The Development of Peer-to-Peer Matrix

Hilmar Gústafsson `<hgusta16@student.aau.dk>`

March 10, 2020

Abstract

...

1 Introduction

In 1971, the first email was sent [9]. Email is currently implemented using decentralized protocols for sending simple messages between recipients, specified for example in the SMTP RFC [8]. One may register with an email provider, and write messages to any other provider implementing the same protocols, as long as they have the recipient's address, known as an email address. Mobile providers similarly allow people to contact others outside of their network. However, there's been a growing trend of *communication silos*, where users may only contact other users of the same service. Examples of such services are WhatsApp, which has announced that its userbase has now exceeded two billion users [7], and Facebook Messenger, which similarly reported having 1.3 billion users [4].

Communication Silos Although these services run over the internet, they are designed in isolation, and do not adhere to an external standard. In order to use such a service, a user must acquire a client designed for this private procotol. This approach does not scale well in terms of battery or network traffic, as each application separately performs similar routine actions, such as checking for and sending messages. We refer to the challenges of communication silos as *the silo problem*.

Security In the typical client-server architecture, the client is forced to operate with some level of trust. In the case of email, the original architecture required that both participants be online, but has now adapted such that some server just needs to receive it and be able to pass it on to the recipient when possible [9]. Platforms which follow this architecture for messaging therefore store the messages of all conversation participants. The storage format and security depends on the software run on the server, which receives, stores and passes the messages on. For closed-source vendors, it is intractable to know whether the messages passed are done so securely, or whether third parties can access the data. Users are therefore forced to either rely on the word of the service providers, or use an alternative. In the cases of WhatsApp and Facebook Messenger, several such features are boasted, but regardless unverifiable, as per [10].

Matrix The Matrix open standard [3] is a specification for a network similar in structure to email. It supports more recent features, such as live messaging, calls, and video conferencing. This standard is a potential solution to *the silo problem*, and has several strong points. It is designed for interoperability. It has multiple clients [2] and SDKs [5]. Through bridges, Matrix allows communicating with different protocols [1]. The reference implementation of the Matrix open standard is Synapse [6], which is writ-

ten mostly in Python.

1.1   Initial Problem Statement

We will explore the current use of the Matrix open standard and its current limitations. We might consider alternatives to current solutions, as well as novel solutions to open problems.

We propose the following initial problem statement:

- How is the Matrix open standard currently being used?

- What are the limitations of the standard?

- What are the limitations of the reference implementation?

- What are the necessary properties of solutions to these limitations?

2   Analysis

We explore the different questions posed in the initial problem statement in section 1.1

2.1   Use of the Matrix Standard

2.2   Matrix Specification Limitations

2.3   Matrix Reference Implementation Limitations

2.4   Properties of Solutions

2.5   Problem Statement

- What are the architectural consequences and trade-offs in peer-to-peer communication?

- What are the properties of a good solution?

- How could one implement such a solution?

References

[1] Bridges | matrix.org. `https://matrix.org/bridges/`. Last accessed: March 10th, 2020.

[2] Clients | matrix.org. `https://matrix.org/clients/`. Last accessed: March 7th, 2020.

[3] Matrix specification. `https://matrix.org/docs/spec/`. Last accessed: March 10th, 2020.

[4] Messenger - more than 1.3 billion people. `https://www.facebook.com/messenger/posts/1530169047102770`. Last accessed: March 6th, 2020.

[5] Sdks | matrix.org. `https://matrix.org/sdks/`. Last accessed: March 7th, 2020.

[6] Synapse: Matrix reference homeserver. `https://github.com/matrix-org/synapse`. Last accessed: March 10th, 2020.

[7] Whatsapp - two billion users - connecting the world privately. `https://web.archive.org/web/20200215075032/https://blog.whatsapp.com/10000666/Two-Billion-Users--Connecting-the-World-Privately`. Last accessed: March 6th, 2020.

[8] J. Klensin. Simple mail transfer protocol. RFC 5321, RFC Editor, October 2008. `http://www.rfc-editor.org/rfc/rfc5321.txt`.

[9] Ray Tomlinson. The first network email. *Site de Ray Tomlinson*, 2009.

[10] Lance R. Vick. Lance r. vick on twitter: "i finally started a spreadsheet to compare relative security, privacy, compatibility, and features of various messenger systems. tl;dr @riotchat / @matrixdotorg is winning on all fronts. https://t.co/7zxczdjwwj" / twitter. `https://twitter.com/lrvick/status/1051260991479013376`, 2018. Last accessed: March 7th, 2020.