Summary of a Bulletin Video entitled:

Rapid Development of a Data Visualization Service in an Emergency Response

Saiful Khan[1,6]* Phong H. Nguyen[2]† Alfie Abdul-Rahman[3]‡ Euan Freeman[4]§
Cagatay Turkay[5]¶ Min Chen[1]‖
[1] University of Oxford, [2] Redsift Ltd., [3] King's College London, [4] University of Glasgow, [5] University of Warwick, [6] STFC, UK

1 Introduction

In this bulletin, we summarize the work presented in [7], where we discussed the rapid development of RAMPVIS service during the COVID-19 pandemic. The RAMPVIS system encompasses various components, including its web-based front-end with thousands of plots and dashboards, advanced visualizations, search, and portal features, among others. Meanwhile, the back-end involves dynamically updated data streams from multiple modalities, e.g., COVID-19, analytical and model data, a search engine, automated agents, and a knowledge base or ontology [4, 5]. To overcome the evolving requirements of the RAMPVIS system, its complexity, and the diverse components involved, a service-oriented architecture approach was adopted, ensuring agile development, integration, and deployment. Developing a service-oriented architecture typically involves a lengthy software life cycle. However, in certain circumstances, such as emergency response situations, the need arises for the development of a service-oriented system within a compressed time frame of weeks or months. A small group of volunteer resources [1, 3] tackled this challenge by creating a data visualization service for a group of epidemiologists and modeling scientists.

2 Functional Role of RAMPVIS Service

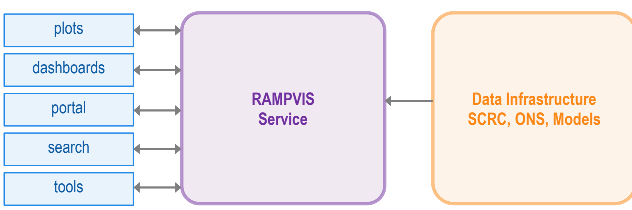

Figure 1: The functional role of the RAMPVIS service.

RAMPVIS service was developed for a crucial role in transferring data from various data infrastructures to multiple plots and dashboards, as shown in Fig. 1. The RAMPVIS service also includes search and personalization features to make it easier for users to access the most relevant visualizations. And eventually expanded to support more complex visualization tools.

3 Requirements and Methodology

The RAMPVIS service was created to effectively visualize COVID-19 data. However, the absence of prior work on visualization services presented a challenge. Traditional visualization system development techniques were limited to high-performance computation and modern visualization libraries like D3.js for easy visualization on common devices.

While the primary requirement was to visualize COVID-19 data streams for a group of epidemiologists and modeling scientists, however, the scale, nature, and modalities of the data were unclear due to the rapid changes in the underdevelopment Scottish COVID-19 Response Consortium (SCRC) [2]. Hence, the overall requirements, such as visualizations, tools, and features required to build a usable system were also unclear. This made it difficult to determine the overall requirements such as visualizations, tools, and features needed for the epidemiologists and modeling scientists to use at the beginning.

To address this obstacle, service-oriented architecture and principles were introduced as a key design choice that can support the evolutionary nature of the system using an agile software development lifecycle cycle. The RAMPVIS service utilized insights from other web applications, service composition, and service engineering to overcome these challenges.

1. Provide data visualization through commonly used web technology, i.e., our service had to meet the standards of services computing.
2. Additionally, the services needed to be developed and deployed quickly in conjunction with the SCRC data infrastructure.
3. To ensure effective and efficient maintenance and operation, i.e., the service also needed to be resilient to changes and easy to maintain.

To meet the requirements, various design choices were incorporated into RAMPVIS, such as:

1. The visualizations were made accessible via web browsers for user convenience.
2. An ontology-centered architecture was developed to manage data stream relationships and visualization functions.
3. Automatic and semi-automatic agents were developed to handle routine tasks and dynamic changes.
4. Database and search services were used to manage, store, and search various data quickly and easily.
5. The D3.js library was utilized to maximize visualization developers' resources and experiences.
6. A template-based development and propagation agent [6] was introduced to facilitate semi-automatic and rapid scaling.
7. The agile software engineering lifecycle enabled collaboration and continuous improvement.
8. All components were developed as open-source software, aligning with the SCRC's requirement for openness and facilitating cross-team communication and coordination.

*e-mail: saiful.khan@stfc.ac.uk
†e-mail: phong.nguyen@redsift.io
‡e-mail: alfie.abdulrahman@kcl.ac.uk
§e-mail: euan.freeman@glasgow.ac.uk
¶e-mail: cagatay.turkay@warwick.ac.uk
‖e-mail: min.chen@oerc.ox.ac.uk

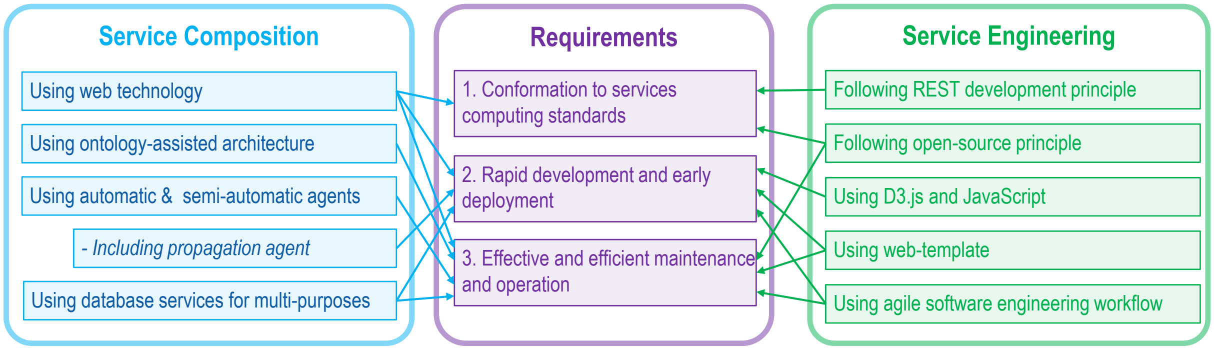

Figure 2: A summary of the service components and engineering strategies created for producing the RAMPVIS visualization service during the COVID-19 emergency response.

The changing needs of the RAMPVIS system affected the way the service was composed and engineered (Fig. 2). The requirements, design, and architecture evolved simultaneously as discussed in [7].

4 Proposed Tiered Architecture

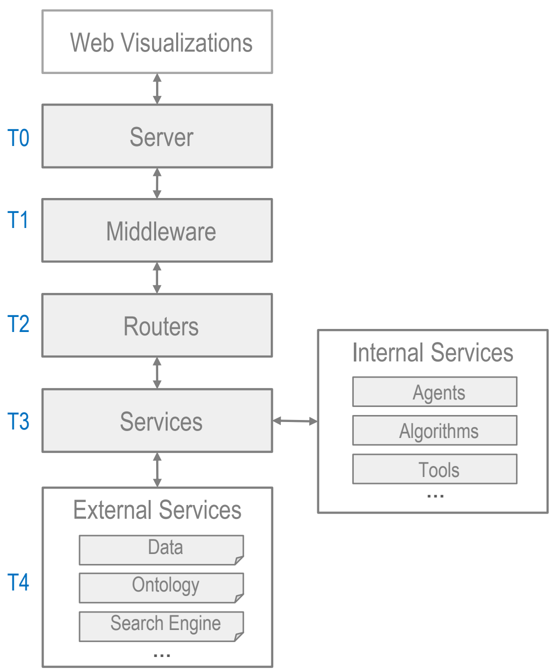

Figure 3: A simplified illustration depicting the functionality of each tier in our n-tier architecture.

The services requirements discussed in Fig. 2 were implemented using a tiered architecture, as in Fig. 3. The architecture comprises five tiers (T0—T4).

1. T0 balances incoming HTTP requests across REST API instances.
2. T1 middleware forwards requests to the appropriate controller.
3. T2 manages the flow and operations.
4. T3 services take action and communicate with other services.
5. T4 integrates with external resources for enhanced capabilities.

By employing this tiered architecture, the RAMPVIS service achieves a modular and scalable structure, enabling efficient request processing, seamless communication between components, and effective utilization of external services. A detailed description of this architecture can be found in [7].

5 Reflection

We adopted agile principles in developing the RAMPVIS services by using a services architectural pattern that allowed for the decoupling of our system into smaller, independent services. Our experience underscores the importance of test-driven development from the beginning to enhance efficiency. We experienced trade-offs between using basic and advanced frameworks, both impacting scalability and development speed. Our work can serve as a blueprint for rapidly developing a scalable data visualization system. Further advancements are encouraged in the areas of easily adaptable or zero-configuration services development, service diagnosis, log monitoring, and visualization to streamline the development and maintenance of service-based systems.

Acknowledgments

We thank all those who have been part of the RAMPVIS. This work was supported in part by the UKRI/EPSRC grant EP/V054236/1.

References

[1] RAMPVIS Volunteers. https://sites.google.com/view/rampvis/volunteers/.

[2] SCRC: Scottish COVID-19 response consortium. https://scottishcovidresponse.github.io/.

[3] M. Chen, A. Abdul-Rahman, D. Archambault, J. Dykes, P. Ritsos, A. Slingsby, T. Torsney-Weir, C. Turkay, B. Bach, R. Borgo, A. Brett, H. Fang, R. Jianu, S. Khan, R. Laramee, L. Matthews, P. Nguyen, R. Reeve, J. Roberts, F. Vidal, Q. Wang, J. Wood, and K. Xu. RAMPVIS: Answering the challenges of building visualisation capabilities for large-scale emergency responses. *Epidemics*, 39:100569, 2022. doi: 10.1016/j.epidem.2022.100569

[4] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann. Knowledge Graphs. *ACM Comput. Surv.*, 54(4), 2021. doi: 10.1145/3447772

[5] S. Khan, U. Kanturska, T. Waters, J. Eaton, R. Bañares Alcántara, and M. Chen. Ontology-Assisted Provenance Visualization for Supporting Enterprise Search of Engineering and Business Files. *Advanced Engg. Informatics*, 30(2):244–257, 2016. doi: 10.1016/j.aei.2016.04.003

[6] S. Khan, P. Nguyen, A. Abdul-Rahman, B. Bach, M. Chen, E. Freeman, and C. Turkay. Propagating Visual Designs to Numerous Plots and Dashboards. *IEEE Trans. Visualization & Computer Graphics*, 28(1):86–95, 2022. doi: 10.1109/TVCG.2021.3114828

[7] S. Khan, P. H. Nguyen, A. Abdul-Rahman, E. Freeman, C. Turkay, and M. Chen. Rapid Development of a Data Visualization Service in an Emergency Response. *IEEE Trans. on Services Computing*, 15:1251–1264, 2022. doi: 10.1109/TSC.2022.3164146