

# Electric Vehicle Fast-Charging Software: Architectural Considerations Towards Trustworthiness

Vick Dini, Damian Andrew Tamburri, and Elisabetta Di Nitto

Politecnico di Milano, DEIB, Via Golgi 42, Milan, Italy  
{vickpierce.dini,damianandrew.tamburri,elisabetta.dinitto}@polimi.it  
<https://www.deepse.deib.polimi.it>

**Abstract.** The industry of electric vehicle charging stations is rapidly evolving. As more people purchase electric vehicles, the need for an efficient and reliable charging service value chain becomes paramount. In this paper we perform a literature review and topic modeling in the field to identify the most relevant topics and challenges therein as well as any standard or reference software architectures proposed to date. Subsequently, by means of ethnomethodological research and interview study with industry practitioners from an emerging Charging Point Operator (CPO) we identify major practical architectural concerns. We conclude that typical -Ops issues related to monitoring and autonomous runtime management of fast-charging software architectures are not addressed in the literature so far, and yet are the most critical for practitioners; this inspired a revised augment of state-of-the-art reference architecture frameworks—offered as the first key original contribution of this paper—for practitioners and researchers to use in the future. Finally, evaluating such framework with practitioners’ insights, we elaborated a second key contribution of this work, namely, six architectural considerations to be taken into account towards trustworthy electric vehicle fast-charging software architectures of the future.

**Keywords:** charging station · cloud-edge computing · electric vehicle · fast charging · software architecture · software engineering.

## 1 Introduction

Nowadays, electric vehicles (EVs) are considered as one of the main solutions to reduce the carbon footprint in highly urbanized areas in the most industrialized countries [1]. However, a critical requirement for their diffusion is the availability of fast charging services [10].

Offering such services on a large scale requires several issues to be addressed, such as the need to: (a) install the EV charging stations (EVCSs) in strategic locations, which require planning and management software services close to the demand and to electrical power sources [38]; (b) fulfill an increasing customer demand—39.5% YOY [12], projected revenue of US\$623.3bn in the US

in 2024 [30]—keeping into account the limitations of the electrical grid [26]; (c) support all the EV charging standards as well as the interaction protocols with multiple stakeholders, e.g. vehicle owners and manufacturers, energy distributors (DSOs), and e-mobility service providers (eMSPs) [9].

Several big players of the EV industry are developing proprietary software systems capable of addressing such challenges. For example, Tesla not only covers the electric vehicle production chain, but also creates and manages EV charging stations to provide full support to its customers [25]. Moreover, multiple companies, including Charging Point Operators (CPOs), are backing the standardization of communication protocols (e.g. OCPP, OCPI, OSCP, and ISO 15118) and software solutions, such as the Charge Point Management System (CPMS) and Energy Management System (EMS), that address the most critical aspects concerning the interoperability and efficiency of the electric vehicle charging infrastructure. This concerted effort aims to streamline the deployment and operation of charging stations across diverse platforms.

In particular, CPOs are highly relevant to our research, as they are the *last mile* of the smart grid with respect to EVs and are thus crucial to the extension of the EVCS network and consequently to the generalized adoption of EVs.

However, through a pilot study of the literature, we report that the development of a comprehensive and open architecture framework capable of addressing all aspects of EVCS management and ensuring trustworthy, orchestrated cooperation between different stakeholders and their software assets is still missing. Thus, we defined the following research questions:

- RQ1: What is the state of the art in software architectures for EVCS?
- RQ2: Is research in this context focusing on trustworthy DevOps?
- RQ3: What are the main problems perceived by EVCS operators (CPOs)?

In addition to providing answers to these questions (RQ1 in Section 3, RQ2 in Section 4, and RQ3 in Section 5), our objective in this paper is to contribute to the aforementioned definition by looking into what considerations shall be made from an architectural perspective over such expected research results.

## 2 Research Methodology

The research methodology we have adopted is based on the three steps described in the following subsections.

### 2.1 Literature Analysis Approach

We performed a literature review to answer RQ1 and identify the main research topics in the EVCS state of the art. Our working hypothesis is that work has already been done on trustworthy DevOps for EVCS. To elaborate on such a hypothesis, we adhere to guidelines specifically recommended for information systems research by [32]. To attain our primary studies, we ran the following

query on the well-known Scopus search engine<sup>1</sup> on April 3<sup>rd</sup> 2024 and obtained 4,544 results in computer science.

*electric AND vehicle AND charging AND stations*

We did not include the keywords *software architecture* in the query because we also wanted to identify the trending topics in computer science. Following the aforementioned recommendations, we excluded the articles written in languages other than English (-148) and kept only scientific papers (-77). Then, we manually excluded the results whose titles are not related to stationary EVCS software, control, or energy management systems (-4,245). From the remaining 74 articles we excluded 9 which were inaccessible and applied the Latent Dirichlet Allocation (LDA) [4] on the remaining ones. This allowed us to identify the most relevant topics in this area. We then selected the 14 papers related to software architecture applied to the EVCS industry for further analysis.

## 2.2 Architectural Analysis

To address RQ2, we used literature review sources to distill an architecture overview for EVCS in industry with a twofold approach. On the one hand, we intended to distill an architecture overview which explicitly addressed trustworthiness as a primary architecture asset while, on the other hand, we aimed for such an overview to be eventually validated with the aid of practitioners involved in the matter. Even though multiple architectures for the EVCS industry have been created over time, we indeed found that none of them was designed starting from a hierarchical EVCS control model (such as the one proposed by [37]) and taking into account software feature/component deployment and orchestration, and the respective flexibility and resilience thereby obtained [13].

## 2.3 Distilling Architectural Considerations

To address RQ3, we conducted an industrial confirmation study featuring a six-month ethnographic study [34], which consisted of four steps: (a) daily participation in activities of our target company; (b) four unstructured interviews [6] to gauge current architectural status; (c) two semi-structured interviews [6] to understand current challenges; (d) an interview analysis [18] to distill a taxonomy of concepts. An architecture refinement approach, as well as problem detection and cause identification [36], was employed to model current baselines, together with a card-sorting exercise [39]<sup>2</sup>. This approach allowed us to distill six key architectural considerations to be made when reflecting on the trustworthiness of electric vehicle fast-charging software; these considerations, grouped into four

<sup>1</sup> <https://www.scopus.com/>

<sup>2</sup> For confidentiality, we excluded the generated artifacts from this paper.

categories: (1) EVCS management, (2) Data Management, (3) Service Continuity, (4) IoT & Electronics, are intended as design principles to drive architecture decision-making, as well as constraints to orchestrate architectures and their self-adaptation.

### 3 Electric Vehicle Charging Stations Overview

To analyse our primary literature studies, we applied the well-known LDA topic modeling approach [4] to the text of the 65 accessible articles and removed the keywords included in the query, as well as adjectives, adverbs, and verbs, and considered the top 5 words in each cluster to obtain the topics in Table 1.

**Table 1.** LDA Topics.

#	Topic	% of tokens	Top 5 words
1	User management	20.3	system, user, datum, service, reservation
2	Power management	18.3	power, converter, battery, voltage, system
3	System control	14.7	power, system, energy, battery, control
4	Edge scheduling	12.5	power, energy, time, datum, node
5	Event management	12.4	time, energy, event, user, system
6	Application data management	12.0	datum, system, user, application, service
7	Mobility data management	4.3	network, datum, mobility, service, node
8	Edge control	3.6	terminal, station, microcontroller, time, process
9	Energy forecast	1.8	time, energy, battery, source, traffic

We gave these topics their names by reflecting on feasible relationships among the terms associated with, or contained in each cluster. We further refined the topic names by modifying the values of the relevance metric ( $\lambda$ ) parameter between 1.0 and 0.5, while observing the changes in the order of the top 5 words. In Table 1 the column about the percentage of tokens indicates the size of each cluster, which may also be interpreted as the frequency and relevance of each topic with respect to the contents of the papers under analysis. The intertopic distance map and ordered list of most salient terms are illustrated in Fig. 1. Additionally, according to the location of the clusters on the principal components (i.e. PC1 and PC2), we named each of the quadrants coherently as follows: Q1: Event management and power control; Q2: User management; Q3: Application and scheduling data management; Q4: Edge control and forecasting.

Following a snowballing approach [35], we added a group of publications about topics other than software to the set of articles under analysis, in addition to key standards in the EVCS context: (a) the Open Charge Alliance<sup>3</sup> (OCP and OSCP), (b) EVRoaming Foundation<sup>4</sup> (OCPI), and (c) International Standards Organization (ISO 15118<sup>5</sup>), as shown in Table 2.

<sup>3</sup> Open Charge Alliance: <https://openchargealliance.org/>

<sup>4</sup> EVRoaming Foundation: <https://evroaming.org/>

<sup>5</sup> ISO 15118: <https://www.iso.org/standard/77845.html>

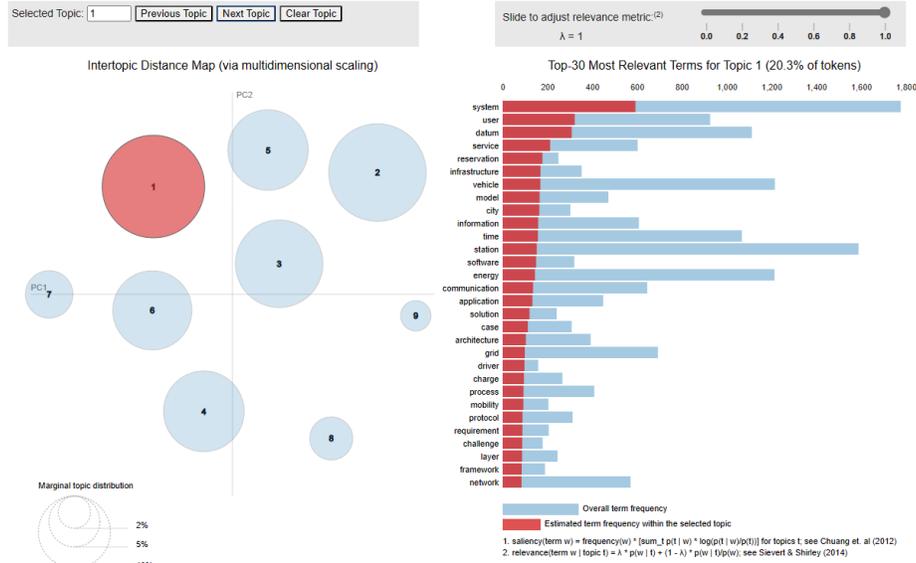


Fig. 1. Intertopic Distance Map & Most Salient Terms.

### 3.1 Main Stakeholders, Components, and Protocols

By analyzing the standards, we gained an overview of the key players in EVCSs and a top level perspective of the organization of each station. Figure 2 illustrates the scenario of a CPO with two EVCSs in different locations. Each station may include several charging points (EVSE: Electric Vehicle Supply Equipment), batteries to accumulate and distribute energy as needed (managed by a BESS: Battery Energy Storage System), other sources of energy (e.g. solar panels), smart meters that connect the station to the external grid, and an Energy Management System (EMS) in charge of controlling the energy exchange among the different sources (e.g. the external grid, batteries, and solar panels) and consumers. The CPO usually relies on a CPMS, which acts as an intermediary between EVCSs and eMSPs, enabling the CPO to communicate with its customers. The CPO also interacts with the DSOs, which are intermediaries in the energy supply market. The interactions among parties occur through the specific protocols highlighted in Figure 2.

Table 2. Parties and Associated Interaction Protocols.

Parties	Protocol	Acronym
eMSP - CPO/CPMS	Open Charge Point Interface	OCPI
CPO - DSO	Open Smart Charging Protocol	OSCP
CPO/CPMS - CPO/EVSE	Open Charge Point Protocol	OCPP
EV - EVSE	Road vehicles – Vehicle to grid communication interface	ISO 15118

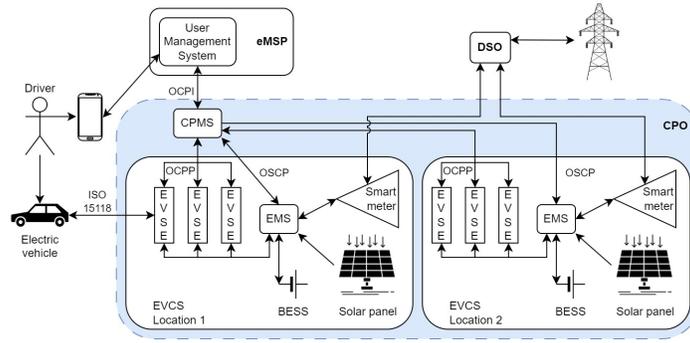


Fig. 2. EVCS Communications and Systems.

### 3.2 Critical EMS Service Functions

A general overview of EVCSs from an energy management perspective is provided by Wu et al. in [37]. Here the authors highlight the hierarchical EVCS control structure, present a list of the main standards and protocols used in this industry, and mention several features that a software system should include to support the EVCS control system, such as the implementation of strategies for energy allocation, trading, and demand forecasting.

The prominent role of EMSs in EVCSs is highlighted in multiple publications. [29] states that EMSs “are automation systems that collect energy measurement data from the field and make it available to users through graphics, online monitoring tools, and energy quality analyzers, thus enabling the management of energy resources.” Specifically, according to [3], their features could include:

- Power Limitation: The EMS can receive power limitation signals and adjust the energy flow of connected devices, including EVSEs, accordingly. This ensures compliance with physical and grid-initiated constraints at the grid connection point, supporting stable local grid operation.
- Dynamic Pricing: The EMS can receive the rates from the DSO and adjust the overall power consumption of connected devices.
- Load Management: The EMS can safely limit the overall EV power draw on the system so that it does not exceed the available capacity. It can also efficiently balance and distribute charging power to consumers.
- EV Charging: The EVSE or EV submits its energy demand and estimated departure time to the EMS. If supported, the end user could choose CO<sub>2</sub> friendly charging and the EMS would take care of charging with a higher portion of solar energy, when available.

### 3.3 EMS Architecture

As of the EMS’s architecture, [28] highlights that it is “based on different layers including management, data communication, vehicle control, electric transmission, and electric backup and storage modules.” Moreover Torreglosa et al. [31]

make a distinction between centralized and decentralized EMSs. In a centralized EMS, a single control system manages all energy sources. Integrating additional sources requires modifications to the EMS and installation of a communication infrastructure between the new sources and the centralized system. Instead, in a decentralized EMS, each energy source works independently, managing energy flow among the system’s sources without requiring communication interfaces between sources or with a supervisory energy system.

More recently, [22] proposed a charging station management system featuring decentralized, fair, and impartial energy trading via a blockchain distributed ledger and smart contracts; as of the system’s control hierarchy, the authors placed centralized control at the station level and decentralized control on the EVSEs, while the cloud server is mostly dedicated to the energy trading platform.

### 3.4 EVCS Service Orchestration and Monitoring

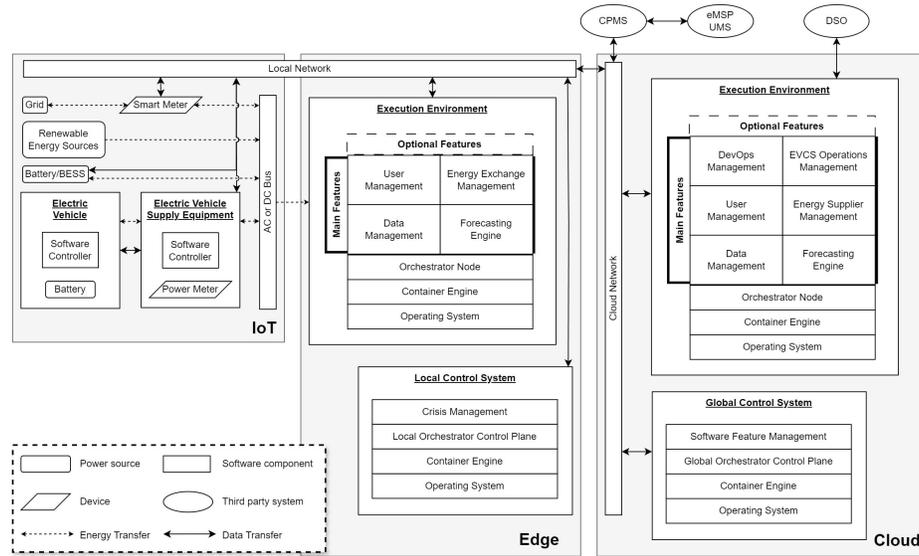
In [27] the authors present a framework for the interoperation and integration of heterogeneous systems in the industrial automation domain and evaluate its application to the EVCS industry through a case study, although it must be noted that their focus is on service orchestration (on “local clouds”) and IoT, and that by service orchestrator they intend a service dispatcher or router. Later on, [21] developed a cloud-IoT monitoring system for solar panel stations, which they described as “robust, reliable, workable, and suitable.”

### 3.5 EMS Control Software Architecture

Some papers that emerged from the LR proposed software architectures that are similar to the OSI model, while others implemented variants of the client/server model. [33] designed an architecture comparable to the cloud-edge-IoT model. Among the others, we consider [19] to be a key paper in this research because its authors applied the Smart Grid Architecture Model (SGAM) to the e-mobility industry, focusing on interoperability and complexity management.

## 4 EVCS: a Revised Architecture Framework

Taking into account the findings of our literature review, we designed a distributed cloud-edge architecture framework (AF) (Fig. 3) for the fast charging domain, in which we mapped the topics that surfaced from the LDA as software components (Table 3) and allocated them in the respective execution environments. This AF should support the computing continuum and interactions from several stakeholders (multiple ownership), in addition to extending the stations’ service continuity by making them more autonomous in local decision making. Optimal EVCS management is of critical importance to EV drivers, especially those who urgently need to charge their vehicles and do not have any options nearby. In the following subsections we delve into the components of our proposed solution.



**Fig. 3.** Proposed Distributed Cloud-Edge Fast Charging Architecture Framework.

**Internet of Things** From left to right, the EVCS architecture starts with the power sources, which may be renewable or fossil-based. The former refers to the power grid (to which the station connects via a smart meter) [11], whereas the latter includes wind turbines [37], train generative braking [17], and solar panels. It should be noted that some or all of these renewable sources might not be available in all EVCSs. Regarding energy storage, EVCSs could house batteries (with their respective BESS) and electric vehicles contain their own battery cells, which, if bidirectional energy transfer is supported, may store and return energy to a common bus. In fact, all of these sources are connected to a common AC or DC bus, which provides energy to charging points and electric vehicles (EVs) [37]. As of the latter, their batteries (controlled by the BMS: battery management system) provide energy to vehicle devices and motors, and the respective processes are managed by a software controller. Charging points have microcontrollers [5] that manage their operation and react to user input and power meter data, in addition to receiving settings and transmitting data about a charging point’s operation [14]. In particular, the smart meter, BESS, and EVSE are connected to the station’s local network.

**Control Systems** EVCS systems must be kept under continuous control to ensure safety of users and continuity of service. Given the presence in EVCSs of the three contexts, IoT and edge at the periphery and cloud at the center, we deem the three-level control hierarchy presented in [37] particularly relevant for the EVCS industry. We could consider the primary control level to be in charge of overseeing EVSEs, whereas the secondary control level manages EVCS devices

**Table 3.** Topic-Component Mapping.

Topic	Component
User Management	User Management
Power Management	Energy Exchange Management, Energy Supplier Management
System Control	EVCS Operations Management, DevOps Management
Edge Scheduling	User Management, Forecasting Engine
Event Management	Crisis Management, EVCS Operations Management
Application Data Management	Data Management, DevOps Management
Mobility Data Management	Data Management, User Management
Edge Control	Energy Exchange Management, EVCS Operations Management
Energy Forecast	Forecasting Engine

and, in turn, communicates with the tertiary control level located in the cloud. Also, there is a bidirectional data flow that starts in the cloud, passes through the EVCS controllers, and reaches the EVSEs, and vice versa. Building on this idea, the tertiary control level should be deployed to execution environments on the cloud to better exercise the global (centralized) monitoring and control of the EVCS microgrid via industrial and single-board computers, which lie at the secondary control level and perform local (decentralized) tasks. The primary control level should manage the digital electronics of each EVCS.

Such multi-layer control system is primarily in charge of guaranteeing EVCS architectures *trustworthiness*, defined as the combination of a system’s reliability and recoverability as they are described in the ISO 25010 standard<sup>6</sup>:

**Reliability:** *Capability of a product to perform specified functions under specified conditions for a specified period of time without interruptions and failures.*

**Recoverability:** *Capability of a product in the event of an interruption or a failure to recover the data directly affected and re-establish the desired state of the system.*

We deem that by placing container orchestrators both locally (edge) and globally (cloud) we can increase the trustworthiness of a distributed architecture, because orchestrators offer benefits such as lower mean time to repair (MTTR) and higher availability by detecting and correcting issues automatically. They improve infrastructure performance through efficient resource management and workload distribution and automate container management tasks like deployment, scaling, and operations. Containers package applications with their dependencies, ensuring consistent functionality and reducing overhead by providing a tailored minimalistic execution environment. They streamline continuous deployment through automated testing, integration, and deployment. Containers enhance security and stability by isolating applications from the underlying OS and other applications. They also enable faster deployment times and more efficient resource utilization compared to traditional virtual machines.

**Execution Environments and Functional Components** We placed execution environments at the cloud and edge to run the components that emerged

<sup>6</sup> ISO 25010: <https://www.iso.org/standard/78176.html>

from our literature review and others that may arise over time. We therefore analyzed several aspects of these components to discern the most appropriate execution environment(s) for each of them, resulting in Table 4.

**Table 4.** Key Architectural Drivers for EVCS: A Comparative Analysis.

Component	Cloud (Global)	Edge (Local)	Resource Intensive	Continuous Execution	Execution Environment Specific	Industry Specific
Data Management	✓	✓	✓			
Forecasting Engine	✓	✓	✓			
User Mgmt	✓	✓				
Energy Exchange Mgmt		✓		✓	✓	✓
Energy Supplier Mgmt	✓				✓	✓
DevOps Mgmt	✓			✓	✓	
EVCS Operations Mgmt	✓			✓	✓	✓
Crisis Mgmt		✓		✓	✓	

These components may be translated to microservices, which are “small applications with a single responsibility that can be deployed, scaled, and tested independently”[20], to accommodate new features, address emerging challenges, and optimize performance without compromising reliability. However, designing an architecture based on this principle has several challenges, such as lower performance (vs. monolithic applications), the increased complexity in monitoring operations and debugging code, and how to guarantee data consistency [23]. Nevertheless, applications designed for our architecture framework could dynamically scale by leveraging the expansive capabilities of the cloud [24], while ensuring that latency-sensitive tasks are executed at the edge [8]. This dynamic scalability accommodates varying workloads and provides a cost-effective solution by optimizing resource allocation [16].

As of their functionalities, the energy exchange and energy supplier management components should focus on automated dynamic energy procurement and sales among DSOs, CPOs, and drivers, in addition to adaptive energy storage and allocation, depending on changing market conditions. Similarly, the DevOps and EVCS operations management components should cover the DevOps life cycle and the centralized monitoring and control of charging stations.

Meanwhile, we placed the forecasting engine and the data and user management components in both the cloud and edge execution environments, because they could offer functionalities that are useful, and even necessary, at both ends, considering the limitations of the underlying hardware.

Finally, we designed the crisis management component to handle all unforeseen adversities and events at the stations and guarantee a certain level of service continuity while the system recovers or adapts to the new circumstances.

## 5 Trustworthy EVCS: Architectural Considerations

### 5.1 Ethnographic Study about the Proposed Architecture

For industrial grounding and confirmation of the aforementioned architecture framework, we interviewed practitioners who work for a CPO that is a key

player in the Southern European market of fast-charging stations, powered by renewables, energy storage, and vehicle-grid integration. They intend to deploy 5,000 remotely managed fast charging points by 2025 and more than 35,000 by 2030. Additionally, they outsource certain services and systems to focus on their core business. To delve into the architectural design, implementation, and maintenance of an EMS infrastructure, we carried out four unstructured interviews at the company, each one lasting one work day, which helped us perform an initial validation of our theoretical EVCS model. Then, we held two semi-structured interviews focused on the software aspects of EVCSs and problems in this industry that might be solved with software. Next, we analyzed the interviews by reviewing the recordings and transcripts through content analysis. Afterwards, we categorized the synthesized data according to each question and produced conceptualizations to ground the software architecture to the industry. Finally, we elicited the most significant problems by reviewing the interviews and diagrams with the practitioners. Specifically, a thematic analysis [18] of the interview data revealed recurring challenges and issues raised by participants, providing qualitative insights into critical areas of concern in this industry. These are discussed in the following subsections.

## 5.2 EVCS Software Architecture Management

Neither the state of the art nor the insights from practitioner interviews provide clear guidelines on how to divide EVCS management software into components and which execution environment is more convenient to converge towards a computing continuum. Yet, this is reported as a critical aspect for the stations to function in a highly autonomous fashion. The following consideration is made:

**1: Computing Continuum of EVCSs is a First-Class Citizen.**

The plurality and heterogeneity of devices and stakeholders that compose and interact with an EVCS lead to a greater relevance of the continuum in this industry. It is thus crucial to determine which features shall be allocated to which layer of the computing continuum.

Subsequently, we report emerging needs for the underlying software architecture to offer a high degree of flexibility. For example, the deployment of updates to the industrial computers that control each EVCS involves a certain degree of complexity. These devices may run a variety of component containers, serverless functions, native OS programs, or even virtual machines, and thus require architectural flexibility based on the capabilities of each computer and its components. Additionally, in such a continuously evolving industry, the underlying technologies could change anytime. The following consideration therefore stands:

**2: DevOps-oriented Architecture Flexibility.**

Metrics shall be defined to measure the extent of flexibility (e.g. modifiability analysis [2]) of EVCS architectures and how they exert continuity over the surrounding DevOps pipelines. The use of abstraction layers should help increase system flexibility[7].

Finally, the absence of established guidelines, standards, or universally acknowledged best practices in this realm increases the degree of complexity of the architectural decision-making process.

**5.3 Data Management**

Regarding data management, the interviews helped us quantify the data volume transmitted and received daily by each EVCS (~100 MB). This empirical data is significant for further (e.g., dataflow) architecture-based experimentation given its absence within existing literature and the related lack of ad-hoc experimentation on the matter. The following consideration is made:

**3: Controlled Architectural Experimentation Shall Reflect Dataflow Modelling & Analysis.**

Many approaches that feature dataflow modelling and analysis may be leveraged to better define the constraints and execution policies under which EVCS management design can be performed.

At the same time, disparities between the data stored in the cloud and the edge can emerge, resulting in *data divergence* [15]. This phenomenon introduces a significant challenge for architectural decision-making processes, as insights derived from cloud-based data may no longer accurately reflect the real-time conditions and dynamics occurring at the edge and vice versa. Consequently, decisions based on outdated or divergent data may prove inadequate or even counterproductive, leading to sub-optimal outcomes and operational inefficiencies. The following consideration is made:

**4: EVCS Architecture Decision-Making is a Continuous Architecturing Exercise.**

Fostering closer alignment between cloud and edge data entails leveraging advanced analytics and machine learning algorithms to detect and reconcile discrepancies in architectural decision-making in real-time, reflecting a continuity in such decisions; new adaptation mechanisms shall hence assume such decision continuity and operate decision-making in account of it.

Thus, organizations shall proactively identify cases of data discrepancy and implement corrective measures to re-align cloud and edge data streams, e.g. by continuously monitoring data integrity and consistency across distributed nodes.

#### 5.4 Service Continuity

During the interviews, the participants elucidated various factors contributing to service interruptions within the EVCS infrastructure, including that:

*Fault logging is a whole issue on its own, because typically when the system fails, it either brings down components or even the entire site [i.e. EVCS]. Therefore, understanding how to ensure the collection and retention of information, and the ability to recover when the site is down can save you from physically going there to conduct diagnostics.*

Hence, the potential for industrial hardware to experience system halts or encounter bugs within its software is a notable concern, as it may result in operational disruptions, especially because usually no one is present at the EVCSs. In the event of a system hang-up in a containerized environment, an orchestrator mechanism can swiftly detect and rectify the issue by initiating the deployment of a new container while decommissioning the malfunctioning one. Similarly, in the case of bugs affecting serverless functions, the system's automated processes can seamlessly execute error-handling protocols to ensure uninterrupted service delivery. Thus, the following consideration is made:

**5: EVCS Architecture Management is an Orchestration-centric Exercise.**

By adopting containerization or serverless architectures, EVCS operators can enhance system robustness and minimize the likelihood of prolonged service disruptions. These deployment strategies not only facilitate rapid fault recovery but also afford greater scalability and agility in adapting to evolving operational requirements.

The interviewees also highlighted the risk of communication disruptions in charging stations and charging points. This stresses the criticality of maintaining seamless connectivity to ensure uninterrupted service delivery. Therefore, future experimentation must be directed towards implementing robust redundancy measures over said orchestration exercises to mitigate the impact of potential communication misalignment or other failures.

#### 5.5 IoT & Electronics

The practitioners gave particular importance to phenomena concerning charging-points malfunctioning due to firmware bugs. Unlike industrial computers, IoT devices typically lack the resources to run containerized environments or lambda functions. However, the imperative to swiftly detect and address charging point failures is crucial in ensuring seamless service continuity and enhancing user experience. In light of this challenge, the following consideration is made:

**6: EVCS Architectures Require Fault Detection by Design.**

Proactive measures such as automated system alerts and remote troubleshooting protocols enable operators to expedite the resolution of firmware-related issues, thus maximizing service uptime and availability.

While IoT devices may not inherently support containerization or serverless computing paradigms, alternative architecture approaches—and the connected fault-tolerance analysis—can be employed to mitigate the impact of firmware bugs on EVCS operations. For instance, leveraging edge computing capabilities, such as edge analytics and edge processing, can empower IoT devices to perform localized fault detection and remediation tasks autonomously. By distributing computational resources closer to the point of data generation, edge computing frameworks expedite response times and reduce reliance on centralized infrastructure for critical decision-making processes.

**6 Conclusion**

We consider this paper to be a preliminary document on software architecture principles applied to the electric vehicle charging stations industry, upon which we expect more authors to elaborate and develop ad hoc solutions. Throughout its sections, we applied qualitative research methods to gather relevant information that helped us model a cloud-edge fast charging architecture framework and identify current challenges in the industry, thereby grounding the theory to the practice and with hopes of developing an industry-actionable solution. In addition to providing answers to the research questions (RQ1 in Section 3, RQ2 in Section 4, and RQ3 in Section 5), we hope to have contributed to the scientific community by delving into important aspects of EVCS management from a software architecture perspective. In the future we plan to follow up on our considerations with an experimental approach, in search of quantitative guidelines to compound each consideration.

**Data Availability and Acknowledgements**

The Scopus search results, the filtered articles, the LDA Jupyter notebook, and the interview questionnaire are openly available on our online repository<sup>7</sup>. This work is partially supported by the Italian Ministry of University and Research under the PNRR program (financed by the EU, NextGenerationEU), Ministerial Decree n. 352, and Atlante SRL.

**References**

1. Asimow, N.G., Turner, A.J., Cohen, R.C.: Sustained Reductions of Bay Area CO2 Emissions 2018–2022. *Environmental Science & Technology* (2024)

<sup>7</sup> <https://zenodo.org/doi/10.5281/zenodo.11060482>

2. Bengtsson, P., Lassing, N., Bosch, J., van Vliet, H.: Architecture-level modifiability analysis (ALMA). *Journal of Systems and Software* **69**(1–2), 129–147 (Jan 2004)
3. Bhattacharjee, S., Nandi, C.: Advanced Energy Management System (A-EMS) Design of a Grid-Integrated Hybrid System. *Iran J Sci Technol Trans Electr Eng* **47**(3), 1021–1044 (2023)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (Mar 2003)
5. Bose, P., Sivraj, P.: Smart charging infrastructure for electric vehicles in a charging station. In: ICICCS. pp. 186–192. IEEE (2020)
6. Brinkmann, S.: Unstructured and semi-structured interviewing. *The Oxford handbook of qualitative research* **2**, 277–299 (2014)
7. Broniatowski, D.A.: Flexibility due to abstraction and decomposition. *Systems Engineering* **20**(2), 98–117 (2017)
8. Carvalho, G., Cabral, B., Pereira, V., Bernardino, J.: Edge computing: current trends, research challenges and future directions. *Computing* **103**, 993–1023 (2021)
9. Chamberlain, K., Al-Majeed, S.: Standardisation of uk electric vehicle charging protocol, payment and charge point connection. *World Electric Vehicle Journal* **12**(2), 63 (2021)
10. Chatterjee, P., Hermwille, M.: Tackling the challenges of electric vehicle fast charging. *ATZelectronics worldwide* **15**(10), 18–22 (2020)
11. Das, H.S., Li, S., Chowdhury, M.F.F., Ghosh, T.: Game theoretical energy management of ev fast charging station with v2g capability. In: PECon. pp. 344–349. IEEE (2020)
12. Evans, G.: 2024 EV Forecast: The Supply Chain, Charging Network, and Battery Materials Market (12 2023)
13. Flora, J., Gonçalves, P., Teixeira, M., Antunes, N.: A study on the aging and fault tolerance of microservices in kubernetes. *IEEE Access* **10**, 132786–132799 (2022)
14. Germanà, R., De Santis, E., Liberati, F., Di Giorgio, A.: On the participation of charging point operators to the frequency regulation service using plug-in electric vehicles and 5g communications. In: *EEEIC/I&CPS Europe*. pp. 1–6. IEEE (2021)
15. Gonçalves, A.M.A.: Estimating Data Divergence in Cloud Computing Storage Systems. Ph.D. thesis, Universidade Nova de Lisboa (2013)
16. Gonzalez, N.M., Carvalho, T.C.M.d.B., Miers, C.C.: Cloud resource management: towards efficient execution of large-scale scientific applications and workflows on complex infrastructures. *Journal of Cloud Computing* **6**(1), 1–20 (2017)
17. Hernandez, J., Sutil, F.: Electric vehicle charging stations fed by renewables: PV and train regenerative braking. *IEEE Latin America Transactions* **14**(7), 3262–3269 (2016)
18. Hsieh, H.F., Shannon, S.E.: Three approaches to qualitative content analysis. *Qualitative Health Research* **15**(9), 1277–1288 (2005)
19. Kirpes, B., Danner, P., Basmadjian, R., Meer, H.d., Becker, C.: E-mobility systems architecture: a model-based framework for managing complexity and interoperability. *Energy Informatics* **2**, 1–31 (2019)
20. Larrucea, X., Santamaria, I., Colomo-Palacios, R., Ebert, C.: Microservices. *IEEE Software* **35**(3), 96–100 (2018)
21. Lee, H.C., Liu, H.Y., Lin, T.C., Lee, C.Y.: A Customized Energy Management System for Distributed PV, Energy Storage Units, and Charging Stations on Kinmen Island of Taiwan. *Sensors* **23**(11), 5286 (2023)
22. Lin, Y.J., Chen, Y.C., Zheng, J.Y., Shao, D.W., Chu, D., Yang, H.T.: Blockchain-based intelligent charging station management system platform. *IEEE Access* **10**, 101936–101956 (2022)

23. Liu, G., Huang, B., Liang, Z., Qin, M., Zhou, H., Li, Z.: Microservices: architecture, container, and challenges. In: QRS-C. pp. 629–635. IEEE (2020)
24. Mahmood, Z.: Cloud computing for enterprise architectures: concepts, principles and approaches. In: Cloud computing for Enterprise architectures, pp. 3–19. Springer (2011)
25. Naor, M., Coman, A., Wiznizer, A.: Vertically integrated supply chain of batteries, electric vehicles, and charging infrastructure: A review of three milestone projects from theory of constraints perspective. *Sustainability* **13**(7), 3632 (2021)
26. Oliinyk, M., Džmura, J., Kolcun, M., Humenik, J., Kanalik, M., Pavlik, M., Pal, D., Medved', D.: Impact of electric vehicles and demand management systems on electrical distribution networks. *Electrical engineering* **104**(2), 667–680 (2022)
27. Péceli, B., Singler, G., Theisz, Z., Hegedűs, C., Szepessy, Z., Varga, P.: Integrated infrastructure for electro mobility powered by the arrowhead framework. *IEEE Access* **6**, 73210–73222 (2018)
28. Qureshi, K.N., Alhudhaif, A., Jeon, G.: Electric-vehicle energy management and charging scheduling system in sustainable cities and society. *Sustainable Cities and Society* **71**, 102990 (2021)
29. Segatto, M.E.V., de Oliveira Rocha, H.R., Silva, J.A.L., Paiva, M.H.M., do Rosário Santos Cruz, M.A.: 14 - telecommunication technologies for smart grids: Total cost optimization. In: Yahyaoui, I. (ed.) *Advances in Renewable Energies and Power Technologies*, pp. 451–478. Elsevier (2018)
30. Statista: Electric Vehicles, <https://www.statista.com/outlook/mmo/electric-vehicles/worldwide>
31. Torreglosa, J.P., García-Triviño, P., Fernández-Ramirez, L.M., Jurado, F.: Decentralized energy management strategy based on predictive controllers for a medium voltage direct current photovoltaic electric vehicle charging station. *Energy Conversion and Management* **108**, 1–13 (2016)
32. Tranfield, D., Denyer, D., Smart, P.: Towards a methodology for developing evidence-informed management knowledge by means of systematic review. *British journal of management* **14**(3), 207–222 (2003)
33. Vujasinović, J., Savić, G., Prokin, M.: Terminal for Remote Control of Renewable Energy Sources Powered Station for Electric Vehicles Charging. In: MECO. pp. 1–6. IEEE (2021)
34. Wilson, W.J., Chaddha, A.: The role of theory in ethnographic research. *Ethnography* **10**(4), 549–564 (2009)
35. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: EASE. ACM, New York, NY, USA (2014)
36. Wong, K.C.: Using an ishikawa diagram as a tool to assist memory and retrieval of relevant medical cases from the medical literature (2011)
37. Wu, Y., Wang, Z., Huangfu, Y., Ravey, A., Chrenko, D., Gao, F.: Hierarchical operation of electric vehicle charging station in smart grid integration applications—an overview. *International Journal of Electrical Power & Energy Systems* **139**, 108005 (2022)
38. Xiong, Y., An, B., Kraus, S.: Electric vehicle charging strategy study and the application on charging station placement. *Autonomous Agents and Multi-Agent Systems* **35**, 1–19 (2021)
39. Zimmermann, T.: Card-sorting. In: Menzies, T., Williams, L.A., Zimmermann, T. (eds.) *Perspectives on Data Science for Software Engineering*, pp. 137–141. Academic Press (2016)