Project Proposal

Tozti — ENS de Lyon

January 15, 2018

Name Tozti

Coordinator Peio Borthelle

Members

- Léonard Assouline
- Peio Borthelle
- Guillaume Cluzel
- Guillaume Duboc
- Julien Ducrest
- Lucas Escot
- Joël Felderhoff
- Felix Klingelhoefer
- Romain Liautaud
- Pierre Meyer
- Alex Noiret
- Pierre Oechsel
- Lucas Perotin
- Vincent Rebiscoul
- Emmanuel Rodrigez
- Daniel Szilagyi
- Lucas Venturini

Abstract The Tozti project aims to simplify the internal organization of student associations by providing them with a suite of open, intuitive and well-integrated tools to solve all their *communication*, *collaboration*, *archiving* and *planning* needs. It will be built by a team of 17 M1 students as part of the "Projet Intégré" course of the ENS de Lyon.

1 An Observation

After more than a year of involvement in the associative life at the ENS de Lyon, we have come to the realization that most student associations are being dragged down by a lack of efficient internal organization. For the most part, the fault lies with the tools they use, which are often unintuitive and don't integrate well with each other. Notably, this leads to a lot of redundant *plumbing* work, and makes it hard for users to locate the information they want, as it is scattered over multiple sources.

2 Our Contribution

To solve those issues, the TOZTI project will **allow each association to build the unified organization suite that best fits its needs** by choosing from a wide range of well-integrated modules. These modules, which we will develop over the course of the project, will each answer a specific organizational need: *communicating* in a hierarchical manner, reaching *consensus*, *planning* tasks and events, *archiving* files securely, *collaborating* on textual documents, or even *automating* recurring tasks.

Ultimately, we want to build a service that student associations can use right away. To this end, we will maintain close communication with the their representatives, to ensure that what we are building suits their needs.

In an effort to be flexible, our software architecture will revolve around small, loosely-coupled but interoperable modules, which we will build using today's web standards and best practices. This gives us several advantages:

- For the end users, the modular nature of our platform will allow easy integration with the competing tools —e.g. mailing lists or Facebook— so as to make the transition process easier. We are convinced that only a platform which does not exclude competitors will be viable, because it will be impossible to hold a monopoly[1].

- For the module developers, using a conventional stack of web technologies, API structures and data-exchange formats such as REST, JWT or iCalendar will enable us to reuse field-tested and trusted libraries for delicate primitives like cryptography, networking or parsing.

- For the maintainers (or anyone who wants to contribute), this conventional stack will also ensure stability and a gentle learning curve for

[1] This is not a novel statement as this path has already been taken by other projects such as Caliopen, Matrix or DuckDuckGo.

understanding and interacting with the internals. This is a requirement to get people to improve or add features to our platform.

3 The Software Specification

We will split the project into several loosely-connected components with thoroughly documented interactions:

Web application providing the unified interface that users will interact with. It will delegate the actual features to the relevant modules.

Storage backend providing privacy-conscious storage of data with fine-grained access-control. We will start from an existing database system (most likely a document-based one) coupled with standard access-control primitives; and will later try to add cryptography to the mix (e.g. to prevent data leaks if the storage server gets compromised).

Essential modules shaped after the specific needs of student associations. Modules marked with **(opt)** will only be implemented if time allows, i.e. if we finish the rest of the project early. Thus we are likely not to implement them at all —or only partly— but as we already have a clear idea of them we wanted to give their specification too.

- **Calendar** with the ability for each user to have his personal calendar and cherry-pick events from associations he is a member of.
- **Discussion board** specifically designed for free-form discussion of associative matters. These discussion threads will be organized into some form of hierarchy. The users will be able to embed or relate to resources provided by other modules (an event, media files, collaborative summaries, etc.) in their messages.
- **Multimedia storage** which stores its data in an structured and easily searchable fashion (by association, by discussion thread, by event, etc.). It will handle versioning of the stored files.
- **(opt) Wiki** as a collaborative textual knowledge base. Access-control will enable writing association staff meeting summaries as well as lengthy event presentations. It will be able to include files from the multimedia module.
- **(opt) Forms** for users to submit structured data. A simple usecase is member registration information. A more specific but widespread one is the ability for the staff to organize tasks divisions during

events with staff members registering for a task and a given time. The result will be stored as tabular data in the multimedia storage module.

(opt) Scenarios enabling an association to automate sequential actions in the other modules to ease and formalize a part of its workflow. The typical usage is the planning of an event which usually includes creating several calendar items, a staff discussion thread, a folder for media files like posters, etc. It will include an interface for designing the macro-like automation as well as an way for firing the script and providing its arguments.

4 Related Work

Taken independently, there are several existing implementations for each module. These will mostly be studied for their interface because as we said in the introduction, one key common problem they have is the lack of unity and selected integration. Amongst others: Discourse, Slack, Trello, Google Drive, Google Docs, Google Forms, HackMD, Etherpad.

For the specific goal of easing the internal organization of a student association, we believe we fill a pretty empty niche. However, a few other projects aim to unify rich communication between people. We will study their UX and their management of the interactions between the multiple components of rich communication:

- Facebook has as goal to ease the communication between people with a common point (an interest group, relatives, friends). Thus it has a few overlapping features but it isn't centered around the structure of a student association. Adeline is similar: it has the notion of an association, but it treats it in somewhat the same way as a group on Facebook. It's prominent feature are forms, but they don't fill the allegedly big usecase of staff task division.

- Github applies these ideas to the field of code development. It's UX is quite similar to what we want to achieve: it stores organized data and one can discuss or collaboratively write about specific parts of this data. These communications are highly organized inside *issues*, *pull requests* or *projects* and are sensibly integrated to the data with the ability to refer to specific lines, commits or branches. Basecamp aims for the same sort of field but maybe has less the will to propose related features at one place.

On a completely different aspect, the vast majority of these services provide no easy way of verifying whether the sensitive data is indeed end-to-end encrypted or can be endangered by compromise of a server. This part will probably be more directed towards research.

5 Workpackages

Tozti is split into several work packages. The following is the list of the work packages, along with their short descriptions and leaders.

COM (Alex): Communication with the associations, and feedback collection.
Members: Alex, Peio, Lucas E., Pierre O.

META (Daniel): Internal organization and communication, code review, testing and dev-ops.
Members: Daniel, Peio, Romain, Lucas E., Pierre O., Guillaume D., Alex

ARCHI (Romain): Planning the internals of our modular system. Choosing the technologies to use. Defining the module interfaces.
Members: Romain, Joël, Léonard, Peio, Daniel, Pierre O.

STORAGE (Peio): Design and implementation of the storage backend. Permissions and encryption.
Members: Julien, Daniel, Peio, Lucas E., Vincent, Joël, Pierre M.

UX/UI (Lucas E.): Mockups of the desired interface and functionality. Front-end web development.
Members: Felix, Romain, Daniel, Lucas P., Alex, Guillaume C., Vincent, Emmanuel, Lucas E.

CALENDAR (Pierre O.): Design and implementation of the calendar module.
Members: Daniel, Lucas E., Guillaume C., Lucas V., Pierre O., Guillaume D.

DISCUSSION (Felix): Design and implementation of the discussion module.
Members: Romain, Emmanuel, Alex, Felix, Lucas P.

MULTIMEDIA (Léonard): Multimedia storage, responsible for storing photos, videos, and other files. Sharing
Members: Peio, Léonard, Pierre O., Emmanuel, Guillaume D.

DEMO (Alex): Preparation of the final demonstration.
Members: Pierre O., Alex

Other, secondary, work packages can be worked on in case we complete all the primary ones before the presentation deadline. Some examples of these packages include

WIKI: A collaborative knowledge base module, for sharing knowledge inside- and among associations.

FORMS: Data collection mechanism, which enables the associations to easily collect member feedback, and store it in a structured fashion, at a common location.

SCENARIOS: A scripting module, which enables the association to perform common and repetitive tasks more quickly. For example, organizing an event may involve: uploading a picture for the event, typing its description, selecting the audience, and sending them an email notification – all these operations would be automatically performed by a script that takes the event's date, photo, and description as its input.

The leaders for these possible work packages are to be chosen later, once all other work packages are deemed to be of a sufficiently high quality.

6 Roadmap

Our schedule is divided in two big parts. During a first time we will strive for a version of a tool having few, but necessary modules showcasing the power it can have. Then, in a second time we will focus on adding more modules to it so that it can really be usable.

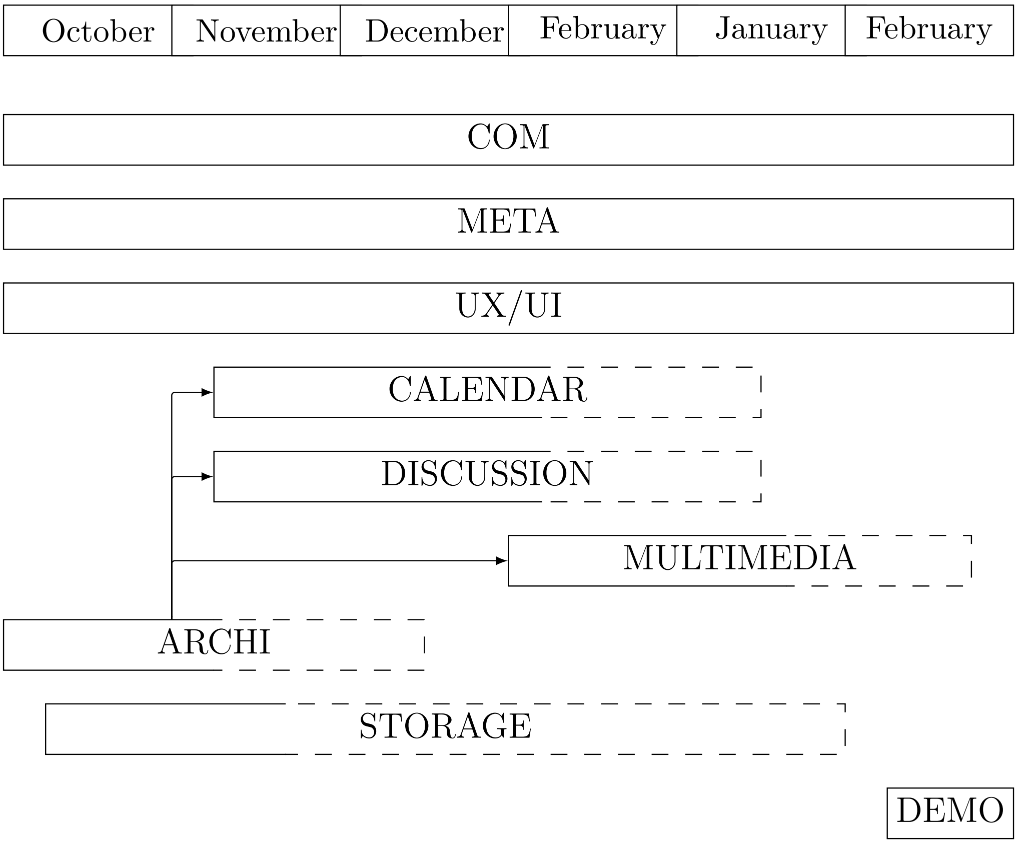

Figure 1: Roadmap of the project

7 Targeted users

We are mainly targeting users involved in the internal organisation of an association. At first, we will focus on the associations of the ENS Lyon. Later on, we would like to extend the project to other schools the have associations.

8 Budget

Eventually, the project will need to be deployed onto a server, for real world testing. Perhaps the DSI could lend us a virtual machine so to approach the conditions of deployment met by associations.