KIMERA: From Evaluation-as-a-Service to Evaluation-in-the-Cloud

Andrea Pasin
Università degli Studi di Padova
Padova, Italy
andrea.pasin.1@phd.unipd.it

Nicola Ferro
Università degli Studi di Padova
Padova, Italy
nicola.ferro@unipd.it

Abstract

Experimental evaluation steers the development of Information Retrieval (IR) systems, and large-scale evaluation campaigns provide the field with a common infrastructure to conduct comparable evaluation exercises. Over the years, tools and platforms have been developed to manage and automate these activities, enhance the reproducibility of conducted experiments and facilitate data sharing. In this context, Evaluation-as-a-Service (EaaS) emerged as an approach to avoid distributing experimental collections, which may contain copyrighted or sensitive data, and instead execute containerised code on that data on remote servers. We propose Kubernetes Infrastructure for Managed Evaluation and Resource Access (KIMERA) as the next step from EaaS into Evaluation-in-the-Cloud (EitC), allowing researchers to directly code and execute their systems through their browsers, requiring only an internet connection. Moreover, recent advancements, such as Large Language Models, or new computing paradigms, such as quantum computers, require external third-party services and computational resources. In this respect, KIMERA streamlines and simplifies access to such services on-demand via their APIs. More in detail, KIMERA relies on state-of-the-art containerization and orchestration tools, such as Docker and Kubernetes, to provide a robust, scalable, secure, and fault-tolerant IR evaluation platform. KIMERA monitors and stores all the participants' submissions, accurately keeping track of the resource usage, allowing for evaluating both the efficiency and the effectiveness of the deployed methods. Moreover, all participants can be assigned workspaces sharing the same resources (i.e., CPU and RAM), thus enhancing reproducibility and comparability among systems. Finally, KIMERA has been designed with modularity and extensibility in mind, allowing it to be easily adapted to new use cases and usage scenarios. KIMERA has been developed and adopted in the context of the QuantumCLEF lab, to allow for mixed experiments, comparing approaches running on traditional hardware and on real quantum annealers provided by external companies. KIMERA has also been used as a learning resource to provide Quantum Computing tutorials for IR at major conferences, such as ECIR and SIGIR. The source code of KIMERA is openly available at https://github.com/MjPaxter/KIMERA.

CCS Concepts

• **Hardware → Quantum computation**; • **Software and its engineering → Software architectures**; • **Information systems → Computing platforms**; **Evaluation of retrieval results**; *Retrieval tasks and goals.*

Keywords

Infrastructure, Kubernetes, Docker, Evaluation, Reproducibility, Quantum Computing, Large Language Models, Information Retrieval

ACM Reference Format:
Andrea Pasin and Nicola Ferro. 2025. KIMERA: From Evaluation-as-a-Service to Evaluation-in-the-Cloud. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3726302.3730298

SIGIR '25, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3730298

1 Introduction

For many years, Information Retrieval (IR) systems have been developed and measured following an experimental approach based on the Cranfield paradigm [5, 17, 33]. This has seen the rise of several shared tasks hosted at conferences such as Text REtrieval Conference (TREC) [18], Conference and Labs of the Evaluation Forum (CLEF) [12], NII Testbeds and Community for Information access Research (NTCIR) [34], and more. While, traditionally, task organisers share experimental collections which are downloaded by participants and processed locally to produce system runs, more recently, Evaluation-as-a-Service (EaaS) has emerged as an alternative paradigm [19, 22] where participants submit their code to some infrastructure which executes such code on centrally managed datasets and produces system runs on behalf of the participants. TIRA [13, 15] is a well-known example of such infrastructures in the IR field. One primary motivation for EaaS is to allow running experiments on datasets which cannot be re-distributed because of size, confidentiality or copyright issues. Another motivation can be giving participant access to computing resources, e.g. GPUs, which they may struggle to have access to on their machines. Finally, EaaS ensures more standardisation, reproducibility, and comparability among the submitted systems.

While EaaS brings several advantages to IR evaluation [25, 35], it also has some limitations. First of all, participants need to fully code, execute, and debug their systems on their own machines before uploading some containerised version of them to the EaaS infrastructure for final execution and benchmarking. This may become troublesome, e.g., when participants locally work with a small sample of data and then the code executed centrally produces errors on the full dataset, calling for several iterations of local coding, uploading and central execution to solve issues. Moreover, this process requires participants to be familiar with how containerisation works, possibly adding a (little) entry barrier. This approach also assumes that participants may need to have locally available some (down-scaled) version of the hardware available on the EaaS infrastructure, e.g. GPUs, in order to perform the local development phase. Finally, nowadays, more and more IR pipelines require to access external services during their execution – accessing some proprietary Large Language Model (LLM) via an Application Program Interface (API) is a very common case[1,2,3]. EaaS infrastructures may or may not allow for this either for security reasons or for some design decisions, e.g. avoid online resources in favour of reproducibility. However, even when EaaS infrastructures allow for accessing online services, this would require participants to share their secrets – API keys or credentials – within the code uploaded to the infrastructure, which is way suboptimal. Moreover, this also assumes that invocation of third-party services from the EaaS infrastructure works as the invocation from the participants' local machines, which is possibly true in theory but in practice can undergo any unforeseen circumstances, e.g. different firewalls inbetween, which lead to errors and troubleshooting.

When it comes to accessing third-party services, in this paper, we focus on a use case even more specific than LLMs, namely quantum computers, applied to IR and Recommender Systems (RS) in our case [9, 10]. Quantum computers are not only extremely expensive but also require specialized infrastructure and maintenance, making them difficult to host for most organizations [16, 24], and it is very unlikely that an average IR researcher has direct access to them, somehow limiting the reserach possibilities of our field. To overcome these barriers, an increasing number of companies offer hosted solutions, housing these computing resources in their data centres and providing remote access via APIs, typically with usage quotas, providing a more cost-effective and scalable way to leverage such cutting-edge technology. The need for such hosted solutions became particularly evident when the QuantumCLEF shared task was proposed [28, 30] since it required the integration of quantum resources in the IR systems pipeline. With respect to the general limitations discussed for EaaS infrastructures, QuantumCLEF posed additional challenges. The organizers of the task had access to actual quantum computers but they could not share with participants the secret API key needed to make calls; as a consequence, if using an EaaS infrastructures, participants would have not been able to access the quantum computers during the local development phase, making it very difficult, if not impossible, to actually run the task. Furthermore, QuantumCLEF investigated performance not only in terms of effectiveness but also in terms of efficiency. Thus, it was necessary to accurately monitor the execution time both on quantum computers and traditional hardware and to control and provide the same resources for all participants, making the execution time easily comparable between different systems; both these requirements are not readily satisfied by current EaaS infrastructures.

For all these reasons and to overcome the current limitation in EaaS, we designed and developed Kubernetes Infrastructure for Managed Evaluation and Resource Access (KIMERA), whose code is openly available at https://github.com/MjPaxter/KIMERA under GNU GPL v3.0 license. It is an infrastructure implemented with state-of-the-art containerization and orchestration tools such as Docker and Kubernetes to enhance reproducibility, easy resource access, and comparability. KIMERA has been used throughout the QuantumCLEF challenge to provide monitored access to quantum computers. KIMERA not only represents a step further in accessing third-party services but it also represents a paradigm shift from EaaS to what we could call Evaluation-in-the-Cloud (EitC). In fact, KIMERA is designed to allow participants to program directly using KIMERA itself: they are only required to have a personal device with internet access (it can even be a smartphone) since they can program and execute their systems directly from the browser. Thus participants are not required to have any powerful machines or specialised hardware to develop and test their systems, and they are not required to have any expertise in containerization tools. Moreover, KIMERA allows organisers to set the same computing resources for all the participants, facilitating the monitoring and comparison of both systems' effectiveness and efficiency. Finally, all the data regarding system runs are collected and stored in a database, to provide insights and statistics such as the overall number of submissions or the overall energy consumption. We remark that even though KIMERA has been specifically tailored to a Quantum Computing (QC) scenario, it can be employed in other areas and tasks, such as tasks where participants use computationally expensive models such as LLMs.

Our infrastructure has already been employed to carry out the QuantumCLEF 2024 [29, 31] lab, where it handled almost 1,000 submitted problems from 7 research groups coming from all over the world. Furthermore, it has been used for teaching purposes during the tutorials delivered at ECIR 2024 [9] and SIGIR 2024 [10], in which we created workspaces for more than 60 people who were interested in understanding the possible future implications and applications of QC in the IR and RS fields.

The paper is organized as follows: Section 2 summarizes the related work; Section 3 provides a background of Kubernetes; Section 4 describes the KIMERA architecture and its main characteristics; Section 5 shows its interface and deployment procedure; Section 6 reports practical use-case scenarios, also beyond QuantumCLEF; finally, Section 7 draws some conclusions and outlines future work.

2 Related Work

2.1 Platforms for IR Shared Tasks

Several platforms have been developed to support shared tasks in IR, aiming to reduce the workload for organisers and participants while increasing reproducibility [14]. The adoption of such platforms has advanced reproducibility and scalability in shared tasks, providing structured environments for researchers to evaluate their systems in a fair and standardised way.

Some of the most used platforms nowadays are the following ones: CodaLab, EvalAI, STELLA, and TIRA [15]. These platforms adopt the EaaS paradigm, providing web services for managing and executing evaluations. While CodaLab and EvalAI are cloud-hosted on commercial services such as Microsoft Azure and Amazon S3, STELLA and TIRA are maintained within universities [14].

TIRA [15], which has been further extended into the TiREx [13] platform, has been employed to host the evaluation of different challenges [1–3, 21]. TIRA allows researchers to submit their systems through containers that will be executed in the infrastructure. These systems must comply with given characteristics in order to be properly executed and evaluated in TIRA. TIRA supports modern IR workflows, making TIRA a robust tool for reproducible IR research.

Differently from TIRA, KIMERA allows participants to directly code in the infrastructure via a plain Web browser, managing access

[1]https://openai.com/index/openai-api/
[2]https://groq.com/
[3]https://nebius.com/

to third-party services by keeping secrets and monitoring usage of resources.

2.2 The QuantumCLEF Shared Task

In the last few years, we have witnessed many groundbreaking changes in several fields due to the latest technological advancements. QC is a new field which is attracting significant research interest in the last decade because it is believed that it could further revolutionise the way we solve complex problems by substantially decreasing the time required to solve them [8]. For this reason, QuantumCLEF[4] is an evaluation challenge that tries to investigate possible applications of QC in IR and RS, to understand whether quantum resources can be integrated into traditional system pipelines to boost their performance. Even though we are still in the early stages of QC development, it is already possible to tackle practical problems with these cutting-edge technologies [26].

The QuantumCLEF lab was proposed in 2023 [28, 30] and in its first edition in 2024 [29, 31] has seen the active participation of 7 research groups coming from all over the world, showing comparable performance obtained by traditional systems with respect to quantum-based solutions showing that QC is starting to become useful to solve practical applications. As the hardware improves, it is expected that quantum solutions start to show better performance, especially in terms of efficiency for problems with high computational complexity, such as combinatorial problems [31]. The QuantumCLEF lab has employed KIMERA to monitor the participants' submissions and to provide them easy access to quantum resources.

3 Background on Kubernetes

KIMERA is implemented with Docker containers and Kubernetes as an orchestrator. Kubernetes is an open-source platform designed to automate the deployment, scaling, and management of containerised applications [4]. It helps manage containers by orchestrating the various tasks involved in their deployment, such as load balancing, resource allocation, and network configuration. It also provides advanced features like automatic restart of containers (e.g., in case of errors). Through Kubernetes, it is possible to deploy infrastructure on several nodes, allowing the distribution of the workload on different computers.

Kubernetes offers several objects that can be used to create very complex but well-structured systems. These objects are thought to be used hierarchically, where some of them can be used as *wrappers* for other objects, offering additional functionalities. Some of its most important objects are:

- **Pod**: The smallest deployable unit in Kubernetes, a Pod encapsulates one or more tightly coupled containers that share the same network namespace and storage volumes. Pods are typically used to run a single instance of an application or service. They are ephemeral in nature and managed by higher-level abstractions such as Deployments.
- **Deployment**: A Deployment is a Kubernetes resource used to manage and scale a set of identical Pods. It provides declarative updates, ensuring that the desired state of the application is maintained over time. Deployments are used to handle tasks like rolling updates and rollbacks, making it easier to manage the lifecycle of applications.
- **Service**: A Service provides a stable network identity and exposes a set of Pods as a single endpoint. It enables communication between different components of an application, even as the underlying Pods are replaced or scaled. Services support multiple types of networking, such as ClusterIP (default), NodePort, and LoadBalancer, depending on the use case.
- **PersistentVolumeClaim (PVC)**: A PVC is a request for storage by a user in Kubernetes. It abstracts the details of underlying storage systems and allows applications to dynamically request and consume storage resources.
- **StorageClass**: A StorageClass provides a way to define the characteristics of storage in a Kubernetes cluster, such as performance or backup policies. It enables dynamic provisioning of storage resources, allowing PVCs to automatically create the required storage volumes.
- **Secret**: Secrets are used to store sensitive information such as passwords, API keys, and certificates in Kubernetes. They are designed to be used by Pods securely without hardcoding sensitive data into the application or configuration files.
- **ConfigMap**: A ConfigMap is used to store non-sensitive configuration data in key-value pairs. It allows one to decouple configuration settings from the application code, making it easier to manage and update application settings without rebuilding or redeploying applications.

4 The KIMERA Architecture

The KIMERA infrastructure leverages state-of-the-art containerization and orchestration tools to provide a robust, scalable, and easily accessible IR development and evaluation platform. KIMERA has been implemented with the main objective of allowing participants to have access to computationally expensive resources such as real quantum computers, without the need to stipulate agreements with companies to obtain API keys and quotas. Additionally, through this infrastructure, we aim to lower the entry barrier for any participants by directly providing them with workspaces where they can code and execute their solutions. These workspaces can be accessed directly from the browser, thus removing any hardware requirement for the participants: they can even code and run their approaches on their smartphones. Furthermore, we designed it to be easily accessible, with clean and practical user interfaces and dashboards to monitor available quotas and executions over time. Finally, this infrastructure monitors and collects data about the execution time of the systems, allowing to draw statistics and to compare them both in terms of effectiveness and efficiency.

4.1 The Different Components

KIMERA is implemented with Docker containers and Kubernetes as an orchestrator. It requires the usage of many of the Kubernetes' components. Figure 1 shows the different components of the infrastructure, their role, and how they are interconnected with each other. In particular, it is possible to see that there is only a single access point from outside the infrastructure network. Some of these components require access to permanent storage, such as the Database and the Workspace(s), while the other components do not. This figure

[4]https://qclef.dei.unipd.it/

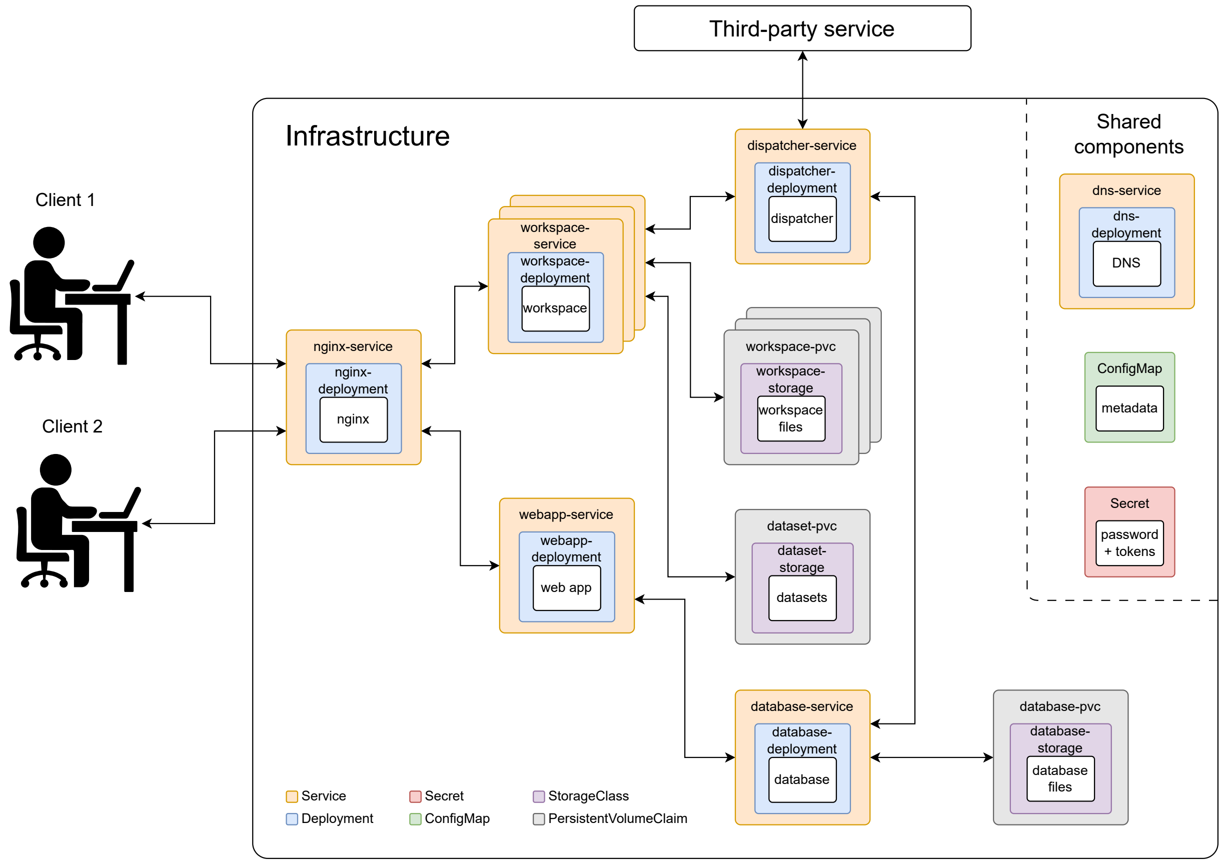


Figure 1: High-level representation of the KIMERA architecture.

also highlights the different levels of abstraction among Kubernetes objects.

4.1.1 nginx. nginx ("engine x") [32] is a high-performance, open-source web server and reverse proxy server designed to handle high traffic with low resource usage. It is widely used for load balancing, serving static content, and acting as a reverse proxy for dynamic web applications. Its event-driven architecture makes it highly scalable and efficient. In our infrastructure, it is used to manage the incoming/outcoming internet traffic. It serves as an HTTP routing mechanism to route traffic to the other components based on the requests, allowing external users to have access to the Web Application or to the Workspaces.

4.1.2 Web Application. The Web Application serves as a public contact point where guests and participants can keep track of the overall ongoing tasks. The Web Application is integrated with the database to show the stored data through practical user interfaces. The Web Application provides a login system that allows registered groups to enter their own private area. In this area, they have access to a dashboard to monitor quotas and the different submissions through lists and charts. Furthermore, the Web Application allows groups to have access to workspaces by providing them with a button that redirects them to the correct URL pointing to their workspace. The Web Application is designed according to the REST API principles.

4.1.3 Workspaces. Each workspace is accessible only to the participating groups through corresponding credentials. Workspaces are directly accessible from the browser, and they provide the participating teams with an environment already pre-configured with the necessary tools to program with. The workspaces are visualised through an interface that resembles Visual Studio Code Integrated Development Environment (IDE), which is fully customizable with extensions and compilers for all programming languages. Since it is directly accessible from the browser, it is possible to program and execute code without having any particular requirements on the machine used (it can even be a smartphone or tablet as long as it is connected to the internet). In our case, since all participants needed to write their solutions in Python, we directly provided the environment configured for Python only. However, if participants want to write code in other languages, it is sufficient to download the proper extensions directly from the available interface. It is important to notice that more than a single person belonging to a group can have access to and work simultaneously on the different files. In fact, usually participating groups are formed by several individuals, and thus it is important to grant the possibility of having access to the workspace at the same time. The suggested good practice is to avoid editing the same file simultaneously to prevent overwriting part of the code that is being written by another person.

All workspaces have a specific amount of CPUs and RAM resources associated. Furthermore, each workspace is associated with its own permanent storage, to store all the created files and configurations by each team. Finally, each workspace has also access to an only-readable folder that contains all the datasets. This prevents a waste of disk space that would happen by copying these resources in all workspaces separately.

While the datasets folder is indeed useful for custom datasets, participants can easily have access to other more standard datasets such as the ones provided by *ir_datasets* [23]. Each workspace comes with Git pre-installed, offering the possibility of making code and submissions freely available in public repositories for enhanced reproducibility [6, 7].

For the QuantumCLEF evaluation lab, we also provided an additional tool called D-Wave Inspector that allows researchers to visualise the status of the qubits when solving a problem. This is an additional feature that we made available inside our infrastructure to give them an even deeper insight into what is happening inside quantum computers when solving a problem.

4.1.4 Dispatcher. The dispatcher is in charge of handling all the API requests that are made by the participants when executing their codes inside the workspaces, formatting the requests correctly, and appending the secret API key to all of them. In this way, it is possible to have access to the computing resources provided by external companies (e.g., D-Wave in the case of QuantumCLEF) while monitoring the resource usage of all participants.

This is a fundamental component that needs to be customised according to the resource needs, since each third-party service may adopt different APIs to provide access to their resources. The dispatcher is also in charge of communicating the results of the API requests to the database to monitor the quotas associated with each group and to store other data that can be useful to draw statistics. Once a research group finishes its quota (e.g., its associated amount of money to use LLMs or its associated QC time), it will no longer be able to make any further API requests, thus ensuring all groups have the same quota.

4.1.5 Database. The infrastructure hosts a PostgreSQL database, which is used to store all the data regarding the tasks. In our case, the data can be, for example, related to the participating groups (i.e., the group name, the group credentials, ...) or related to the execution of the solutions (i.e., running time, description, ...).

The database is crucial to keeping track of the API key usage for each group, thus allowing for the equal partitioning of the quantum resource quotas among them. The database has access to a permanent storage to store all data on the disk persistently. Having a database with all the data related to the participants' submissions and execution times helps in drawing statistics at the end of the shared task, which can also be analysed to further understand how to optimise the infrastructure and the tasks.

Figure 2 illustrates the Entity-Relationship schema of the database. The schema is highly general and can be easily extended with additional attributes without any particular effort. For instance, a CO_2 emissions attribute could be added to the Job entity to track the environmental impact of third-party computing resources, or a Cost attribute could be included to report the expenses incurred by these API calls.

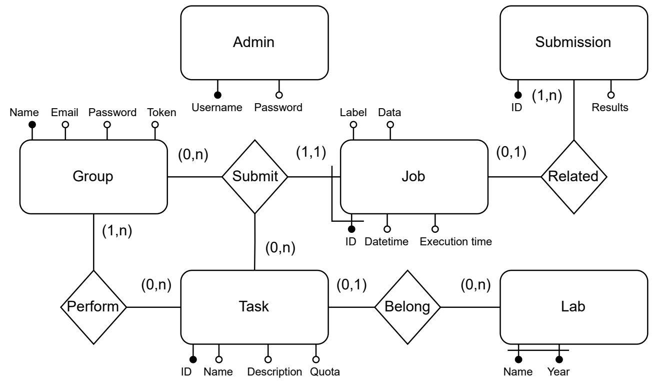


Figure 2: The Entity-Relationship diagram of the database.

4.1.6 DNS. The Domain Name System (DNS) component translates human-readable domain names into IP addresses, enabling computers to locate and communicate with each other over the infrastructure network. In our case, it is critical for the communication between the infrastructure components. It allows them to resolve Service names to their corresponding IPs, enabling the interaction in this dynamic environment. At the core of this component, there is a CoreDNS container acting as a DNS server for the whole infrastructure.

4.1.7 ConfigMap and Secret. These components help for a good organisation of all metadata in the infrastructure. In particular, the ConfigMap holds configuration data while the Secret component holds private data such as passwords or API tokens used in the infrastructure's components.

4.2 Typical workflows

Figure 3 illustrates two typical workflows, referred to as *Workflow A* and *Workflow B*, which are commonly performed within our infrastructure. These workflows represent a series of interactions between infrastructure components to carry out specific tasks.

4.2.1 Workflow A: Code Execution and API Utilisation. Workflow A represents a scenario where the research group is primarily engaged in coding and executing their code within the associated workspace. The workflow involves the following steps:

(1A) **Coding and Execution**: the research group develops their code within the workspace. All coding-related actions are translated into a series of HTTP requests that are done towards our infrastructure, reaching the nginx component.
(2A) **Forwarding**: all requests are forwarded by the nginx component to the corresponding workspace for processing.
(3A) **API-service**: when the code execution requires interaction with external API-based services (e.g., quantum computers, LLMs, etc.), the workspace generates the appropriate API calls to the dispatcher.
(4A) **API-wrapping**: the dispatcher securely appends the necessary API tokens to each request, acting as a wrapper for the API calls. It then forwards these requests to external services that can be provided by third-party companies.
(5A) **API-responses**: the results of the API-based services are returned to the dispatcher.

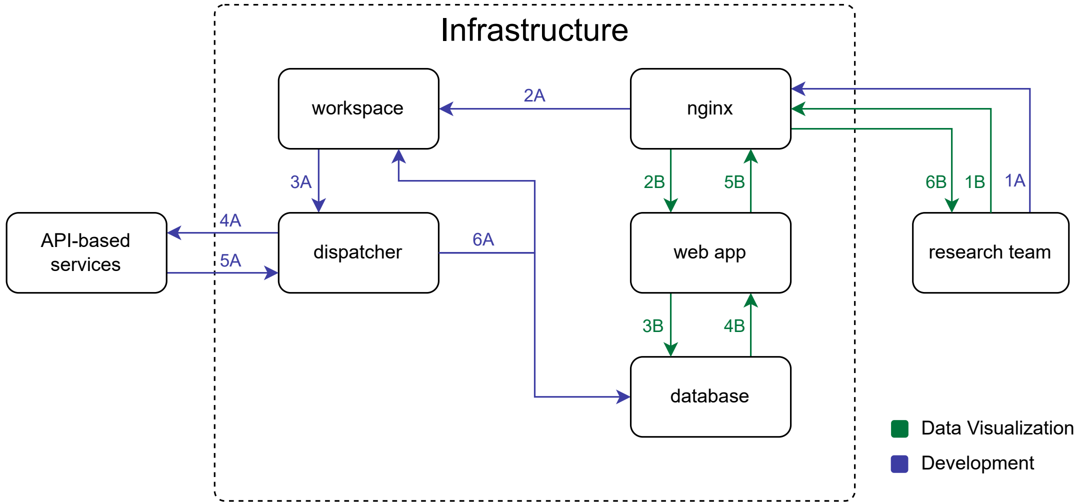


Figure 3: Representation of two possible workflows that can be performed in KIMERA.

(6A) **Results**: the results of the API-based services are returned to the workspace for further processing or analysis. Additionally, the results and metadata related to API usage are stored in the database. This enables monitoring and tracking of API usage by the research group for operational and analytical purposes.

4.2.2 Workflow B: Data Visualisation and Dashboard Interaction. *Workflow B* describes the process where the research group performs data visualisation tasks, such as monitoring API usage or interacting with the dashboard available in the Web Application. The workflow includes the following steps:

(1B) **Browsing**: the research group wants to access the web application to visualise data or interact with the dashboard. Thus, it proceeds by making HTTP requests that reach the nginx component.
(2B) **Forwarding**: the research group requests are forwarded by the nginx component, reaching the Web Application.
(3B) **Querying the database**: the Web Application does not store any permanent data: everything is stored in the database. Thus it proceeds to make the opportune request to retrieve the data from the database.
(4B) **Data retrieval**: the data is retrieved from the database and returned to the Web Application.
(5B) **Data formatting**: the Web Application formats the data in an opportune way and returns it to the nginx component with HTTP responses.
(6B) **Data visualisation**: the HTTP responses are forwarded to the research group, that will then be able to visualise the data with charts and dashboards.

4.3 Characteristics of the Infrastructure

The development of the infrastructure through Docker and Kubernetes allows for several advantages, enhancing scalability, reproducibility, and easy access to computing resources.

4.3.1 Scalability. The KIMERA infrastructure can be deployed on just a single machine, provided that the machine has enough resources to host all groups. However, Kubernetes allows to easily scale the infrastructure by deploying its components on several machines, making it possible to host a large number of participants on several machines. A key factor that needs to be taken into account is that to measure efficiency (i.e., execution time) in the fairest way possible, all the machines where the workspaces are deployed must share the same hardware configuration.

Furthermore, if the Web Application or the Dispatcher are under heavy load, it is possible to create more instances of these components (called *replicates*) and the traffic will be forwarded automatically for load-balancing purposes, thus allowing more participants to take part in the proposed shared tasks.

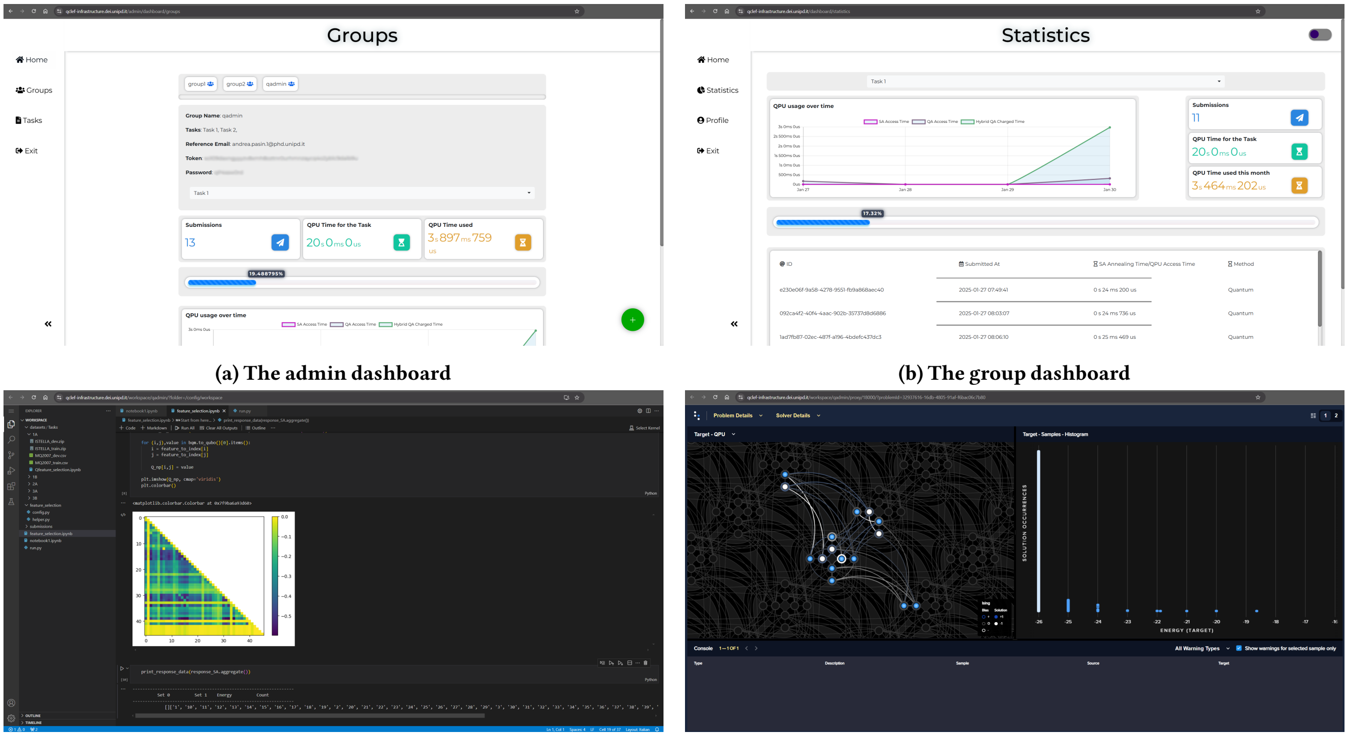
This is also applicable to the Database component. However, this requires careful management to avoid incurring in data inconsistencies due to race conditions and concurrent modifications.

4.3.2 Error-handling. While developing solutions for the proposed tasks, participants may incur errors such as Out-Of-Memory errors or other critical failures that can compromise the workspaces. However, this is not a concern since Kubernetes automatically handles these issues by identifying the crashed workspaces and restarting them within seconds. In this way, the infrastructure provides the maximum level of availability.

Moreover, if the infrastructure is distributed across multiple machines and one of them goes offline, Kubernetes automatically migrates the affected components to the remaining online machines. This process adheres to load-balancing principles, preventing any machine from becoming overloaded.

Finally, if any other components encounter issues that cause them to crash, Kubernetes promptly restarts them, maintaining a resilient and reliable system.

4.3.3 Resource monitoring. The resources of each component are constantly being monitored by Kubernetes to enhance load-balancing



(a) The admin dashboard

(b) The group dashboard

(c) The programming interface

(d) The D-Wave Inspector

Figure 4: The KIMERA User Interface that participants use to interact with the infrastructure during the QuantumCLEF initiative.

and fault-tolerance. It is possible to specify the minimum and maximum amount of resources that each component of the infrastructure can use. This ensures that each workspace is given the same resources to achieve very good levels of comparability and reproducibility, also when considering efficiency.

4.3.4 Security. The infrastructure adopts multiple layers of security measures to safeguard user data and system integrity. Firstly, all communication between clients and the server is conducted over HTTPS, encrypting data in transit. This encryption helps protect sensitive information from being intercepted during transmission. In addition to secure communication, access to the infrastructure's internal components is tightly controlled. Rather than exposing the entire system, only specific APIs are made accessible, and these are served through a NGINX server. Finally, user workspaces are secured with passwords. This ensures that only authorised users can access and interact with the content, providing an extra layer of protection against unauthorised access and maintaining the confidentiality of group data.

4.3.5 Easy access. A key advantage of our infrastructure is its ability to significantly lower access barriers. Existing platforms often require researchers to develop their solutions on personal machines and then upload them—for example in the form of containers—for final execution on the infrastructure. While this approach is practical, it can still pose challenges for researchers who lack adequate hardware resources for the development and debugging of their software or familiarity with containerization technologies. In contrast, our infrastructure offers a user-friendly environment where solutions can be developed and tested directly. This eliminates the need for specialized knowledge or advanced hardware, making the platform accessible to virtually anyone.

5 Running Prototype

In this section, we show what the User Interface looks like and provide the instructions on how to deploy KIMERA for a possible shared task.

5.1 User Interface

Figure 4 shows some screenshots of the User Interface that participants viewed when interacting with KIMERA during the QuantumCLEF 2024 lab [5]. In particular, Figure 4a shows the admin dashboard, where it is possible to retrieve, visualise, and edit the groups' data. Once logged in, research groups can access their dashboard (see Figure 4b) showing data regarding their quotas and solved problems. This allows them to monitor their access to external services to avoid consuming all the available quota. Users can also have access to their workspace with a link that is available in their personal area. Figure 4c shows the programming interface that users see when coding and executing their systems in the provided workspace. As it is possible to see, it resembles the Visual Studio Code interface. On the left, it is possible to access the file system. By using the buttons available on the left vertical bar, it is possible to fully customise the working environment and interface to accommodate all the user's

[5] https://qclef-infrastructure.dei.unipd.it/

needs (e.g., install compilers for different languages, change the interface colouring scheme and fonts, ...). It is also possible to open one or more terminal shells. Finally, Figure 4d shows the interface that we provided to users to examine the status of the qubits inside the quantum computers for the solved problem. This gives really informative insights about how the problem was translated into the physical graph of interconnected qubits, making participants observe what actually happens inside quantum computers.

5.2 Deployment and Usage

To deploy KIMERA, the organisers must first set up the base infrastructure that consists of the Web Application, the Database, and the Dispatcher. This is done automatically by running a few Kubernetes commands that are used to bootstrap the infrastructure, load the Docker containers associated with the aforementioned components, and configure them properly inside the infrastructure according to the definitions that are provided by appropriate `.yaml` files [6]. These files are all already available and define how these components should behave. Furthermore, in a similar way, an initial configuration of the nginx proxy server is created. The organisers can upload the datasets in a read-only folder that will be shared among all participating teams. At this point the basic version of KIMERA is up and running, however, there are not yet any workspaces available. Thus, through the Web Application, the organisers can enter an administration area where they can define the tasks and quotas (in the case of QuantumCLEF, the number of seconds of quantum computation available). In this area, it is also possible to register groups and create their credentials. All of this is done by just compiling some web forms.

The organisers are then supposed to create a workspace for each group and update the network configuration of the nginx component. This is done through some Python scripts that are already available. While at the moment the organisers are supposed to run these scripts manually (i.e., from the terminal), in the future we plan to update the Web Application to make it run the scripts automatically when registering a group.

At this moment, the group has access to its workspace and private area and can proceed with the development and execution of experiments in their environment. In the meanwhile, the organisers wait until the shared task finishes and the group provides some final submissions. During the task, the organisers have access at any time to a dashboard that shows the overall resource usage and the usage associated with each group. This is useful to have real-time monitoring of the resources, for example, to understand whether there can be any issues with the given quotas. On the other side, the groups can have access to their dashboards and to their workspaces to program and execute their solutions.

Once the shared task is finished, the groups will simply leave the final submissions inside a folder called *submissions* in their workspace. Each folder can be easily accessed by the organisers to read and load the submission files and proceed with their evaluation. The scores will then be saved in the database and reported to the teams.

6 Use Cases

In this section, we discuss the two use-case scenarios where our infrastructure has been successfully employed: the QuantumCLEF shared task and the Quantum Computing tutorials delivered at ECIR 2024 and SIGIR 2024. We also discuss how the infrastructure can be easily extended to carry out other evaluation tasks beyond QuantumCLEF.

[6]https://github.com/MjPaxter/KIMERA

Table 1: The teams who participated and submitted at QuantumCLEF 2024.

Team	Affiliation	Country
BIT.UA	IEETA/DETI, LASI, University of Aveiro	Portugal
CRUISE	RMIT University	Australia
NICA	Iran University of Science and Technology, Departement of Computer Engineering	Iran, Islamic Republic Of
OWS	Friedrich Schiller Universität Jena	Germany
qIIMAS	Universidad Nacional Autonoma de Mexico	Mexico
QTB	Universidad Tecnologica de Bolivar	Colombia
shm2024	Madras Christian College, Chennai	India

6.1 QuantumCLEF

Our infrastructure has been used for the first edition of the QuantumCLEF shared task in 2024 [29, 31]. The shared task has seen the active participation of 7 research groups that developed quantum-based solutions for IR and RS tasks, comparing them with traditional approaches. Table 1 shows the 7 teams who actively took part in the QuantumCLEF 2024 lab. These are the teams that delivered final submissions for the tasks, successfully finding some quantum-based solutions for the proposed tasks.

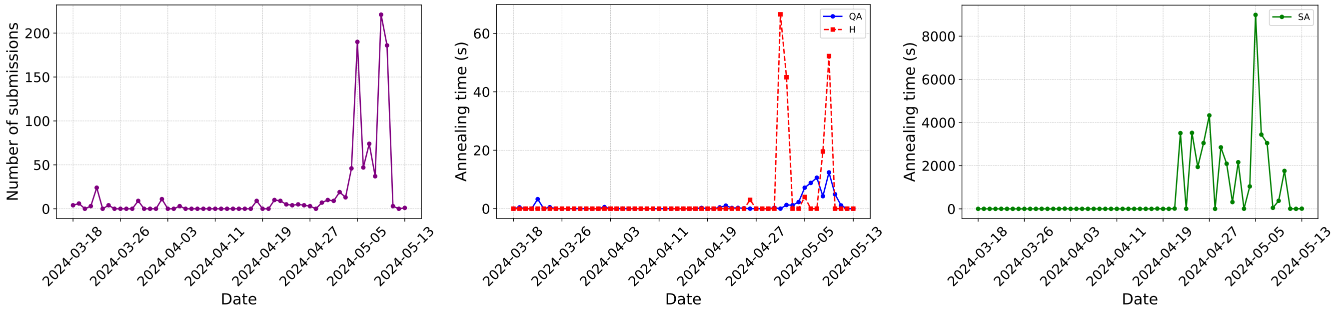
Figure 5 shows the number of submissions over time, with insights also regarding the execution time of quantum-based approaches (marked with **QA** and **H**) and traditional-based approaches (marked with **SA**). As it is possible to see, the participating teams tend to work especially right before the deadline of the task. In particular, our infrastructure handled a total of 976 submissions, requiring more than 12 hours of execution time using traditional hardware, while approximately 4 minutes of QC resources.

This has been a very practical use-case scenario where the infrastructure has been put under heavy load at certain moments. During the QuantumCLEF lab, the infrastructure was deployed on a *m6a.8xlarge* AWS EC2 instance equipped with an AMD EPYC 7R13 processor (32 CPU cores) together with 128 GB of RAM. Each participating team was given a maximum amount of 10 GB of RAM and 1200 CPU millicores.

6.2 Tutorials

Our infrastructure has also been employed for teaching purposes during the ECIR 2024 [9] and SIGIR 2024 [10] conferences. In fact, during these conferences, two tutorials were held about an introduction to QC and Quantum Annealing with specific possible insights into their applications to the IR and RS fields.

The introductory part of the tutorials focused primarily on discussions and did not require any hands-on participation from the attendees. In contrast, the second part featured live coding sessions, where participants were expected to engage directly with quantum computers. However, teaching participants how to use quantum resources in this setting posed a significant challenge, as each individual would have required a personal API key to access them,



(a) Number of submissions over time **(b) Distribution of the Annealing time (QA and H)** **(c) Distribution of Annealing time (SA)**

Figure 5: The distribution of the participating teams' submissions over time, considering also the Annealing time used per day.

which can usually be obtained after the stipulation of an agreement with the company offering these resources.

Our infrastructure was employed to support and streamline the tutorials to address this limitation. Several dedicated workspaces were created, enabling participants—many of whom were interacting with quantum computers for the first time—to gain practical experience. This not only allowed them to explore real quantum hardware in a controlled environment but also demonstrated that KIMERA can be effectively utilised for educational purposes, including tutorials and hands-on training, especially when access to specialised computational resources is restricted.

6.3 IR Evaluation Beyond QuantumCLEF

While our infrastructure has been tailored to QuantumCLEF, it can be easily customised to be adopted for other purposes. Firstly, KIMERA could also be employed to provide resource access to individual research groups requiring computing resources beyond evaluation campaigns. This would allow them to easily study and execute experiments in already configured environments that provide access to powerful computing resources while enhancing reproducibility and comparability of their work. To support this, we plan to introduce an automated evaluation component that can be called from within the workspaces to evaluate the submissions without the intervention of external organisers. We have not yet introduced this feature because we believe this is not a good approach for evaluation campaigns, since participating teams may attempt to repeatedly test their models on the evaluation set, thus tailoring their approach to those specific results rather than ensuring broader generalisation.

Secondly, there exist shared tasks that allow or even require [11, 20] participants to use LLMs. KIMERA could be employed to provide participants with access to these powerful models hosted by external companies due to the computational resource requirements. In that case, it would only be necessary to refine the dispatcher accounting for the different API calls without needing other major architectural changes. Finally, KIMERA could be used to support a wide range of shared tasks not involving the usage of external computing resources. Its user-friendly design, which lowers barriers in both hardware accessibility and required technical expertise, makes it an inclusive platform for participants with varying levels of experience. By enabling broader participation, KIMERA would contribute to more diverse, high-quality submissions and enhance the overall robustness and reliability of shared task results. This expanded accessibility could also foster innovation by encouraging novel approaches from researchers who might otherwise lack the necessary computational resources.

7 Conclusions

In this paper, we presented KIMERA, an infrastructure that can be employed to transition from EaaS to EitC in IR, providing easy and monitored access to computationally expensive resources while lowering the entry barrier in terms of hardware and technical expertise. KIMERA has been used in the QuantumCLEF shared task to give access to participants to quantum resources, enhancing reproducibility, comparability, resource access, and scalability.

While the infrastructure has been tailored specifically to the QuantumCLEF needs, in this paper we have shown that it can be easily customised to further serve as a user-friendly evaluation infrastructure for a wide variety of shared tasks and for individual research purposes. Due to the rise of many quota-based platforms where API keys are used to provide access to resources such as quantum computers and LLMs, our infrastructure can be useful to monitor the quotas given to participants without requiring them to stipulate contracts with these platforms to get their own API keys. Moreover, KIMERA has been developed with the aim of lowering the hardware entry barrier for participating teams in shared tasks by providing them with a complete and fully customizable environment to code and test their systems, without requiring any knowledge of containerization tools.

In the future, we plan to further optimise the infrastructure, making it even easier for organisers to manage all the activities associated with shared tasks. We also intend to refine its components to allow for automated evaluation procedures, thus allowing individual research groups to use KIMERA beyond shared tasks for enhanced resource access, comparability, and reproducibility. Finally, we plan to use this infrastructure also during the second edition of the QuantumCLEF shared task in 2025 [27] , gathering even more data and insights about QC.

Acknowledgments

We acknowledge the CINECA award under the ISCRA initiative for the availability of high-performance computing resources and support.

References

[1] Abinew Ali Ayele, Nikolay Babakov, Janek Bevendorff, Xavier Bonet Casals, Berta Chulvi, Daryna Dementieva, Ashaf Elnagar, Dayne Freitag, Maik Fröbe, Damir Korencic, Maximilian Mayerl, Daniil Moskovskiy, Animesh Mukherjee, Alexander Panchenko, Martin Potthast, Francisco Rangel, Naquee Rizwan, Paolo Rosso, Florian Schneider, Alisa Smirnova, Efstathios Stamatatos, Elisei Stakovskii, Benno Stein, Mariona Taulé, Dmitry Ustalov, Xintong Wang, Matti Wiegmann, Seid Muhie Yimam, and Eva Zangerle. 2024. Overview of PAN 2024: Multi-author Writing Style Analysis, Multilingual Text Detoxification, Oppositional Thinking Analysis, and Generative AI Authorship Verification Condensed Lab Overview. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 15th International Conference of the CLEF Association, CLEF 2024, Grenoble, France, September 9-12, 2024, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 14959)*, Lorraine Goeuriot, Philippe Mulhem, Georges Quénot, Didier Schwab, Giorgio Maria Di Nunzio, Laure Soulier, Petra Galuscáková, Alba García Seco de Herrera, Guglielmo Faggioli, and Nicola Ferro (Eds.). Springer, 231–259. doi:10.1007/978-3-031-71908-0_11

[2] Janek Bevendorff, Ian Borrego-Obrador, Mara Chinea-Rios, Marc Franco-Salvador, Maik Fröbe, Annina Heini, Krzysztof Kredens, Maximilian Mayerl, Piotr Pezik, Martin Potthast, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, Benno Stein, Matti Wiegmann, Magdalena Wolska, and Eva Zangerle. 2023. Overview of PAN 2023: Authorship Verification, Multi-Author Writing Style Analysis, Profiling Cryptocurrency Influencers, and Trigger Detection - Condensed Lab Overview. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 14th International Conference of the CLEF Association, CLEF 2023, Thessaloniki, Greece, September 18-21, 2023, Proceedings (Lecture Notes in Computer Science, Vol. 14163)*, Avi Arampatzis, Evangelos Kanoulas, Theodora Tsikrika, Stefanos Vrochidis, Anastasia Giachanou, Dan Li, Mohammad Aliannejadi, Michalis Vlachos, Guglielmo Faggioli, and Nicola Ferro (Eds.). Springer, 459–481. doi:10.1007/978-3-031-42448-9_29

[3] Alexander Bondarenko, Maik Fröbe, Johannes Kiesel, Ferdinand Schlatt, Valentin Barrière, Brian Ravenet, Léo Hemamou, Simon Luck, Jan Heinrich Reimer, Benno Stein, Martin Potthast, and Matthias Hagen. 2023. Overview of Touché 2023: Argument and Causal Retrieval. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023 (CEUR Workshop Proceedings, Vol. 3497)*, Mohammad Aliannejadi, Guglielmo Faggioli, Nicola Ferro, and Michalis Vlachos (Eds.). CEUR-WS.org, 3065–3089. https://ceur-ws.org/Vol-3497/paper-259.pdf

[4] Brendan Burns, Joe Beda, Kelsey Hightower, and Lachlan Evenson. 2022. *Kubernetes: up and running.* " O'Reilly Media, Inc.".

[5] C. W. Cleverdon. 1967. The Cranfield Tests on Index Languages Devices. *Aslib Proceedings* 19, 6 (1967), 173–194.

[6] Christian S. Collberg and Todd A. Proebsting. 2016. Repeatability in computer systems research. *Commun. ACM* 59, 3 (2016), 62–69. doi:10.1145/2812803

[7] Roberto Di Cosmo and Stefano Zacchiroli. 2017. Software Heritage: Why and How to Preserve Software Source Code. In *Proceedings of the 14th International Conference on Digital Preservation, iPRES 2017, Kyoto, Japan, September 25-29, 2017*, Shoichiro Hara, Shigeo Sugimoto, and Makoto Goto (Eds.). https://hdl.handle.net/11353/10.931064

[8] Maurizio Ferrari Dacrema, Fabio Moroni, Riccardo Nembrini, Nicola Ferro, Guglielmo Faggioli, and Paolo Cremonesi. 2022. Towards Feature Selection for Ranking and Classification Exploiting Quantum Annealers. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 2814–2824. doi:10.1145/3477495.3531755

[9] Maurizio Ferrari Dacrema, Andrea Pasin, Paolo Cremonesi, and Nicola Ferro. 2024. Quantum Computing for Information Retrieval and Recommender Systems. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part V (Lecture Notes in Computer Science, Vol. 14612)*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer, 358–362. doi:10.1007/978-3-031-56069-9_47

[10] Maurizio Ferrari Dacrema, Andrea Pasin, Paolo Cremonesi, and Nicola Ferro. 2024. Using and Evaluating Quantum Computing for Information Retrieval and Recommender Systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (Eds.). ACM, 3017–3020. doi:10.1145/3626772.3661378

[11] Nicola Ferro, Julio Gonzalo, Jussi Karlgren, and Henning Müller. 2024. The CLEF 2024 Monster Track: One Lab to Rule Them All. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part VI (Lecture Notes in Computer Science, Vol. 14613)*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer, 11–18. doi:10.1007/978-3-031-56072-9_2

[12] N. Ferro and C. Peters (Eds.). 2019. *Information Retrieval Evaluation in a Changing World – Lessons Learned from 20 Years of CLEF*. The Information Retrieval Series, Vol. 41. Springer International Publishing, Germany.

[13] Maik Fröbe, Jan Heinrich Reimer, Sean MacAvaney, Niklas Deckers, Simon Reich, Janek Bevendorff, Benno Stein, Matthias Hagen, and Martin Potthast. 2023. The Information Retrieval Experiment Platform. *CoRR* abs/2305.18932 (2023). doi:10.48550/ARXIV.2305.18932 arXiv:2305.18932

[14] Maik Fröbe, Jan Heinrich Reimer, Sean MacAvaney, Niklas Deckers, Simon Reich, Janek Bevendorff, Benno Stein, Matthias Hagen, and Martin Potthast. 2024. The Information Retrieval Experiment Platform (Extended Abstract). In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024.* ijcai.org, 8405–8410. https://www.ijcai.org/proceedings/2024/931

[15] Maik Fröbe, Matti Wiegmann, Nikolay Kolyada, Bastian Grahm, Theresa Elstner, Frank Loebe, Matthias Hagen, Benno Stein, and Martin Potthast. 2023. Continuous Integration for Reproducible Shared Tasks with TIRA.io. In *Advances in Information Retrieval - 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2-6, 2023, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 13982)*, Jaap Kamps, Lorraine Goeuriot, Fabio Crestani, Maria Maistro, Hideo Joho, Brian Davis, Cathal Gurrin, Udo Kruschwitz, and Annalina Caputo (Eds.). Springer, 236–241. doi:10.1007/978-3-031-28241-6_20

[16] Muhammed Golec, Emir Sahin Hatay, Mustafa Golec, Murat Uyar, Merve Golec, and Sukhpal Singh Gill. 2024. Quantum cloud computing: Trends and challenges. *Journal of Economy and Technology* (2024).

[17] D. K. Harman. 2011. *Information Retrieval Evaluation.* Morgan & Claypool Publishers, USA.

[18] D. K. Harman and E. M. Voorhees (Eds.). 2005. *TREC. Experiment and Evaluation in Information Retrieval.* MIT Press, Cambridge (MA), USA.

[19] Frank Hopfgartner, Allan Hanbury, Henning Müller, Ivan Eggel, Krisztian Balog, Torben Brodt, Gordon V. Cormack, Jimmy Lin, Jayashree Kalpathy-Cramer, Noriko Kando, Makoto P. Kato, Anastasia Krithara, Tim Gollub, Martin Potthast, Evelyne Viegas, and Simon Mercer. 2018. Evaluation-as-a-Service for the Computational Sciences: Overview and Outlook. *ACM J. Data Inf. Qual.* 10, 4 (2018), 15:1–15:32. doi:10.1145/3239570

[20] Jussi Karlgren, Luise Dürlich, Evangelia Gogoulou, Liane Guillou, Joakim Nivre, Magnus Sahlgren, Aarne Talman, and Shorouq Zahra. 2024. Overview of ELOQUENT 2024 - Shared Tasks for Evaluating Generative Language Model Quality. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 15th International Conference of the CLEF Association, CLEF 2024, Grenoble, France, September 9-12, 2024, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 14959)*, Lorraine Goeuriot, Philippe Mulhem, Georges Quénot, Didier Schwab, Giorgio Maria Di Nunzio, Laure Soulier, Petra Galuscáková, Alba García Seco de Herrera, Guglielmo Faggioli, and Nicola Ferro (Eds.). Springer, 53–72. doi:10.1007/978-3-031-71908-0_3

[21] Johannes Kiesel, Çagri Çöltekin, Maximilian Heinrich, Maik Fröbe, Milad Alshomary, Bertrand De Longueville, Tomaz Erjavec, Nicolas Handke, Matyás Kopp, Nikola Ljubesic, Katja Meden, Nailia Mirzakhmedova, Vaidas Morkevicius, Theresa Reitis-Münstermann, Mario Scharfbillig, Nicolas Stefanovitch, Henning Wachsmuth, Martin Potthast, and Benno Stein. 2024. Overview of Touché 2024: Argumentation Systems. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 15th International Conference of the CLEF Association, CLEF 2024, Grenoble, France, September 9-12, 2024, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 14959)*, Lorraine Goeuriot, Philippe Mulhem, Georges Quénot, Didier Schwab, Giorgio Maria Di Nunzio, Laure Soulier, Petra Galuscáková, Alba García Seco de Herrera, Guglielmo Faggioli, and Nicola Ferro (Eds.). Springer, 308–332. doi:10.1007/978-3-031-71908-0_14

[22] Jimmy Lin and Miles Efron. 2013. Evaluation as a service for information retrieval. *SIGIR Forum* 47, 2 (2013), 8–14. doi:10.1145/2568388.2568390

[23] Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. 2021. Simplified Data Wrangling with ir_datasets. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 2429–2436. doi:10.1145/3404835.3463254

[24] Nicolas Maring, Andreas Fyrillas, Mathias Pont, Edouard Ivanov, Eric Bertasi, Mario Valdivia, and Jean Senellart. 2023. One Nine Availability of a Photonic Quantum Computer on the Cloud Toward HPC Integration. In *IEEE International Conference on Quantum Computing and Engineering, QCE 2023, Bellevue, WA, USA, September 17-22, 2023*, Brian La Cour, Lia Yeh, and Marek Osinski (Eds.). IEEE, 112–116. doi:10.1109/QCE57702.2023.10193

[25] Henning Müller and Allan Hanbury. 2019. EaaS: Evaluation-as-a-Service and Experiences from the VISCERAL Project. In *Information Retrieval Evaluation in a Changing World - Lessons Learned from 20 Years of CLEF*, Nicola Ferro and Carol Peters (Eds.). The Information Retrieval Series, Vol. 41. Springer, 161–173. doi:10.1007/978-3-030-22948-1_6

[26] Riccardo Nembrini, Maurizio Ferrari Dacrema, and Paolo Cremonesi. 2021. Feature Selection for Recommender Systems with Quantum Computing. *Entropy* 23, 8 (2021), 970. doi:10.3390/E23080970

[27] Andrea Pasin, Maurizio Ferrari Dacrema, Paolo Cremonesi, Washington Cunha, Marcos André Gonçalves, and Nicola Ferro. 2025. QuantumCLEF 2025 - The Second Edition of the Quantum Computing Lab at CLEF. In *Advances in Information Retrieval - 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6-10, 2025, Proceedings, Part V (Lecture Notes in Computer Science, Vol. 15576).*

[28] Andrea Pasin, Maurizio Ferrari Dacrema, Paolo Cremonesi, and Nicola Ferro. 2023. qCLEF: A Proposal to Evaluate Quantum Annealing for Information Retrieval and Recommender Systems. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 14th International Conference of the CLEF Association, CLEF 2023, Thessaloniki, Greece, September 18-21, 2023, Proceedings (Lecture Notes in Computer Science, Vol. 14163)*, Avi Arampatzis, Evangelos Kanoulas, Theodora Tsikrika, Stefanos Vrochidis, Anastasia Giachanou, Dan Li, Mohammad Aliannejadi, Michalis Vlachos, Guglielmo Faggioli, and Nicola Ferro (Eds.). Springer, 97–108. doi:10.1007/978-3-031-42448-9_9

[29] Andrea Pasin, Maurizio Ferrari Dacrema, Paolo Cremonesi, and Nicola Ferro. 2024. Overview of QuantumCLEF 2024: The Quantum Computing Challenge for Information Retrieval and Recommender Systems at CLEF. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 15th International Conference of the CLEF Association, CLEF 2024, Grenoble, France, September 9-12, 2024, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 14959)*, Lorraine Goeuriot, Philippe Mulhem, Georges Quénot, Didier Schwab, Giorgio Maria Di Nunzio, Laure Soulier, Petra Galuscáková, Alba García Seco de Herrera, Guglielmo Faggioli, and Nicola Ferro (Eds.). Springer, 260–282. doi:10.1007/978-3-031-71908-0_12

[30] Andrea Pasin, Maurizio Ferrari Dacrema, Paolo Cremonesi, and Nicola Ferro. 2024. QuantumCLEF - Quantum Computing at CLEF. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part V (Lecture Notes in Computer Science, Vol. 14612)*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer, 482–489. doi:10.1007/978-3-031-56069-9_66

[31] Andrea Pasin, Maurizio Ferrari Dacrema, Paolo Cremonesi, and Nicola Ferro. 2024. QuantumCLEF 2024: Overview of the Quantum Computing Challenge for Information Retrieval and Recommender Systems at CLEF. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), Grenoble, France, 9-12 September, 2024 (CEUR Workshop Proceedings, Vol. 3740)*, Guglielmo Faggioli, Nicola Ferro, Petra Galuscáková, and Alba García Seco de Herrera (Eds.). CEUR-WS.org, 3032–3053. https://ceur-ws.org/Vol-3740/paper-297.pdf

[32] Will Reese. 2008. Nginx: the high-performance web server and reverse proxy. *Linux Journal* 2008, 173 (2008), 2.

[33] Stephen Robertson. 2008. On the history of evaluation in IR. *J. Inf. Sci.* 34, 4 (2008), 439–456. doi:10.1177/0165551507086989

[34] T. Sakai, D. W. Oard, and N. Kando (Eds.). 2021. *Evaluating Information Retrieval and Access Tasks – NTCIR's Legacy of Research Impact.* The Information Retrieval Series, Vol. 43. Springer International Publishing, Germany.

[35] Ellen M. Voorhees, Jimmy Lin, and Miles Efron. 2014. On run diversity in Evaluation as a Service. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, Shlomo Geva, Andrew Trotman, Peter Bruza, Charles L. A. Clarke, and Kalervo Järvelin (Eds.). ACM, 959–962. doi:10.1145/2600428.2609484