Analysis of an Internet Service or Product

Título: Analysis of an Internet Service or Product

Author: Bernardo Rodrigues – nmec 88835
João Marques – nmec 89234
João Silva – nmec 88813
Tomás Costa – nmec 89016

Date: **27/03/2020**

Index

1. Summary / Abstract

Matrix is an open network for secure, decentralized communication. It is a **decentralized conversation store** rather than a messaging protocol.

Matrix gives you simple HTTP APIs and SDKs (iOS, Android, Web) to create chat rooms, direct chats and chat bots, complete with end-to-end encryption, file transfer, synchronized conversation history, formatted messages, read receipts and more.

2. Framework

Information and communication technologies have undergone considerable changes during the last decades and have acquired major importance in present day forms of life, leaving us with no choice but to embrace this inevitability.

In previous times there was a clear line between the “public” and “private” spheres of daily life. Because communication technology could only be transmitted in very specific forms like a television set or radio signal.

It is a given fact that these technologies grant us possibilities that would be impossible otherwise, one simple example is what is going on in these last few months. The corona virus hit us hard, self-quarantine is a must right now, but for those who can, and thanks to the growing variety of internet products, this doesn't mean work, communication and interaction need to stop as well.

But as these services evolve and our dependency to them grows with them, it is important to understand a bit more about this subject and specific products in order to comprehend what is better suited for our likes and needs, what is safe and what is not, what is private and what isn´t, and for those, who care, like us, how they actually work.

3. Architecture

Matrix tries to solve common problems in communication platforms (regarding security, privacy and reliability) with it's well-designed architecture.

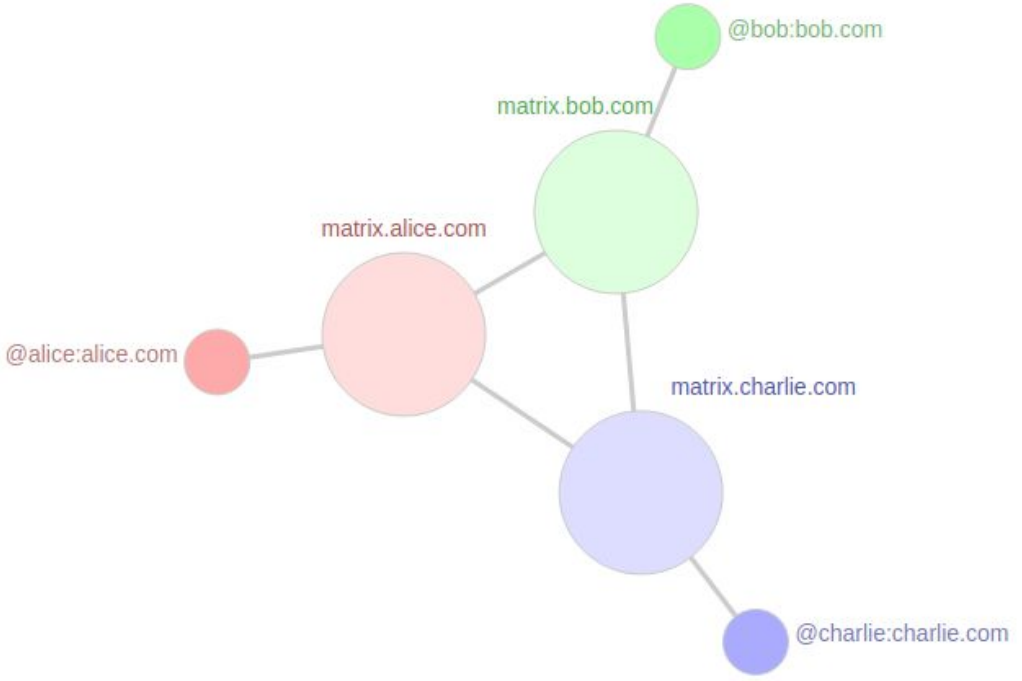
As explained before, Matrix follows a decentralized communication approach. Every user has a **homeserver**, where the client connects to, which stores the communication history and account information for that user.
The homeserver shares data with the wider Matrix ecosystem by synchronizing communication history with other homeservers. This way, communication is always made between 2 or more homeservers, and is sent across them in a way no central entity must process any information.
In the case of a group chat, each message is replicated in every homeserver present in the conversation.

On the other hand, the fact that the communication is decentralized in homeservers, and not in clients, allows that each client can send and receive information using several devices, and they all synchronize and share both messages and encryption data.

Besides, the fact that no central entity is responsible for processing the information, means that the user's homeserver can be anywhere in the world (in a house, in a company's server room, or in a datacenter), which poses significant advantages in terms of **privacy and security**, as will be discussed later.

3.1 Distribution

The Matrix protocol and network can be visualized next:



[1] https://matrix.org

In this diagram, we can see 3 homeservers represented, for 3 users, respectively (matrix.alice.com, matrix.bob.com, matrix.charlie.com). As explained before, each of these can be placed on any part of the world, or in the same physical server. For the Matrix network, it represents no difference.

In this case, if all 3 users are communicating in a group chat, and user Alice wants to send a message, the same is sent from the client (@alice:alice.com) to the homeserver (matrix.alice.com), where the message is kept as registry, and then replicated to all 3 homeservers of the other members of the room (homeservers matrix.bob.com and matrix.charlie.com), and consequently propagated to each client (@bob:bob.com and @charlie:charlie.com).

This way, this protocol assures all users can connect between themselves (be it on a 1:1 chat room, or a larger group room) without a central party.

3.2 Encryption

On the other hand, the Matrix protocol also provides a very interesting and practical way of handling encrypted chat rooms.

End-to-end encryption (E2EE) consists on a message being encrypted on the sender's client application (running on a smartphone, tablet or computer) and only being decrypted on the receiver's. As oppose to client-server encryption, this means that the server handling the message distribution (in this case, the homeserver), will not be able to access the message's content.

For this to work, an encryption key must be generated by one of the parties involved in the messaging, and then shared with the other member of the conversation. Many chat systems (like WhatsApp) implement this feature in 1 to 1 messaging. However, when the chat room contains more than 2 people, the number of key exchanges that would have to take place between all the users rapidly becomes hard to manage, so many of these chat services opt not to implement this feature on group chats.

Besides, another problem E2EE raises is presence of the encryption key in all the user's devices. If using this feature, the encryption key would have, not only to be shared between all the users, but also between all their devices.

Matrix solves this issue by implementing a simple peer-to-peer key-sharing algorithm. For a conversation to start, each user shares it's encryption key with all the others. If at some point a key is missing, the user can re-request it from the sender, who should accept the request, or not, if the behavior seems suspicious.

If the user is using more than one device, he can choose to share the encryption keys every time between his devices (by accepting a simple pop-up message), or by storing the encryption keys in the homeserver. This would be, of course, encrypted themselves with another key created by the user.

This way, secure and encrypted communication is easily achieved.

Matthew Hodgson, project lead for Matrix, clarified that they "use AES-256 in CBC mode with PKCS#7 padding for encryption and HMAC-SHA-256 (truncated to 64 bits) for authentication. The 256 bit AES key, 256 bit HMAC key, and 128 bit AES IV are derived from the message key using HKDF-SHA-256." [2]

3.3 Performance

This algorithm was built under the idea of, not only permitting a good way of achieving the goals mentioned earlier, but also of implementing a fast and efficient protocol.

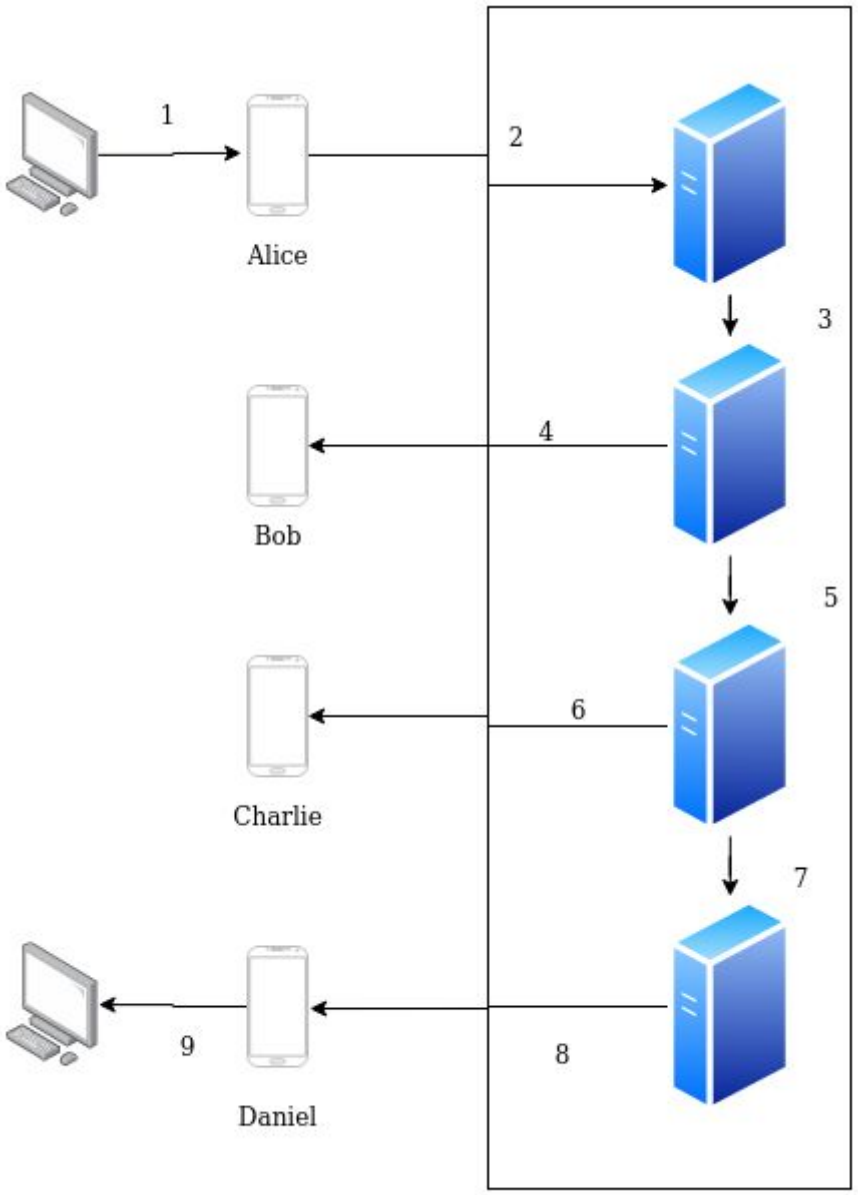
In terms of bandwidth usage, the Matrix protocol is similar to other commonly used ones, using about 1500 bytes for each message in the network.

However, as this is an open protocol, and allows for customization and change, several efforts have been made in order to achieve the lowest bandwidth possible with one message. In that sense, it has already been possible to achieve a message exchange that only demands 100 bits-per-second of bandwidth in order to make real time communication.[3]

This show the amount of customization that can be made in order to achieve one's goals on a specific use case for the platform.

On the other hand, we can compare the number of packets that would have to be exchanged on a single communication with other commonly used protocols.

For this example, let's take an example of a group chat with 4 users. If we consider a common client-server-client centralized protocol, like WhatsApp, and assume user 1 (Alice) is using her computer to chat, and user 4 (Daniel) is using it too, we will get a packet exchange diagram like the following (simplified):



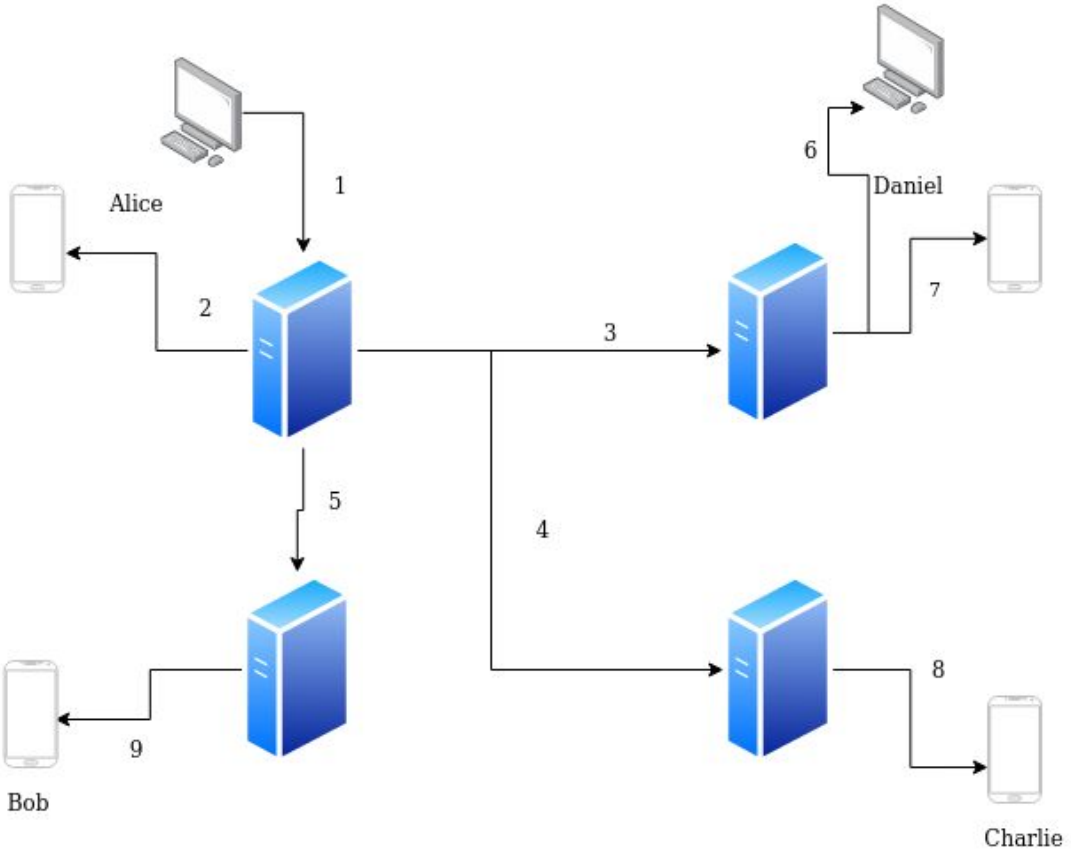
We can see that 9 packets are transmitted. Packets 3, 5 and 7 would be for replication between WhatsApp's servers, and can be less, or more than these. Packets 1 and 9 would be a "forward" between the user's phone and web client, and packets 2, 4, 6 and 8 would represent the communications between the users and WhatsApp's servers.

So, we would be able to reach the conclusion that this communication implies:

$$N = (NumServers - 1) + NumComputers + NumUsers$$

where **N** represents the number of packets, **NumServers** the number of WhatsApp servers involved in the communication, **NumComputers** the number of computers being used by the users and **NumUsers** the number of users in the room.

On the other hand, we can translate this situation to the Matrix protocol:



We can see that, again, 9 packets are transmitted. Packets 3, 4 and 5 would be for replication between each homeserver (taking in consideration that we can have several users on the same homeserver, and the packet would never have to leave it). The remaining would represent the communications between the clients and the respective homeservers.

In this case, the communication implies:

$$N = (NumHomeServers - 1) + NumDevices$$

, where **N** represents the number of packets, **NumHomeServers** represents the number of homeservers and **NumDevices** the number of devices presents.

As we can see, although the number of packets is the same in this case, in the case of Matrix, it doesn't depend on the number of users in the chat room, nor the number of computers.

So, we easily realize that it, not only isn't worse than WhatsApp in terms of performance but can also be better in many situations.

4. Functionalities

Besides including the basic features of a messaging protocol, such as messaging, Matrix also has some features that other protocols have started to include, such as:

- Video and Voice conferencing
- File sharing
- Eventually-consistent cryptographically secure synchronization of room state across a global open network.
- Creation and management of fully distributed chat rooms with no single points of control or failure

Matrix excels when it comes to external integration, it can write to channels outside of Matrix (Slack, Gitter, and IRC), it can host bots to stay in channels and listen and respond to commands (GitHub, Giphy, and RSS Bot) and it also features widgets for a full-blown interactive matrix room (Jitsi, Grafana and Etherpad).

The Jitsi-Matrix integration [4] allows you to self-host your video conferencing room.

Besides the ones listed above, Matrix offers an API to extend its Client-Server functionalities [5].

5. Business Model

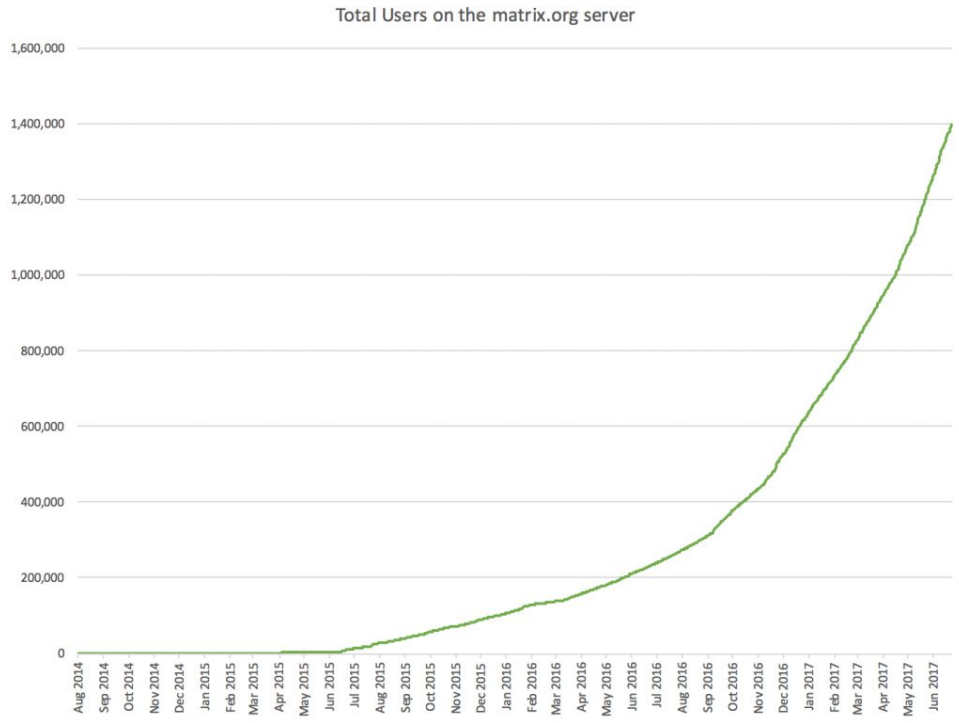
Along the development progress, the original development team split into 2, one to work on the **matrix protocol** and the other to work on **riot.im**, which is a client that uses matrix in a messaging app, with both free and paid options for its customers. In order to keep matrix free and independent, the matrix protocol is currently owned by **The Matrix.org Foundation**, whilst the riot.im is owned by the original development team **New Vector Foundation**, that both back up these projects.

The goal of the matrix protocol is to create an open-source communication protocol. As such, their license falls under the Apache Licence v2 [6], with states that any company is free to use its protocol for both free and commercial use.

The company responsible for the development of the protocol was initially funded by large multinational telecommunications infrastructure company (Amdocs).

But, after funding was cut in August of 2017, the company has decided to accept donations from private entities via Patreon, Bitcoin and Ethereum [18]. However, most of the funding comes from companies interested in the protocol, like **Status.im,** with a 5 Million partnerships in 2018. As of today, the developers of matrix also offer a paid prioritized or custom development for corporations.

Although the goal of matrix is to create a protocol where every user can communicate regardless of its client application, it inevitably has to compete with other protocols used by the major communication platforms, like the “Extensible Messaging and Presence Protocol” (XMPP) - used by WhatsApp - or the custom made protocol by Slack. Even with the competition, the number of users for matrix has being increasing.



[18] matrix.org

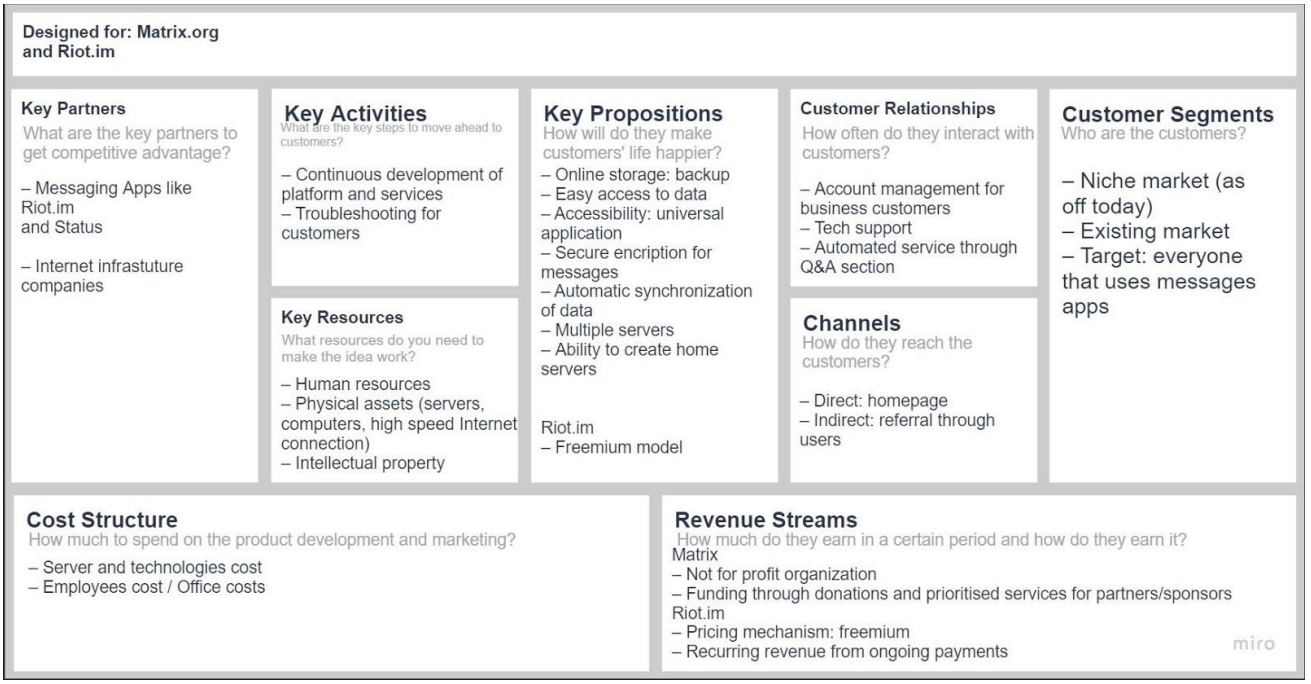
We believe that this is because of matrix’s (and, by extent, matrix clients’) distinct competitive advantage in which it allows its users to communicate with different apps. For example, a person with a riot.im account can link it with its WhatsApp account (through matrix’s bridges) and receive the messages from both riot.im and WhatsApp on a single app.

With this in mind it becomes clear that the success of this messaging protocol doesn’t depend on the rise or fall of messaging apps, since its goal is to allow users to communicate with everyone regardless of protocol.

So, in a way it belongs in a different category, where it can function as the sole protocol used by all messaging apps, or in its own niche as a way to bridge different apps together (or somewhere in between).

To better visualize the business model of matrix, a business model canvas has been designed. [7]

Since both the matrix and the Riot.im teams are closely related and follow similar goals (in fact, they were once part of the same team), we decided to join the two businesses models of matrix and its principal client/business partner Riot.im.



As for customer segments, and as already mentioned, the protocol is free for everyone to use, so any future messaging app can create a client to matrix, like Riot did.

Riot tries to enter an already existing market with well-established participants (WhatsApp, Facebook Messenger, Slack). Two off the main selling features of Riot/Matrix are the security it claims to be able to provide to its users and the amount of control it gives to its user by allowing the use of homebrew servers. This has resulted in a user base composed of more experienced users that wish for more control over their messages.

On the other hand, by not being as easy to use as the more prominent apps, it has created a large barrier of entry for newcomers, which in turn results in less users.

Therefore, as of today they fill a role in a more niche market. It's worth to keep in mind that riot's client is relatively new, as such their team is still working on improving accessibility hoping to reach a broader market.

Their main channels (the way they get new clients) is mostly through their webpage and referral from users, not different from how the actual major messaging applications started.

Their main offer (Key Propositions) is the existence of a secure universal app for users to communicate between multiple apps, along with a free protocol that allows for communication between different clients.

Their key activities consist of the continuous development of the matrix and riot.im apps. Additionally, matrix also offers help for other messaging applications that choose to use their protocol.

The key resources consist on the code created by the respective teams, along with said teams. Also, the infrastructure necessary to keep the app running like the servers and internet connections

As for key partners, matrix chooses to partner with apps that use their protocol like **Riot.im** and **Status.im**. Additionally, since their protocol is much more lightweight compared to others, they also have favorable relationships with internet infrastructure companies (it's worth remembering that they were founded by one).

For customer relationships, both offer Q&A for anyone interested, and tech support when needed.

Finally, the cost structure is mostly the costs associated with managing an office, the employee's salary, the costs the servers (maintenance) and the technologies needed to make the app/protocol.

These costs are offset by their revenue Streams. Whereas matrix is a not for profit organization, they get their money mainly through donations and sponsors, to which they offer additional help and personalized development. Riot.im relies on a "freemium" model where they offer a free plan to their clients, along with a paid option consisting of recurring payments, with better servers and more customization options.

6. Social and Economic impacts

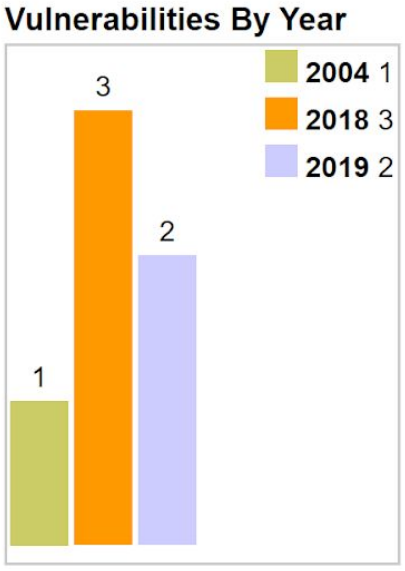
One of the main features is the amount of security and privacy it allows to its users. As such, many users that wish to keep their data hidden may use this protocol/client to exchange messages between them.

One such case of a big entity using the matrix protocol is the French government [17], that forked a project from Riot.im, and made its own custom implementation of a messaging app called Tchap. Even though the app was meant for government officers requiring a French issued government email (@elysee.fr), soon before it was released a hacker was able to fake its email by adding the extension to its regular email, gaining access to public channels. The hacker then reported the security

flaw to the matrix team and they deployed a fix. The bug was due to a bug in a python module that hasn't been fixed since July of 2018. [17]

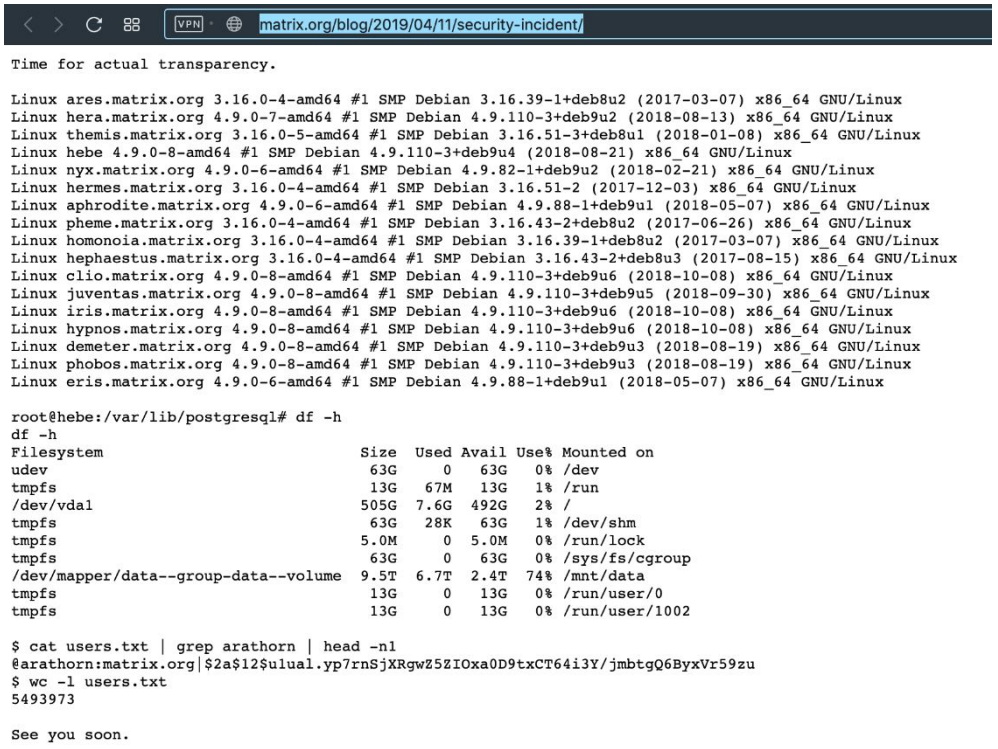
Such major flaw could have caused a massive breach in the French internal affairs. In case it would have happened, it's unclear whether the matrix developers would be responsible or the French government.

Even through matrix claims to be safe, as off today it has 6 identified security vulnerabilities listed on the CVE database. [8]



For comparison WhatsApp has 8 [9]. Facebook messenger has only 1 [10].

Another incident was the hacking of one off the matrix servers in April of 2019 [16], where a hacker got access to one of the production databases, gaining access to unencrypted message data, private messages, password hashes and access tokens. Furthermore, the hacker also published the following image in the matrix blog:



[16] matrix.org

Matrix identified the security as a vulnerability in their testing software: “The version of Jenkins we were using had a vulnerability (CVE-2019-1003000, CVE-2019-1003001, CVE-2019-1003002) which allowed an attacker to hijack credentials” [16].

The security breach affected mostly the matrix productions servers, where their user information is stored making user’s credentials possibly compromised, it's worth pointing that their paid alternative servers where not affected (modular.im). Although riot and matrix both claims to be safe for everyone, in this specific situation the paying customers where safer than the latter.

Although it affected matrix users directly, it was not an issue involving the matrix protocol’s code.

However, it is important to mention that one of the advantages of having a seamless end-to-end-encrypted system is that, in case of an attack to the servers storing the messages, these wouldn’t be able to be decrypted by any hacker.

Lastly, we need to mention that this exact feature, end-to-end-encryption, also raises a social and economical question. The fact that no central entity is responsible for processing the information gives users a great deal of privacy and security for their communications. However, this in turn allows users to use the platform for more nefarious uses like sharing copyrighted files or terrorism plots.

There has been various instances where criminals hide their information behind encrypted data. For example, the dispute between Apple and the FBI [11], where the FBI wasted time and money to access the data. In this situation, Apple could have accessed their devices and deliver the data to the FBI.

As of now, matrix.org and riot.im both fall under the UK government laws, but should a similar situation happen due to the privacy measures that matrix chooses to implement, the amount of assistance Matrix would be able to provide would most likely be very limited.

7. Other solutions

It is time to take a look at how matrix and riot (matrix client) place in relation to other similar solutions [12].
The closest most known services that aim for the same objectives as matrix are probably Slack and Mattermost, since they're the biggest virtual workspaces out there. Even so, in some cases it is worth to make a comparison with other messaging services like WhatsApp.

Times have changed, old services need to evolve or watch new solutions get ahead. Matrix is one of the new options we now have.

Where and who can access the data? This is the question of the moment, security and privacy have become mandatory requirements.

Riot is definitely a model to look at in terms of security, with end-to-end security in private and group conversations as decentralized communication requires encryption.
Slack, just as Mattermost supports two factor authentication and secure client-server communication, however, neither support end-to-end encryption. Anyone that gains access to a Slack server can read everything you have written and exchanged.

It Is because of this rising concern that WhatsApp (one of the biggest chat apps out there) adopted the encryption developed by yet another chatting Service – Signal. Signal is popular for secure chats and secure calling, and, if they are 100% honest about how they handle information, does not provide any information to anyone.
Despite this, because you don't need to give your telephone number when using Riot, your account won't be linked with your identity. Another aspect, Signal is centralized. This means data is sitting in one place and although we are sure it is well protected, it is a prime and clear target to be attacked or regulated. Also, Matrix allows for encrypted chat rooms with more than 2 people, as oppose to WhatsApp and Signal.

So, if you use Matrix servers with no encryption, signal will be the winner here. But if you use your own homeserver, or use a homeserver you trust or encrypt your rooms, Matrix looks a lot better.

Matrix is Open source and, obviously, there are plenty advantages in choosing an open source solution for overcoming an issue in a project, in this case, communication. Open source is on average more secure and more agile in reaction and discovery of bugs. Open source is transparent. Open source removes the dependency on a commercial organization and the threats of their future decisions, not to mention the freedom you gain with lesser costs.
Although Mattermost is Open source, you need to perform a monthly payment to unlock its full potential. Slack is not Open Source and has higher costs than Mattermost.

We cannot forget one of the prime features of Matrix, which are the **bridges**. Using Matrix, you can bridge chatting rooms to Whatsapp, Slack, etc. None of the competitors can provide a similar service.
Another advantage is that you don't have to use a specific client as there is a big variety you can choose from as for example Riot.

Comparing Riot to other services [15]:

Client	Discord	Mattermost	**Riot**	Signal	Slack	Whatsapp
Windows	YES	YES	YES	YES	YES	YES
macOs	YES	YES	YES	YES	YES	YES
Linux	YES	YES	YES	YES	YES	NO
Web	YES	YES	YES	NO	YES	YES
Android	YES	YES	YES	YES	YES	YES
iOS	YES	YES	YES	YES	YES	YES
E2EE User chat	NO	NO	OPTIONAL	YES	NO	YES
E2EE Group chat	NO	NO	OPTIONAL	YES	NO	YES
File transfer	YES	YES	YES	YES	YES	YES
Voice recording	NO	PLUG IN	NO	YES		YES
Voice chat	YES	PLUG IN	YES	YES		YES
Video chat	YES	PLUG IN	YES	YES		YES
Read receipts	NO	NO	YES	YES		YES
Editing sent messages	YES	YES	PARTIAL	NO		NO

Deleting sent messages	YES	YES	YES	PARTIAL		PARTIAL
Custom emoji	YES	YES	YES	NO	YES	NO
Stickers	NO	NO	YES	YES		YES
Self-destruction messages	NO	NO	NO	YES		NO
Lock screen	NO	NO	NO	YES		NO
Themes	YES	YES	YES	YES	YES	NO
Plugins	YES	YES	YES	NO	YES	NO
P2P	NO	NO	YES	NO	NO	NO
Open Source	NO	YES	YES	YES	NO	NO
Federated Servers	NO	NO	YES	NO	NO	NO

8. Conclusion

In this report we went in depth about the Internet product - matrix, which is a virtual workspace for communication purposes.
We went in detail on what is matrix prime features, how do they work and how matrix stands in comparison to his competitors.

The use of E2EE allows Matrix users to feel secure, private and protected. On the other hand, It's unique implementation that allows bridging other communication services establish a viable path for communication with people using other tools without the need of installing them.

We hope to have accomplished both to describe and analyse this service, it's benefits, dangers and impacts in our web society.

9. References

[1] https://matrix.org

[2] https://www.reddit.com/r/privacy/comments/da219t/im_project_lead_for_matrixorg_the_open_protocol/

[3] https://matrix.org/blog/2019/03/12/breaking-the-100-bps-barrier-with-matrix-meshsim-coap-proxy

[4] https://matrix.org/blog/2020/04/06/running-your-own-secure-communication-service-with-matrix-and-jitsi

[5] https://matrix.org/docs/api/client-server/#/

[6] http://www.apache.org/licenses/LICENSE-2.0.html

[7] https://vimeo.com/78350794

[8] https://www.cvedetails.com/vulnerability-list/vendor_id-2044/Matrix.html

[9] https://www.cvedetails.com/product/54433/Whatsapp-Whatsapp.html?vendor_id=19851)

[10] (https://www.cvedetails.com/product/28702/Facebook-Facebook-Messenger.html?vendor_id=7758)

[11] https://en.wikipedia.org/wiki/FBI%E2%80%93Apple_encryption_dispute

[12] https://www.workzone.com/blog/slack-alternatives/

[13] https://www.troopmessenger.com/blogs/mattermost-vs-matrix

[14] https://medium.com/ignation/time-to-replace-slack-who-will-win-mattermost-or-riot-matrix-a090e9cdc219

[15] https://en.wikipedia.org/wiki/Comparison_of_cross-platform_instant_messaging_clients

[16] https://matrix.org/blog/2019/04/11/we-have-discovered-and-addressed-a-security-breach-updated-2019-04-12

[17] https://techcrunch.com/2019/04/19/security-flaw-in-french-government-messaging-app-exposed-confidential-conversations/

[18] https://matrix.org/blog/2017/07/07/a-call-to-arms-supporting-matrix

10. Bibliography

[1] https://matrix.org/

[2] https://jmarques.icu/posts/matrix/

[3] https://www.securemessagingapps.com/