The Challenges of Open Source Software Alternatives

Aristofanis Chionis-Koufakos[1], *Maria* Dimou[2], and *Michal* Kolodziejski[2]

[1]National and Kapodistrian University of Athens, School of Science, Athens, Greece
[2]CERN IT Department, Esplanade des Particules 1, 1217 Meyrin, Switzerland

Abstract. Developing an Open Source Software application is a challenge. Mainly because there are commercial alternatives that have an army of expert developers behind them, experienced supporters and well-established business processes in their development and promotion. Nevertheless, web-based applications, that securely handle the users' personal data are an area of freedom and ease of use, features that make such applications very attractive. The "ease-of-use" part is very hard to achieve, for the developers and the end-users. Dependencies change often in OSS packages, so the fear that something breaks is always around the corner.
If the application looks attractive, additional user requirements fall like rain. This poses a problem of continuity, maintenance and operational quality of the packages. In this paper and presentation we shall share our experience in building such a tool, using https://cern.ch/slides, as a showcase and a learning exercise. We shall describe what was available, what was missing, how it was put together, how much effort it took, and what was achieved.

1 Setting the scene

The IT Department at CERN is there to offer high-quality operational services to physicists and engineers working in the laboratory. From this point of view any off-the-shelf professional solution seems to be the natural choice. Nevertheless, some of our services have to be tailored to the special requirements of the accelerator, detectors and data production, storage and management at CERN. This is why a certain degree of in-house software development is necessary. This can hardly apply to “utilitarian” software packages, like email and office tools. On the other hand, such widely used applications, when trusted to commercial companies come with a cost. The level of this cost may not appear immediately and/or may not be only financial. Dependency on a vendor is also costly because one has to adopt all paraphernalia that perform optimally with a given proprietary solution.

In addition, one cannot always say in advance the extent of importance a given in-house open source development may take in the future or its long-term impact in the community. Think of the Web in 1991, and its effect on all aspects of life today, as an example. In this spirit, re-evaluating what we have is always healthy and, sometimes, necessary.

1.1 The MAlt project

The Microsoft Alternatives (MAlt) project [1] started at the end of 2017 seeking open source products with simple exit strategies and low switching costs, in order to minimise CERN's exposure to the risks of unsustainable commercial solutions.

The project aims to deliver services inclusive of all the CERN community. The project's principles of engagement are to deliver the same service to every category of CERN user, to avoid vendor lock-in so as to decrease risk and dependency, to ensure data sovereignty and to serve the common use-cases.

2 The CERN Slides application

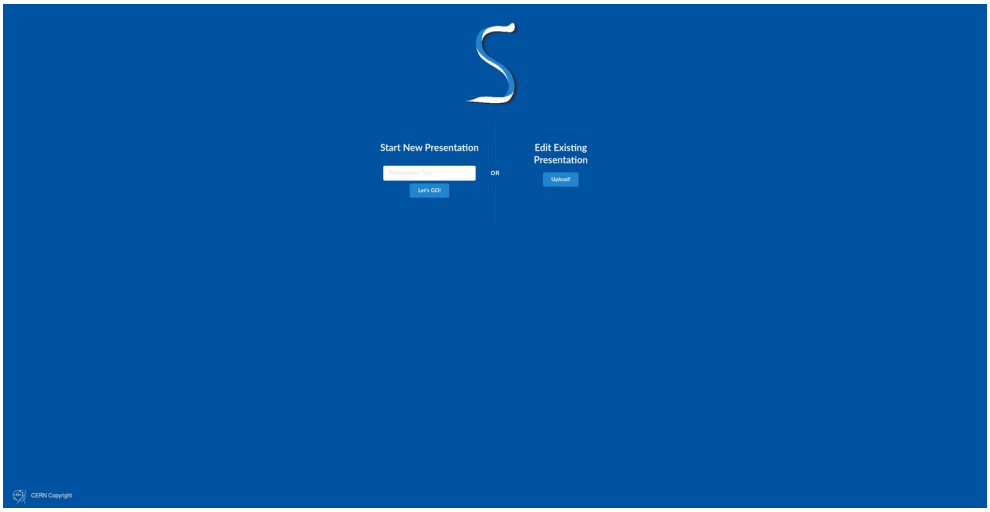

Fig. 1. Entry point to the application – what you see when you open page https://cern.ch/slides

Writing, editing and presenting slides is a common activity for most users. https://cern.ch/slides [2] is an open source web-based software application, developed at CERN in 2020, for users to create presentation slides. This project is built from scratch, using only open source libraries.

The aim of this project was to satisfy all basic requirements and remain open, vendor independent and extensible. Thus the community can propose functionality extensions and contribute code to implement these extensions. Migration of existing Microsoft PowerPoint slides was never part of the project mandate. Covering the needs of users, who find CodiMD [3] slides hard to write, was the main purpose of the project.
The application was developed as a BSc thesis [4] subject in the Microsoft Alternatives (MAlt) spirit. The result is a working prototype that offers the basic functionalities of a slides' maker open source software product. The project would benefit from a few final development touches to be perfected. It also needs some publicity in its favour, in order for it to be used throughout CERN.

2.1 Slides' functionality requirements

Several iterations took place before we concluded on a list of features that were possible to develop in the time available (9 months) and would be necessary and sufficient for the users to try the application. Here is what users can now do, using this slides' maker:

- Add, edit, format and place text anywhere in a slide.
- Add, resize and position images in a slide.
- Select a CERN theme for their presentation, which can be changed at any time.
- Save their presentation as a ".slides" file.
- Upload their existing presentation (".slides" file) to continue editing.
- Present their presentation from the web browser.
- Export a ".slides" presentation, to a PDF file.

2.2 Evaluation of existing products

Given the always limited resources and the wealth of open source components available, re-inventing the wheel would be unacceptable. This is why an inventory and evaluation of existing solutions was made before embarking in development. Their attractive features were noted for our product. When the products were commercial, rejection was unavoidable. Else, the basic requirement to avoid vendor dependency wouldn't be satisfied. This was the case for Slides.com, Google Slides and LibreOffice Impress. Even if free for personal use, at the level of an organisation like CERN, support and license costs are involved. More over, we wanted a web-based slides' maker, with no need for the user to install anything.

2.3 Technologies evaluated

A slides' maker requires an engine running in the background, to translate user's actions via the web browser into code that can be processed by a presentation framework in order to display the presentation. Always with open source technologies in mind, Revealjs was evaluated, which is used in the background of slides.com and CodiMD. Its integration was problematic, so Revealjs was abandoned. The solution adopted was to develop a ReactJS [5] frontend and NodeJS [6] backend and exploit existing functionality of Spectacle [7], a ReactJS library for creating presentations.

2.4 The application architecture

The application design had to handle data format, **storage** and **access** method and rights to a given presentation.

For the presentation **storage** a ".slides" filetype was decided, which is, in fact, a ".zip" file. It contains an "assets" folder for images and animated ".gifs" as well as a "presentation.JSON" file that describes the Redux state that has to be loaded, once the presentation is uploaded to the CERN Slides' application. The part of the Redux state stored in a ".slides" file contains data like the presentation title, the slides' theme, the background colour, the number of slides, the content of each slide and more. To store a ".slides" file the users download it from the browser and save it locally.

For the presentation **access** by the users the https://cern.ch/slides web interface was developed. It requires a CERN login via the new CERN SSO (Single Sign-On). A future integration with CERNBox and/or Indico would be ideal, as other packages are stored and displayed from these popular applications, where the data are kept centrally, safe and secure.

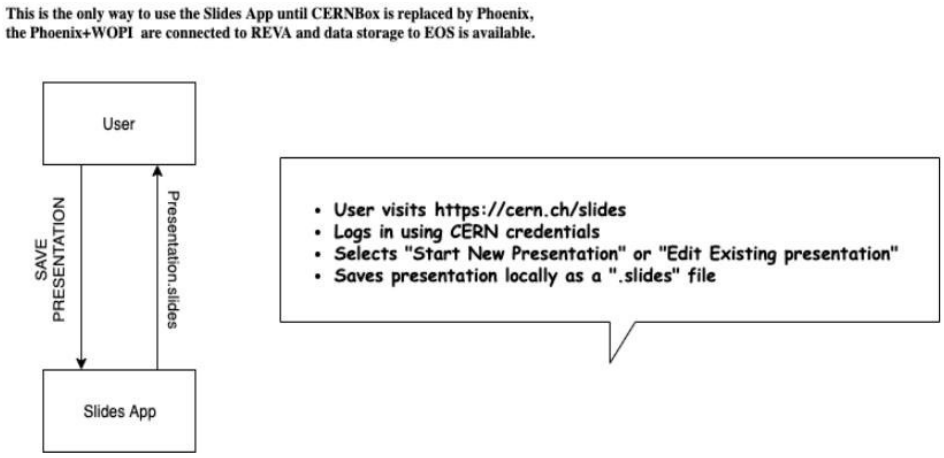

Fig. 2. Current application architecture – The user's interaction with the Slides' app. Storage in local disk.

2.5 Text Editing and Image handling

A text **editor** is obviously mandatory in order to enter content in slides and to give a personal style to one's presentation. The need for this project was to identify a text-editor that would be free and open source, feels modern and customisable, works well with ReactJS and can easily save the text content in the Redux store. After evaluating and rejecting several open source projects like Electron, CKEditor v.4 & 5, Froala and many more (details and product references can be found in [3]), React-draft-wysiwyg [8] was

adopted, an open source project offering ReactJS integration for the draft-js editor and working well with Redux.

The **image** upload, *positioning* on the slide and *resizing* were interesting to program. The option of binary files to store the images (as opposed to base64 string encoding) was adopted for smaller file size and better process performance reasons.

2.6 The CERN Slides' app User Interface

The User Interface (UI) shows, in a user-friendly and intuitive way, the actions possible on the given slide or on the presentation as a whole. This approach follows the principles, look and feel of https://slides.com/. Their UI includes two sidebars on the left and the presentation deck of slides that takes up the rest of the space. The first sidebar on the left is for "presentation-wide" functions (Export, Import, Download, Preview, etc.) and the second one for "slide-wide" functions, adding different types of elements in each slide (Text, Image, Shape, etc.). A user guide is available [9] containing all the usage details.

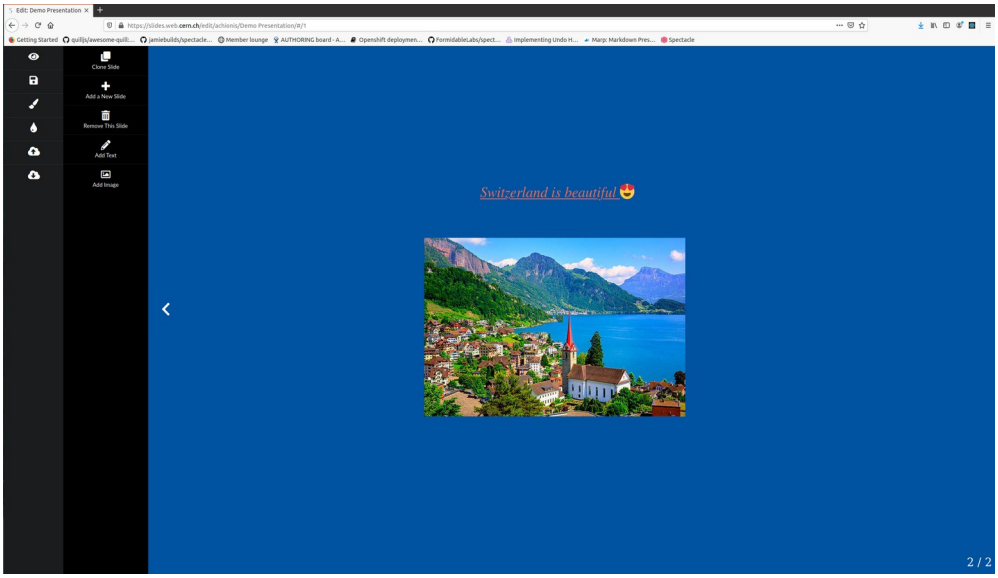

Fig. 3. The CERN Slides application in Edit mode. The menus on the left show how the possible actions that users can perform on the given slide and the presentation as a whole.

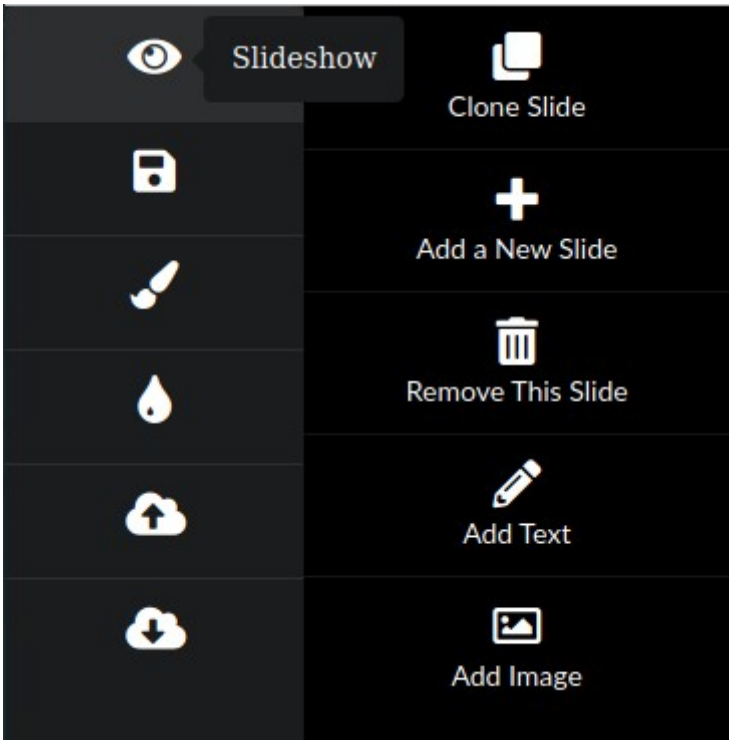

Fig. 4. The left sidebars close-up; CERN Slides' application in Edit mode.

3 The Challenges during this development

The most challenging parts of the application development were the Text Editor, the resizing and moving of the text boxes and images, and the image storage in the production server. The functions were not trivial to implement. There were not enough tutorials to get information from and the time was pressing. We wanted to create a web-based Slides' application from scratch, while satisfying basic user requirements to make the tool attractive to users. These milestones were necessary but difficult to be implemented. The CERN Slides' application, as a full stack web application required work in many areas, from front-end ReactJS code, to CSS, to API design, to proper authentication and securing of the application, to image transfer and storage in the server, to deployment and docker container knowledge. It touched many areas of expertise, which had to be learned. The best coding practices had to be followed in order to assure the longevity of the project.

3.1 Conclusions for the application

Creating a software application from scratch can be tricky. There are a lot of parameters that the developers should take into consideration, different teams that have to cooperate and challenges that need teamwork to be solved. Asking for feedback from the users of the

application is crucial to pave the way of the application development. Also, be adaptable to new user requirements and software limitations is something to always keep in mind. As it always happens with web applications, the user views received sometimes conflicting feedback due to personal taste and habits, so the optimal solution was not always obvious. Another challenge that was faced, was the pessimism of the environment. In the beginning, some colleagues thought the task is gigantic and unachievable considering the time and resources available.

3.2 Good result

The most important thing is that the initial idea/inspiration of the application design turned out to be possible. The use of the Spectacle library is the best solution and the approach that was taken was the best possible. The result is a working prototype of a web application that one can use to create slides, present them using a web browser and export them as PDF files for offline presentations. The initial goal has been achieved and the future seems bright, even if some more improvements and features need yet to come.

4 General and Future Challenges

To some extent, in today's world, developing a software solution in-house for such a utilitarian application is pointless. Many open source applications appear every day partially doing what one needs in order to make slides. Unfortunately, these applications are often abandoned soon after they appear and the relevant software repositories remain exposed to security vulnerabilities. The evaluation and our community requirements show that it is important to identify the right components, assemble all the relevant bits of software and integrate them between themselves and with the CERN ecosystem. This can give an open solution that works. The question now is the application maintenance, its preservation from security vulnerabilities and its evolution to satisfy more user requirements.

Passing from slides written by hand on plastic transparent foils in the 1980ies, several generations of tools for making slides were used by the community. Those of us in favour of open solutions, started by writing slides in HTML [10], and recently used reveal [11] and tried to instruct users [12] to adopt MAlt solutions including CodiMD slides [13], widely used today in CERN IT. Nevertheless, the large majority of users find such tools difficult.

We consider that the CERN Slides' application https://cern.ch/slides is a web-based slides'maker that works for the basic essential functionality, well integrated in our Authentication/Authorisation methods. And it is better than a proprietary solution with many bells and jiggles. Unfortunately, we all have more than enough work already and not enough resources to attribute to such applications' maintenance and promotion.

This is why we appeal to the community to create a forum of code contributions and issues' reporting in order to make this tool and other such tools known and achieve good maintenance and promotion.

Thanks to the IT CDA group management for approving this project and giving functionality advice throughout the development process.

References

[1] https://cern.ch/malt/
[2] https://cern.ch/slides
[3] https://demo.codimd.org
[4] A. Chionis-Koufakos https://cds.cern.ch/record/2724152 (2020)
[5] ReactJS https://reactjs.org
[6] NodeJS https://nodejs.org
[7] Spectacle https://github.com/FormidableLabs/spectacle
[8] React-draft-wysiwyg https://github.com/jpuri/react-draft-wysiwyg
[9] The CERN Slides' application User Guide https://slides.docs.cern.ch (2020)
[10] HTML slides https://cern.ch/dimou/SAslides/ (1999)
[11] Reveal slides https://cern.ch/reveal/maria/atlas-20171211.html (2017)
[12] Non .pptx slides' instructions https://twiki.cern.ch/Edutech/NonPowerPointSlides (2018)
[13] CodiMD slides' example: https://codimd.web.cern.ch/p/SJaroUFSV#/ (2019)