VIBE Project Report 2025

Guillaume Witz, Data Science Lab, VIBE Team Lead

The **V**irtual **I**maging at University of **Be**rn (VIBE) project aims to provide a platform for bioimage analysis for the imaging community at the University of Bern. Its goal is to address recurring issues related to software installation and maintenance, access to powerful hardware, and the storage and processing of large datasets.

VIBE received funding in 2024 through an SDIG grant, with a planned start in January 2025. Due to hiring timelines, the project effectively started in March 2025 with the recruitment of Dr. Ruben Lopez as Bioimage Analyst and Jens Müller as Systems Engineer. Over the course of 2025, the team's efforts were distributed across several areas: selecting and designing the platform architecture, assessing user needs through interviews, developing and testing application containers, developing and testing the remote desktop environment, documenting the platform for users, and establishing the processes for user management and billing. Throughout this first year of the project, the ID-Sys team was a key partner and greatly facilitated platform design and deployment.

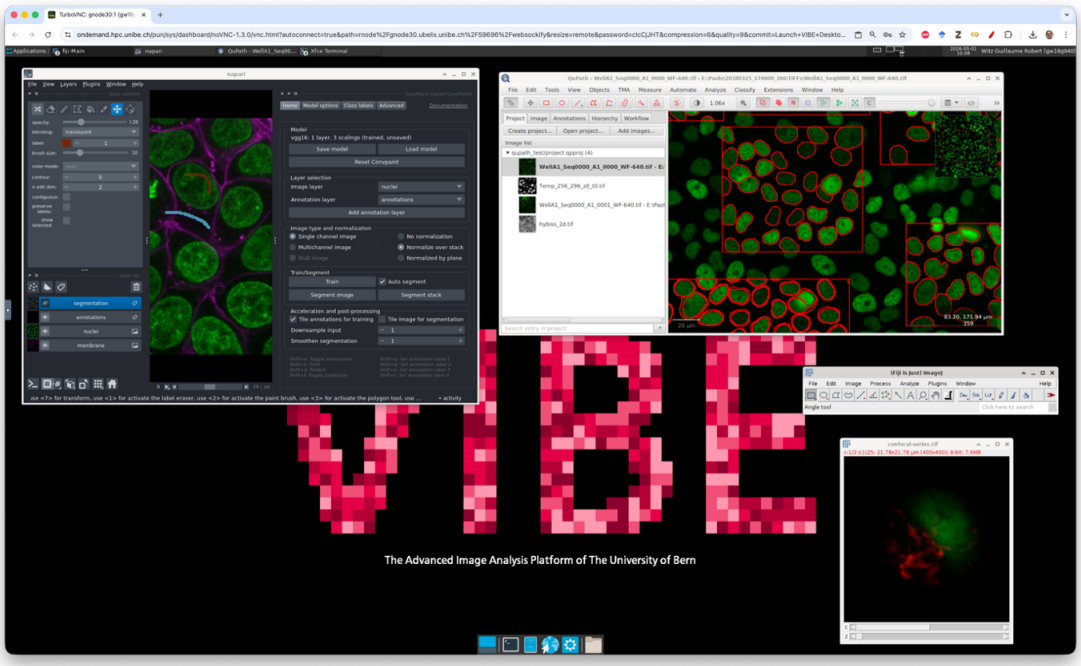

Figure 1: Screenshot of the VIBE Desktop with a few open applications: napari (with ConvPaint plugin, QuPath and Fiji).

Platform Choice

The VIBE team first reviewed existing solutions for providing a bioimage analysis platform. For this work, contacts with other platforms from Switzerland and abroad, facilitated through bioimage analyst networks such as SwissBISA and GloBIAS, were extremely helpful.

The final decision was to build a remote desktop environment running via Open OnDemand on the University cluster UBELIX. Open OnDemand is a popular open-source software developed to provide simplified and interactive access to traditional High-Performance Computing (HPC) resources.

We chose this approach for three main reasons. First, it avoids IT infrastructure fragmentation by extending the existing UniBE cluster architecture instead of, for example, establishing a separate pool of virtual machines. Second, Open OnDemand had just been deployed at UniBE at the outset of the project, making the timing favorable. Third, other institutions, such as VIB in Flanders, had already successfully implemented a similar approach and were able to provide practical guidance.

Architecture

We validated the platform concept during a proof-of-concept phase, in which both the OnDemand Remote Desktop and key applications such as Fiji, napari and QuPath were successfully run on the cluster. Building on this proof of concept, the team then designed a production architecture around several core principles.

Isolation. One key goal is to isolate the VIBE platform as much as possible from UBELIX. The rationale is to avoid making the platform dependent on specific features of the cluster, thereby giving maximal flexibility to the VIBE team for customization. Additionally, isolation ensures that the platform does not create new constraints on the cluster itself, which must serve many other users with diverse use cases.

To achieve this isolation, applications such as Fiji, napari plugins, QuPath, and Huygens are packaged into Apptainer containers, which provide an efficient way to manage complex software installations within the constraints of the cluster environment. The desktop environment itself is also containerized, again allowing VIBE to maximize customization and isolation.

Open-source. All files and scripts required for the platform are shared via GitHub repositories. These typically include container definition files, scripts to build containers, and utilities to create menus, logos and related components. The intention is for the platform to be reproducible on similar infrastructures with only minor modifications.

Automation. To ensure that the platform can be updated easily, significant effort was devoted to automating key deployment steps. Updated and new applications are automatically built on the cluster based on changes identified in the relevant GitHub repositories, such as modifications to the container definition files.

Great care is also taken to avoid disrupting users through failed updates. The platform operates with three stages, development, testing and production, allowing the VIBE team and users to validate applications in development and testing environments before deployment to production. Billing is not in place yet but will also be automated using an approach adapted from that already in place for the regular cluster usage.

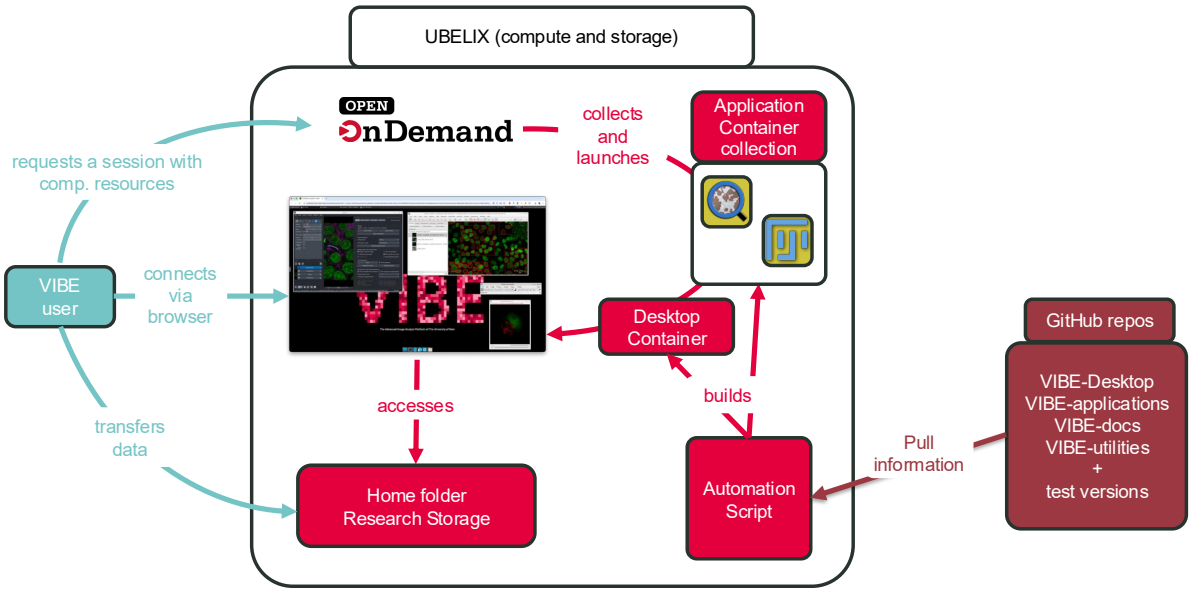

Figure 2: General description of the platform architecture. All information regarding the containerization of the Remote Desktop and the applications offered on the platform is stored in GitHub repositories (right). The information is copied to the cluster where it is used to build the containers automatically. To launch a session, users request resources via OnDemand. Once a session has started, users connect through their browser and gain access to the Desktop and all applications. Users can also transfer data to the cluster, either in their home folder or to a Research Storage share if available. Applications can then access these data directly.

User Needs Assessment

As an important early step in the project, we conducted fifteen interviews with potential platform users to identify software and hardware requirements. These users were selected from the VIBE Working Group, which was a key driving force behind the establishment of the project.

These interviews produced a prioritized list of applications, distinguishing between tools considered essential and those that would be useful but could be addressed at a later stage. Additionally, several interviewees provided examples of complete workflows, including software dependencies and demo datasets, which were used to test the platform.

Container Development and Testing

With user needs established and the general architecture in place, work began building containers for the identified applications. Each application presented its own set of challenges, typically involving dependency conflicts and ensuring correct hardware access, particularly for GPUs. A selection of applications was also thoroughly tested to verify feature completeness and confirm that computational performance met expectations.

Alongside this work, the team developed a standardized approach to container creation in order to ensure consistency in both building and usage. This standardization includes aspects such as container naming, default software (e.g. for 3D rendering), container information (e.g. labels), and software installation (e.g. Python dependencies). To ensure consistency, a container template published on GitHub was designed to guide the creation of future containers.

Desktop and User Experience

Special attention was given to making the desktop environment feel familiar to users accustomed to a standard workstation. Applications are presented in a dedicated menu with categories and icons, and the overall environment is designed to require as little onboarding as possible. Despite the use of containerization, applications retain access to

all user data, for example in the home folder or a Research Storage share, just as they would in a conventional environment.

User Management and Billing

A detailed user management process was established in collaboration with ID-Sys. Research groups will order UBELIX projects and add members through the existing ID-Sys platform (based on Active Directory), after which those groups will be granted access to VIBE. Usage data will then be provided by ID-Sys to the VIBE team for billing, with the billing workflow partially automated by adapting the existing UBELIX process.

Documentation and Outreach

Extensive work was carried out to create user-facing documentation, a key feature of the project to attract users to the platform. The current documentation, still in development, is available at https://dsl-unibe-ch.github.io/vibe-documentation/. The documentation includes a *Getting Started* section, *How-to Guides* (e.g. moving data, estimating hardware needs), *In-depth technical explanations* for advanced users, and practical examples of full workflows with screenshots and videos. This documentation should address the needs of users with varying levels of technical expertise.

To support visibility, a logo was designed to resemble a fluorescence microscopy image with a pixelated rendering of "VIBE" in shades of UniBE red.

Finally, the platform was presented at two events during the year 2025: the MIC Research Day and the MIC Summer School.

Hardware

The VIBE project will use the investment scheme of the UBELIX cluster to provide users with dedicated hardware. This means that while any UBELIX user will technically be able to access it, VIBE users will receive priority and pre-emptible access.

Through the investment scheme, VIBE will not be billed by UBELIX for cluster usage. However, VIBE will bill its own users using the same billing scheme as UBELIX to avoid confusion.

Currently, VIBE plans to invest in two half-nodes, each providing four GPUs of type RTX4090 or RTX6000. The latter type is expected to be operated in virtualized mode, with each GPU split into four virtual instances. Altogether, this infrastructure should support 20 concurrent users at platform launch.

As the UBELIX infrastructure is currently saturated, additional hardware will be added once a new server aisle becomes available. Hardware procurement is also being affected by the current GPU shortage driven by large-scale industry investment.

Collaborations

Throughout the project, exchanges with other institutions proved highly valuable. Within Switzerland, discussions took place through SwissBIAS, including a roundtable on existing platforms at the SwissBIAS Annual Meeting 2025.

At the international level, VIBE participates in the GloBIAS Cloud Computing Working Group, which brings together teams from the Crick Institute, VIB, UniBE, EMBL, University of Queensland, and other institutions. The VIB team in particular, already operating an OnDemand platform, shared concrete insights into container building and infrastructure that directly informed the VIBE setup.

Outlook for 2026

The key objective is to have the platform in operation by Q4 2026. To achieve that goal, the following milestones are planned:

- May 2026: Open the platform to selected alpha-testers, in particular Working Group members who shared workflows.
- June 2026: Finalize the user management system.
- July 2026: Conduct the first platform test during the CEM Summer School.
- July-September 2026: Open the platform for broader testing (free usage). This phase includes testing by all interested users on default UBELIX resources, as well as testing by selected users, including working group members, on VIBE hardware.
- September 2026: Deploy automated billing.
- October 2026: Official platform launch and discontinuation of access through generic UBELIX resources.