EDITOR: Schahram Dustdar, dustdar@infosys.tuwien.ac.at

DEPARTMENT: INTERNET OF THINGS, PEOPLE, AND PROCESSES

On Causality in Distributed Continuum Systems

Víctor Casamayor Pujol, Boris Sedlak, Praveen Kumar Donta, and Schahram Dustdar, *Vienna University of Technology, Vienna, 1040, Austria*

As distributed continuum systems (DCSs) are envisioned, they will have a massive impact on our future society. Hence, it is of utmost importance to ensure that their impact is socially responsible. Equipping these systems with causal models brings features such as explainability, accountability, and auditability, which are needed to provide the right level of trust. Furthermore, by combining causality with graph-based service-level objectives, we can cope with dynamic and complex system requirements while achieving sustainable development of DCSs' capacities and applications.

DCSs: NOVEL CHALLENGES

Distributed systems, specifically cloud computing, support many of our daily activities, spanning from the home to the workspace and including many facets of our social life and entertainment activities. These services have become ubiquitous due to their business model, performance, and security guarantees. Computing infrastructure centralization, i.e., the data center, has been a critical enabler for this technology, providing an efficient and scalable solution to manage worldwide-scale services.

Interestingly, the upcoming generations of Internet-distributed systems are moving away from these large and isolated data centers to be closer to the users at the *edge* of the network. This trend is justified by several reasons, such as reducing network congestion, reducing service latency, or improving privacy guarantees. The list is long, and many scientific publications emphasize this change, e.g., Shi et al.[1] and Satyanarayanan.[2] Furthermore, moving computational resources closer to the user brings new applications for our society, such as autonomous cars, individualized health tracking and assistance, optimized resource management, holographic communications, and many other opportunities.

The potential business opportunities are enormous. Hence, it is inevitable that a new generation of Internet-distributed systems, also known as *distributed continuum systems* (*DCSs*),[3] will eventually be a part of our societal and technological ecosystems. Figure 1 shows a high-level view of this type of system. At the top, there are the resources from cloud computing; farther down we find fog and edge computing resources, which are more constrained, and, finally, at the bottom, we can find Internet of Things (IoT) devices and specific applications. This figure shows devices and their interconnections, but it is fundamental to understanding that there is no centralized logic behind it. The cloud does not govern all the devices in the figure, but each subset of devices is responsible for its tasks, and they are related in a loosely coupled manner. Similarly, as services would do in a service-oriented architecture, that is the reason behind using bidirectional arrows in Figure 1.

The technological challenges for these systems are manifold. DCSs are composed of a large diversity of devices and interconnections, which belong to different providers and are mutual to several tenants. Given the open and shared environment these systems inhabit, they need to adapt dynamically to changes while keeping strict constraints fulfilled in terms of performance, quality of service (QoS), or cost. There exist remarkable similarities between DCSs and natural ecosystems when their behaviors need to be explained, and both are self-adaptive systems built from self-adaptive components. We are studying the technological challenges presented by these new systems.[4] We are confident that, sooner rather than later, all the challenges identified will be surmounted and these systems will become a reality.

DCS applications will have many interactions with people and a significant impact on society. Hence, it is

1089-7801 © 2024 IEEE
Digital Object Identifier 10.1109/MIC.2023.3344248
Date of current version 16 April 2024.

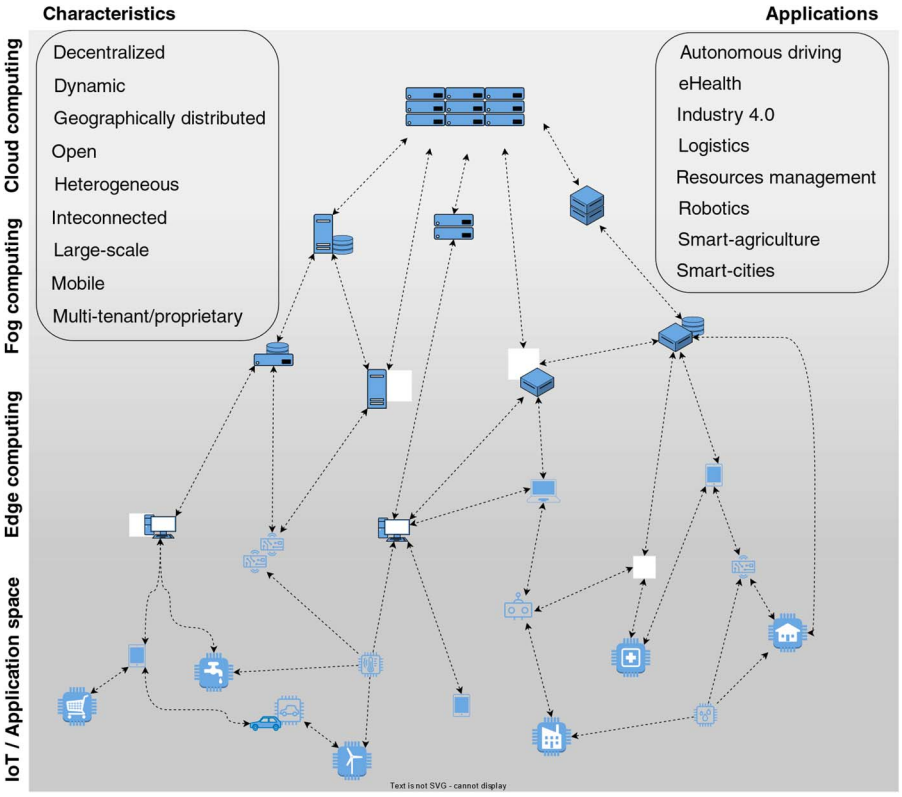

FIGURE 1. General overview of a distributed computing continuum system. IoT: the Internet of Things.

required that their complex behaviors be explainable, fair, and auditable. Causality is a technique that is able to offer this by composing models that explain each of the system's components. In this article, we start motivating causality from a bird's-eye view and take a closer look at how we envision its applicability in DCSs.

Top-Level View on Causality

At the highest conceptual level, causality aims at answering the why question. This question is rooted in human nature: our inherent way of comprehending how everything works. From this view, including causal models in new developments enables the capacity to understand the underlying reasons for their behaviors. This may be seen as an overhead, i.e., one just needs the system to do its job. However, this opens the door to conscious, open, and fair development of new systems. We claim that causality has to be the driving force for sustainable development of DCSs.

This is in contrast to having business growth as the main driving force. We have witnessed how cloud computing and big data analytics have allowed companies to provide new and exciting free services to users: e-mail, storage, social networks, and so on. However, these services only looked free because their users had become valuable products for these companies.[5] Similarly, machine learning (ML) and artificial intelligence (AI) have experienced exponential growth for the last one or two decades. Although all new capacities are discovered and enhanced through extensive training, many inconvenient aspects of this technology, such as a lack of trustworthiness[6] or being resource devouring,[7] have only emerged since ML/AI have become essential in our society. Fortunately, we are now aware of these perils, and legislation is slowly but steadily trying to solve these deficiencies. Our lesson learned is that business growth cannot be the main driving force for future DCSs.

Causality, as an overarching technique for DCS, can bring the necessary mechanisms to grasp their behavior and footprint fully. Sustainable development is not only about minimizing energy consumption, it is also about developing a holistic view of how these new systems interact with our society, and their impact. Causality is a cornerstone of DCSs' sustainable development as it accompanies concepts such as fairness, trustworthiness, resource efficiency, environmental responsibility, and so on.

CAUSALITY

In general, causality studies the cause and effect relationship between events and variables. A key difference with previous statistical analyses is that causality disconnects from spurious correlations. Hence, it looks for meaningful relationships between events and variables and provides methods for modeling them.

Causal models can be leveraged at three levels, or rungs, as explained in Pearl and Mackenzie.[8] The first rung is observational, allowing inference on a system that cannot be interacted with, and the only available data are from observations. The second rung is interventional; in this case, the system can be interacted with, and the data obtained reflect this possible interaction. This offers the possibility to make inferences on the system's behavior after performing changes on it. Finally, the third rung is counter factual, which aims to build models that can infer possibilities that cannot happen, such as, *What would have happened to the system if I had done action x instead of y*? Indeed, obtaining data and models for this type of query is challenging, but it is valuable reasoning to understand how a system can or cannot work. Further, it is a type of reasoning very familiar to us (people) as we commonly imagine situations and their possible outcomes (POs) before or after they have happened.

Embedding causal mechanisms within DCSs ensures human-interpretable backdoors to achieve explainability, accountability, and auditability, among the other benefits that this technology, such as parameter optimization, can bring to DCSs. As DCSs must be self-adaptive, they require knowledge about themselves, i.e., system knowledge, and the environment to which they need to adapt, i.e., context awareness. Crucially, by using causal-based techniques, it is possible to bring a third leg to this autonomy that considers its relationship with society, bringing the opportunity to become socially responsible and sustainable, i.e., societal responsibility (see Figure 2).

It is important to emphasize that causality is not merely a buzzword or trendy concept that we are introducing into distributed systems. It is a fundamental principle that is essential to understand to ensure the development of these new systems. In Figure 2, we can see all the features causality can provide to DCSs. Furthermore, it has been used in computer science for some time now. However, in most cases, it is used to solve specific problems for a specific domain and has never been used as the driving force for a new development.

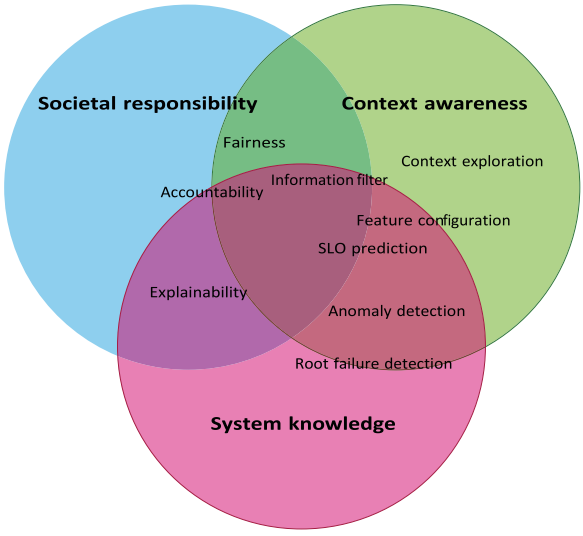

FIGURE 2. Causality features framed over the three main pillars to build DCSs. SLO: service-level objective.

Causality in Computer Science

In this section, we provide an overview of the use of causality in computer science to shed light on the possibilities of this technology for DCSs. Causality has been used in computer science for anomaly detection in distributed cloud-based systems. The complex relationships among services have required the development of fine-grained causal inference techniques to detect the root cause of failures or service-level objective (SLO) violations.[9] Indeed, this complexity is increased in DCSs. However, the need to detect root-cause failures and SLO violations remains the same for these systems as an expansion of the computing units and services toward the network's edge.

Configurable software systems have also benefited from causality to better understand how system features relate among them and how the system performance is affected by proper configuration. It has been shown that causal relations help identify responsibilities and reasons for features and feature combinations, paving the way for future tailored optimized configurations. Moreover, this also offers the possibility to perform analysis on counter-factual configurations, enabling reasoning over different contexts, an utmost feature for self-adaptive systems.[10] DCS are composed of a diversity of devices and connections. Hence, beyond horizontal or vertical scaling, properly configuring their usage is crucial to offering the expected QoSs.

Recommendation systems are another field where causality is starting to play a major role. These systems

can be understood as the primary information filters on the Internet, as they aim at providing only relevant information for its users. Hence, it is crucial for these systems to differentiate spurious correlations from real causal relations. In this regard, causality is essential to their inference models.[11] The volume of data that DCSs generate is huge; beyond the data related to the specific application, there is also a vast amount of data internal to the system, known as *big data*.[12] Hence, filtering these data for system management requires timely and effective processing and understanding.

The ubiquitousness of AI- and ML-inferring systems raises the need to find methods to explain these systems' outcomes; otherwise, it is unknown how much we can trust them. In that regard, causality is being explored to provide those systems with features such as interpretability, explainability, accountability, and fairness. In general, the main goal is to make these systems trustworthy and fair.[13] Recently, several works have started to study the capacity of causal reasoning for large language models (LLMs).[14] In brief, some causal queries are properly addressed, however, future research aims at incorporating causal models to provide this capacity to LLMs. This shows the relevance of causal reasoning in how we interact with anything. DCSs host sensitive applications, e.g., autonomous driving or e-health. Hence, explainability, fairness, and accountability are mandatory characteristics; otherwise, these applications will need more trustworthiness to be acceptable.

Many computer science fields use causality to boost or provide these missing functionalities to their state-of-the-art techniques. We opt for making causality a fundamental pillar for DCSs as it has the potential to revolutionize them by bringing capacities such as root failure and anomaly detection, SLO violation prediction, feature analysis and configuration, context exploration, information filtering, and decision explainability, while at the same time ensuring fairness and accountability.

Causally Enabled DCSs

Now our main focus is to identify the decisive elements from DCSs that are able to leverage causal models for seamless integration of all the benefits. In general, DCSs comprise cloud infrastructure, several fog nodes, more edge nodes, and many IoT devices, as shown in Figure 1. Their necessary functionalities are spread along the continuum. However, their requirements tend to be more specific to the computing tier. Hence, we need a mechanism that links functionalities and requirements, one that is aware of the specific idiosyncrasies of the component and can embed the causal models. By the end of the "Control and Management Subsystems" section, we present the artifact able to do this.

Subsystem-Based View of DCSs

Providing a modular view of the system's necessary functionalities eases the understanding of where causal features are required. Further, we comment on how requirements affect functionalities according to the computer tier in which the functionality sits. Hence, the following is organized into subsystems,[a] given that they represent system-specific functionalities. In short, any DCSs have the following subsystems: hardware, data, analytics, control, management, and network. Interestingly, in a DCS, there can be hardware components that are also a part of the payload, however, this is outside the scope of this article. Further, there can be other cross-cutting concerns, such as security, transversal to all others, which we understand as if each subsystem requires a *module* on security.

The hardware subsystem aims to manage the hardware components of the system. In that regard, it becomes more relevant in DCSs for two reasons. On the one hand, edge or IoT hardware has power constraints and mobility capacities and is geographically distributed. These aspects challenge the hardware requirements of any previous cloud-based system. On the other hand, future computing systems must be sustainable. Hence, most of the required considerations must be applied in this subsystem.

Hardware subsystem functionalities span from monitoring and tracking the health and performance of the hardware components to ensuring their desired behavior by performing adequate maintenance, including the required firmware updates and component substitutions. As a result, for this subsystem, it is crucial to incorporate anomaly detection and root-cause failure identification from causal methods. This would therefore enable fast and precise action at the hardware level with a high degree of accountability for the actions taken.

The data subsystem is in charge of handling data through its lifecycle, i.e., generation, (pre-)processing, storage, distribution, consumption, and deletion. Indeed, the specific needs of the application, combined with its location along the continuum, affects the requirements for each phase. Further, aspects such as data gravity and friction require special care in DCSs,[15] where

[a]They are also known as planes or layers, but within a system's context, we found subsystems a more appropriate wording.

data may have to be moved through jurisdictional boundaries.

The two main causal features required for this subsystem are 1) context awareness, for defining the specific policies that have to affect the data, and related to that, 2) system accountability for the decisions made, given that they are sensitive.

The analytics subsystem collects system metrics and performs analytics to support the other subsystems. Indeed, the requirements for this subsystem are very different at the edge than at the cloud; however, the expected functionalities are very similar. Simply put, the edge requires lightweight and fast analytics, while cloud requirements can lead to cost-effective requirements. Indeed, performing analytics includes forecasting metric trends and SLO violations and providing optimization possibilities for system configurations.

In that regard, this is a crucial subsystem for applying causal methods. Causality can provide information filters to process only relevant features, i.e., causally dependent. Further, system configuration capabilities can be explored considering the context-awareness capability of causality, and moreover, it can use interventional and counter-factual reasoning to optimize the configuration options of the system. This subsystem must also have explainability, fairness, and accountability features derived from the use of causal models.

Control and management subsystems subsystems need conceptually similar mechanisms related to the causal features that can be used. However, in terms of functionalities, they have different perspectives on the system. The control subsystem is responsible for the lower-level needs of the system, typically in short timescales. Hence, it takes care of local tasks that require a fast response. As an example, the access control of resources can be managed by the control subsystem. On the contrary, the management subsystem takes a higher-level perspective on the system: it has larger timescales and works toward global tasks that have slower paces. Hence, provisioning or updating a computing edge cluster can be a part of the management subsystem's tasks. Similarly, as in previous subsystems, specific requirements for the functionalities are linked to their location in the continuum.

These subsystems require input from the system's analytics to act accordingly. In that regard, it is fundamental that they have causal models able to offer explanations for the decisions made. Further, in a more technical view, these systems can also benefit from feature configuration and counter-factual analysis to align the information obtained from the analytics to its specific domain of action. They therefore need to be able to track previous decisions using the input from the analytics subsystem.

The network subsystem is in charge of managing the communication layer of DCSs. It takes control over specific functions of the communication network. For instance, it must ensure a certain level of throughput while having a dynamic number of requests. However, the techniques and resources vary depending on the computational tier. Generally, in cloud tiers, there are fast and reliable wired connections, while toward the edge, wireless and low-power communication is more typical.

The network subsystem requires to causally model its connectivity efficiency to account for the overall system performance. Furthermore, similarly to the previous subsystems, it also needs the feature configuration and counter-factual analysis features brought by causality.

CAUSALITY INTEGRATED IN DCS

A holistic integration of causality in DCSs is needed to harness their benefits; many of their features are required in all subsystems as key enablers for DCSs. From our experience in distributed systems, an enhanced version of SLOs can be the missing artifact to link causality with distributed systems. Functionalities need to be steered by requirements to be properly adapted to the computing continuum (CC) environment (see Figure 3). Further, DCSs are service oriented; therefore, SLOs can describe these functionalities while determining a specific requirement for them. It is worth noting that SLOs are commonly used in cloud computing systems, however, our enhanced version would reflect all the needs that services throughout the CC may have. This implies that SLOs reflect lower-level system needs, e.g., the maximum CPU usage of a

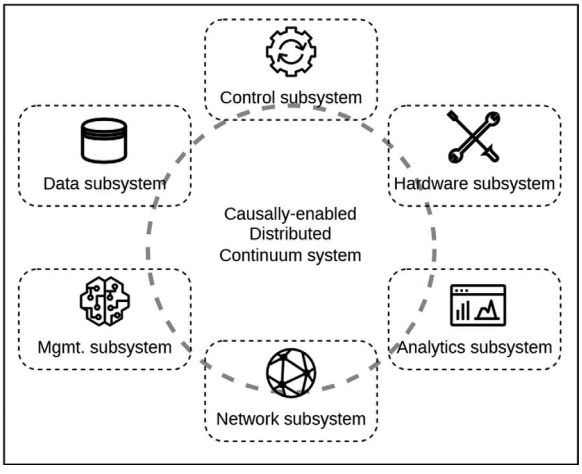

FIGURE 3. Subsystem view for distributed continuum systems.

device as well as higher-level application needs, e.g., the minimal accuracy for a medical-related inference, given that both relate to service requirements. Interestingly, DCSs are highly dependent on their underlying infrastructure; hence, accuracy may be as a result of the ML model as well as the sensor data's granularity.

Enhanced SLOs are built as causal graphs, where the leaf node is the SLO's compliance value, and its parents and grandparents are variables that causally influence the SLO's behavior. Moreover, these variables that influence the SLO's compliance can be parameterized. Looking at the accuracy example, the frames per second of a camera influences the SLO's compliance, but it can be adjusted to different rates. This implies that it is also possible to work at the interventional or counterfactual rungs with causally enhanced SLOs. As shown in Figure 4, an SLO is a causality graph where the values of its variables explain its behavior. These variables can also be deliberately modified, providing a framework for applying do-calculus and counterfactual reasoning in distributed systems. Further, these variables or parameters can relate to different SLOs, providing the capacity to explore the system's behavior beyond a single SLO.

Currently, there are two main frameworks for modeling causality: structural causal models (SCMs),[16] based on causal graphs and structural equations, and the POs framework, which inherits from the development of randomized experiments (see Yao et al.[17] for a comprehensive survey). From our perspective, SCMs better suit the needs of DCSs, given that the graph view and its equations can easily model DCSs. Causal graphs can be expressed in terms of SCMs, where each parent of the SLO (*Pa*) together with a set of exogenous (nonobservable) variables (U) define the probability of SLO compliance [*P*(SLO)].

$$P(\text{SLO}) \leftarrow f_i(Pa_i, U_i). \tag{1}$$

Consider that f is a function that relates a parent and the probability distribution of U to the probability distribution of SLO compliance. Also, the subindex on the right side of the equation accounts for different parents and exogenous variables, explaining that each can have a different relationship (i.e., function f) with the SLO. Hence, this formulation also offers the possibility to study the sensitivity of SLO compliance to its variables.[18]

Modeling these systems through SLO-based causal graphs brings the following crucial benefits:

- Defining SLOs through causal graphs embeds an information filter for each SLO. Given that every time an SLO is required to be analyzed, e.g., to understand why its compliance value has changed, only the information of the variables in its causal graph has to be assessed, which is the minimal set of information. This also relates to and emphasizes the use of Markov's blanket concept.[3]

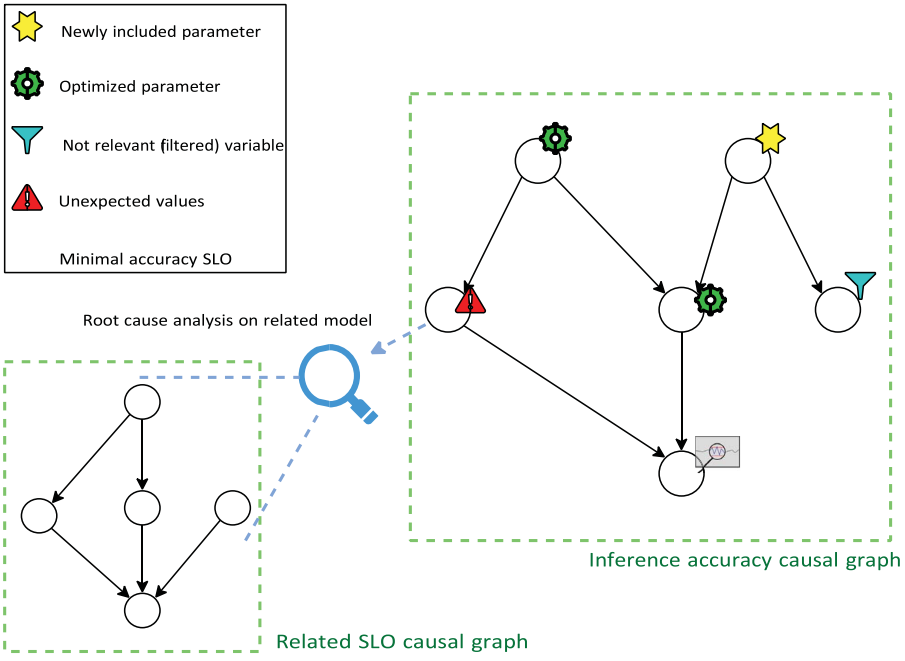

FIGURE 4. Causal graphs for an SLO-based designed system.

- Causal models provide context awareness, given that they are specific to a service in a determined environment. However, they can also be updated, which means that changes in the service context can be integrated into the causal model, which adapts it to its new reality.
- Exploring possible configurations of the system and their implications for the SLOs is possible when the configuration variables are included in the SLO-based causal graph. Hence, parameter explainability and optimization go together with the causal-enabled SLO.
- The granularity of causal models is linked to the described SLO. Further, it is expected to find variables in a causal graph that are also related to other SLOs, which may have a different level of granularity. Hence, by following these relations, root cause and anomalies can be detected regardless of the distance between the observed effect and the root cause.
- Causal graphical models are interpretable; hence, using them at the service-requirement level integrates the possibility of explaining and therefore accounting for the reasons for a service behavior.

Currently, we are making progress on defining SLOs for DCSs through a Bayesian network. Our next step is ensuring that the Bayesian network behaves as a causal graph so that we unfold all the benefits given by them.

CONCLUSION

The development of DCSs will have an enormous impact on our future society. Hence, they require embedding the capacity for being explainable, fair, accountable, and auditable. This is on top of all the other technical challenges that still need to be solved for these large-scale, heterogeneous, and dynamic systems. In this article, we motivated causality, in the form of causal graphs and SCMs, as the technique to be embraced by these systems to overcome all their technical challenges while also bringing these crucial capacities for being socially responsible. Further, we proposed their integration with SLOs to obtain this holistic framework for developing DCSs.

REFERENCES

1. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.
2. M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017, doi: 10.1109/MC.2017.9.
3. S. Dustdar, V. C. Pujol, and P. K. Donta, "On distributed computing continuum systems," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 4092–4105, Apr. 2023, doi: 10.1109/TKDE.2022.3142856.
4. V. Casamayor Pujol, P. K. Donta, A. Morichetta, I. Murturi, and S. Dustdar, "Edge intelligence—Research opportunities for distributed computing continuum systems," *IEEE Internet Comput.*, vol. 27, no. 4, pp. 53–74, Jul./Aug. 2023, doi: 10.1109/MIC.2023.3284693.
5. T. Morey, T. Forbath, and A. Schoop, "Customer data: Designing for transparency and trust," *Harvard Business Review*, May 2015. [Online]. Available: https://hbr.org/2015/05/customer-data-designing-for-transparency-and-trust
6. D. Kaur, S. Uslu, K. J. Rittichier, and A. Durresi, "Trustworthy artificial intelligence: A review," *ACM Comput. Surv.*, vol. 55, no. 2, pp. 39:1–39:38, Jan. 2022, doi: 10.1145/3491209.
7. E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" in *Proc. ACM Conf. Fairness, Accountability, Transparency*, 2021, pp. 610–623, doi: 10.1145/3442188.3445922.
8. J. Pearl and D. Mackenzie, *The Book of Why: The New Science of Cause and Effect*. New York, NY, USA: Basic Books, 2018.
9. P. Chen, Y. Qi, and D. Hou, "CauseInfer: Automated end-to-end performance diagnosis with hierarchical causality graph in cloud environment," *IEEE Trans. Services Comput.*, vol. 12, no. 2, pp. 214–230, Mar./Apr. 2019, doi: 10.1109/TSC.2016.2607739.
10. C. Dubslaff, K. Weis, C. Baier, and S. Apel, "Causality in configurable software systems," in *Proc. 44th Int. Conf. Softw. Eng. (ICSE)*, New York, NY, USA: ACM, Jul. 2022, pp. 325–337, doi: 10.1145/3510003.3510200.
11. C. Gao, Y. Zheng, W. Wang, F. Feng, X. He, and Y. Li, "Causal inference in recommender systems: A survey and future directions," Aug. 2022. [Online]. Available: http://arxiv.org/abs/2208.12397
12. P. K. Donta, B. Sedlak, V. C. Pujol, and S. Dustdar, "Governance and sustainability of distributed continuum systems: A big data approach," *J. Big Data*, vol. 10, no. 1, Apr. 2023, Art. no. 53, doi: 10.1186/s40537-023-00737-0.
13. N. Ganguly et al., "A review of the role of causality in developing trustworthy AI systems," Feb. 2023. [Online]. Available: http://arxiv.org/abs/2302.06975
14. C. Zhang et al., "Understanding causality with large language models: Feasibility and opportunities," Apr. 2023. [Online]. Available: http://arxiv.org/abs/2304.05524
15. B. Sedlak, V. C. Pujol, P. K. Donta, and S. Dustdar, "Controlling data gravity and data friction: From metrics to multidimensional elasticity strategies,"

in *Proc. IEEE Int. Conf. Softw. Services Eng. (SSE)*, Jul. 2023, pp. 43–49, doi: 10.1109/SSE60056.2023.00017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10234353

16. J. Pearl, "Causal inference in statistics: An overview," *Statist. Surv.*, vol. 3, pp. 96–146, Jan. 2009, doi: 10.1214/09-SS057. [Online]. Available: https://projecteuclid.org/journals/statistics-surveys/volume-3/issue-none/Causal-inference-in-statistics-An-overview/10.1214/09-SS057.full
17. L. Yao, Z. Chu, S. Li, Y. Li, J. Gao, and A. Zhang, "A survey on causal inference," *ACM Trans. Knowl. Discovery Data*, vol. 15, no. 5, pp. 74:1–74:46, May 2021, doi: 10.1145/3444944.
18. C. Cinelli, D. Kumor, B. Chen, J. Pearl, and E. Bareinboim, "Sensitivity analysis of linear structural causal models," in *Proc. 36th Int. Conf. Mach. Learn.*, PMLR, May 2019, pp. 1252–1261. [Online]. Available: https://proceedings.mlr.press/v97/cinelli19a.html

VÍCTOR CASAMAYOR PUJOL is a project assistant (postdoctoral researcher) with the Distributed Systems Group, Vienna University of Technology, Vienna, 1040, Austria. Contact him at v.casamayor@dsg.tuwien.ac.at.

BORIS SEDLAK is a Ph.D. candidate with the Distributed Systems Group, Vienna University of Technology, Vienna, 1040, Austria. Contact him at b.sedlak@dsg.tuwien.ac.at.

PRAVEEN KUMAR DONTA is a postdoctoral researcher with the Distributed Systems Group, Vienna University of Technology, Vienna, 1040, Austria. Contact him at p.donta@dsg.tuwien.ac.at.

SCHAHRAM DUSTDAR is a full professor of computer science and heads the Research Division of Distributed Systems, Vienna University of Technology, Vienna, 1040, Austria. Contact him at dustdar@dsg.tuwien.ac.at.