A Systematic Review and Recommendation of Software Architectures for SARS-CoV-2 Monitoring

Kay Smarsly[1][a], Yousuf Al-Hakim[1], Patricia Peralta[1], Silvio Beier[2] and Claudia Klümper[3]

[1]Institute of Digital and Autonomous Construction, Hamburg University of Technology, Germany

[2]Chair of Urban Bioengineering for Resource Recovery, Bauhaus University Weimar, Germany

[3]Laboratory of Environmental Analysis, Department HAM 2, Hamm-Lippstadt University of Applied Sciences, Germany

Keywords: SARS-CoV-2, COVID-19, Wastewater, Monitoring, Surveillance, Software Architectures.

Abstract: The coronavirus disease 2019 (COVID-19) is a highly infectious respiratory disease caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), which has led to the ongoing global pandemic with more than half a billion cases worldwide. Response measures to COVID-19 outbreaks suffer from lagging between detecting and reporting COVID-19 cases and by underreporting. By contrast, using wastewater allows detecting SARS-CoV-2 ribonucleic acid (RNA) in human feces, serving as a timely and reliable basis for devising effective measures to prevent and control COVID-19 outbreaks. As a technological basis, software systems for monitoring SARS-COV-2 RNA in wastewater are required, which are capable of (i) interlinking COVID-19-related data from different sources, (ii) providing user interfaces with remote access, (iii) implementing software design concepts that are well-established, and (iv) deploying on-demand SARS-CoV-2 data analysis. To ensure reliable operation, it is crucial to set up SARS-COV-2 monitoring systems based on sound software architectures. This paper systematically reviews and categorizes software architectures for SARS-CoV-2 monitoring systems, considering journals, book series, and conference proceedings indexed in the Scopus database. Then, a software architecture for SARS-CoV-2 monitoring systems is proposed. In future work, the proposed software architecture may be implemented and validated for SARS-CoV-2 monitoring.

1 INTRODUCTION

The global pandemic of the coronavirus disease 2019 (COVID-19), with hundreds of millions of confirmed cases, has caused an ongoing health crisis affecting countries, which are facing COVID-19-related challenges that have triggered severe social and economic consequences. The COVID-19 cases reported to health authorities, together with COVID-19-related hospitalizations and death rates, are key metrics used to devise response measures to COVID-19 outbreaks (Huber and Langen, 2020). However, the practice of relying on reports of COVID-19 key metrics when devising response measures comes mainly with two downsides. First, the time period between detecting and reporting COVID-19 cases to public health authorities is relatively long (Cheng, et al., 2021). Second, not all infections are detected and/or reported, entailing high rates of underreported COVID-19 cases (Schneble, et al., 2021). As a result, the timeliness and the effectiveness of response measures are limited, having a non-negligible effect on public health and on the economy.

To overcome the limitations of the current practice, the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), causing COVID-19, may be monitored in wastewater (Tandukar, et al., 2022). The SARS-CoV-2 RNA can be detected in human feces days to a week before the onset of symptoms (Wölfel, et al., 2020), thus serving as a proactive approach towards identifying potential COVID-19 outbreaks and complementing the COVID-19 metrics reported to public health authorities (Nag, et al., 2022). In general, genomic monitoring of viruses in wastewater has proven to be a valuable tool for detecting and identifying variants of concern and for establishing early warning systems to support response measures on a scientifically

[a] https://orcid.org/0000-0001-7228-3503

Smarsly, K., Al-Hakim, Y., Peralta, P., Beier, S. and Klümper, C.
A Systematic Review and Recommendation of Software Architectures for SARS-CoV-2 Monitoring.
DOI: 10.5220/0011593000003414
In *Proceedings of the 16th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2023) - Volume 5: HEALTHINF*, pages 211-217
ISBN: 978-989-758-631-6; ISSN: 2184-4305

sound basis (Karthikeyan, et al., 2022). As a consequence, response measures may be devised earlier and more goal-driven, causing milder social, economic, and cultural impacts as compared to the current practice. For monitoring SARS-CoV-2 variants, influent samples are collected from wastewater treatment plants and analyzed in laboratories using reverse transcription quantitative real-time PCR (RT-qPCR) and/or genomic sequencing. To devise response measures as precisely as possible, results achieved by SARS-CoV-2 monitoring must be correlated with data sets (e.g. COVID-19 metrics) stemming from other data sources.

To efficiently and accurately correlate the results achieved by SARS-CoV-2 monitoring with data from other sources, software systems need to be implemented that are capable of (i) interlinking COVID-19-related data from third parties, (ii) providing user interfaces with remote access, (iii) implementing software design concepts that are well-established, and (iv) deploying on-demand data analysis. Hence, a software architecture must be defined, on which SARS-CoV-2 monitoring systems may be built. Therefore, this paper systematically reviews software architectures of SARS-CoV-2 monitoring systems, and of monitoring systems in general, considering peer-reviewed journals, book series, and conference proceedings.

The remainder of this paper is organized as follows. First, the research methodology of the review is described. Second, the results of the systematic review are presented, and requirements of SARS-CoV-2 monitoring systems are identified. Third, based on a discussion of the results, a software architecture for SARS-CoV-2 monitoring systems is proposed. Finally, conclusions are drawn, and a brief outlook on future research is presented.

2 RESEARCH METHODOLOGY

To ensure high quality of the literature screened in the systematic review, journals, book series and conference proceedings indexed in the Scopus database are used (Elsevier, 2022). The review includes literature published between 2019 to 2022 because the first COVID-19 case was registered in December 2019. The research methodology follows three main steps, (i) data collection, (ii) data organization, and (iii) data analysis. In the first step, data collection, two search strings are used, and the first search string is *("software architecture" and ("monitoring" or "surveillance"))*. Owing to the novelty of SARS-CoV-2 monitoring, SARS-CoV-2-related terms are omitted in the first search string, i.e. the systematic review starts with software architectures for monitoring and surveillance applications in general, which results in an initial search result of 1964 papers. The second search string is *("software architecture" and ("SARS-CoV-2" or "covid*" or "corona"))*, covering literature on software architectures for SARS-CoV-2 monitoring, including monitoring efforts based on wastewater, as well as literature reporting on software architectures for SARS-CoV-2 platforms without monitoring features, leading to 39 search results.

In the second step of the research methodology, data organization, the results are subject to inclusion and exclusion criteria, to ensure high quality of the search results. It should be noted that, in general, research papers that consider software architectures of monitoring systems as well as literature reviews that summarize software architectures for monitoring systems are included. Additionally, papers written in different languages are included. The primary exclusion criterion is related to the question whether the software architecture has been described in a way that allows a clear identification for the systematic review.

In the third step of the research methodology, data analysis, the following research question is to be answered: "What types of software architectures have been implemented for SARS-CoV-2 monitoring systems?" When answering the research question, identified literature is analyzed, starting from a broader analysis of monitoring system architectures in general, ending up in architectures of systems specifically designed for SARS-CoV-2 monitoring. The research methodology is visualized in Figure 1.

3 REVIEW RESULTS

Regarding the first search string, i.e. monitoring system architectures in general, from 1964 papers initially found, 100 papers are relevant to the review. Regarding the second search string, i.e. monitoring system architectures related to SARS-CoV-2 monitoring, from 39 papers initially found, 3 papers are relevant to the review. The software architectures of the monitoring systems presented in the papers found from the first search string are categorized in Figure 2 and described in the following subsection, followed by a description of the software architectures proposed for SARS-CoV-2 monitoring systems.

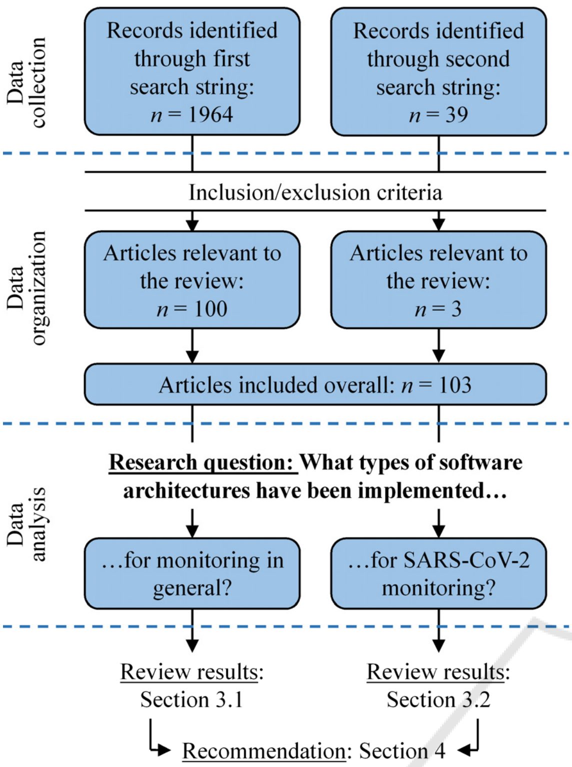

Figure 1: Research methodology.

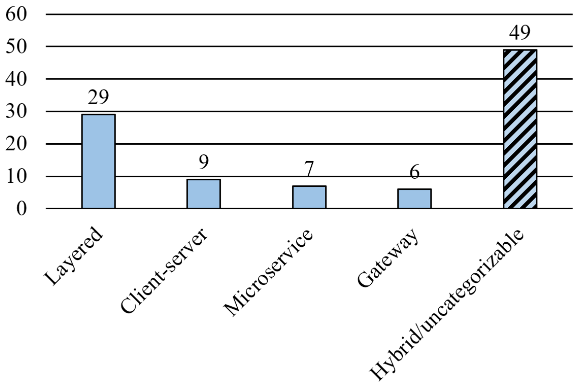

Figure 2: Categorization of architectural design patterns used for implementing monitoring systems that have been published between 2019 and 2022.

3.1 Monitoring System Architectures in General

As can be seen from Figure 2, the software architectures used for implementing monitoring systems are characterized by four predominant design patterns, shown in Figure 3, which are materialized in four corresponding software architectures,

- the layered architecture,
- the client-server architecture,
- the microservice architecture, and
- the gateway architecture.

The software architectures used for implementing monitoring systems will be described in the following paragraphs. For further details on the software architectures and the underlying architectural design patterns, the interested reader is referred to Fowler (2002). From Figure 2, it is remarkable that the architectural design patterns of most software architectures are hybrid or uncategorizable, the reason of which will be addressed in the discussion section.

The ***layered architecture*** is most frequently used. The layered architecture consists of three layers, the presentation layer, the domain layer, and the data source layer. The presentation layer ensures interactions between users and the monitoring system, the domain layer contains the logic of the monitoring system, such as validations of user requests, and the data source layer enables communication with a database (Fowler, 2002). The layer-wise separation has multiple benefits to monitoring systems, as each layer is understood as a single, independent unit. The separation is particularly useful for comprehensive monitoring processes with various parties involved in monitoring or with respect to a substitution of layers, e.g. in case of updates of monitoring system elements.

The ***client-server architecture*** comprises two main elements, client and server. The client requests data, services, or other resources from the server. The server answers the requests and provides the corresponding results and resources to the client. Multiple clients may be connected to a server, where not all clients need to be situated in the same network. Since the client-server architecture centralizes data within a monitoring system, a distinct advantage of the architecture, besides security aspects owing to a central server, is the ease of maintenance. Centralized data not being distributed over several systems can easily be managed, as compared to decentralized data.

The ***microservice architecture*** is used in various recent monitoring systems. The most important characteristic of the microservice architecture is the "componentization" through microservices. Essentially, microservices are small software components that can be deployed, scaled, and tested independently (Thönes, 2015). The componentization through services allows independent deployment with services being independently scalable and being written in different programming languages. The microservice architecture is therefore particularly helpful for comprehensive, heterogeneous monitoring systems with multiple parties involved in monitoring.

The ***gateway architecture*** encapsulates decoupled elements of a monitoring system. The gateway acts as an interface, through which clients may request a service from a server. Monitoring systems based on the gateway architecture are structured clearly, since coordination and communication are managed by the gateway. The gateway architecture has advantages when implemented as a physically distributed system because system changes do not affect every element, and only the gateway needs to be adapted to the changes.

3.2 Monitoring System Architectures for SARS-CoV-2 Monitoring

The three SARS-CoV-2 monitoring systems identified in the systematic review implement the layered architecture. Regarding the research topic, Tabbiche, et al. (2021) propose a modeling tool for designing applications that monitor real-time context information. A prototype application, using a 4-layered architecture, is presented in a pervasive environment that tracks the movements of persons who have tested positive, to anticipate COVID-19 contaminations. Sethi and Pal (2021) introduce a monitoring system for detecting COVID-19 exposure in public transportation vehicles. The monitoring system is implemented on three layers, where a cloud layer and a user layer are connected through a communication layer. Finally, the monitoring approach pursued within health care software proposed by De Moura Costa, et al. (2022) aim to evaluate performance metrics related to COVID-19, such as latency, throughput, and send rates in different locations. Two layers with multiple sub-layers and individual services have been implemented, and privacy, scalability, and latency reduction have been considered important requirements when choosing the layered architecture.

4 DISCUSSION AND RECOMMENDATION

From the systematic review, it can be concluded that the software architectures of monitoring systems strongly depend on the monitoring objective. In general, the layered architecture is most frequently used, as it is easy to implement, modular, and allows substituting layers (i.e. elements of the system).

As mentioned earlier, it is apparent from Figure 2 that most architectures proposed for monitoring systems in general are hybrid or uncategorizable. A reason, as expert surveys indicate, is that many monitoring systems have been implemented from a pragmatic engineering standpoint, sometimes in a trial-and-error manner, without following professional software design concepts from a computer science point of view.

Regarding software architectures explicitly proposed for SARS-CoV-2 monitoring, the layered architecture has been used in all papers found in this review. It is evident from the systematic review that software architectures for SARS-CoV-2 monitoring systems have rarely been reported because of the novelty of the research problem. In addition, the urgency of this matter has resulted in SARS-CoV-2 monitoring systems being implemented without explicit architectural design and/or without reporting the software architectures in scientific literature.

Taking the results summarized above as a starting point, the software requirements relevant to SARS-CoV-2 monitoring systems will be described in the following subsection. Thereupon, a software architecture for SARS-CoV-2 monitoring systems will be proposed.

4.1 Software Requirements for SARS-CoV-2 Monitoring Systems

To design the software architecture for SARS-CoV-2 monitoring systems, functional and non-functional requirements are considered. The following discussion will exemplarily focus on designing a prototype SARS-COV-2 monitoring system within the "CoMoTH" project using wastewater at 23 wastewater treatment plants in the Free State of Thuringia, Germany (CoMoTH, 2022). Project partners, such as federal, state, and local health authorities as well as companies, are involved in the design process to provide advice according to their needs in data analysis and visualization.

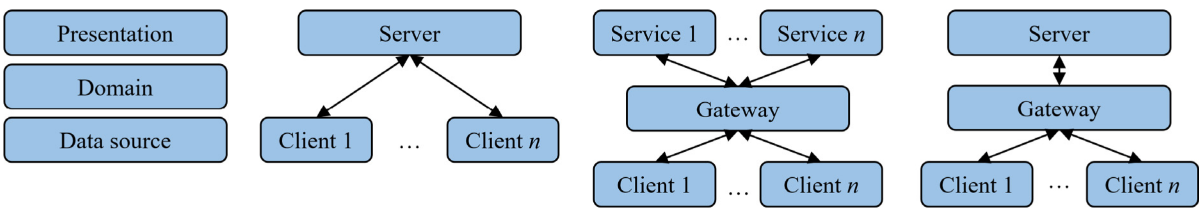

Figure 3: Layered, client-server, microservice, and gateway architecture.

As for the ***functional requirements***, the monitoring system must be accessible through web browsers and must provide user-friendly visualizations of the data and of the data analysis results, the latter based on well-established mathematical statistics as well as on artificial intelligence techniques. Representing another functional requirement, the needs of the project partners as well as the needs of the general public (being users of the monitoring system) are to be implemented. A further functional requirement is the ability to couple the wastewater monitoring system with COVID-19-related software platforms operated by third parties. The German Government's central scientific institution in the field of biomedicine, the Robert Koch Institute (RKI), provides a digital interface used by hospitals, laboratories, and test centers to forward COVID-19-related data to the RKI, which is to be interfaced with the proposed monitoring system. Furthermore, the monitoring system must be modular, allowing for easy modifications. The functional requirements are summarized as follows: The software architecture must ensure that the monitoring system

- is accessible through web browsers and provides user-friendly visualizations of the data and of the data analysis results to be achieved based on well-established mathematical statistics as well as on artificial intelligence techniques,
- is, in a first step, adjusted to wastewater treatment plants in Thuringia, Germany, while the concept is generally valid,
- implements the needs of the project partners and of the general public (being users of the monitoring system),
- is modular, allowing for easy modifications, and is interlinked with COVID-19-related software platforms operated by third parties.

In addition to the functional requirements listed above, the architecture must ensure that the following ***non-functional requirements*** will be met by the monitoring system. The non-functional requirements are basically the quality constraints that the monitoring system must satisfy:

- ***Usability:*** The users are enabled to perform the monitoring tasks safely, effectively, and efficiently.
- ***Integrity:*** Data accuracy and consistency are ensured when storing, processing, and retrieving monitoring data.
- ***Extensibility:*** Without much effort, the monitoring system is expandable.
- ***Portability:*** The monitoring system can be implemented on different platforms.
- ***Security:*** The monitoring system is protected against disclosure, theft and damage of data and disruption or misdirection of services.
- ***Reliability/availability:*** For the specified periods of time (reliability) and at the specified intervals of time (availability), the monitoring system functions under stated conditions.
- ***Maintainability:*** The monitoring system is easily maintainable to correct defects, to repair or replace faulty software components, and to maximize reliability.
- ***Robustness:*** The monitoring system copes with errors during execution and with erroneous input.
- Further non-functional requirements known from software engineering must be met, such as scalability, performance, efficiency, safety, flexibility, and reusability.

4.2 A Software Architecture for SARS-CoV-2 Monitoring Systems

As an outcome of the review and drawing from the requirements summarized in the previous subsection, the following software architecture is proposed for SARS-CoV-2 monitoring systems and recommended for the specific SARS-CoV-2 monitoring system described above. Unlike software architectures identified in the review, a model-view-controller (MVC) architecture is proposed, which divides a software system into three elements, model, view, and controller, as shown in Figure 4. The ***model***, independently from user interfaces, contains domain logic and manages the data of the system (e.g. in form of one or more databases that contain data relevant to SARS-CoV-2 monitoring). The ***view*** acts as an interface between users and the application, where different views may be devised, e.g. through web browsers, remote application programming interfaces (APIs), or command-line interfaces. Moreover, the views can be tailored to a user or a user group (such as federal, state, and local health authorities or the general public), and the same information provided by the model can be presented differently, depending on the view. Finally, the ***controller*** receives user inputs (such as requests regarding SARS-CoV-2 data), performs interactions with the model, and initiates view updates.

The MVC architecture specifically supports the processes relevant to SARS-CoV-2 monitoring and the specific needs of the users and user groups involved in monitoring. As can be seen from Figure

4, using a web browser, a user interacts with a dashboard provided by the view (1) to visualize data and to conduct online data analyses. Thereupon, the view alerts the controller (2), which contacts the model to retrieve the data and conducts the analyses, applying the algorithms that are available to the view. Then, the controller updates the model (3) and sends the requested data and the analysis results to the view (4), which updates the visualization in the dashboard for the user (5).

In summary, every element can be designed, implemented, and validated independently from each other because of the strict separation of the elements. The separation of view and model is most important, as different presentations, tailored to different user groups, may be implemented independently from the model. Vice versa, different databases or other sources for integrating SARS-CoV-2 data may be included into the systems (such as RKI databases or other databases operated by health authorities) without affecting the presentation or the controller, which handles user requests and makes use of algorithms (such as AI algorithms) to analyze the data of the model before passing the results to the view. Moreover, the view may provide different presentations, tailored to different users or user groups, as the presentation depends on the model, but the model does not depend on the presentation.

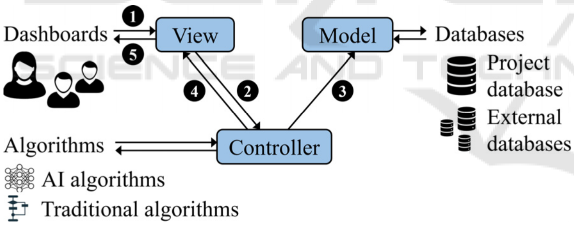

Figure 4: Proposed software architecture for SARS-CoV-2 monitoring systems.

5 SUMMARY AND CONCLUSIONS

Response measures to COVID-19 outbreaks primarily depend on the number of COVID-19 cases, on hospitalization rates, and on the death rates reported to health authorities. Reporting is a practice that depends on timely and reliable reactions (from infected individuals, test centers, and local authorities). The current practice thus suffers from lagging between detecting and reporting COVID-19 cases and by underreporting. By contrast, using wastewater allows detecting SARS-CoV-2 RNA in human feces, serving as a timely and reliable basis for devising effective measures to prevent and control COVID-19 outbreaks. To ensure reliable operation of SARS-CoV-2 monitoring systems, it is crucial to design monitoring systems based on sound software architectures. This paper has systematically reviewed and categorized software architectures for monitoring systems in general and for SARS-CoV-2 monitoring systems in particular, considering journals, book series, and conference proceedings indexed in the Scopus database that have been published between 2019 and 2022. Building upon the findings achieved in the review, a software architecture for SARS-CoV-2 monitoring systems has been proposed.

From the systematic review, it has been concluded that the software architectures of monitoring systems strongly depend on the monitoring objective. Regarding software architectures of monitoring systems in general, most architectures are hybrid or uncategorizable. A reason, as expert interviews indicate, is that many monitoring systems have been implemented from a pragmatic engineering standpoint, sometimes in a trial-and-error manner, without conducting professional software design concepts from a computer science point of view. With respect to SARS-CoV-2 monitoring systems, only very few software architectures have been reported, which may be attributed to the novelty of this matter.

Considering the results of the review and building upon the analysis of the software requirements relevant to SARS-CoV-2 monitoring systems, an architecture for SARS-CoV-2 monitoring system has been be proposed. The architecture is based on an MVC pattern because of its modularity and the separations of view and controller as well as of view and model, the latter being fundamental, as different models may be exchanged and integrated into a SARS-CoV-2 monitoring system independently from the view. In future work, the authors aim to implement the proposed SARS-CoV-2 monitoring system and, using the monitoring system, to validate the proposed approach.

ACKNOWLEDGEMENTS

This research is partially funded by Bauhaus University Weimar, Germany, under award number 2360200276-2/21, which is gratefully acknowledged. The authors would also like to thank the Free State of Thuringia, Germany, as this work has partially been conducted within the framework of the CoMoTH project, funded by Thüringer Aufbaubank under grant number 2021 FE 9143/44. Finally, the authors would like to record their deep gratitude for the invaluable support and suggestions provided by the research teams at Robert Koch Institute.

REFERENCES

Cheng, C., et al. (2021). The incubation period of COVID-19: A global meta-analysis of 53 studies and a Chinese observation study of 11 545 patients. *Infectious Diseases of Poverty*, 10(2021), 119.

CoMoTH (2022). Research project “Monitoring of SARS-CoV-2 in wastewater in Thuringia [SARS-CoV-2-Abwassermonitoring in Thüringen]”, funded by the Thuringian Development Bank under grant number 2021 FE 9143/44.

De Moura Costa, H. J., et al. (2022). A Fog and Blockchain Software Architecture for a Global Scale Vaccination Strategy. *IEEE Access*, 10(2022), pp. 44290-44304.

Elsevier (2022). Scopus Content Coverage Guide. Elsevier B.V, Amsterdam, The Netherlands. Online available: https://www.elsevier.com/__data/assets/pdf_file/0007/69451/Scopus_ContentCoverage_Guide_WEB.pdf. Accessed: August 18, 2022.

Fowler, M. (2002). Patterns of Enterprise Application Architecture. Cambridge, UK: Addison-Wesley Professional.

Huber, M., Langen, H. (2020). Timing matters: the impact of response measures on COVID-19-related hospitalization and death rates in Germany and Switzerland. *Swiss Journal of Economics and Statistics*, 10(2020), 156.

Karthikeyan, S., Levy, J. I., De Hoff, P., et al. (2022). Wastewater sequencing reveals early cryptic SARS-CoV-2 variant transmission. *Nature*, 2022, https://doi.org/10.1038/s41586-022-05049-6.

Nag, A., et al. (2022). Monitoring of SARS-CoV-2 Variants by Wastewater-Based Surveillance as a Sustainable and Pragmatic Approach – A Case Study of Jaipur (India). *Water*, 14(3), 297.

Schneble, M., De Nicola, G., Kauermann, G., Berger, U. (2021). A statistical model for the dynamics of COVID-19 infections and their case detection ratio in 2020. *Biometrical Journal*, 63(8), pp. 1623-1632.

Sethi, V., Pal, S. (2021). Controlling Spread of COVID-19 Using VANETs. In: *Proceedings of the 2021 IEEE International Conference on Communications*, Montreal, QC, Canada, 06/14/2021.

Tabbiche, M. N., Khalfi, M. F., Adjoudj, R. (2021). A Smart Modeling Tool for Model-Driven Engineering of Ubiquitous Applications: Covid-19 Contact-Tracer. In: *Proceedings of the 2021 International Conference on Recent Advances in Mathematics and Informatics*, Tebessa, Algeria, 09/21/2021.

Tandukar, S., et al. (2022). Detection of SARS-CoV-2 RNA in wastewater, river water, and hospital wastewater of Nepal. *Science of The Total Environment*, 824(2022), 153816.

Thönes, J. (2015). Microservices. *IEEE Software*, 32(1), pp. 113-116.

Wölfel, R., et al. (2020). Virological assessment of hospitalized patients with COVID-2019. *Nature*, 581(2020), pp. 465-469.