Knowledge and innovation center in the field of Smart Charging infrastructure in the Netherlands.

Founded in 2009 by the Dutch grid operators.

- Prognoses and outlooks.
- Smart charging innovation and implementation.
- EU largest testing site.
  - Interoperability
  - Smart Charging
  - Cyber security
  - PQ
- Tender support.
- Promoting open innovation & open protocols.

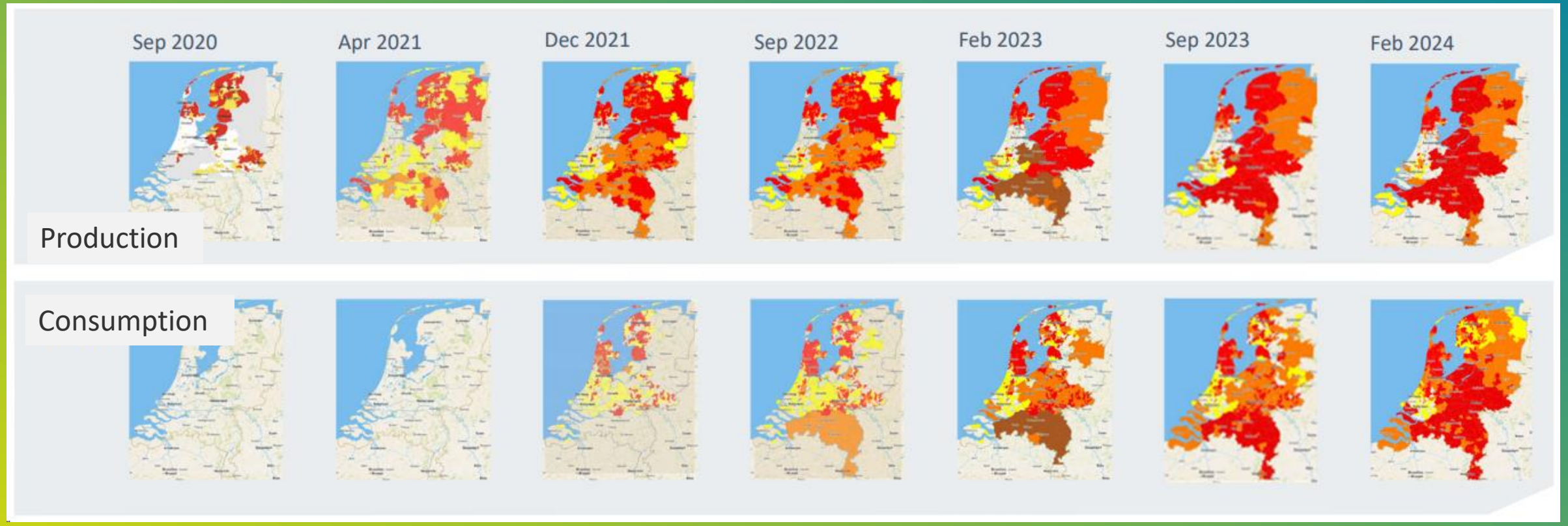
Arjan.wargers@elaad.nl

OpenADR
for Grid-Aware Charging

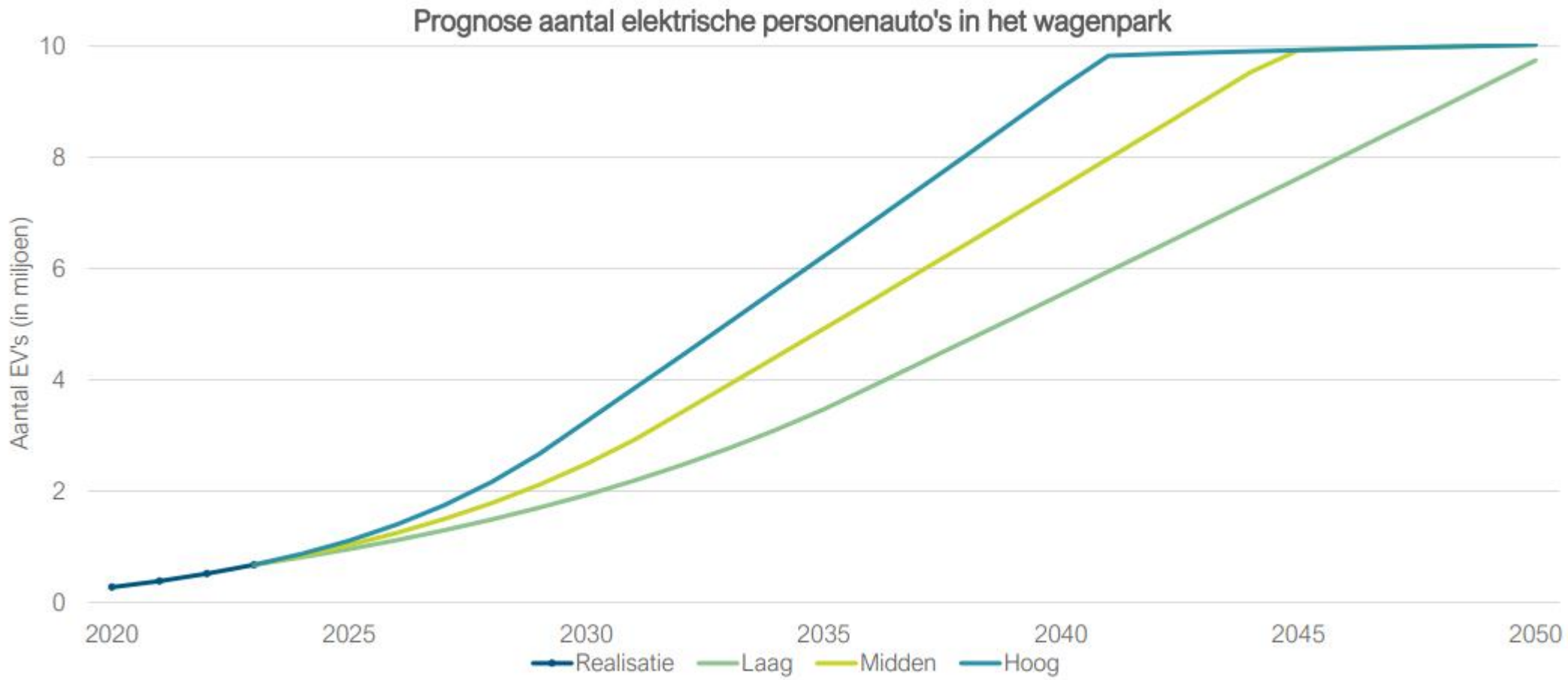
Agenda

- Introduction (Congestion, e-Mobility prognoses and solutions)
- DSO – CPO Interface
- OpenADR 3.0

Development of Congestion



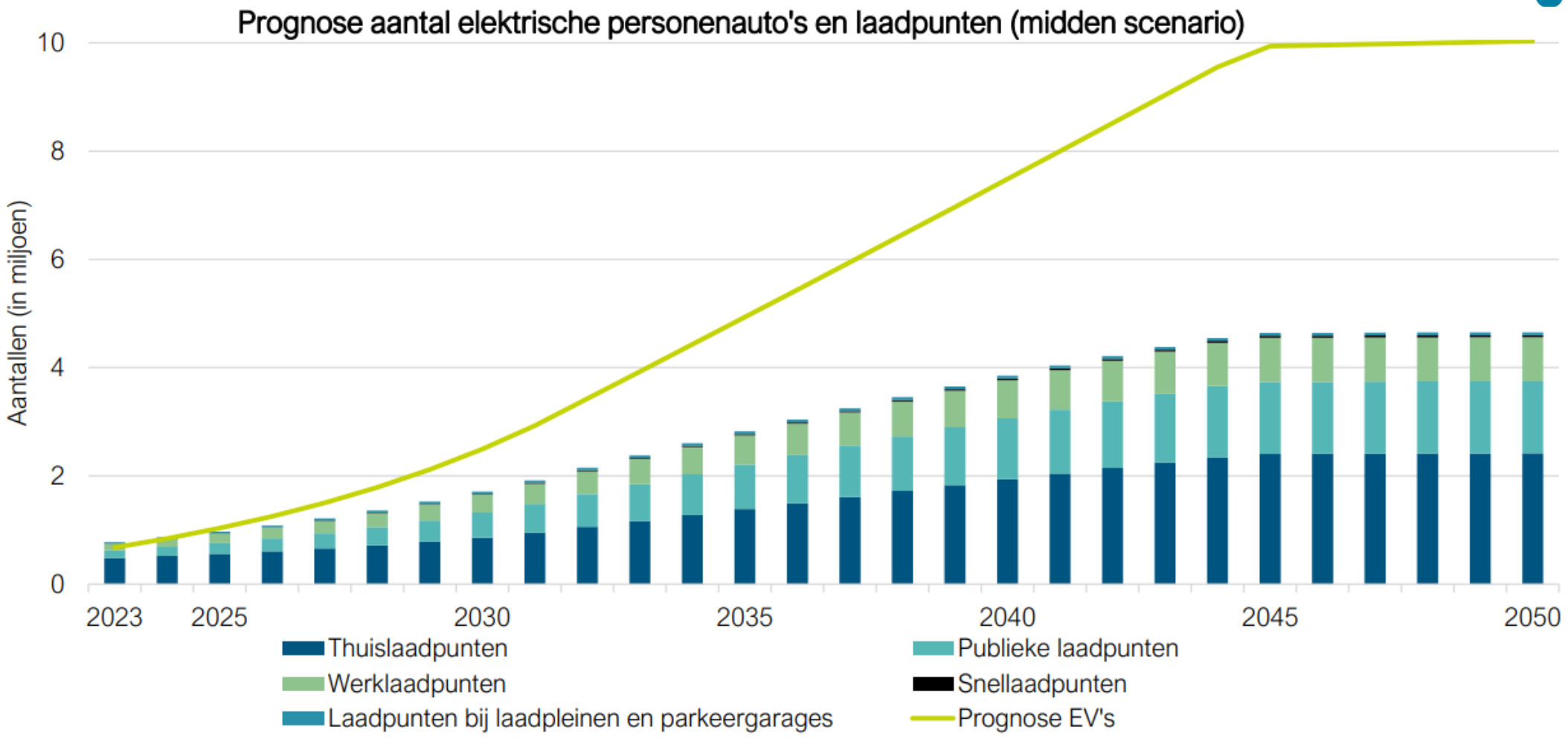
EV growth prognoses



Figuur 6: Prognoses aantal elektrische personenauto's in het wagenpark tot en met 2050

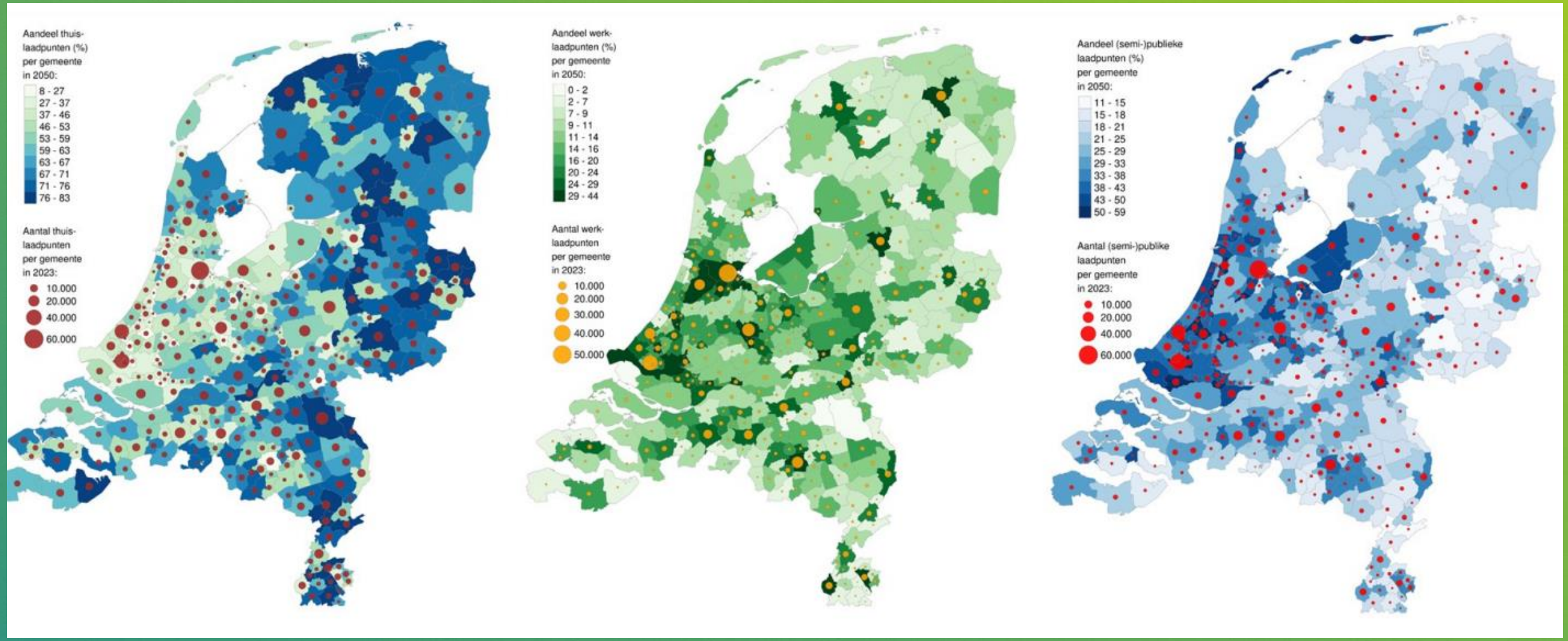
Source: ElaadNL outlook: Elektrificatie van personenauto's tot 2050

EV + EVSE growth prognoses



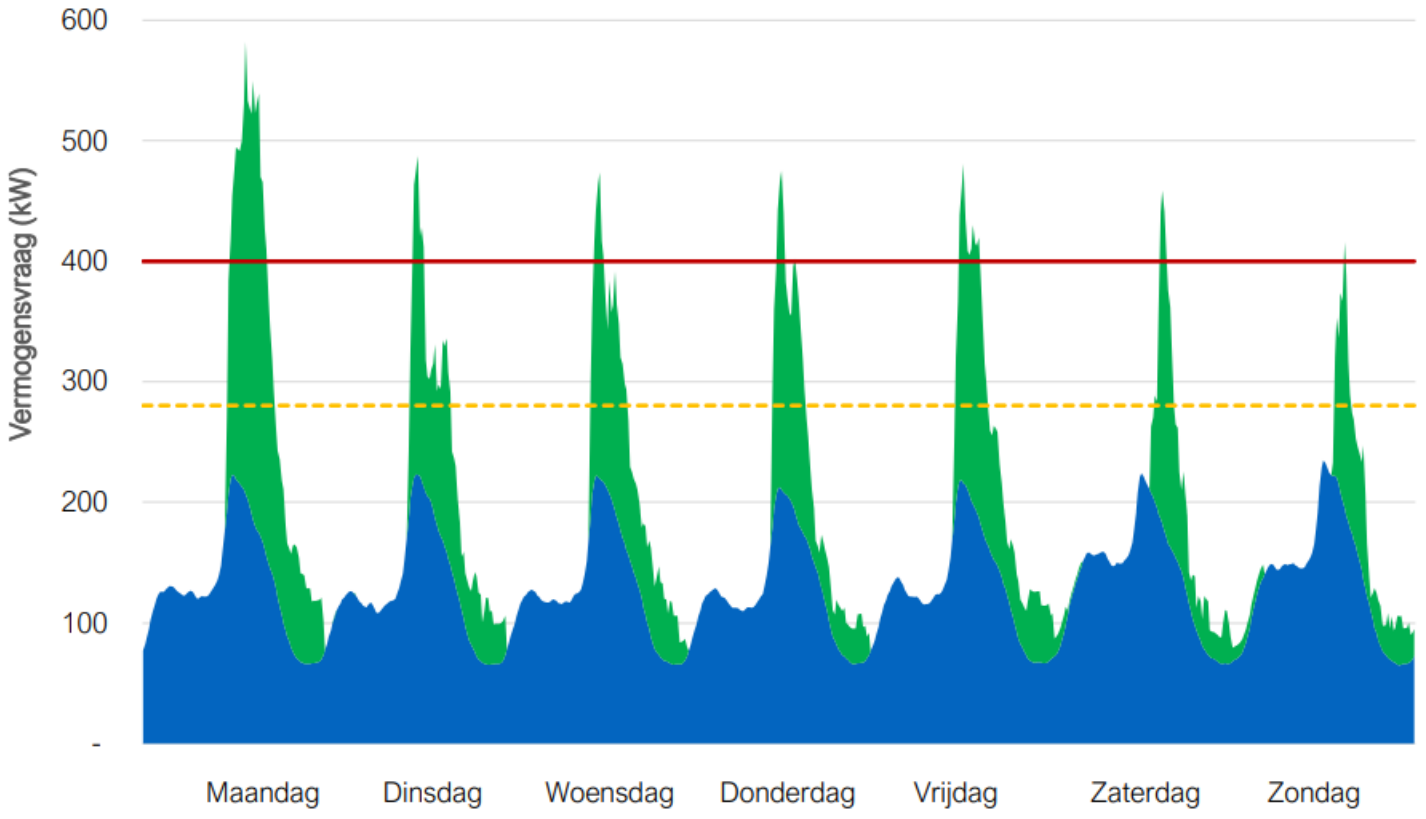
Source: ElaadNL outlook: Elektrificatie van personenauto's tot 2050

Regional spread EV adoption leads to EV hotspots



Source: ElaadNL outlook: Elektrificatie van personenauto's tot 2050

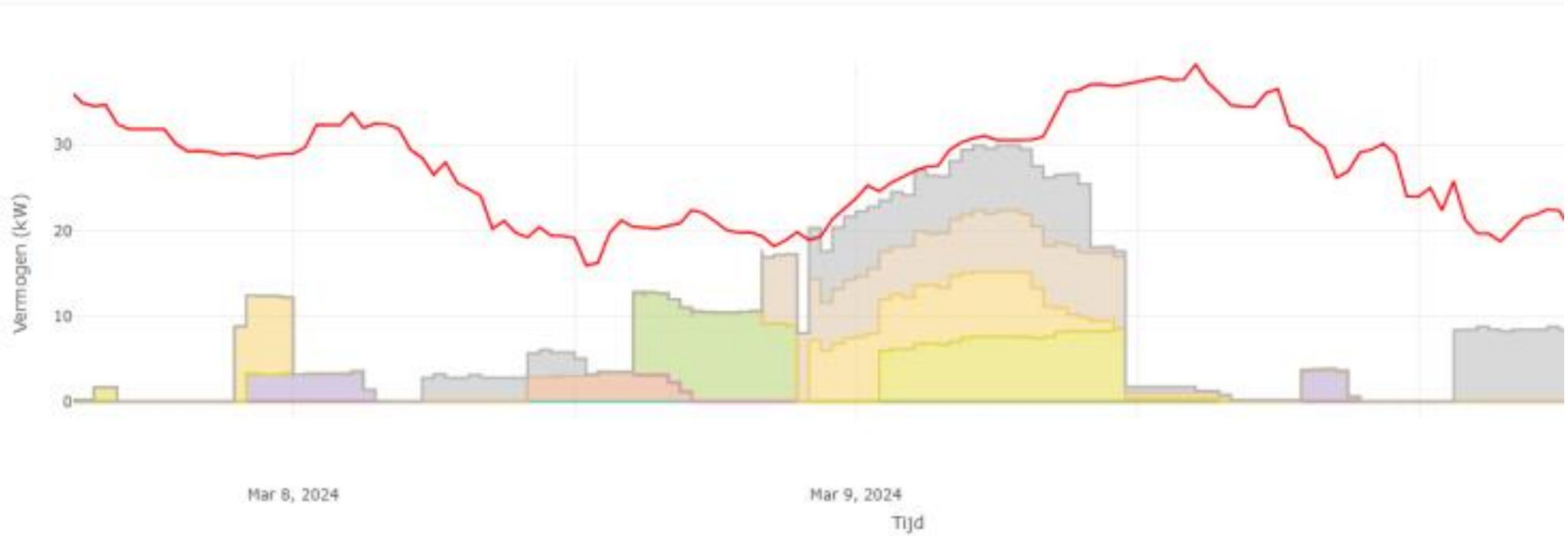
EV peak demand



Source: ElaadNL outlook: Elektrificatie van personenauto's tot 2050

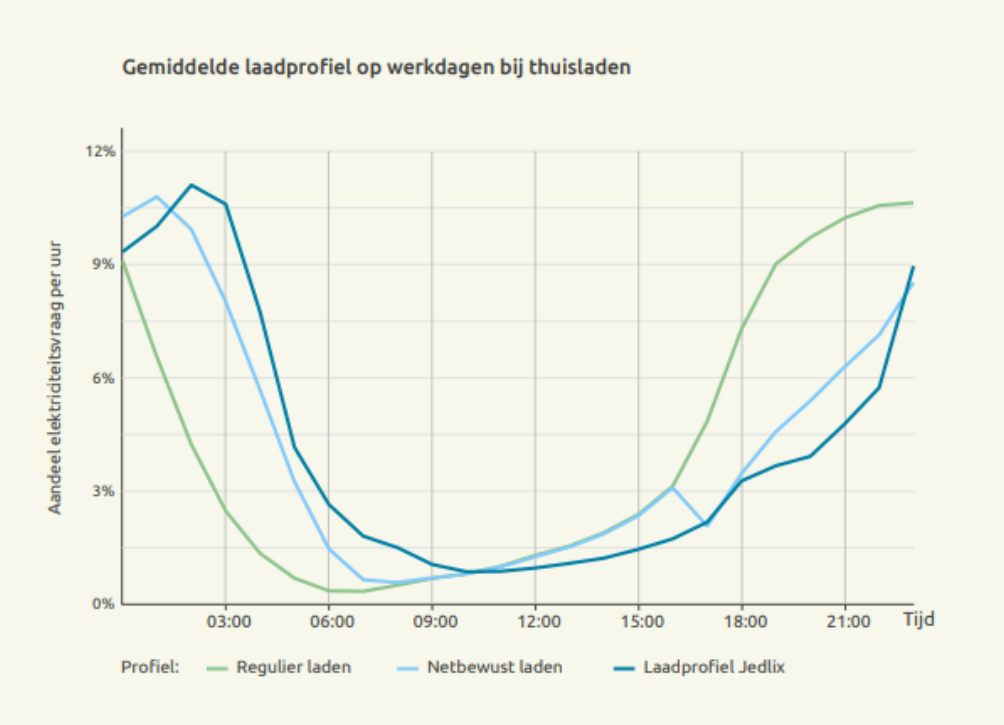
Smart charging pilots and PoCs

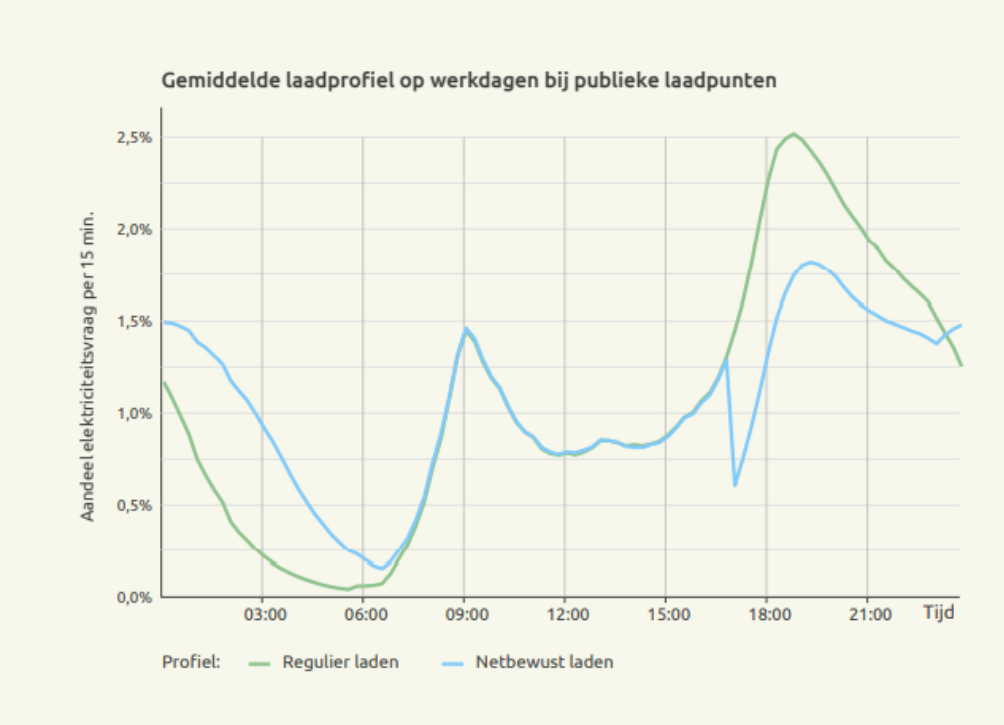
Grid aware charging



- Peak load reduction of 35%-49%
- Charged volume is between 2% and 4% lower

Smart charging profiles
Grid aware charging





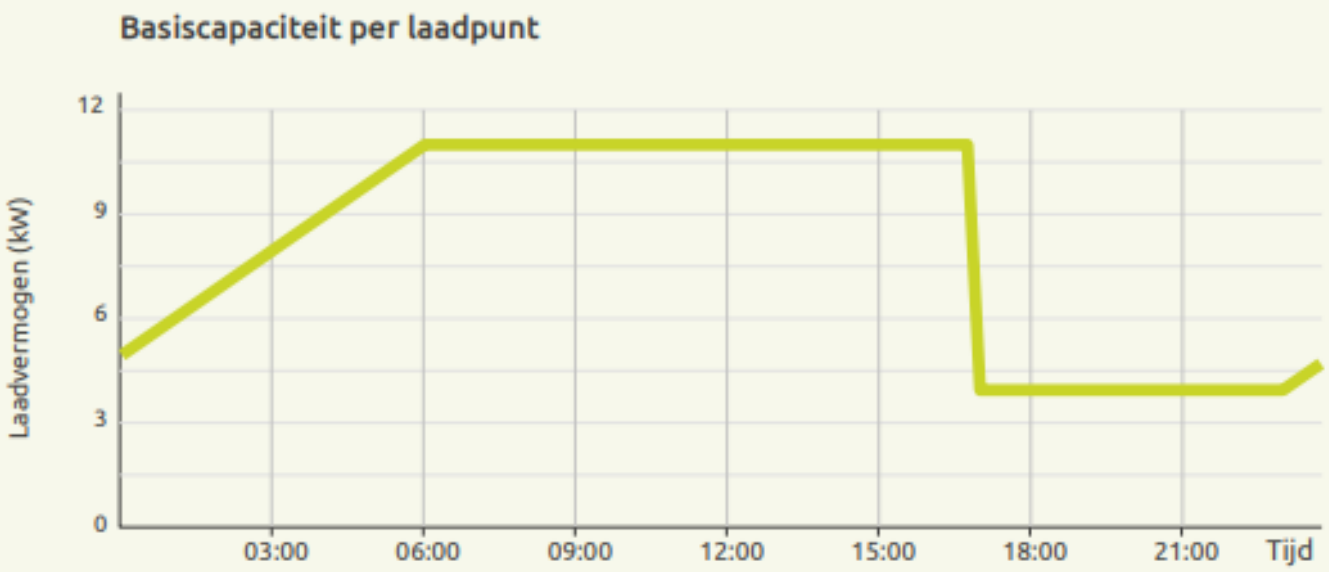
Application of grid aware charging:
Potential peak reduction of 44% for private chargers.
Potentail peak reduction of 28% for public charging.

Bron: ElaadNL Outlook Laadprofielen personenauto's

OpenADR applied

DSO-CPO Interface

Solution: Grid aware charging



Grid aware charging

In phases towards increasing automation and accuracy

Static profile (MVP)

Via grantor

- Capacity profile and application area determined based on raw measurement data.
- Capacity profile and application area is sent manually without linking systems.
- Update frequency is maximum 2x per year.

Automated static profile (NBL 1.0)

From DSO to CPO

- Capacity profile and application area determined where possible based on actual data (Dali boxes in combination with algorithm)
- Capacity profile and application area is sent automatically via a link between systems (e.g. OpenADR)
- Frequency of update is on a weekly/monthly basis.

Dynamic profile (NBL 2.0)

From DSO to CPO

- Data is read in real time and the capacity profile and application area are controlled day ahead on available grid capacity.
- Capacity profile and application area are sent day ahead via a system link (e.g. OpenADR)

Overview

The DSO – CPO interface is a interface between the Dutch Distribution System Operators (DSOs) and the Charge Point Operators (CPOs) active in the Netherlands.

Part of the National Action Plan on Charge Infrastructure

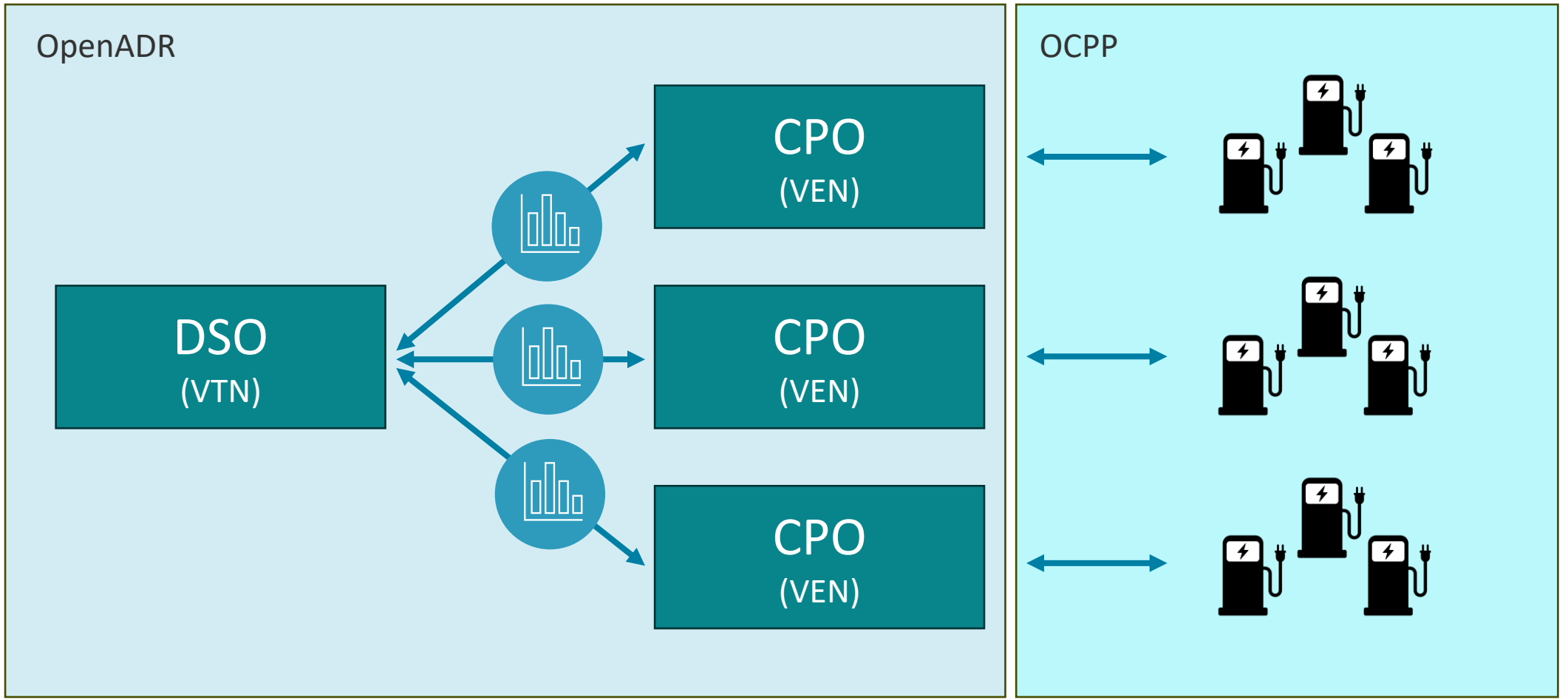
The standard that will be used for this is OpenADR 3.0. This standard is for communicating Demand Response signals.

As agreed between the DSOs and CPOs, this interface will be a polling based interface, where it is the CPO responsibility to fetch events that are published by the DSO.

Goal

- Sending grid connection limits to clusters of charging stations (grouping)
- Target each cluster based on the underlying EAN18 identifiers of the charging stations
- Clusters defined by DSO, based on grid topology
- Cluster information needs to be in sync between DSO and CPO

Architecture



Message example

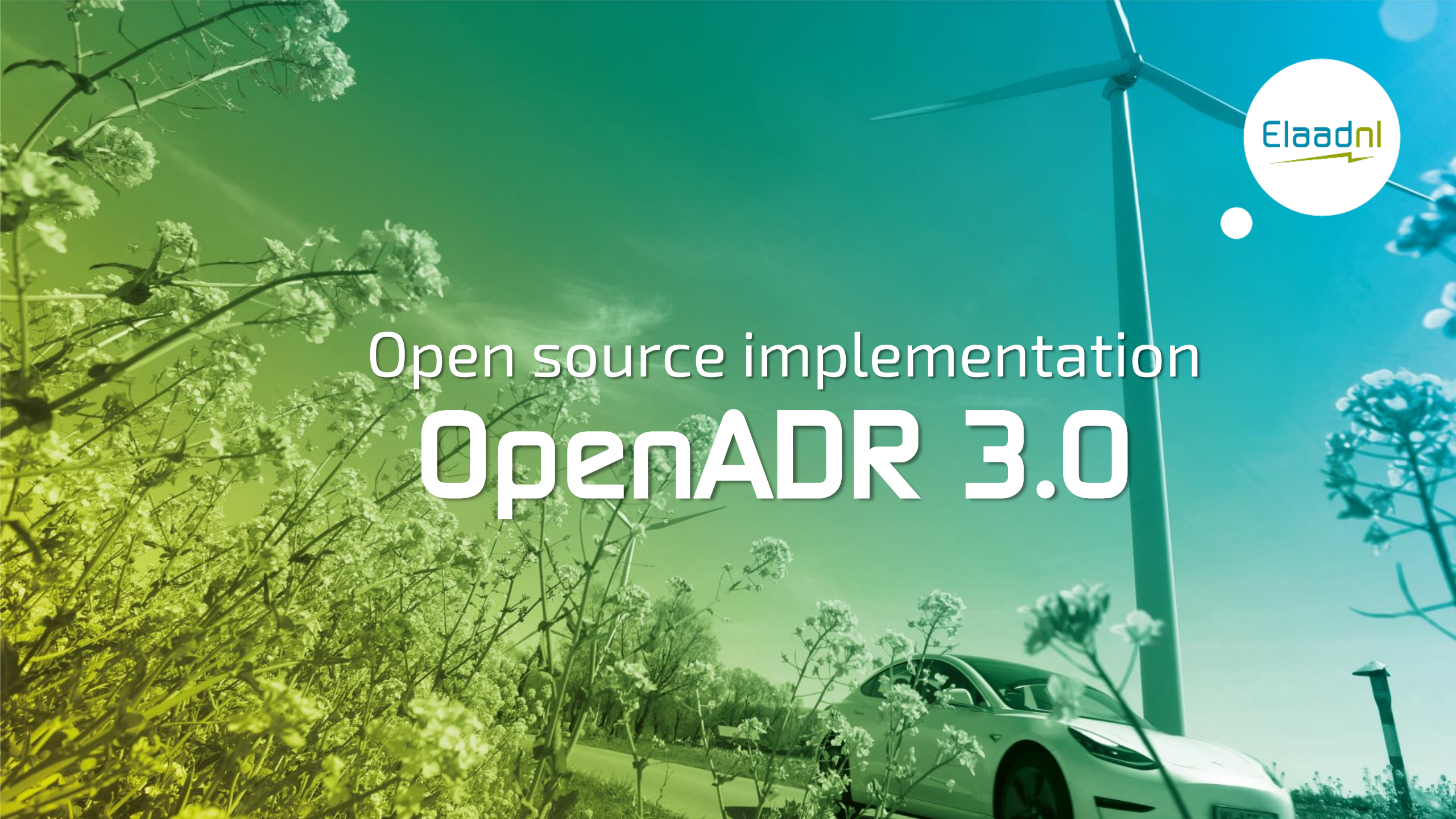

```
{
    "id":"f81d4fae-7dec11d0-a765-00a0c91e6bf6",
    "createdDateTime":"2024-05-03T09:30:00.000Z",
    "modificationDateTime":"2024-05-03T09:30:00.000Z",
    "objectType":"EVENT",
    "programID":"DSO_CPO_INTERFACE_NL",
    "eventName":"GAC signal 2024-10-01",
    "priority":200,
    "targets":[
        {
            "type":"GROUP",
            "values":[
                "LP_CLUSTER-001"
            ]
        },
        {
            "type":"POWER_SERVICE_LOCATION",
            "values":[
                "EAN87654321123456",
                "EAN123123123123123"
            ]
        }
    ],
    "reportDescriptors":null,
    "payloadDescriptors":[
        {
            "payloadType":" IMPORT_CAPACITY_LIMIT",
            "units":"KW"
        }
```

Message example


```
"payloadDescriptors":[
    "payloadType":"IMPORT_CAPACITY_LIMIT",
    "units":"KW"
  }
],
"intervalPeriod":{
  "start":"2024-11-19T00:00:00.000Z",
  "duration":"PT24H"
},
"intervals":[
  {
    "id":0,
    "intervalPeriod":{
      "start":"2024-11-19T00:00:00.000Z",
      "duration":"PT15M"
    },
    "payloads":[
      {
        "type":"IMPORT_CAPACITY_LIMIT",
        "values":[
          40
        ]
      }
    ]
  },
  {
    "id":1,
    "intervalPeriod":{
      "start":"2024-11-19T00:15:00.000Z",
      "duration":"PT15M"
    },
    "payloads":[
      {
```

Message example


```
    "id":1,
    "intervalPeriod":{
        "start":"2024-11-19T00:15:00.000Z",
        "duration":"PT15M"
    },
    "payloads":[
        {
            "type":"IMPORT_CAPACITY_LIMIT",
            "values":[
                10
            ]
        }
    ]
},
"...",
{
    "id":23,
    "intervalPeriod":{
        "start":"2024-11-19T23:45:00.000Z",
        "duration":"PT15M"
    },
    "payloads":[
        {
            "type":"IMPORT_CAPACITY_LIMIT",
            "values":[
                35
            ]
        }
    ]
}
```

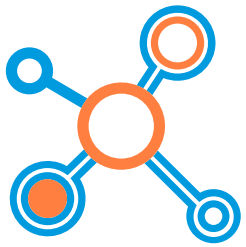


Open source implementation

OpenADR 3.0

Background

- OpenLEADR OpenADR 2.0b implementation
- Created in 2020 by ElaadNL
- Open source
- Written in Python3
- Part of Linux Foundation (LF) Energy
- ~2K downloads/month



LF ENERGY openLEADR

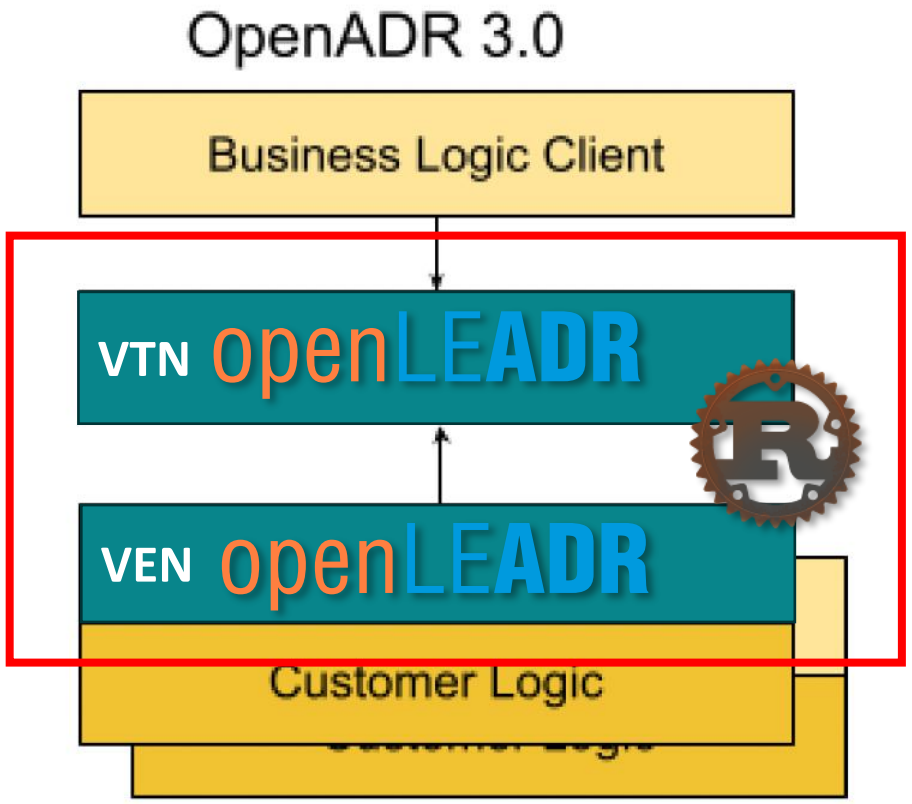
Next step

- OpenADR open source implementations needed
- New approach OpenADR 3.0 based on OpenAPI specification allows for more integration flexibility
- Stable VTN/VEN implementations that can connect to custom business logic
- Enable fast adoption

OpenLEADR for 3.0

- OpenLEADR OpenADR 3.0 implementation
- Created in 2024 in collaboration between ElaadNL and Tweede Golf
- Open source
- Written in Rust
- Part of Linux Foundation (LF) Energy

OpenLEADR for 3.0



- Robust, memory-safe implementation of both VTN and VEN
- Not yet fully feature complete but a great start
- Built and maintained by Tweede Golf in collaboration with ElaadNL

GitHub OpenLEADR 2.0

OpenLEADR is a Python 3 module that provides a convenient interface to OpenADR systems. It contains an OpenADR Client that you can use to talk to other OpenADR systems, and it contains an OpenADR Server (VTN) with convenient integration possibilities.

It currently implements the OpenADR 2.0b specification.

https://github.com/OpenLEADR/openleadr-python

GitHub OpenLEADR 3.0

OpenADR 3.0 in Rust

This repository contains an OpenADR 3.0 client (VEN) library and a server (VTN) implementation, both written in Rust. OpenADR is a protocol for automated demand-response in electricity grids, like dynamic pricing or load shedding. The OpenADR alliance is responsible for the standard, which can be downloaded free of charge. This implementation is still work-in-progress, and we aim for a first stable release in December 2024.

Thanks to our sponsors Elaad and Tweede golf for making this work possible.

https://github.com/OpenLEADR/openleadr-rs
https://trifectatech.org/initiatives/automated-demand-response/


Elaadnl

Q & A

RESEARCHING AN
D TESTING SMART
AND SUSTAINABL
E CHARGING