Synchronising Medical Data in Lower- and Middle Income Countries

Stefanie Wiegand and Alex Dickinson
Mechanical Engineering Dept.,
Faculty of Engineering & Physical Sciences,
University of Southampton, UK
Email: {s.wiegand},{alex.dickinson}@soton.ac.uk

Gary Wills
School of Electronics and Computer Science,
Faculty of Engineering & Physical Sciences,
University of Southampton, UK
Email: gbw@ecs.soton.ac.uk

***Abstract*—Distributed database research often focuses on use-cases involving big data and frequent updates. We present an approach that allows asynchronous database synchronisation in geographical areas that lack a fully developed digital infrastructure while ensuring data privacy by design and taking patients' consent into consideration when synchronising sensitive personal data.**

I. Introduction

Digital patient management systems in lower- and middle income countries (LMICs) have been used for over a decade [1]. Service providers have long understood many of the advantages these systems have over conventional, paper-based records, such as a reduction in the data duplication overhead, the possibility to easily gather statistics from existing data, a reduction in physical storage space or the possibility of backups. Some restrictions exist in LMICs that prevent clinics from introducing modern systems or upgrading their existing systems, such as high cost, incompatibility with existing databases and lack of know-how to facilitate a smooth transition.

However, even if these restrictions didn't exist, there would still be a difference in the use-cases for employing patient management systems in a production environment:

- Systems may not always be connected to a network [2] (or even the power grid); some nodes may never be connected and rely entirely on "offline synchronisation".
- Dependency on a central server (e.g. "in the cloud") creates a single point of failure.
- Synchronisation may not always be desired: lack of disk space, restricted bandwidth or privacy issues require a more granular synchronisation.
- Encryption becomes even more necessary when information is transferred via other nodes that have no right to access the data.

Our approach is based on a use-case for prosthetics and orthotics (P&O) clinics in Cambodia and aims to address these issues to create a secure, flexible data synchronisation layer.

II. Current System

The current solution is a mixture of paper-based records and a legacy desktop application. The data is captured on paper by prosthetists, physiotherapists and community caseworkers. Subsequently, it is digitised by the receptionist and then undergoes a quality assessment by the system administrators. Once the data is in the database, it can be manually compiled into spreadsheets that need further adjustment before a report can be generated, for example to capture the inventory or create statistics about the number of patients treated. Data is not encrypted nor does the current system generate an auditable access log. Because the system is a Windows desktop application, it can only run on Windows PCs, not mobile devices or other platforms.

Because of these drawbacks, the Royal Government of Cambodia has decided to harmonise the different systems in use in non-government organisations providing P&O services. However, it has not yet been decided which system will be used although it looks likely that it will be an OpenMRS-compatible solution.

OpenMRS is widely used for managing electronic healthcare records in developing countries [3]. While it does provide synchronisation capabilities [4], these are unsuitable for our use-case because:

- The synchronisation is based on a parent/child architecture as opposed to peer-to-peer,
- it requires both nodes to be online at the same time in order to synchronise data, meaning forwarding messages to other nodes is not supported, and
- no encryption is used as messages are always sent directly to their ultimate destination.

III. Proposed System

We captured the requirements of different stakeholders of the system and identified the following:

- The synchronisation should be de-centralised and thus work using a peer-to-peer methodology. This would ensure synchronisation can happen between individual nodes, even when they are not connected to the rest of the network.
- The synchronisation required for our use-case is not based on consistency [5]. As opposed to replication, we adhere to the principle of least privilege [6], synchronising only data, for which consent has been obtained and only to nodes where it is needed.

- To ensure data protection, it is essential that all sensitive data is not only encrypted at rest and in transit, but that encryption is more granular, allowing only designated recipients to decrypt messages.
- The system should be web-based in order to be accessible from a variety of end-user devices without requiring the data to be stored on the device.
- We aim to use the principles for digital design [7], incorporating the users in the design process and making our open architecture available for others to implement compatible solutions for other technologies and use-cases.

We make the following assumptions:

- All nodes know all other nodes in the network and their public keys as they have been set up in advance.
- The synchronisation layer needs to be able to access the database log or the database itself using dedicated credentials.
- The synchronisation has been configured, defining which parts of the database should be included. For granularity on dataset level, the application needs to use a schema that provides unique IDs for datasets independent of the order of insertion.
- The administrator of the database has taken every possible measure to keep the data safe at rest in order to be compliant with data protection laws, such as the General Data Protection Regulation (GDPR), i.e. encryption of the database and/or disk, access control and appropriate logging.

The architecture is simple: each node contains its own database, where patient records are created and managed. The synchronisation layer accesses the database directly, extracting only new or updated data that is to be synchronised. This data is processed, encrypted and passed to the messaging module, which distributes it to other nodes in the group. All messages are encrypted, using keys only the designated recipient(s) can decrypt. Each node keeps a local log of the recipients of each message it receives. Only if a node learns that the message has been received by all designated recipients, it is removed from its message queue no longer propagated.

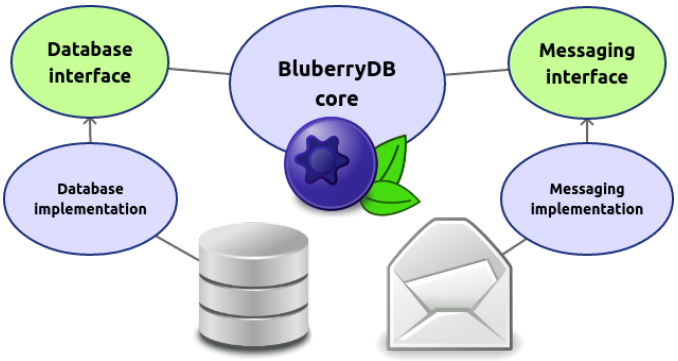

Fig. 1. High-level software architecture.

By separating the implementation from the interfaces and providing an open specification, we ensure that the system is future-proof as new implementations for other technologies can easily be added. While our proof of concept is being written in Python, it can also be implemented in other languages to suit a particular set-up or legacy hardware. Different implementations would maintain compatibility by adhering to the specification.

IV. Discussion

A large body of research focuses on synchronising distributed databases containing a high volume of data, experiencing a high frequency of updates and being highly partitioned, often in a cloud environment. The main focus thus lies in optimising for two of the three guarantees of the CAP theorem (Consistency, Availability and Partition Tolerance). Our use-case is different in that consistency is not only unnecessary but not desired because of consent and potential physical limitations, removing the need to compromise on the other two guarantees. Our priority is not speed, but ensuring data protection, making our approach more pragmatic and user-driven.

V. Future Work

The project runs until the end of January 2021, by which we aim to use the system in Exceed Worldwide's three comprehensive physical rehabilitation centers in Cambodia, helping staff eliminate overhead and free up their time to focus on their patients. However, we are also working with the National Institute of Social Affairs in Cambodia, who are working on harmonising patient data across non-government organisations. By keeping the open architecture flexible and allowing different technologies to be integrated, we hope to eventually achieve a bigger impact and expand our synchronisation layer beyond the scope of the current project.

Acknowledgment

Advisors: Exceed Worldwide, BluPoint Ltd.
Funders: EPSRC/NIHR (EP/R014213/1) and RAEng (RF/130). ERGO/FPSE/46271.

References

[1] B. Wolfe, B. Mamlin, P. Biondich, et al., *The OpenMRS system: collaborating toward an open source EMR for developing countries*, AMIA Annu Symp Proc. 2006;2006:1146.
[2] S. Wiegand et al., *Designing a Distributed Prosthetics Database for Use in Lower- and Middle Income Countries*, Asian Prosthetic and Orthotic Scientific Meeting (APOSM), 2018.
[3] C. Paton and M. Malik, Open Source and Free, Web-based Medical Software, 2008.
[4] Digital Square and Soldevelo, *Sync 2.0 OpenMRS module, https://github.com/openmrs/openmrs-module-sync2 (last accessed 11/04/2019)*, 2017.
[5] E. Brewer, *Towards robust distributed systems.* (Invited Talk), Principles of Distributed Computing, Portland, Oregon, July 2000.
[6] J. Saltzer and M. Schroeder, *The Protection of Information in Computer Systems*, Proceedings of the IEEE, vol. 63, no. 9 (Sept 1975), pp. 1278-1308.
[7] Principles for DigitalDevelopment Forum, *https://digitalprinciples.org/ (last accessed 11/04/2019)*, 2017.