36th International Electric Vehicle Symposium and Exhibition (EVS36)
Sacramento, California, June 11-14, 2023

A Smart Charging System to Efficiently Run EV Fleets

Tobias Fleck[1], Sascha Gohlke[2], Zoltan Nochta[3]
[1]*SAP Deutschland, tobias.fleck@sap.com*
[2]*Karlsruhe University of Applied Sciences, sascha.gohlke@h-ka.de*
[3]*Karlsruhe University of Applied Sciences (corresponding author), zoltan.nochta@h-ka.de*

Executive Summary

Smart charging is a means of monitoring and actively controlling EV chargers to optimize the distribution and consumption of energy, with a focus on peak-load avoidance. In our paper, we discuss requirements on smart charging in the specific context of enterprise fleets. We consider smart charging as a component of a more comprehensive system architecture and deal with problems related to the integration with other functional modules and data sources. We present a practical software implementation of smart charging, which was realized as an extension of the open-source charge-point management solution ”Open E-Mobility”[1], and share evaluation results collected in a real-world deployment scenario.

Keywords: fleet, smart charging, infrastructure, ICT, load management

1 Introduction

In the past decade, the global market share of electric vehicles (EV) has been growing rapidly. A significant proportion of EVs of all types belong to enterprise fleets and are used for commercial purposes or as company cars by employees. For example, in Germany 58% of all electric cars sold in 2021 were registered to companies [2]. Companies are increasingly using their EV fleets for business-related, sometimes even for mission-critical purposes, as EVs prove to be more and more reliable. To ensure high operational readiness of EVs and reduce dependency on publicly accessible charging stations, many companies build and operate their own EV charging infrastructures (CI) on their premises. Those facilities are also often used by employees to charge privately owned EVs at work. Establishing and operating a CI poses a number of economic challenges to a company, including high capital and operating costs (TCO), volatile and less predictable utilization (during and outside business hours), complex tax regulations, etc. [1, 2, 3]. In addition, businesses must take several technical boundaries into consideration, such as missing or insufficient cabling at parking areas, grid power limitations, bad network connectivity, etc.

A properly designed software system can help enterprises master many of the operational challenges during the entire life-cycle of charging stations and related other assets. A crucial task thereby is to optimize the distribution of available, in many cases limited amount of power among multiple, often heterogeneous EVs and chargers in a safe and cost-efficient manner.

In this contribution we present a Smart Charging System (SCS), which is a software system mainly designed to serve the needs of companies that operate EV fleets and have one or more sites equipped with charging stations. Technically, the SCS is part of the open-source system ”Open E-Mobility”, which is used to manage thousands of charging stations at different locations meanwhile. It can be deployed and operated as an on-premise system or as a containerized cloud solution and communicate with several other systems via the provided interfaces.

[1]https://github.com/sap-labs-france/ev-server
[2]https://www.kba.de/DE/Statistik/Produktkatalog/produkte/Fahrzeuge/fz28/fz28_gentab.html?nn=3547466

The remainder of the paper is organized as follows: In Section 2, we present the architecture of the SCS, summarize the main functional and non-functional requirements, explain details of the core EV charging algorithm and show its integration with various data sources in the current implementation. Section 3 describes major evolutionary steps of a three-year continuous development and evaluation effort that took place at SAP Labs France in Mougins, France. Section 4 revisits publications dealing with related aspects of EV charging. Finally, directions for our future work are outlined in Section 5.

2 System Design

The high-level architecture of SCS contains four main functional components as shown in Figure 1. The main task of the component *Smart Charging Core* is to calculate and dynamically adjust the distribution of available power among the active charging sessions in the given CI (see also in Section 2.2). The *Data Manager* stores permanent data, such as the system configuration and master data about the capabilities of the installed charging stations. It also maintains temporary information needed to carry out calculations, for example. As part of "Open E-Mobility", the SCS interacts with other components of the entire charge-point management system. For example, it logs relevant events using the *Logging* interface and provides information about the status of active charging sessions for EV drivers via the *Mobile App* as well as for the operator of the CI via *Browser/Portal*. The SCS can communicate and exchange data with further external systems, including Energy Management System (*EMS*), Enterprise Resource Planning (*ERP*) or EV vendors' *Vehicle Backend*, if they are available and made accessible within the CI-owner's IT-environment. These systems are mainly used by the SCS as data sources for ongoing calculations related to load management. The required connections to these systems and to the charging stations on site, incl. protocol- and API-specific messaging, are handled by the *Communication Manager*. The component named *Integration Layer* is mainly responsible for collecting the required data from the different connected sources (in a synchronous or asynchronous way) and for the preparation of the gathered data for further processing by the core component (see details in Section 2.3).

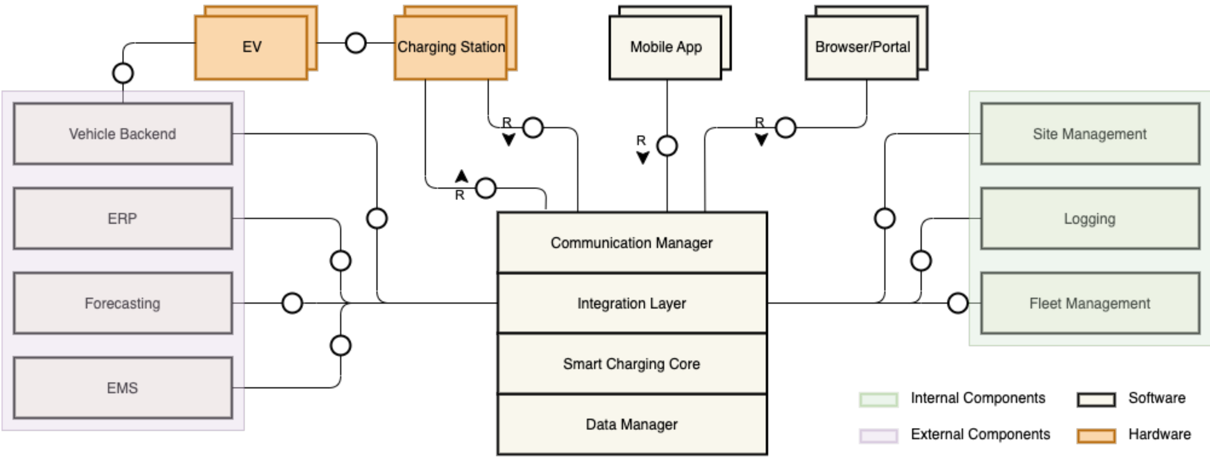

Figure 1: High-level architecture diagram of the Smart Charging System (SCS)

The deployment of the four main components is sufficient to operate the SCS with basic functionality. In this case, the *Smart Charging Core* works only with predefined configuration values, such as fixed safety limits for power consumption, and does not take into account dynamic information, such as instantaneous solar power generation. All additional internal and external components can be added optionally and independently from each other to adapt the system to specific requirements in the given scenario.

2.1 Functional and Non-Functional Requirements

In designing and implementing the SCS, we focused particularly on the following capabilities as important requirements for the system:

- **Infrastructure protection:** During the simultaneous charging of EVs, huge demand peaks can occur damaging the infrastructure or even leading to outages. The SCS must deal with several related thresholds at the same time, such as the mains connection power of the site, limitations of the local electrical infrastructure according to fuse hierarchies, capacity of individual power lines and transformers, etc. In addition, it is important to communicate with the local EMS, if deployed on site, to quickly react on fluctuations of the available power caused by electricity consuming devices (e.g., machinery, HVAC devices) or by energy producing assets (PV, CHP).

- **Management of heterogeneous equipment:** A company's CI can contain AC and DC chargers of various vendors, types and versions. Considering only "abstracted" equipment in the software system can lead to severe problems, because "real" devices behave differently with respect to their, e.g., charging curve characteristics, in-/output ratios, means of data provisioning, interfaces, configurable parameters, etc. The larger the CI, the greater the cumulative effect of these factors can be.

- **Support of EV-specific charging:** During a charging session, the EV's battery management system may autonomously increase (or lower) the demanded power. As a reaction, the SCS may limit the maximum available power, or provide the EV with additional power, e.g., by re-scheduling other EVs' charging sessions. Accordingly, the SCS requires up-to-date information about connected vehicles, incl. maximum allowed current/power, the number of phases used, etc.

- **Context-aware prioritization:** In the business context, a prioritization of charging sessions is often needed: A salesperson, who wants to visit a customer and needs a "full" battery within two hours, has higher priority than another employee, who leaves the office at the evening. To determine prioritization, data from different sources are required, e.g., planned arrival time (*ERP*), estimated departure time (*Forecasting*), capacity of EV batteries (*Fleet Management*), current SoC (*Vehicle Backend, Mobile app, DC Charging Stations*), etc.

- **Interoperability and scalability:** The SCS must seamlessly interact with other system components over available interfaces and network protocols. It should also be able to serve CIs of different size and allow adding (removing) locations to the overall setup.

- **Flexibility:** CI sites have different properties and characteristics, for example, with regard to the number and type of served EVs, usual charging times, local infrastructure limitations, etc. Consequently, the structure and operational complexity of the SCS also varies between deployment sites. In order to address this, the SCS needs to be built modular and thus adaptable to the given infrastructure, EV-fleet, user needs and prioritization requirements. In general, the SCS must be able to work in different complexity levels and enable adding/removing components independently from each other.

- **Exception handling:** In case of errors, e.g., due to malfunctioning charging stations or EVs, a proper exception handling in near real-time is needed. Thereby, vendor- and device-specific error messages must be captured and properly interpreted. It must also be ensured that failing or bad network connectivity (HTTP, WebSocket, TCP/IP) does not jeopardize running charging sessions and missing data is handled when planning new sessions. If there was an outage in the local electrical system, a safe restart of charging procedures is required.

2.2 Smart Charging Core

The scheduling algorithm (see Algorithm 1) implemented by the *Smart Charging Core* component is based on previous research results [4]. It shares the basically limited charging power at a given location among connected EVs in a fair manner. The SCS triggers the calculation when a new charging session starts, or an ongoing session ends. Scheduling can also be done periodically (e.g., every 15 minutes) or when significant changes in the amount of available energy are detected (e.g., through solar production). The algorithm initially creates a "greedy" charge plan for each ev in $evList$ for n time slots of duration d represented in $tsList$. In a practical setup, for example, with $n = 96$ and $d = 0.25$ hours, a charge plan for the next 24 hours can be created.
By executing Algorithm 2 for each EV (Line 2-4), the maximum possible charging current will be assigned to each EV - taking into account limitations of the given EV ($ev.I_{max}$) and the charging point ($cp.I_{max}$). This is repeated for the next time slots until the sum of the EV's initial charge capacity $ev.cap_{init}$ (measured in Ah) and charged capacity $ev.cap_{cha}$ reaches/exceeds the battery's maximum capacity $ev.cap_{max}$. Note that $ev.cap_{cha}$ is calculated based on the charging current I assigned to the EV and the total duration of passed k time slots.
To face potential conflicts that could occur if the total scheduled charging power within one or more time slots exceeds power limitations of the charging infrastructure, some EVs' initially created charge plans must be adjusted, i.e., delayed. For that purpose, the EVs are ranked by executing Algorithm 3 in Line 5 (see explanation below). In order to determine the critical time slots, $sumI_{ts}$, the sum of charging currents assigned to all EVs in $evList$ in each time slot is calculated. A particular time slot will be $blocked$ (see Line 13) if the resulting value is not below the relevant technical limitation of the charging site's electrical system (called *fuse limit*). The $sumI_{ts}$ is reduced by the previously given charging current I of the lowest ranked EV (see Line 14), whose charge plan will be re-filled.
Afterwards, the EV with the lowest priority is rescheduled by applying Algorithm 2 in Line 15. The reduction of the charging current to zero in all blocked time slots (see Line 6 in Algorithm 2) leads to a delayed/prolonged charging of the particular EV, because the intended cap_{max} value cannot be reached otherwise. This shifting procedure is repeated for the next ranked EVs until the violation of the fuse limit within the time slot is solved.
Note, that an adjustment of the charging current in the last unblocked time slot to match fuse limits more exactly is implemented, but not included in the pseudo-code due to readability and space reasons.

Algorithm 1 Scheduling procedure

procedure SCHEDULE($evList, tsList$)
for $i \leftarrow 1$ to $evList.length$ **do**
FILLPLAN($ev[i], tsList$) ▷ Algorithm 2 called
end for
PRIORITIZE($evList$) ▷ Algorithm 3 called
for $k \leftarrow 1$ to $tsList.length$ **do**
$sumI_{ts} \leftarrow 0$
for $i \leftarrow 1$ to $evList.length$ **do**
$sumI_{ts} \leftarrow sumI_{ts} + evList[i].tsList[k].I$
end for
$index \leftarrow 1$
while $sumI_{ts} \geq$ fuse limit **do** ▷ Check if total current exceeds limit
$tsList[k] \leftarrow blocked$ ▷ Block time slot for rescheduling
$sumI_{ts} \leftarrow sumI_{ts} - evList[index].tsList[k].I$
FILLPLAN($evList[index], tsList$) ▷ Reschedule EV with lowest priority
$index + +$
end while
end for
end procedure

Algorithm 2 Procedure to fill EV charge plans

procedure FILLPLAN($ev, tsList$)
for $k \leftarrow 1$ to $tsList.length$ **do**
if $tsList[k]$ **not** $blocked$ & $ev.cap_{init} + ev.cap_{cha}(k) \leq ev.cap_{max}$ **then**
$ev.tsList[k].I \leftarrow min(ev.I_{max}, cp.I_{max})$ ▷ Assign lower value of CP/EV max current
else
$ev.tsList[k].I \leftarrow 0$
end if
end for
end procedure

Algorithm 3 Prioritization procedure

procedure PRIORITIZE($evList$)
for $i \leftarrow 1$ to $evList.length$ **do**
$mCap_{minSoC} \leftarrow evList[i].cap_{des} - (evList[i].cap_{init} + evList[i].cap_{cha})$
$\Delta t \leftarrow evList[i].t_{dep} - evList[i].t_{now}$
if $mCap_{minSoC} \geq 0$ **then** ▷ Assign higher prio if EV is below minimum SoC
$evList[i].priority \leftarrow mCap_{minSoC}/((\Delta t * evList[i].I_{max}) + 1e - 8)$
else
$mCap_{maxSoC} \leftarrow evList[i].cap_{max} - (evList[i].cap_{init} + evList[i].cap_{cha})$
$evList[i].priority \leftarrow mCap_{maxSoC}/((\Delta t * evList[i].I_{max}) + 1e - 8) - 1000$
end if
end for
sort $(evList, priority)$ ▷ Sort according priority
return
end procedure

The aforementioned prioritization of EVs for being potentially re-scheduled is done in Algorithm 3. To rank EVs in $evList$, the missing capacity to reach the minimum SoC $mCap_{minSoC}$ (measured in Ah) is calculated for each EV. It is the difference between the EV's desired charge capacity cap_{des} (calculated from the desired SoC, in %, as entered by the user) and the sum of its initial capacity cap_{init} on arrival and the capacity charged since then cap_{cha} (Line 3). The urgency of charging depends on the available time Δt between departure time t_{dep} (e.g., entered by the EV-driver) and current time t_{now}. If the minimum SoC is not yet reached, the priority is calculated based on $mCap_{minSoC}$, the urgency Δt and the maximum charging current I_{max} of the EV (see Line 6). The applied formula basically ensures that EVs/drivers with higher energy demand and less time for charging will get a higher priority in average and thus will not be taken as first candidates for being "shifted". The other EVs that already reached their minimum expected SoC will be ranked based on the charge capacity that is missing to reach the maximum capacity of the vehicle's battery $mCap_{maxSoC}$. The chosen formula (see Line 9) gives in average a higher rank for those EVs with higher energy demand and less available time to fully charge their batteries. Accordingly, first candidates for re-scheduling will be those EVs that almost reached their batteries' maximum capacity and still have time to wait.

2.3 Integration Layer

To leverage the capabilities of the generic *Smart Charging Core* component and to configure the implemented algorithms properly, information from several, heterogeneous data sources with regard to, e.g., APIs, security settings, data formats, etc. must be gathered. In case these sources are not deployed in the given environment and/or (temporarily) unavailable, the algorithms must be provided with preset values to ensure operational safety at any point in time. Similarly, a calculated charging plan must be transmitted to all connected charging stations and the respective EVs, so that they can interpret received messages (commands) and set configuration parameters or return data as requested. Within the SCS, these data-oriented tasks are carried out by the *Integration Layer*. This component basically allows the adaptation of the *Smart Charging Core* to the given context and operational environment. Note that the SCS currently supports OCPP version 1.6 [3]. Accordingly, the *Integration Layer* creates and maintains a Charging Profile for each connected, OCPP-compatible charging station within the CI. A fundamental task thereby is to handle misbehaving or incompatible charging stations. For that purpose, the *Integration Layer* monitors and reflects the current status of the CI, and can react on events that occurred. It can also collect data about ongoing charging sessions in near real-time, and help re-distribute the available power according to the actual power consumption of ongoing sessions. Below, the tasks and functionality of the *Integration Layer* are explained in more detail.

- **Error and Exception Handling:** The SCS presupposes a proper implementation of OCPP by the charging stations and the support of OCPP charging profiles. However, OCPP implementations vary by charging station manufacturer and model. Compatibility problems often appear in specific setups and cases, which were not known upfront. The *Integration Layer* offers different mechanisms to master such situations. When collecting data to properly configure the core scheduling algorithm, the capabilities of connected stations are checked. It is especially examined, whether each station is able to work with the generated OCPP profiles. If not, the given charging station will be excluded from the optimization, because it cannot be limited. In order to not endanger the electrical infrastructure, the SCS will automatically adjust infrastructure limits for the next optimization cycles by subtracting the maximum power that the incompatible station can draw. The adjustment of these limits only happens, if the affected charging station is charging. Otherwise, the full capacity can be considered by the optimization. A similar mechanism is applied if a charging station is rejecting or not answering to charging profile requests, e.g., due to network issues. In this case, the faulty stations are collected and handled as incompatible charging stations in a separate optimization cycle. At the same time, a notification framework informs the administrators about the stations, which are not working correctly in order to take action if the issue persists.

- **AC/DC Handling:** The *Integration Layer* supports both AC and DC charging sessions according to their specifics. AC sessions can use one, two or all three phases depending on the given charging station and connected EV. When triggering the *Smart Charging Core*, this information must be taken account to determine the demanded charging current per phase. DC stations usually use all three phases, which makes phase assignments redundant. For DC chargers, however, the efficiency values need to be taken into account, because the conversion from AC to DC is carried out by the charging station and not by the EV (as in case of AC chargers).

- **Vendor-specific handling:** Charging station vendors tend to vary in how they handle OCPP charging profiles, for example by using their preferred measurement units (kW or Ampere). Therefore, the *Integration Layer* provides a framework and mechanism to adapt a generic charging profile template to vendor specific requirements.

[3] https://www.openchargealliance.org/protocols/ocpp-16/

- **EV-specific handling:** With the help of the *Fleet Management* component, the SCS is able to retrieve data about converters and batteries of almost every EV-model on the market, by using the Electric Vehicle Database [4] and other similar repositories. To keep the vehicle data up to date, synchronization jobs with the respective data sources are implemented. The data can be used to instantiate vehicles of a certain type in *Fleet Management*. By assigning these vehicles to users, the system can determine which EV model is charging at which station, without the need to establish a communication channel to the EV itself. The extracted information (converter data, battery size) is used to send power- and battery capacities to the *Smart Charging Core* without waiting for monitored consumption data and adopting to it later. In addition, the system provides implementations of service interfaces offered by OEMs, such as Mercedes, and also by third-party service platforms, like Tronity, to receive live information about the current state of charge during AC-sessions. The stored data of the EVs is extended with this information and can be provided for different purposes such as priority handling.

- **Real-time behavior adaptation:** The process of EV charging (both AC and DC) can be influenced by many factors. The charging curve, i.e., the power drawn over time, depends not only on the type, age and condition of the hardware on the vehicle side, but also on external parameters such as temperature. In some cases significant deviations from the expected model-specific behavior can be observed when charging a particular EV. A negative implication can be that EVs consume less power than expected and allocated to the session when it started. Especially DC chargers manage power usage actively, by monitoring the connected battery's charging status. An efficient charging system must react to varying (in general unpredictable) power curves. The *Integration Layer* captures the momentary power drawn in the charging sessions and supports the SCS in calculating the real charge demand of a particular vehicle. The implemented mechanism puts a buffer on top of the observed power output of the charging station and uses the increased value as a new power limit for the session, whereby the new limit remains below the connector's maximum limit. By supervising whether the EV uses the buffer, the algorithm can determine if the car would be able to draw more power and provide it to the session if available. This way, it is also ensured that incorrect or missing vehicle data does not lead to the allocation of later unused power capacity.

- **Dynamic power limits:** In most cases charging stations are operated in combination with other energy consuming or producing devices. The amount of demanded and produced power within an electrical system can heavily vary depending on season, time of the day, temperature, weather, etc. Setting a fixed, safety-oriented power limit for the CI could make it basically independent from the fluctuations, but lead to lower throughput and efficiency. For this reason, the SCS can be integrated with external EMS that monitors and controls the overall electrical infrastructure on site. This integration should be as flexible as possible, to support as many EMS-providers as possible. Thus, the SCS provides an API endpoint to push energy data, but also integrates with external APIs to pull/request data from. By taking into account EMS-data, it is possible to dynamically adapt the available power for the CI according to the current solar production, building consumption, etc.

- **Priority Handling:** To support the prioritization of EVs (as shown in Algorithm 3) and the related charging sessions, the SCS collects as much information as possible. For instance, the *Mobile App* provides a dialogue for the driver to enter her planned departure time, required state of charge and also the current state of charge (if the data is not provided by another integrated source or service). After the data is collected, it is processed and passed to the *Smart Charging Core*, which uses the received parameters to determine, which EVs are prioritized and can thus charge faster. This ensures a fair sharing of power among trustworthy EV drivers and helps minimize inactivity times.

3 Implementation and Deployment in a Real-World Testbed

The SCS, along with other components of "Open E-Mobility", was implemented and tested iteratively in a period of three years. The process included multiple development phases and related test cycles. In each development phase, a new, encapsulated and independent component was added to the existing system so that the improvements introduced were feasible and measurable. The evaluation of the system in an operative environment took place at the premises of SAP Labs France in Mougins, France. It started on April 1, 2020, which was a good time to start field testing because the charging infrastructure was less stressed as usual (due to COVID-19) and users were therefore more tolerant of potential technical problems. As time went on, the number of charging operations increased again, and so that the scalability of the system could be tested as well. The testbed on site comprises currently 38 charging points (31 AC and 7 DC) from different vendors, incl. Schneider Electric, Legrand, ABB, Delta, IES, Webasto, Ebee, Mennekes, Keba, StarCharge, Wall Box Chargers and Joinon. The CI has been initially set up and maintained by local team members. The system served in total over 650 employees to charge their cars. Overall 291 cars of various vendors (incl. Tesla, Jaguar, Kia, Renault, Volkswagen, Audi, Mercedes, Hyundai, BMW, Fiat, Volvo, Nissan, etc) have been registered in the system as part of the EV-fleet.

[4] https://ev-database.org/

In total, more than 25,000 EV charging sessions were executed successfully, consuming in total almost 700 MWh energy with a combined session time of almost 3400 days. The system protected the power-constrained infrastructure well, as there were no overloads throughout the entire test period.
Before the deployment of SCS, it was possible to exceed the power limit of the site, for example, if a large number of EVs had to be charged simultaneously. With the roll-out of the first version of the SCS core system with its main components (i.e. without using any other external data sources), it was ensured for the first time that the maximum power limit of the local infrastructure could no longer be exceeded.
However, this "safety-first" strategy did not take into account the actual power limits of the vehicles' converters. Instead, the algorithm assigned to each charging session the maximum charging current, which was derived from the chargers' maximum output power, e.g., 22 kW in case of AC chargers. The actual assignment of the determined power to a particular charger takes place in updating the charger's OCPP Charging Profile using the message *SetChargingProfile.req*. As a result, the fixed maximum power limit of the CI was reached quickly, so only a few chargers could operate in parallel while the other charging stations received no power. The disadvantage of this approach is also illustrated in the upper part of Figure 2. In the example, a Tesla Model 3 charges constantly at 11 kW, although the connector has a maximum power of 22 kW. Without adjusting the limit to the actually demanded power, the SCS statically allocates 22 kW for the entire session duration. The unused, yet blocked 11 kW are "wasted", i.e., cannot not be given to other stations at the same time. For instance, in a CI segment created for testing with a power limit of 110 kW, only 5 EVs could be charged simultaneously.
Such inefficiencies motivated the incorporation of additional information into the charging-power calculation. The required data sources were added step-by-step by continuously extending and enhancing the *Integration Layer* and related other components. When retrieving the connected EV's actual demanded power at the beginning of a charging session (using the OCPP message *MeterValues.req*), the allocated power limit can be adjusted (lowered) by updating the OCPP Profile limit of the station. This adaptive adjustment of the power limit for a session is shown in the lower part of Figure 2. As a result, the charging algorithm can re-distribute the otherwise unused power among other charging stations. For technical and safety reasons, the actual limits per charger were calculated by adding a safety buffer to the observed power consumption. In the example shown in Figure 2, the buffer is set to approx. 20%. Accordingly, the limit for the charging session of the example Tesla Model 3 is set to 13.5 kW. Using this enhancement, the number of parallel powered sessions increased significantly, since 8 (instead of only 5) EVs with a power consumption of 11 kW each could be charged.
However, at the beginning of each session, the maximum connector power remains allocated and thus blocked for other sessions at least until the next execution of the scheduling algorithm. The applied safety buffer per station (approx. 2.5 kW in case of the exemplary Tesla) will not be usable by any other session at all.

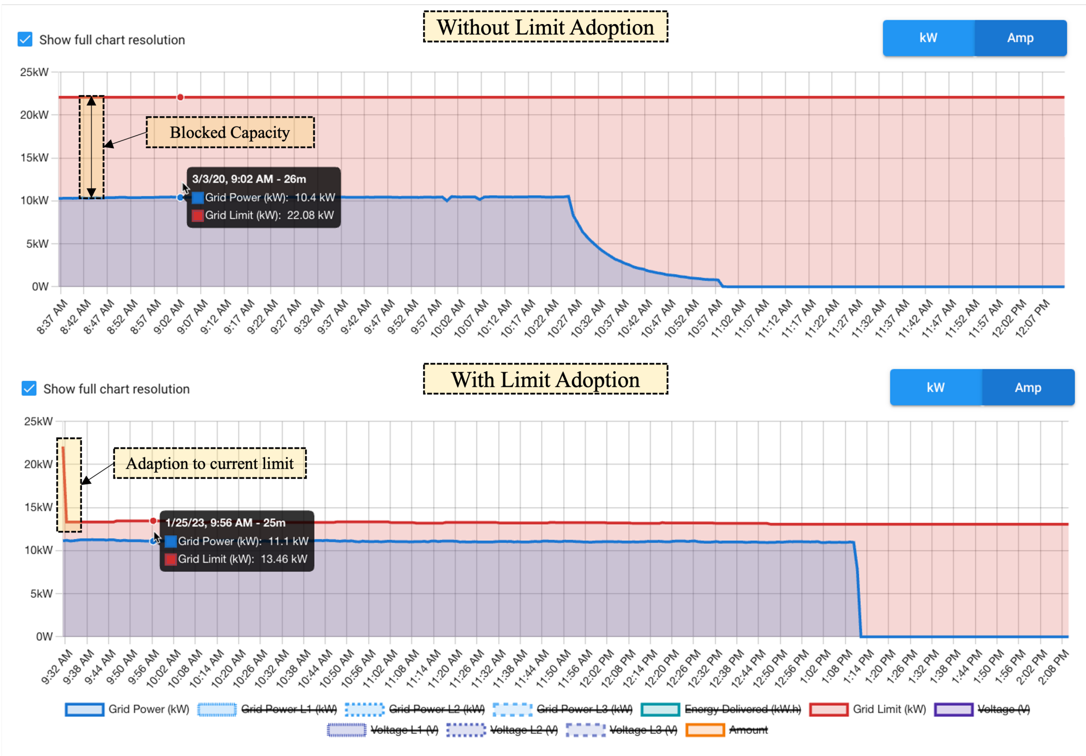

Figure 2: Power limit adaption to current consumption (screenshots from "Open E-Mobility")

To address this issue, the component *Fleet Management* was introduced. It provides model-specific master data about EVs and enables the assignment of particular EVs to drivers. When starting a charging session at a charging station, the driver is authenticated and thus a linkage to the data about the respective EV is established. By retrieving the electrical properties of the vehicle from the database, the maximum charging power of the EVs can be used in the optimization from the beginning on. It was now possible to assign 11 kW as the definite limit to the exemplary Tesla Model 3 without allocating any additional safety buffer. Figure 3 shows a comparison of charging the same EV with the above discussed limit adoption (in the upper part) and with the initially set model-specific power limit (in the lower part). In essence, it became possible to utilize the freed power at other stations in parallel. On the 110 kW infrastructure segment 10 (average) EVs could be charged at the same time without safety risks.

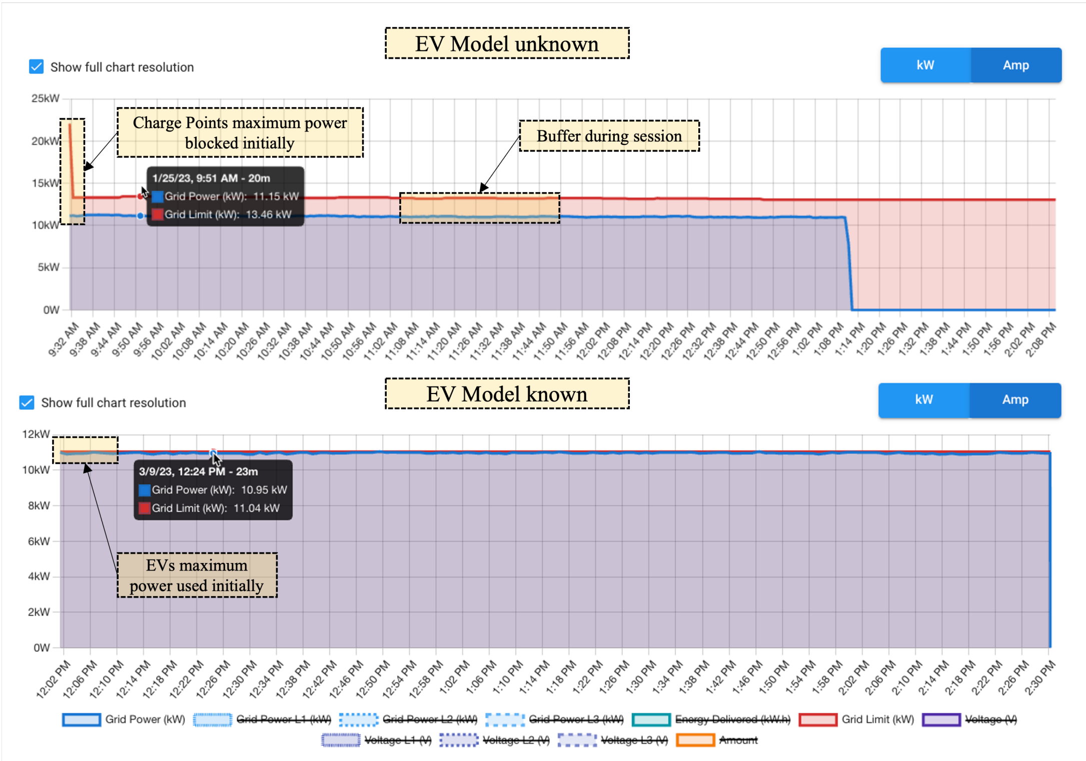

Figure 3: Utilization of vehicle-specific data in charging limit calculations (screenshots from "Open E-Mobility")

At that stage, the SCS was only able to efficiently distribute power within the CI according to a fixed maximum power that was set as a strict upper limit. The limit was determined, as a proportion of the maximum power consumed by the entire facility (mainly office buildings). Thereby, neither the actual power consumption nor the power provided by the building's PV system were considered as input parameters. Since a large number of energy consuming devices are not always in operation, and/or do not constantly draw a high amount of power, ignoring their actual energy consumption leads to a rather low power limit assigned to the CI. Similarly, considering the actual on-site power generation can help safely increase the CI's maximum power consumption limit.
The integration of the SCS with the locally installed *EMS* solved this issue. The EMS-vendor provides a REST API to query collected data about all connected and monitored devices, incl. solar panels and the stationary battery installed in the building. The continuous retrieval of the actual power consumption and production on site allowed the dynamic updating of the CI's maximum power limit. Using this integration feature, it was possible to allocate 50 kW additional power in average to the charging infrastructure. On the above mentioned 110 kW infrastructure it was now possible to charge up to 15 EVs at the same time on average. Figure 4 shows the power distribution of the testing facility while taking into account building consumption, solar power production and charging station consumption.
By combining all of the above system components and associated "live" data, the SCS was able to efficiently distribute power, while treating each EV charging session equally. This "democratic" approach is beneficial in some use cases, for example, when a logistics company's delivery vehicles shall be recharged during the night. However, in other settings, some vehicles must be served faster and/or charge a higher amount of energy than others to fulfil business-related requirements. Some EVs/drivers can have a longer stay at the charging facility and thus more time to charge than others. The vehicles' total charging demand may vary depending on the planned driving distances or specific routes the users need to drive till the next charge can occur.

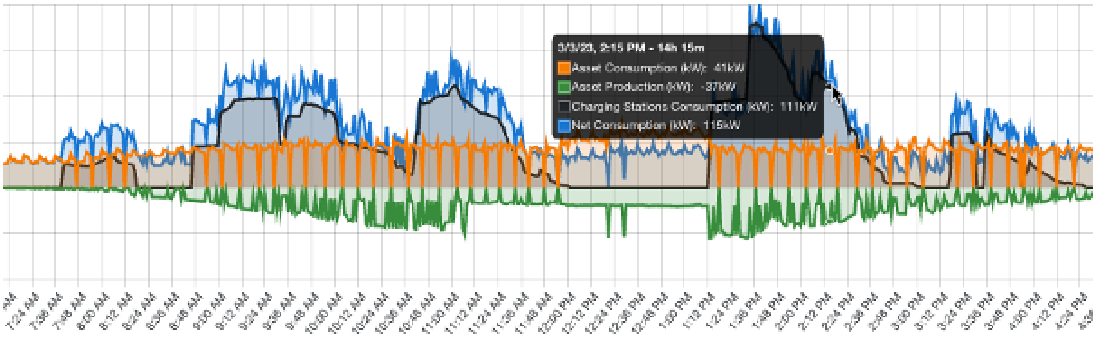

Figure 4: EV charging as part of the electrical infrastructure (screenshot from "Open E-Mobility")

To meet these requirements and preferences, the incorporation of further data items, such as the given EV's current and target SoC, as well as its planned departure time is required. These parameters can be provided, for example, by the user manually, via the *Mobile App* (see in 1). If user authentication is done without using the app, e.g., by presenting an RFID card at the charging station, default values for the above mentioned parameters are taken. By passing the collected data to the *Smart Charging Core*, the scheduling of sessions can be carried out according to users' actual needs, and energy can also be provided/distributed in a more efficient way. Figure 5 shows how prioritization effects the start of powering a charging session in the system according to the users' known planned departure time.
In the depicted example, two EVs, EV_1 and EV_2, arrive at 2:00 PM and start charging at the same time. EV_1 can stay till 6:00 PM, while EV_2 must leave earlier, at 4:00 PM. Due to the limited available power of approx. 11 kW (see the red line) only one EV can be charged at its maximum current. If EV_1 would be charged before EV_2, EV_2 would not have enough time to charge until it must leave, and EV_1 will be inactive after it was fully charged. If the system can take the planned departure times into account, it schedules EV_2 first, allowing to charge to full capacity before it has to leave at 4:00 PM. After that EV_1 can start and will have another two hours to charge until 6:00 PM. Viewing it from the involved drivers' perspective, in this particular example, the EV prioritization helps double the efficiency of the power-constrained infrastructure.

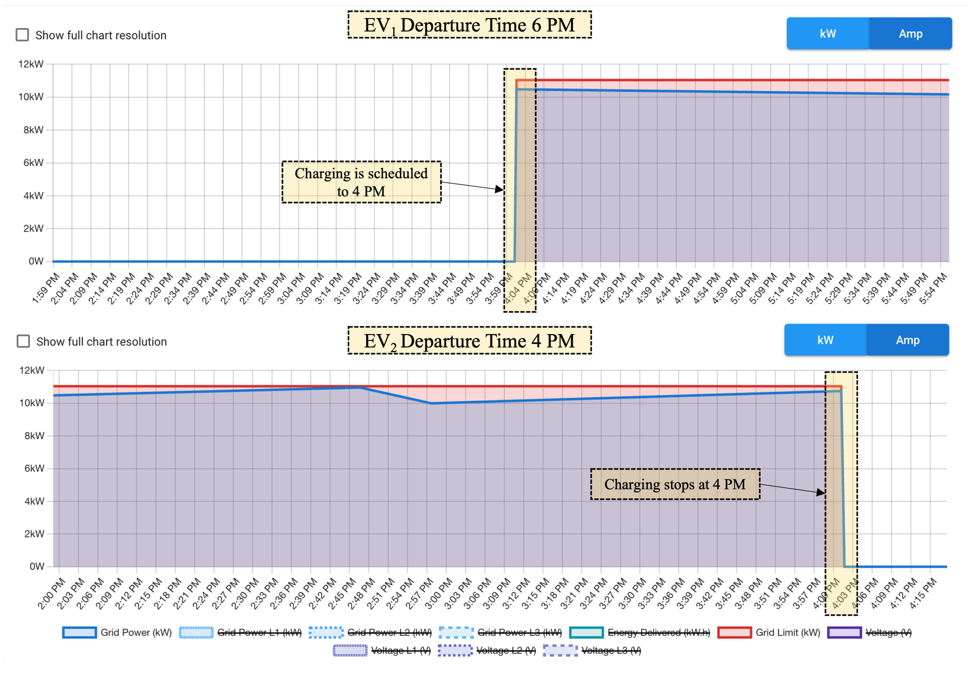

Figure 5: Effects of prioritization on two concurrent sessions (screenshot from "Open E-Mobility")

In addition to the support of rather passive AC chargers, the SCS is also able to deal with DC chargers that actively control charging processes while connected to an EV's battery. Figure 6 shows an example how our SCS combines different data to dynamically adjust the power limit (in red) during an ongoing DC charging session.

The information about the plugged vehicle's battery (in the example a Jaguar I-PACE EV400 with battery capacity of 90 kWh, and initial SoC of 30% which corresponds to 27 kWh) is used to initially set the maximum allowed power, which is 104 kW in this case. The DC-charger in the example, which is capable to deliver up to 150 kW, is regulated accordingly. The remaining 46 kW can be distributed among other chargers (as long as the resulting total power does not violate other thresholds). Over time, the car's battery management system automatically reduces the power drawn, in order to protect the battery. As a consequence, the power limit in the example session is re-adjusted (lowered) three times, making an increasing amount of power available for other (newly started or ongoing) charging sessions. The battery's increasing SoC is further used to re-calculate prioritization decisions (the effects of those are not depicted here).

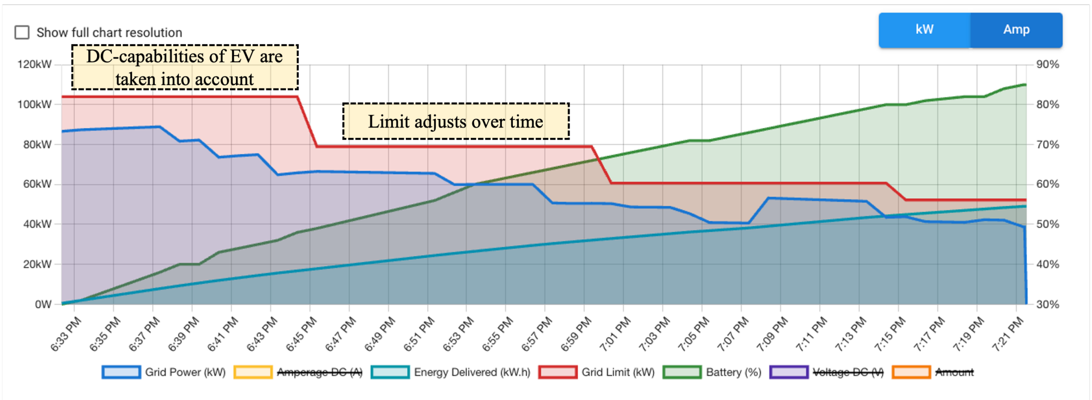

Figure 6: Example charging process on a Delta Ultra-fast Charger (screenshot from "Open E-Mobility")

As illustrated above, the SCS, in combination with the external components and data, can almost triple the efficiency of the power-limited charging infrastructure. To achieve similar results with a non-controlled hardware solution, the infrastructure limit would have to be tripled. For the above test infrastructure, this could require an increase in transformer power by 200 kW, which would result in very high costs.

4 Related Work

According to [5], smart charging can increase driver satisfaction by maximizing the average SoC across all EVs. To reach this goal in the context of a company fleet, in [2] different charging strategies called "baseline", "intelligent" and "multi-location" were proposed. Based on a dataset from a company-fleet, simulations showed that intelligent charging almost doubles the utilization of the infrastructure and the available power compared to the uncontrolled baseline charging.
In recent years, a lot of research has been done on EV charging, and the results have been summarized in various studies. For example, [6] reviewed seven case-study simulations in the context of smart charging (including those in [5]). To identify challenges of commercial EV charging, in [7] charging strategies were analyzed, including problems related to return-to-base scenarios. A comprehensive overview of smart charging applications together with an overview of publicly known pilot projects is provided in [8]. Case studies often include real-world testing in technically limited environments with a very small number of EVs. For example, in [9] a possible design of a charging infrastructure for company locations is presented, while considering charging preferences and trip data of a bakery in Germany. A three-day experiment in a so-called "mobility house" (containing student housing, a grocery store and a parking garage) showed that rule-based peak-shaving and load-demand forecasting can reduce load demand peaks by 25.4–38.5% while ensuring a minimum SoC of 50% [10].
The applicability of smart charging approaches that were designed specifically for charge-at-work scenarios, such as [11, 12], and various related scheduling strategies, e.g., [13, 14, 15, 16, 17] is usually evaluated in simulations rather than operational environments. The same holds for [3], which proposed a charging simulation model to support the design of a corporate charging infrastructure based on employees' driving data. Further challenges in the context of scheduling charging processes and related requirements for a software system are presented by [1].
In contrast to the above mentioned research, our smart charging approach and system is actively used to power a large number of EVs in resource-constrained environments. Nevertheless, simulation is a useful instrument to initially test and evaluate new features or the applicability of improvements before putting them into operation. The smart charging approach presented in [4] has since been further improved and evaluated using simulations, whereas the respective extensions have not yet been rolled out as part of live system deployments. The smart charging algorithm (see in 2.2) was extended by pre-computing day-ahead charging schedules using a mixed integer programming (MIP) model.

Thereby, EVs receive their pre-computed schedule, if they are in time and have the expected SoC. Simulations show that randomness in real-life settings makes pre-computed schedules impractical and thus prove the need for charge plan adaptation in (near) real-time [13]. Due to the non-linear charging behaviour of EVs in practice, an approach to predict the necessary charging power over the charging process based on historical charging processes is proposed. It helps avoid gaps between charge plans and the actual power consumption of EVs, which can lead to better utilization of available energy [5]. To predict EV departure times based on an existing dataset from a workplace reflecting employees' daily routine arrivals and departures, different regression methods can be used, incl. Oracle, Constant, Linear Model, XGBoost and Artificial Neural Network. In the simulations XGBoost performed best with a mean absolute error of 82 minutes. Smart charging simulation results showed how a higher prediction accuracy of departure times leads to a higher mean fraction of minimum SoC [18]. Furthermore, an objection function for time-slot prioritization is implemented, but not yet actively used in production. It can be used for ordering time slots by different prioritization objectives [13]: *Fair share* as the penalty cost of missing SoC, depending on the fraction of the minimum SoC that an EV lacks and the gap between minimum and full SoC. *Energy cost* for charging at a time slot with a high energy price. *Peak shaving* expresses the penalty cost as a system usage fee for the highest consumption peak.

5 Conclusions and Future Work

In this paper, we presented an extension to the open-source charging-point management system ”Open E-Mobility” that enables intelligent control of electric vehicle charging at enterprise sites. The Smart Charging System has already been successfully deployed and used in various charging infrastructures. Thanks to the system's modular structure and the realized multiple interfaces to numerous external data sources, different factors and data can be included in the calculation of charging plans, for both AC and DC charging stations.
We validated the positive impact of this flexible design in a real-world environment at SAP Labs France in Mougins, France. As illustrated by examples, the usage of specific information led to better power utilization and helped increase the overall effectiveness of the charging infrastructure.
The SCS will be enhanced by several features and functions in the near future. Currently, for example, it is not possible to create charging plans depending on variable or time-dependent electricity tariffs. The mainly economic impact of such tariffs on the calculation of charging plans has been studied theoretically in numerous publications, but has hardly been implemented in practice. Another aspect concerns the realization and integration of predictive algorithms to forecast the departure time and power demand of EVs. The current scheduler implementation assigns power limits to ongoing EV charging sessions based on actual information, i.e., previously set data without taking potential future data and related uncertainties into account. Regarding the communication with charging stations, it is also planned to support OCPP version 2.0.

Acknowledgments

As part of the research project *Green4EVer* [5] this work has been funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK, Ref. 01ME20009). We especially thank to our colleagues in the SAP Labs France team led by Hanno Klausmeier for their continuous support and Oliver Frendo for his expertise provided.

References

[1] N. Bodenschatz, M. Eider, and A. Berl, “Challenges and Requirements of Electric Vehicle Fleet Charging at Company Parking Sites,” Sep. 2021, pp. 623–628.

[2] O. Frendo, S. Karnouskos, N. Gaertner, O. Kipouridis, K. Rehman, and N. Verzano, “Charging strategies and implications for corporate electric vehicle fleets.” Porto: IEEE, Jul. 2018, pp. 466–471.

[3] M. F. Voss, S. P. Haveman, and G. M. Bonnema, “In-Company Smart Charging: Development of a Simulation Model to Facilitate a Smart EV Charging System,” *Energies*, vol. 14, no. 20, p. 6723, Jan. 2021.

[4] O. Frendo, N. Gaertner, and H. Stuckenschmidt, “Open Source Algorithm for Smart Charging of Electric Vehicle Fleets,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 6014–6022, Sep. 2021.

[5] https://www.h-ka.de/en/iaf/green4ever

[5] O. Frendo, J. Graf, N. Gaertner, and H. Stuckenschmidt, "Data-driven smart charging for heterogeneous electric vehicle fleets," *Energy and AI*, vol. 1, p. 100007, Aug. 2020.

[6] T. K. Bera, A. K. Bohre, I. Ahmed, A. Bhattacharya, and A. Yadav, "Smart Charging for Electric Vehicles (EVs): A Short Review," Sep. 2022, pp. 1–6.

[7] B. Al-Hanahi, I. Ahmad, D. Habibi, and M. A. S. Masoum, "Charging Infrastructure for Commercial Electric Vehicles: Challenges and Future Works," *IEEE Access*, vol. 9, pp. 121 476–121 492, 2021.

[8] S. Deb, M. Pihlatie, and M. Al-Saadi, "Smart Charging: A Comprehensive Review," *IEEE Access*, vol. 10, pp. 134 690–134 703, 2022.

[9] A. Waclaw, T. Aloise, and M. Lienkamp, "Charging Infrastructure Design for Commercial Company Sites with Battery Electric Vehicles: A Case Study of a Bavarian Bakery," Sep. 2020, pp. 1–10.

[10] A. Wallberg, C. Flygare, R. Waters, and V. Castellucci, "Peak Shaving for Electric Vehicle Charging Infrastructure - A Case Study in a Parking Garage in Uppsala, Sweden," *World Electric Vehicle Journal*, vol. 13, no. 8, p. 152, Jan. 2022.

[11] T. Ma and O. A. Mohammed, "Optimal Charging of Plug-in Electric Vehicles for a Car-Park Infrastructure," *IEEE Transactions on Industry Applications*, vol. 50, no. 4, pp. 2323–2330, Jul. 2014.

[12] G. Zhang, S. T. Tan, and G. G. Wang, "Real-Time Smart Charging of Electric Vehicles for Demand Charge Reduction at Non-Residential Sites," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4027–4037, Sep. 2018.

[13] O. Frendo, N. Gaertner, and H. Stuckenschmidt, "Real-Time Smart Charging Based on Precomputed Schedules," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6921–6932, Nov. 2019.

[14] W. Jiang and Y. Zhen, "A Real-Time EV Charging Scheduling for Parking Lots With PV System and Energy Store System," *IEEE Access*, vol. 7, pp. 86 184–86 193, Jan. 2019.

[15] S. Deb, A. K. Goswami, P. Harsh, J. P. Sahoo, R. L. Chetri, R. Roy, and A. S. Shekhawat, "Charging Coordination of Plug-In Electric Vehicle for Congestion Management in Distribution System Integrated With Renewable Energy Sources," *IEEE Transactions on Industry Applications*, vol. 56, no. 5, pp. 5452–5462, Sep. 2020.

[16] C. Rottondi, G. Neglia, and G. Verticale, "Complexity Analysis of Optimal Recharge Scheduling for Electric Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 4106–4117, Jun. 2016.

[17] K. Sepetanc and H. Pandzic, "A Cluster-Based Model for Charging a Single-Depot Fleet of Electric Vehicles," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3339–3352, Jan. 2021.

[18] O. Frendo, N. Gaertner, and H. Stuckenschmidt, "Improving Smart Charging Prioritization by Predicting Electric Vehicle Departure Time," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6646–6653, Oct. 2021.

Presenter Biography

Prof. Dr.-Ing. Zoltan Nochta received his PhD from the University Karlsruhe in 2004. He is Chief Development Expert at SAP SE and works as part-time professor at Karlsruhe University of Applied Sciences since 2014. At SAP he was responsible for several R&D projects on e-mobility. Currently, he coordinates the project Green4EVer, which aims the integration of e-mobility into company car fleets and related processes.