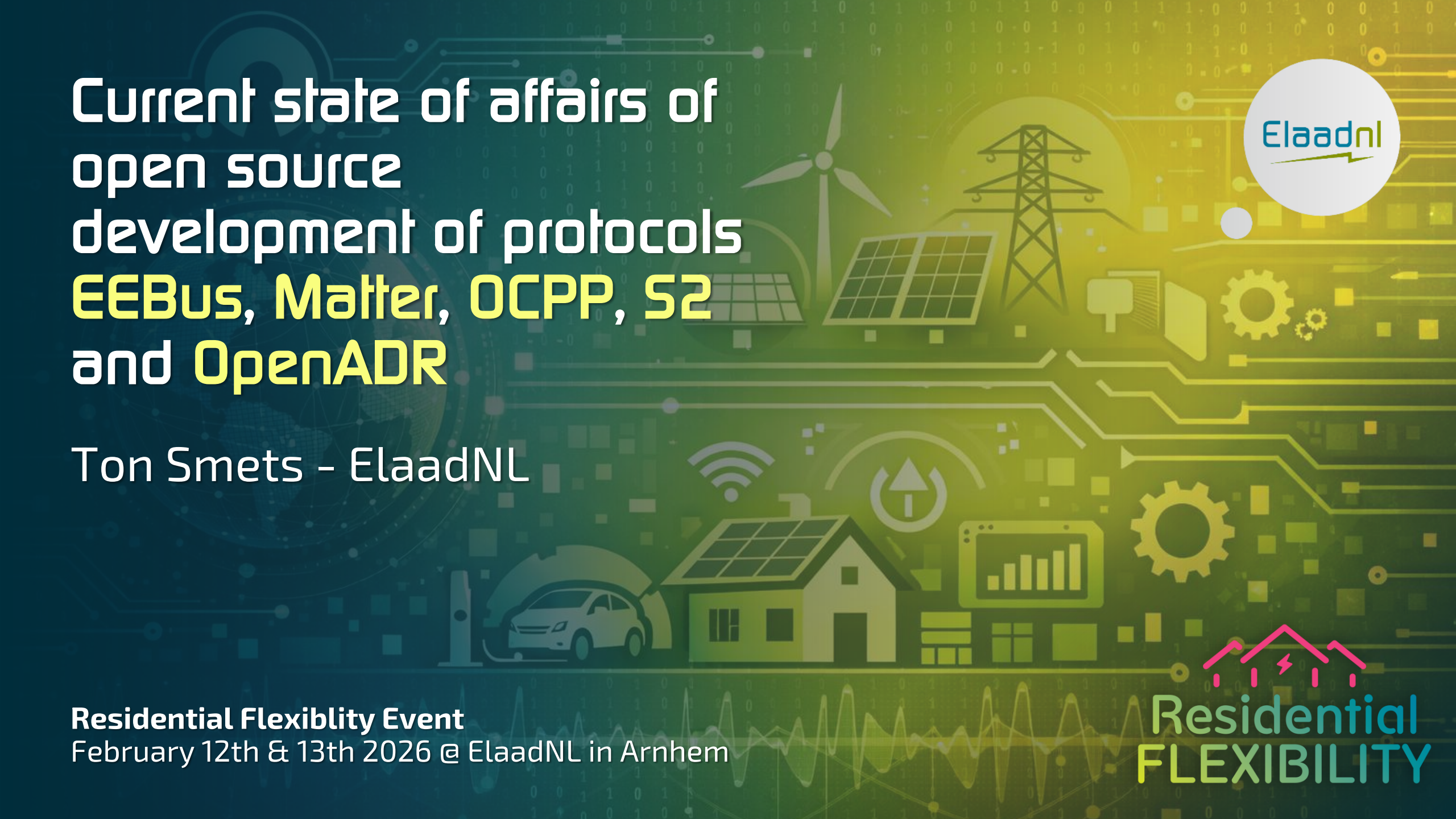
Current state of affairs of open source development of protocols EEBus, Matter, OCPP, S2 and OpenADR

Ton Smets - ElaadNL



Residential Flexiblity Event
February 12th & 13th 2026 @ ElaadNL in Arnhem

Residential FLEXIBILITY

About me

Ton Smets

- Software engineering background
- 8+ years in the energy domain
- Interoperability

ton.smets@elaad.nl

Agenda

- Why the Residential Flexibility project exists
- What problem we are solving
- What we asked the market to build
- Where we are today
- What this means going forward

Why this project exists

Residential FLEXIBILITY

The urgency

- Net congestion increasing (LS-net issues)
- “Saldering” ends in 2027
- Growth in EVs, heat pumps, batteries, PV inverters
- Dynamic tariffs increasing

“Residential flexibility is no longer optional,
it is structurally required”

What is residential flexibility?

flexibility

noun [U]

UK /ˌflek.səˈbɪl.ə.ti/ **US** /ˌflek.səˈbɪl.ə.ţi/

“the ability to change or be changed easily according to the situation”

What is residential flexibility?

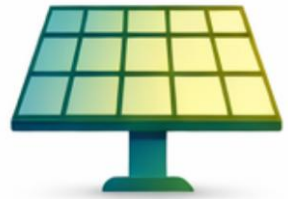
Flexibility = the ability to

- Limit peak demand
- Shift energy use in time
- Store surplus PV energy
- Modulate power
- Respond to grid signals

Main flexible assets in homes:

"THE BIG 4"

- Heat pump
- EV charger (EVSE)
- PV inverter
- Home battery



Residential FLEXIBILITY

The hard truth: No interoperability

Market research findings:

- 26 protocols and communication methods identified
- Modbus most used inside homes
- Custom APIs mostly inhomes (cloud-based)
- No interoperability

We have connectivity, but no shared language

The market research

"**Residentiële Flexibiliteit**" conducted by **Powerfolio** on behalf of **ElaadNL** and **Flexiblepower Alliance Network (FAN)** – *February 2025*

- Powerfolio identified 26 different communication protocols and methods currently used in the Dutch residential flexibility landscape
- The most widely used protocol inside homes is Modbus (RTU/TCP), but implementations differ per vendor and project, meaning there is connectivity without a shared semantic language.
- Market parties agree that interoperability is essential to scale residential flexibility, but they are hesitant to commit to a single standard without coordinated action and architectural clarity.

https://nl.flexible-energy.eu/nieuws-events/rapport-residentiele-flexibiliteit-nu-beschikbaar/

Architectures

Two interoperability challenges:

- HEMS → Devices inside home
- Grid / Aggregator → HEMS

Hybrid future:

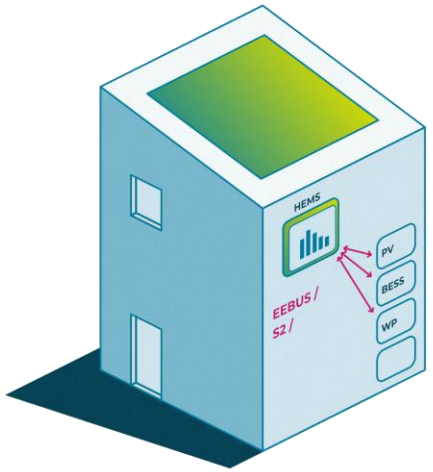
- Local HEMS
- Cloud HEMS
- OEM cloud integrations

Different routes

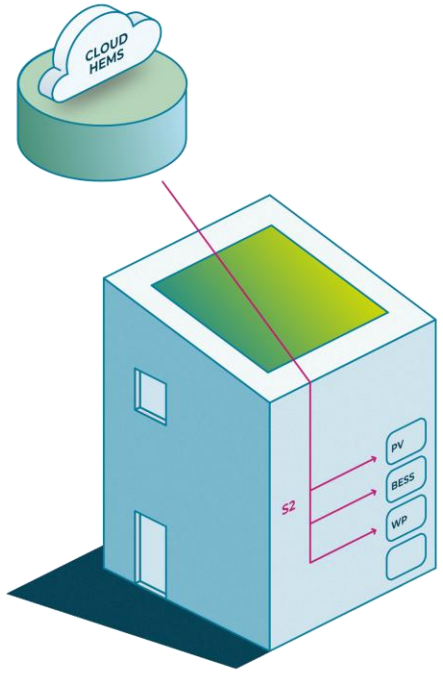
- Countries around us prescribe how signals should be communicated to HEMS or directly to assets
  - Example: Germany §14a legislation where a DSO is able to control/limit assets via a physical “Steuerbox” connected to the smart meter
- Many different options possible
  - Direct control (example: DSO → heatpump)
  - HEMS control (example: DSO → HEMS)
  - Aggregated control (example: DSO → Aggregator → HEMS)
- Possibilities for future markets

Different routes

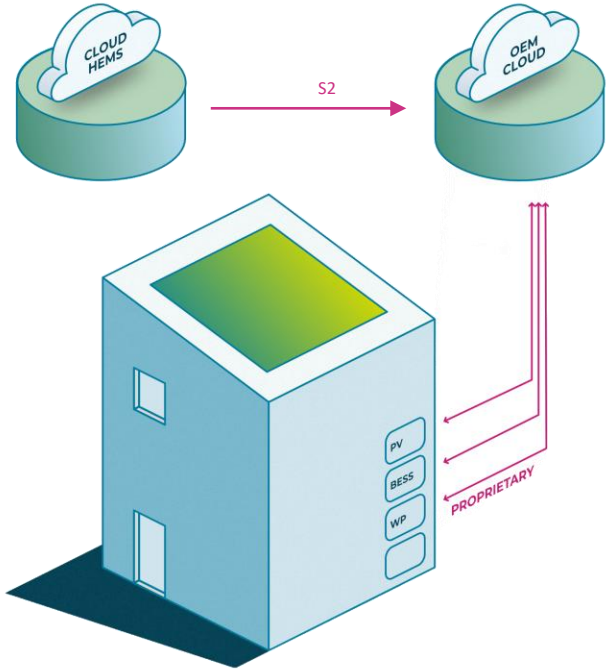
Local HEMS



Cloud HEMS



Cloud HEMS
+ Device OEM Cloud



From research to action

Residential FLEXIBILITY

What the market told us

Market feedback:

- Need interoperability
- Avoid vendor lock-in
- Prefer limited number of standards
- Align internationally
- Avoid overly complex frameworks

But:

- No clear market leader
- No party wants to move first

Someone needs to take a first step

Already available solutions

- A small amount open source software implementations for some protocols were available already
- Our research showed a variety flaws in most of them
  - Feature incompleteness
  - No maintainer for the open source repository
  - Non-permissive licensing
  - Limited tests and test results
  - Less popular programming languages used

So we launched an RFP

Objective

Develop open-source standardized connectors for:

- S2
- Matter
- EEBUS
- Modbus bridge
- OCPP

Requirements

- Open Source
- Permissive licensing (Apache 2.0)
- Easily accessible (Public GitHub)
- Quality
- Cybersecure
- Written in popular programming languages
- Interoperability testing (like today, tomorrow and later this year)

Open source experience

OpenLEADR (Python)

OpenADR 2.0 implementation

openleadr-rs (Rust)

OpenADR 3.0 implementation

Already deployed in Dutch Grid Aware Charging (live)

Used for:

- Grid capacity limiting
- Real EVs responding to real DSO signals

Difference in protocols

Home-Level Protocols

- S2
- EEBUS
- Matter
- OCPP

Grid-Level Protocol

- OpenADR (OpenLEADR)

Clear separation

- Grid signal transport
- Home device coordination

RFP ideology

- A HEMS should offer multiple options for protocols to connect with, ideally all 4 from the RFP list
- A device OEM implements at least 1 protocol from the RFP list
- A user should be able to buy for example a home battery and a HEMS that have the same protocol printed on the box. Installation should be plug-and-play

What we asked the market to build

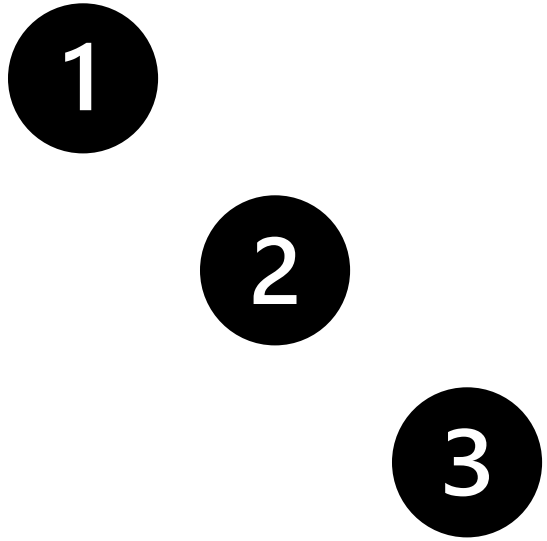
Residential FLEXIBILITY

The 5 work packages

WP	Protocol	Purpose
WP1	S2	Energy flexibility
WP2	Matter	Energy flexibility
WP3	EEBUS	Energy flexibility
WP4	Modbus	Support for installed base
WP5	OCPP	EV charging control integration

The 3 core usecases

1. Peak Grid Demand Limiting
   - DSO capacity profiles
   - OpenADR at grid level
   - S2 / EEBUS / Matter inside home
2. Dynamic Tariff Optimization
3. Self-Consumption Optimization



Current state with 3 selected parties

Residential FLEXIBILITY

RFP outcome

Coverage for most work packages

- WP1 – S2 ☑
- WP2 – Matter ☑
- WP3 – EEBUS ☑
- WP4 – Modbus bridge ☑
- WP5 – OCPP ☒
  - WP 5A – HEMS OCPP connector ☑
  - ~~WP 5B – OCPP Proxy~~ (intentionally not yet filled) ☒

Why WP5 is still partly open

OCPP was designed for EVSE <-> CPO backend

In the home, new challenge:

- HEMS wants to influence charging
- CPO must retain control and billing functions
- Profiles must not conflict

Therefore we are working together with Open Charge Alliance (OCA) to

- Define the correct architecture
- Avoid fragmented interpretations

WP5 is paused strategically, not abandoned

Technologies

Embedded side (the flexible assets)

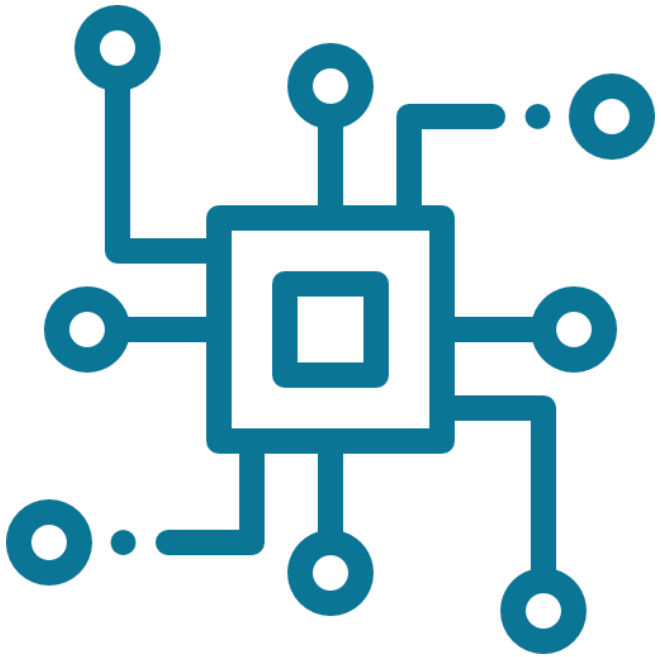
- C/C++

HEMS / cloud

- Python
- Hybrid solutions

OpenADR

- Rust
- Python



Where we are now

S2 / Matter / EEBUS

- Active development
- GitHub repositories running (in private for now)

Modbus

- Bridge architecture under implementation with focus on interchangable configuration files

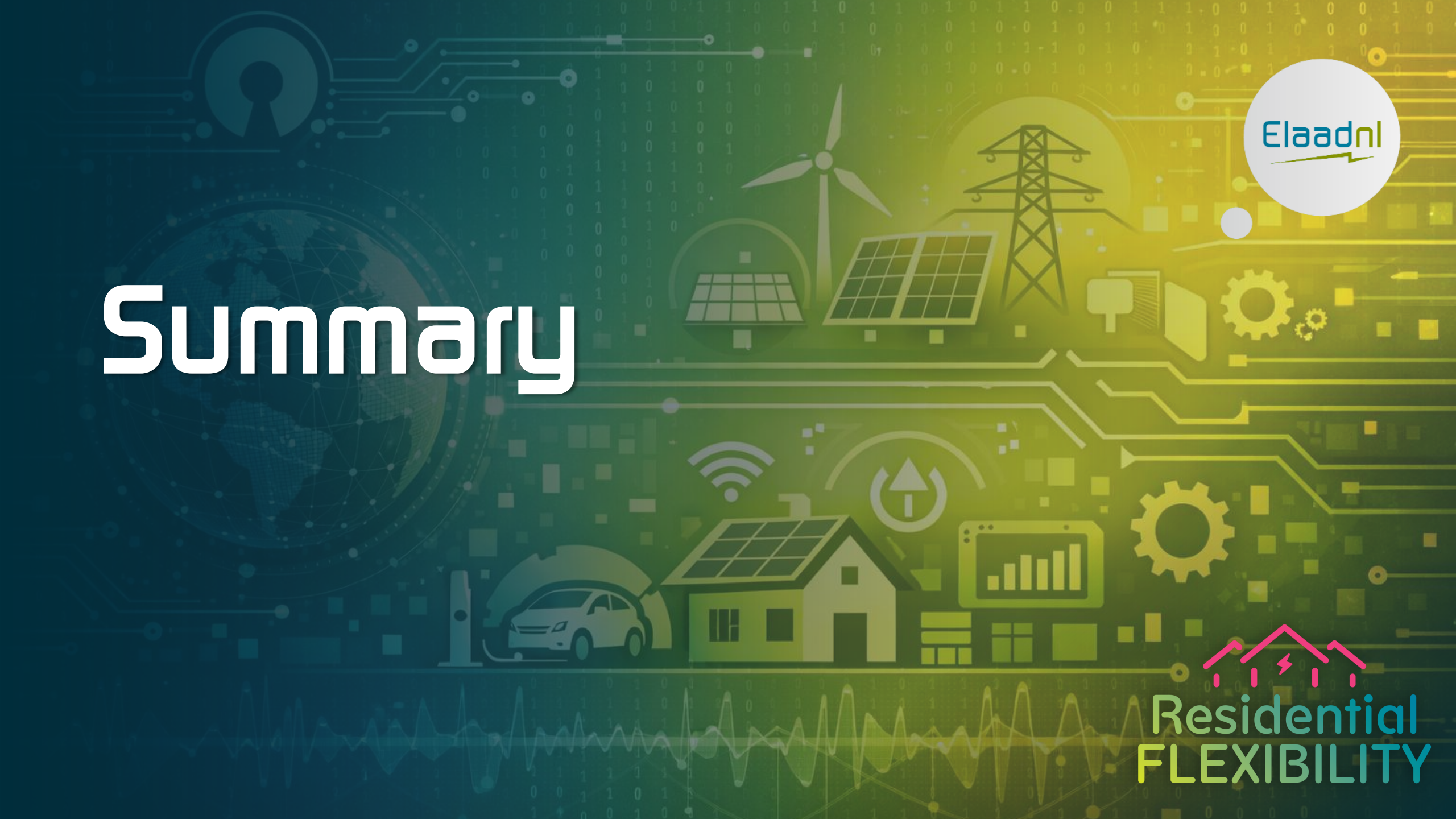
OCPP

- Alignment phase with OCA ongoing

OpenADR

- Python for OpenADR 2.0 stable
- Rust for OpenADR 3.0 still evolving (active development)
- Live Grid Aware Charging

Summary



Residential FLEXIBILITY

Why this is important

Without this effort:

- Continued API fragmentation
- Vendor lock-in
- Conflicting control models
- Slower growth & scaling of flex ecosystems

With this effort:

- Open reference implementations
- Reduced integration cost & time
- Foundation for (inter)national interoperability

Next phase

- Integration testing at ElaadNL Test Lab
- Interoperability demonstrations
- OCPP architectural decision in collaboration with OCA
- Preparation for broader market adoption
- Community building
- Expansion of Grid Aware Charging to support new OpenADR 3.x versions

Elaadnl
Residential
FLEXIBILITY