Applying Distributed Case-Based Reasoning in the Edge-Cloud Continuum*

Alexander Schultheis[1,2], David Jilg[1,2], Ralph Traphöner[3], Roman Schander[1,2], and Ralph Bergmann[1,2]

[1] German Research Center for Artificial Intelligence (DFKI)
Branch Trier University, Behringstraße 21, 54296 Trier, Germany
{Alexander.Schultheis,David.Jilg,Roman.Schander,Ralph.Bergmann}@dfki.de
[2] Artificial Intelligence and Intelligent Information Systems, Trier University, 54296 Trier, Germany, http://www.wi2.uni-trier.de
{Alexander.Schultheis,David.Jilg,s4rnscha,bergmann}@uni-trier.de
[3] Empolis Information Management GmbH, Europaallee 10, 67657 Kaiserslautern, Germany;
Ralph.Traphoener@empolis.com

Abstract *Case-Based Reasoning* (CBR) is well suited for recurring but context-dependent industrial decisions and is therefore a promising artificial intelligence approach for the *Edge-Cloud Continuum* (ECC). The ECC denotes a distributed infrastructure in which tasks are placed dynamically across resource-constrained nodes near data sources and centralized cloud resources. However, the systematic transfer of CBR to distributed edge-cloud settings has received little attention. This paper presents a project-grounded approach for operationalizing CBR in the ECC. Based on experience from the *EASY* project and exchanges with industry and research partners, industrial application scenarios are structured, load balancing metrics are adopted, and an architecture for distributed CBR is introduced. The paper further discusses opportunities, challenges, and cross-organizational perspectives. To demonstrate technical feasibility, a ProCAKE-based prototype is implemented and investigated in a distributed case base scenario. The results demonstrate that distributed CBR in the ECC is technically feasible, while highlighting the need to assess potential runtime benefits in relation to case base characteristics, heterogeneous processing capabilities, and distribution overhead.

Keywords: Distributed Case-Based Reasoning · Edge-Cloud Continuum · Load Balancing · ProCAKE

1 Introduction

Edge-Cloud Systems (ECS) distribute computation and data across heterogeneous resources ranging from edge nodes close to data sources, machines, sensors, or

* The final authenticated publication is available online at https://doi.org/10.1007/978-3-032-45236-8_18

controllers to centralized cloud platforms [30,36,38]. The *Edge-Cloud Continuum* (ECC) denotes a distributed infrastructure in which computations are placed dynamically across these tiers rather than statically assigned to edge or cloud [30, 36]. Such systems combine low-latency, data-local processing on resource-constrained edge devices with powerful cloud resources when needed [24,30,39,43]. Runtime decisions must therefore be made under changing constraints such as resource availability, network conditions, privacy requirements, and latency demands [2, 14, 30]. *Case-Based Reasoning* (CBR) is well suited to such recurring decisions because similar past cases can capture previous problem situations, including placements, constraints, and solution outcomes, that can be reused and adapted for new edge-cloud situations [1, 4]. This is particularly relevant in industrial manufacturing and *Industry 4.0* (I4.0), where *Artificial Intelligence* (AI) methods, data analysis techniques, the *Internet of Things* (IoT), and distributed systems are integrated into industrial processes and decisions recur under varying machine states, production conditions, and data characteristics [9, 30, 31].

The motivation for this paper arises from the *EASY* project (*Energy-Efficient Analysis and Control Processes in the Dynamic Edge-Cloud Continuum for Industrial Manufacturing*) [30][4] and exchanges with industry and research partners, which highlight the relevance of operationalizing CBR in industrial edge-cloud settings [2, 14, 18, 31]. However, a systematic transfer of CBR to distributed edge-cloud environments has received little attention. In particular, it remains unclear which industrial scenarios motivate such a transfer, how CBR knowledge containers [28] and tasks of the CBR cycle [1] should be distributed across edge and cloud, and which criteria should guide such placement decisions. While *Distributed Case-Based Reasoning* (DCBR) extends centralized CBR towards distributed architectures [26], existing approaches mainly structure this space in terms of knowledge organization and reasoning execution, and framework support remains limited [34]. Moreover, distributed execution cannot be assumed to be beneficial per se, since coordination and communication overhead may outweigh expected gains [2, 14, 43]. Unlike more general DCBR settings, the ECC requires placement decisions to account for heterogeneous infrastructure tiers, ranging from resource-constrained edge devices to more powerful cloud resources.

To address this gap, this paper presents an approach for operationalizing CBR in the ECC grounded in project experience from *EASY* and exchanges with industrial and research partners. The main contributions are a structured set of industrial application scenarios for CBR in edge-cloud systems and a concept for DCBR in the ECC comprising load balancing metrics, an architecture-oriented discussion of how CBR knowledge containers and CBR tasks can be distributed, and an analysis of opportunities, challenges, and cross-organizational potentials. To demonstrate technical feasibility, a prototype based on the ProCAKE framework [5, 6] is investigated in a feasibility test for distributed case bases.

The remainder of this paper is structured as follows. Sect. 2 introduces the required foundations. Sect. 3 presents industrial application scenarios for CBR in edge-cloud systems. Sect. 4 develops the proposed approach for CBR in the

[4] `https://easy-edge-cloud.de/`

ECC. Sect. 5 presents its prototype-based implementation and feasibility test. Sect. 6 concludes the paper and outlines future work.

2 Foundations

This section provides the background required for this contribution. Sect. 2.1 introduces distributed systems and load balancing, while Sect. 2.2 outlines DCBR. Finally, Sect. 2.3 introduces ECS and the ECC.

2.1 Distributed Systems

The term distributed system refers to a collection of autonomous computing elements, called nodes, appearing to a user as a single coherent system [38]. Such systems distribute computation, data, and other resources across multiple nodes to improve efficiency, scalability, and reliability. In practice, they are heterogeneous and dynamic, since nodes may differ in capabilities and operating conditions and available resources may change over time. All architectural forms require decisions about where data and computation are placed and executed.

The distribution of requests, tasks, or services across available resources is addressed by load balancing [8]. It aims to optimize objectives such as system utilization, response time, availability, resource consumption, and energy efficiency while taking distribution overhead into account. Since these objectives may conflict, decisions rely on metrics such as duration, CPU usage, RAM usage, permanent storage, network traffic, energy, and, where relevant, GPU usage.

2.2 Distributed Case-Based Reasoning

CBR is an experience-based problem-solving paradigm in which new problems are addressed by retrieving and reusing similar previously observed cases [4]. The CBR cycle [1] comprises retrieve, reuse, revise, and retain: prior solutions are adapted, evaluated, and stored as new experience when appropriate. Since retrieval depends on case representation and similarity assessment, both are central. CBR knowledge can be structured into vocabulary, case base, similarity measures, and adaptation knowledge [28].

DCBR generalizes the CBR setting, often realized as a single problem solver operating on a centrally maintained case base, towards architectures in which CBR resources and tasks are distributed across multiple computational entities [26]. A solver denotes the problem-solving component that performs CBR functionality, while agents denote computational entities that can own or execute such functionality. Cases, retrieval and adaptation functionality, or complete CBR applications can be distributed to improve scalability, responsiveness, and robustness or to respect organizational constraints such as separate ownership of experience repositories [26, 27]. Privacy and scalability are key motivations for moving beyond a single-agent, single-case-base architecture.

Plaza and McGinty [26] distinguish DCBR approaches along two orthogonal criteria: how experiential knowledge is organized and where knowledge is processed. This yields recurring patterns spanning *single-agent/multiple-case-base*, *multi-agent/single-case-base*, and *multi-agent/multiple-case-base* configurations. These configurations differ in whether case bases, reasoning entities, or both are distributed and therefore require different forms of coordination. While distribution can reduce bottlenecks and support heterogeneous knowledge sources, it also increases the importance of communication costs, trust boundaries, coordination policies, and coherence across evolving local case bases [26]. Existing DCBR work discusses efficiency gains from distributed computation and identifies privacy and scalability as motivations for distribution [26]. For the ECC, these considerations need to be complemented by explicit resource and placement criteria.

An overview of CBR frameworks [34] indicates that DCBR is rarely supported as a first-class feature and is typically only available through extensible architectures or bespoke integrations. Partial support is provided by *CloodCBR*, which decomposes CBR functionality into deployable microservices [23], and by *eXiT*CBR*, which realizes a cooperative multi-agent setup with aggregated assessments [25].

2.3 Edge-Cloud Computing

As a specific form of distributed system composed of cooperating nodes [38], ECS distribute computation and data management across geographically dispersed and heterogeneous resources spanning from edge devices close to data sources to centralized cloud platforms [36]. According to Schultheis et al. [30], the ECC denotes a distributed infrastructure that comprises edge devices and a central cloud and enables the dynamic placement of computations across all participating devices.

Driven by increasing data volume and dynamic workloads as well as cloud-centric limitations such as latency, bandwidth pressure, and local data-handling requirements, ECS relocate selected stages of the data value chain closer to sources [3, 7, 30, 36]. In the ECC, edge and cloud tiers differ systematically regarding compute and storage capacity, latency, data locality, and connectivity [30, 36]. Distributing CBR therefore becomes not only a question of knowledge organization and task execution, but also a placement problem under tier-specific resource asymmetries.

In industrial manufacturing, an ECC may range from physical production assets, such as robots and IoT controllers, to nearby edge nodes and a central cloud [16, 20, 22, 30]. Assets denote the physical or digital production resources connected to the continuum. Services can be executed on the closest tier with sufficient resources so that preprocessing and low-latency control remain near the machines while compute-intensive analytics may be shifted upward. The cloud can orchestrate placement decisions and provide centralized service management, for example, via a Gaia-X-aligned service repository [37], while objectives such as energy efficiency and sustainability can also be incorporated as optimization goals for AI service deployment [15, 30, 40].

3 Application Scenarios for Case-Based Reasoning in Edge-Cloud Systems

This section presents application scenarios for CBR in the ECC. The scenarios were collected in the context of the research project *EASY* [30] and through an expert task force [2, 14] within the German *Edge-Datenwirtschaft* technology program[5]. *EASY* was a collaborative project involving partners from industry and research [30] that ran from 2022 to 2025. The *Edge-Datenwirtschaft* technology program addresses decentralized data processing together with opportunities of the data economy, i.e., value creation from shared and usable data. Within this program, the task force investigated potentials and evaluation possibilities of ECS.

A central motivation for using CBR in industrial ECS is its knowledge-driven nature. In contrast to data-driven *Machine Learning* (ML) approaches, it can be effective even when only limited training data is available, which is common for rare failure modes and evolving production conditions [31]. Accordingly, CBR can support analytics-oriented scenarios such as predictive maintenance, activity detection, and the identification of data quality issues in IoT data [2, 14, 18, 29–31, 33, 35], as well as control tasks such as guiding resource-allocation decisions and adapting workflows across the ECC [2, 14, 17, 30]. Its distributed use is motivated by knowledge locality and computation, but its benefit depends on the trade-off between privacy or sovereignty constraints, scalability or latency gains, and communication, coordination, and aggregation overhead. Relevant evaluation dimensions include solution quality, computation time, energy consumption, and data traffic.

Beyond standalone CBR, the collected scenarios emphasize hybrid approaches that combine CBR with AI planning to address complex control problems in distributed industrial environments [2, 14, 18]. AI planning generates action sequences from an initial to a goal state [10, 12] and can support internal continuum processes, such as automated resource allocation, and manufacturing-related control processes in the ECC [2, 14, 19, 32]. Since planning is computationally expensive and distributed planning introduces overhead, it should be evaluated by criteria such as throughput time, energy consumption, plan length, or plan costs [2, 14]. Hybridization with CBR can mitigate this overhead. Fig. 1 illustrates such a hybrid scenario from the *EASY* project [30], in which a hybrid edge-cloud architecture generates production plans from customer requirements and semantic planning information, i.e., machine-readable information about available resources, capabilities, and constraints, potentially provided through *Asset Administration Shells* (AASs) [41]. Case knowledge can provide coarse solutions, control knowledge, or heuristics that accelerate planning, while planning methods refine and validate these candidates [2, 14, 18]. A representative scenario is adaptive replanning in production, where an edge node detects deviations

[5] `https://www.digitale-technologien.de/DT/Navigation/DE/ProgrammeProjekte/AktuelleTechnologieprogramme/Edge_Datenwirtschaft/edge_dw.html`

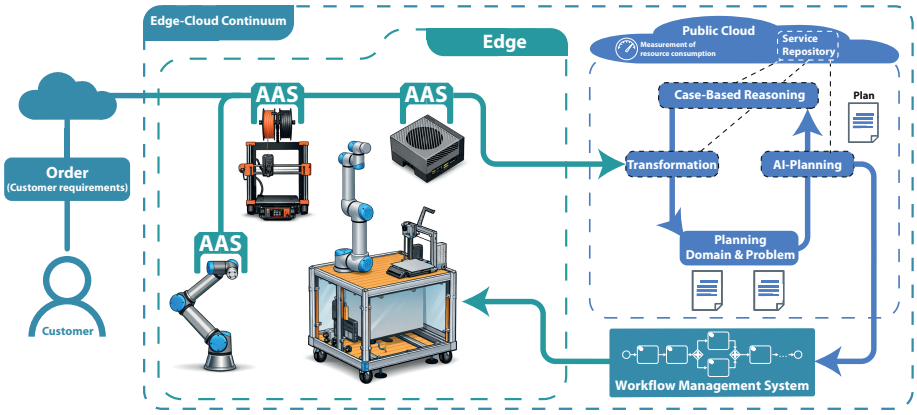


Fig. 1. Hybrid CBR Application within the *EASY* Project [30].

and triggers replanning. If local resources are insufficient, a coarse plan can be generated via CBR on another node, then refined and returned to the workflow engine [2, 14].

4 Case-Based Reasoning in the Edge-Cloud Continuum

This section introduces the proposed use of CBR in the ECC. First, relevant metrics for placement decisions for CBR tasks in the ECC are presented in Sect. 4.1. Building on this, Sect. 4.2 introduces the architecture for distributed CBR in the ECC. Sect. 4.3 then discusses the resulting opportunities and challenges. Finally, Sect. 4.4 outlines additional potentials of this form of CBR use, particularly for cross-organizational settings.

4.1 Metrics for Load Balancing in the Edge-Cloud Continuum

As discussed in Sect. 2.1, load balancing requires explicit metrics to decide how requests, tasks, or services should be distributed across available resources. This also applies to the distribution of CBR tasks in the ECC, where shifting retrieval, adaptation, or complete CBR cycles between edge nodes and the cloud introduces coordination and communication overhead. Consequently, the benefit of distribution has to be assessed using metrics selected according to the requirements of the application scenario.

The metrics considered for CBR in the ECC are adopted from Guldner et al. [11], whose reference model derives metrics from the measurement goal and the measured object. This matches the present context because the measured object is the distributed execution of CBR services and the measurement goal is to determine whether distribution provides measurable advantages over centralized alternatives within one organization. Tab. 1 summarizes these metrics.

Recent work on guided measuring and benchmarking in the ECC supports this perspective by emphasizing comparable evaluation procedures for heterogeneous edge-cloud environments [13]. In the *EASY* project, these metrics were discussed for distributed AI methods, including federated learning, AI planning, and CBR [30].

For CBR in the ECC, the system boundaries of the measurement have to be defined explicitly, since the measured object may comprise only retrieval, the full CBR cycle, or also communication and aggregation steps. Following Guldner et al. [11], the assessment also has to distinguish between baseline, idle, usage, and load situations. The metrics capture measurable performance-related effects such as execution time, resource consumption, energy demand, and communication overhead. Further design criteria, including data sovereignty, privacy, security, trust, and interoperability, are discussed separately because they may determine whether CBR functionality can be distributed at all.

4.2 Architecture for Case-Based Reasoning in the Edge-Cloud Continuum

Architecture Overview. Building on the general *EASY* architecture for the ECC, CBR in this setting can be operationalized as a service distributed between edge nodes and the cloud. Fig. 2 illustrates an architecture in which physical production assets are connected to local edge nodes that host CBR functionality and local knowledge containers, while the cloud side provides an orchestrator for load balancing and placement decisions as well as a central service repository for discovery. In line with *EASY*, this repository serves as an interoperable entry

Table 1. Relevant Metrics for Load Balancing in the ECC (Adopted and Adapted From Guldner et al. [11])

Measure / Metric	Example Unit	Prerequisites and Required Measures
Duration	s	Recording timestamps, usage scenario
Mean Power Draw	W	Duration, power meter or internal power logger, a usage scenario
Energy	Wh or J	Power, duration, baseline, usage scenario
CPU Usage	%	Resource logger, usage scenario, baseline
RAM Usage	MB or %	Resource logger, usage scenario, baseline
Permanent Storage	MB	Resource logger, usage scenario, baseline
Network Traffic	MB	Resource logger, usage scenario, baseline
GPU Usage	%	Resource logger, usage scenario, baseline
GRAM Usage	%	Resource logger, usage scenario, baseline
Mean GPU Power Draw	W	Supported hardware with internal power logger, usage scenario
GPU Energy	Wh or J	Mean GPU power draw, usage scenario, baseline
Energy Efficiency Factor	$\frac{\text{item}}{\text{J}}$	Useful work and energy

point for describing and providing CBR services, with Gaia-X as a reference model for sovereign and interoperable service and data exchange [37]. Services can be executed locally, cooperatively across multiple nodes, or centrally in the cloud depending on the scenario. In terms of the DCBR typology, the proposed architecture primarily targets a multiple-agent setting, while a centralized single-agent cloud configuration remains an option when metrics or hardware limitations indicate that local execution is unsuitable [26, 30].

Distribution Dimensions. Building on the distinction between knowledge organization and processing in DCBR [26], the ECC motivates a broader view along two dimensions: 1) the distribution of knowledge containers and 2) the distribution of CBR phases. Accordingly, knowledge is not limited to the case base, but also comprises vocabulary, similarity measures, and adaptation knowledge [28]. Case knowledge may remain local because it originates close to the assets and may contain sensitive operational information. Vocabulary may differ across nodes and is required to interpret local assets, services, and data models consistently, while a cloud vocabulary may only contain a higher-level view. Similarity measures may depend on machine type, sensor setup, or locally available features. Adaptation knowledge may be tied to local process constraints, available resources, or permitted actions.

Consequences for the CBR Cycle. The placement of knowledge containers directly constrains the distribution of the CBR cycle [1]. If case knowledge is distributed, retrieval and retention must account for access to distributed

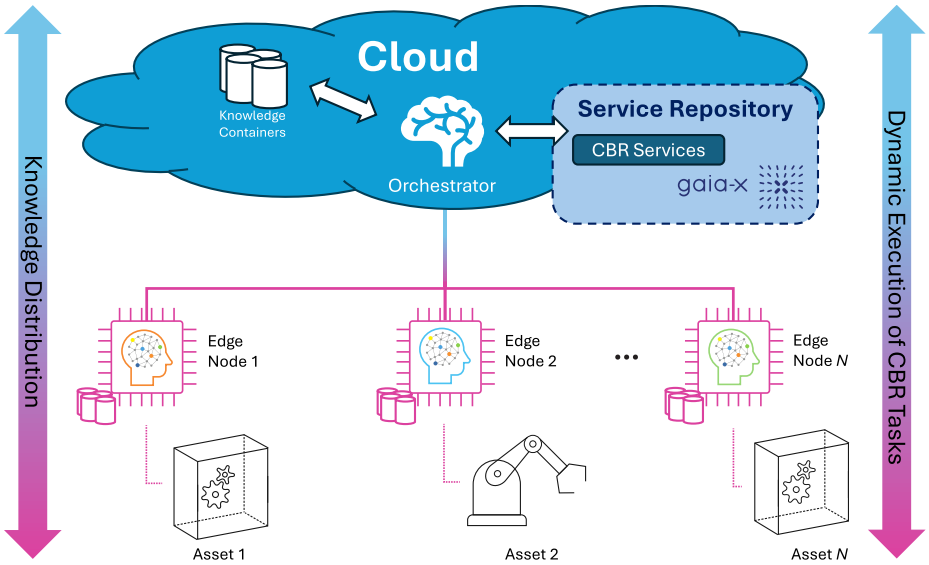


Fig. 2. Architecture for Application of CBR in the ECC, Based on *EASY* [30].

case bases. If similarity knowledge is available only on specific nodes, similarity assessment must be executed there unless the model can be transferred. Likewise, reuse must account for access to the relevant distributed adaptation knowledge. In industrial scenarios, a machine-related diagnostic request may be handled locally on the edge, forwarded to specialized nodes, or coordinated between nodes if knowledge is partitioned by machine type, process step, or site. Such interaction can support decentralized discovery and coordination of distributed CBR tasks, but distributed case bases and similarity knowledge require additional coordination, for example, to exchange cases, candidates, or similarity models.

Orchestrator. To operationalize these variants, the cloud requires an orchestration component that acts as the load balancer for distributed CBR and decides where CBR tasks are executed and which knowledge sources are involved. As indicated in Fig. 2, the service repository provides available CBR services, whereas the orchestrator performs placement decisions based on metrics and constraints such as resource availability, transfer costs, latency, and ownership restrictions [30]. In the *EASY* context, this orchestration should rely on semantically described services, resources, and interfaces and remain compatible with Gaia-X-aligned requirements for interoperable and sovereign service and data exchange [37].

Consequences for Knowledge Access. This separates central discoverability from central knowledge storage. A node may keep local knowledge containers under its control while still exposing CBR functionality or selected metadata to the continuum. In this way, the ECC supports distributed CBR execution and knowledge use while retaining flexible local, cooperative, or centralized processing.

4.3 Opportunities and Challenges of the Approach

The proposed use of CBR in the ECC offers several opportunities. First, requests can be processed close to the data source if relevant cases and knowledge containers are available locally. This can reduce latency and data transfer in industrial diagnostic and monitoring scenarios. Second, heterogeneous resources can be used more efficiently because compute-intensive steps can be shifted to more powerful nodes, while lightweight or latency-critical steps remain local. Third, distribution can support scalability and robustness because requests can be processed in parallel, specialized case bases can be queried selectively, and failures of individual nodes can be mitigated by rerouting requests or offloading tasks [30].

At the same time, distribution introduces challenges. Resource-constrained edge devices may not execute all CBR tasks efficiently, and communication, coordination, and aggregation can increase response times. Distributed vocabularies, cases, similarity measures, or adaptation knowledge also have to remain sufficiently consistent to support reliable reasoning. This is especially relevant when knowledge is updated locally, heterogeneous data representations are used, or results from several nodes have to be combined. Finally, distributed CBR in the ECC requires resource-aware scheduling, compatible service descriptions,

shared interfaces, and sufficiently aligned semantic models for assets, services, and data [30].

4.4 Further Potentials

A further perspective arises when the ECC is considered across organizational boundaries. The general *EASY* architecture [30] already points in this direction, since edge nodes may belong to different industrial actors, while a central cloud and repository support orchestration and service discovery. For CBR, knowledge does not have to be centralized even if requests can be handled across the continuum. Organizations may keep cases and further knowledge containers under local control while exposing only selected CBR services or restricted metadata. This is attractive in industrial ecosystems in which machine operators, suppliers, and external service providers contribute complementary experience without disclosing internal data or processes.

A representative example is a service scenario in which machine-related experience is documented in local shift books and case bases, while one production shift is supported by an external service provider. The provider may require problem-solving knowledge and should also contribute new observations and solutions, while the operating company may not want to reveal the full process context, configurations, or all case attributes. Cross-organizational CBR in the ECC should therefore allow selective knowledge sharing, for example, by exposing fault descriptions, similarity-relevant features, or validated solution fragments. Requests can then be forwarded to specialized nodes without requiring the requesting actor to know where the relevant cases are stored. If organizations use different case structures or terminologies, semantic alignment becomes essential. Shared vocabularies and semantically described assets and services, for example based on AASs [41], can relate heterogeneous knowledge representations. At the same time, cross-organizational CBR depends on interoperability, access control, and data-sovereign discoverability mechanisms, since even metadata or index entries may be sensitive [30].

5 Implementation and Feasibility Test

This section presents the prototype-based realization of the proposed approach for CBR in the ECC and an initial feasibility test. Sect. 5.1 introduces the technical requirements and the resulting ProCAKE-based load balancer. Sect. 5.2 then reports on an initial runtime-oriented feasibility test.

5.1 Technical Requirements and Implementation

To move from the conceptual architecture to an executable prototype, technical requirements for using CBR in the ECC were derived in *EASY* from the use cases of the involved research and industry partners. For distributed CBR, ProCAKE had to run as separate instances on edge and cloud nodes, remain

compatible with an orchestration component, be deployable in the Kubernetes[6] and KubeEdge[7] runtime environment, rely on uniform and serializable exchange formats, and allow edge instances to continue ongoing CBR tasks in partially disconnected settings. Kubernetes provides an open-source platform for deploying, scaling, and managing containerized applications, while KubeEdge extends it to edge environments by supporting deployment, networking, and metadata synchronization between cloud and edge nodes.

As distributed CBR is only rarely supported explicitly by existing CBR frameworks [34], ProCAKE [5, 6] provided a suitable basis for an extension towards the ECC. This is because ProCAKE already covers several targeted application scenarios and supports the required CBR types, such as temporal and process-oriented CBR. This choice implied additional requirements for edge devices, namely Linux compatibility, sufficient memory and storage capacities for KubeEdge, communication capabilities, and enough local compute resources for the framework and its components.

ProCAKE was integrated into the service-based runtime environment by exposing its functionality through the existing REST interface, describing it semantically using AASs in the project context, and encapsulating it in a Docker container deployable in Kubernetes and KubeEdge. Thus, ProCAKE became discoverable and deployable as a CBR service in the ECC. The resulting ProCAKE load balancer was published[8].

5.2 Feasibility Test

To obtain initial evidence on the practical applicability of the prototype, an initial feasibility test was conducted for a distributed case base scenario. It focuses on cumulative runtime under different placement strategies for this scenario, covering only part of the proposed architecture and the metric perspective introduced in Sect. 4.1. Since the prototype currently exposes only a subset of metrics as system-accessible routing criteria, the feasibility test relies on CPU usage, permanent storage, and energy efficiency classes derived from the energy efficiency factor. Runtime serves as the observed outcome for comparing the resulting placement decisions. The heterogeneous hardware setup comprises 1) a cloud server with 50 CPU cores at 2850 MHz, 1 TB RAM, and 196 GB storage, 2) a Jetson-based edge node with 6 CPU cores at 115–1510 MHz, 6 GB RAM, and 100 GB storage, and 3) a Raspberry-Pi-based edge node with 4 CPU cores at 2400 MHz, 15 GB RAM, and 64 GB storage.

Three freely available case bases were considered to cover simple attribute-based cases, workflow-like cases, and smart-factory time series cases. The used-car case base is structurally flat and contains more than 100,000 cases. The recipes case base contains 40 complex workflow cases [21] and has been used in the context of combining AI planning and CBR [17]. The third case base contains 4,811 cases

[6] `https://kubernetes.io/docs/concepts/overview/`
[7] `https://kubeedge.io/`
[8] `https://gitlab.rlp.net/procake/procake-load-balancer`

from a smart factory scenario, where each case comprises ten time series and additional meta-attributes [18, 42]. For each case base, multiple CBR requests were issued over a longer experiment session. The orchestrator distributed the requests according to four placement strategies: even distribution and metric-based strategies derived from CPU usage, permanent storage, and energy efficiency classes. In addition, pure cloud execution without edge distribution was considered as a centralized reference.

The resulting runtimes are shown in Tab. 2. Across the evaluated settings, the centralized cloud-only reference yields the lowest cumulative runtime for all three case bases. This suggests that distribution overhead and slower computation on edge nodes may outweigh the benefits of processing cases closer to the edge in these scenarios. Their individual contributions are not measured separately. Within the distributed strategies, the resource-aware variants based on CPU usage and permanent storage substantially reduce cumulative runtime compared to even distribution and the energy-efficiency-based strategy. Overall, the results show that distributed CBR in the ECC is technically feasible but not necessarily advantageous from a pure runtime perspective. At the same time, runtime is not the only relevant decision criterion. Privacy, data sovereignty, or ownership restrictions may prohibit transferring cases or other knowledge containers and can therefore act as exclusion criteria for an otherwise runtime-efficient centralized strategy [11, 30].

6 Conclusion and Future Work

This paper investigates the application of CBR in the ECC. Relevant industrial application scenarios were collected and structured based on project experience and exchanges with industrial partners. Building on this motivation, an approach for distributed CBR in the ECC was presented. It comprises load balancing metrics for placement decisions, an architecture-oriented discussion of how CBR knowledge containers and phases can be distributed across edge and cloud, and a discussion of opportunities, challenges, and cross-organizational potentials. To demonstrate technical feasibility, a ProCAKE-based prototype was integrated into the ECC runtime environment. The feasibility test shows that DCBR in the ECC is technically possible, but also that distributed execution is not necessarily

Table 2. Runtime of the Feasibility Test Across the Three Case Bases for Metric-Based Placement and Reference Strategies (Values in HH:MM:SS).

Metric / Strategy	Used Cars	Recipes	Time Series
Even Distribution	3:37:16	190:53:31	394:07:10
CPU Usage	2:12:02	32:37:52	179:24:26
Permanent Storage	1:49:44	36:04:31	98:14:43
Energy Efficiency Class	3:58:50	86:29:32	407:44:19
Cloud (centralized)	0:15:56	30:06:52	91:09:44

advantageous regarding runtime. This initial runtime-oriented test does not yet cover the broader metric perspective and application diversity discussed in this paper.

Future work should extend the current prototype beyond the scenario of a distributed case base and investigate additional forms of distributed CBR in the ECC. This includes the separate and combined distribution of vocabulary, similarity knowledge, adaptation knowledge, and CBR phases, as well as an up-to-date, comprehensive literature review of DCBR. The interactions between these decisions require closer study, since knowledge placement directly constrains where CBR tasks can be executed efficiently and consistently. Moreover, the cross-organizational perspective should be explored regarding controlled knowledge sharing, semantic interoperability, and data-sovereign access to DCBR services, including cross-organizational case distribution in data spaces, understood as governed environments for data sharing, in the follow-up research project Factory-X[9]. A further important direction is the use of AASs as a shared vocabulary and semantic grounding for manufacturing-oriented I4.0 applications. In such a setting, the current AAS of a machine could be integrated into the query and case representation, allowing cases to be interpreted as episodes in the life of a machine. This would support identifying comparable machines, analyzing whether similar faults have occurred on equivalent assets, and relating past cases to the current machine state. This could also reduce modeling effort for industrial users if compatible AASs are already provided by machine suppliers. More generally, such a shared structure could help address the compatibility of distributed knowledge representations and the transfer of knowledge across nodes or organizations. Finally, future evaluations should broaden the current runtime-oriented perspective and align the assessment of CBR in the ECC with broader evaluation concepts for edge-cloud systems [2,14]. This includes quantified criteria such as resource usage, energy demand, and data transfer, as well as use-case-specific trade-offs and non-performance-related exclusion criteria such as privacy, trust, and data sovereignty.

Acknowledgments. This work is funded by the Federal Ministry of Research, Technology and Space under grant No. 01MD22002C *EASY* [30]. Further, this work was partially funded by the Federal Ministry for Economic Affairs and Energy under grant No. 13MX001F *Factory-X*.

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Commun. **7**(1), 39–59 (1994). `https://doi.org/10.3233/AIC-1994-7104`
2. Alt, B., Bast, S., Babel, M., Buitmann, J., Creutz, L., Dartmann, G., Ehaus, M., Engel, T., Gast, F., Großegesse, N., Guldner, A., Jahnke, N., Jilg, D., Körner, M.F., Leskow, J., Mundorf, J., Naumann, S., Pehlken, A., Schacht, M., Schick, L.,

[9] `https://factory-x.org/`

Schlagenhauf, T., Schultheis, A., Schörner, K., Straub, S., Strüker, J., Traphöner, R., Weiher, M.A.: Evaluation of Edge-Cloud Systems: Approaches, tools and metrics (2025). https://doi.org/10.5281/zenodo.17233607, Begleitforschung Edge Datenwirtschaft.
3. Angel, N.A., Ravindran, D., Vincent, P.M.D.R., Srinivasan, K., Hu, Y.C.: Recent Advances in Evolving Computing Paradigms: Cloud, Edge, and Fog Technologies. Sensors (Basel, Switzerland) **22**(1) (2021). https://doi.org/10.3390/s22010196
4. Bergmann, R.: Experience Management: Foundations, Development Methodology, and Internet-Based Applications, LNCS, vol. 2432. Springer (2002). https://doi.org/10.1007/3-540-45759-3
5. Bergmann, R., Grewenig, L., Hoffmann, M., Jung, L., Malburg, L., Schultheis, A., Zeyen, C.: ProCAKE Framework (8.0.0) (2026). https://doi.org/10.5281/zenodo.20524277
6. Bergmann, R., Grumbach, L., Malburg, L., Zeyen, C.: ProCAKE: A Process-Oriented Case-Based Reasoning Framework. In: 27th ICCBR Workshop Proc. vol. 2567, pp. 156–161. CEUR-WS.org (2019)
7. Cajas Ordóñez, S.A., Samanta, J., Suárez-Cetrulo, A.L., Carbajo, R.S.: Intelligent Edge Computing and Machine Learning: A Survey of Optimization and Applications. Future Internet **17**(9), 417 (2025). https://doi.org/10.3390/fi17090417
8. Eager, D.L., Lazowska, E.D., Zahorjan, J.: Adaptive Load Sharing in Homogeneous Distributed Systems. IEEE Trans. Software Eng. **12**(5), 662–675 (1986). https://doi.org/10.1109/TSE.1986.6312961
9. Gilchrist, A.: Industry 4.0: The Industrial Internet of Things. Apress (2016). https://doi.org/10.1007/978-1-4842-2047-4
10. Green, C.: Application of Theorem Proving to Problem Solving. In: 1st IJCAI. pp. 219–240. IJCAI'69, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1969)
11. Guldner, A., Bender, R., Calero, C., Fernando, G.S., Funke, M., Gröger, J., Hilty, L.M., Hörnschemeyer, J., Hoffmann, G., Junger, D., Kennes, T., Kreten, S., Lago, P., Mai, F., Malavolta, I., Murach, J., Obergöker, K., Schmidt, B., Tarara, A., Veaugh-Geiss, J.P.D., Weber, S., Westing, M., Wohlgemuth, V., Naumann, S.: Development and evaluation of a reference measurement model for assessing the resource and energy efficiency of software products and components - Green Software Measurement Model (GSMM). Future Gener. Comput. Syst. **155**, 402–418 (2024). https://doi.org/10.1016/J.FUTURE.2024.01.033
12. Haslum, P.: Admissible Heuristics for Automated Planning. Ph.D. thesis, Institutionen för datavetenskap (2006)
13. Jahnke, B.F., Schmook, R., Howar, F.: GMB-ECC: Guided Measuring and Benchmarking of the Edge Cloud Continuum. IEEE Access **14**, 64191–64201 (2026). https://doi.org/10.1109/ACCESS.2026.3685833
14. Jahnke, N., Großegesse, N., Jahnke, B.F., Jilg, D., Leskow, J., Schmook, R., Schultheis, A., Wang, J.: Toward a Holistic Evaluation of Edge-Cloud Systems: Insights from a Task Force. SSRN Electronic Journal pp. 1–16 (2025). https://doi.org/10.2139/ssrn.7132701, Preprint, posted on SSRN.
15. Kern, E., Hilty, L.M., Guldner, A., Maksimov, Y.V., Filler, A., Gröger, J., Naumann, S.: Sustainable software products – Towards assessment criteria for resource and energy efficiency. FGCS **86**, 199–210 (9 2018). https://doi.org/10.1016/j.future.2018.02.044
16. Khalyeyev, D., Bures, T., Hnetynka, P.: Towards Characterization of Edge-Cloud Continuum. In: ECSA 2022 Proc. LNCS, vol. 13928, pp. 215–230. Springer (2022). https://doi.org/10.1007/978-3-031-36889-9_16

17. Malburg, L., Brand, F., Bergmann, R.: Adaptive Management of Cyber-Physical Workflows by Means of Case-Based Reasoning and Automated Planning. In: EDOC 2022 Proceedings. LNBIP, vol. 466, pp. 79–95. Springer (2022). https://doi.org/10.1007/978-3-031-26886-1_5
18. Malburg, L., Schultheis, A., Bergmann, R.: Modeling and Using Complex IoT Time Series Data in Case-Based Reasoning: From Application Scenarios to Implementations. In: 31st ICCBR Workshop Proc. vol. 3438, pp. 81–96 (2023)
19. Marrella, A.: What Automated Planning Can Do for Business Process Management. In: BPM 2017 Workshops. LNBIP, vol. 308, pp. 7–19. Springer (2017). https://doi.org/10.1007/978-3-319-74030-0_1
20. Milojicic, D.: The edge-to-cloud continuum. Computer **53**(11), 16–25 (2020). https://doi.org/10.1109/MC.2020.3007297
21. Minor, M., Montani, S., Recio-García, J.A.: Process-Oriented Case-Based Reasoning. Inf. Syst. **40**, 103–105 (2014). https://doi.org/10.1016/j.is.2013.06.004
22. Moreschini, S., Pecorelli, F., Li, X., Naz, S., Hästbacka, D., Taibi, D.: Cloud Continuum: the definition. IEEE Access **10**, 131876–131886 (2022). https://doi.org/10.1109/ACCESS.2022.3229185
23. Nkisi-Orji, I., Wiratunga, N., Palihawadana, C., Recio-García, J.A., Corsar, D.: Clood CBR: Towards Microservices Oriented Case-Based Reasoning. In: 28th ICCBR Proc. pp. 129–143. Springer (2020). https://doi.org/10.1007/978-3-030-58342-2_9
24. Noghabi, S.A., Cox, L., Agarawal, S., Ananthanarayanan, G.: The Emerging Landscape of Edge Computing. GetMobile: Mobile Computing and Communications **23**, 11–20 (2019). https://doi.org/10.1145/3400713.3400717
25. Pla, A., López, B., Gay, P., Pous, C.: eXiT*CBR.v2: Distributed Case-Based Reasoning Tool for Medical Prognosis. Decis. Support Syst. **54**(3), 1499–1510 (2013). https://doi.org/10.1016/j.dss.2012.12.033
26. Plaza, E., McGinty, L.: Distributed case-based reasoning. Knowl. Eng. Rev. **20**(3), 261–265 (2005). https://doi.org/10.1017/S0269888906000683
27. Redmond, M.: Distributed Cases for Case-Based Reasoning: Facilitating Use of Multiple Cases. In: AAAI-90 Proceedings. pp. 304–309. AAAI Press / The MIT Press (1990)
28. Richter, M.M.: Knowledge Containers. In: Readings in Case-Based Reasoning. Morgan Kaufmann Publishers (2003)
29. Schultheis, A.: Exploring a Hybrid Case-Based Reasoning Approach for Time Series Adaptation in Predictive Maintenance. In: 32nd ICCBR Workshop Proc. vol. 3708, pp. 230–235. CEUR-WS.org (2024)
30. Schultheis, A., Alt, B., Bast, S., Guldner, A., Jilg, D., Katic, D., Mundorf, J., Schlagenhauf, T., Weber, S., Bergmann, R., Bergweiler, S., Creutz, L., Dartmann, G., Malburg, L., Naumann, S., Rezapour, M., Ruskowski, M.: EASY: Energy-Efficient Analysis and Control Processes in the Dynamic Edge-Cloud Continuum for Industrial Manufacturing. Künstliche Intelligenz **39**, 161–166 (2024). https://doi.org/10.1007/s13218-024-00868-3
31. Schultheis, A., Bertrand, Y., Grüger, J., Malburg, L., Bergmann, R., Serral Asensio, E.: Case-based Data Quality Management for IoT Logs: A Case Study Focusing on Detection of Data Quality Issues. IoT **6**(4) (2025). https://doi.org/10.3390/iot6040063
32. Schultheis, A., Jilg, D., Malburg, L., Bergweiler, S., Bergmann, R.: Towards Flexible Control of Production Processes: A Requirements Analysis for Adaptive Workflow Management and Evaluation of Suitable Process Modeling Languages. Processes **12**(12) (2024). https://doi.org/10.3390/pr12122714

33. Schultheis, A., Späth, M., Weich, J., Bergmann, R.: Towards Explainable and Reusable Temporal Case-Based Reasoning in Predictive Maintenance: A Research Environment. In: 44th SGAI Proc. LNCS, Springer (2026). https://doi.org/10.1007/978-3-032-45236-8_17
34. Schultheis, A., Zeyen, C., Bergmann, R.: An Overview and Comparison of Case-Based Reasoning Frameworks. In: 31st ICCBR Proc. LNCS, vol. 14141, pp. 327–343. Springer (2023). https://doi.org/10.1007/978-3-031-40177-0_21
35. Seiger, R., Schultheis, A., Bergmann, R.: Case-Based Activity Detection from Segmented Internet of Things Data. In: 33rd ICCBR Proc. LNCS, vol. 15662, pp. 438–453. Springer (2025). https://doi.org/10.1007/978-3-031-96559-3_29
36. Strnadl, C.F.: End-to-End-Architekturen zur Datenmonetarisierung im Industrial Internet of Things (IIoT). In: Monetarisierung von technischen Daten, pp. 169–206. Springer Berlin Heidelberg, Berlin, Heidelberg (2021). https://doi.org/10.1007/978-3-662-62915-4_10
37. Tardieu, H.: Role of Gaia-X in the European Data Space Ecosystem. In: Designing Data Spaces: The Ecosystem Approach to Competitive Advantage, pp. 41–59. Springer (2022). https://doi.org/10.1007/978-3-030-93975-5_4
38. Van Steen, M., Tanenbaum, A.S.: Distributed Systems. Prentice Hall (2017)
39. Varghese, B., de Lara, E., Ding, A.Y., Hong, C.H., Bonomi, F., Dustdar, S., Harvey, P., Hewkin, P., Shi, W., Thiele, M., Willis, P.: Revisiting the Arguments for Edge Computing Research. IEEE Internet Computing **25**(5), 36–42 (2021). https://doi.org/10.1109/MIC.2021.3093924
40. Weber, S., Guldner, A., Fazlic, L.B., Dartmann, G., Naumann, S.: Sustainability in Artificial Intelligence – Towards a Green AI Reference Model. In: Künstliche Intelligenz in der Umweltinformatik. LNI, GI (2023). https://doi.org/10.18420/inf2023_154
41. Wei, K., Sun, J., Liu, R.: A Review of Asset Administration Shell. In: 2019 IEEM. pp. 1460–1465. IEEE (2019). https://doi.org/10.1109/IEEM44572.2019.8978536
42. Weich, J., Schultheis, A., Hoffmann, M., Bergmann, R.: Integration of Time Series Embedding for Efficient Retrieval in Case-Based Reasoning. In: 33rd ICCBR Proc. LNCS, vol. 15662, pp. 328–344. Springer (2025). https://doi.org/10.1007/978-3-031-96559-3_22
43. Yang, C., Lan, S., Wang, L., Shen, W., Huang, G.G.Q.: Big Data Driven Edge-Cloud Collaboration Architecture for Cloud Manufacturing: A Software Defined Perspective. IEEE Access **8**, 45938–45950 (2020). https://doi.org/10.1109/ACCESS.2020.2977846