Implementation Report

OpenADR 3.0 Reference Implementation for Smart Energy Systems

OpenADR 3.0 in the Energy Sector

Connecting Grid Operators, Energy Suppliers, District Operators, E-Mobility Backends and Charge Point Operators via OpenADR

m8mit / Kraftwerk Software Systems GmbH (formerly msu solutions GmbH)
Lead Author: Dr.-Ing. Hannes Schwarz
July 2026

AVAT Automation GmbH
Andreas Jeni

Zentrum für Sonnenenergie- und Wasserstoff-Forschung Baden-Württemberg
Linda Roth

VIVAVIS AG
Thomas Krämer

Further Consortium Partners:
Power Plus Communications AG
Technische Hochschule Ulm
Stadtwerke Schwäbisch Hall GmbH
Stadtwerke Ulm/Neu-Ulm Netze GmbH
Stadtwerke Ludwigsburg Kornwestheim GmbH
Stadtwerke Bernburg GmbH

Table of Contents

Table of Contents 2
1. Introduction 3
2. Objectives and Motivation 4
2.1 Selection of the Communication Standard 4
3. OpenADR Standard Fundamentals 5
3.1 Role Model: VTN and VEN 5
3.2 Core OpenADR Objects 5
3.3 VEN Registration and Setup 5
3.4 Flexibility Querying and Tariff-Based Control 6
4. System Architecture 7
4.1 Role Distribution 7
4.2 Technical Infrastructure and Endpoints 7
4.3 Communication Layers 7
5. Master Data Exchange 8
5.1 Master Data ZSW – AVAT 8
5.2 Master Data AVAT – KSS 8
6. Operational Data Exchange 9
6.1 Scenario A: Automated Price Signal Response (Grid → District → CPO) 9
6.2 Scenario D: Direct Grid Operator Communication 9
7. Use Case Specification 10
7.1 Scenario A: Automated Response to Dynamic Grid Price Signals 10
7.1.1 Sequence Diagram (Scenario A: Automated Price Signal Response) 10
7.2 Scenario B: EV Charging Flexibility Management and Tariffing 11
7.2.1 Sequence Diagram (Scenario B) 11
7.2.2 Operating States (OPERATING_STATE) 12
7.3 Scenario C: Revenue Transparency from Grid-Serving Operation 13
7.3.1 Workflow 13
8. Technical OpenADR Object Specification 14
8.1 Price Event (Scenario A) 14
8.2 Forecast Report (Scenario A) 14
8.3 Usage Report (Scenario A) 15
8.4 Operation State Event / Anchor Event (Scenario B, Step 2) 15
8.5 Operation State Report (Scenario B, Step 4) 16
8.6 Setpoint Event (Scenario B, Step 5) 16
8.7 Usage Report for Load Management (Scenario B, Step 6) 17
9. Detailed Communication Workflow (Round-Trip) 18
9.1 Initialization Phase 18
9.2 Ongoing Operation 18

10. Test Execution and Results 19
Test Protocol 19
Dynamic Grid Tariffs (ZSW) 20
Schedule Optimization (AVAT) 20
Charge Management (KSS) 21
11. Collaboration with the OpenADR Alliance 22
12. Summary and Outlook 22
12.1 Achieved Results 22
12.2 Open Items 22
12.3 Outlook 22

1. Introduction

This report documents a reference implementation of OpenADR 3.0 in the energy sector. The objective was to connect Charge Point Operators (CPO) to grid management systems via the international OpenADR standard, focusing on standardized communication between distribution system operators, district optimization systems, and charging infrastructure.

This report documents the results of the implementation work in its final form. It summarizes the technical developments, the system architecture, the defined data exchange mechanisms, the specified use cases, and the tests conducted.

The implementation work was led by Kraftwerk Software Systems GmbH (KSS, formerly msu solutions GmbH). The following project partners played a significant role: Zentrum für Sonnenenergie- und Wasserstoff-Forschung Baden-Württemberg (ZSW), AVAT Automation GmbH, VIVAVIS AG, and Kraftwerk Software Systems GmbH.

2. Objectives and Motivation

The increasing electrification of the transport sector poses significant challenges for distribution system operators. In particular, the growing number of EV charging points requires coordinated charge management to prevent grid congestion while leveraging user flexibility in a market-based manner.

The project aimed to develop and implement a communication interface based on the international OpenADR 3.0 standard that enables coordinated charge management across actors from grid operation (VIVAVIS), district optimization (AVAT), and e-mobility (KSS / m8mit). Additionally, transparent revenue reporting from grid-serving operation (ZSW) was to be established.

2.1 Selection of the Communication Standard

During the conceptual phase, a thorough evaluation of various communication standards was conducted. In addition to OpenADR, OSCP (Open Smart Charging Protocol) was given particular consideration. The decision in favor of OpenADR 3.0 was primarily based on its significantly broader object model, which enables not only simple capacity limits but also complex tariff and price signals, forecasts, and detailed reporting.

Furthermore, as an internationally recognized standard (IEC 62746-10-1), OpenADR provides a broad basis for interoperability and future-proofing. In particular, the use of RESTful APIs and JSON-based data structures facilitates integration into modern IT architectures.

3. OpenADR Standard Fundamentals

OpenADR (Open Automated Demand Response) is an international communication standard for the automated control of energy consumers and distributed generation assets. Version 3.0 defines a clearly structured role model and standardized objects for communication.

3.1 Role Model: VTN and VEN

OpenADR 3.0 distinguishes between two central roles:

- **Virtual Top Node (VTN):** The VTN is the higher-level node that creates programs, events, and price signals and distributes them to the subordinate VENs. In this implementation, both ZSW (grid regulation algorithm) and AVAT (district optimization) assume the role of a VTN.
- **Virtual End Node (VEN):** The VEN is the subordinate node that reacts to the VTN's signals, creates reports, and returns measurement data. KSS (Charge Point Operator) acts as a VEN. AVAT simultaneously takes on the roles of a VEN (towards ZSW) and a VTN (towards KSS).
- **Resource:** The actual energy resources (e.g., charging stations, districts) that are controlled by the VEN.
- **Business Logic (BL):** The business logic that performs optimization and decision-making outside the OpenADR protocol.

3.2 Core OpenADR Objects

Communication in OpenADR 3.0 is based on the following core object types:

- **Program:** A program represents a demand response tariff or a control program. It contains metadata describing which types of events and reports are expected. In this project, programs are created for price incentives (ZSW → AVAT) and for charging sessions (AVAT → KSS).
- **Event:** An event is a time-bound control signal sent from the VTN to the VEN. It can contain price signals (price_event), operating state requests (operation_state_event), or setpoints (setpoint_event).
- **Report:** A report provides feedback from the VEN to the VTN. It contains measurement or forecast data such as consumption values (usage_report), load forecasts (ops_forecast_report), or operating states (operation_state_report).
- **Subscription/Webhook:** VENs register for notifications about events and reports via a webhook mechanism. This enables event-driven communication without the VEN having to continuously poll the VTN.

3.3 VEN Registration and Setup

VEN registration follows a multi-stage process. First, the VEN creates a representation of itself on the VTN (register VEN). The actual resource registration (charging points, plants) occurs in a subsequent step. Webhooks for event and report notifications are registered separately.

In the project context, this means the following: KSS registers a list of charging points with AVAT. AVAT in turn registers as a VEN with the ZSW VTN and reports aggregated data per grid connection point (MaLo-ID).

3.4 Flexibility Querying and Tariff-Based Control

A central aspect of OpenADR-based charge management is the querying of end-user flexibility. The process is as follows:

1. The end user (EV driver) is asked about their flexibility via a user interface (app/webapp) connected to the VEN (KSS). The user can specify their desired departure time, minimum range, or tariff preference. This choice is closely linked to the selected tariff model and the active program.
2. The selected flexibility is transmitted from the VEN to the VTN. In the simplest case, flexibility is predefined by the selected flex tariff (e.g., "green" = full flexibility, "red" = no flexibility).
3. Based on the available flexibility, dynamic grid tariffs (from ZSW), and spot market electricity prices, AVAT's business logic creates an optimized charging schedule per charging point.
4. The calculated schedule is transmitted from the VTN (AVAT) to the VEN (KSS) and converted into control signals for the charging station.

This concept enables market-based and grid-serving utilization of end-user charging flexibility without unnecessarily compromising user convenience.

4. System Architecture

The system architecture developed as part of the OpenADR implementation is based on the OpenADR 3.0 role model and comprises three communication layers: grid management (ZSW), district optimization (AVAT), and charge point operation (KSS).

4.1 Role Distribution

Partner	OpenADR Role	Function	Server/Endpoint
ZSW	VTN	Grid regulation algorithm, price incentives	OpenADR server
AVAT	VTN / VEN	District optimization	OpenADR server, OpenADR client
KSS / m8mit	VEN	Charge Point Operator (CPO)	OpenADR client
VIVAVIS	(VTN)	Grid control center	Separate connection

4.2 Technical Infrastructure and Endpoints

The technical infrastructure is based on a client-server architecture with webhook-based notifications. The specific configuration is as follows:

```
Communication Flow:
  ZSW (VTN) ── price_events → AVAT (VEN)
  AVAT (VEN) ── usage_reports → ZSW (VTN)
  KSS (VEN) ── program_reports → AVAT (VTN)
  AVAT (VTN) ── setpoint_events → KSS (VEN)
  KSS (VEN) ── usage_reports → AVAT (VTN)
```

AVAT operates both a VTN (for communication with KSS) and a VEN (towards ZSW). AVAT registers webhook subscriptions with the ZSW VTN and receives price events as notifications.

4.3 Communication Layers

- **Grid Control Center – Assist&Control (VIVAVIS – ZSW):** Transmission of grid status. The OpenADR connection between VIVAVIS and ZSW was not implemented within the project scope; communication takes place via a separate interface.
- **Assist&Control – District Optimization (ZSW – AVAT):** Transmission of price incentives and dynamic grid tariffs (ZSW → AVAT) as well as electricity consumption/feed-in forecasts (AVAT → ZSW). Communication via OpenADR (price_event, forecast_report, usage_report).
- **District Optimization – CPO (AVAT – KSS):** Transmission of charge setpoints (AVAT → KSS) as well as charge power, charging point status, and charging session information (KSS → AVAT). Communication via OpenADR (setpoint_event, operation_state_report, usage_report).
- **CPO – Charging Stations:** Transmission of charge power setpoints to individual charging stations. Reading of charging session information for reporting back to AVAT. Communication via OCPP (not OpenADR).

5. Master Data Exchange

The master data exchange forms the foundation for operational communication. It takes place once during registration and is updated when changes occur.

5.1 Master Data ZSW – AVAT

Sender	Recv.	Content	Purpose	OpenADR
AVAT	ZSW	MaLo-ID	Grid connection identification	(future)
AVAT	ZSW	Geocoordinates (lat, lon)	Localization	(future)
AVAT	ZSW	MaStR-Nr.	Renewable energy plant identification	(future)

5.2 Master Data AVAT – KSS

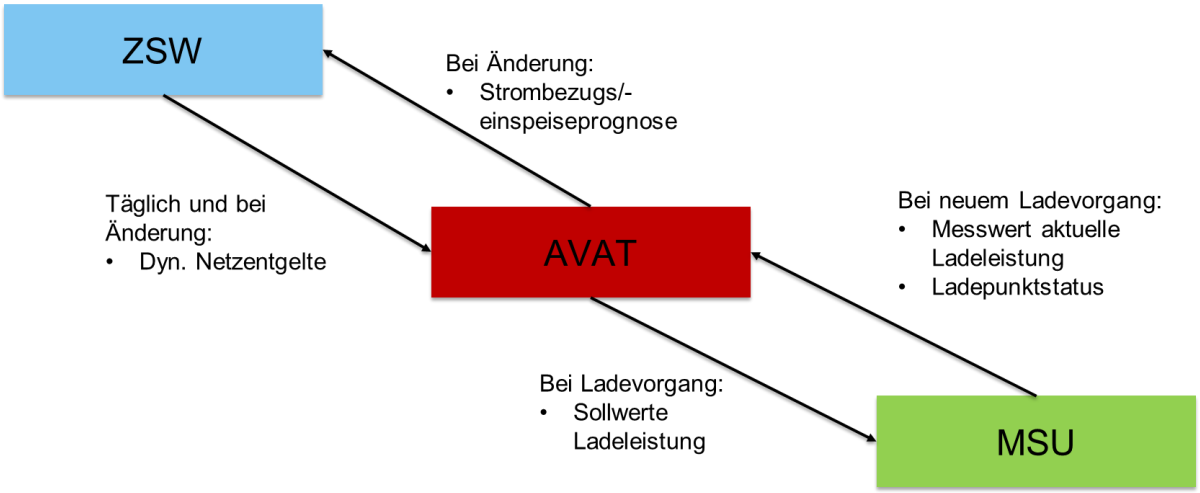
Sender	Recv.	Content	Purpose	OpenADR
KSS	AVAT	EVSEID	Charging point identification	(future)
KSS	AVAT	Geocoordinates (lat, lon)	Localization	(future)
KSS	AVAT	Location	Localization	(future)
KSS	AVAT	Rated power of charging point (alternatively voltage/current)	Modeling	(future)
KSS	AVAT	Type AC/DC	Correct modeling	(future)
KSS	AVAT	Controllable yes/no	Correct modeling	(future)
KSS	AVAT	Available tariffs	Charging strategies	(future)

6. Operational Data Exchange

Operational data exchange comprises time series and live data and varies depending on the use case.

6.1 Scenario A: Automated Price Signal Response (Grid → District → CPO)

- **ZSW → AVAT: Price incentives (on change) via OpenADR Program + Events**
- **AVAT → ZSW: Electricity consumption/feed-in forecast (on change) as OpenADR Event**
- **AVAT → KSS: Charge power setpoints (live) as OpenADR Event**
- **KSS → AVAT: Current charge power, charging point status (live), charging session start with tariff selection as Usage Reports**
- **VIVAVIS → ZSW: Grid status (on change), communication via separate interface**
- **ZSW → KSS/AVAT: Dynamic grid tariffs (daily and on change)**



6.2 Scenario D: Direct Grid Operator Communication

Simplified variant: direct communication between VIVAVIS and KSS. Charge power setpoints from VIVAVIS to KSS, feedback of live data (charge power, charging point status, operating state) from KSS to VIVAVIS.

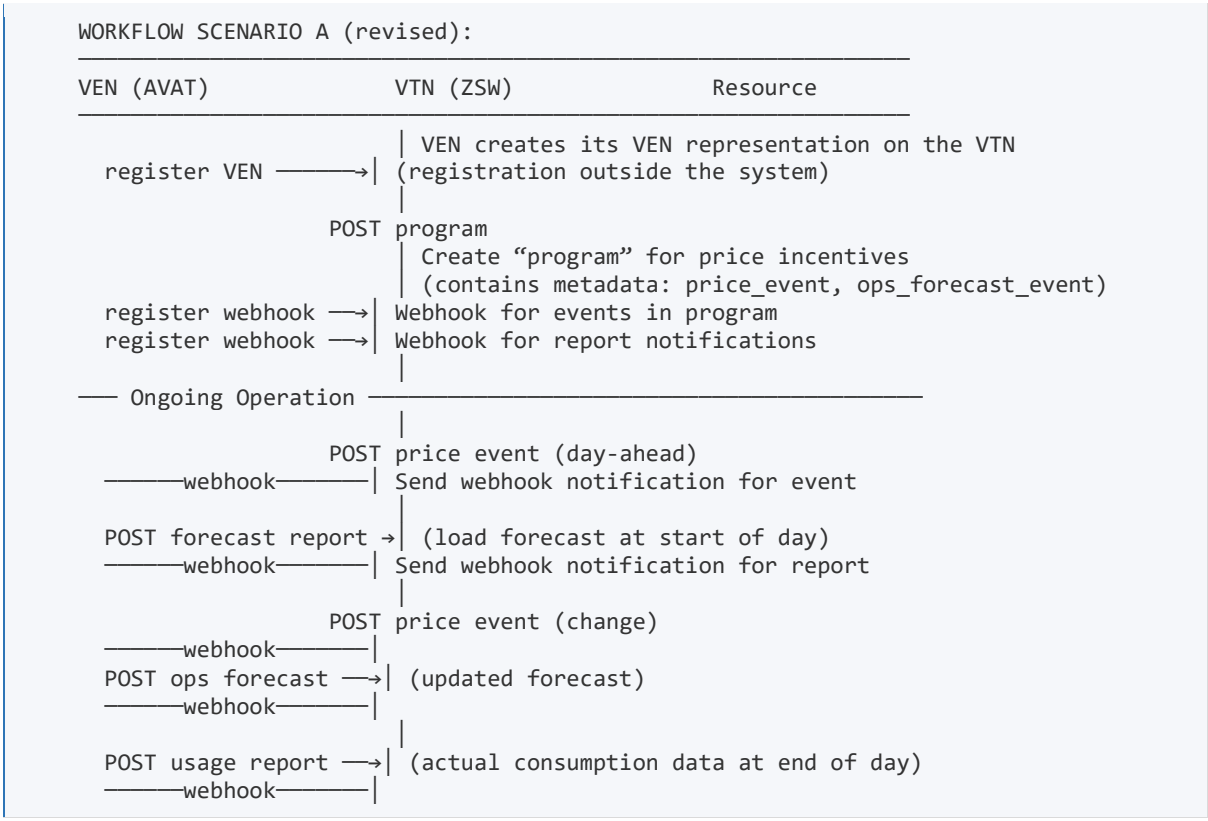
7. Use Case Specification

7.1 Scenario A: Automated Response to Dynamic Grid Price Signals

This use case describes the workflow in which customer assets (e.g., a district) automatically respond to price incentives from the grid management system. The overall process is coordinated by the grid regulation algorithm (ZSW) and the district optimization system (AVAT).

7.1.1 Sequence Diagram (Scenario A: Automated Price Signal Response)

Participants: VEN (AVAT), VTN (ZSW), Resource (e.g., district), Business Logic (BL)



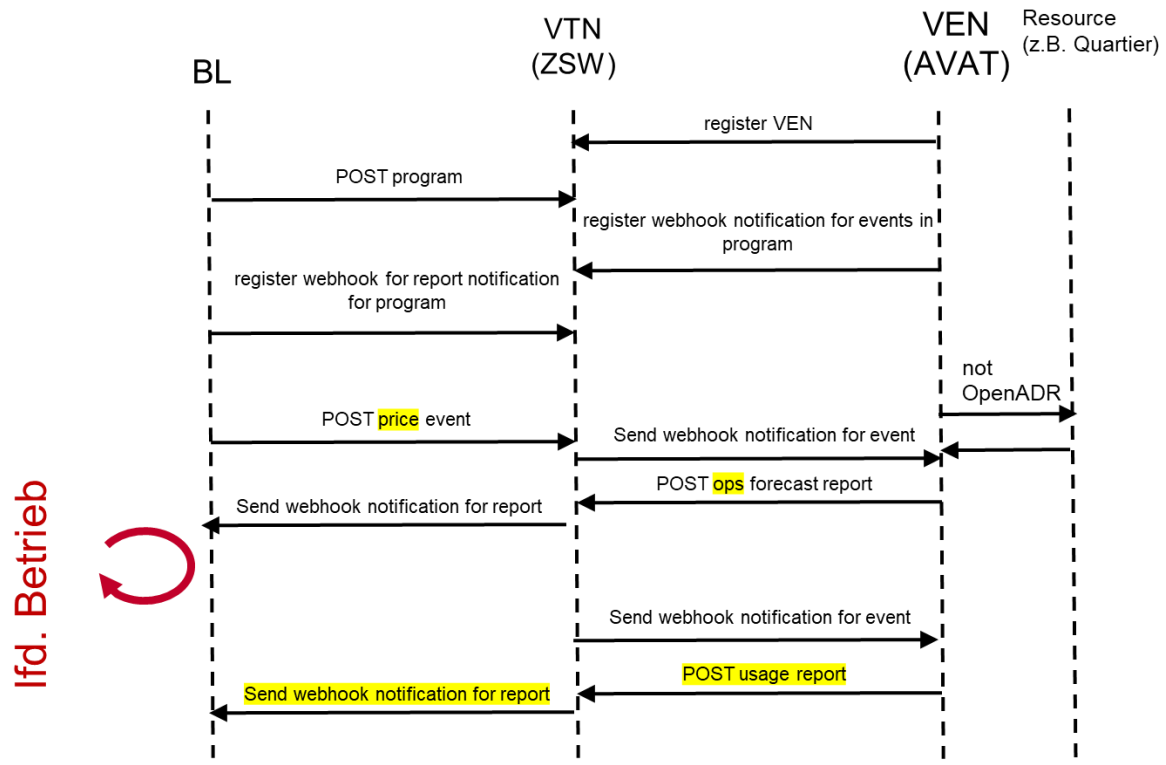


Figure: Sequence diagram for Scenario A: exchange of price events, forecast and usage reports between VEN (AVAT) and VTN (ZSW).

7.2 Scenario B: EV Charging Flexibility Management and Tariffing

This use case describes the communication between the district optimization system (AVAT as VTN) and the CPO (KSS as VEN) for controlling individual charging sessions. It covers the complete process from the start of a charging session to continuous metering data transmission.

7.2.1 Sequence Diagram (Scenario B)

Participants: Resource (charging point), VTN (AVAT), VEN (KSS), Business Logic (BL)

```
WORKFLOW SCENARIO B (Flexibility Tariffing and Billing):
──────────────────────────────────────────────────────────────────
Resource (charging point)  VEN (KSS)          VTN (AVAT)        BL
──────────────────────────────────────────────────────────────────
STEP 1: Registration
Each charging point registers as a VEN
                         register VEN ──→ VTN

STEP 2: Program Creation
  VTN creates a “program” per tenant + Anchor Event (“SIMPLE”)
                                  POST program
                                  POST dummy-event in program
                         register webhook for events

STEP 3: Webhook Registration
VTN registers webhooks for reports and events
                                  register webhook for event report

STEP 4: New Charging Session (non-OpenADR)
Report for charging session with tariff info
  OPERATING_STATE → IDLE_NORMAL
```

```
                              POST report ———→ VTN —→ BL calculates
                              —— webhook notification for report ——

    STEP 5: Charge Setpoints (ongoing operation)
    New setpoints are dispatched (CONTROL_SETPOINT)
    Can be sent multiple times during a charging session
                                        POST event ———→ VEN
                              register webhook for event report
                              —— webhook notification for event ——

    STEP 6: Metering Data Transmission (every 15 min)
    Usage Report until charging session completes
      OPERATING_STATE → NORMAL
                              POST event-report —→ VTN
                              —— webhook notification for report ——
```

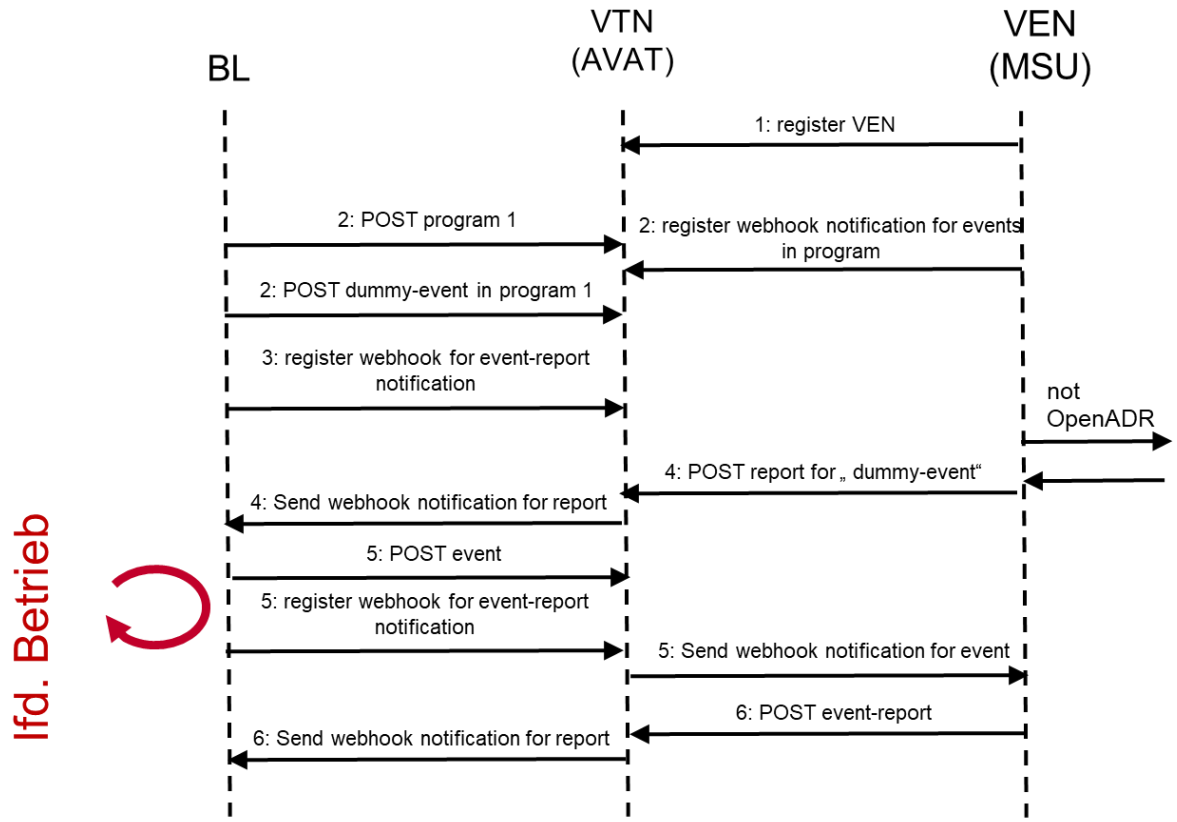


Figure: Sequence diagram for Scenario B: registration, anchor event, setpoint events and 15-minute usage reports between VEN (MSU) and VTN (AVAT).

7.2.2 Operating States (OPERATING_STATE)

Status	Description
IDLE_NORMAL	Vehicle connected, not charging
RUNNING_NORMAL	Vehicle connected, charging unrestricted
RUNNING_CURTAILED	Vehicle connected, charging curtailed (power limit active)
NORMAL	No vehicle connected, no charging session

7.3 Scenario C: Revenue Transparency from Grid-Serving Operation

This use case builds directly on Scenario A (Automated Price Signal Response). While Scenario A covers the customer assets' response to dynamic price signals, Scenario C focuses on the transparent presentation of the resulting additional revenues from grid-serving operation.

7.3.1 Workflow

Participants: Plant operator (the utility partner), VEN (AVAT), VTN (ZSW), Resource

The combined heat and power (CHP) plant is operated by the AVAT optimization system in an electricity-price-driven mode (based on the EPEX spot market). Within Scenario A (Automated Price Signal Response), the incentive signals from the grid management system lead to shifts in the CHP unit's operating schedule. These shifts are only carried out if they result in a financial benefit for the plant operator.

Scenario C (Revenue Transparency) makes the resulting additional revenues from grid-serving operation transparent in the AVAT optimization system for the combined heat and power plant.

8. Technical OpenADR Object Specification

This chapter describes in detail the OpenADR objects defined in the project. The specifications are based on the OpenADR 3.0.1 documentation and define the specific payloads, intervals, and report descriptors for each use case.

8.1 Price Event (Scenario A)

The Price Event is created by the VTN (ZSW) and contains price signals in 15-minute intervals. It requests both a Forecast Report (prospective load forecast) and a Usage Report (retrospective actual consumption data) from the VEN.

```
objectType: EVENT
eventName: price_event
ID: <uuid>
programID: <uuid>
createdDateTime: 2023-02-09T10:30:00.000Z  // Creation timestamp

intervalPeriod:
  start: 2023-02-10T00:00:00.000Z  // Desired start (e.g., 00:00 the next day)
  duration: PT15M                   // 15-minute intervals

// Report Descriptor 1: Requested load forecast (USAGE_FORECAST)
reportDescriptors[0]:
  payloadType: USAGE_FORECAST
  startInterval: 0       // Report sent from this interval
  numIntervals: 96       // 96 intervals = 24 hours
  historical: false      // Reporting for future intervals
  frequency: -1          // All intervals at once
  repeat: 1              // Generate exactly 1 report
  readingType: FORECAST

// Report Descriptor 2: Requested metering data (USAGE)
reportDescriptors[1]:
  payloadType: USAGE
  startInterval: -1      // From last interval
  numIntervals: 96       // 96 intervals
  historical: true       // Retrospective (last interval)
  frequency: -1          // All intervals at once
  repeat: 1              // Generate exactly 1 report
  readingType: DIRECT_READ
```

8.2 Forecast Report (Scenario A)

The Forecast Report is created by the VEN (AVAT) in response to a Price Event. It contains the forecasted consumption values per resource point in 15-minute intervals.

```
objectType: REPORT
reportName: usage_forecast_report
ID: <uuid>
programID: <uuid>
eventID: <uuid>          // Reference to the triggering Price Event
clientName: Avat
createdDateTime: 2023-02-10T00:00:00.000Z

intervalPeriod:
  start: 2023-02-10T00:00:00.000Z  // Copied from the Event
  duration: PT15M
```

```
payloadDescriptors:
  payloadType: USAGE
  readingType: FORECAST
  units: kWh

resources:
  resourceName: <connection point>  // e.g., #1 Bernburg, #2 Ludwigsburg
  intervalPeriod:
    start: 2023-02-10T00:00:00.000Z
    duration: PT15M
  intervals:
    payloads:
      type: USAGE
      values: [10.3, 2.4, 4.3, ...]  // kWh per 15-min interval
```

8.3 Usage Report (Scenario A)

The Usage Report contains the actual consumption data (Direct Read) in the same interval format. It is transmitted at the end of the day.

```
objectType: REPORT
reportName: usage_summary_report
ID: <uuid>
programID: <uuid>
eventID: <uuid>
clientName: Avat
createdDateTime: 2023-02-10T00:00:00.000Z

intervalPeriod:
  start: 2023-02-10T00:00:00.000Z  // Copy from Event
  duration: PT15M

payloadDescriptors:
  payloadType: USAGE
  readingType: DIRECT-READ
  units: kWh

resources:
  resourceName: <connection point>
  intervalPeriod:
    start: 2023-02-10T00:00:00.000Z
    duration: PT15M
  intervals:
    payloads:
      type: USAGE
      values: [10.3, 2.4, 4.3, ...]  // Actual kWh values
```

8.4 Operation State Event / Anchor Event (Scenario B, Step 2)

The Anchor Event is created by the VTN (AVAT) for each program. It serves as an anchor point to which charging session notifications (reports) can be sent. It contains a single report descriptor for OPERATING_STATE.

```
objectType: EVENT
eventName: operation_state_event
id: <uuid>
programID: <uuid>
createdDateTime: 2023-02-10T00:00:00.000Z
```

```
intervalPeriod:
  start: 2023-02-10T00:00:00.000Z
  duration: PT15M

reportDescriptors:
  payloadType: OPERATING_STATE
  startInterval: 0
  numIntervals: 1
  historical: false
  frequency: 0     // Once per interval
  repeat: 1
  readingType: DIRECT_READ
```

8.5 Operation State Report (Scenario B, Step 4)

The Operation State Report is sent by the VEN (KSS) at the start of a new charging session. It references the Anchor Event and contains the current operating state.

```
objectType: REPORT
reportName: operation_state_report
id: <uuid>
programID: <uuid>
eventID: <uuid>          // Reference to the Anchor Event
clientName: <ven-name>
createdDateTime: 2023-02-10T00:00:00.000Z

payloadDescriptors:
  payloadType: OPERATING_STATE
  readingType: DIRECT_READ

resources:
  resourceName: <charging-point-ID>
  intervalPeriod:
    start: 2023-02-10T00:00:00.000Z
    duration: PT15M
  intervals:
    id: 0
    payloads:
      type: OPERATION_STATE
      value: IDLE_NORMAL  // or RUNNING_NORMAL, RUNNING_CURTAILED, NORMAL
```

8.6 Setpoint Event (Scenario B, Step 5)

The Setpoint Event is sent by the VTN (AVAT) to the VEN (KSS) to curtail the charge power of a specific charging point. It contains the setpoint in kW and requests three report types.

```
objectType: EVENT
eventName: setpoint_event
id: <uuid>
programID: <uuid>
createdDateTime: 2023-02-10T00:00:00.000Z

targets: RESOURCE_NAME <charging-point-ID>

intervals:
  id: 0
  intervalPeriod:
```

```
        start: 2023-02-10T00:00:00.000Z  // Start of power curtailment
        duration: PT15M
      payloads:
        type: CONTROL_SETPOINT
        values: <charge setpoint in kW>

  reportDescriptors: (3x, one per payloadType)
    payloadType: READING | OPERATING_STATE | SETPOINT
    startInterval: 0
    numIntervals: -1     // Unlimited
    historical: false
    frequency: 1         // Every interval
    repeat: -1           // Repeat indefinitely
    readingType: DIRECT_READ
```

8.7 Usage Report for Load Management (Scenario B, Step 6)

The Usage Report for charge power curtailment is sent every 15 minutes by the VEN (KSS) and contains three payload types: metering values, operating state, and the current setpoint.

```
  objectType: REPORT
  reportName: usage_report
  id: <uuid>
  programID: <uuid>
  eventID: <uuid>          // Reference to the Setpoint Event
  clientName: <ven-name>
  createdDateTime: 2023-02-10T00:00:00.000Z

  payloadDescriptors: (3x, one per payload type)
    payloadType: READING          | readingType: DIRECT_READ | unit: kWh
    payloadType: OPERATING_STATE  | readingType: DIRECT_READ | unit: -
    payloadType: SETPOINT         | readingType: DIRECT_READ | unit: kW

  resources:
    resourceName: <charging-point-ID>
    intervalPeriod:
      start: 2023-02-10T00:00:00.000Z
      duration: PT15M
    intervals:
      id: 0
      payloads:
        type: OPERATION_STATE  |  value: IDLE_NORMAL
      payloads:
        type: READING          |  value: <meter reading>
      payloads:
        type: SETPOINT         |  value: <current power in kW>
```

9. Detailed Communication Workflow (Round-Trip)

9.1 Initialization Phase

No.	Action	Sender	Recv.	Trigger
1	VEN Registration	KSS	AVAT	Start
1	Webhook for setpoint_event, operation_state_event	KSS	AVAT	Start
1	Webhook for operation_state_report, usage_report	AVAT	KSS	Start
1	Webhook for price_events	AVAT	ZSW	Start

9.2 Ongoing Operation

No.	Name	Sender	Recv.	Auto	Trigger
1	price_event	ZSW	AVAT	–	Start
2	ops_forecast_report	AVAT	ZSW	✓	price_event
2	operation_state_event	AVAT	KSS	–	
3	operation_state_report	KSS	AVAT	✓	New charging session
4	setpoint_event	AVAT	KSS	✓	operation_state_report
4	ops_forecast_report	AVAT	ZSW	✓	operation_state_report
5	usage_report	KSS	AVAT	✓	setpoint_event
6	usage_summary_report	AVAT	ZSW	✓	End

Legend: “Auto” = automatic in response to the trigger (✓) or manually initiated (–).

10. Test Execution and Results

Test Protocol

UC3_TC2_Test Run 1		
Testzeit	26.03.2026	
Testgebiet	THU (Ladesäule), virtuelles Netzgebiet	
Beteiligte User/Anlagen	THU: Ladesäulen & BEV	
	ZSW, Avat, MSU: OpenADR VTN bzw. VEN	
Versionsstand der wesentlichen Komponenten	Vor jedem Test-Durchlauf prüfen !	
	VTN und VEN	OpenADR 3.0.1
	Ladesäule Easee Home	FW: v340; OCPP 1.6 + proprietäres Backend
	Ladesäule ABL emH3 Twin	FW: 1.8p4; OCPP 1.6
	BEV Hyundai Ioniq 5 (Modell: NEAE.S5BLC.AEU)	NE_E.EUR.S5W_L.001.001.231102
	BEV Cupra Born	1701
Abweichung von Vorbedinungen	BHKW in Ludiwgsburg nicht live gesteuert, Umsetzung simulativ	
	Quartier mit Strom- und Wärmebedarfen simuliert	
Arbeitsschritte		
	1. Server starten	
	2. Webhooks setzen / abbonnieren	
	3. Verbindung zu Ladesäulen prüfen	
	4. Preise senden, Fahrpläne berechnen, Steuersignale an die Ladesäulen	
Auswertung		
Prüfer	Linda Roth	ZSW
Test-Ergebnisse	→ wenn als Testkriterium definiert	
	Definierte Prozesskette läuft durch (events, reports werden korrekt gesendet und empfangen)	[x] Bestanden
	Simulationszeit:	[ ]
	Ladesäulen regeln entsprechend Steuerungsbefehlen	[x] bestanden
Erwähnenswerte Vorkommnisse	→ wenn nicht als Test-Kriterium definiert	
	MSU: Easee Ladesäule hat Abregelbefehl erhalten und wird nicht geladen, Fehlersuche: bei MSU lange Queue und Verzögerung beim versenden des Operating State Event	
	Avat Genauigkeitsfehler: kleine Leistungszahlen wurden gerundet und Abregelung auf 0 wurde als 1 kW angezeigt --> gefixed	

Dynamic Grid Tariffs (ZSW)

For the test, the dynamic grid tariff time series shown below was created. It was generated manually and was intended to mimic realistic tariff structures as closely as possible. The time series was transmitted from ZSW to AVAT via OpenADR as a price_event.

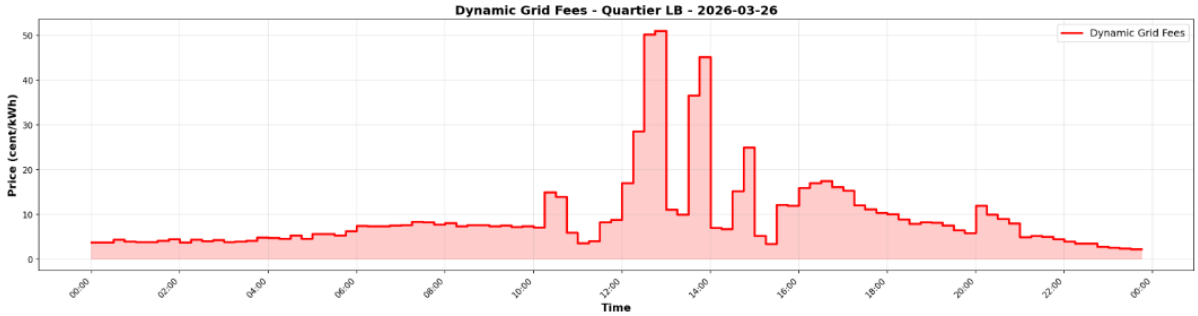


Figure: Dynamic grid tariff time series for the test day (ZSW).

Schedule Optimization (AVAT)

The dynamic grid tariffs received from ZSW and the day-ahead market price forecast were combined into a price time series and fed to the AVAT optimizer. The optimizer used this time series to calculate optimized charge schedules for the six charging stations in the district, as shown in the figure below.

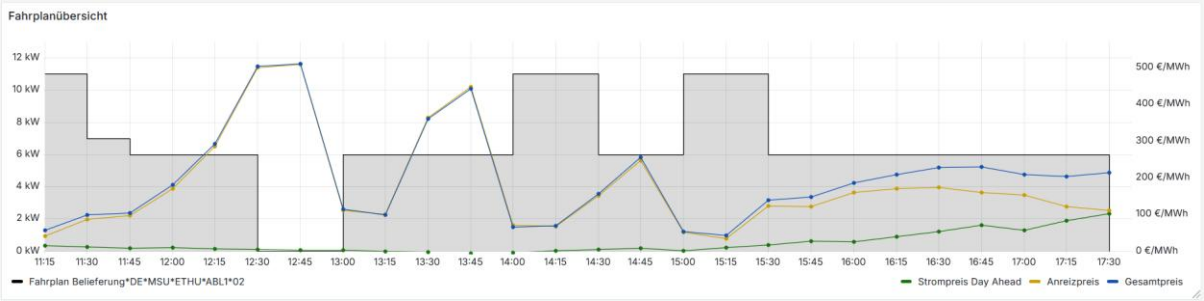


Figure: Charging station supply as a function of total price (Day Ahead Auction + Incentive Price).

Additionally, the planned CHP operation shifted relative to pure day-ahead optimization, so that the charging stations' electricity demand was covered during periods with lower overall costs (day-ahead price + incentive price). This demonstrates that grid tariff incentives effectively influence district optimization.

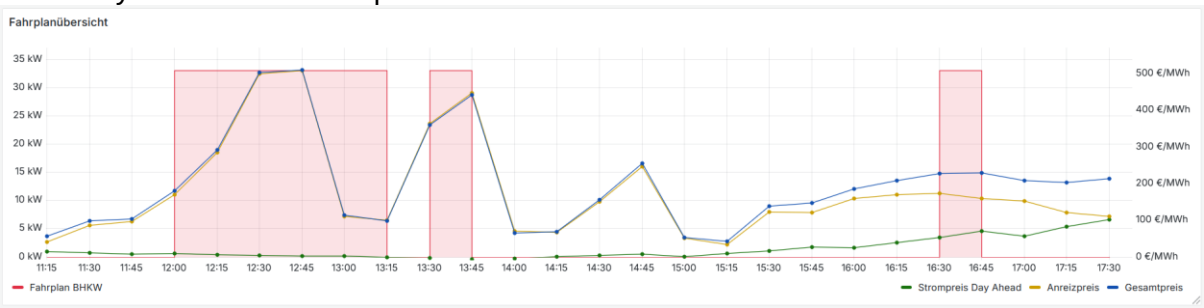


Figure: CHP operation and grid consumption as a function of total price.

To display the additional revenues from grid-serving operation, as required by Scenario C (Revenue Transparency), the AVAT optimization system software was extended accordingly and deployed at the utility partner's plant. These revenues are shown in the figure below.

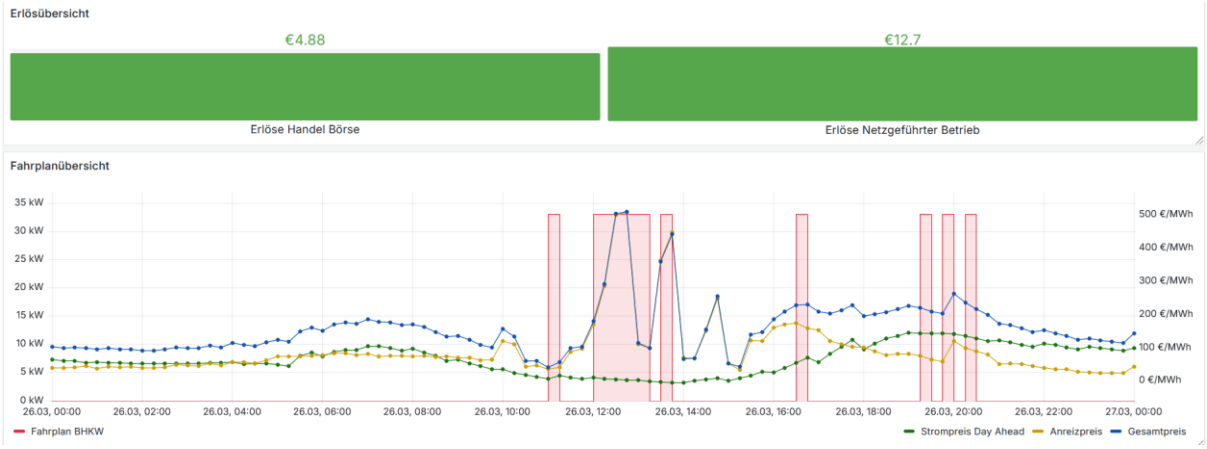


Figure: Additional revenues in the AVAT optimization system: Revenue overview (exchange trading revenues vs. grid-serving operation revenues) and CHP schedule overview with electricity, incentive and total price.

Charge Management (KSS)

During the test on 2026-03-26, the m8mit backend acted as a VEN (Virtual End Node) towards AVAT's VTN. KSS's OpenADR client received the charge setpoints calculated by AVAT as setpoint_events via the registered webhook. The received setpoints were automatically forwarded to the connected charging points and the charge power was adjusted accordingly. In parallel, usage_reports containing current metering values (charge power in kW, meter reading in kWh) and the operating state (OPERATING_STATE) were sent back to the VTN at 15-minute intervals. The following figure shows the charge power profile during the test: Initially, charging occurred at full power (~11 kW); after the setpoint event with a reduced setpoint was received, power dropped to approx. 4 kW. After the curtailment phase ended, full charge power was available again. The entire communication workflow, from webhook notification through setpoint adjustment to feedback reporting, ran fully automatically via the OpenADR 3.0 interface.

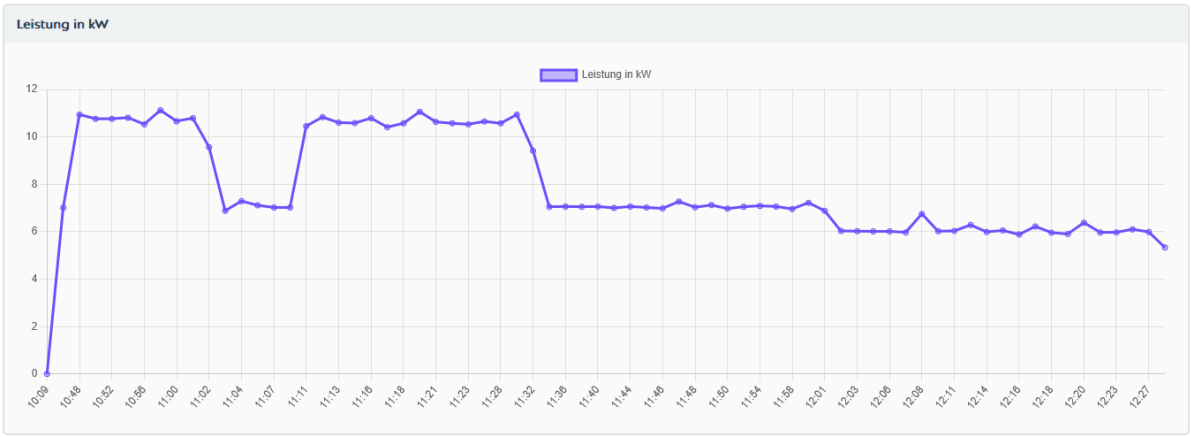


Figure: Charge power over time during the test – reduction from approx. 11 kW to approx. 4 kW after receipt of the setpoint event.

11. Collaboration with the OpenADR Alliance

A key component of the OpenADR implementation was the ongoing dialogue with the OpenADR Alliance. As part of the project, a complimentary membership was arranged for the project consortium, which provided the following benefits:

- Access to additional information and documents from the OpenADR Alliance
- Participation in OpenADR Alliance webinars
- Contribution to the further development of the OpenADR standard (application-specific conventions in the energy and grid operations context)
- Optional official certification of the OpenADR interface per project partner

Furthermore, planning was initiated for publishing the solution developed in the project as a reference implementation for other OpenADR members.

12. Summary and Outlook

12.1 Achieved Results

- Successful evaluation and selection of OpenADR 3.0 over OSCP as the communication standard
- Specification of a three-tier system architecture with clearly defined roles (VTN/VEN)
- Definition and implementation of two core scenarios (Scenario A: Price Signal Response, Scenario B: EV Charging Flexibility Management) with detailed OpenADR object specification
- Setup of the technical infrastructure with OpenADR servers (ZSW, AVAT) and client (KSS) based on the OpenADR 3.0.1 specification
- Successful round-trip test across the entire communication chain
- Consolidation of communication work packages for consistent standardization
- Membership in the OpenADR Alliance and planning of a reference publication

12.2 Open Items

- Integration into existing grid management systems requires further work
- Field test with real production data is being prepared
- Integration of master data into the OpenADR protocol is planned for future implementation

12.3 Outlook

The OpenADR-based communication architecture developed as part of the implementation forms a solid foundation for the grid-serving deployment of charging infrastructure. The results demonstrate that OpenADR 3.0 is suitable in principle for handling the communication between distribution system operators, district optimization systems, and charge point operators, although some application-specific adaptations were necessary (e.g., the use of Anchor Events for charging session notifications).

Looking ahead, the planned publication as a reference implementation through the OpenADR Alliance provides the opportunity to feed the insights gained in the project into the further development of the standard and to serve as a template for other market participants.