitor.
transaction	boolean	1..1	Required. Monitor only active when a transaction is ongoing on a component relevant to this transaction.
value	decimal	1..1	Required. Value for threshold or delta monitoring. For Periodic or PeriodicClockAligned this is the interval in seconds.
type	MonitorEnumType	1..1	Required. The type of this monitor, e.g. a threshold, delta or periodic monitor.

Field Name	Field Type	Card.	Description
severity	integer	1..1	Required. The severity that will be assigned to an event that is triggered by this monitor. The severity range is 0-9, with 0 as the highest and 9 as the lowest severity level. The severity levels have the following meaning: **0-Danger** Indicates lives are potentially in danger. Urgent attention is needed and action should be taken immediately. **1-Hardware Failure** Indicates that the Charging Station is unable to continue regular operations due to Hardware issues. Action is required. **2-System Failure** Indicates that the Charging Station is unable to continue regular operations due to software or minor hardware issues. Action is required. **3-Critical** Indicates a critical error. Action is required. **4-Error** Indicates a non-urgent error. Action is required. **5-Alert** Indicates an alert event. Default severity for any type of monitoring event. **6-Warning** Indicates a warning event. Action may be required. **7-Notice** Indicates an unusual event. No immediate action is required. **8-Informational** Indicates a regular operational event. May be used for reporting, measuring throughput, etc. No action is required. **9-Debug** Indicates information useful to developers for debugging, not useful during operations.

2.52. VariableType

Class

Reference key to a component-variable.

VariableType is used by: Common:ComponentVariableType, GetVariablesRequest.GetVariableDataType, GetVariablesResponse.GetVariableResultType, NotifyMonitoringReportRequest.MonitoringDataType, NotifyReportRequest.ReportDataType, SetVariableMonitoringRequest.SetMonitoringDataType, SetVariableMonitoringResponse.SetMonitoringResultType, SetVariablesRequest.SetVariableDataType, SetVariablesResponse.SetVariableResultType, NotifyEventRequest.EventDataType

Field Name	Field Type	Card.	Description
name	identifierString[0..50]	1..1	Required. Name of the variable. Name should be taken from the list of standardized variable names whenever possible. Case Insensitive. strongly advised to use Camel Case.
instance	identifierString[0..50]	0..1	Optional. Name of instance in case the variable exists as multiple instances. Case Insensitive. strongly advised to use Camel Case.

2.53. VPNType

Class

VPN Configuration settings

VPNType is used by: SetNetworkProfileRequest.NetworkConnectionProfileType

Field Name	Field Type	Card.	Description
server	string[0..512]	1..1	Required. VPN Server Address
user	string[0..20]	1..1	Required. VPN User
group	string[0..20]	0..1	Optional. VPN group.
password	string[0..20]	1..1	Required. VPN Password.
key	string[0..255]	1..1	Required. VPN shared secret.
type	VPNEnumType	1..1	Required. Type of VPN

3. Enumerations

3.1. APNAuthenticationEnumType

Enumeration

APNAuthenticationEnumType is used by: setNetworkProfile:SetNetworkProfileRequest.APNType

Value	Description
CHAP	Use CHAP authentication
NONE	Use no authentication
PAP	Use PAP authentication
AUTO	Sequentially try CHAP, PAP, NONE.

3.2. AttributeEnumType

Enumeration

AttributeEnumType is used by: Common:VariableAttributeType, getVariables:GetVariablesRequest.GetVariableDataType, getVariables:GetVariablesResponse.GetVariableResultType, setVariables:SetVariablesRequest.SetVariableDataType, setVariables:SetVariablesResponse.SetVariableResultType

Value	Description
Actual	The actual value of the variable.
Target	The target value for this variable.
MinSet	The minimal allowed value for this variable
MaxSet	Thne maximum allowed value for this variable

3.3. AuthorizationStatusEnumType

Enumeration

Status of an authorization response.

AuthorizationStatusEnumType is used by: Common:IdTokenInfoType

Value	Description
Accepted	Identifier is allowed for charging.
Blocked	Identifier has been blocked. Not allowed for charging.
ConcurrentTx	Identifier is already involved in another transaction and multiple transactions are not allowed. (Only relevant for the response to a transactionEventRequest(eventType=Started).)

Value	Description
Expired	Identifier has expired. Not allowed for charging.
Invalid	Identifier is invalid. Not allowed for charging.
NoCredit	Identifier is valid, but EV Driver doesn't have enough credit to start charging. Not allowed for charging.
NotAllowedTypeEVSE	Identifier is valid, but not allowed to charge at this type of EVSE.
NotAtThisLocation	Identifier is valid, but not allowed to charge at this location.
NotAtThisTime	Identifier is valid, but not allowed to charge at this location at this time.
Unknown	Identifier is unknown. Not allowed for charging.

3.4. AuthorizeCertificateStatusEnumType

Enumeration

Status of the EV Contract certificate.

AuthorizeCertificateStatusEnumType is used by: authorize:AuthorizeResponse

Value	Description
Accepted	Positive response
SignatureError	<not used>
CertificateExpired	If the contract certificate in the AuthorizeRequest is expired.
CertificateRevoked	If the Charging Station or CSMS determine (via a CRL or OCSP response) that the contract certificate in the AuthorizeRequest is marked as revoked.
NoCertificateAvailable	<not used>
CertChainError	If the contract certificate contained in the AuthorizeRequest message is not valid.
ContractCancelled	If the EMAID provided by EVCC is invalid, unknown, expired or blocked.

3.5. BootReasonEnumType

Enumeration

BootReasonEnumType is used by: bootNotification:BootNotificationRequest

Value	Description
ApplicationReset	The Charging Station rebooted due to an application error.
FirmwareUpdate	The Charging Station rebooted due to a firmware update.

Value	Description
LocalReset	The Charging Station rebooted due to a local reset command.
PowerUp	The Charging Station powered up and registers itself with the CSMS.
RemoteReset	The Charging Station rebooted due to a remote reset command.
ScheduledReset	The Charging Station rebooted due to a scheduled reset command.
Triggered	Requested by the CSMS via a TriggerMessage
Unknown	The boot reason is unknown.
Watchdog	The Charging Station rebooted due to an elapsed watchdog timer.

3.6. CancelReservationStatusEnumType

Enumeration

Status in CancelReservationResponse.

CancelReservationStatusEnumType is used by: cancelReservation:CancelReservationResponse

Value	Description
Accepted	Reservation for the identifier has been canceled.
Rejected	Reservation could not be canceled, because there is no reservation active for the identifier.

3.7. CertificateActionEnumType

Enumeration

CertificateActionEnumType is used by: get15118EVCertificate:Get15118EVCertificateRequest

Value	Description
Install	Install the provided certificate.
Update	Update the provided certificate.

3.8. CertificateSignedStatusEnumType

Enumeration

CertificateSignedStatusEnumType is used by: certificateSigned:CertificateSignedResponse

Value	Description
Accepted	Signed certificate is valid.

Value	Description
Rejected	Signed certificate is invalid.

3.9. CertificateSigningUseEnumType

Enumeration

CertificateSigningUseEnumType is used by: signCertificate:SignCertificateRequest, certificateSigned:CertificateSignedRequest

Value	Description
ChargingStationCertificate	Client side certificate used by the Charging Station to connect the the CSMS.
V2GCertificate	Use for certificate for 15118 connections. This means that the certificate should be derived from the V2G root.

3.10. ChangeAvailabilityStatusEnumType

Enumeration

Status returned in response to ChangeAvailabilityRequest.

ChangeAvailabilityStatusEnumType is used by: changeAvailability:ChangeAvailabilityResponse

Value	Description
Accepted	Request has been accepted and will be executed.
Rejected	Request has not been accepted and will not be executed.
Scheduled	Request has been accepted and will be executed when transaction(s) in progress have finished.

3.11. ChargingLimitSourceEnumType

Enumeration

Enumeration for indicating from which source a charging limit originates.

ChargingLimitSourceEnumType is used by: notifyChargingLimit:NotifyChargingLimitRequest.ChargingLimitType, clearedChargingLimit:ClearedChargingLimitRequest, getChargingProfiles:GetChargingProfilesRequest.ChargingProfileCriterionType, reportChargingProfiles:ReportChargingProfilesRequest

Value	Description
EMS	Indicates that an Energy Management System has sent a charging limit.
Other	Indicates that an external source, not being an EMS or system operator, has sent a charging limit.
SO	Indicates that a System Operator (DSO or TSO) has sent a charging limit.
CSO	Indicates that the CSO has set this charging profile.

3.12. ChargingProfileKindEnumType

Enumeration

Kind of charging profile.

ChargingProfileKindEnumType is used by: Common:ChargingProfileType

Value	Description
Absolute	Schedule periods are relative to a fixed point in time defined in the schedule. This requires that startSchedule is set to a starting point in time.
Recurring	The schedule restarts periodically at the first schedule period. To be most useful, this requires that startSchedule is set to a starting point in time.
Relative	Charging schedule periods should start when the EVSE is ready to deliver energy. i.e. when the EV driver is authorized and the EV is connected. When a ChargingProfile is received for a transaction that is already charging, then the charging schedule periods should remain relative to the PowerPathClosed moment. No value for *startSchedule* should be supplied.

3.13. ChargingProfilePurposeEnumType

Enumeration

Purpose of the charging profile.

ChargingProfilePurposeEnumType is used by: Common:ChargingProfileType, clearChargingProfile:ClearChargingProfileRequest.ClearChargingProfileType, getChargingProfiles:GetChargingProfilesRequest.ChargingProfileCriterionType

Value	Description
ChargingStationExternalConstraints	Additional constraints that will be incorporated into a local power schedule. Only valid for a Charging Station. Therefore evse.Id MUST be 0 in the SetChargingProfileRequest message.
ChargingStationMaxProfile	Configuration for the maximum power or current available for an entire Charging Station.
TxDefaultProfile	Default profile that can be configured in the Charging Station. When a new transaction is started, this profile SHALL be used, unless it was a transaction that was started by a RequestStartTransactionRequest with a ChargingProfile that is accepted by the Charging Station.
TxProfile	Profile with constraints to be imposed by the Charging Station on the current transaction, or on a new transaction when this is started via a RequestStartTransactionRequest with a ChargingProfile. A profile with this purpose SHALL cease to be valid when the transaction terminates.

3.14. ChargingProfileStatusEnumType

Enumeration

Status returned in response to SetChargingProfileRequest.

ChargingProfileStatusEnumType is used by: setChargingProfile:SetChargingProfileResponse

Value	Description
Accepted	Request has been accepted and will be executed.
Rejected	Request has not been accepted and will not be executed.

3.15. ChargingRateUnitEnumType

Enumeration

Unit in which a charging schedule is defined.

ChargingRateUnitEnumType is used by: Common:ChargingScheduleType, Common:CompositeScheduleType, getCompositeSchedule:GetCompositeScheduleRequest

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

3.16. ChargingStateEnumType

Enumeration

The state of the charging process.

ChargingStateEnumType is used by: transactionEvent:TransactionEventRequest.TransactionType

Value	Description
Charging	The contactor of the Connector is closed and energy is flowing to between EVSE and EV.
EVConnected	There is a connection between EV and EVSE, in case the protocol used between EV and the Charging Station can detect a connection, the protocol needs to detect this for the state to become active. The connection can either be wired or wireless.
SuspendedEV	When the EV is connected to the EVSE and the EVSE is offering energy but the EV is not taking any energy.
SuspendedEVSE	When the EV is connected to the EVSE but the EVSE is not offering energy to the EV, e.g. due to a smart charging restriction, local supply power constraints, or when charging has stopped because of the authorization status in the response to a transactionEventRequest indicating that charging is not allowed etc.
Idle	There is no connection between EV and EVSE.

3.17. ClearCacheStatusEnumType

Enumeration

Status returned in response to ClearCacheRequest.

ClearCacheStatusEnumType is used by: clearCache:ClearCacheResponse

Value	Description
Accepted	Command has been executed.
Rejected	Command has not been executed.

3.18. ClearChargingProfileStatusEnumType

Enumeration

Status returned in response to ClearChargingProfileRequest.

ClearChargingProfileStatusEnumType is used by: clearChargingProfile:ClearChargingProfileResponse

Value	Description
Accepted	Request has been accepted and will be executed.
Unknown	No Charging Profile(s) were found matching the request.

3.19. ClearMessageStatusEnumType

Enumeration

Result for a ClearDisplayMessageRequest as used in a ClearDisplayMessageResponse.

ClearMessageStatusEnumType is used by: clearDisplayMessage:ClearDisplayMessageResponse

Value	Description
Accepted	Request successfully executed: message cleared.
Unknown	Given message (based on the id) not known.

3.20. ClearMonitoringStatusEnumType

Enumeration

ClearMonitoringStatusEnumType is used by: Common:ClearMonitoringResultType

Value	Description
Accepted	Monitor successfully cleared.
Rejected	Clearing of monitor rejected.
NotFound	Monitor Id is not found.

3.21. ComponentCriterionEnumType

Enumeration

ComponentCriterionEnumType is used by: getReport:GetReportRequest

Value	Description
Active	Components that are active, i.e. having *Active* = 1
Available	Components that are available, i.e. having *Available* = 1
Enabled	Components that are enabled, i.e. having *Enabled* = 1
Problem	Components that reported a problem, i.e. having *Problem* = 1

3.22. ConnectorEnumType

Enumeration

Allowed values of ConnectorCode.

NOTE	This enumeration does not attempt to include every possible power connector type worldwide as an individual type, but to specifically define those that are known to be in use (or likely to be in use) in the Charging Stations using the OCPP protocol. In particular, many of the very large number of domestic electrical sockets designs in use in many countries are excluded, unless there is evidence that they are or are likely to be approved for use on Charging Stations in some jurisdictions (e.g. as secondary connectors for charging light EVs such as electric scooters). These light connector types can be represented with the enumeration value Other1PhMax16A. Similarly, any single phase connector not otherwise enumerated that is rated for 16A or over should be reported as Other1PhOver16A. All 3 phase connector types not explicitly enumerated should be represented as Other3Ph.

ConnectorEnumType is used by: reserveNow:ReserveNowRequest

Value	Description
cCCS1	Combined Charging System 1 (captive cabled) a.k.a. Combo 1
cCCS2	Combined Charging System 2 (captive cabled) a.k.a. Combo 2
cG105	JARI G105-1993 (captive cabled) a.k.a. CHAdeMO
cTesla	Tesla Connector (captive cabled)
cType1	IEC62196-2 Type 1 connector (captive cabled) a.k.a. J1772
cType2	IEC62196-2 Type 2 connector (captive cabled) a.k.a. Mennekes connector
s309-	1P-16A 16A 1 phase IEC60309 socket
s309-	1P-32A 32A 1 phase IEC60309 socket
s309-	3P-16A 16A 3 phase IEC60309 socket
s309-	3P-32A 32A 3 phase IEC60309 socket

Value	Description
sBS1361	UK domestic socket a.k.a. 13Amp
sCEE-	7-7 CEE 7/7 16A socket. May represent 7/4 & 7/5 a.k.a Schuko
sType2	IEC62196-2 Type 2 socket a.k.a. Mennekes connector
sType3	IEC62196-2 Type 2 socket a.k.a. Scame
Other1PhMax16A	Other single phase (domestic) sockets not mentioned above, rated at no more than 16A. CEE7/17, AS3112,
NEMA	5-15, NEMA 5-20, JISC8303, TIS166, SI 32, CPCS-CCC, SEV1011, etc.
Other1PhOver16A	Other single phase sockets not mentioned above (over 16A)
Other3Ph	Other 3 phase sockets not mentioned above. NEMA14-30, NEMA14-50.
Pan	Pantograph connector
wInductive	Wireless inductively coupled connection (generic)
wResonant	Wireless resonant coupled connection (generic)
Undetermined	Yet to be determined (e.g. before plugged in)
Unknown	Unknown; not determinable

3.23. ConnectorStatusEnumType

Enumeration

A status can be reported for the Connector of an EVSE of a Charging Station. States considered Operative are: Available, Reserved and Occupied. States considered Inoperative are: Unavailable, Faulted.

ConnectorStatusEnumType is used by: statusNotification:StatusNotificationRequest

Value	Description
Available	When a Connector becomes available for a new User (Operative)
Occupied	When a Connector becomes occupied, so it is not available for a new EV driver. (Operative)
Reserved	When a Connector becomes reserved as a result of ReserveNow command (Operative)
Unavailable	When a Connector becomes unavailable as the result of a Change Availability command or an event upon which the Charging Station transitions to unavailable at its discretion. Upon receipt of ChangeAvailability message command, the status MAY change immediately or the change MAY be scheduled. When scheduled, StatusNotification SHALL be send when the availability change becomes effective (Inoperative)
Faulted	When a Connector (or the EVSE or the entire Charging Station it belongs to) has reported an error and is not available for energy delivery. (Inoperative).

3.24. CostKindEnumType

Enumeration

CostKindEnumType is used by: Common:CostType

Value	Description
CarbonDioxideEmission	Absolute value. Carbon Dioxide emissions, in grams per kWh.
RelativePricePercentage	Relative value. Price per kWh, as percentage relative to the maximum price stated in any of all tariffs indicated to the EV.
RenewableGenerationPercentage	Relative value. Percentage of renewable generation within total generation.

3.25. CustomerInformationStatusEnumType

Enumeration

Status in CancelReservationResponse.

CustomerInformationStatusEnumType is used by: customerInformation:CustomerInformationResponse

Value	Description
Accepted	The Charging Station accepted the message.
Rejected	When the Charging Station is in a state where it cannot process this request.
Invalid	In a request to the Charging Station no reference to a customer is included.

3.26. DataEnumType

Enumeration

DataEnumType is used by: Common:VariableCharacteristicsType

Value	Description
string	This variable is of the type string.
decimal	This variable is of the type decimal.
integer	This variable is of the type integer.
dateTime	DateTime following the [RFC3339] specification.
boolean	This variable is of the type boolean.
OptionList	Supported/allowed values for a single choice, enumerated, text variable.
SequenceList	Supported/allowed values for an ordered sequence variable.

Value	Description
MemberList	Supported/allowed values for a mathematical set variable.

3.27. DataTransferStatusEnumType

Enumeration

Status in DataTransferResponse.

DataTransferStatusEnumType is used by: dataTransfer:DataTransferResponse

Value	Description
Accepted	Message has been accepted and the contained request is accepted.
Rejected	Message has been accepted but the contained request is rejected.
UnknownMessageId	Message could not be interpreted due to unknown messageId string.
UnknownVendorId	Message could not be interpreted due to unknown vendorId string.

3.28. DeleteCertificateStatusEnumType

Enumeration

DeleteCertificateStatusEnumType is used by: deleteCertificate:DeleteCertificateResponse

Value	Description
Accepted	Normal successful completion (no errors).
Failed	The Charging Station either failed to remove the certificate or rejected the request. A Charging Station may reject the request to prevent the deletion of a certificate, if it is the last one from its certificate type.
NotFound	Requested resource not found.

3.29. DisplayMessageStatusEnumType

Enumeration

Result for a SetDisplayMessageRequest as used in a SetDisplayMessageResponse.

DisplayMessageStatusEnumType is used by: setDisplayMessage:SetDisplayMessageResponse

Value	Description
Accepted	Request to display message accepted.
NotSupportedMessageFormat	None of the formats in the given message are supported.

Value	Description
Rejected	Request cannot be handled.
NotSupportedPriority	The given MessagePriority not supported for displaying messages by Charging Station.
NotSupportedState	The given MessageState not supported for displaying messages by Charging Station.
UnknownTransaction	Given Transaction not known/ongoing.

3.30. EnergyTransferModeEnumType

Enumeration

Enumeration of energy transfer modes.

EnergyTransferModeEnumType is used by: Common:ChargingNeedsType

Value	Description
DC	DC charging.
AC_single_phase	AC single phase charging according to IEC 62196.
AC_two_phase	AC two phase charging according to IEC 62196.
AC_three_phase	AC three phase charging according to IEC 62196.

3.31. EventNotificationEnumType

Enumeration

Specifies the event notification type of the message.

EventNotificationEnumType is used by: notifyEvent:NotifyEventRequest.EventDataType

Value	Description
HardWiredNotification	The software implemented by the manufacturer triggered a hardwired notification.
HardWiredMonitor	Triggered by a monitor, which is hardwired by the manufacturer.
PreconfiguredMonitor	Triggered by a monitor, which is preconfigured by the manufacturer.
CustomMonitor	Triggered by a monitor, which is set with the setvariablemonitoringrequest message by the Charging Station Operator.

3.32. EventTriggerEnumType

Enumeration

EventTriggerEnumType is used by: notifyEvent:NotifyEventRequest.EventDataType

Value	Description
Alerting	Monitored variable has passed an Lower or Upper Threshold. Also used as trigger type for a HardWiredNotification.
Delta	Delta Monitored Variable value has changed by more than specified amount
Periodic	Periodic Monitored Variable has been sampled for reporting at the specified interval

3.33. FirmwareStatusEnumType

Enumeration

Status of a firmware download.

A value with "Intermediate state" in the description, is an intermediate state, update process is not finished.

A value with "Failure end state" in the description, is an end state, update process has stopped, update failed.

A value with "Successful end state" in the description, is an end state, update process has stopped, update successful.

FirmwareStatusEnumType is used by: firmwareStatusNotification:FirmwareStatusNotificationRequest

Value	Description
Downloaded	Intermediate state. New firmware has been downloaded by Charging Station.
DownloadFailed	Failure end state. Charging Station failed to download firmware.
Downloading	Intermediate state. Firmware is being downloaded.
DownloadScheduled	Intermediate state. Downloading of new firmware has been scheduled.
DownloadPaused	Intermediate state. Downloading has been paused.
Idle	Charging Station is not performing firmware update related tasks. Status Idle SHALL only be used as in a FirmwareStatusNotificationRequest that was triggered by TriggerMessageRequest.
InstallationFailed	Failure end state. Installation of new firmware has failed.
Installing	Intermediate state. Firmware is being installed.
Installed	Successful end state. New firmware has successfully been installed in Charging Station.
InstallRebooting	Intermediate state. Charging Station is about to reboot to activate new firmware. If sent before installing the firmware, it indicates the Charging Station is about to reboot to start installing new firmware. If sent after installing the new firmware, it indicates the Charging Station has finished installing, but requires a reboot to activate the new firmware, which will be done automatically when idle. This status MAY be omitted if a reboot is an integral part of the installation and cannot be reported separately.
InstallScheduled	Intermediate state. Installation of the downloaded firmware is scheduled to take place on installDateTime given in UpdateFirmware request.
InstallVerificationFailed	Failure end state. Verification of the new firmware (e.g. using a checksum or some other means) has failed and installation will not proceed. (Final failure state)

Value	Description
InvalidSignature	Failure end state. The firmware signature is not valid.
SignatureVerified	Intermediate state. Provide signature successfully verified.

3.34. GenericDeviceModelStatusEnumType

Enumeration

GenericDeviceModelStatusEnumType is used by: getBaseReport:GetBaseReportResponse, getMonitoringReport:GetMonitoringReportResponse, getReport:GetReportResponse, setMonitoringBase:SetMonitoringBaseResponse

Value	Description
Accepted	Request has been accepted and will be executed.
Rejected	Request has not been accepted and will not be executed.
NotSupported	The content of the request message is not supported.
EmptyResultSet	If the combination of received criteria result in an empty result set.

3.35. GenericStatusEnumType

Enumeration

Generic message response status

GenericStatusEnumType is used by: getCompositeSchedule:GetCompositeScheduleResponse, notifyEVChargingSchedule:NotifyEVChargingScheduleResponse, signCertificate:SignCertificateResponse, setMonitoringLevel:SetMonitoringLevelResponse, publishFirmware:PublishFirmwareResponse

Value	Description
Accepted	Request has been accepted and will be executed.
Rejected	Request has not been accepted and will not be executed.

3.36. GetCertificateIdUseEnumType

Enumeration

GetCertificateIdUseEnumType is used by: Common:CertificateHashDataChainType, getInstalledCertificateIds:GetInstalledCertificateIdsRequest

Value	Description
V2GRootCertificate	Use for certificate of the V2G Root.
MORootCertificate	Use for certificate from an eMobility Service provider. To support PnC charging with contracts from service providers that not derived their certificates from the V2G root.

Value	Description
CSMSRootCertificate	Root certificate for verification of the CSMS certificate.
V2GCertificateChain	ISO 15118 V2G certificate chain (excluding the V2GRootCertificate).
ManufacturerRootCertificate	Root certificate for verification of the Manufacturer certificate.

3.37. GetCertificateStatusEnumType

Enumeration

GetCertificateStatusEnumType is used by: getCertificateStatus:GetCertificateStatusResponse

Value	Description
Accepted	Successfully retrieved the OCSP certificate status.
Failed	Failed to retrieve the OCSP certificate status.

3.38. GetChargingProfileStatusEnumType

Enumeration

GetChargingProfileStatusEnumType is used by: getChargingProfiles:GetChargingProfilesResponse

Value	Description
Accepted	Normal successful completion (no errors).
NoProfiles	No ChargingProfiles found that match the information in the GetChargingProfilesRequest.

3.39. GetDisplayMessagesStatusEnumType

Enumeration

GetDisplayMessagesStatusEnumType is used by: getDisplayMessages:GetDisplayMessagesResponse

Value	Description
Accepted	Request accepted, there are Display Messages found that match all the requested criteria. The Charging Station will send NotifyDisplayMessagesRequest messages to report the requested Display Messages.
Unknown	No messages found that match the given criteria.

3.40. GetInstalledCertificateStatusEnumType

Enumeration

GetInstalledCertificateStatusEnumType is used by: getInstalledCertificateIds:GetInstalledCertificateIdsResponse

Value	Description
Accepted	Normal successful completion (no errors).
NotFound	Requested resource not found.

3.41. GetVariableStatusEnumType

Enumeration

GetVariableStatusEnumType is used by: getVariables:GetVariablesResponse.GetVariableResultType

Value	Description
Accepted	Variable successfully set.
Rejected	Request is rejected.
UnknownComponent	Component is not known.
UnknownVariable	Variable is not known.
NotSupportedAttributeType	The AttributeType is not supported.

3.42. HashAlgorithmEnumType

Enumeration

HashAlgorithmEnumType is used by: Common:CertificateHashDataType, Common:OCSPRequestDataType

Value	Description
SHA256	SHA-256 hash algorithm.
SHA384	SHA-384 hash algorithm.
SHA512	SHA-512 hash algorithm.

3.43. IdTokenEnumType

Enumeration

Allowable values of the **IdTokenType** field.

IdTokenEnumType is used by: Common:IdTokenType

Value	Description
Central	A centrally, in the CSMS (or other server) generated id (for example used for a remotely started transaction that is activated by SMS). No format defined, might be a UUID.

Value	Description
eMAID	Electro-mobility account id as defined in ISO 15118
ISO14443	ISO 14443 UID of RFID card. It is represented as an array of 4 or 7 bytes in hexadecimal representation.
ISO15693	ISO 15693 UID of RFID card. It is represented as an array of 8 bytes in hexadecimal representation.
KeyCode	User use a private key-code to authorize a charging transaction. For example: Pin-code.
Local	A locally generated id (e.g. internal id created by the Charging Station). No format defined, might be a UUID
MacAddress	The MacAddress of the EVCC (Electric Vehicle Communication Controller) that is connected to the EVSE. This is used as a token type when the MAC address is used for authorization ("Autocharge").
NoAuthorization	Transaction is started and no authorization possible. Charging Station only has a start button or mechanical key etc. IdToken field SHALL be left empty.

3.44. InstallCertificateStatusEnumType

Enumeration

InstallCertificateStatusEnumType is used by: installCertificate:InstallCertificateResponse

Value	Description
Accepted	The installation of the certificate succeeded.
Rejected	The certificate is invalid and/or incorrect OR the CSO tries to install more certificates than allowed.
Failed	The certificate is valid and correct, but there is another reason the installation did not succeed.

3.45. InstallCertificateUseEnumType

Enumeration

InstallCertificateUseEnumType is used by: installCertificate:InstallCertificateRequest

Value	Description
V2GRootCertificate	Use for certificate of the V2G Root, a V2G Charging Station Certificate MUST be derived from one of the installed V2GRootCertificate certificates.
MORootCertificate	Use for certificate from an eMobility Service provider. To support PnC charging with contracts from service providers that not derived their certificates from the V2G root.
CSMSRootCertificate	Root certificate, used by the CA to sign the CSMS and Charging Station certificate.
ManufacturerRootCertificate	Root certificate for verification of the Manufacturer certificate.

3.46. Iso15118EVCertificateStatusEnumType

Enumeration

Iso15118EVCertificateStatusEnumType is used by: get15118EVCertificate:Get15118EVCertificateResponse

Value	Description
Accepted	exiResponse included. This is no indication whether the update was successful, just that the message was processed properly.
Failed	Processing of the message was not successful, no exiResponse included.

3.47. LocationEnumType

Enumeration

Allowable values of the optional "location" field of a value element.

LocationEnumType is used by: Common:SampledValueType

Value	Description
Body	Measurement inside body of Charging Station (e.g. Temperature).
Cable	Measurement taken from cable between EV and Charging Station.
EV	Measurement taken by EV.
Inlet	For the Charging Station (*evseId* = 0): measurement at network ("grid") inlet connection. For measurements with *evseId* > 0, these are measurements taken at the EVSE inlet (This can be useful for a DC charger).
Outlet	Measurement at a Connector. Default value.

3.48. LogEnumType

Enumeration

LogEnumType is used by: getLog:GetLogRequest

Value	Description
DiagnosticsLog	This contains the field definition of a diagnostics log file
SecurityLog	Sent by the CSMS to the Charging Station to request that the Charging Station uploads the security log.

3.49. LogStatusEnumType

Enumeration

Generic message response status

LogStatusEnumType is used by: getLog:GetLogResponse

Value	Description
Accepted	Accepted this log upload. This does not mean the log file is uploaded is successfully, the Charging Station will now start the log file upload.
Rejected	Log update request rejected.
AcceptedCanceled	Accepted this log upload, but in doing this has canceled an ongoing log file upload.

3.50. MeasurandEnumType

Enumeration

Allowable values of the optional "measurand" field of a Value element, as used in MeterValuesRequest and TransactionEventRequest with eventTypes *Started*, *Ended* and *Updated*. Default value of "measurand" is always "Energy.Active.Import.Register".

Note 1 : Two measurands (Current.Offered and Power.Offered) are available that are strictly speaking no measured values. They indicate the maximum amount of current/power that is being offered to the EV and are intended for use in smart charging applications.

Note 2: Import is energy flow from the Grid to the Charging Station, EV or other load. Export is energy flow from the EV to the Charging Station and/or from the Charging Station to the Grid. Except in the case of a meter replacement, all "Register" values relating to a single charging transaction, or a non-transactional consumer (e.g. Charging Station internal power supply, overall supply) MUST be monotonically increasing in time.

Note 3: The actual quantity of energy corresponding to a reported ".Register" value is computed as the register value in question minus the register value recorded/reported at the start of the transaction or other relevant starting reference point in time. For improved auditability, ".Register" values SHOULD be reported exactly as they are directly read from a non-volatile register in the electrical metering hardware, and SHOULD NOT be re-based to zero at the start of transactions. This allows any "missing energy" between sequential transactions, due to hardware fault, meter replacement, mis-wiring, fraud, etc. to be identified, by allowing the CSMS to confirm that the starting register value of any transaction is identical to the finishing register value of the preceding transaction on the same connector.

MeasurandEnumType is used by: Common:SampledValueType

Value	Description
Current.Export	Instantaneous current flow from EV
Current.Import	Instantaneous current flow to EV
Current.Offered	Maximum current offered to EV
Energy.Active.Export.Register	Numerical value read from the "active electrical energy" (Wh or kWh) register of the (most authoritative) electrical meter measuring energy exported (to the grid).
Energy.Active.Import.Register	Numerical value read from the "active electrical energy" (Wh or kWh) register of the (most authoritative) electrical meter measuring energy imported (from the grid supply).
Energy.Reactive.Export.Register	Numerical value read from the "reactive electrical energy" (varh or kvarh) register of the (most authoritative) electrical meter measuring energy exported (to the grid).
Energy.Reactive.Import.Register	Numerical value read from the "reactive electrical energy" (varh or kvarh) register of the (most authoritative) electrical meter measuring energy imported (from the grid supply).

Value	Description
Energy.Active.Export.Interval	Absolute amount of "active electrical energy" (Wh or kWh) exported (to the grid) during an associated time "interval", specified by a Metervalues ReadingContext, and applicable interval duration configuration values (in seconds) for ClockAlignedDataInterval and TxnMeterValueSampleInterval.
Energy.Active.Import.Interval	Absolute amount of "active electrical energy" (Wh or kWh) imported (from the grid supply) during an associated time "interval", specified by a Metervalues ReadingContext, and applicable interval duration configuration values (in seconds) for ClockAlignedDataInterval and TxnMeterValueSampleInterval.
Energy.Active.Net	Numerical value read from the “net active electrical energy" (Wh or kWh) register.
Energy.Reactive.Export.Interval	Absolute amount of "reactive electrical energy" (varh or kvarh) exported (to the grid) during an associated time "interval", specified by a Metervalues ReadingContext, and applicable interval duration configuration values (in seconds) for ClockAlignedDataInterval and TxnMeterValueSampleInterval.
Energy.Reactive.Import.Interval	Absolute amount of "reactive electrical energy" (varh or kvarh) imported (from the grid supply) during an associated time "interval", specified by a Metervalues ReadingContext, and applicable interval duration configuration values (in seconds) for ClockAlignedDataInterval and TxnMeterValueSampleInterval.
Energy.Reactive.Net	Numerical value read from the “net reactive electrical energy" (varh or kvarh) register.
Energy.Apparent.Net	Numerical value read from the "apparent electrical energy" (VAh or kVAh) register.
Energy.Apparent.Import	Numerical value read from the "apparent electrical import energy" (VAh or kVAh) register.
Energy.Apparent.Export	Numerical value read from the "apparent electrical export energy" (VAh or kVAh) register.
Frequency	Instantaneous reading of powerline frequency
Power.Active.Export	Instantaneous active power exported by EV. (W or kW)
Power.Active.Import	Instantaneous active power imported by EV. (W or kW)
Power.Factor	Instantaneous power factor of total energy flow
Power.Offered	Maximum power offered to EV
Power.Reactive.Export	Instantaneous reactive power exported by EV. (var or kvar)
Power.Reactive.Import	Instantaneous reactive power imported by EV. (var or kvar)
SoC	State of charge of charging vehicle in percentage
Voltage	Instantaneous DC or AC RMS supply voltage. For *location* = `Inlet` and *evseId* = 0: voltage at charging station grid connection. For *location* = `Outlet` and *evseId* > 0: voltage at EVSE outlet towards the EV.

3.51. MessageFormatEnumType

Enumeration

Format of a message to be displayed on the display of the Charging Station.

MessageFormatEnumType is used by: Common:MessageContentType

Value	Description
ASCII	Message content is ASCII formatted, only printable ASCII allowed.
HTML	Message content is HTML formatted.
URI	Message content is URI that Charging Station should download and use to display. for example a HTML page to be shown in a web-browser.
UTF8	Message content is UTF-8 formatted.

3.52. MessagePriorityEnumType

Enumeration

Priority with which a message should be displayed on a Charging Station.

MessagePriorityEnumType is used by: Common:MessageInfoType, getDisplayMessages:GetDisplayMessagesRequest

Value	Description
AlwaysFront	Show this message always in front. Highest priority, don't cycle with other messages. When a newer message with this MessagePriority is received, this message is replaced. No Charging Station own message may override this message.
InFront	Show this message in front of the normal cycle of messages. When more messages with this priority are to be shown, they SHALL be cycled.
NormalCycle	Show this message in the cycle of messages.

3.53. MessageStateEnumType

Enumeration

State of the Charging Station during which a message SHALL be displayed.

MessageStateEnumType is used by: Common:MessageInfoType, getDisplayMessages:GetDisplayMessagesRequest

Value	Description
Charging	Message only to be shown while the Charging Station is charging.
Faulted	Message only to be shown while the Charging Station is in faulted state.
Idle	Message only to be shown while the Charging Station is idle (not charging).
Unavailable	Message only to be shown while the Charging Station is in unavailable state.

3.54. MessageTriggerEnumType

Enumeration

Type of request to be triggered by trigger messages.

MessageTriggerEnumType is used by: triggerMessage:TriggerMessageRequest

Value	Description
BootNotification	To trigger BootNotification.
LogStatusNotification	To trigger LogStatusNotification.
FirmwareStatusNotification	To trigger FirmwareStatusNotification.
Heartbeat	To trigger Heartbeat.
MeterValues	To trigger MeterValues.
SignChargingStationCertificate	To trigger a SignCertificate with certificateType: ChargingStationCertificate.
SignV2GCertificate	To trigger a SignCertificate with certificateType: V2GCertificate
StatusNotification	To trigger StatusNotification.
TransactionEvent	To trigger TransactionEvent.
SignCombinedCertificate	To trigger a SignCertificate with certificateType: ChargingStationCertificate AND V2GCertificate
PublishFirmwareStatusNotification	To trigger PublishFirmwareStatusNotification.

3.55. MonitorEnumType

Enumeration

MonitorEnumType is used by: Common:VariableMonitoringType, setVariableMonitoring:SetVariableMonitoringRequest.SetMonitoringDataType, setVariableMonitoring:SetVariableMonitoringResponse.SetMonitoringResultType

Value	Description
UpperThreshold	Triggers an event notice when the actual value of the Variable rises above monitorValue
LowerThreshold	Triggers an event notice when the actual value of the Variable drops below monitorValue.
Delta	Triggers an event notice when the actual value has changed more than plus or minus *monitorValue* since the time that this monitor was set or since the last time this event notice was sent, whichever was last. For variables that are not numeric, like boolean, string or enumerations, a monitor of type Delta will trigger an event notice whenever the variable changes, regardless of the value of *monitorValue*.
Periodic	Triggers an event notice every *monitorValue* seconds interval, starting from the time that this monitor was set.
PeriodicClockAligned	Triggers an event notice every monitorValue seconds interval, starting from the nearest clock-aligned interval after this monitor was set. For example, a *monitorValue* of 900 will trigger event notices at 0, 15, 30 and 45 minutes after the hour, every hour.

3.56. MonitoringBaseEnumType

Enumeration

MonitoringBaseEnumType is used by: setMonitoringBase:SetMonitoringBaseRequest

Value	Description
All	Activate all pre-configured monitors.
FactoryDefault	Activate the default monitoring settings as recommended by the manufacturer. This is a subset of all preconfigured monitors.
HardWiredOnly	Clears all custom monitors and disables all pre-configured monitors.

3.57. MonitoringCriterionEnumType

Enumeration

MonitoringCriterionEnumType is used by: getMonitoringReport:GetMonitoringReportRequest

Value	Description
ThresholdMonitoring	Report variables and components with a monitor of type UpperThreshold or LowerThreshold.
DeltaMonitoring	Report variables and components with a monitor of type Delta.
PeriodicMonitoring	Report variables and components with a monitor of type Periodic or PeriodicClockAligned.

3.58. MutabilityEnumType

Enumeration

MutabilityEnumType is used by: Common:VariableAttributeType

Value	Description
ReadOnly	This variable is read-only.
WriteOnly	This variable is write-only.
ReadWrite	This variable is read-write.

3.59. NotifyEVChargingNeedsStatusEnumType

Enumeration

NotifyEVChargingNeedsStatusEnumType is used by: notifyEVChargingNeeds:NotifyEVChargingNeedsResponse

Value	Description
Accepted	A schedule will be provided momentarily.

Value	Description
Rejected	Service not available.
Processing	The CSMS is gathering information to provide a schedule.

3.60. OCPPInterfaceEnumType

Enumeration

Enumeration of network interfaces.

OCPPInterfaceEnumType is used by: setNetworkProfile:SetNetworkProfileRequest.NetworkConnectionProfileType

Value	Description
Wired0	Use wired connection 0
Wired1	Use wired connection 1
Wired2	Use wired connection 2
Wired3	Use wired connection 3
Wireless0	Use wireless connection 0
Wireless1	Use wireless connection 1
Wireless2	Use wireless connection 2
Wireless3	Use wireless connection 3

3.61. OCPPTransportEnumType

Enumeration

Enumeration of OCPP transport mechanisms. SOAP is currently not a valid value for OCPP 2.0.

OCPPTransportEnumType is used by: setNetworkProfile:SetNetworkProfileRequest.NetworkConnectionProfileType

Value	Description
JSON	Use JSON over WebSockets for transport of OCPP PDU's
SOAP	Use SOAP for transport of OCPP PDU's

3.62. OCPPVersionEnumType

Enumeration

Enumeration of OCPP versions.

OCPPVersionEnumType is used by: setNetworkProfile:SetNetworkProfileRequest.NetworkConnectionProfileType

Value	Description
OCPP12	OCPP version 1.2
OCPP15	OCPP version 1.5
OCPP16	OCPP version 1.6
OCPP20	OCPP version 2.0 The OCPP 2.0 release of OCPP has been deprecated, so this value OCPP20 must now be used for OCPP 2.0.1 implementations in the NetworkConnectionProfile. Note that OCPP 2.0.1 does have its own Websocket subprotocol name: ocpp2.0.1.

3.63. OperationalStatusEnumType

Enumeration

Requested availability change.

OperationalStatusEnumType is used by: changeAvailability:ChangeAvailabilityRequest

Value	Description
Inoperative	Charging Station is not available for charging.
Operative	Charging Station is available for charging.

3.64. PhaseEnumType

Enumeration

Phase specifies how a measured value is to be interpreted. Please note that not all values of Phase are applicable to all Measurands.

PhaseEnumType is used by: Common:SampledValueType

Value	Description
L1	Measured on L1
L2	Measured on L2
L3	Measured on L3
N	Measured on Neutral
L1-N	Measured on L1 with respect to Neutral conductor
L2-N	Measured on L2 with respect to Neutral conductor
L3-N	Measured on L3 with respect to Neutral conductor
L1-L2	Measured between L1 and L2

Value	Description
L2-L3	Measured between L2 and L3
L3-L1	Measured between L3 and L1

3.65. PublishFirmwareStatusEnumType

Enumeration

Status for when publishing a Firmware.

PublishFirmwareStatusEnumType is used by: publishFirmwareStatusNotification:PublishFirmwareStatusNotificationRequest

Value	Description
Idle	
DownloadScheduled	Intermediate state. Downloading of new firmware has been scheduled.
Downloading	Intermediate state. Firmware is being downloaded.
Downloaded	Intermediate state. New firmware has been downloaded by Charging Station.
Published	The firmware has been successfully published.
DownloadFailed	Failure end state. Charging Station failed to download firmware.
DownloadPaused	Intermediate state. Downloading has been paused.
ChecksumVerified	Intermediate state. The Firmware checksum is successfully verified.
PublishFailed	Publishing the new firmware has failed.
InvalidChecksum	Failure end state. The firmware checksum is not matching.

3.66. ReadingContextEnumType

Enumeration

Values of the context field.

ReadingContextEnumType is used by: Common:SampledValueType

Value	Description
Interruption.Begin	Value taken at start of interruption.
Interruption.End	Value taken when resuming after interruption.
Other	Value for any other situations.
Sample.Clock	Value taken at clock aligned interval.

Value	Description
Sample.Periodic	Value taken as periodic sample relative to start time of transaction.
Transaction.Begin	Value taken at start of transaction.
Transaction.End	Value taken at end of transaction.
Trigger	Value taken in response to TriggerMessageRequest.

3.67. ReasonEnumType

Enumeration

Reason for stopping a transaction.

ReasonEnumType is used by: transactionEvent:TransactionEventRequest.TransactionType

Value	Description
DeAuthorized	The transaction was stopped because of the authorization status in the response to a transactionEventRequest.
EmergencyStop	Emergency stop button was used.
EnergyLimitReached	EV charging session reached a locally enforced maximum energy transfer limit
EVDisconnected	Disconnecting of cable, vehicle moved away from inductive charge unit.
GroundFault	A GroundFault has occurred
ImmediateReset	A Reset(Immediate) command was received.
Local	Stopped locally on request of the EV Driver at the Charging Station. This is a regular termination of a transaction. Examples: presenting an IdToken tag, pressing a button to stop.
LocalOutOfCredit	A local credit limit enforced through the Charging Station has been exceeded.
MasterPass	The transaction was stopped using a token with a MasterPassGroupId.
Other	Any other reason.
OvercurrentFault	A larger than intended electric current has occurred
PowerLoss	Complete loss of power.
PowerQuality	Quality of power too low, e.g. voltage too low/high, phase imbalance, etc.
Reboot	A locally initiated reset/reboot occurred. (for instance watchdog kicked in)
Remote	Stopped remotely on request of the CSMS. This is a regular termination of a transaction. Examples: termination using a smartphone app, exceeding a (non local) prepaid credit.
SOCLimitReached	Electric vehicle has reported reaching a locally enforced maximum battery State of Charge (SOC)

Value	Description
StoppedByEV	The transaction was stopped by the EV
TimeLimitReached	EV charging session reached a locally enforced time limit
Timeout	EV not connected within timeout

3.68. RecurrencyKindEnumType

Enumeration

RecurrencyKindEnumType is used by: Common:ChargingProfileType

Value	Description
Daily	The schedule restarts every 24 hours, at the same time as in the startSchedule.
Weekly	The schedule restarts every 7 days, at the same time and day-of-the-week as in the startSchedule.

3.69. RegistrationStatusEnumType

Enumeration

Result of registration in response to BootNotificationRequest.

RegistrationStatusEnumType is used by: bootNotification:BootNotificationResponse

Value	Description
Accepted	Charging Station is accepted by the CSMS.
Pending	CSMS is not yet ready to accept the Charging Station. CSMS may send messages to retrieve information or prepare the Charging Station.
Rejected	Charging Station is not accepted by CSMS. This may happen when the Charging Station id is not known by CSMS.

3.70. ReportBaseEnumType

Enumeration

ReportBaseEnumType is used by: getBaseReport:GetBaseReportRequest

Value	Description
ConfigurationInventory	Required. A (configuration) report that lists all Components/Variables that can be set by the operator.
FullInventory	Required. A (full) report that lists everything except monitoring settings.

Value	Description
SummaryInventory	Optional. A (summary) report that lists Components/Variables relating to the Charging Station's current charging availability, and to any existing problem conditions. For the Charging Station Component: - AvailabilityState. For each EVSE Component: - AvailabilityState. For each Connector Component: - AvailabilityState (if known and different from EVSE). For all Components in an abnormal State: - Active (Problem, Tripped, Overload, Fallback) variables. - Any other diagnostically relevant Variables of the Components. - Include TechCode and TechInfo where available. All monitored Component.Variables in Critical or Alert state shall also be included. - Charging Stations that do not have Monitoring implemented are NOT REQUIRED to include Connector Availability, monitoring alerts, and MAY limit problem reporting detail to just the active Problem boolean Variable.

3.71. RequestStartStopStatusEnumType

Enumeration

The result of a RequestStartTransactionRequest or RequestStopTransactionRequest.

RequestStartStopStatusEnumType is used by: requestStartTransaction:RequestStartTransactionResponse, requestStopTransaction:RequestStopTransactionResponse

Value	Description
Accepted	Command will be executed.
Rejected	Command will not be executed.

3.72. ReservationUpdateStatusEnumType

Enumeration

ReservationUpdateStatusEnumType is used by: reservationStatusUpdate:ReservationStatusUpdateRequest

Value	Description
Expired	The reservation is expired.
Removed	The reservation is removed.

3.73. ReserveNowStatusEnumType

Enumeration

Status in ReserveNowResponse.

ReserveNowStatusEnumType is used by: reserveNow:ReserveNowResponse

Value	Description
Accepted	Reservation has been made.
Faulted	Reservation has not been made, because evse, connectors or specified connector are in a faulted state.
Occupied	Reservation has not been made. The evse or the specified connector is occupied.
Rejected	Reservation has not been made. Charging Station is not configured to accept reservations.
Unavailable	Reservation has not been made, because evse, connectors or specified connector are in an unavailable state.

3.74. ResetEnumType

Enumeration

Type of reset requested.

ResetEnumType is used by: reset:ResetRequest

Value	Description
Immediate	Immediate reset of the Charging Station.
OnIdle	Delay reset until no more transactions are active.

3.75. ResetStatusEnumType

Enumeration

Result of ResetRequest.

ResetStatusEnumType is used by: reset:ResetResponse

Value	Description
Accepted	Command will be executed.
Rejected	Command will not be executed.
Scheduled	Reset command is scheduled, Charging Station is busy with a process that cannot be interrupted at the moment. Reset will be executed when process is finished.

3.76. SendLocalListStatusEnumType

Enumeration

Type of update for SendLocalListRequest.

SendLocalListStatusEnumType is used by: sendLocalList:SendLocalListResponse

Value	Description
Accepted	Local Authorization List successfully updated.
Failed	Failed to update the Local Authorization List.
VersionMismatch	Version number in the request for a differential update is less or equal then version number of current list.

3.77. SetMonitoringStatusEnumType

Enumeration

SetMonitoringStatusEnumType is used by: setVariableMonitoring:SetVariableMonitoringResponse.SetMonitoringResultType

Value	Description
Accepted	Monitor successfully set.
UnknownComponent	Component is not known.
UnknownVariable	Variable is not known.
UnsupportedMonitorType	Requested monitor type is not supported.
Rejected	Request is rejected.
Duplicate	A monitor already exists for the given type/severity combination.

3.78. SetNetworkProfileStatusEnumType

Enumeration

Possible values of SetNetworkProfileStatus as used in SetNetworkProfileResponse.

SetNetworkProfileStatusEnumType is used by: setNetworkProfile:SetNetworkProfileResponse

Value	Description
Accepted	Setting new data successful
Rejected	Setting new data rejected
Failed	Setting new data failed

3.79. SetVariableStatusEnumType

Enumeration

SetVariableStatusEnumType is used by: setVariables:SetVariablesResponse.SetVariableResultType

Value	Description
Accepted	Variable successfully set.
Rejected	Request is rejected.
UnknownComponent	Component is not known.
UnknownVariable	Variable is not known.
NotSupportedAttributeType	The AttributeType is not supported.
RebootRequired	A reboot is required.

3.80. TransactionEventEnumType

Enumeration

TransactionEventEnumType is used by: transactionEvent:TransactionEventRequest

Value	Description
Ended	Last event of a transaction
Started	First event of a transaction.
Updated	Transaction event in between 'Started' and 'Ended'.

3.81. TriggerMessageStatusEnumType

Enumeration

Status in TriggerMessageResponse.

TriggerMessageStatusEnumType is used by: triggerMessage:TriggerMessageResponse

Value	Description
Accepted	Requested message will be sent.
Rejected	Requested message will not be sent.
NotImplemented	Requested message cannot be sent because it is either not implemented or unknown.

3.82. TriggerReasonEnumType

Enumeration

Reason that triggered a transactionEventRequest.

TriggerReasonEnumType is used by: transactionEvent:TransactionEventRequest

Value	Description
Authorized	Charging is authorized, by any means. Might be an RFID, or other authorization means.
CablePluggedIn	Cable is plugged in and EVDetected.
ChargingRateChanged	Rate of charging changed by more than LimitChangeSignificance.
ChargingStateChanged	Charging State changed.
Deauthorized	The transaction was stopped because of the authorization status in the response to a transactionEventRequest.
EnergyLimitReached	Maximum energy of charging reached. For example: in a pre-paid charging solution
EVCommunicationLost	Communication with EV lost, for example: cable disconnected.
EVConnectTimeout	EV not connected before the connection is timed out.
MeterValueClock	Needed to send a clock aligned meter value
MeterValuePeriodic	Needed to send a periodic meter value
TimeLimitReached	Maximum time of charging reached. For example: in a pre-paid charging solution
Trigger	Requested by the CSMS via a TriggerMessageRequest.
UnlockCommand	CSMS sent an Unlock Connector command.
StopAuthorized	An EV Driver has been authorized to stop charging. For example: By swiping an RFID card.
EVDeparted	EV departed. For example: When a departing EV triggers a parking bay detector.
EVDetected	EV detected. For example: When an arriving EV triggers a parking bay detector.
RemoteStop	A RequestStopTransactionRequest has been sent.
RemoteStart	A RequestStartTransactionRequest has been sent.
AbnormalCondition	An Abnormal Error or Fault Condition has occurred.
SignedDataReceived	Signed data is received from the energy meter.
ResetCommand	CSMS sent a Reset Charging Station command.

3.83. UnlockStatusEnumType

Enumeration

Status in response to UnlockConnectorRequest.

UnlockStatusEnumType is used by: unlockConnector:UnlockConnectorResponse

Value	Description
Unlocked	Connector has successfully been unlocked.
UnlockFailed	Failed to unlock the connector.
OngoingAuthorizedTransaction	The connector is not unlocked, because there is still an authorized transaction ongoing.
UnknownConnector	The specified connector is not known by the Charging Station.

3.84. UnpublishFirmwareStatusEnumType

Enumeration

Status for when publishing a Firmware.

UnpublishFirmwareStatusEnumType is used by: unpublishFirmware:UnpublishFirmwareResponse

Value	Description
DownloadOngoing	Intermediate state. Firmware is being downloaded.
NoFirmware	There is no published file.
Unpublished	Successful end state. Firmware file no longer being published.

3.85. UpdateEnumType

Enumeration

UpdateEnumType is used by: sendLocalList:SendLocalListRequest

Value	Description
Differential	Indicates that the current Local Authorization List must be updated with the values in this message.
Full	Indicates that the current Local Authorization List must be replaced by the values in this message.

3.86. UpdateFirmwareStatusEnumType

Enumeration

Generic message response status

UpdateFirmwareStatusEnumType is used by: updateFirmware:UpdateFirmwareResponse

Value	Description
Accepted	Accepted this firmware update request. This does not mean the firmware update is successful, the Charging
Station	will now start the firmware update process.

Value	Description
Rejected	Firmware update request rejected.
AcceptedCanceled	Accepted this firmware update request, but in doing this has canceled an ongoing firmware update.
InvalidCertificate	The certificate is invalid.
RevokedCertificate	Failure end state. The Firmware Signing certificate has been revoked.

3.87. UploadLogStatusEnumType

Enumeration

UploadLogStatusEnumType is used by: logStatusNotification:LogStatusNotificationRequest

Value	Description
BadMessage	A badly formatted packet or other protocol incompatibility was detected.
Idle	The Charging Station is not uploading a log file. Idle SHALL only be used when the message was triggered by a TriggerMessageRequest.
NotSupportedOperation	The server does not support the operation
PermissionDenied	Insufficient permissions to perform the operation.
Uploaded	File has been uploaded successfully.
UploadFailure	Failed to upload the requested file.
Uploading	File is being uploaded.
AcceptedCanceled	On-going log upload is canceled and new request to upload log has been accepted.

3.88. VPNEnumType

Enumeration

Enumeration of VPN Types.

VPNEnumType is used by: setNetworkProfile:SetNetworkProfileRequest.VPNType

Value	Description
IKEv2	IKEv2 VPN
IPSec	IPSec VPN
L2TP	L2TP VPN
PPTP	PPTP VPN

Referenced Components and Variables

1. Controller Components

This section gives an overview of the 'Controller' components, which are introduced in OCPP 2.0. A controller component can be recognized by the 'Ctrlr' suffix and is responsible for the configuration of a certain functionality. Most of the 'Referenced' components that are described in this document, are 'Controller' components.

The table below contains a summary of all Controller components, for more details, please refer to Part 2 - Appendices.

Controller	Component Description
AlignedDataCtrlr	Responsible for configuration relating to the reporting of clock-aligned meter data.
AuthCacheCtrlr	Responsible for configuration relating to the use of a local cache for authorization for Charging Station use.
AuthCtrlr	Responsible for configuration relating to the use of authorization for Charging Station use.
CHAdeMOCtrlr	Responsible for configuration relating to the CHAdeMO controller
ClockCtrlr	Provides a means to configure management of time tracking by Charging Station.
CustomizationCtrlr	Responsible for configuration relating to custom vendor-specific implementations, using the DataTransfer message and CustomData extensions.
DeviceDataCtrlr	Responsible for configuration relating to the exchange and storage of Charging Station device model data.
DisplayMessageCtrlr	Responsible for configuration relating to the display of messages to Charging Station users.
ISO15118Ctrlr	Responsible for configuration relating to the ISO 15118 controller
LocalAuthListCtrlr	Responsible for configuration relating to the use of local authorization lists for Charging Station use.
MonitoringCtrlr	Responsible for configuration relating to the exchange of monitoring event data.
OCPPCommCtrlr	Responsible for configuration relating to information exchange between Charging Station and CSMS.
ReservationCtrlr	Responsible for configuration relating to reservations.
SampledDataCtrlr	Responsible for configuration relating to the reporting of sampled meter data.
SecurityCtrlr	Responsible for configuration relating to security of communications between Charging Station and CSMS.
SmartChargingCtrlr	Responsible for configuration relating to Smart Charging.
TariffCostCtrlr	Responsible for configuration relating to tariff and cost display.
TxCtrlr	Responsible for configuration relating to transaction characteristics and behaviour.

Every Controller component has an 'Enabled' variable. This variable can be used to enable/disable a certain functionality. Any data in the charging station is not part of the controller component, so when disabling a functionality, any relating data stored in the Charging Station will not be changed or removed.
For example: if ReservationCtrlr is disabled when there is an active reservation, the EVSE will become available, but the reservation entries will still be there - they are just not used. If afterwards ReservationCtrlr is enabled again, the reservation entries will become active again as long as they have not expired and no transaction is in progress. If a transaction has started in the mean time, that transaction remains active. The reservation is then considered expired.

2. Referenced Components and Variables

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

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

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

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

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

Requirements for all the Configuration Variables in this document:

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

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

2.1. General

NOTE	WebSocket-related variables are described in "OCPP-2.0.1 Part 4 JSON over WebSockets".

2.1.1. ActiveNetworkProfile

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

2.1.2. AllowNewSessionsPendingFirmwareUpdate

Required	no	
Component	**componentName**	ChargingStation

Variable	**variableName**	AllowNewSessionsPendingFirmwareUpdate	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	Indicates whether new sessions can be started on EVSEs, while Charging Station is waiting for all EVSEs to become Available in order to start a pending firmware update. When a firmware update is waiting to be installed and this variable exists and has the value *true*, then, the Charging Station will not set free EVSEs to Unavailable, pending the update. This means that it may take longer until there is a point in time when all EVSEs of the Charging Station are free and it can perform the firmware update.		

2.1.3. DefaultMessageTimeout

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

2.1.4. FileTransferProtocols

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

2.1.5. HeartbeatInterval

Required	no

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

2.1.6. NetworkConfigurationPriority

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

2.1.7. NetworkProfileConnectionAttempts

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

2.1.8. OfflineThreshold

Required	yes

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

2.1.9. QueueAllMessages

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

2.1.10. MessageAttemptsTransactionEvent

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

2.1.11. MessageAttemptIntervalTransactionEvent

Required	yes		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	MessageAttemptInterval	
	variableInstance	TransactionEvent	
	variableAttributes	**attributeType**	Actual
		mutability	ReadWrite
	variableCharacteristics	**unit**	s
		dataType	integer
Description	How long the Charging Station should wait before resubmitting a TransactionEventRequest message that the CSMS failed to process.		

2.1.12. UnlockOnEVSideDisconnect

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

2.1.13. WebSocketPingInterval

This configuration variable is described in "OCPP-2.0.1 Part 4 JSON over WebSockets".

2.1.14. ResetRetries

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

2.1.15. MessageFieldLength

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

2.1.16. ItemsPerMessageGetReport

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

2.1.17. ItemsPerMessageGetVariables

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

2.1.18. BytesPerMessageGetReport

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

2.1.19. BytesPerMessageGetVariables

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

2.1.20. ConfigurationValueSize

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

2.1.21. ReportingValueSize

Required	no		
Component	**componentName**	DeviceDataCtrlr	
Variable	**variableName**	ReportingValueSize	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	2500
Description	This Configuration Variable can be used to limit the following fields: GetVariableResult.attributeValue, VariableAttribute.value and EventData.actualValue. The max size of these values will always remain equal.		

2.1.22. ItemsPerMessageSetVariables

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

2.1.23. BytesPerMessageSetVariables

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

2.1.24. DateTime

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

2.1.25. NtpSource

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

2.1.26. NtpServerUri

Required	no		
Component	**componentName**	ClockCtrlr	
Variable	**variableName**	NtpServerUri	
	variableInstance	Single digit, multiple servers allowed, primary NtpServer has instance '1', the secondary has instance '2'. etc	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	When an NTP client is implemented, this variable can be used to configure the client: This contains the address of the NTP server. Multiple NTP servers can be configured. These can be back-up NTP servers. If the NTP client supports it, it can also connect to multiple NTP servers simultaneous to get a more reliable time source.		

2.1.27. TimeOffset

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

2.1.28. NextTimeOffsetTransitionDateTime

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

2.1.29. TimeOffsetNextTransition

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

2.1.30. TimeSource

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

2.1.31. TimeZone

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

2.1.32. TimeAdjustmentReportingThreshold

Required	no	
Component	**componentName**	ClockCtrlr

Variable	**variableName**	TimeAdjustmentReportingThreshold	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	s
		dataType	integer
Description	When the clock time is adjusted forwards or backwards for more then TimeAdjustmentReportingThreshold number of seconds, a SecurityEventNotification("SettingSystemTime") is sent by the charging station. A reasonable value is 20 seconds.		

2.1.33. CustomImplementationEnabled

Required	no		
Component	**componentName**	CustomizationCtrlr	
Variable	**variableName**	CustomImplementationEnabled	
	variableInstance	<VendorId>	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	This standard configuration variable can be used to enable/disable custom implementations that the Charging Station supports. It is recommended to first check if the custom behavior is able to be implemented using the device model, otherwise DataTransfer message(s) and/or CustomData fields can be used.		

2.2. Security related

2.2.1. BasicAuthPassword

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

Required	no		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	BasicAuthPassword	
	variableAttributes	**mutability**	WriteOnly
	variableCharacteristics	**dataType**	string
		maxLimit	40 (Max length of the BasicAuthPassword)

Description	The basic authentication password is used for HTTP Basic Authentication. The password SHALL be a randomly chosen passwordString with a sufficiently high entropy, consisting of minimum 16 and maximum 40 characters (alpha-numeric characters and the special characters allowed by passwordString). The password SHALL be sent as a UTF-8 encoded string (NOT encoded into octet string or base64). This configuration variable is write-only, so that it cannot be accidentally stored in plaintext by the CSMS when it reads out all configuration variables. This configuration variable is required unless only "security profile 3 - TLS with client side certificates" is implemented.

2.2.2. Identity

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

2.2.3. OrganizationName

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

2.2.4. CertificateEntries

Required	yes	
Component	**componentName**	SecurityCtrlr

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

2.2.5. SecurityProfile

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

2.2.6. AdditionalRootCertificateCheck

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

2.2.7. MaxCertificateChainSize

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

2.2.8. CertSigningWaitMinimum

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

2.2.9. CertSigningRepeatTimes

Required	no		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	CertSigningRepeatTimes	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	integer

Description	This variable can be used to configure the amount of times the Charging Station SHALL double the previous back-off time, starting with the number of seconds configured at CertSigningWaitMinimum, every time the back-off time expires without having received the CertificateSignedRequest containing the from the CSR generated signed certificate. When the maximum number of increments is reached, the Charging Station SHALL stop resending the SignCertificateRequest, until it is requested by the CSMS using a TriggerMessageRequest.

2.3. Authorization related

2.3.1. AuthEnabled

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

2.3.2. `AdditionalInfoItemsPerMessage`

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

2.3.3. `OfflineTxForUnknownIdEnabled`

Required	no		
Component	**componentName**	AuthCtrlr	
Variable	**variableName**	OfflineTxForUnknownIdEnabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean

Description	If this key exists, the Charging Station supports Unknown Offline Authorization. If this key reports a value of *true*, Unknown Offline Authorization is enabled.

2.3.4. AuthorizeRemoteStart

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

2.3.5. LocalAuthorizeOffline

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

2.3.6. LocalPreAuthorize

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

2.3.7. MasterPassGroupId

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

2.3.8. DisableRemoteAuthorization

Required	no		
Component	**componentName**	AuthCtrlr	
Variable	**variableName**	DisableRemoteAuthorization	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	When set to true this instructs the Charging Station to not issue any AuthorizationRequests, but only use Authorization Cache and Local Authorization List to determine validity of idTokens. Note: The difference between **AuthCtrlr.DisableRemoteAuthorization** and **AuthCacheCtrlr.DisablePostAuthorization** is that the latter only disables re-authorization of tokens that are as not-Accepted in the Authorization Cache or Local Authorization List, whereas **AuthCtrlr.DisableRemoteAuthorization** disables all authorization with CSMS.		

2.4. Authorization Cache related

2.4.1. AuthCacheEnabled

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

Required	no		
Component	**componentName**	AuthCacheCtrlr	

Variable	**variableName**	Enabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable exists and reports a value of *true*, Authorization Cache is enabled.		

2.4.2. AuthCacheAvailable

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

2.4.3. AuthCacheLifeTime

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

2.4.4. AuthCacheStorage

Required	no	
Component	**componentName**	AuthCacheCtrlr

Variable	**variableName**	Storage	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	The maximum number of bytes
Description	Indicates the number of bytes currently used by the Authorization Cache. MaxLimit indicates the maximum number of bytes that can be used by the Authorization Cache.		

2.4.5. AuthCachePolicy

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

2.4.6. AuthCacheDisablePostAuthorize

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

2.5. Local Authorization List Management related

2.5.1. LocalAuthListEnabled

Required	no	
Component	**componentName**	LocalAuthListCtrlr

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

2.5.2. `LocalAuthListEntries`

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

2.5.3. `LocalAuthListAvailable`

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

2.5.4. `ItemsPerMessageSendLocalList`

Required	when `LocalAuthListAvailable` is *true*	
Component	**componentName**	LocalAuthListCtrlr

Variable	**variableName**	ItemsPerMessage	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer

2.5.5. `BytesPerMessageSendLocalList`

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

2.5.6. `LocalAuthListStorage`

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

2.5.7. `LocalAuthListDisablePostAuthorize`

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

2.5.8. LocalAuthListSupportsExpiryDateTime

Required	no		
Component	**componentName**	LocalAuthListCtrlr	
Variable	**variableName**	SupportsExpiryDateTime	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	When set to *true* Charging Station will disregard idTokens for authorization as if not present in the Local Authorization List when current date/time is past the value of *cacheExpiryDateTime*. Note, that *cacheExpiryDateTime* does not affect the behavior of SendLocalListRequest or GetLocalListRequest messages.		

2.6. Transaction related

2.6.1. EVConnectionTimeOut

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

2.6.2. StopTxOnEVSideDisconnect

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

2.6.3. TxBeforeAcceptedEnabled

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

2.6.4. TxStartPoint

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

2.6.5. TxStopPoint

Required	yes		
Component	**componentName**	TxCtrlr	

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

2.6.6. TxStartStopPoint values

2.6.6.1. TxStartPoint values

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

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

2.6.6.2. TxStopPoint values

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

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

table:

Value	Description
ParkingBayOccupancy	An object (probably an EV) is no longer detected in the parking/charging bay.
EVConnected	One or both ends of the Charging Cable have been disconnected (if this can be detected, else detection of a cable being unplugged from the socket), or for wireless charging: communication between EVSE and EV is lost.
Authorized	Driver or EV is no longer authorized, this can also be some form of anonymous authorization like a start button. The end of authorization will cause the Charging Station to stop the energy transfer, after which the TransactionEventRequest with eventType = Ended will be transmitted.
PowerPathClosed	All preconditions for charging are no longer met. This event is the logical OR of EVConnected and Authorized and should be used if a transaction is supposed to end when EV is disconnected and/or deauthorized. This will cause the Charging Station to stop the energy transfer, after which the TransactionEventRequest with eventType = Ended will be transmitted. It is exactly the same as having the values EVConnected, Authorized in `TxStopPoint`. Despite its name, this event is not related to the state of the power relay.
EnergyTransfer	Energy is not being transferred between EV and EVSE. This is not recommended to use as a `TxStopPoint`, because it will stop the transaction as soon as EV or EVSE (temporarily) suspend the charging.
DataSigned	This condition has no meaning as a `TxStopPoint` and should not be used as such.

2.6.7. `MaxEnergyOnInvalidId`

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

2.6.8. `StopTxOnInvalidId`

Required	yes	
Component	**componentName**	TxCtrlr

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

2.7. Metering related

2.7.1. SampledDataEnabled

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

2.7.2. SampledDataAvailable

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

2.7.3. SampledDataSignReadings

Required	no		
Component	**componentName**	SampledDataCtrlr	
Variable	**variableName**	SignReadings	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean

Description	If set to *true*, the Charging Station SHALL include signed meter values in the TransactionEventRequest to the CSMS. Some Charging Stations might only be able to sign `Transaction.Begin` and `Transaction.End` meter values. When a Charging Station does not support signed meter values it SHALL NOT report this variable.

2.7.4. SampledDataTxEndedMeasurands

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

2.7.5. SampledDataTxEndedInterval

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

2.7.6. SampledDataTxStartedMeasurands

Required	yes

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

2.7.7. SampledDataTxUpdatedMeasurands

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

2.7.8. SampledDataTxUpdatedInterval

Required	yes		
Component	**componentName**	SampledDataCtrlr	
Variable	**variableName**	TxUpdatedInterval	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	s
		dataType	integer

Description	Interval between sampling of metering (or other) data, intended to be transmitted via TransactionEventRequest (eventType = Updated) messages. For transaction data (evseId>0), samples are acquired and transmitted periodically at this interval from the start of the charging transaction. A value of "0" (numeric zero), by convention, is to be interpreted to mean that no sampled data should be transmitted during the transaction.

2.7.9. AlignedDataEnabled

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

2.7.10. AlignedDataAvailable

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

2.7.11. AlignedDataMeasurands

Required	yes		
Component	**componentName**	AlignedDataCtrlr	
Variable	**variableName**	Measurands	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	MemberList
		maxLimit	The maximum length of the CSV formatted string, to be defined by the implementer.

Description	Clock-aligned measurand(s) to be included in MeterValuesRequest or TransactionEventRequest, every `AlignedDataInterval` seconds. For all the allowed values see: Measurand. The Charging Station reports the list of supported Measurands in VariableCharacteristicsType.valuesList of this variable. This way the CSMS knows which Measurands it can put in the `AlignedDataMeasurands`.

2.7.12. `AlignedDataInterval`

Required	yes		
Component	**componentName**	AlignedDataCtrlr	
Variable	**variableName**	Interval	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	s
		dataType	integer
Description	Size (in seconds) of the clock-aligned data interval, intended to be transmitted in the MeterValuesRequest or TransactionEventRequest message. This is the size (in seconds) of the set of evenly spaced aggregation intervals per day, starting at 00:00:00 (midnight). For example, a value of 900 (15 minutes) indicates that every day should be broken into 96 15-minute intervals. When clock aligned data is being transmitted, the interval in question is identified by the start time and (optional) duration interval value, represented according to the ISO8601 standard. A value of "0" (numeric zero), by convention, is to be interpreted to mean that no clock-aligned data should be transmitted.		

2.7.13. `AlignedDataSendDuringIdle`

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

2.7.14. `AlignedDataSignReadings`

Required	no	
Component	**componentName**	AlignedDataCtrlr

Variable	**variableName**	SignReadings	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If set to *true*, the Charging Station SHALL include signed meter values in the SampledValueType in the TransactionEventRequest to the CSMS for those measurands defined in `AlignedDataTxEndedMeasurands`. When a Charging Station does not support signed meter values it SHALL NOT report this variable.		

2.7.15. `AlignedDataTxEndedMeasurands`

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

2.7.16. `AlignedDataTxEndedInterval`

Required	yes		
Component	**componentName**	AlignedDataCtrlr	
Variable	**variableName**	TxEndedInterval	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**unit**	s
		dataType	integer
Description	Size (in seconds) of the clock-aligned data interval, intended to be transmitted in the TransactionEventRequest (eventType = Ended) message. This is the size (in seconds) of the set of evenly spaced aggregation intervals per day, starting at 00:00:00 (midnight). For example, a value of 900 (15 minutes) indicates that every day should be broken into 96 15-minute intervals. When clock aligned data is being collected, the interval in question is identified by the start time and (optional) duration interval value, represented according to the ISO8601 standard. All intervals are transmitted (if so enabled) at the end of the transaction in 1 TransactionEventRequest (eventType = Ended) message. This is not a recommended practice, since the size of the message can become very large.		

2.7.17. PublicKeyWithSignedMeterValue

Required	no		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	PublicKeyWithSignedMeterValue	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	OptionList
		valueList	Never,OncePerTransaction,EveryMeterValue
Description	This Configuration Variable can be used to configure whether a public key needs to be sent with a signed meter value. Note, that the field is required, so it needs to be present as an empty string when the public key is not sent.		

2.7.18. SampledDataRegisterValuesWithoutPhases

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

2.8. Reservation related

2.8.1. ReservationEnabled

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

2.8.2. ReservationAvailable

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

2.8.3. ReservationNonEvseSpecific

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

2.9. Smart Charging related

2.9.1. SmartChargingEnabled

Required	no		
Component	**componentName**	SmartChargingCtrlr	
Variable	**variableName**	Enabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	Whether Smart Charging is enabled.		

2.9.2. SmartChargingAvailable

Required	no	
Component	**componentName**	SmartChargingCtrlr

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

2.9.3. ACPhaseSwitchingSupported

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

2.9.4. ChargingProfileMaxStackLevel

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

2.9.5. ChargingScheduleChargingRateUnit

Required	yes		
Component	**componentName**	SmartChargingCtrlr	
Variable	**variableName**	RateUnit	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	MemberList

Description	A list of supported quantities for use in a ChargingSchedule. Allowed values: 'A' and 'W'		

2.9.6. PeriodsPerSchedule

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

2.9.7. ExternalControlSignalsEnabled

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

2.9.8. NotifyChargingLimitWithSchedules

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

2.9.9. Phases3to1

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

2.9.10. ChargingProfileEntries

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

2.9.11. LimitChangeSignificance

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

2.10. Tariff & Cost related

2.10.1. TariffEnabled

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

2.10.2. TariffAvailable

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

2.10.3. TariffFallbackMessage

Required for Charging Stations supporting Tariff Information.

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

2.10.4. CostEnabled

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

2.10.5. CostAvailable

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

2.10.6. TotalCostFallbackMessage

Required for Charging Stations supporting Tariff Information.

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

2.10.7. Currency

Required for Charging Stations supporting Tariff Information.

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

2.11. Diagnostics related

2.11.1. MonitoringEnabled

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

2.11.2. MonitoringAvailable

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

2.11.3. ItemsPerMessageClearVariableMonitoring

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

2.11.4. ItemsPerMessageSetVariableMonitoring

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

2.11.5. BytesPerMessageClearVariableMonitoring

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

2.11.6. BytesPerMessageSetVariableMonitoring

Required	yes

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

2.11.7. `OfflineMonitoringEventQueuingSeverity`

Required	no		
Component	**componentName**	MonitoringCtrlr	
Variable	**variableName**	OfflineQueuingSeverity	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	integer
Description	When set and the Charging Station is offline , the Charging Station shall queue any NotifyEventRequest messages triggered by a monitor with a severity number equal to or lower than the severity configured here. Value ranging from 0 (Emergency) to 9 (Debug).		

2.11.8. `ActiveMonitoringBase`

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

2.11.9. `ActiveMonitoringLevel`

Required	no		
Component	**componentName**	MonitoringCtrlr	

Variable	**variableName**	ActiveMonitoringLevel	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Shows the currently used MonitoringLevel. Valid values are severity levels of SetMonitoringLevelRequest: 0-9.		

2.12. Display Message related

2.12.1. DisplayMessageEnabled

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

2.12.2. DisplayMessageAvailable

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

2.12.3. NumberOfDisplayMessages

Required	yes	
Component	**componentName**	DisplayMessageCtrlr

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

2.12.4. DisplayMessageSupportedFormats

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

2.12.5. DisplayMessageSupportedPriorities

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

2.13. Charging Infrastructure related

2.13.1. Available

Required	yes

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

2.13.2. AvailabilityState

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

2.13.3. AllowReset

Required	no

Component	**componentName**	EVSE	
	evse	*	
Variable	**variableName**	AllowReset	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	Component can be reset. Can be used to announce that an EVSE can be reset individually.		

2.13.4. `ConnectorType`

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

2.13.5. `PhaseRotation`

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

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

2.13.6. SupplyPhases

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

2.13.7. Power

Required	yes (*maxLimit only*)

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

2.13.8. Example Reporting of EVSEs and Connectors via device model

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

Component				**Variable**		**VariableAttribute**		**VariableCharacteristics**		
name	**evseid**	**evseconnectorId**	**instance**	**name**	**instance**	**type**	**value**	**dataType**	**maxLimit**	**supportsMonitoring**
ChargingStation				Available		Actual	true	boolean		false
ChargingStation				AvailabilityState		Actual	Available	boolean		false
ChargingStation				SupplyPhases		Actual	integer	3		false
ChargingStation				ACCurrent	"L1"	Actual	decimal	45.0		true
ChargingStation				ACCurrent	"L2"	Actual	decimal	44.9		true
ChargingStation				ACCurrent	"L3"	Actual	decimal	44.9		true
EVSE	1		"left"	Available		Actual	true	boolean		false
EVSE	1		"left"	AvailabilityState		Actual	Available	optionList		false
EVSE	1		"left"	SupplyPhases		Actual	3	integer		false

Component				Variable	VariableAttribute		VariableCharacteristics		
EVSE	1		"left"	Power	Actual	0.0	decimal	22000.0	true
Connector	1	1		Available	Actual	true	boolean		false
Connector	1	1		ConnectorType	Actual	sType2	string		false
Connector	1	1		SupplyPhases	Actual	3	integer		false
EVSE	2		"right"	Available	Actual	true	boolean		false
EVSE	2		"right"	AvailabilityState	Actual	Occupied	optionList		false
EVSE	2		"right"	SupplyPhases	Actual	0	integer		false
EVSE	2		"right"	Power	Actual	41000.0	decimal	50000.0	true
Connector	2	1		Available	Actual	true	boolean		false
Connector	2	1		AvailabilityState	Actual	Occupied	optionList		false
Connector	2	1		ConnectorType	Actual	cCCS2	string		false
Connector	2	1		SupplyPhases	Actual	0	integer		false
Connector	2	2		Available	Actual	true	boolean		false
Connector	2	2		AvailabilityState	Actual	Unavailable	optionList		false
Connector	2	2		ConnectorType	Actual	cG105	string		false
Connector	2	2		SupplyPhases	Actual	0	integer		false

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

2.14. ISO 15118 Related

2.14.1. `CentralContractValidationAllowed`

Required	no	
Component	**componentName**	ISO15118Ctrlr

Variable	**variableName**	CentralContractValidationAllowed	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable exists and has the value *true*, then Charging Station can provide a contract certificate that it cannot validate, to the CSMS for validation as part of the AuthorizeRequest.		

2.14.2. ContractValidationOffline

Required	yes		
Component	**componentName**	ISO15118Ctrlr	
Variable	**variableName**	ContractValidationOffline	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable is *true*, then Charging Station will try to validate a contract certificate when it is offline.		

2.14.3. ProtocolSupportedByEV

Required	no		
Component	**componentName**	ConnectedEV	
	evse	*	
Variable	**variableName**	ProtocolSupportedByEV	
	variableInstance	<Priority>	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	string
Description	A string with the following comma-separated items: "<uri>,<major>,<minor>". This is information from the SupportedAppProtocolReq message from ISO 15118 Each priority is given its own variable instance. Priority is a number from 1 to 20 as a string. E.g. "1" or "2". Example: - ConnectedEV.ProtocolSupportedByEV["1"] "urn:iso:15118:2:2013:MsgDef,2,0" - ConnectedEV.ProtocolSupportedByEV["2"] "urn:iso:15118:2:2010:MsgDef,1,0"		

2.14.4. ProtocolAgreed

Required	no

Component	**componentName**	ConnectedEV	
	evse	*	
Variable	**variableName**	ProtocolAgreed	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	string
Description	A string with the following comma-separated items: "<uri>,<major>,<minor>". This is the protocol uri and version information that was agreed upon between EV and EVSE in the supportedAppProtocolReq handshake from ISO 15118. Example: "urn:iso:15118:2:2013:MsgDef,2,0"		

2.14.5. ISO15118PnCEnabled

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

2.14.6. ISO15118V2GCertificateInstallationEnabled

Required	no		
Component	**componentName**	ISO15118Ctrlr	
Variable	**variableName**	V2GCertificateInstallationEnabled	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	If this variable is *true*, then ISO 15118 V2G Charging Station certificate installation as described by use case A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station is enabled. If this variable is *false*, then ISO 15118 V2G Charging Station certificate installation as described by use case A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station is disabled.		

2.14.7. ISO15118ContractCertificateInstallationEnabled

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

2.14.8. ISO15118RequestMeteringReceipt

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

2.14.9. ISO15118SeccId

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

2.14.10. ISO15118CountryName

Required	no		
Component	**componentName**	ISO15118Ctrlr evse * (optional)	
Variable	**variableName**	CountryName	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	The countryName of the SECC in the ISO 3166-1 format. It is used as the **countryName** © of the SECC leaf certificate. Example: "DE"		

2.14.11. ISO15118OrganizationName

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

2.14.12. ISO15118EvseId

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