



HAL
open science

MUTE: A Peer-to-Peer Web-based Real-time Collaborative Editor

Matthieu Nicolas, Victorien Elvinger, Gérald Oster, Claudia-Lavinia Ignat,
François Charoy

► **To cite this version:**

Matthieu Nicolas, Victorien Elvinger, Gérald Oster, Claudia-Lavinia Ignat, François Charoy. MUTE: A Peer-to-Peer Web-based Real-time Collaborative Editor. ECSCW 2017 - 15th European Conference on Computer-Supported Cooperative Work, Aug 2017, Sheffield, United Kingdom. pp.1-4, 10.18420/ecscw2017_p5 . hal-01655438

HAL Id: hal-01655438

<https://inria.hal.science/hal-01655438v1>

Submitted on 4 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MUTE: A Peer-to-Peer Web-based Real-time Collaborative Editor

Matthieu Nicolas, Victorien Elvinger, Gérald Oster, Claudia-Lavinia Ignat, and François Charoy

Inria, F-54600, Université de Lorraine, LORIA, F-54506, CNRS, F-54506

Contact Author: claudia.ignat@inria.fr

Abstract. Real-time collaborative editing allows multiple users to edit shared documents at the same time from different places. Existing real-time collaborative editors rely on a central authority that stores user data which is a perceived privacy threat. In this paper, we present Multi-User Text Editor (MUTE), a peer-to-peer web-based real-time collaborative editor without central authority disadvantages. Users share their data with the collaborators they trust without having to store their data on a central place. MUTE features high scalability and supports offline and ad-hoc collaboration.

Introduction

Real-time collaborative editing allows multiple users to edit shared documents at the same time from different places and from different devices. It receives a lot of attention from both industry and academia, and gains in popularity due to the availability of free online services such as Google Docs and Etherpad. Real-time online collaborative editors have multiple benefits. Firstly, they provide a ready-to-use platform for all users to view and modify documents on their web browsers, without installing any software. Secondly, users co-contribute to shared documents in a fast and easy manner as merging of concurrent changes is automatic. Users do not need to manually deal with concurrent revisions and conflict resolution.

These real-time collaboration services rely on a central authority. This places confidential information contained in the shared documents in the hands of a single organisation. Users perceive it as a privacy threat. They have no control on the

usage of their data and may even lose their ownership after sharing them with the authority. These collaboration services generally rely on a centralised architecture that does not scale well in terms of the number of users as shown by Ignat et al. (2015); Dang and Ignat (2016). Moreover, users have to be connected in order to contribute to shared documents. Editors are neither offline capable, nor partition-tolerant: if users lose Internet connection, the changes they performed while they worked on isolation, are lost when they go back online. Furthermore, users cannot collaborate in an ad-hoc manner where they work separately in different subgroups and then synchronise their changes among subgroups.

Peer-to-Peer (P2P) collaboration eliminates disadvantages of systems based on a central authority. Users maintain their data and decide with whom to share it. P2P collaboration reduces the risk for privacy breach as only part of the protected data is exposed at any time. P2P collaboration is highly scalable. It supports a large number of users. It also supports online and offline collaboration as well as ad-hoc mode of collaboration. Users can intentionally split the group into separate subgroups. This enables implementing subgroup-scoped obliviousness such that some exchanged information in the subgroup is not disclosed to the rest of the group. For instance, a subgroup can privately brainstorm on modifications.

In this paper we present Multi-User Text Editor (MUTE), a P2P web-based real-time collaborative editor that supports ad-hoc collaboration. We first describe the different collaboration modes and then the demonstration we propose that highlights subgroup collaborations.

Supported Collaboration Modes

MUTE supports various modes of collaboration. The default one is the online mode. In this mode, all users join the editing session and collaborate on the document at the same time. Firstly, to set up a P2P network between browsers, MUTE relies on the *WebRTC* standard. It allows, using a participant discovery system known as *signalling server*, to connect peers and to broadcast messages. Secondly, to ensure high availability, participants hold a copy of the document. MUTE relies on a Conflict-free Replicated Data type (CRDT) proposed by André et al. (2013) to merge participant contributions. As shown in Shapiro et al. (2011), a CRDT ensures that participants that receive the same set of contributions in an arbitrary order from other users get a convergent view of the document without the need of extra exchanges. To make up for the unreliability of the network and to ensure that users get all contributions eventually, user contributions are exchanged using an anti-entropy mechanism such as presented by Demers et al. (1987).

In offline mode, users can continue working on their copy of the shared document while they have no Internet connection. When a user switches back to the online mode, an anti-entropy mechanism is performed and her local copy of the document is synchronised with the shared document.

To introduce the ad-hoc collaboration mode, we use the following example. Suppose four users Alice, Bob, Dave, and Carol collaborate on a project proposal

as shown in Figure 1a. Suppose that Alice and Bob have to attend a conference and they take a train together. Further suppose that during their travel Alice and Bob have no Internet connection to work online on the proposal. Alice and Bob can set up a Wi-Fi connection between them and collaborate during their travel on a copy of the proposal that will integrate contributions of these two users. In the meantime, Dave and Carol that had network connection contributed to the shared project as shown in Figure 1b. After their travel, when one of the users Alice or Bob has access to an Internet connection they can synchronise their changes with the ones of Dave and Carol as illustrated in Figure 1c. Changes of all users are integrated and they see the same view of the project proposal.

Demonstration

Our demonstration will illustrate this ad-hoc collaboration mode. We first set up a network of two wireless routers and one Raspberry Pi server. This server delivers the web editor code and acts as a signalling server. Alice, Bob, Carol, and Dave share a common document using MUTE. They connect their devices to one of the routers and access the same document URL. They download the web editor shown in Figure 2. Thanks to the discovery service, participants connect to each other and as shown in Figure 1a, they can edit the same document. In order to simulate the subgroup collaboration from Figure 1b, Alice and Bob will connect to another Wi-Fi network provided by our second router, while Carol and Dave will stay connected to the first one. As a result, the group of participants is divided into two subgroups. Participants of each subgroup are no longer connected to the participants of the other subgroup. Their modifications are only shared in their respective subgroup. After each subgroup performs several contributions to their copy, Alice and Bob reconnect to the first Wi-Fi network. Subgroups join into the initial group, as illustrated in Figure 1c. Then, users synchronise to get a convergent document. During the demonstration, we remove the server in order to

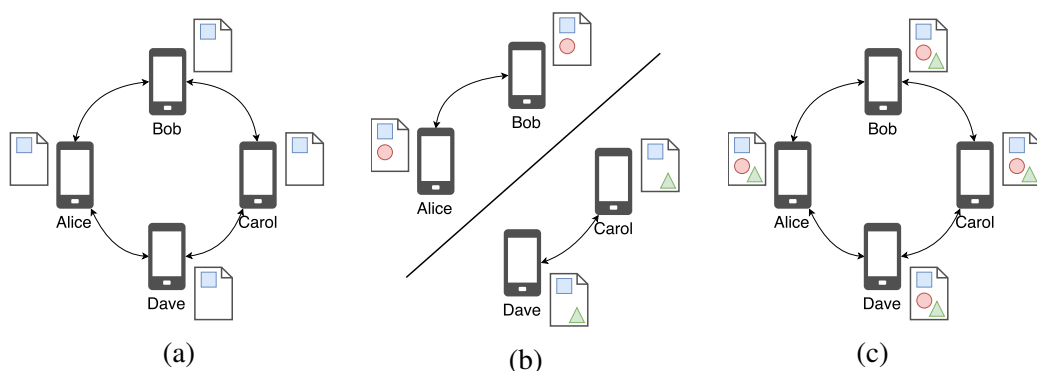


Figure 1: Ad-hoc collaboration: (a) A group of users share a common document. (b) A partition occurred, collaboration continues within the two subgroups. (c) Subgroups join together and then users share their contributions.

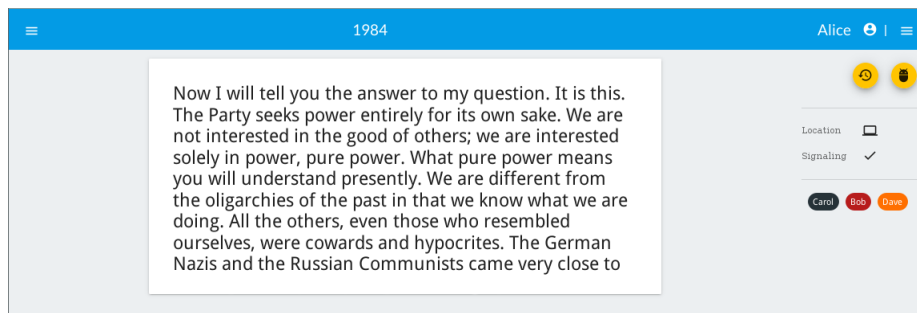


Figure 2: Screenshot of the web editor from the point of view of Alice.

show that it is unnecessary once the editor downloaded and the P2P network established. Conference participants can join the collaboration by using their laptops or smartphones.

Conclusion

We presented MUTE, a P2P web-based real-time collaborative editor and we showed its advantages over existing collaborative editors relying on a central architecture: user control over their data, better scalability and support for online, offline, and ad-hoc collaboration modes.

References

- André, L., S. Martin, G. Oster, and C.-L. Ignat (2013): ‘Supporting Adaptable Granularity of Changes for Massive-Scale Collaborative Editing’. In: *International Conference on Collaborative Computing: Networking, Applications and Worksharing - CollaborateCom 2013*. Austin, TX, USA, pp. 50–59, IEEE Computer Society.
- Dang, Q. V. and C.-L. Ignat (2016): ‘Performance of Real-time Collaborative Editors at Large Scale: User Perspective’. In: *Internet of People Workshop, 2016 IFIP Networking Conference*. Vienna, Austria, pp. 548–553, IEEE Computer Society.
- Demers, A., D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry (1987): ‘Epidemic Algorithms for Replicated Database Maintenance’. In: *ACM Symposium on Principles of Distributed Computing - PODC’87*. Vancouver, British Columbia, Canada, pp. 1–12, ACM Press.
- Ignat, C.-L., G. Oster, O. Fox, V. L. Shalin, and F. Charoy (2015): ‘How Do User Groups Cope with Delay in Real-Time Collaborative Note Taking’. In: *European Conference on Computer Supported Cooperative Work - ECSCW 2015*. Oslo, Norway, pp. 223–242, Springer International Publishing.
- Shapiro, M., N. Preguiça, C. Baquero, and M. Zawirski (2011): ‘Conflict-free Replicated Data Types’. In: *International Symposium on Stabilization, Safety, and Security of Distributed Systems - SSS 2011*. Grenoble, France, pp. 386–400, Springer.