







DIY Challenge Blueprint: From Organization to Technical Realization in Biomedical Image Analysis

Leonard Klausmann^{*1,2}, Tobias Rueckert^{*1,3}, David Rauber¹, Raphaela Maerkl¹, Suemeyye R. Yildiran¹, Max Gutbrod¹, and Christoph Palm^{1,2}
leonard.klausmann@oth-regensburg.de

¹ Regensburg Medical Image Computing (ReMIC), OTH Regensburg, Regensburg, Germany

² Regensburg Center of Health Sciences and Technology (RCHST),
OTH Regensburg, Regensburg, Germany

³ AKTORMed Robotic Surgery, Neutraubling, Germany

Abstract. Biomedical image analysis challenges have become the de facto standard for publishing new datasets and benchmarking different state-of-the-art algorithms. Most challenges use commercial cloud-based platforms, which can limit custom options and involve disadvantages such as reduced data control and increased costs for extended functionalities. In contrast, Do-It-Yourself (DIY) approaches have the capability to emphasize reliability, compliance, and custom features, providing a solid basis for low-cost, custom designs in self-hosted systems. Our approach emphasizes cost efficiency, improved data sovereignty, and strong compliance with regulatory frameworks, such as the GDPR.

This paper presents a blueprint for DIY biomedical imaging challenges, designed to provide institutions with greater autonomy over their challenge infrastructure. Our approach comprehensively addresses both organizational and technical dimensions, including key user roles, data management strategies, and secure, efficient workflows. Key technical contributions include a modular, containerized infrastructure based on Docker, integration of open-source identity management, and automated solution evaluation workflows. Practical deployment guidelines are provided to facilitate implementation and operational stability. The feasibility and adaptability of the proposed framework are demonstrated through the MICCAI 2024 PhaKIR challenge with multiple international teams submitting and validating their solutions through our self-hosted platform. This work can be used as a baseline for future self-hosted DIY implementations and our results encourage further studies in the area of biomedical image analysis challenges.

Keywords: Biomedical challenges · Image analysis · Blueprint · Guidelines · Self-hosting · Do-It-Yourself

* L. Klausmann (corresponding author) and T. Rueckert contributed equally.

Supplementary Information The full code and documentation are available at https://github.com/remic-othr/PhaKIR_DIY.

1 Introduction

Biomedical image analysis challenges serve as key benchmarks for algorithm evaluation, providing curated datasets, standardized metrics, and public leaderboards [16]. Most challenges rely on commercial cloud-based platforms such as Grand Challenge [23] or Synapse [29], which streamline workflows like data hosting, user registration, and leaderboard management. Figure 1 illustrates the increasing adoption of such platforms in the challenges conducted at the annual Medical Image Computing and Computer Assisted Interventions (MICCAI) conference from 2020 to 2024. However, hosting medical data on external platforms often raises compliance concerns with institutional and national regulations. In addition, commercial solutions may impose fees or technical constraints that limit flexibility and control over challenge pipelines.

Motivation and Gap A self-hosted approach offers institutions greater control over data, infrastructure, and evaluation pipelines, enabling regulatory compliance and customization. Storing data internally allows tailored security policies and better adherence to frameworks such as the General Data Protection Regulation (GDPR) [17]. However, technical and organizational barriers often deter organizers, and existing guidance on setting up and managing such platforms remains sparse.

Contributions This paper provides a structured blueprint for DIY biomedical image analysis challenges, covering organizational aspects (e.g., role definitions, data management, challenge workflows) and technical considerations (e.g., infrastructure, authentication, storage). We showcase practical implementation, best practices, and key learnings through the Surgical Procedure Phase Recognition, Keypoint Estimation, and Instrument Instance Segmentation (PhaKIR) challenge [27,26,28], conducted during MICCAI 2024.

Paper Outline Section 2 outlines the typical challenge lifecycle and stakeholder roles. In Section 3, core infrastructure components are introduced and linked to the system architecture. Section 4 details the PhaKIR challenge as a case study. A discussion regarding lessons learned and limitations is given in Section 5, while Section 6 concludes with future directions.

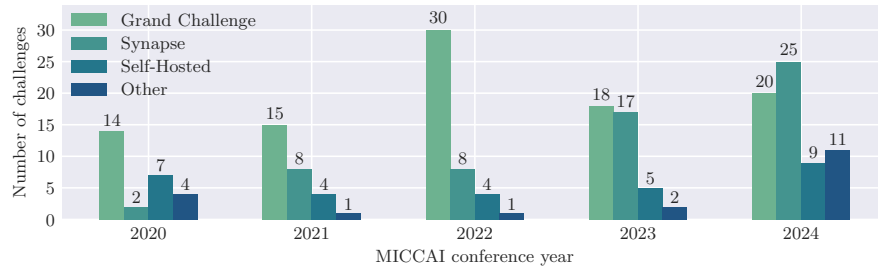


Fig. 1. Number of challenges conducted at the annual MICCAI conference between 2020 and 2024, categorized by the used hosting platforms.

2 User Stories: Stakeholders and Challenge Lifecycle

To provide an overall picture before discussing technical components, the stakeholders and phases of a typical biomedical imaging challenge are described below.

2.1 Key Roles and Their Needs

A challenge relies on organizers, participants, and data providers, each with essential roles and needs. Following stakeholder listed aren't exhaustive and can be expanded as needed (e.g., with data curators or other specialized roles).

Challenge Organizer The challenge organizer is responsible for establishing the required infrastructure, defining challenge rules and evaluation metrics, managing participant registrations, and ensuring the integrity of the data. Accordingly, organizers need fine-grained access control, robust data security mechanisms, automated evaluation pipelines, and transparent systems for displaying results.

Challenge Participant The challenge participant registers for the competition, downloads the relevant data, develops or refines algorithms, submits solutions, and interprets the resulting evaluation. Participants thus require straightforward access to the challenge data, a user-friendly submission process, timely and frequent feedback on performance and technical issues, and clear documentation outlining the challenge details.

Data Provider The data provider supplies raw or partially processed imaging data, ensuring compliance with privacy and usage agreements, and may also contribute image-related annotations. To fulfill these responsibilities, the data provider needs to ensure that the data are used within ethical and legal bounds, maintain robust logs of data access or downloads, and accommodate potential updates if, for example, incorrect annotations are identified. Although the challenge organizer and data provider roles can be held by the same entity, it is often helpful to treat them as distinct when planning the challenge infrastructure.

2.2 Phases in a Challenge Lifecycle

A biomedical image analysis challenge follows a structured lifecycle, consisting of distinct phases that guide its planning, execution, and outcome analysis.

Planning Phase In this initial phase, organizers secure the necessary ethical approvals and data-sharing permissions, thereby establishing a solid regulatory foundation. The overarching objective of the challenge is then defined, and specific tasks are identified in alignment with the intended research goals. Concurrently, appropriate evaluation metrics are selected or developed to rigorously assess participant performance. Budgetary considerations, infrastructural limitations, and a comprehensive timeline are also determined during this phase.

Pre-Challenge Phase Prior to the challenge launch, substantial preparatory work is undertaken. Datasets are curated and pre-processed, including normalization procedures and strategic splitting into training and validation sets to ensure robust experimental design. Key backend services such as identity and

access management, data storage, and the submission portal, are deployed and pilot-tested to preempt operational issues. In addition, detailed user documentation and tutorials are produced to facilitate a smooth onboarding process for prospective participants.

Challenge Execution Once the challenge is open and running, participants undergo a structured registration and onboarding process that grants them access to the curated dataset. Participants develop and submit their containerized solutions, which are automatically evaluated by a dedicated evaluation system. This phase is supported by ongoing communication through forums, emails, or helpdesk services to address inquiries and provide clarifications. The inclusion of a dynamic leaderboard further enables participants to monitor their performance relative to peers, fostering a competitive and transparent environment.

Post-Challenge Phase Following the challenge, the organizers consolidate and analyze the final results, often generating an official, final leaderboard as a part of this review. Advanced statistical analyses may be employed to ascertain the significance of the top-performing methods. Furthermore, all relevant data, code, and solutions are systematically archived to support reproducibility and facilitate future research endeavors. The outcomes of the challenge are then disseminated through a comprehensive summary paper, which is created together with the participants and which serves as a valuable reference for subsequent collaborative studies.

By outlining the temporal interactions between user roles and specific system components, the requisite infrastructure elements are systematically derived in Section 3.1, and the overall system architecture is presented in Section 3.2.

3 Blueprint: Defining the Challenge Framework

This chapter presents a comprehensive blueprint that translates user requirements into infrastructure components and proposes an integrated system architecture tailored to the needs of a biomedical research challenge.

3.1 Infrastructure Requirements

Based on the previously outlined lifecycle and roles, this section matches the identified organizational needs with the corresponding technical services.

Data Annotation Observing research data management principles ensures structured planning, storage, and reproducibility of datasets [10]. In medical image processing, annotation quality is crucial, as it defines the ground truth for model training and evaluation. Expert dependency and variability require multiple validation steps, while scalability remains a challenge [1]. Because these data typically involve standard medical imaging formats such as Digital Imaging and Communications in Medicine, the annotation environment should be both browser-based and equipped to handle these formats.

Data Distribution and Storage Ensuring secure and consistent access to

large biomedical imaging datasets is essential for many research challenges, particularly when confidentiality requirements must be met. These datasets may involve endoscopic videos or substantial 3D volumes like Computed Tomography or Magnetic Resonance Imaging scans, necessitating a storage solution that is both scalable and capable of limiting access to authorized participants only. In addition, maintaining detailed logs of data downloads is often required for compliance and auditing purposes.

Identity and Access Management (IAM) For effective user management, challenge organizers must control system access while ensuring a smooth registration process. This typically involves robust account creation and verification. Where possible, authentication should be delegated to a trusted institutional identity provider, while authorization should remain clearly separated in a local IAM. Moreover, it is essential to differentiate between user roles (organizers, participants, data providers) and comply with data protection regulations such as the GDPR by minimizing personal data collection and securing user consent [19].

Submission and Evaluation Mechanism The submission and evaluation mechanism is crucial for handling containerized code uploads from participants and automatically evaluating these submissions. This process requires an automated evaluation framework employing script-based methodologies, which may incorporate a dynamic leaderboard for real-time metric visualisation. This follows the recommendations of the Metrics Reloaded framework [15]. Compatibility with experiment-tracking tools enables organisers to manage reproducible experiments effectively [22]. To ensure consistency, stochastic models must use fixed random seeds. The modular architecture enables horizontal scaling to arbitrary throughput. Additionally, version control is important if participants are allowed to submit multiple versions of their work.

Reverse Proxy Ensuring secure external access in a single domain is often essential, especially when multiple microservices must be exposed, and Transport Layer Security (TLS) [31] must be managed. This setup typically requires handling TLS certificate management through external services or institution-issued certificates, and can also demand load balancing or path-based routing if the system comprises multiple services [30].

Documentation and Communication Clear and comprehensive documentation, paired with effective communication channels, is essential to ensure participants can navigate submission processes and use data responsibly, while organizers can promptly address inquiries and disseminate updates. A multi-channel communication strategy enhances engagement and challenge organization [20].

3.2 Proposed System Architecture

In the following, the services described above are integrated into a cohesive system. This provides a high-level schematic of how a user’s request flows from a public domain to secure backend services.

System Overview In the proposed architecture, participants primarily interact through a web portal for registration, information access, and submission

management. Requests pass through a reverse proxy that terminates secure connections and forwards traffic to appropriate back-end services. An identity management system handles authentication and authorization by issuing access tokens. A dedicated storage component hosts training datasets for algorithm development. Submitted solutions are processed by an evaluation engine running containerized or script-based workflows, and results are stored and presented via a leaderboard and associated database.

Integration and Workflow A typical end-to-end scenario could proceed as follows: a new participant visits the website and is directed to an IAM system to create an account and receive a confirmation. During registration, participants provide challenge-related information and upload a signed agreement outlining data usage rights, legal obligations, and ethical considerations prior to the data access. Once participants are logged in, they obtain permission to download the challenge dataset on the designated release date. They then train a model locally and submit a containerized solution through the challenge portal. For multiple submissions, a containerized script automatically computes metrics and logs results in a database. Optionally, a real-time leaderboard displays updated scores, allowing participants to compare outcomes and monitor progress.

4 Case Study: The PhaKIR Challenge

To validate this blueprint, we applied the DIY approach to the PhaKIR challenge within the Endoscopic Vision Challenge (EndoVis) at MICCAI 2024 [27,26]. A curated dataset of anonymized endoscopic images from multiple centers was created, containing several thousand annotated frames [28]. Ground truth annotations were established through a three-step consensus: initial annotation by trained annotators, followed by two expert reviews to ensure consistency.

Timeline The timeline of the challenge, as outlined in Chapter 2.2, was structured into multiple phases. Starting in January, the organizers coordinated with the EndoVis team, obtained regulatory approvals, and defined challenge scope, evaluation criteria, and timelines, concluding planning by March. During the pre-challenge phase (spring to mid-July), registration opened, the annotated dataset was released, and evaluation metrics and labeling instructions were published. The execution phase began in late July with the release of submission guidelines and opening of the submission portal in early September. Participants submitted containerized solutions and methodology reports by late September, followed shortly after by final submissions. The post-challenge phase concluded on October 10 with result presentations, submission reviews, statistical analyses, archival of findings, and a final summary report.

4.1 Infrastructure Setup

Docker was employed for containerization, ensuring isolated execution, consistency, and portability [11]. It offers greater flexibility than traditional virtualization, thus streamlining deployment and scaling for DIY challenges [24]. However,

static configurations may introduce security vulnerabilities [11]. The PhaKIR challenge employed a container-based core platform that integrated open-source infrastructure with challenge services. The organizers operated with a dedicated on-premise server for hosting and evaluating submissions. The implementation based on Chapter 3.1 was exclusively self-hosted and open-source:

Data Annotation The usage of the Computer Vision Annotation Tool (CVAT) [5] streamlines annotations and improves consistency [10]. This dedicated annotation platform facilitated coordinated ground-truth creation, and data pre-processing followed best practices [1].

Data Distribution and Storage To securely provide challenge data, a Secure Shell (SSH) File Transfer Protocol (SFTP) server was implemented, ensuring encrypted file transfer via SSH [14]. Challenge submissions were saved via MinIO [18], an S3-compatible object store which was chosen as the implementation, as it can be efficiently operated in Docker environments [6].

IAM For IAM, Authentik [2] was implemented, an open-source identity provider based on Open Authorization (OAuth) 2.0 [8] that centralizes user management and allows automated control driven by the Representational State Transfer [25] interface. Role-based access control restricted access, requiring users to review and accept the terms of use before approval [14]. It supports secure Single Sign-on and token-based authentication through embedded outposts, reducing redundant accounts and security risks while requiring careful configuration to avoid known vulnerabilities [9,12]. In this case study, no institutional OIDC/SAML identity provider was used; instead, both authentication and authorization were fully handled within the local IAM system.

Submission and Evaluation Python-based scripts computed standard metrics based on [15] and custom metrics for the keypoint estimation task. These scripts were executed automatically upon participant submission, pulling the submissions from the Git repository to predict the test results and comparing them to the ground truth.

Reverse Proxy A nginx proxy [21] secured inbound connections to each back-end service, automatically renewing TLS certificates via Let’s Encrypt [13]. This approach minimized public-facing endpoints and restricted direct server access.

Documentation and Communication A dedicated email account, website, and Git platform facilitate structured information access. WordPress [3], with custom plugins, supports team organization, secure OAuth 2.0 login, and participant interaction [7]. Gitea [4], a lightweight Git platform, enables version-controlled information sharing and serves as the submission management system. These tools streamline communication while reducing administrative effort [20].

4.2 Participant Workflow

After registering, accompanied by a signed data usage agreement, via the Authentik portal, which serves as the OAuth provider, the participants gained access to download the training dataset via an SFTP server with OAuth 2.0 token-based restrictions, secured via Authentik outposts. Submission guidelines and data descriptions were provided on Git. Containerized submissions were

securely uploaded to Gitea and stored in MinIO. Automated Python-based evaluation scripts were then executed on a GPU server to compute the results using the ground truth, annotated with CVAT. For all connections, an nginx reverse proxy provides TLS certificates, automatically renewed via Let’s Encrypt.

4.3 Lessons Learned

The PhaKIR challenge provided several insights, leading to key improvements for the design of future self-hosted DIY challenges:

Submission Issues and Technical Support Many participants submitted invalid Docker containers due to misinterpretation of the guidelines, which required extensive manual review. Future iterations should implement automated pre-submission validation and interactive debugging tools to streamline error detection, reduce administrative effort, and improve the overall user experience.

Real-Time Leaderboard Participants requested dynamic ranking updates, which were not initially planned. Integrating an automated leaderboard in future challenges could enhance transparency and engagement.

Post-Challenge Feedback Despite a survey, insights were restricted by the limited number of responses ($n = 4$). Future approaches should offer incentives for participation or structured debriefings for more comprehensive feedback.

Self-Hosted Infrastructure To prevent system overload, per-team submission limits and container resource quotas were enforced, ensuring stable performance during peak usage. The DIY challenge setup proved effective in data management, authentication, and evaluation, demonstrating a viable alternative to commercial platforms with greater flexibility and security.

5 Discussion

The following discusses the theoretical and practical aspects of our approach.

Blueprint By applying the proposed blueprint and considering the key roles outlined in Chapter 2.1, the stakeholder requirements were fully addressed. Detailed role analysis enabled identification of the essential workflows for successful implementation of challenges. Although the guideline is not exhaustive given the specificity of individual challenges, it provides a flexible, customizable basis.

Case Study The case study shows that the DIY approach offers significant advantages over commercial solutions, particularly in terms of enhanced flexibility and improved data sovereignty. Even basic services demand manual configuration, deployment, and dedicated IT security administration; with guidelines, initial setup takes roughly two weeks of moderate DevOps work, followed by a few hours of maintenance each week. We note that the criteria defined in the Blueprint (Section 3) were intended as design goals to illustrate feasibility within this case study, rather than as formal validation metrics. For future deployments, it would have been possible to configure Authentik to delegate authentication to an institutional OIDC/SAML identity provider, while the local IAM instance continues to handle authorization. Although the blueprint supports experiment

tracking integration (e.g. MLflow), it was not implemented; future work will explore its incorporation. Feedback from challenge participants at the MICCAI 2024 conference consistently confirmed the robust organization and technical implementation of the PhaKIR challenge, thus validating the proposed approach.

6 Conclusion and Outlook

This paper presents a comprehensive blueprint for self-hosting biomedical imaging challenges that addresses key organizational and technical requirements. A representative case study validates the framework’s potential for improved data sovereignty, enhanced customization, and greater cost efficiency relative to conventional solutions. Future research should focus on streamlining deployment, strengthening security, and scaling the approach to more complex challenge designs. Additionally direct integration with public challenge directories, enabling seamless listing and discovery of hosted challenges should be tackled. In general, the proposed framework offers a solid foundation for decentralized institution-controlled solutions in biomedical imaging. The biomedical imaging community is encouraged to refine this blueprint to meet emerging technical and regulatory demands, fostering innovation and progress.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Aljabri, M., AlAmir, M., AlGhamdi, M., Abdel-Mottaleb, M., Collado-Mesa, F.: Towards a better understanding of annotation tools for medical imaging: a survey. *Multimedia Tools and Applications* (2022). <https://doi.org/10.1007/s11042-022-12100-1>
2. Authentik Security Inc.: Authentik. <https://goauthentik.io/> (2025), [Online; accessed 25-February-2025]
3. Automattic Inc.: Wordpress. <https://wordpress.com/> (2025), [Online; accessed 25-February-2025]
4. CommitGo, Inc.: Gitea. <https://about.gitea.com/> (2025), [Online; accessed 25-February-2025]
5. CVAT.ai Corporation: CVAT: Leading Image & Video Data Annotation Platform. <https://www.cvat.ai/> (2025), [Online; accessed 25-February-2025]
6. Factor, M., Meth, K., Naor, D., Rodeh, O., Satran, J.: Object storage: The future building block for storage systems. IBM Haifa Research Laboratories (2004), <https://www.research.ibm.com/haifa/projects/storage/objectstore/>
7. Fernandes, S., Vidyasagar, A.: Digital marketing and wordpress. *Indian Journal of Science and Technology* (2015). <https://doi.org/10.17485/ijst/2015/v8iS4/60375>
8. Fett, D., Küsters, R., Schmitz, G.: A comprehensive formal security analysis of oauth 2.0. In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. pp. 1204–1215 (2016)

9. Fett, D., Küsters, R., Schmitz, G.: A comprehensive formal security analysis of oauth 2.0. CCS'16 (2016), <http://dx.doi.org/10.1145/2976749.2978385>
10. Hanbury, A.: A survey of methods for image annotation. Journal of Visual Languages & Computing (2008). <https://doi.org/10.1016/j.jvlc.2008.01.002>
11. Ibrahim, M.H., Sayagh, M., Hassan, A.E.: A study of how docker compose is used to compose multi-component systems. Empirical Software Engineering **26**, 128 (2021), <https://doi.org/10.1007/s10664-021-10025-1>
12. Indu, I., Anand, P.R., Bhaskar, V.: Identity and access management in cloud environment: Mechanisms and challenges. Engineering Science and Technology, an International Journal **21**, 574–588 (2018), <https://doi.org/10.1016/j.jestch.2018.05.010>
13. Internet Security Research Group: Let's Encrypt - Free SSL/TLS Certificates. <https://letsencrypt.org/> (2025), [Online; accessed 25-February-2025]
14. Krishna, M., Jamwal, P., Chaitanya, K.S.R., Kumar, B.V.: Secure file multi transfer protocol design. Journal of Software Engineering and Applications (2011). <https://doi.org/10.4236/jsea.2011.45034>
15. Maier-Hein, L., Reinke, A., Godau, P., Tizabi, M.D., Buettner, F., Christodoulou, E., Glocker, B., Isensee, F., Kleesiek, J., Kozubek, M., et al.: Metrics reloaded: recommendations for image analysis validation. Nature methods **21**(2), 195–212 (2024)
16. Maier-Hein, L., Reinke, A., Kozubek, M., Martel, A.L., Arbel, T., Eisenmann, M., Hanbury, A., Jannin, P., Müller, H., Onogur, S., Saez-Rodriguez, J., van Ginneken, B., Kopp-Schneider, A., Landman, B.A.: Bias: Transparent reporting of biomedical image analysis challenges. Medical Image Analysis **66**, 101796 (2020). <https://doi.org/10.1016/j.media.2020.101796>
17. Mattsson, U.: Data security: On premise or in the cloud. ISSA Journal (December 2019)
18. MinIO, Inc.: MinIO. <https://min.io/> (2025), [Online; accessed 25-February-2025]
19. Morkonda, S.G., Chiasson, S., van Oorschot, P.C.: Empirical analysis and privacy implications in oauth-based single sign-on systems. 20th Workshop on Privacy in the Electronic Society (WPES '21) (2021), <https://doi.org/10.1145/3463676.3485600>
20. Murphy, M.: Internal strategies for assessing organizational communication channel effectiveness. Walden Dissertations and Doctoral Studies Collection (2017), <https://scholarworks.waldenu.edu/dissertations>
21. nginx: NGINX Proxy Manager. <https://nginxproxymanager.com/> (2025), [Online; accessed 25-February-2025]
22. Quaranta, L., Calefato, F., Lanubile, F., et al.: A taxonomy of tools for reproducible machine learning experiments. In: DP@ AI* IA. pp. 65–76 (2021)
23. Radboud University Medical Center: Grand Challenge. <https://grand-challenge.org/> (2025), [Online; accessed 25-February-2025]
24. Reis, D., Piedade, B., Correia, F.F., Dias, J.P., Aguiar, A.: Developing docker and docker-compose specifications: A developers' survey. IEEE Access **10**, 2318–2326 (2022), <https://doi.org/10.1109/ACCESS.2021.3137671>
25. Richards, R., Richards, R.: Representational state transfer (rest). Pro PHP XML and web services pp. 633–672 (2006)
26. Rueckert, T., Maerkl, R., Rauber, D., Klausmann, L., Gutbrod, M., Reiter, J., Rueckert, D., Feussner, H., Wilhelm, D., Palm, C.: A video benchmark dataset for surgical procedure phase recognition, keypoint estimation, and instrument instance segmentation in endoscopy. In preparation (2025)

27. Rueckert, T., Rauber, D., Maerkl, R., Klasumann, L., Yildiran, S.R., Gutbrod, M., Weber Nunes, D., et al.: Comparative validation of surgical procedure phase recognition, keypoint estimation, and instrument instance segmentation in endoscopy: Results of the PhaKIR 2024 challenge. In preparation (2025)
28. Rueckert, T., et al.: PhaKIR Dataset - Surgical Procedure Phase Recognition, Keypoint Estimation, and Instrument Instance Segmentation (2025). <https://doi.org/10.5281/zenodo.15740620>
29. Sage Bionetworks: Synapse. <https://www.synapse.org/> (2025), [Online; accessed 25-February-2025]
30. Sommerlad, P.: Reverse proxy patterns. Pattern-oriented Software Architecture (2003), <https://www.research.ibm.com/haifa/projects/storage/objectstore/>
31. Turner, S.: Transport layer security. IEEE Internet Computing **18**(6), 60–63 (2014)