Container-based Data Pipelines on the Computing Continuum for Remote Patient Monitoring

Nikolay Nikolov
SINTEF AS / University of Oslo, Norway

Arnor Solberg
Tellu, Norway

Radu Prodan
University of Klagenfurt, Austria

Ahmet Soylu
Oslo Metropolitan University, Norway

Mihhail Matskin
Royal Institute of Technology, Sweden

Dumitru Roman
SINTEF AS, Norway

***Abstract*—Diagnosing, treatment, and follow-up care of patients is happening increasingly through telemedicine, especially in remote areas where direct interaction is hindered. Over the past three years, following the COVID-19 pandemic, the utility of remote patient care has been further field-tested. Tackling the technical challenges of a growing demand for telemedicine requires a convergence of several fields: 1) software solutions for reliable, secure, and reusable data processing, 2) management of hardware resources (at scale) on the Cloud/Fog/Edge Computing Continuum, and 3) automation of DevOps processes for deployment of digital healthcare solutions with patients. In this context, the emerging concept of *big data pipelines* provides relevant solutions and is one of the main enablers. In what follows, we present a data pipeline for remote patient monitoring and show a real-world example of how data pipelines help address the stringent requirements of telemedicine.**

1. Introduction

TELEMEDICINE, over recent years during the COVID-19 pandemic, has provided healthcare solutions to patients, municipalities, and hospitals with reduced costs and improved patient treatment [1]. Typical scenarios involve treatment and care for elderly people and for chronic diseases (e.g., diabetes, kidney, cancer) or quarantined (e.g., COVID-19) patients [2]. Examples of digital healthcare services include personal safety alarms, care phones, healthcare call centers, remote patient monitoring, and digital supervision.

The uptake of information technologies in Europe is the leading accelerator for telemedicine. The market potential of telemedicine was valued at USD 83.5 billion and is expected to grow at a compound annual growth rate of 24% in the coming years. Accordingly, the well-being market, enabled by digital technologies (mobile applications, sensing devices), is also rapidly growing[1].

Digital healthcare services control mainly two types of sensors gathering raw information from thousands of patients through tablets, mobile phones, sensors, or other devices [3]:

1) *remote supervision sensors* such as patient motion sensors and video cameras;
2) *care and wellness-specific sensors* monitoring specific patient data, such as blood pressure, scales, glucose, and medicine quantity.

Gathering data on behalf of thousands of patients implies data processing solutions for automated, private, and secure processing, including filtering, encryption, anonymization, and storage. The processes may take dynamic structures depending on their functionality, patient type (e.g., elderly, quarantined), disease, data acquisition device, data format (e.g., alarm, measurement, video stream), stakeholder (e.g., doctor, nurse, response center), or third-party integration (e.g., electronic health records).

The large number and variability of different telemedicine scenarios require organizing data processing into reusable and deployable units that can be easily managed across distributed settings. Big Data pipelines [4] are composite workflows for processing and communication of data with non-trivial properties and characteristics, referred to as the "Vs" of Big Data. Examples include volume, velocity, variety, veracity, validity, value, variability, etc. They represent data processes as sets of discrete and often independent steps or pipes that can be individually managed and reused.

In a distributed setting, modern frameworks for data pipelines enable the use of the so-called Cloud/Edge/Fog *Computing Continuum* [5]. The Computing Continuum extends traditional Cloud computing with emerging Edge and Fog computing paradigms, reducing overheads and compliance issues for transferring distributed data into remote data centers.

The contributions of this paper can be summarised as follows:

- We provide a generic example of remote patient monitoring use case and an outline of a set of data processing challenges, as well as how data pipelines can be used to address them;
- We describe a real-world industrial example of a remote patient monitoring data pipeline;
- We introduce the concept of container-based data pipelines in the domain of remote patient monitoring and illustrate how they can be used to address the challenges outlined.

2. Motivating use case: remote patient supervision

An example remote patient supervision scenario is depicted in Figure 1. A set of remote supervision or care and wellness sensor devices are installed in a patient's home, along with an edge device that connects them with the respective telecare service provider. The patients can then use the devices actively or passively to perform measurements related to their current health. The edge device sends these values (raw or pre-processed) to a centralized Cloud web service. The data are analyzed according to a pre-defined care or aftercare plan. If deviations occur, the central service may notify the care physician or the patient to take action or contact.

The data processing corresponding to this scenario can be generalized into three main stages, shown in Figure 2 (a):

1) *Data generation, pre-processing, and routing*: during this stage, the measurements gathered from remote supervision and wellness sensors are translated to signals, potentially pre-processed at the edge, and routed to the centralized service.
2) *Data storage and analytics*: in this stage, data are securely stored in the (centralized) Cloud of the service provider, and analytical services are performed to determine whether additional intervention is necessary by the patient or the medical doctor.
3) *End user application logic*: during this stage, based on the analytical results, specific ap-

[1] https://www.grandviewresearch.com/industry-analysis/telehealth-market-report

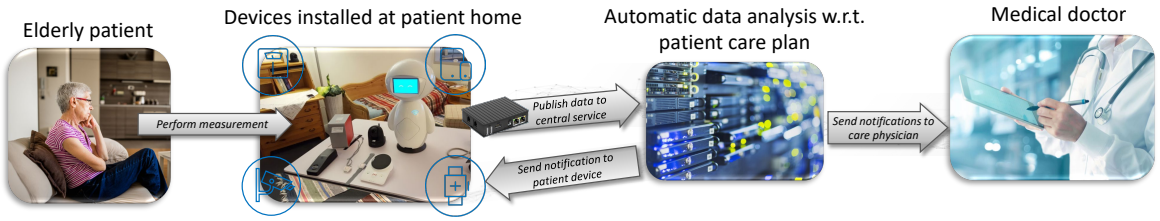

Figure 1. General setup for a remote patient monitoring process

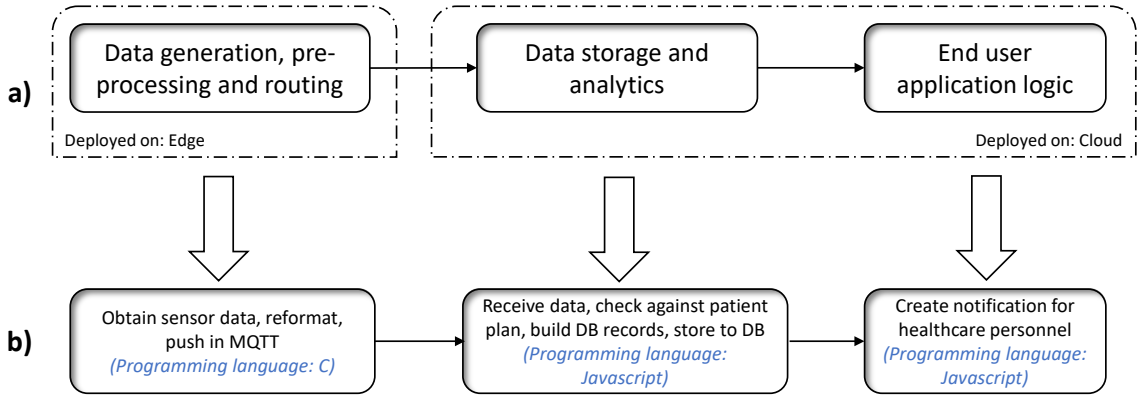

Figure 2. Mapping between a) generic telecare pipeline and b) sample pipeline based on Telluhealth Remote Patient Monitoring solution

plication logic is executed for the end user, which can be both the care physician and the patient.

2.1. Challenges

For many patients, the instantiation and maintenance of the aforementioned process at scale, securely, and privately on hybrid hardware infrastructures come with technical challenges, some of which are discussed in the following list.

1) **Scalability**: Due to the variability of the sensor packages sold and continuous updates/improvements in sensing technology, deploying a completely new setup for each patient is often necessary. This issue is exacerbated by the need to efficiently and flexibly introduce new sensors and monitor different aspects of a patient recovery process. Upgrades to the software caused by this can lead to service outages [6]. And thus, although sensing devices do not often produce significant volumes of data, several different processes equal to at least the number of patients served by a service provider must be deployed and managed.

2) **High cost of testing**: As device sets are determined by the need of each remote patient and devices get shipped and deployed together, compatibility tests are needed to ensure the software and hardware are working well and in tandem [7]. This means that test data must be produced or generated per each case and used with the specific installed version of the software to be delivered to the patient. Even if test data are available, all testing must be performed in an equal setting to the one targeted for the patient, which means that the exact (type of) device shipped must be used, making testing difficult and impractical. Finally, at runtime, multiple process instances (maybe in the thousands) must be used in parallel without downtime, making simulations at scale particularly relevant.

3) **Ensuring control and trustworthiness**: Medical data, in particular, has very high requirements for privacy and security, and ensuring the patient control over their data and trust in the system's security are critical [8]. And in the context of a complex, multi-instance, variable data processing system, ensuring transparency and consistent security and privacy measures is

difficult.

4) **Difficult DevOps process**: Complex data processing systems are difficult to efficiently orchestrate (automate, monitor, tune at runtime) and deploy across the Computing Continuum [9]. Finally, each step has different requirements in terms of hardware resources, such as memory and processing, and finding the optimal resource allocation for a given set of available resources adds additional complexity.

2.2. (Big) data pipelines to the rescue

Big Data pipeline frameworks provide facilities for deploying and managing resources on the Computing Continuum and, depending on the implementation, enable easy reuse and scalability at the individual step or pipeline level. Thus, telemedical data processing pipelines can be specified in a standard form in compliance with security and privacy requirements and then deployed according to the purpose. Alternatively, at runtime, many existing telecare systems produce and, for compliance purposes, store execution traces that are often standardized. They contain enough information to reconstruct one data pipeline or the set of all running pipelines (e.g., using process discovery techniques) and to reuse them, making it possible to discover and specify data pipelines [10]. Authors in [11] propose a framework for the lifecycle of Big data pipelines that covers their management and execution from their inception to their deployment and runtime. Thus, migrating to a mode of operation that relies on data pipelines for processing, service providers must take care of both design and runtime aspects - how to identify the pipelines, configure them, potentially test them, provision the necessary resources, create a deployment topology and finally deploy on the Computing Continuum.

2.3. TelluCare pipeline

TelluCare[2] is a telehealth system for supporting and helping various patients stay at home to the maximum possible extent during treatment and care. It is built for patients who need regular follow-ups caused by chronic diseases such as diabetes, non-working kidneys, chronic obstructive pulmonary disease, or patients with temporary diseases that need to be regularly followed up for some time, such as COVID-19 and cancer. Patients report their physical and mental well-being from home, and about 1000 people have been onboarded.

The digital health system controls various medical devices and specific sensors supporting the care and wellness of the specific patient - i.e., blood pressure meters, scales, glucose meters, and medicine reminders. It can also be connected to sensors deployed for remote supervision, such as bed sensors, motion sensors, and video cameras, as needed. In addition, the system allows integration with third-party systems, e.g., to provide information or alarms to response centers, caregivers, physicians, family members, etc., or to give information to medical systems such as electronic health record systems.

The TelluCare system is provided to patients, including a personal health gateway device deployed at the homes of patients. The software stack of the gateway can be deployed both on a smartphone or in a stationary device provided by the service provider. This device can integrate medical-technical equipment with more proprietary protocols and equipment that can only be connected via USB port, serial port, Ethernet, Wifi, etc. The implementation communicates with the remote patient monitoring system using RESTful APIs[3] or MQTT[4].

Figure 2 (b) shows a subset of the concrete pipeline corresponding to a part of the data processing of the TelluCare system. Data from the sensors is obtained, reformatted, and then securely communicated to a centralized service through an MQTT broker. This process is executed at the gateway device deployed in the home of the patient receiving care. The transmitted data are then checked against the patient plan for any deviations from the prescribed regiment by the doctor and recorded in a database. All data from the pipeline, including execution traces, are stored in conformance to the Fast Healthcare Interoperability Resources (FHIR)[5] standard. If

[2] https://tellu.no/en/services/

[3] Application programming interfaces (APIs) using the Representational State Transfer (REST) architectural style are used to communicate securely over the Internet.

[4] MQTT is a network protocol designed for communication using a publish-subscribe model for message queueing services (https://mqtt.org/)

[5] https://hl7.org/FHIR/

any deviations from the patient plan have been identified in the data, a notification is created and sent to the care physician of the patient. Notably, besides the hardware heterogeneity in the implementation, the programming languages of the steps are also different.

3. Data Pipelines in Telemedicine Using the DataCloud Toolbox

The DataCloud toolbox[6] offers tools that cover the entire lifecycle of (big) data pipelines on the Computing Continuum. The toolbox relies on software containers for implementing pipelines. The lifecycle consists of six stages as depicted in Figure 3 - discovery, design/specification, simulation, provisioning, scheduling (or adaptation at runtime), and deployment. The tools that comprise the toolbox are interoperable and have been designed around their uses by different types of stakeholders at the different stages of the lifecycle [11]. Thereby, business domain experts are involved in the discovery and design of the pipelines and can contribute their knowledge of the domain and value in the data their organization manages. During the pipeline design and simulation stages, data scientists can implement programming logic flexibly, independent of their preferred programming language, and insert pipeline step implementations using a common template approach. Computing continuum infrastructure maintainers can manage the heterogeneous infrastructures, provision new resources, specify adaptation policies, deploy pipelines across the computing continuum, and monitor them during runtime. The computing continuum's resources include Cloud (public/private/hybrid), Edge, Fog, and resource providers (public companies and even private individuals) can incorporate new resources using a shared blockchain-based resource marketplace.

Below we discuss how the data pipeline by Tellucare is implemented using the tools and technologies offered.

3.1. Telecare pipelines using software containers

To solve some of the challenges of telemedical pipelines, the DataCloud toolbox uses software container-based data pipelines as described in [12]. One of the approach's key aspects is using containers to package standalone steps. Such packaging provides an easy means to efficiently orchestrate the pipeline execution across resources. The container-based approach is horizontally scalable and performs better than state-of-the-art data workflow tools (challenge 1). Additionally, control and trustworthiness can be implemented as integral components in each step by including relevant software libraries for secure/private communication and storage (challenge 3). Container systems manage components' deployment, scaling, and networking, which grants the pipeline execution framework the ability to scale each pipeline step independently. Furthermore, container orchestration technologies can manage the distributed heterogeneous hardware infrastructures (in our case, the Edge and Cloud components), constituting the Computing Continuum (challenge 4). Finally, representing the pipeline as an easily deployable collection of program elements enables service providers (using the right tools) to test each step independently or simulate multiple configurations of all steps in different settings and determine requirements for the hardware to support many concurrent pipelines (challenge 2).

3.2. Pipeline discovery

The process of discovering or decomposing existing processes of telemedical applications is enabled through the analysis of execution traces (Figure 3-1) from existing systems [10]. In the DataCloud toolbox, this is done by importing the logs stored in the databases for compliance (in the case of the TelluCare pipeline, this is the FHIR standard). The pipeline discovery tool[7] can then be used to analyze the torrents of stored data and convert it to reveal the sets of events that occurred during the execution of the health application. The data are then segmented, abstracted, and filtered to extract discreet execution traces associated with one or more specific pipelines. The tool can automatically identify and discard unnecessary events that do not contribute to the pipeline execution. The final clean set of execution traces is then stored in an event log repository, where process mining algorithms are

[6] https://datacloud-project.github.io/toolbox/

[7] https://github.com/DataCloud-project/DIS-PIPE

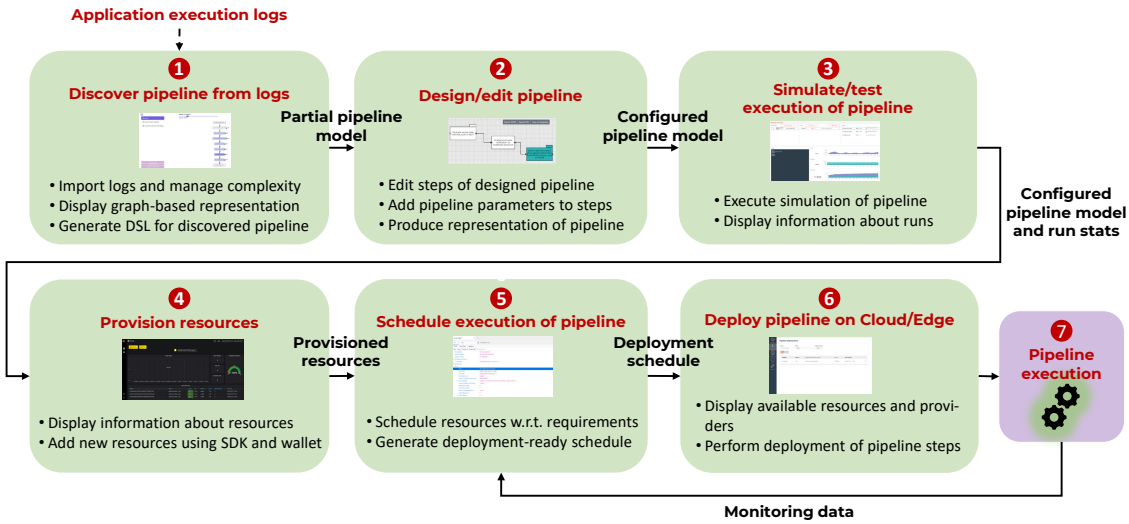

Figure 3. Lifecycle of a telemedicine data pipeline on the computing continuum

applied to discover and learn the structure of the underlying pipelines. The pipeline representations are in a standard format[8]. By analyzing execution traces and identifying the key steps and events in a telemedical pipeline, the discovery tool makes it easier for users to synthesize partial data processes that can be reused and deployed later in the pipeline lifecycle (challenges 3 and 4).

3.3. Pipeline specification

Once the structure of the data process is discovered, it can be translated to a data pipeline (Figure 3-2). This is done by the use of pipeline specification tools and standards [14]. The pipeline specification tool in the DataCloud toolbox[9] can be used to change the discovered process to an executable pipeline by specifying the containers that correspond to the intended functions of the steps and configuring the pipeline with the proper step-specific parameters. Such parameters and configuration may include the execution environment (Edge or Cloud), required hardware capabilities (e.g., the required RAM and CPU on the device to deploy a step), as well as the step communication/data transmission configurations (e.g., using MQTT or web APIs). The tool allows for specifying and maintaining standard pipelines or pipeline fragments (i.e., sub-pipelines) that can be reused when creating solutions for new patients with variations that the specific illness can cause they receive care for. This tool also specifies privacy- and security-compliant pipelines following user and authority requirements. The resulting fully specified data pipeline can then be used directly for deploying on the required hardware at the appropriate scale (challenges 1 and 4).

3.4. Simulation and testing before deployment

As was mentioned, telemedical pipelines can have variations based on the requirements of the patient and the installed medical devices. This means that testing of pipelines before their deployment with patients (Figure 3-3) is needed. Furthermore, as the number of patients a service provider supports grows, predicting the computing resources needed to maintain the infrastructure responsive and running is difficult. Simulation frameworks for data pipelines provide a means to predict such requirements [15]. The DataCloud toolbox contains a tool[10], specifically designed to test container-based data pipelines and provide insights into the requirements at runtime [16]. It extracts the steps from pipeline definitions and performs dry runs by instantiating the containers in an isolated environment. It then estimates their throughput and computing requirements and displays useful monitoring information about their execution state (e.g., this

[8] https://github.com/DataCloud-project/DEF-PIPE-DSL
[9] https://github.com/DataCloud-project/DEF-PIPE-Frontend
[10] https://github.com/DataCloud-project/SIM-PIPE

can be useful if a pipeline variation fails). With this information, a DevOps team can quickly and easily estimate the hardware requirements, and a scheduler can use it to identify the right nodes from the available infrastructure to deploy the pipeline (challenge 2).

3.5. Resource provisioning

In the context of telemedical pipelines, it is sometimes the case that there is a need to provide computing infrastructure outside of available Cloud resources - e.g., when a client is in a remote area outside of availability zones provided by public cloud (Figure 3-4). Furthermore, as more and more pipelines are supported by the service to the point where capacity is exceeded, a service provider may need on-demand provisioning. The resources acquired need to minimally conform to step requirements of the pipeline to enable successful execution and minimize the cost to the service provider. Resource provisioning tools and frameworks in the Edge/Fog frameworks allow for dynamic management of the needed resources [13]. The DataCloud toolbox provides a decentralized tool for provisioning heterogeneous virtualized resources[11] from various providers. It comprises a decentralized marketplace based on a permissionless blockchain to federate virtual machines and containers registered for on-demand usage. It can be used by service providers to dynamically add and remove new nodes to their infrastructure as demand for data processing fluctuates (challenges 1 and 4).

3.6. Scheduling of resources

The deployment of a telemedical pipeline needs to be performed in a distributed setting, possibly over multiple deployment targets in both Cloud and Edge (Figure 3-5). The deployment must, nevertheless, conform to the requirements of the pipeline steps. For example, a step that needs to be performed on the gateway needs to be deployed precisely there, whereas a step that needs to be deployed over the Cloud infrastructure must be deployed on a node that has just enough free resources to host it. The task of determining the deployment targets is done through the use of Computing Continuum schedulers [17]. The DataCloud toolbox provides an advanced tool that performs dynamic scheduling and runtime adaptation for the produced definitions container-based pipelines[12]. The tool can analyze the pipeline structure and requirements and, taking into consideration the capabilities managed by the resource provisioning tool, produce a schedule for the deployment of the pipelines. It also considers the dependencies between the steps and provides a schedule with the specific needed deployment sequence that can be enacted by a deployment engine with minimal overuse of resources. In addition, the tool uses monitoring information and, based on the load of the pipeline, may trigger runtime adaptations of the pipeline, such as scaling up steps that are bottlenecks in the pipeline or releasing unnecessary resources (challenges 1 and 4).

3.7. Pipeline deployment on Cloud/Edge

Once a deployable and sequenced schedule is available, the telemedical pipeline is ready to be deployed (Figure 3-6). For this purpose, the middleware is needed to orchestrate deployment over the provisioned and available Edge and Cloud infrastructures. The DataCloud toolbox provides an advanced orchestrator[13] that is specifically designed to support the distributed deployments based on the aforementioned sequenced schedules. It deploys the pipeline steps on the resources that were identified by the scheduler through the MAESTRO orchestrator[14] (challenge 4). It includes a monitoring system, modified to support multiple compound metrics, such as communication latency and bandwidth during the runtime of the pipeline, and enact adaptation actions (provided by the scheduler), e.g., concerning the requirements for scaling of steps.

4. Summary and Outlook

This paper has argued for using (big) data pipelines to solve some of the challenges related to data processing in telemedicine. It outlined the general structure of a telemedicine setup and pointed out corresponding technical and non-technical challenges that emerge from the increased demand for such services. It continued

[11] https://github.com/DataCloud-project/R-MARKET

[12] https://github.com/DataCloud-project/ADA-PIPE

[13] https://github.com/DataCloud-project/DEP-PIPE

[14] https://themaestro.ubitech.eu/

to describe how data pipelines can address challenges by using a motivating use case with a real-world telemedicine service provider using data pipelines. We also showed how the outlined challenges are concretely addressed using the DataCloud toolbox for supporting container-based data pipelines on the Computing Continuum.

There is a clear emerging direction in telemedicine with the development and deployment of large-scale AI models in medicine, such as Med-PaLM 2 [18], and others [19] that have demonstrated effectiveness in data analysis and automatic identification of medical conditions. We foresee that these AI models will become integral parts of the data analysis in telemedical pipelines. And with the rapid development of other technologies, such as quantum cryptography for secure data transmission and Artificial Intelligence for online analysis of video streams [20], there is significant future potential to revolutionize remote patient monitoring pipelines.

ACKNOWLEDGMENT

This work received partial funding from the Norwegian research council (NFR 323325), the projects DataCloud (H2020 101016835), enRichMyData (HE 101070284), Graph-Massivizer (HE 101093202), BigDataMine (NFR 309691), and NorwAI (NFR 309834).

REFERENCES

1. Kelly, J. T. et al., The Internet of Things: Impact and Implications for Health Care Delivery. *Journal of Medical Internet Research*, 22(11), e20135. https://doi.org/10.2196/20135, 2020.
2. DeFranco, J. F. and Metro, M. J., Internet of Telemedicine. *IEEE Computer*, 55(4), 56–59. https://doi.org/10.1109/MC.2022.3143625, 2022.
3. Perera, M. S. et al., Internet of Things in Healthcare: A Survey of Telemedicine Systems Used for Elderly People. *IoT in Healthcare and Ambient Assisted Living* (pp. 69–88). Springer. https://doi.org/10.1007/978-981-15-9897-5_4, 2021.
4. M. Barika, Orchestrating big data analysis workflows in the cloud: research challenges, survey, and future directions. *ACM Computing Surveys*, https://doi.org/10.1145/3332301, 2019.
5. Balouek-Thomert et al., Towards a computing continuum: Enabling edge-to-cloud integration for data-driven workflows. *International Journal of High Performance Computing Applications*, 33(6), 1159–1174. https://doi.org/10.1177/1094342019877383, 2019.
6. Yoon, J. et al., Remote Virtual Spinal Evaluation in the Era of COVID-19. *International Journal of Spine Surgery*, 14(3), 433–440. https://doi.org/10.14444/7057, 2020.
7. Ftouni, R. et al., Challenges of Telemedicine during the COVID-19 pandemic: A systematic review. BMC Medical Informatics and Decision Making, 22(1), 207. https://doi.org/10.1186/s12911-022-01952-0, 2022.
8. Nittari, G. et al., Telemedicine Practice: Review of the Current Ethical and Legal Challenges. *Telemedicine and E-Health, 26(12)*, 1427–1437. https://doi.org/10.1089/tmj.2019.0158, 2020.
9. Sulis, E. et al., An ambient assisted living architecture for hospital at home coupled with a process-oriented perspective. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-022-04388-6, 2022.
10. A. Augusto et al., "Automated discovery of process models from event logs: Review and benchmark," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 4, pp. 686–705, 2019, https://doi.org/10.1109/TKDE.2018.2841877, 2019.
11. Roman, D. et al., "Big Data Pipelines on the Computing Continuum: Tapping the Dark Data", *IEEE Computer*, 55(11), 74–84. https://doi.org/10.1109/MC.2022.3154148, 2022.
12. Nikolov, N. et al., "Conceptualization and scalable execution of big data workflows using domain-specific languages and software containers", *Internet of Things*, 16, 100440. https://doi.org/10.1016/j.iot.2021.100440, 2021.
13. A. Shakarami et al., "Resource provisioning in edge/fog computing: A comprehensive and systematic review," J. Syst. Architecture, vol. 122, p. 102,362, doi: 10.1016/j.sysarc.2021.102362, Jan. 2022.
14. R. Ranjan et al., "Orchestrating BigData analysis workflows," *IEEE Cloud Comput.*, vol. 4, no. 3, pp. 20–28, https://doi.org/10.1109/MCC.2017.55, 2017.
15. I. Bambrik, "A survey on cloud computing simulation and modeling," *SN Comput. Sci.*, vol. 1, no. 5, Art. no. 249, https://doi.org/10.1007/s42979-020-00273-1, 2020.
16. Thomas, A. et al., "SIM-PIPE DryRunner: An approach for testing container-based big data pipelines and generating simulation data", *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, 1159–1164. https://doi.org/10.1109/COMPSAC54236.2022.00182, 2022.

17. Kimovski, D. et al., Cloud, Fog, or Edge: Where to Compute? *IEEE Internet Computing*, 25(4), 30–36. https://doi.org/10.1109/MIC.2021.3050613, 2021.
18. Wang, D.-Q. et al., Accelerating the integration of ChatGPT and other large-scale AI models into biomedical research and healthcare. *MedComm – Future Medicine*, 2(2), e43. https://doi.org/10.1002/mef2.43, 2023.
19. Singhal, K. et al., Towards Expert-Level Medical Question Answering with Large Language Models (arXiv:2305.09617). *arXiv*. https://doi.org/10.48550/arXiv.2305.09617, 2023.
20. Gill, S. S. et al., AI for next generation computing: Emerging trends and future directions. *Internet of Things*, 19, 100514. https://doi.org/10.1016/j.iot.2022.100514, 2022.

Nikolay Nikolov is a research scientist at SINTEF AS, Norway. His research interests include big data, data enrichment, and the semantic web technologies. Nikolov received his International MSc in Service Engineering from Stuttgart University, University of Crete and Tilburg University. Contact him at https://www.linkedin.com/in/nikolay-vl-nikolov/ or nikolay.nikolov@sintef.no.

Arnor Solberg is Chief DevOps Officer at Tellu AS at Lysaker, Norway. His research interests include Software Engineering, IoT, edge and cloud based systems, eHealth. Solberg received his PhD in Software Engineering from University of Oslo. Contact him at arnor.solberg@tellu.no.

Radu Prodan is a professor in distributed systems at the Institute of Information Technology, University of Klagenfurt, Austria. His research interests include middleware system tools for Cloud, Fog and Edge computing. Prodan received his Habilitation degree in computer science from the University of Innsbruck, Austria. He is a member of ACM. Contact him at radu.prodan@aau.at.

Ahmet Soylu is a full professor at Oslo Metropolitan University, Oslo, Norway. His research interests include data management, knowledge representation, and Semantic Web. Soylu received his PhD degree in computer science from KU Leuven, Belgium. Contact him at ahmet.soylu@oslomet.no.

Mihhail Matskin is a professor at the Royal Institute of Technology - KTH at Stockholm, Sweden. His research interests include software engineering, big data systems, intelligent software systems. Matskin received his PhD in Software Technology from the Institute of Cybernetics, Tallinn, Estonia. Contact him at misha@kth.se.

Dumitru Roman is a senior research scientist at SINTEF AS, Norway and associate professor (adjunct) at University of Oslo, Norway. His research interests span across various aspects of data science and engineering. He received his PhD from University of Innsbruck, Austria. Contact him at https://www.linkedin.com/in/titiroman.