

Automerger: Real-time data sync between edge devices

Martin Kleppmann

Alastair R. Beresford

Department of Computer Science and Technology, University of Cambridge, UK

Abstract

We introduce Automerger, a JavaScript library for data synchronisation between mobile devices such as laptop computers and smartphones. It allows users to read and modify data even while their device is offline, and it automatically merges changes made concurrently on different devices. Unlike most existing data synchronisation systems, Automerger does not require data to be sent via a centralised server, but rather allows local and peer-to-peer networks to be used.

Background

Many mobile apps require data to be synchronised (or *replicated*) between several devices owned by the same user: for example, contacts, calendars, reminders, and passwords. Moreover, there are many collaborative cloud-based apps, such as Google Docs (text documents and spreadsheets), Evernote (notes and ideas), and various business collaboration tools (such as Slack or Trello), which users may access from mobile devices.

Currently, these kinds of application are typically implemented by storing the primary copy of the data on a server. Some applications allow the user to access and modify this data while offline, by storing a local copy of the data on the user's device, while other applications only work while online, because they require synchronous interaction with the server.

Allowing users to read and write a local copy of the data gives rise to performance and availability benefits, since the users do not need to wait for network communication when interacting with the application. However, a challenge with this architecture is that potentially conflicting changes can be made concurrently on different devices that have copies of the data. These changes need to be merged while resolving any conflicts that may have occurred.

Existing collaborative applications such as Google Docs use a technique called *Operational Transformation* to perform this conflict resolution [1, 5]. This technique has the downside that all changes to the data must be sent via a central server, potentially located on another continent, even if the collaborating devices are in the same room. The server must be trusted to correctly process document edits, ruling out the use of end-to-end encryption.

Introducing Automerger

Automerger is a data synchronisation and conflict resolution library that is compatible with any network topology. It allows servers to be used optionally, but it also enables mobile devices to exchange data directly via a local

radio link such as Bluetooth, via a local network, or via peer-to-peer networks. To improve user privacy, it enables the use of end-to-end encryption, which ensures that servers never need to handle unencrypted data, and which cryptographically ensures the integrity of data exchanged between devices [4].

Automerger provides a JSON data model [3] implemented as a *Conflict-free Replicated Data Type* or *CRDT* [6]. This approach allows arbitrary concurrent changes to a shared dataset or document to be merged automatically in a principled manner. We have formally verified that these algorithms provide *strong eventual consistency* [2].

To demonstrate the generality of Automerger, we have implemented several proof-of-concept sample applications: a project management application similar to Trello, a collaborative graphics editor [7], a note-taking tool, and more. Thanks to Automerger, these applications support real-time collaboration: if users are online, edits made by one user are seen by their collaborators in a fraction of a second. On the other hand, these applications are also fully functional while offline, and they synchronise their state with instances of the application on other devices when a network connection is available.

References

- [1] John Day-Richter. What's different about the new Google Docs: Making collaboration fast, September 2010. URL <https://drive.googleblog.com/2010/09/whats-different-about-new-google-docs.html>.
- [2] Victor B F Gomes, Martin Kleppmann, Dominic P Mulligan, and Alastair R Beresford. Verifying strong eventual consistency in distributed systems. *Proceedings of the ACM on Programming Languages (PACMPL)*, 1(OOPSLA), October 2017. doi:10.1145/3133933.
- [3] Martin Kleppmann and Alastair R Beresford. A conflict-free replicated JSON datatype. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 28(10):2733–2746, April 2017. doi:10.1109/TPDS.2017.2697382.
- [4] Martin Kleppmann, Stephan A. Kollmann, Diana A. Vasile, and Alastair R. Beresford. From secure messaging to secure collaboration. In *26th International Workshop on Security Protocols*. 2018.
- [5] David A Nichols, Pavel Curtis, Michael Dixon, and John Lamping. High-latency, low-bandwidth windowing in the Jupiter collaboration system. In *8th Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 111–120, November 1995. doi:10.1145/215585.215706.
- [6] Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. Conflict-free replicated data types. In *13th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, pages 386–400, October 2011. doi:10.1007/978-3-642-24550-3_29.
- [7] Peter van Hardenberg. Pixelpusher: Real-time peer-to-peer collaboration with React, February 2018. URL <https://medium.com/@pvh/pixelpusher-real-time-peer-to-peer-collaboration-with-react-7c7bc8ecbf74>.