NeXt generation Techno-social Legal Encryption Access and Privacy nextleap.eu
Grant No. 688722 Project Started 2016-01-01. Duration 36 months

DELIVERABLE D3.3
Draft Decentralized Case Studies

Ksenia Ermoshina (CNRS), Francesca Musiani (CNRS), with the collaboration of Harry Halpin (INRIA)

Beneficiaries:	CNRS
Internal Reviewers:	Vincent Puig (IRI), Holger Krekel (Merlinux)
Workpackage:	**WP3, D3.3 Draft Decentralized Case Studies**
Description:	This deliverable presents the results of the M6-M18 in-depth investigation of a limited number of end-to-end encrypted messaging applications and their communities of developers and users. This research is a follow-up to the initial survey of 30 such applications (D3.1) and draws the portrait of the current state of e2e encrypted messaging, by using qualitative methods borrowed from science and technology studies (STS).
Version:	**1.0**
Nature:	**Report (R)**
Dissemination level:	**Public (P)**
Pages:	**71**
Date:	2017-06-30

Intended Audience: Developers (including members of the NEXTLEAP team); informational security trainers; usability researchers; other actors of the encrypted messaging ecosystem; Scholars of Internet architecture and infrastructure from a social science standpoint. May be of interest to regulators as well (e.g. authors of the Privacy directive at the European level).

Project co-funded by the European Commission within the Horizon 2020 Programme

Table of contents

0. Introduction

This deliverable accounts for the first eighteen months of the NEXTLEAP social sciences-driven inquiry into end-to-end encrypted secure messaging applications, the development of the protocols underlying them, and the communities supporting them. This inquiry has been led by CNRS with the collaboration of INRIA and Merlinux especially, and is eventually meant to contribute to NEXTLEAP's protocol, implementation and specification developments, what 5.2 defines as a 'non-hierarchical triangularity'. The present deliverable builds heavily upon 3.1, which accounts for a survey of 30 secure messaging projects which we have conducted in Months 1 to 6 of the project and has been instrumental to guide the selection of the cases to be addressed in more depth here.

As we delve further into below, this deliverable actually does more than account for three detailed case studies. Rather, it paints the portrait of an "ecosystem". In the wake of the Snowden revelations on mass surveillance, secure messaging applications and end-to-end encryption have gained leverage and become a matter of public concern. Yet, developers remain in a state of flux about how to implement security and privacy properties and users have not converged on a single application. For example, there is still debate on cryptographic properties such as forward and future secrecy, group messaging, and repudiation. Furthermore, there is no clear standard to adopt for these properties, with older standards such as Pretty Good Privacy (PGP) not offering them. In terms of privacy, work is yet immature; even the most popular secure messaging applications such as Signal or Wire expose metadata via associating users with their the phone number to use the application.

For all these reasons, next-generation secure messaging appears as un-standardized and fragmented, leading to a state where secure messaging users currently exist in dozens of "silos" unable to inter-operate with each other (see D 3.1). The "silos" effect has, by the way, been considered among the most important obstacles to the adoption of secure messaging apps: "The common trend of creating new secure communication tools and assessing the usability of these tools is a significant obstacle to adoption due to creating fragmented user bases. Also, to reach their communication partners, participants needed to use tools that are interoperable"[1]. This is in stark contrast to the email model, where any email service can openly communicate with another in a federated fashion. Thus, developers of modern secure messaging protocols are facing a number of trade-offs between various design issues, including security and privacy properties, the introduction of group support features, the degree of decentralization of the application, standardization and licensing attempts. In the meantime, vivid discussions are happening around adapting open federated protocols (e.g. XMPP) to integrate the most recent security features (such as future secrecy): new initiatives, based on decentralized (federated or peer-to-peer) protocols are emerging, yet

[1] Abu-Salma, R. et al. (2017) Obstacles to the Adoption of Secure Communication Tools, In *Proceedings of the 38th IEEE Symposium on Security and Privacy* (Oakland), San Jose, CA, USA

still suffering from a number of limitations related to usability and scalability. All these dynamics will be recurring throughout the deliverable as we describe developers' actions and their interactions with other stakeholders (users, security trainers, standardizing bodies, funding organizations…)

0.1. Rationale and structure of the deliverable

This deliverable is meant to provide an in-depth understanding of three different end-to-end encrypted mail and messaging applications. After a survey of 30 cases of encrypted messaging applications and a history of encryption protocols[2], we proceeded to select a few applications that could be studied in more detail using a combined STS and usability studies methodology, described below. We have initially selected three applications based on their underlying protocols and because of the relative accessibility of the developer communities for interviews and observations. The three applications originally selected were Signal (a centralized end-to-end encrypted instant messaging app), LEAP/Pixelated (a federated end-to-end encrypted asynchronous messaging protocol and client) and Briar (a peer-to-peer end-to-end encrypted messaging application for resilient communication using network-layer protection, such as Tor hidden services).

These three cases offered the possibility to address various research questions such as the reasons behind architectural choices (centralized, federated or peer-to-peer), choice of licensing, UI/UX design choices, relations between the underlying protocols and the application level (the three cases develop both a protocol that can potentially be separated from the user-facing client application: Signal protocol, LEAP and Bramble); various solutions for privacy properties (metadata protection); various design choices for group communication; various approaches to security properties (forward and future secrecy; server-side archives; cryptographic deniability; ephemeral or disappearing messaging) etc.

However, when we started our fieldwork on the three selected projects in September 2016, we quickly understood that these projects can hardly be singled out with respect to their connections with other initiatives in the field of encrypted messaging and email. In fact, the underlying protocols used by these three projects gave birth to a number of client-side implementations, forked or actively interacted with various applications in the field. We thus decided to "follow" the three projects as they grow and transform, and use them as our "threads of Ariadne", respecting the loops and knots that these threads were naturally forming on their way. During our fieldwork we had the opportunity to meet and talk with a large number of professionals, ranging from cryptographers to UI/UX designers, trainers and

[2] Ksenia Ermoshina, Francesca Musiani and Harry Halpin, 2016. "End-to-end encrypted messaging protocols: An overview," In: Franco Bagnoli, Anna Satsiou, Ioannis Stavrakakis, Paolo Nesi, Giovanna Pacini, Yanina Welp, Thanassis Tiropanis and Dominic DiFranzo (editors). *Internet science: Third international conference, INSCI 2016, Florence, Italy, September 12–14, 2016, proceedings*. *Lecture Notes in Computer Science*, volume 9934. Berlin: Springer-Verlag, pp. 244–254. doi: http://dx.doi.org/10.1007/978-3-319-45982-0_22, accessed 10 November 2016.

users, who mentioned in our discussions (both recorded and off-the-record) the protocols and tools we focus on here.

Thus, this deliverable is an attempt to tell the complex story of protocols and communities, and eventually, to approach questions of governance of encryption protocols, that will be further developed in our last deliverable. Through a comparative analysis of centralized and decentralized protocols and their implementations, we address here several important questions described in the NEXTLEAP DoW: What are the architectural patterns of successful decentralised systems? How can we build scalable and high-performing privacy-preserving decentralised architectures? Can decentralisation help privacy and anonymity? What are the concrete motivations, values, and characteristics of a user community that leads to success of certain decentralized systems?

This deliverable is structured as follows: after describing our methodology and the ethical requirements that guided our research, the **first part** describes the Signal protocol and its various implementations and forks. We discuss properties and limits of this protocol, and several community-related problems engendered by an important debate over the centralized and Google-dependent nature of Signal protocol. We then move forward to the **second part** dedicated to federated protocols. We analyze OMEMO (and its implementations such as ChatSecure and Conversations), as an attempt to bring some of the recent security properties, such as forward and future secrecy, provided by triple Diffie-Hellman ratchet, to XMPP. We also focus on asynchronous messaging (email) and the problems related to end-to-end encrypted email infrastructures, focusing on LEAP/Pixelated, and the very recent project Autocrypt (described more in details in the deliverable D 5.2). We then move on to Matrix.org as one of the recent attempts to provide interoperability and federation to the field of instant messaging. In the **third part** of the deliverable, we introduce what will be the main focus of D3.5, decentralized and peer-to-peer solutions. We present our initially-selected case study for this type of tools, Briar, and hint at the different questions that will be addressed in the next deliverable: the variety of network-layer protection protocols and tools (Tor, mixnets…), their ability to achieve anonymity and privacy, as well as important problems related to decentralized architectures (group messaging, reputation, synchronization, deanonymization risks…). In **conclusion**, we open up some important governance-related questions, including current attempts of evaluation and classification of various end-to-end encrypted messaging solutions, and draw the lines for our future work.

0.2. Methodology

We adopt the qualitative methods of Science and Technology Studies (STS) to analyze the interfaces of messaging apps as “meeting points” between the intentional goals of developers and the needs of users[3]. In complement with approaches such as traditional quantitative survey-based or protocol-based security usability studies on particular software, STS aim at providing a fieldwork-driven sense-making of emerging systems and

[3] Nelly Oudshoorn and Trevor Pinch (2005}. How users matter: The co-construction of users and technology, Cambridge, United States, The MIT Press.

communities of practice, doing ‘analytical thick descriptions’ of events, artifacts, organizations – in particular, moments of crises, debates, controversies – to try and understand the life of a technical artifact, from its creation to its appropriation and reconfigurations by users, to its becoming a subject of public debate, of governance, of lobbying. A commonly-found term to describe this in STS literature is “problematization”, which refers to the process of inquiring into “how and why certain things (behavior, phenomena, processes) became a problem”[4].

The primary methodology to achieve this goal is to observe, for relatively prolonged periods of time, specific case-study groups or communities, conducting on the side in-depth interviews with their members and reading appropriate documentation such as release notes, accounts of working sessions, etc. This generally requires to carefully select a limited number of case studies, which will be covered in depth, by making hypotheses on their meaningfulness. Ideally these case studies should be representative of wider trends and be cross-checked via multiple interviews or backed up with quantitative studies. Thus, STS employs primarily qualitative techniques from anthropology such as ethnography, but aimed primarily at the role of technology in society, which can provide insight that can form the foundation for quantitative work and future development of the field.

We argue that secure messaging very much needs this perspective at the present time, as an emerging field that is increasingly becoming a matter of interest for the general public. It is a moment in time when users cannot be taken as a ‘separate sample’ from the rest of the ecosystem, including developers themselves (and the different forms they choose to give to their projects, their level of openness, etc.), alongside a variety of trainers, regulators, the media, and so on. At the current level of maturity of encryption as a public concern and a concern of governance, this approach is very much needed and would do well to precede more systematic and quantitative endeavors[5].

Just as we seek to have a nuanced understanding of developers’ motivations and the representations they have of users and their needs, in the tradition of “user studies” developed within STS, we understand users not as a homogeneous and passive group, but as active contributors participating in innovation and co-shaping technologies, which is possible in software development via routes such as bug reporting, pull requests on code, mailing list comments, and in person contact of users with developers. We distinguish users as high-risk or low-risk, depending on their own analysis and description of their situation. Our interviews include both tech-savvy users (who become trainers and teach other users) and low-knowledge users who are nonetheless possibly in a very high-risk situation (i.e. a situation where the misuse of secure messaging would likely lead to death or high prison sentences). In our methodology, at first we focused on interviewing users from western Europe who were not likely in high-risk situations (in particular, in Germany, France, Austria) as well as activists and journalists from Eastern Europe and the Middle East in high-risk situations in Ukraine, Iran and Egypt.

[4] M. Foucault and J. Pearson (2001). *Fearless Speech*, Semiotexte, distributed by MIT Press, Cambridge, MA.

[5] One possible outcome of this qualitative approach from STS is that the very concepts and schemas traditionally used in cryptography and usability themselves can be globally revised if needed.

0.3. Interview Selection Process

Interview subjects that were developers were selected due to pre-existing personal relationships with the cryptographic research community of NEXTLEAP research team members. Although this does provide bias, we believe it can be countered by doing a large number of interviews as well as also recognizing the relatively small size of the global developer community. We also reached out to some developers via the GitLab and GitHub pages of the projects without personal connections (e.g. Ricochet, Conversations).

In contrast, user studies were done with individuals that were selected more by chance via their attendance at training events in their local environments (both high-risk, in the case of Ukraine and Russia, and low-risk in the case of France, Germany, Austria and the United Kingdom) or conferences in pre-selected venues that were determined to be likely to attract high-risk users that lived in areas that, due to the level of repression, made it difficult if not impossible to interview them in their native environment, or would make it such that they could not speak openly in their native environment due to repression. This was the case for users from Egypt, Turkey, Kenya, Iran, where the interviews took place in March 2017 at the Internet Freedom Festival and at RightsCon. All interviews were made between Fall 2016 and Spring 2017, for a total of 52 interviews. We interviewed (17) developers, experts from NGOs focused on privacy and security, such as EFF, Tactical Tech and Privacy International (3) and everyday users (32). Developers from LEAP and Pixelated (PGP), ChatSecure (OTR), Signal protocol and its implementations and forks (including Wire and Conversations(OMEMO)) were interviewed, as well as developers from Tor, Briar and Ricochet that use their own custom protocols.

Within user groups we distinguish between high-risk users (14) and users (including researchers and students) from low-risk countries (18). The developers were all from the USA/Western Europe, and the high-risk users included users from Ukraine, Russia, Egypt, Lebanon, Kenya and Iran. Some high-risk users, due to the conditions in their country, had left (4) or maintained dual residency (2) between their high-risk environment and a low-risk environment. The “users” category also includes a subset (18) of security trainers, e.g. users involved in organizing seminars on security, disseminating privacy-enhancing technologies, practices and knowledge. We interviewed between trainers from high-risk (9) and low-risk countries (9).

0.4. Ethical Guidelines

A specific protocol was developed in order to protect the privacy of our respondents. We let users and developers suggest us a tool of communication of their choice if they wish to do the interview online. These tools ranged from PGP to Signal, meet.jitsi, Wire or WhatsApp. If an “in person” interview was preferred, the interview was recorded with an audio recorder

isolated from the Internet. We use a dedicated encrypted hard-drive to store the interviews. Before the interview we asked our respondents to carefully read two user-consent forms related to the study and ask all the questions regarding their privacy, their rights and our methodology. The two forms were written in collaboration with UCL usability researchers and based on the European General Data Protection Regulation. The documents included an Information Sheet and an Informed Consent Form. The first document explained the purpose of the interview, described the research project and clearly mentioned the sources of funding for the project; provided information on the length of the interview, but also information about the researcher, including her email, full name, academic affiliation and the address of the research institution. The second form described the procedures regarding data processing methods, the period and conditions of data storage; it emphasized the right of the interviewees to demand, at any moment, to withdraw their data from the research. A copy of each document was given to the interviewee. Different forms were used for users and developers.

Additional measures have been taken to ensure better privacy for our interviewees. Thus, the name of the interviewee was not mentioned during the recording. We also adapted some questions to withdraw any elements of context (such as the country or the city, the precise social movement or affinity group a user was involved in and so on), if interviewees asked for this. We respected the right of our interviewees to refuse answering a specific question. However, our questions were specifically designed in order to focus on the tools, with no biographical questions. In this deliverable, the names of both developers and users are mentioned when the user gave permission, but are otherwise, by default, kept anonymized and we resort to qualifying labels such as ‘lead developer of…’ or ‘high-risk user from...’.

1. Signal

“Federated protocols are a thing of the past in this world of ours”
Moxie Marlinspike[6]

As seminal secure messaging protocols such as PGP and OTR started showing their age in terms of security and usability, cryptographers -- unsurprisingly -- wanted to develop new and better protocols in the wake of the Snowden revelations. Open-source developers started deploying efforts to create a next-generation secure messaging protocol. The most advanced and popular protocol in this regards is the Signal Protocol, used by applications such as Signal (formerly TextSecure).

1.1. The protocol history: from Axolotl to Signal

In a nutshell, Signal used per-conversation key material in a similar manner to OTR, and thus unlike PGP did not force complex key management on the users. Like OTR, it

[6] https://github.com/LibreSignal/LibreSignal/issues/37#issuecomment-217231557

maintained properties of repudiation and forward secrecy by virtue of the Axolotl Diffie-Hellman key ratchet[7], but added “future secrecy” so that messages indefinitely in the future cannot be read at any point in the future in the case of a key material compromise[8]. It solved the asynchronous messaging problem by virtue of allowing longer-term pre-keys managed by the Signal server, and offered group messaging implemented as point-to-point messaging.

This protocol then started to attract attention from the academic cryptographic community, and only minor flaws were found[9]. Although alternative approaches were developed and widely deployed, like MTProto by Telegram, these protocols developed their own cryptographic primitives and so received less attention from the academic community, although a number of bugs were revealed[10].

With minor variants implemented in the vastly popular WhatsApp messenger, the core Signal Protocol seems well on its way to clearly replace the use of XMPP+OTR and even a competitive, if somewhat “boutique”, feature for mainstream messaging services (as shown by the adoption of the Signal protocol as an option by both Google Allo and Facebook Messenger). Encrypted messaging applications like WhatsApp, Telegram, and Signal are now the default encrypted messaging application for users that consider themselves to be high-risk. Usability studies have shown that although Signal (similar to OTR) is easy to setup and use, even highly-skilled users fail to use verification correctly. Currently, Signal is centralized, as a single server mediates the setup of the protocol in most widespread deployments (WhatsApp, Google Allo, Facebook Messenger, Wire).

Open-source alternatives that claim to use the Signal protocol or its forks exist, such as the centralized application Wire that uses a fork of Axolotl protocol called Proteus. Parts of the Signal Protocol were copied by a draft XMPP Foundation standard called OMEMO, for use by applications such as Conversations and Chat Secure, which led to usage of Signal’s double ratchet in federated projects. While Signal appears to be widely adopted and considered an improvement over both OTR and PGP, the core Signal protocol remains officially unstandardized, even though the protocol’s creators Trevor Perrin and Moxie Marlinspike have produced an informal draft after considerable “*community pressure*” [as Matrix.org lead developer puts it]

[7] https://github.com/trevp/double
13 https://signal.org/docs/specifications/x3dhratchet/wiki
[8] Katriel Cohn-Gordon, Cas Cremers, and Luke Garratt. On post-compromise security. In Computer Security Foundations Symposium (CSF), 2016 IEEE 29th, pages 164–178. IEEE, 2016.
[9] Tilman Frosch, Christian Mainka, Christoph Bader, Florian Bergsma, Jörg Schwenk, and Thorsten Holz. How secure is TextSecure? In European Symposium on Security and Privacy (EuroS&P), pages 457–472. IEEE, 2016.
[10] Jakob Jakobsen and Claudio Orlandi. On the CCA (in)security of MTProto. In Proceedings of the Workshop on Security and Privacy in Smartphones and Mobile Devices, pages 113–116. ACM, 2016

1.2. A “quasi-standardization” process

While most users we have interviewed, including high-risk users, do not appear to have standardization as an explicit priority, developers care deeply about standards as “something they would eventually be working on”, namely for increasing the ‘dialogue’ between applications and reduce the silo effect:

> *In the long term I am not opposed to the idea of standardizing, it’s great to have a reference for interoperability. [Briar lead developer]*

Standardization is understood as a *reference* and thus as an important communication or mediation instrument, that helps the security community understand each other and build a ground for common knowledge (such as cryptographic libraries), and also guarantees a smoother development of new applications on top of standardized protocols.

Yet a widespread discontent with existing standards bodies is expressed by developers, and that for several reasons. Developers underline recent transformations of these organizations, referring to a previous ‘golden age’ of standardizing bodies, when their mode of existence was closer to that of FOSS communities. Our respondents note the growing importance of private actors as stakeholders within standardizing bodies.

> *My impression of the IETF is that it’s not the same beast it was in the early days. There was a time when it was a group of enthusiastic people who would come to the IETF with an idea that was sort of halfway finished and they’d say look I wanna let everybody know about this, let’s knock it into shape and we’ll all build on it. I think it’s become a much slower moving and more adversarial environment. This area of technologies has attracted more money and more corporate participation and therefore, conflicts of interest [Briar lead dev]*

This institutionalization of the standardizing bodies and their progressive “detachment” from coding communities creates an environment that is less suitable for experiments and unfinished projects:

> *“Recently Moxie wrote a post about disappearing messages, and he phrased it as “automated data hygiene”, and I thought that is more like a convenience. Instead of periodically clearing out your message history you could set a little time out thing and it makes it easier. And I wrote a message to XMPP standards mailing list [about it], and people were not interested, because of various objections to the idea. But I think that’s something that we will implement, it’s just probably not will be standardized because the XMPP community is very conservative. I don’t think… they don’t fully get it. It’s something that users want… so why?... I don’t know. They end up in that old school stuff”. [Developer, ChatSecure]*

Standardization implies ‘translation’ of a protocol as a sociotechnical experiment into a pre-standard, able to ‘enroll’ and convince various agents within evaluation bodies.

Standardization involves collective work that opens up the core-set of protocol authors to include external experts from standardizing organizations, some of them being far from users experiences and needs, and from the “real” economy of the encrypted messaging field -- a process that is hardly appealing to some developers as it is seen as time-consuming on early stages of the project development:

> *I wouldn’t really think about submitting something to the IETF on early stage these days because I think that would probably involve a lot of work to convince a lot of other people to allow it to become a standard… and obviously everybody would have their own thoughts of how better to work. [Briar lead developer]*

Instead, most developers share the philosophy that they would build the application first, and then focus on standardization and decentralization via the use of open standards:

> *I used to work with W3C a long time ago and I am very aware of how they work and that they may have some limitations. We want to get Matrix as mature enough and solid and stable enough then we can pass it over to a proper governance organization but right now it’s still evolving very rapidly. [Matrix.org lead developer]*

In the case of secure messaging, it is still felt that more development is needed on the code, and standardization would only slow down existing development efforts.

Indeed, a new way of ‘*quasi-standardization*’ or ‘standardization by running code’ is being practiced in the field of end-to-end encrypted messaging applications, namely around the Signal protocol. In this process a quasi-standard is defined as “*something that works*” and that’s been “iterated” and redeployed by others. In this sense, all of the various Signal protocol deployments (Wire, WhatsApp, Facebook Messenger, OMEMO-based apps such as Conversations and Chat Secure…) work as ‘crash-tests’ for the protocol, where the protocol gets ‘forged’ by usage:

> *I think the direction to take is a bit more the direction that Signal is taking, when you kind of build a system for a while, you iterate a bit until you think you have something that works well and then you start to document it and if other people wanna interoperate, you talk to them about standardizing on that stage, on a much later stage [...] Recently Signal people started to release specifications for some of the bits and pieces that they using and the idea I think will eventually to have spec of all the things they are using. I wonder whether they will think about standardizing at some point, may be the idea is about delaying it to a later stage of the project, and not about avoiding it [Briar lead developer]*

The Signal protocol, characterized by the “triple” Diffie-Hellman ratchet, is now considered the ‘best practice’ in the field and becomes a ‘trend setter’ for other projects in terms of privacy and security features (e.g. forward secrecy and future secrecy for example). Developers, even those working in federated (Conversations) or peer-to-peer (Briar) projects, see Signal solution as one of the best designs available, even if it is not fully standardized.

1.3. Signal's multiple implementations: markets, governance and encryption

The field of instant messaging applications has been deeply transformed with a number of implementations of Signal protocol, but also because of the growing popularity of other secure messaging tools, such as Telegram, Threema or Wickr that use their own protocols. The turn to encryption has modified the market and brought considerable changes on the level of governance, engaging important private sector players in the game:

> *What is happening last two years [interview was done in 2016] is something that's fantastic, with a number of messengers popping up and also greater publicity around Axolotl or Signal… Snowden also talking about it... So this is something what is really good for the industry. And we've seen it's triggered even the big ones who started using encryption [Alan, Wire CTO]*

According to former Pirate Bay founder Peter Sunde -- creator of Heml.is, an end-to-end encrypted IM application -- the adoption of end-to-end encryption by WhatsApp did not happen without market influence. In fact, before being purchased by Facebook, WhatsApp's cost was evaluated at 19 billion dollars. After an unfruitful attempt to purchase Peter's protocol, they turned to Signal and acquired the Signal protocol for 1 million dollars. After the implementation of end-to-end encryption in WhatsApp, the app was sold to Facebook for a better cost of 21 billion dollars. Peter Sunde recalls:

> *"[WhatsApp CEO] was upset that I and other guys [from Heml.is team] were talking about ideology and politics because he was not interested in politics. He was more interested in the government staying out of his life, and not touching him so much and so on. And he did not wanna call that ideology, which was probably the first problem we had between us. And then he said encryption is not something people care about because no one is interested in what people talk about on WhatsApp because there was nothing important. For him it was more like we need to have something because people are complaining in the media, rather than it's actually about privacy… It was just a business thing. He has no clue what people are using his app for… I don't think he was in touch with his user base, he was more interested in user growth. [Heml.is]*

The element of "political ideology" raised by Peter in the above citation is another important factor that influences interactions and collaborations among different IMs and cryptographic projects. Our fieldwork within developer communities shows that common ideological grounds may become a crucial factor of adopting - or not - of specific protocol choices and design decisions. We will show later here, in the case of Wire, the importance of specific values that cement FLOSS communities, and their influence on fundamental development choices.

However, in the case of Signal's cooperation with WhatsApp (and, by consequence, with Facebook), reflections over social consequences of adoption of end-to-end encryption are used to justify this controversial deal, that became one of the most important collaborations in the recent history of instant messaging. Our respondents, especially among the audience of Ukrainian trainers, underline the positive consequences of WhatsApp's turn to encryption, that, according to them, has solved major problems related to "adoption costs" of encryption:

> *"Since WhatsApp adopted end-to-end encryption, we usually do not spend that much time on instant messaging encryption [during trainings], and recommend to stay with WhatsApp if people already use it. So they can still communicate with all of their friends, and also… it looks familiar, and it does not shock. And people say [during trainings] if they use WhatsApp it's less suspicious than if they use a special app for activists" [I, female informational security trainer, Ukraine]*

The last argument in the quote above mentions an important concern addressed by a number of interviewed users and also observed during cryptoparties and informational security trainings: *does the very fact of using an activist-targeted app constitute a threat in itself?* This refers to the famous "Cute Cat Theory of Digital Activism" by Ethan Zuckerman[11], according to which it is safer and easier for activists to use the same mainstream platforms as those used for sharing "lolcats" pictures, whereas using a tool marked as "activist" may put users under targeted (and thus, easier and cheaper) surveillance.

This concern reveals a shared (but often underexplored) users' anxiety over their "metadata" (even though this particular term is not always used explicitly). Often, in our interviews, we were confronted to an extensive critique of all the existing tools by both informational security trainers and non-technical users. This echoes the findings of another recent usability study of end-to-end encryption tools, stating that "most participants did not believe secure tools could offer protection against powerful or knowledgeable adversaries"[12]. Among user stories about the reasons of adopting encryption, an important number mentioned an experience of their social graphs and "activist" lifestyle being exposed to adversaries because of the usage of specific tools (such as emails on riseup.net, for example, as mentioned by our Russian and Ukrainian respondents). The opposite effect (using an activist-targeted tool as means of "earning trust") was also mentioned by a Russian user, with a story on an undercover police officer using a @riseup.net email account as one of the means to penetrate a student movement mailing list during mass protests in 2011-2012.

The quintessence of this "tool-scepticism" may be illustrated with the following drawing made by one of our respondents, a European high-risk journalist working on war conflicts in Middle Eastern countries:

[11] http://www.ethanzuckerman.com/blog/2008/03/08/the-cute-cat-theory-talk-at-etech/
[12] Abu-Salma et al. (2017), cit., p. 2.

[drawing collected during the interview on Feb 16, 2017]

The user commented the drawing in this way:

> *"In case of a truly secure communication I say something but no one knows what I said and to whom [...] I could have just given you a blank sheet of paper, it would have meant that no traces of a communication act are visible. But as soon as you asked me to draw you something…" [C, male, journalist, high-risk]*

The adoption of encryption by "mainstream" messaging applications (as opposed to "activist-targeted" applications) leads to a specific effect that our respondents called "fish in the sea":

> *"Imagine if I have nothing to hide, but I still use an end-to-end encrypted app, then people who need to hide themselves... like whistleblowers for example... it will be easier for them, say, to disappear in this big flow of cat photos or love messages. So I feel like I am helping someone when I use encryption all the time for all of my communications". [female, low-risk user, tech-journalist, Austria]*

An interesting phenomena of "shared responsibility" arises from the mass adoption of encryption: according to the "fish in the sea" thesis, the more users opt for end-to-end encryption tools, the more secure it becomes for everyone to use these tools, but specifically for high-risk users, whose life and freedom depend on these tools. While mass adoption of distributed (peer-to-peer) apps has a real technical correlation between number of users and privacy protection level (example: Pond or Tor), for centralized apps (like Signal and WhatsApp) or for email encryption the consequences of mass adoption are often described from a "social" or economic standpoint:

> *"The more people use encryption, the more expensive it will be for the governments to read everything. It's not about reaching 100% security… This simply does not exist! It's about making them waste their time and money to decrypt our stuff and in the end they are reading something like "Let's go to eat pizza tonight"... " [male, informational security trainer, Ukraine]*

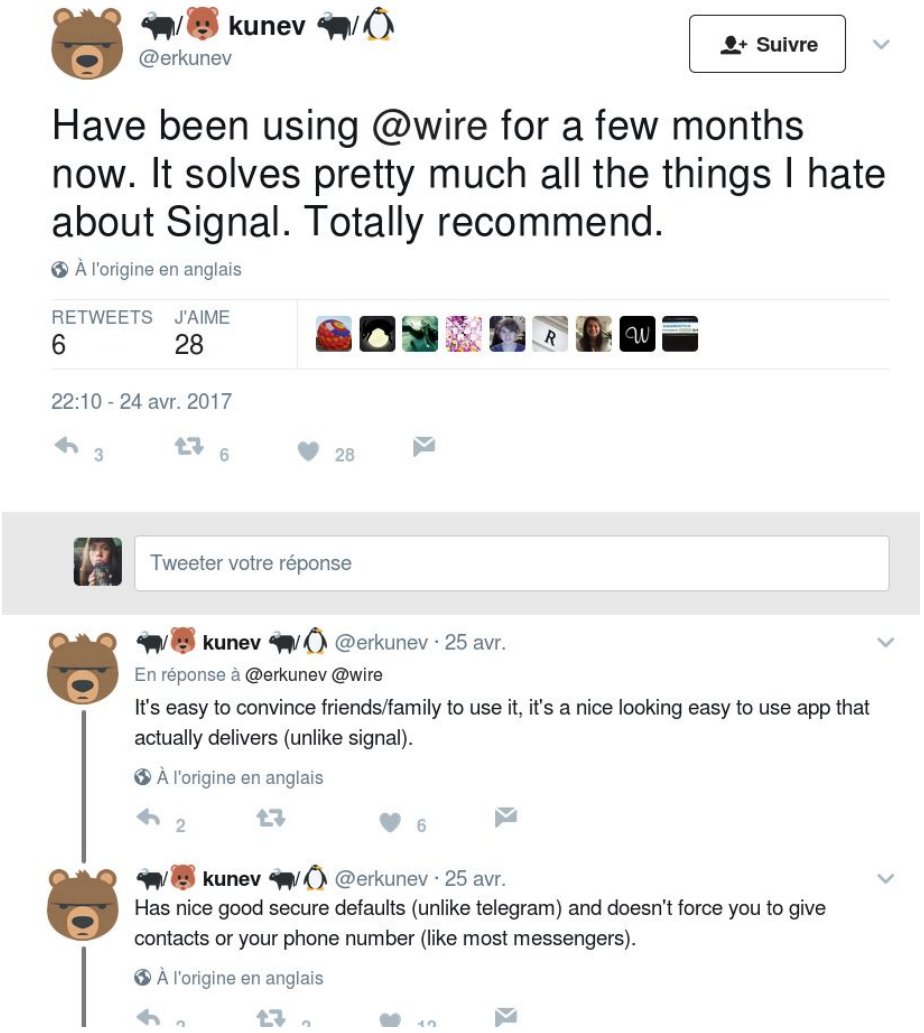
Even though the collaboration of Moxie with WhatsApp and Facebook was subject to controversies and critiques among a number of tech-savvy FLOSS circles, mass adoption of end-to-end encryption had an impact on Internet governance. A critical discourse bridging encryption and terrorism was also present in mass media and at important community gatherings, such as Internet Freedom Festival or RightsCon, where specific sessions on regulation of encryption were held in 2017, bringing together representatives of technical community and EU regulators.

1.3.1. Forking the Signal protocol: Wire and Proteus. Licensing problems and non-standardization as a business-model

One of the most well-known and popular forks of the Signal protocol is called Proteus and is used by the application Wire, aimed at mass adoption. Wire was launched by ex-Skype developers, with a desire to respond to "one of the biggest gaps that was missing on the market, related to privacy and security" [Alan, Wire CTO]. Wire's primary targeted user group is "privacy-aware consumers":

> *"Our users in the first phase are primarily consumers, who care about privacy. And we hope that more and more people would care about it. When I was a kind my parents did not know passive smoking was bad for the kids. When it's started to rising awareness, they stopped smoking. I like to compare passive smoking with rising awareness about privacy when people understand how information can be misused. It's something what needs to be spread more, more people need to be aware" [Alan, Wire CTO]*

Wire is not aimed at activists or tech-savvy audience, but at "ordinary people", as Alan puts it. One of their main concerns was thus to build a usable interface and integrate new features that would distinguish them from other end-to-end encrypted messengers. Thus, Wire supports drawings, GIF exchange, end-to-end encrypted group chats up to 128 members, group video calls up to 6 participants, disappearing timed messages, file transfer. A number of our interviewees as well as social media users have underlined the aesthetic aspect of Wire's UI as an advantage, helping to quickly adopt the app, as opposed to Signal.



Another of Wire's selling arguments, frequently advertised on their social media pages, is the voice calls quality and encryption: Wire offers end-to-end encrypted voice calls using a specific protocol based on constant bit rate encoding[13].

The underlying Wire encryption protocol, called Proteus, is a fork of Axolotl with "*pieces that were needed to have support for the multiple devices as a standalone*" [Alan, Wire CTO]. However, difficulties and tensions have been observed around Wire's attempts to reimplement the Signal protocol. Some of these difficulties are connected with the lack of specifications (documentation)[14]. There is only a draft specification[15] produced recently, as our respondents explain, not without the pressure from other communities:

> *OWS did not prioritize standardizing [the protocol] both because it gave them flexibility to change it as well as allow it to be more valuable to them as intellectual property. However they have just finished standardizing a lot of it, and last 2 weeks they published a very good specification of how it works, and I think to some extent that was basically because of the pressure coming from community like us, because*

[13] https://medium.com/@wireapp/a-major-upgrade-to-calling-9ac8780741a1
[14] This independent audit of OMEMO protocol includes a dedicated part on Signal protocol and refers to lack of documentation: https://conversations.im/omemo/audit.pdf
[15] https://whispersystems.org/docs/specifications/xeddsa/

> *we were writing our own standards for stuff, and a year ago we sent it to them saying hey guys in case it helps here's our standard version of it. But obviously they did not adopt it and maybe they did not read it at all. We've written our own mathematical definition of how it works and send it to them for review. [Matrix.org lead developer]*

So, this lack of specification obliges developers to re-code from scratch sometimes using other programming languages.

> *The problem there was with Axolotl, if you wanted to build it completely from the specification, there was, I would even say on purpose, not enough available documentation. But if you wanted to develop your own implementation, you were pushed or bullied that you have copied their implementation. But their implementation is first of all a different one. If you are working in Rust or Java, it can not be copied it's like 2 different paradigms So then what I did, I was very naive and went to Moxie last year in June, met with him in San Francisco and asked him to review our Rust implementation and we would pay him a very good money for that. Instead he said you can pay 1,5 million, and I will keep your binaries and will help you to get going the implementation. And then I was like… yeah, exactly… What happened afterwards – he said he would sue us and started threatening about it. And you know that's a threat, and our legal guys said that's a threat and it needs to be legally handled. So we sued him for a threat. And then it was just settled. He dropped his charges, we dropped our charges and we are using Axolotl the way we do and how we would like. [Alan, Wire]*

One of ChatSecure's developers explains this conflict as a consequence of a specific licensing politics, that lead to tampering and modifications in the legal terms and agreements between Signal team and other implementers:

> *"Signal protocol is open source under the GPL that means you can't integrate it into a commercial product that's why Whisper Systems were getting large licensing agreements from Facebook and Google and WhatsApp to integrate without opening up all of their source code. Part of that they were incompatibilities with GPL and AppStore specifically. So we needed to get some of the legal language exempted [...] Moxie needs to protect his revenue. So part of his arguments with Wire was that they [Signal] hadn't documented Signal protocol, so there was no open specification, so if you wanted to write a compatible reimplementation, you would have to read the source code which would create a derivative work, which would not allow you to use it commercially because he would argue he still has copyright of the majority of the work". [ChatSecure]*

As pointed out by the LEAP lead developer, the Signal developers are concerned about the technical competence of having third-party developers standardize their protocol ("Moxie is a very good coder and his standards are very high") as well as not be able to update the protocol rapidly enough in response to research and bugs.

This makes it possible to use the non-standardization as a 'business model', where the 'missing parts' of expertise and specification necessary for a proper deployment of the protocol can be offered by the Signal team as a service:

> *You can say OK we will license this technology which is not something I am interested in because I would like it to remain free software. But you can also say "we are the people who understand this technology, it makes sense to hire us if you want to deploy it." If people build systems on top of it, then they pay somebody to contribute changes down into that codebase [Briar lead developer]*

Wire's politics regarding open source have changed during our fieldwork, with an important influence of our partners from Merlinux [Autocrypt project]. Wire started as partially open source, with the server code being closed. They attended the Autocrypt hackathon in December 2016, where they could present their application in front of an audience constituted of well-known contributors to various cryptographic open source decentralized projects (from K9 and Mailpile to Scuttlebutt, GPG and Enigmail). Wire was interesting for Autocrypt because of their move away from phone numbers towards usernames, and potential federation. However, as the only centralized and closed-source project in the room, Wire and its team were exposed to a number of rather critical questions. In our exchange with the Wire team, they underlined the importance of meeting "this community" with whom, they said, they were not very familiar. At this event, the Wire team could discuss with the Matrix.org team over potential federation, and bridging Wire within Matrix.

During the Autocrypt hackathon, Wire's chief of marketing, Siim, shared with us his concern about "reaching out to all these communities" and asked about events to attend in order to present Wire to relevant audiences of open source and privacy activists. After this event, it took Wire 5 months to go fully open source: an announcement was posted on Twitter on April 7, 2017. We have seen the Wire team at different events such as IFF, RightsCon and Re:publica, outreaching to the privacy community. In April 2017, our conversations with Alan revealed that Wire was planning to become federated shortly.

As we have seen from our user survey and observation of security trainings, open-source and licensing choices are less covered in high-risk trainings, as high-risk users do not always associate open-source with security. Open-source was perceived as a less important criteria in the context of an immediate physical threat, as when a proprietary but "efficient" and "easy to explain" solution exists, trainers will give priority to it. For example, in Ukraine WhatsApp is the most recommended application, because it is considered to be easy to install. Trainers consider WhatsApp's proprietary license and collaboration with Facebook in terms of metadata less important than the immediate security it can give to the users. The primary task in high-risk contexts with low-knowledge users is to help them quickly abandon unencrypted tools as well as tools that collaborate with their adversaries.

However, users do care about the sources of funding and the business models of end-to-end encrypted messaging applications. Questions about business models were very frequent on different chats on cybersecurity that we have been observing since September 2016. Users ask for transparency of funding but at the same time show a certain scepticism regarding crowdfunding models (donations) that seem not sustainable enough for an application to be properly maintained.

Recent critiques addressed to Signal concern their dependency on US government funding:

> *"Signal was created by the same spooky regime change outfits that fund the Tor Project. The money primarily comes through the federal government's premier Internet Freedom venture capital outfit: Open Technology Fund, which works closely with the State Department's regime change arm and is funded through several layers of Cold War CIA cutouts — including Radio Free Asia and the Broadcasting Board of Governors".*[16]

Telegram creator Pavel Durov's critics of Signal goes in the same direction, noticing that no US-government funded application could be trusted:

Centralized end-to-end encrypted secure messaging applications propose different business models, though it seems that no ideal solution has yet been found. Wire's project is to propose paid services to users for supplementary storage space (encrypted cloud services). Wire also aims at providing business solutions for end-to-end encryption of the Internet of Things. Threema, one of the few paid end-to-end encrypted applications, asks for $2 contribution per user. Some users, namely members of "Privacy Week" and "Cryptoparty" Austria (2 main privacy-related events in the country), underline that as an advantage:

16 https://surveillancevalley.com/blog/government-backed-privacy-tools-are-not-going-to-protect-us-from-president-trump

> *All of us use Threema [...] I prefer to pay, at least I am sure that I am not the product. [Cryptoparty organizer, male, Austria]*

The licensing choices, business models and politics of open/closed source turn out to be complex sociotechnical processes that are embedded in both community-related interactions, economic context and legal arrangements. This question will be further investigated in our next deliverable focused on the governance of encryption.

1.3.2. Problems and limitations of Signal and other centralized messaging applications

1.3.2.1. Google Cloud Messaging dependencies

Though the Signal protocol has enjoyed extensive attention from academia and has been formally verified, with only minor flaws found, users and tech-savvy communities address a number of critiques to the tool -- critiques that were highlighted during our interviews. Some of these critiques do not concern cryptographic properties of the protocol per se, but are focused on the application's UI/UX, privacy features and particular design choices that bind Signal to other applications and protocols that are, in their turn, criticized by open source communities.

First of all, tech-savvy users (especially crypto trainers from low-risk countries: Austria, Germany, France) point out Signal's dependence on Google services. They see it as a negative point both from the standpoint of open source community ethics (this group of users has manifested an important "allergy" to Google and GAFAM as such), and from the point of view of privacy. The two main critiques concerned Google Cloud Messaging services and Google Play dependencies[17].

Two initiatives were launched in order to "decentralize" or "degooglize" Signal: the F-Droid repository at https://fdroid.eutopia.cz/, which aimed to distribute Signal builds outside of the Google Play Store, but still talks to Google Cloud Messaging (GCM); and a Github repository, which aimed at completely removing the Google components of Signal, even from the server side. The second project was called "LibreSignal". This open source client was using Signal protocol and was defined as "The truly private and Google-Free messenger for Android"[18].

LibreSignal authors stated on their GitHub page that the problem of Signal relying on GCM "is only an issue for people who use a custom Android ROM without Google Play Services. For the vast majority of people who do have Google Play on their phone, this issue is completely irrelevant"[19]. Thus, the Google dependencies of Signal become a problem for a very specific user community, both privacy-aware and tech-savvy, who opt for decentralized and anti-GAFAM communication tools. In this context, the choice of a "Google-free" messenger can also be perceived as a "lifestyle" choice. We have noticed that this choice

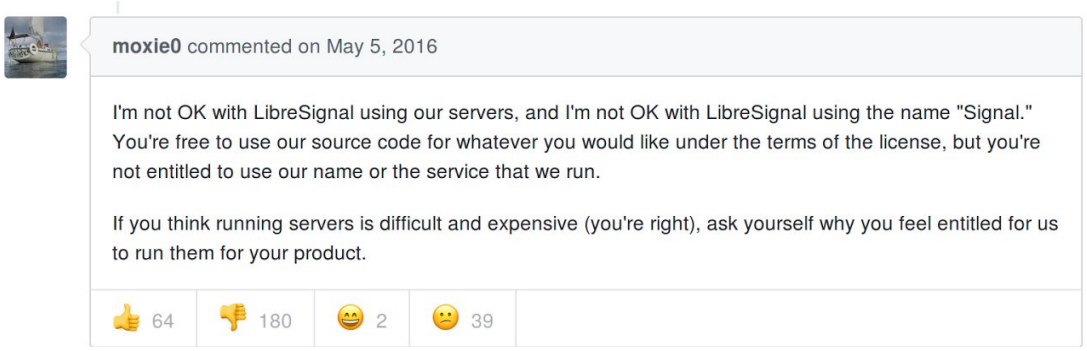
[17] https://github.com/WhisperSystems/Signal-Android/issues/127
[18] https://github.com/LibreSignal/LibreSignal
[19] https://github.com/LibreSignal/LibreSignal

often coexists with alternative hardware choice (Linux phone, Fair Phone, Copperhead OS...), or Google Play-free customized Android. As a tech-savvy user puts it:

> *"If I don't like mainstream in media, if I don't like mainstream in music – why would I like mainstream on my computer?" [Daniel, mail service provider, festival organizer]*

LibreSignal was launched in 2015 but its development was stopped in November 2016, after a long online discussion between the LibreSignal team and Moxie Marlinspike. LibreSignal started as a fork of TextSecure (former name of Signal): several open source contributors built a fork of TextSecure integrating an alternative to Google Cloud Messaging, namely, WebSocket. A bounty hunt was launched to develop a fork of Signal with WebSocket[20].

LibreSignal published a build on FDroid. However, Moxie Marlinspike demanded first of all to change the name of the fork (not using TextSecure or Signal), and secondly not to use Open Whisper System servers. He also insisted that no distribution of Signal via FDroid was possible.



Source: https://github.com/LibreSignal/LibreSignal/issues/37#issuecomment-217211165

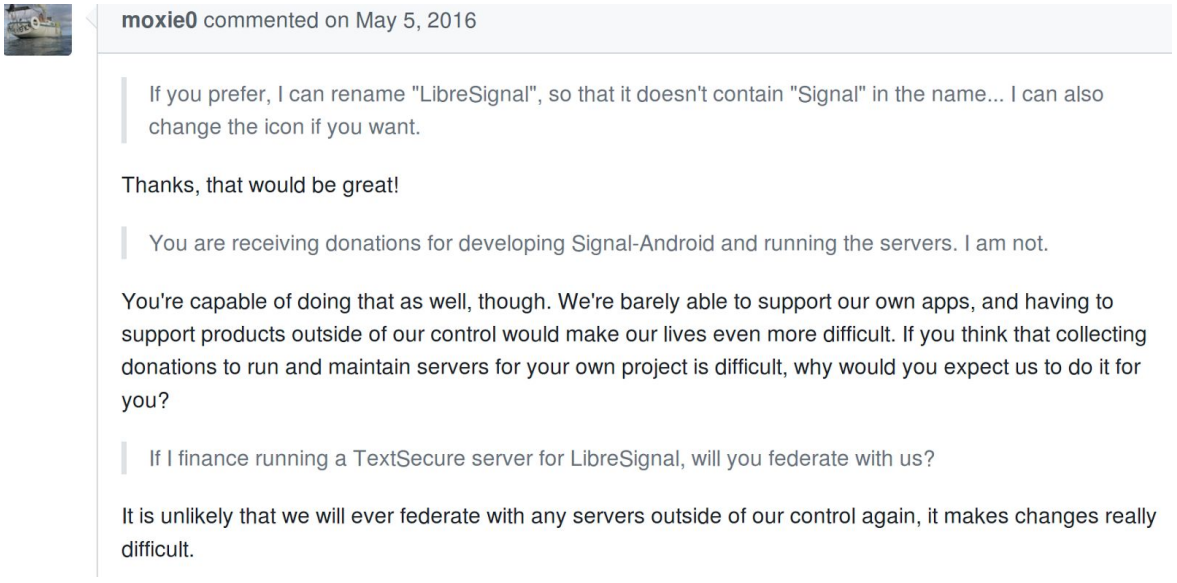
The debate between LibreSignal and OWS raised several issues that are important for our research and for the overall understanding of the socio-technical and legal problems related to the debate on federation/centralization:

- the infrastructure for federated projects (servers) seems complicated to set up and maintain, compared to the 'forking' of the client-side code and implementations of patches (such as Websocket for example). The latter does not demand coordination efforts between different agents: developers can fork and experiment on their own without having to care for the infrastructure (servers).
- however, the process of publishing a build on any of the app repositories (would it be Fdroid or GooglePlay) demands a greater coordination and enrolment of a multitude of actors (and respect of a number of legal and technical standards). This problem of federated protocols was also underlined in our interviews:

20 https://www.bountysource.com/issues/35722527-create-proper-pull-request-to-add-libresignal-s-websocket-support-to-ows-signal

> *"Obviously I think that decentralization is good otherwise I wouldn't be doing Conversations or using XMPP. But of course it's more challenging to do. Because you have to coordinate with a lot of different vendors". [ChatSecure dev]*

- In the case of Signal the problems of forking and federating the protocol turn out to be both legal (legal threats around the Signal trademark), technical (quality of LibreSignal builds criticized by Moxie) and economical (costs of running servers);



moxie0 commented on May 5, 2016

> If you prefer, I can rename "LibreSignal", so that it doesn't contain "Signal" in the name... I can also change the icon if you want.

Thanks, that would be great!

> You are receiving donations for developing Signal-Android and running the servers. I am not.

You're capable of doing that as well, though. We're barely able to support our own apps, and having to support products outside of our control would make our lives even more difficult. If you think that collecting donations to run and maintain servers for your own project is difficult, why would you expect us to do it for you?

> If I finance running a TextSecure server for LibreSignal, will you federate with us?

It is unlikely that we will ever federate with any servers outside of our control again, it makes changes really difficult.

Source: https://github.com/LibreSignal/LibreSignal/issues/37#issuecomment-217231557

This long debate on LibreSignal GitHub page brought Moxie Marlinspike to publish his ideas regarding federation and centralization in the famous blog post "The Ecosystem is Moving"[21]. Signal team was accused of not being "truly" open source because of their refusal to draw upon open federated protocols:

> *"The problem is that Moxie and OWS don't want Signal to be 100% free/libre software (it is not important for them...)" [user mimi89999 commented on May 4, 2016[22]]*

> *"They are not interested in removing the Google GCM dependencies, so unfortunately this is not possible. Moxie has been quite explicit on this point, several times". [user paride commented on May 4, 2016]*

[21] https://whispersystems.org/blog/the-ecosystem-is-moving/
[22] https://github.com/LibreSignal/LibreSignal/issues/37#issuecomment-217211165

cjeanneret commented on May 6, 2016

I love the fake opensource provided by OWS : use the code, but go f*** yourself for federation, out-of-store distribution and acceptation.
Really an open mind as we want to see more and more. 👎

Seriously, if OWS continues that way, even libreSignal will be dropped from my devices, and no other apps from this organization will ever come back.

Currently, people seem to be wanting to help, provide resources (like servers) and are apparently just asking for federation (that was already working with CM), but all we hear are "no", and not alternative.

People that actually *care* for privacy just don't want to have gapps on their device. This is a basic step toward a bit more control over data and privacy.
How do you think those people will use your apps, **@moxie0** ? Seriously? Locking out non-gapps-ed users is probably the worst idea you might have, since the drop of sms encryption with TextSecure (and the rebranding to signal an so on).

OWS policy just goes against real privacy, letting people with gapps, thus google services (mails, sync and so on) without the possibility to NOT let that on their phone.
That's against all logic.

Source: https://github.com/LibreSignal/LibreSignal/issues/37#issuecomment-217231557

Starting from February 21, 2017 Signal can be used without Google services. As LEAP creator explains it, the long process of “degooglizing” Signal is related to an important technical problem, that of notifications, that demand infrastructure investment. The connection with Google Play is explained as a desire to keep track/stats on the usage:

> *LibreSignal was an attempt to avoid the Google services push. The problem with push on mobile is that it’s a very difficult problem to be able to push events to mobile devices without burning the battery and without setting up a complex infrastructure for that. So you can rely on Google infrastructure for that and then it’s more efficient and they do it for free. But it’s controversial for some reasons mostly because it requires to install all the google services on your device, which means you basically have to turn your device over to Google that always finds ways to gather your data so you can’t run Signal on a Google-free device. That’s recently changed, Signal very recently supports push notifications without Google services. Moxie still insists that no one should distribute Signal application outside the Google play market because he wants to get bug reports and data who’s running and how they are running it when it’s crushing. [LEAP lead developer]*

The problem of Google Cloud Messaging dependency and push-notifications was addressed by other projects that found different solutions to this problem. For example, Chat Secure uses their own system for push notifications that claims to have better privacy properties than the one used in Signal:

> *“Our push messaging server… Your device registers with a randomized username and password first time you install it. And then your device fetches a bash of randomized tokens from the server that if sent back in post to the server can then trigger a message back to you and these tokens you can distribute however you*

> *want. So you can give them directly to your contacts and then they send a post and we don't know who your contacts are, you can send them from everywhere, you can use Tor, and all we know is that it's a randomized token. And you can rotate and cycle through these tokens, so if an end contact wants to use a token once and then throw it out, if they have other ones they can swap those and you sort of rotate the tokens periodically and what we ended up doing for XMPP specifically is there's another mode where instead of handling a token to your contacts directly, if you hand a token to your XMPP server the server can then send pushes to you rom contacts that don't support this alternative mechanism. So anyone using any XMPP client on the other end can still send pushes without having to support anything". [ChatSecure lead developer]*

Wire claims to be "pretty much Google-free", however it uses Amazon for push-notifications and analytics. Wire's CTO describes the transition from Google services as long and difficult and also dependent on Google's "reputation" privacy-wise:

> *"We are pretty much now Google-free. But it took time… It wasn't easy. As you know may be, when we started Wire 3+ years ago, Google was already under suspicion but there were some people in the company who were Google fans, now they are completely disillusioned but it took us also some time to take away all of the Google stuff. [Alan, Wire CTO]*

1.3.2.2. Privacy-related issue: phone number as identification mechanism

Another critique addressed to Signal concerns the usage of phone numbers. The Signal Protocol provides confidentiality but requires exposing phone numbers to the server and so allows the server (although the server currently minimizes logs) or a passive adversary to capture all the metadata, including the social graph of users. Today, phone number is used by the majority of popular IM apps as an identification mechanism which is considered as one of the central problems of post-Snowden end-to-end encryption IM apps.

Peter Sunde, Heml.is and Pirate Bay developer[23], as well as Wire's CTO, Alan, argue that phone number and the possibility to share user's phone contacts are important for contact discovery: they help users build their network of contacts and quickly find other users in their phonebook who use the same messaging app. An easy contact discovery is important, according to Alan, to 'fidelize' users and make them 'stick with' a new messaging app:

> *In the beginning we didn't have a possibility to have your phone number to register, we wanted to have just emails. But then we had amazing 2 days where hundreds of thousands of downloads, even went to millions, that happened. But we could not build a good network effect. People were just downloading it individually and not connect to each other. So we went towards this side, that you have your address book uploaded so it will help you to have some contacts, established and then once*

[23] Even though Peter Sunde is negative about exposing users' phone numbers, he mentioned in the interview that the usage of phone numbers helps to solve the problem of "users acquisition". The alternative way of identification, involving usernames instead of phone numbers, demands, according to Sunde, to already have "*a big user community*".

> *people have contacts, they stick with Wire. But we had a lot of people who had not a single contact and they left. [Alan, Wire]*

That is also the reason why Wire has implemented a chatbot called "Otto" who becomes user's first contact and a way to get familiar with the app and test the app:

> *Since our UI is different from WhatsApp or Viber, feature wise we have new features that they did not have. Otto is here to help you to learn how to use these features. And also for people before they add contacts, it's a possibility to chat and to call him, just to keep them still on Wire, show them the potential of it so that they feel comfortable inviting their friends. And Otto is also end-to-end encrypted. [Alan, Wire]*

Indeed, Alan points out an important problem, also underlined by users in our interviews, that we call "migration problem": how do we migrate our contacts from one IM to another? How do we 'stick with' a new IM if no one from our social group is using it? An important number of our respondents among non-tech savvy users explains weak adoption of PGP or Signal because of the unpopularity of these tools among their peer groups: "no one uses it". Moxie Marlinspike also underlines this social network paradox:

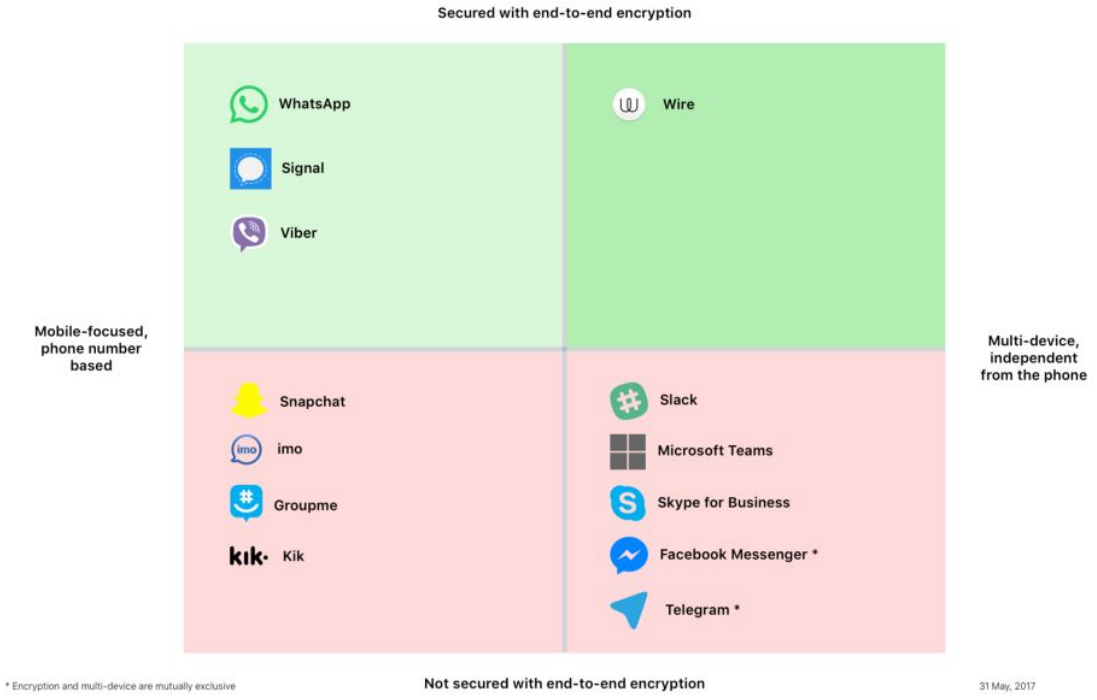
> *Social networks have value proportional to their size, so participants aren't motivated to join new social networks which aren't already large. It's a paradox where if people haven't already joined, people aren't motivated to join. [Moxie Marlinspike, The Difficulty Of Private Contact Discovery]*[24]

This effect also becomes an important factor for advising IMs during security trainings. For example, several Ukrainian trainers that we have interviewed tend to recommend WhatsApp and Gmail over Signal, and PGP over Thunderbird, partly because users already have some of their contacts using these tools, and the transition/adoption costs are lower.

Wire has recently moved away from using phone numbers as authentication method, and now gives users an option to sign up with email address. On May 25, 2017 they added a possibility to delete a previously-registered phone number and adopt email-only authentication. This design solution was partly made because of the critiques addressed to Signal, and as a way to distinguish from it (as a selling argument). Wire has largely promoted this new feature on social media; it was recommended by Edward Snowden. Wire also published specific guides on "staying anonymous on Wire" that became one of their "killer features", giving users a possibility to subscribe using a fake email account, over VPN or Tor, providing almost no personal information[25].

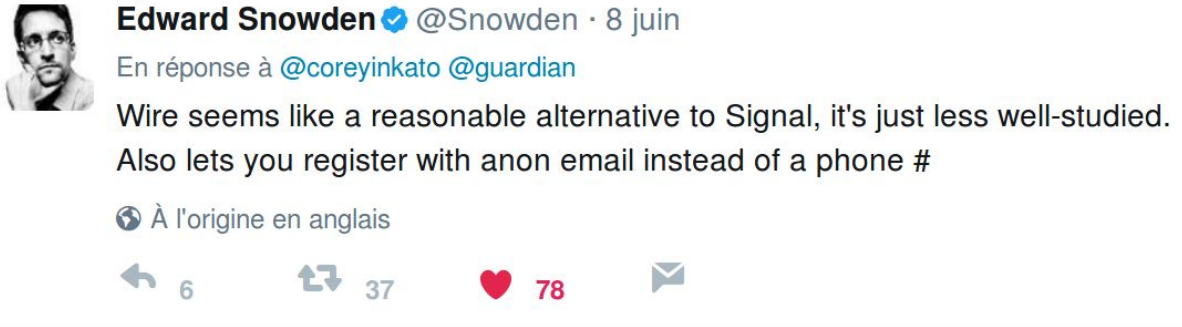
[24] https://whispersystems.org/blog/contact-discovery/

[25] (https://medium.com/@wireapp/staying-anonymous-on-wire-22faa13aba4d



Source:
https://medium.com/@wireapp/product-design-decisions-for-secure-messengers-e8a5e7d1a373

The anonymity feature has become a popular argument in favor of Wire over Signal, as we could see both from our interviews and from social media and press analysis. This was also the reason for Edward Snowden to recommend Wire as an alternative to Signal:



Source: https://twitter.com/Snowden/status/872880404780503040

Usage of phone numbers appears to be related to the centralized architecture as such, and makes it harder to make a transition towards federated systems. This issue is closely linked to the critique of telecom operators, perceived as fundamentally driven by a proprietary culture, and untrustful privacy-wise. The phone number as authentication method increases metadata collection. It is opposed to alternative methods of identification, used in federated systems:

> *"With Signal it's impossible to create a decentralized system because phone numbers aren't decentralized. With XMPP it's like an email address. Even users who*

> *aren't technologically savvy can understand: this is my user ID, and this is my server. We have no idea what server you use and what's your ID. It's on the client side, and we don't need it for pushing messages".* [ChatSecure]

Usage of telephone number for authentication has been source for serious leaks and attacks on other centralized secure messaging apps like Telegram. Namely, on April 29, 2016, the accounts of two opposition activists from Moscow have been hacked by the state: in collaboration with the telephone company MTS (both activists were using this operator for their mobile communications), they deactivated SMS delivery service during the night of April 29 for the two users, and reclaimed recovery codes for their Telegram accounts. When those codes arrived to the MTS servers, the police could intercept them and use to get access to the accounts. Pavel Durov highly recommended using two-factor authentication in order to prevent this kind of attacks (the method is also recommended during cryptoparties, to reinforce protection of accounts on centralized e2e messengers).

Overall, our research has shown that high-risk and tech-savvy privacy concerned users are concerned by their phone number being used for identification. The usage of existing users' social graphs (and especially using 3d party oAuth mechanisms, such as twitter, gmail or facebook oAuth services) was widely criticized. Wire's "matching" feature suggesting new contacts based on user's Wire contact list has received a wide critique on Twitter and was qualified as "creepy" by some of our respondents (namely, tech-savvy security trainers from Austria and Germany).

Thus, there is a strong need for an alternative contact discovery mechanism, that would give users control over their social graphs. Some alternative design ideas of decentralized identity-management systems, based on blockchain technologies[26], have been suggested recently. These alternatives were discussed and presented at the European Lab Forum (May 2017) and Elevate Festival (March 2017). Nextleap's project ClaimChain, as well as a newborn serverless email-based end-to-end encrypted Delta Chat[27] messenger, using Autocrypt specification, could constitute potential answers to this demand.

1.4. Metadata collection: an unsolved problem for centralized end-to-end encrypted messaging applications

While some secure messaging solutions, like Ricochet or Briar, use Tor to hide the IP address of their users, none of the popular Signal-based messengers hide metadata[28]. However, users often believe Signal protects metadata and keeps their conversations anonymous. Even though Signal does not log metadata (outside the last time a user installed the application and the time of last connection), a real-world adversary could simply watch the encrypted messages going in and out of the Signal server in order to capture the

[26] For example, Jolocom, https://github.com/jolocom or Xdi2 protocol https://xdi2.org/ and so on. A large number of such decentralized identity systems is listed here: https://github.com/peacekeeper/blockchain-identity

[27] https://delta.chat

[28] Jakob Jakobsen, Claudio Orlandi (2016). On the CCA (in)security of MTProto. In Proceedings of the Workshop on Security and Privacy in Smartphones and Mobile Devices, pages 113–116. ACM, 2016.

social graph of users. More easily, a captured device would reveal the phone numbers of all other Signal contacts. Although some applications such as Ricochet do achieve protection against this kind of adversary that high-risk users worry about, high-risk users are in general much more aware of Signal and find it easy to use, while all interviewed users were unaware of the existence of the anonymized Ricochet application.

Wire claims, however, to reduce metadata storage of their messages to 72 hours. Metadata become a selling argument that distinguishes them from other IM apps:

> *With Wire what was important was end-to-end encryption as a first step. But besides there is a whole new dynamics that needs to happen that's related to all of the metadata and what is it used for. How is it used. So the first step is to absolutely minimize all of the metadata. This is also the part we are currently heavily working on, and some of the metadata we have was here for historic reasons. When you launch an application first without end-to-end encryption and then you upgrade it to support end-to-end encryption there is still some of the junk left from the previous implementation. This is something that we are cleaning it up, during next few weeks some of the last and very important stuff related to metadata is going to disappear. Some of the metadata that's gonna be in the system is gonna be for 72 hours so that during that time we either can help people if needed in case of a troubleshoot, or if certain abuse is happening than you can follow it up. [Alan, Wire CTO]*

Recently, Wire one-to-one voice calls are said to leave no metadata:

> *Call setup messages are sent as end-to-end encrypted messages and Wire servers have no knowledge of the contents or type of these messages. That means the servers don't have access to who called who, when and for how long. To further conceal any potential calling message patterns, messages to tear down a call are sent peer-to-peer as part of the media stream*[29]*.*

Many developers have come to consider the improvement of privacy by minimizing metadata collection to be the second most important feature after the development of end-to-end encryption. Yet many developers confuse whether or not a third-party adversary could be passively monitoring the communication, thus collect metadata, with whether or not they as developers personally collected data -- as exemplified by one developer that stated simply "*I do not have anything in my code that would let me know how many people are watching the app right now.*" However, many developers also think they would have to collect some metadata in order to interoperate with features such as push notifications of arriving messages, but they try to limit the harm:

> *"With introducing the push messaging it's the first time we're exposed to possible metadata. But we don't log anything, we don't know who is talking to who, we don't log any information. We designed it specifically to be as useless as possible as far as any information. There's no logging and whatnot… And I think we're gonna enable server operators to run additional components that reduce exposures as well. There's now some level of registration with our servers but it's optional. It's required to use push-messaging, but if you use an account with TOR, we disable this information. There's nothing we can do to figure out who users are based on the*

[29] https://medium.com/@wireapp/a-major-upgrade-to-calling-9ac8780741a1

> *information we do have. If someone says we want all the records of the person with this Jabber ID, it's not possible to correlate any information. We don't ask for phone numbers". [ChatSecure lead developer]*

Most developers who were aware of third-party data collection of metadata were supportive of using Tor and on disabling the collection of phone numbers in particular, but lacked a comprehensive plan to minimize the collection of data as such. High-risk and low-risk users generally supported reducing data collection and increasing privacy, although often the encryption of data was assumed to hide metadata by non-trained users. Developers and information security trainers underlined the urgency to find a reliable solution to the metadata collection problem and state that nothing in the field of end-to-end encrypted instant messaging apps offers good metadata protection. Developers and trainers associated the leaking of metadata with centralization:

> *"Metadata connects you weirdly with other people, and there's more sense in the metadata than in the data itself for technological reasons [...] No one from the messaging apps is trying to solve that. Instead they suggest to sync your address books so they know exactly who you're talking to even though you trust them to somehow make it into hashes or whatever. That's the issue we are not solving with the apps, we make it worse. We now have centralized servers that become honeypots, and it's not about the data, it's about the metadata" [Peter S., Heml.is].*

1.5. Alternative (also centralized) to Signal: Telegram

Telegram remains the most popular secure messaging app for some user populations -- in our research, we have particularly focused on Russian users, for whom it is the case. The usage of Telegram varies according to users' goals and threat models. Several functionalities of the app make it convenient for different user groups: chats, secret chats, group chats, bots and channel broadcasting. Users faced with a low level of threat, not associated with any political activities, tend to adopt Telegram as an alternative to WhatsApp and SMS for everyday conversations with their peer groups. Many activists and privacy-concerned users are aware of the absence of "privacy by default" in Telegram chats (client-to-server encryption), however they do not always opt for a "secret chat" option that offers end-to-end encryption. This user group also adopts two-step authentication and self-destruct timer options. Functions such as "Group chat" are used for group conversations and are popular among activists, journalists or researchers for organizational purposes, as an alternative to Google Groups or mailing lists. For example, one of our use-cases, a group of researchers working in Eastern Ukraine, use Telegram on a daily basis to coordinate research activities, discuss fieldwork, materials and other organizational information. However, they do not rely on Telegram for very sensitive discussions and prefer face-to-face offline meetings. During Russian anti-Putin protests in spring 2017, Telegram was also very popular regardless security flaws that had been previously detected and the absence of end-to-end encryption in group chats. Telegram group chats remain popular among high-risk users despite the fact that encryption for group chat offered in Telegram is very basic, defaulting to simple TLS rather than the more advanced M-PROTO protocol for group chat.

The popularity of Telegram in Russia can be partly explained by the reputation of its founders, Nicolai and Pavel Durov, Russian-born developers and entrepreneurs. Pavel Durov, the founder of Vkontakte, the most famous Russian social network, is colloquially referred to as the "Russian Zuckerberg" and became *persona non grata* in Russia after his refusal to collaborate with the FSB[30]. Telegram's quick rise on the market of messaging apps is of particular interest as it tells us a lot about the socio-economic factors that influence the success of an innovation in the field: it was when Facebook bought WhatsApp (followed by a several hours black-out for the latter), that the Telegram download rate exploded. As opposed to WhatsApp, Telegram can publicly underline its non-for-profit character and lack of ties with any commercial or governmental services.

While the Russian version of Telegram was released in 2012, before the Snowden revelations, Durov claims that the international version of his tool was inspired by the whistleblower:

> *"In 2012 my brother and I built an encrypted messaging app for our personal use—we wanted to be able to securely pass on information to each other, in an environment where WhatsApp and other tools were easily monitored by the authorities. After Edward Snowden's revelations in 2013 we understood the problem was not unique to our situation and existed in other countries. So we released the encrypted messaging app for the general public"*[31].

As Telegram servers are located in five different countries around the world, outside Russia, its broadcasting function is used by censored media as a way to bypass the blockage, and by bloggers as an alternative to Facebook and traditional blogging platforms (for example, Alexey Navalny's popular bot on Telegram and the Grani.ru channel and bot, amongst others). However, unlike private communications on Telegram, public channels may be read and blocked by the Telegram technical team. As of January 2016, 660 channels attributed to ISIS were blocked.

While recent research in usability showed that Telegram was suffering from a number of important usability problems[32], our survey has shown that this application was widely adopted by high-risk users, in Russia, Ukraine, but also in Iran. In Russia, Pirate Party activists are using Telegram widely (the group of PPR members on Telegram counts 240 users as on June 17, 2017). Iranian users were pointing out several factors in favor of Telegram, not all of them being directly connected with privacy or security properties of the app. Users were especially happy with Telegram's functions such as stickers: Telegram offers a possibility for users to generate their own stickers. Iranian users explained in the interviews that adding stickers representing muslim in their everyday environment, but also

[30] http://www.reuters.com/article/idUS74722569420130830
[31] http://www.dazeddigital.com/artsandculture/article/24279/1/pavel-durov
[32] Ruba Abu-Salma, Kat Krol, Simon Parkin, Victoria Koh, Kevin Kwan, Jazib Mahboob, Zahra Traboulsi, and M. Angela Sasse. The Security Blanket of the Chat World: A Usability Evaluation and User Study of Telegram. In Proceedings of the 2nd European Workshop on Usable Security (EuroUSEC), Paris, France, 2017.

specific stickers on Iranian political situation was a very important feature that differs Telegram from “first world” apps that only focus on “western” lifestyle and emoticons.

During our web ethnography focused on usages of Telegram we have observed how Telegram stickers were used to attract attention to a specific cause, as an alternative way of spreading information and setting a trend. Thus, after the arrest of the Tor exit-node operator Dmitry Bogatov in Russia on April 10, 2017, a group of Russian Internet freedom activists (some of them members of Pirate Party Russia), who used Telegram group chat to coordinate #freeBogatov campaign, generated a stickerpack dedicated to the Dmitry Bogatov case.

Apart from stickers -- that become at the same time tools for community-building and for personnalisation of this messaging app -- other elements of personnalisation are present in Telegram, such as the possibility to set any image as wallpapers/background for the app, use a wide range of colour themes. All these functions are absent in Signal.

The trust in Telegram is, according to our interviews, not a trust in technology, but a trust in the person and his political position:

> *“User1: Maybe you should not discuss that over Telegram?*
> *User2: Why not? Pashka Durov will never give away any of our data, he does not care about Russian police” [from online discussion in a group chat “Soprotivlenie” [Resistance], posted on June 11, 2017]*

In the case of Telegram, it is strikingly interesting to observe how the actual cryptographic protocol and security and privacy properties (Signal does use mobile numbers and stores users metadata) lose their importance for users, compared to UI/UX features and to the reputation of the app's creator. Thus, motivations for adoption of privacy-enhancing tools are also dependent on the reputation of their creators, as well as shifting geopolitical alliances that may affect the reach of government agencies.

2. Federated projects for end-to-end encrypted communications

When it comes to the community of developers debating online, the tensions between centralization and decentralization go hand-in-hand with debates over standards. A well-known argument in favor of centralization and against standards was published by Moxie Marlinspike (Signal lead developer) in his blog. His blog-post called "The eco-system is moving" has attracted considerable attention and is widely quoted by developers, trainers and advanced users as a reason not to use open standards but opt for centralization. According to this point of view, centralization offers a better control 'by infrastructure' or 'by design', while federation can be "dangerous" in terms of security, as it is hard to audit all the different implementations of the protocol and ensure correct updates.

2.1. Decentralization: to which extent? Social and technical split in security community

The Signal protocol itself does not impose centralization technically, however, the evolution of the market of end-to-end encrypted messengers demands centralization, according to Moxie:

> *"One of the controversial things we did with Signal early on was to build it as an unfederated service. Nothing about any of the protocols we've developed requires centralization; it's entirely possible to build a federated Signal Protocol based messenger, but I no longer believe that it is possible to build a competitive federated messenger at all" [Moxie, https://whispersystems.org/blog/the-ecosystem-is-moving/]*

Briar lead developer tends to agree with that position when he explains why the Briar app public release is taking that much time:

> *We have an app that runs, it's fairly stable, it looks like a fairly complete finished application. The reason that we do not release it now is because we need to change data format and protocol details, and keeping interoperability with older versions in the decentralized network when you're changing data format and protocol details is really difficult. So we wanna postpone having to deal with that problem as late as*

> *possible by just doing limited test until those things stabilize before doing public releases [Michael, Briar]*

Decentralization seems to bring more problems into keeping older and newer versions of an application interoperable. Thus, Briar implements 'expiry dates' into their releases in order to make it technically impossible to communicate between users of newer and older versions. This problem was also present in Matrix's transition towards end-to-end encryption, with a multitude of older clients being incompatible with e2e. Peter Sunde also underlines problems related to federation and speaks in terms of trade-offs that developers have to make when designing protocols:

> *The idea to begin with* [when developing Heml.is messaging app] *was to do federated servers, but then we realized that it was not possible because we also wanted to protect social graphs of how people communicate and in order to do that we needed to make sure that we have control over the servers… All these ideas and technological solutions are always a trade-off. [Peter Sunde; Pirate Bay, Heml.is]*

In his blog post, Moxie brings forward a market-centered argument, focusing on constantly moving consumer's demands and the overall IM 'ecosystem'. He explains success of popular online communication systems as a result of transforming federated protocols into centralized services:

> *Cannibalizing a federated application-layer protocol into a centralized service is almost a sure recipe for a successful consumer product today. It's what Slack did with IRC, what Facebook did with email, and what WhatsApp has done with XMPP. In each case, the federated service is stuck in time, while the centralized service is able to iterate into the modern world and beyond [Moxie, https://whispersystems.org/blog/the-ecosystem-is-moving/].*

He also claims social centralization (in terms of control over changes in protocols) is needed to successfully respond to unsolved challenges, such as metadata protection:

> *If anything, protecting metadata is going to require innovation in new protocols and software. Those changes are only likely to be possible in centralized environments with more control, rather than less. Just as making the changes to consistently deploy end to end encryption in federated protocols like email has proved difficult, we're more likely to see the emergence of enhanced metadata protection in centralized environments with greater control [Moxie, ibid.].*

Other critiques addressed to decentralized protocols, such as XMPP, concern weak support for rich media and difficulties in development of XMPP on mobile:

> *"XMPP is an example of a federated protocol that advertises itself as a "living standard." Despite its capacity for protocol "extensions," however, it's undeniable that XMPP still largely resembles a synchronous protocol with limited support for rich media, which can't realistically be deployed on mobile devices". [Moxie, ibid.]*

Finally, because of the relative freedom in client deployment, according to Moxie's critique, XMPP suffers from a fractured client support, and the difference between various XMPP

clients results in lack of consistency regarding feature support in various clients. According to Moxie, that may create confusion for users:

> *Someone's choice to use an XMPP client or server that doesn't support video or some other arbitrary feature doesn't only effect them, it effects everyone who tries to communicate with them. It creates a climate of uncertainty, never knowing whether things will work or not. In the consumer space, fractured client support is often worse than no client support at all, because consistency is incredibly important for creating a compelling user experience [Moxie, ibid.].*

However, developers from PGP, XMPP, and other projects (LEAP, Ricochet, etc.) strongly oppose this critique from Signal in their own blog-posts. For example, a developer involved in Conversations and OMEMO states that the "extensibility of XMPP is not a danger in itself. A good extension will spread naturally. Moreover, there's a permanent incentive to innovate in XMPP." This has led developers in certain communities to try to standardize versions of the Signal protocol applied to federated projects, such as the development of the OMEMO standard in the XMPP Foundation.

In terms of PGP, developers from encrypted e-mail providers such as LEAP are still trying to implement the PGP standards correctly, and hope in the future that the PGP standards can be extended to have properties such as future secrecy and easier key management like Signal, but they see fixing the standards as a far-off long-term goal. In contrast, Signal developers believe these older protocols like PGP and XMPP actually harm user security, and should be abandoned. The LEAP team strongly defends open federated protocols:

> *Moxie came out recently and 'arrowed' himself as a centralist in his blogpost about this topic. I disagree with him. His argument is that… open federated protocols are simply too slow to adapt and they are too difficult to implement because there's too many different players, you make a change and you break everything […] I think just because there are examples in the past of being difficult to change protocols does not mean that the idea of open protocols is dead. I think there's definitely ways in which we can recognize the problems with open protocols so that we still can have interoperability but not be locked in an unchanging [ecosystem]". [Elijah, LEAP]*

The arguments against centralized architectures that were advanced by our respondents concern several aspects: first of all, the danger of "trusting one person behind the service" ["putting all eggs in one basket", as Elijah puts it]. From this standpoint, the personal responsibility of a centralized service provider is very high and puts the provider itself and all the users at risk. A decentralized solution would help to distribute this responsibility:

> *Signal is trusted because it's Moxie, but if I was the government I could force Moxie to hand over the keys and tell him like… you can't say anything anymore. But he would at least shoot the system down, like the case with Lavabit, the email service that Snowden used. That guy closed the service down instead of working for the government. But most people will not do that… The way to solve is to have a federated system like you have your own data at your own location so it's not spread in the central location. So decentralization is what we need. [Peter Sunde]*

Secondly, in our interviews, centralization was also questioned in relation to the metadata leaks. Another argument in favor of decentralized architectures emphasizes the freedom for users to choose between various services, as Matrix lead developer explains:

> *I had a long running dispute with Moxie Marlinspike about [decentralization versus centralization] basically because Matrix obviously is an interoperable decentralized network and he considers that a privacy risk because you can't control the whole network, there may be implementations that may not be secure, there may be bugs that you can not control, that leak sensitive information and that is a thing for privacy. I would argue however that decentralization is also important for freedom, and the user's ability to pick which service to use. [Matrix lead developer]*

Finally, federated protocols are said to provide better accountability and better control over the service providers, as well as an ability to choose:

> *There's no way to establish any kind of accountability for centralized services that lock people in. It is very easy if people have federated services, they can just pick up and leave if their provider does something they may not like. So tomorrow Signal started to do that people were critical of, it would be very difficult to get people out of Signal to use something else. [Elijah, LEAP]*

Indeed, in our user interviews, we have witnessed a specific attitude of "proximity" with XMPP service providers among frequent XMPP users, as well as a specific form of feedback between users and providers. The possibility to develop a client on top of an open federated protocol, with features needed by a specific user community, was also underlined as a positive aspect of federation, however limiting the usage of these solutions to a more tech-savvy audience.

While centralized projects such as Telegram, Signal, or Wire prevail on the market and have larger user-base, most developers were more enthusiastic about decentralization. Even though they agree on the fact that decentralized systems are harder to design, their motivation to work on decentralized systems was grounded in both the political and technical aspects of decentralization. Politically, decentralization offers 'empowerment' to the user, as it gives users a means to 'control' their own data and developers believe it enables better metadata protection:

> *"I know there are journalist organizations that run their own XMPP server, primarily using desktop clients. I'd like to allow people to run their own infrastructure around their own data as much as possible. There are other great tools that do encryption, at that point you can use WhasApp or Facebook, they are using the same Signal protocol. But you're still not owning your data, all the metadata is controlled by a centralized system, they know all your contacts, who you're messaging at what time. I want people to run their own infrastructure. I just think that's good for… I just think decentralization is a good thing, and it's a hard problem to solve in a usable way. So… Other people feel differently. Like Moxie really wanted to do a vertically integrated centralized solution because it's a lot more usable if you control all the technology pieces". [ChatSecure]*

Some developers believed the choice of decentralization is inherently connected to not collecting metadata, and felt that models existed which were usable and decentralized: "*With Signal it's impossible to create a decentralized system because phone numbers aren't decentralized. With XMPP it's like an email address. Even users who aren't technologically savvy can understand this is my user ID, and this is my server.*" *[OMEMO lead developer]*

Developers involved into production of decentralized protocols noticed that the realities of the "market" in secure messaging make both federated and distributed projects less privileged in terms of financial investments than centralized projects:

> *I believe that the problems with decentralization we have right now are more a problem of the lack of funding than a conceptual problem that decentralization does not work. Most XMPP clients are developed by volunteers in their spare time. And it's kinda obvious that one single developer that donates two hours a week of his spare time to develop an IM client is unable to compete with five full time developers like Open Whisper Systems for example. [ChatSecure dev]*

Unlike developers, many high-risk users did not bring up the need for decentralization explicitly, but they brought it up implicitly in how they formed trust relationships. Decentralization is seen both as technical challenge and social experiment, as it provides infrastructure for specific communities to organize without having to rely on intermediaries. In this sense, developers, high-risk users, and trainers tend to build associations between political organization and technical infrastructure. For example, some developers and trainers justified decentralization as mirroring the organization of anti-authoritarian social movements. In terms of choice, there was a preference for systems that were considered trustworthy politically by high-risk users, and decentralization was generally viewed as a positive in this regard by the minority of high-risk users that wanted decentralization. These high-risk users expressed concerns about centralized systems collecting their metadata, although few realized this would be possible in most decentralized systems as well, albeit in a distributed form.

In what follows we will briefly present three federated protocols/applications : Conversations (and OMEMO protocol), Matrix.org and LEAP/Pixelated.

2.2. XMPP "revival", OMEMO protocol and other innovations in federation

2.2.1. Off the record messaging protocol and its problems

Released in 2004, "Off the Record" (OTR) messaging is a plug-in for synchronous instant messaging XMPP that features a number of radical changes in contrast to PGP[33]. On the

[33] Nikita Borisov, Ian Goldberg, and Eric Brewer (2004). Off-The-Record communication, or, why not to use PGP. In *Proceedings of the Workshop on Privacy in the Electronic Society*, pages 77–84. ACM.

level of security, different keys are generated per conversation when a conversation is started, so there is no long-term key material that is vulnerable to compromise. However, by virtue of this design choice OTR messaging limits itself to synchronous messaging between only two participants. Key management is much easier and verification of contacts is still encouraged via a shared secret established offline rather than key verification in PGP. Off the Record messaging also enables forward secrecy, i.e. that a key compromise cannot lead to the reading of past messages, by simply deriving a new key for every message in the conversation. It has undergone thorough academic analysis in terms of security, leading to newer versions of OTR being produced in response to various attacks[34]. Importantly, messages can not be repudiated, i.e. it can not be proven that a message was actually sent by the sender.

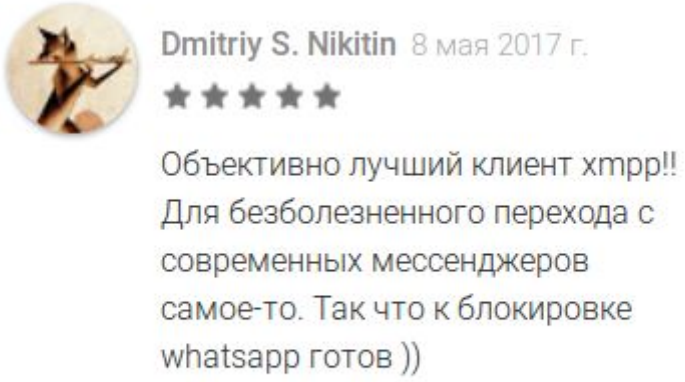
Furthermore, it could also not be proven that they were not tampered with, as malleable encryption was used. OTF was built as an extension to XMPP, an IETF standard that "provides a technology for the asynchronous, end-to-end exchange of structured data by means of direct, persistent XML streams among a distributed network of globally addressable, presence-aware clients and servers" that was mostly used for synchronous chat. Like SMTP, XMPP does not provide any content confidentiality and so does not, by itself, count as secure messaging without OTR. OTR has application support via clients such as Adium and Pidgin that can be used on a number of platforms, and could even be used on mobile platforms via Conversations and ChatSecure.

While XMPP itself is standardized by the XMPP Foundation and the various versions of OTR are authoritatively described by its academic authors in a specification on their webpage, OTR's "current usage" is itself is clearly described by the XMPP Foundation (a small standards body devoted only to XMPP) as a XEP (XMPP Extension Protocol). Therefore, although not as authoritatively standardized as PGP, XMPP+OTR is informally considered an open standard. A usability study was done on OTR that demonstrated users did not have trouble setting up keys but did not understand the offline authentication process[35]. The usability and user perception of more complex properties such as forward secrecy and repudiation were not studied in usability studies. Although academic work examined how to improve the authentication process[36], the combination of the restriction of OTR to synchronous messaging and the confusing authentication process as well as underlying dependencies on the increasingly unused XML technology stack, unmaintained insecure clients, and excessive extensibility – all led to a decline in usage of OTR in the decade after its publication.

However, our analysis of the Russian context shows that XMPP-based messaging solutions are now experiencing a rebirth, namely in response to the growing control of online communications by the state. A recent law project (25 May 2017) proposes to ban usage of

[34] Mario Di Raimondo, Rosario Gennaro, and Hugo Krawczyk (2005). Secure off-the-record messaging. In *Proceedings of the Workshop on Privacy in the Electronic Society*, pages 81–89. ACM.
[35] Ryan Stedman, Kayo Yoshida, and Ian Goldberg (2008). A user study of Off-The-Record messaging. In *Proceedings of the Symposium on Usable Privacy and Security*, pages 95–104. ACM.
[36] Chris Alexander and Ian Goldberg (2007). Improved user authentication in Off-The-Record messaging. In *Proceedings of the Workshop on Privacy in Electronic Society*, pages 41–47. ACM.

anonymous messaging by obliging instant messaging applications to guarantee user authentication via phone numbers. Several IMs have since been blocked in Russia, such as WeChat, Zello (massively used by Russian truck drivers movement), Line and Blackberry. In this context, we encountered growing interest of Russian users to XMPP/OTR as alternative to centralized IMs. Conversations, the federated encrypted IM, has an important Russian userbase. On Google Play users mention it as an alternative after popular centralized messengers are blocked:



"The best XMPP client: for a painless transition from modern messengers. I am ready for WhatsApp blockage" [comment by Dmitry S. Nikitin, May 8, 2017][37]

Russians run a considerable number of XMPP servers (15[38]). The Russian user community of Matrix.org is also constantly growing. Federation seems to find its adepts in the context of state-level censorship where centralized servers can be easily blocked. Possibility of relatively easy development of new clients compatible with older ones also makes it relevant to the context of state-controlled Internet.

2.2.2. Conversations and OMEMO: bringing future secrecy to XMPP

The authors of federated end-to-end encrypted messaging apps like Conversations (Android) and ChatSecure (iOS), both using XMPP + OTR/OMEMO, claim to develop their tools in response to an existing problem, namely the absence of a trusted and properly coded XMPP-client for mobile. Both projects were basically one-man apps, with developers working alone, driven by their own technical curiosity and their own problems as XMPP/OTR users:

> *"I am an XMPP user. When I got my first Android phone there was no good examples available for the platform. I actually decided to write my own app. I'm working pretty much alone. If you exclude the OMEMO that was written during the Google summer*

[37] https://play.google.com/store/apps/details?id=eu.siacs.conversations&referrer=utm_source%3Domemo

[38] https://list.jabber.at/

> *of code in 2015 I did maybe 95% of the code by myself. I do have occasional in contributors on github but was pretty much alone". [Conversations and OMEMO developer]*
>
> "*I started working on [ChatSecure] in 2011, I was learning iPhone programming at the time, it was back when there were not quite as many apps on the AppStore. I was a user of Adium, the Mac messenger, and I was familiar with its OTR integration, and I wondered: 'Oh, what if I could compile… cross-compile OTR and have some equivalent app for iPhone'. And it took me a couple of months to get a basic prototype, I shipped it to the AppStore, and it was found by people from the Guardian project, and they funded my work to improve the product. Then I approached other funders and got other sources of funds… And was growing it from there"*. [ChatSecure lead developer]

Conversations is an end-to-end encrypted instant messaging application that proposes users a choice of three encryption algorithms and an unencrypted (TLS-only) mode. This design solution, where the protocol is transparent for users, and these can choose their encryption method, is unique in the field and makes Conversations an interesting case-study. The choice between 3 different algorithms was, as Conversations developer explained us, his version of a solution to the interoperability problem:

> *XMPP is now up for I think 18 years. Overtime there have been many attempts to introduce end-to-end encryption into XMPP. One of the earliest attempts was OpenPGP based. I think it's 10 years now that OTR came out, OTR being transport agnostic that means that you can speak OTR over any instant messaging protocol. When I start at Conversations these end-to-end encryption methods were available in different clients. Some clients had only Openpgp others had only OTR. Conversations tried to be compatible with all those clients, so conversations had to implement openpgp as well as OTR. [Daniel, OMEMO/Conversations developer]*

XMPP offers more freedom (one can choose an encryption protocol to build on top of it), while adding complexity as it demands additional adjustments in order to make "translation" happen (bridging or solving the interoperability problem).

The OMEMO protocol (OMEMO is shorthand for OMEMO Multi-End Message and Object Encryption) provides future and forward secrecy and deniability, and gives the possibility of message synchronization and offline delivery (important features for a federated protocol). OMEMO was developed to solve specific limitations and problems that existed both in OpenPGP and in OTR:

> *"One of the main limitations of OTR was that it didn't work with multiple clients. If you are logged in your mobile client as well as your desktop client OTR does not work. Open PGP the other hand had downsides as well: there wasn't any verification built in, it did not have any forward secrecy or any other nice crypto traits that you wanted to have" [Conversations lead developer]*

Conversations nowadays supports all of listed above encryption protocols. This design choice is based on user behavior and habits:

> *"We still have a big user base using OTR or OpenPGP [...] There are still XMPP clients around especially on desktop which only support OTR for example. And in my user base there still are people who want to communicate with those who are using desktop clients which only support OTR. [...] We couldn't throw these encryption schemes away so that's how we end up with providing users with a choice from three different end-to-end encryption or if we include no end to end encryption it means 4 schemes". [Conversations lead developer]*

Conversations is giving users a good amount of freedom in choosing the encryption scheme that they prefer or are more used to. This also helps to attract more users, regardless of the encryption algorithms that their peers support: Conversations can talk to OTR- and to OpenPGP-only clients:

> *I give user a choice. They can choose Open PGP and will not have forward secrecy but will have some other features in return. [Conversations lead developer]*

This freedom of choice is, as we have demonstrated in the section above, mentioned as one of the advantages of federation. This also gives users possibility to gain certain security, privacy or usability features that exist in some protocols and do not exist in others. For example, OpenPGP offers archiving, which, for low-risk users, may sometimes be more important than future and forward secrecy:

> *If you look at people who are only using Conversations, you could say it's a minority of users but there are a couple of users who don't want to move away from OpenPGP because all these modern encryption schemes that provide forward secrecy do not have the ability to retrieve your messages. so when you're setting up a new client you can't access the archive with all your messages. And OpenPGP allows you to do that. That's arguably a very niche use case, less secure in some ways because it doesn't have forward secrecy and stuff like that but some users really prefer have the ability to access the archive they are not paranoid that much or are not under the high risk. [Conversations lead developer]*

As the app's developer explains it, Conversations is aimed at a specific user-group, that could be defined as high-knowledge, low-risk and privacy-aware users, who strongly support XMPP, federation and open-source, opt for Google-free and privacy-preserving (minimum metadata storage) solutions. While applications as Signal, Telegram, Wire are often covered in media, in case of Conversations, users' enrollment happens via their technical knowledge and interest in XMPP and federation. These users are, in their turn, bringing their close network and peer groups into using the app:

> *The majority of my users are either people who are more educated about how security works and how federation works or close family members or friends of these people. [...] Conversations is not like the mainstream app you read about on the media. It's rather the app that requires some knowledge about different protocols and then you particularly pick federated protocol like xmpp and then do search what clients are available and then choose conversations for example. And then maybe you'll convince a couple of close friends to use that as well. [Conversations lead developer]*

OMEMO implements double ratchet, which is starting to be considered as a new “norm”, a “golden standard” of instant messaging promoted by the widespread acceptance of the Signal protocol:

> *“I personally don’t have a very strong cryptography background, I try to rely on what’s already there, what’s already popular right now. Like Signal protocol with double ratchet is quite widespread these days and seems to be well received by various cryptographers. So it makes sense to base OMEMO on something that’s similar to the double ratchet or Signal protocol.” [Conversations lead developer]*

OMEMO was developed during the Google Summer of code in 2015, using the LibSignal library. OMEMO lead developer states that the lack of standardization and lack of specification of Signal protocol was never a problem for them to work with the LibSignal library:

> *“If you are interested in working code, you’re not really concerned by the underlying crypto protocol. If you can basically use the library for the input and output, it does not really matter what happens in the library. OMEMO was basically just an XMPP wrap around this library, in the times when OMEMO was developed, 2 years ago. The underlying cryptography was not very well documented, and we were just basically making through the library. But it has changed may be about 6 months ago, OWS provided documentation about what they call the Double ratchet algorithm and how the Signal library works and how the cryptography in there works” [Conversations lead developer]*

OMEMO, being an “XMPP wrap around LibSignal”, brings Signal’s features such as future secrecy to XMPP. In this sense, protocol design choices of Conversations and OMEMO are determined by the overall development and “state of the art” of modern cryptography:

> *“If you are designing a new protocol for end-to-end encryption now or even 2 years ago, for instant messaging having forward secrecy in it it's just a good practice. It's just what all the other IM encryption schemes are doing as well. Signal does it, WhatsApp does it. At least to me instant messaging is kind of like temporary information and at least I don't think that a lot of users need access to archives and the forward secrecy is adding more security here. They don't miss the lack of archive.” [Conversations lead developer]*

However, Conversations does not support the function that leads to the disappearing of messages, even though users, according to our interviews, have been asking for it. Instead of that, Conversations opts again for giving users more choice and more responsibility:

> *“If you are looking into data hygiene, not keeping around the history is something that you should do for yourself on your own device according your own rules and Conversations has actually an option, a setting that allows you to delete all messages both sent and received on your device and regulate intervals for example a month, a week or something like that. If you are interested in data hygiene, it is a more honest solution. Because you can control when these messages are deleted. Conversations gives the user choice and an option to delete all messages on the*

> *current device after a configurable interval. It is up to each individual user to decide what to do with the data." [Conversations lead developer]*

This approach of "giving user a choice" actually differs from the automatic/opportunistic encryption approach that "hides" a lot of options from users, while some users have expressed a desire to have more control over the application and to "see" encryption happening (for example, during our observations of Pixelated usability tests at CCC in December 2016). Even though in Signal, disappearing messages are also including time choice, the encryption protocol is embedded in the tool and is not giving any alternatives to the user.

OMEMO is not yet a standardized protocol, though it has already been adopted by a number of XMPP clients (listed on a dedicated website[39]). At the time of writing (as on June 13, 2017) the website lists 38 XMPP clients supporting OMEMO. OMEMO has been submitted to XMPP foundation for standardization and is currently on the first stage of the standardization process.

However, our interviews with both Conversations and ChatSecure have revealed that these projects, though well known within the XMPP scene (developers mention that they regularly attend XMPP-related events), are almost invisible within more hybrid events destined to privacy advocates and high-risk users and other stakeholders (Internet governance organizations, lawyers, researchers), such as, for instance, Internet Freedom Festival or RightsCon. These applications could have had a wider adoption, especially given the strong modern-state cryptography implemented in both tools, however, the authors stay disconnected from the "high-risk low-tech" communities. The media coverage is also partly responsible for that disconnect: both applications are much less visible in non tech-savvy oriented media, do not benefit of a support from well-known public figures such as Edward Snowden, or organizations such as EFF, AccessNow or Tactical Tech. The latter are more eager to cover centralized applications (except for Tor browser) and are not really involved into advocating for technical decentralization.

2.2.3. Matrix.org: facing the interoperability problem

During European Lab session on privacy-enhancing technologies and decentralized identity management systems (Lyon, France, May 25, 2017) we have attended an interesting discussion that illustrated the state of the art in the field of encrypted messaging quite well. The panel moderator, a PGP, Signal, Wire, Jabber (etc.) user himself, asked the audience: "How many messaging applications do you use on your mobile phone?" After a brief round of answers from the public, he counted that among the 40 people present, the average number of messaging applications was 5 per person, while several people said to regularly use more than 7 messaging applications. This question served him to bring on the table an important problem of silos: how do people communicate being separated by the walled gardens of their favorite messaging clients, but also, more deeply, by various encryption protocols that are incompatible with each other.

[39] https://omemo.top/

While Conversations suggest a solution of interoperability between encryption protocols (OpenPGP/OTR/OMEMO), it still runs over XMPP and requires users to have an account on an XMPP server (or to "even better - run [their] own XMPP server for [them] and [their] friends"[40]). However, interoperability remains a problem for users of centralized messaging applications. Individual projects have attempted to solve the interoperability problem, such as bridging Jabber to centralized applications. For example, the Russian Pirate Party has developed a "Jabber bot" for Telegram, that helps Pirate Party members that do not trust Telegram to still communicate with their community using Jabber. The actual bot is called Jabbergram[41] and the one used before is Goatway[42].

Another project that addresses the interoperability problem is Matrix.org, that gives the possibility to "bridge" various messaging applications and let users connect without having to migrate from their favorite tools. The idea of Matrix.org goes beyond an instant messaging application, but is framed as an ecosystem that could be used for any kind of data sharing and is first of all trying to solve the interoperability problem:

> *"Matrix itself is I guess an ambitious project to create a new ecosystem for interoperable communication really extending the web or the Internet itself, so that there is a common fabric for sending messages and VoIP calls, and indeed sharing information of any kind so it could be Internet of things data, it could be … I don't know … literally any kind of communication. Pretty much like you can store anything in the web, you can store anything in Matrix [...] We build Matrix to provide a consistent real-time interface to be able to publish and consume real time data. And the model is very much inspired by the phone network, in terms of the phone network being a global way to exchange communication data. But in the end it's closed, it's not open, it's quite centralised to the telcos. So we wanted to create a complementary network which has the openness and the decentralization baked in". [Matrix lead developer]*

Matrix is an open source, non-profit project under Apache license, financed by Amdocs on the long term. The team currently counts 16 members, with 11 based in London and 5 based in France. They support Android, iOS and Desktop versions. Currently there are 5000+ chat rooms registered on Matrix focused mostly on "the techno privacy aware community, computer scientists" and "normal developers", as Matrix creator puts it.

Unlike LEAP or Signal, the Matrix team does not take a political stand and does not aim to provide software for left-wing political activists. The creator of Matrix positions his team as "sort of moderate, really centrist":

> *"I am very sympathetic to the human rights issue and privacy but we haven't set up with Matrix to build something like Tor or Signal or Ricochet or any of those tools which are optimized very specifically for privacy at all costs. Instead we're trying to do*

---

40 https://play.google.com/store/apps/details?id=eu.siacs.conversations&referrer=utm_source%3Domemo

41 https://pypi.python.org/pypi/jabbergram/0.1.6

42 https://github.com/hdghg/goatway

> *the best we can and be very mindful of it, whilst also building an ecosystem that works and is practical". [Matrix lead developer]*

Matrix's creator identifies his position as "liberal pluralism", which is reflected in the very architecture and the users of his system. From the point of view of the architecture, it is a decentralized system of "translation" that "bridges" a great variety of different messaging tools, thus leaving a certain amount of freedom to the users that can stay with their usual interface, while making it possible for them to connect with others. As Matrix's lead developer points it, Matrix is trying to respond to the problem of "fragmentation" and the "walled gardens" problem produced by the quick and somewhat "chaotic development of the messaging ecosystem", especially on mobile. During our interview, he showed us his mobile phone with around 10 different instant messaging applications (from Wire, Threema, WhatsApp and iMessage to Signal).

In terms of user pluralism, Matrix has very different rooms, from cryptography and open-source, cryptocurrency and decentralization to psychological help, furries, subcultures and fan communities, left-wing groups and alt-right Donald Trump supporter rooms. One of the problems that Matrix tries to address is spam management and decentralized reputation system:

> *It's an open area of research, there isn't a solution yet [...] We do not want to have a single silo that declares who's good and bad. And we also need it to be morally relative because just because I think someone is a spammer and he should be ignored, for you it may be a valuable direct marketer and you're actually interested in whatever product they're happening to sell. It needs to be morally relative or needs to support liberal pluralism. I think that's the correct jargon. [Matrix lead developer]*

Within the ecosystem of mail and messaging applications, Matrix puts itself "in the middle" between centralized and completely decentralized systems:

> *"Signal is very much saying like... we'll run it as a centralized service therefore we can guarantee its security. We will be more in the middle. And I guess something like email in the current state has no privacy and security but it's being 100% decentralized. So we're basically more middle of the ground approach".*

Matrix's lead developer discusses relations between decentralization and security, saying that "the two things [decentralization and privacy] definitely pull against each other". He sees email as an unsecure system, because of the independence of email service providers. In this sense, an attempt to introduce a standardized automatic mail encryption such as Autocrypt, that could be implemented by various Mail User Agents with relatively low resources, could be a possible step against the passive attacks described by the Matrix developer. It is not by chance that Matrix was invited to present its current work at the Autocrypt hackathon in December 2016, gathering different projects that discussed not only email encryption but also the decentralized reputation and identity systems, such as Scuttlebutt. Introducing encryption through headers also becomes a form of 'governance' via federation, between the multitude of mail service providers: this kind of governance is, unlike

Signal, based on a very detailed specification, which could be approved by the galaxy of email service providers who take part in Autocrypt effort.

For Matrix as a decentralized system, the implementation of end-to-end encryption was a crucial step to undertake, because of the passive attacks and a specific relations between trust and infrastructure where trust is spread across a large number of server administrators:

> *In decentralized systems you end up with a large attack surface. If you were running a server and I am running a server, and I send a message, the message is on both of our servers that means that we have to trust a sysadmins of both of the servers. In a big room with 5000 people, and, say, 1000 servers, with average of 5 people per servers, you have even a bigger problem. If you're sending something sensitive or private, as much as you can send something private to 5000 people, it will be shared across all of the servers. So we found critical to get end-to-end encryption so that the servers get an encrypted copy of the message rather than the real message so that you don't need to worry about system administrators spying on messages. [Matrix lead developer]*

In the case of Matrix, end-to-end encryption was adopted two years after the beginning of the project -- while, as we have seen from our research, when a system has not been designed with end-to-end encryption from the very beginning, the transition is rather slow and difficult. In the case of Wire, originally built with no end-to-end encryption, the team had to remove part of the server-side code that still contained possibility of a plain text message being sent before going fully open-source. In the case of a decentralized system, the problem of implementation is even harder, as some of the older clients run with no encryption and additional steps must be taken in order not to block users who use older clients:

> *We're obviously being very careful when rolling end-to-end encryption out because we don't want to break everyone's existing communications and clients. Plus Matrix as an ecosystem has many different apps, probably 35-40 different clients and many different bridges, not all of them have turned to end-to-end yet. We don't want to block the old clients. So by now it's an opt-in on a room basis. But once we're out of beta we'll be turning e2e on by default by every private room, and we'll have a proxy migration path for other clients so that simple clients that know nothing about cryptography can easily join and participants. [Matrix lead developer]*

The protocol implemented for end-to-end encryption in Matrix is called Olm[43] and is a version of Signal double-ratchet written in C and C++11 and exposed as a C API. This library also includes an implementation of the Megolm cryptographic ratchet. It has been audited in 2016 [44]. Olm version of Double ratchet has been designed for a decentralized system, so that it can be easily ported to different platforms and bindings can easily be written for it. It does not include certain elements baked in such as random numbers generator. Instead the caller must provide the random data. This makes it easier to port the library to different platforms since the caller can use whatever cryptographic random number generator their platform provides.

[43] https://matrix.org/git/olm/about/
[44] https://www.nccgroup.trust/us/our-research/matrix-olm-cryptographic-review/

Olm stands for a troglodyte salamander, that resembles an Axolotl. Olm's second name is Proteus. The three names of salamanders are used by three applications that have implemented double ratchet protocol: Signal (ex-Axolotl), Wire (Proteus) and Olm (Matrix).

Olm

While Wire had difficulties in implementing Axolotl, Matrix did not encounter hostility or tensions from the Signal team, except for an explicit condition to avoid using names or references to Axolotl or Signal. Matrix developer explains this relatively successful collaboration with Signal by the political economy of open source and by the very architecture and nature of Matrix (everyone can be "bridged" there, and a large number of other projects can benefit from it):

> *The difference between us and Wire is that Matrix.org initiative is non-profit and entirely open source. The Apache license that we use is aggressively permissive [...] And I spoke to Moxie and explained what we were doing. And he said: you're crazy, and if it works than go for it, it will be great, but you guys have fun doing that and I will keep doing Signal. He does not want to persecute us. [...] Whereas other companies, which are doing proprietary commercial messaging solutions even if some parts of it are open-source, he [Moxie] will see them as competition. We are more like idealistic altruistic hippies who should probably fail while building this white elephant of Matrix, and if we don't it will be a good thing for everybody including him [Moxie]. So why not support us? [Matrix lead developer]*

However, XMPP community is undermining Matrix.org's "innovative" aspect. The most popular critique to Matrix is that they "reinvent XMPP":

> *I do think they reinvent the wheel, XMPP has been around for 15 years and Matrix to my view is not that different. It does not provide you with something that an XMPP ID could not provide. So I don't have any plans of being compatible with them. Both Matrix and XMPP have the concept of transports built in though, so we could have an XMPP server bridged into the Matrix network or vice versa but that's not something I*

am interested in personally. Someone else could do this if he was interested in this. [Conversations lead developer]

2.3. Towards political and technical federation of email encryption: Autocrypt, LEAP and Pixelated

SMTP, the protocol originally used for transferring email, is one of the first messaging standards, but SMTP has no confidentiality of content or even authentication of headers for network-level routing. However, it is one of the oldest and most widely deployed standards for asynchronous messaging[45]. PGP (Pretty Good Privacy) was created to add end-to-end encryption capabilities to e-mail in 1991 by Phil Zimmerman, while the OpenPGP set of standards was finally defined years later in 1997 in IETF to allow the open implementation of PGP without conflicts with RSA patents or proprietary software[46].

OpenPGP is implemented in both desktop and mobile email apps, including Outlook, Apple Mail, and Thunderbird through plug-ins. An alternative standard for encrypted email called S/MIME is also supported via plugins by most major email clients, the main difference between OpenPGP and S/MIME being that S/MIME requires the installation of certificates provisioned by centralized certificate authorities[47]. In contrast to centralized approaches, OpenPGP offloads the key management to the users via a decentralized “Web of Trust” model. In general, PGP was considered to have poor usability as users could not understand key management and judge the trust relationships in keys, or even understand the interface [48].

OpenPGP and S/MIME also work on mobile devices, such as the PGPMail for iOS and K-9 Mail (via plug-ins such as Openkeychain) for Android, but as OpenPGP binds the key to the particular device, there has often been concern about how to securely transport any long-term private key material between devices, and so mobile adoption of encrypted email is considered to be low among users and problematic in terms of security. Although these challenges of PGP on the mobile platform are well-known[49], mobile PGP has not been subject to usability studies in the same manner that PGP itself has. S/MIME has had some usability studies and in general shows better usability than PGP, insofar as key management does not have to be maintained by the end-user, but users still have trouble understanding the interface[50].

[45] https://tools.ietf.org/html/rfc821
[46] https://tools.ietf.org/html/rfc2440
[47] https://tools.ietf.org/html/rfc2633
[48] Alma Whitten and J Doug Tygar. Why Johnny Can’t Encrypt: A Usability Evaluation of PGP 5.0. In Usenix Security, 1999.
[49] Audun Jøsang and Gunnar Sanderud. Security in mobile communications: Challenges and opportunities. In Proceedings of the Australasian Information Security Workshop, pages 43–48. Australian Computer Society, Inc., 2003.
[50] Simson L Garfinkel and Robert C Miller. Johnny 2: a user test of key continuity management with S/MIME and Outlook express. In Proceedings of the Symposium on Usable Privacy and Security, pages 13–24. ACM, 2005.

In terms of the underlying protocol, there are a number of flaws. First, PGP tends to allow all combinations of usages of encryption and signatures based on the preference of the user, but does not offer authentication of the headers (i.e. the "to" and "from" fields), allowing messages to be surreptitiously forwarded and otherwise redirected via signature stripping attacks[51]. Despite these problems being well-known, the IETF OpenPGP Working Group did not address any of these concerns, and so far has only re-convened in order to address upgrades in the underlying primitives in order to support elliptic curve cryptography and remove known-broken hash functions in fingerprint verification from the standard[52]. In general, PGP has been considered an open standard that has serious problems in terms of both security and usability, and this provoked the generation of competing technology such as Off the Record Messaging.[53]

2.3.1. Autocrypt: email encryption as a community effort

E-mail stubbornly remains unencrypted, due, to a large extent, to problems with key management. Proprietary closed-source projects exist that offer "encrypted email solutions", such as Protonmail or Tutanota. However, they are not solving the interoperability and fragmentation problem. End-to-end encryption in Protonmail, for instance, works only if both users have Protonmail installed; this reminds of how centralized IM clients work, that also bind users to specific applications. Our surveys have shown that users do not always know about this particularity of Protonmail and think that all Protonmail emails are encrypted by default.

However, email "remains the largest open federated identity and messaging eco-system, anchors the web, mobiles and continues to relay sensitive information between people and organisations"[54]. In the context of centralization in the field of IM apps, with telephone numbers being massively used as unique identifiers, several initiatives have recently been launched to "make e-mail great again"[55] and to revive encrypted e-mail, such as Autocrypt[56], pEp, the Google End-to-End project[57] and LEAP/Pixelated[58]. These efforts have not yet been finalized or have not reached widespread adoption; this process "in-the-making" should be analyzed, from an STS perspective, as a technosocial, community-building effort that may have important consequences on the whole encrypted messaging/mail ecosystem (suggesting a "(re)turn to federation"; proposing alternative approaches to identity and key management).

[51] Don Davis. Defective Sign & Encrypt in S/MIME, PKCS# 7, MOSS, PEM, PGP, and XML. In USENIX Annual Technical Conference, pages 65–78, 2001.
[52] https://datatracker.ietf.org/doc/charter-ietf-openpgp/
[53] Nikita Borisov, Ian Goldberg, and Eric Brewer. Off-The-Record communication, or, why not to use PGP. In Proceedings of the Workshop on Privacy in the Electronic Society, pages 77–84. ACM, 2004.
[54] https://autocrypt.readthedocs.io/en/latest/
[55] Source: Autocrypt chat on IRC.
[56] https://autocrypt.org/en/latest/
[57] https://github.com/e2email-org/e2email
[58] https://leap.se

Because of its decentralized architecture, automatic email encryption implementation demands a strong community effort and enrolment of a multitude of actors (namely, Mail User Agents). These actors need to come to an agreement on the protocol, its documentation, modalities of implementation, but also UI/UX features, logo, funding sources and so on.

Unlike centralized mobile applications for instant messaging, that can impose their own protocols, often unstandardized, and "own" their users who interact "within" the IM app, federated email ecosystem remains, to a large extent, open. It blurs borders between users and service providers, as the barrier of running a privately-owned email service has become lower, involving more people in the ecosystem. Thus, there is fundamentally no way (nor will) to control all of the MUAs and service providers at both the technical and the ideological level. However, there are several ongoing attempts to create a "translation" tool that can connect different actors without centralizing the whole ecosystem.

One of the most recent innovations in this field is Autocrypt, a project run by a group of email program developers and crypto enthusiasts from the open-source privacy-aware community. Autocrypt provides an answer to the problem of key discovery, exchange and key management, thus addressing two important challenges, firstly, from the point of view of usability and secondly, in the context of a movement towards "re-decentralization". Instead of keeping public keys on a centralized public key server (that may create vulnerability), and proposing users to retrieve, exchange and verify keys by themselves (that may create confusion) Autocrypt puts key material in the header of the email. As LEAP creator explains it: "*Autocrypt is no validation but it's trying to standardize the initial key discovery using the TOFU model which is the trust on first use". [Elijah, LEAP founder]*

Usability-wise, this automatic key discovery process helps to solve important problems related to the key exchange, verification and management. Our user survey has shown that users rarely verify keys and rarely can define what a key is. High-risk users tend to verify the authenticity of their contacts when they receive notifications on an IM app about the changes in the key material, using "real-life" non-cryptographic solutions such as voice calls or social networks. However, this behavior is rare. And when it comes to email, advanced key verification rarely happens. Moreover, as our interviews with trainers show, there is a general trend in the trainers community to avoid explaining keys during informational security seminars, as "the old metaphors of padlocks are not really explaining what's going on there, and we waste a lot of time on it", - a French trainer from the hackerspace Le ReSet explains. The general trend to "automatic" or opportunistic encryption makes key discovery process invisible to the user, and this is seen to be a positive step towards mass adoption of encryption.

> *"Different encrypted messaging apps have popped up that made it a lot easier to have just an app that will pass on your communication, and the encryption part will be transparent to the user. Having encryption as a default mode is the key part in making encryption popular. Transparent means a user does not have to learn about all the maths beyond the crypto" [McLemon, cryptotrainer, Austria]*

Even though there are some demands of better visibility of encryption (people want to “see encryption happening”). Autocrypt sees that as an opportunity to implement a “pedagogical” aspect within the UI: this mode would let users “see what your service provider sees”. By now this mode has not yet been adopted.

From the point of view of re-decentralization, Autocrypt solution addresses the problem of centralized public key storage, especially when coupled with other projects, such as ClaimChain, for key validation. It also proposes a new form of coordination or federation of email app developers community, which, as Autocrypt people witness, has been for a long time rather dispersed and has not really had a place to “meet each other”). In this sense, headers become instruments of “self-governance”, communication and coordination of MUAs, that leave a lot of freedom to MUAs while providing a standard that helps to keep encryption working across different agents.

For automatic email encryption these two challenges (usability and federation of MUAs) are interconnected. In order for automatic encryption to work, the solution for it should be scalable, which implies collaboration between the multitude of third-party projects. It relies on the success of enrolment of email app developers.

Both LEAP and Autocrypt address not only the "end user" but in order to work and to be spread widely, relies on joint efforts of professional communities (service providers in one case; mail client developers in the other). This community-based approach is very different from the one offered by Signal-like centralized applications where the authors of the protocol seem to have greater control over the implementations, and various implementations are not offering interoperability.

However, while both Autocrypt and LEAP demonstrate a comparable dynamics in terms federated, community-oriented efforts, their approaches to the key transmission and, by consequences, the ways how their communities are organized, are very different. Autocrypt opts for a key-serverless key transmission, or in band approach, “putting all logic for encryption into the end-user mail program code” while LEAP adopts the “provider-mediated key approach”, and thus needs to rely on the service providers support:

> *The goal is that you should never have to do anything. If there’s a way to automatically discover and verify your key, we wanna be able to always discover and verify your key ASAP and automatically. If there’s not a way to verify it, we wanna discover it as soon as possible and then continue to use that. So… what we don’t wanna do is to teach people about what keys are, what fingerprints are. We want most people to be able to use it and have strong guarantees without having to know anything about that. In order to be able to do it at scale we will need to have more service providers to support methods of automatic validation. [LEAP founder]*

In this case, email apps then depend on their providers. Autocrypt has made a long way, starting from the comparable provider-mediated key approach, to the actual key-serverless key distribution. The bifurcation started at the PGP summit in July 2016, where the project (not yet called Autocrypt at that time) was discussed with email encryption community,

including such projects as K-9 and OpenKeyChain. Important vulnerabilities of public key servers were discussed, namely the privacy-related issues: “the decade old PGP keyserver model along with cryptoparty practises have known issues, notably leaking the social graph of users and all the difficulties arising from involving users in key management decisions (import, transfer, export, trust)”. [See D 5.2 section 2.1]

However, the provider-mediated approach, that could be an alternative to a centralized public key server, also demonstrated important weak points. First of all, this method of key distribution adds complexity in the already complex infrastructure by introducing a new dependency. Secondly, it involves high responsibility (and thus, power) of mail service providers, that can become yet another point for hacking attacks: “One particular problem with provider mediated schemes is that they not only require support from mail programs but from providers which need to serve encryption keys. Serving encryption keys turns providers into a kind of “certificate authority”, which in turn makes them more interesting hacking targets”. [See D 5.2.section 2.3]

By now, Autocrypt has been supported by an important number of email app developers, and since June 2017 Autocrypt specification has been adopted by a new project, Delta Chat, that brings Autocrypt encryption scheme into the field of instant messaging. This may in the nearest future provide interoperability between email and the new instant messenger.

By now, Autocrypt is focused on passive attacks only and as such is targeted more at a low-risk audience:

> *For most people it’s sufficient to trust the first time you’ve encountered a key and keep trusting or have a trust chain after initial contact so that… When someone comes later and tries to disrupt or MITM your communication they are not able to, because they can not roll back time. For most situations it’s OK and it’s much easier to do. That’s why there is much excitement about doing that first, it’s way easier and it covers most of the basics. But […] there are many many people who have a different threat model where they are actually under active attack and we do need proper strong validation for those people. [LEAP founder]*

Standardizing TOFU and making automatic mail encryption easier to deploy for MUAs is thus a first step to make, before providing more sophisticated solutions against active attacks and targeted surveillance.

Even though in the current level 1 Autocrypt is focusing on low-risk users, the Autocrypt motto “email encryption for everyone” has a benefit for high-risk users, for the reasons that we have discussed in the part 1.3. of this deliverable: mass adoption of email encryption makes both mass surveillance and targeted surveillance more expensive and harder.

2.3.2. LEAP/Pixelated

While Autocrypt engages email app developers and opts for an in-band approach, LEAP is addressing primarily email service providers. The server-side part of LEAP, the LEAP

Platform, is also called “provider in a box”[59] and is a “set of complementary packages and server recipes automated to lower the barriers of entry for aspiring secure service providers” [60]. The client side part is called BitMask, a cross-platform application including a local proxy that a standard email client can connect to, and an easy one-click Virtual Private Network (VPN) service. Bitmask app offers full end-to-end encryption, while public keys are automatically discovered and validated. The LEAP project was developed in response to both the crisis of email encryption infrastructure, and usability issues that make it difficult for users to use encryption without compromising their confidentiality and the confidentiality of the people they communicate with.

Behind the LEAP project, a certain political vision has been developed, and initially, members of LEAP have been for a long time involved in what they called “radical tech *collectifs*” during our interviews. This intention is highlighted on the official website of the project. However, if LEAP’s initial target was related to high-risk users, the scope of the project englobes a wider audience of Internet users audience, as LEAP people define encryption as a human right: “Like free speech, the right to whisper is a necessary precondition for a free society. Without it, civil society and political freedom become impossible. As the importance of digital communication for civic participation increases, so does the importance of the ability to digitally whisper. LEAP is devoted to making the ability to whisper available to all internet users”[61].

LEAP’s architecture and protocol design solutions have been forged through discussions with other tech collectifs, questioning the ‘coherence’ between certain types of politics and certain architectural solutions, namely, anti-authoritarian leftist politics and peer-to-peer models. LEAP develops its own philosophy of decentralization, that is far from being an apology of radical decentralization or peer-to peer:

> *“There are three models.. three architectural models: centralized model, federated model and a peer-to-peer model. And LEAP came out of a shared… understanding… [...] about the way in which people with certain type of politics, with anti-authoritarian politics, they bind their politics to a decentralized model and they believe very strongly that all of the technology must follow a decentralized model. And our critique of that was that there are a lot of technical problems with decentralized model, and that you can’t actually… the politics and the tech architecture… actually there are some correlation but trying to [correlate them directly] does not work at all. So for all these reasons and many more we were upset with how people from anti-authoritarian politics were mapping this directly to a decentralized architecture which we felt had potential but there are a lot of hard research problems that are unsolved”. [Elijah, LEAP founder]*

In response to problems found in both p2p and centralized models, LEAP proposes to initiate a transition of the whole modern encryption ecosystem towards open federated protocols that can also become, according to LEAP founder, an important turn in Internet governance, as it redistributes power-relations and re-decentralizes infrastructures, nowadays owned by a minority of big actors:

[59] https://leap.se/en/about-us
[60] https://leap.se/en/about-us
[61] https://leap.se/en/about-us

> *"We felt that we needed a revival of 1990s. 1990 s were like before the craziness of the dotcoms, everything if you wanted to communicate you had to use an open protocol that was federated, that was the way everything worked. [...] We felt that… the time was right to take some of the new innovations in the last 20 years and start to turn those into open protocols that can be federated and do not lock people… do not chain them to the monopoly powers of the internet, Google, Amazon, Microsoft and Apple…" [Elijah, LEAP founder]*

In this sense, LEAP refers to a specific vision of the history of Internet(s)[62], that postulates the existence of a "golden age" of decentralized open protocols. Among Russian tech and privacy aware user communities, the same kind of "nostalgic" turn to federation has been observed: from the growing interest in Matrix.org to the revival of such formats as "XMPP microblogging" as alternatives to Twitter[63]. However, the early-stage federated protocols were accessible to specific communities of tech-savvy users, and nowadays, as ChatSecure developer notes it, setting up and maintaining federated infrastructures is far from being easy:

> *"I'd say the most difficult thing right now is to make it easier for people to run their own infrastructure." [ChatSecure]*

In this context, one of the main goals of LEAP is to spread federation by deploying "kits" of interlaced protocols, sets of packages necessary for a quick deployment of a secure infrastructure:

> *"Specifically we try to make it really easy to have a federated model [...] One reason that holds back the federated model is that properly hosting secure services on the web now is very very difficult. Beyond the reach of people who do not specialize in keeping their servers secure. So we wanna be able to encapsulate all the skills and best practices for maintaining an infrastructure into an automated suite that allows people with moderate skill without infinite knowledge to be able to do it properly". [Elijah, LEAP]*

LEAP proposes several original and new protocols, such as SOLEDAD and BONIFIED[64]. SOLEDAD is a

> *"protocol that does synchronized client-encrypted databases, [...] searchable databases that are encrypted on client device but synchronize with a cloud and with all your devices. SOLEDAD is actually the secret sauce that makes LEAP possible, it's the thing we're excited about but it's also abstract. You could build lots of other*

[62] Minar, N. and Hedlund, M. (2001). A network of peers – Peer-to-peer models through the history of the Internet. In A. Oram (Ed.), Peer-to-peer: Harnessing the Power of Disruptive Technologies, 9-20. Sebastopol, CA: O'Reilly.

[63] One of the most popular is BNWach https://bnw.im/

[64] "We have defined [BONIFIED] to standardize solving problems of the client interaction with the service provider: registering users, changing passwords, logging in, authenticating, getting session tokens" [Elijah]

> *applications on top of it, everything you could think about. [...]We could have a password manager on top of it, or secure note taking app where your notes you have access to on an encrypted device". [Elijah, LEAP]*

However, the biggest part of LEAP is built on existing open federated protocols; LEAP's contribution is the combination of these protocols and their "translation" for easier deployment:

> *"We are mostly building upon open protocols and trying to use the best aspects of those, that are properly deployed [...] There are lots of new standard protocols that can dramatically increase the security stands of service providers but they are very difficult to get ready. And that's the dream or the hope or may be the secret sauce of LEAP project is to offer a whole bunch of best practices that are really hard to set up on your own. But they are very useful if you do it". [Elijah, LEAP]*

LEAP is also interacting with other solutions such as mixnets[65], in order to solve important hard research problems such as keeping keys updates and reducing metadata leaks.

> *Initial key discovery is only half of the problem. The harder half is your keys up to date. You have to be constantly refreshing them and that potentially leaking a lot of information. That's particularly where the metadata leaking becomes important. So that's… a separate project called Panoramix, LEAP also works on it, is to do… build a mixnet infrastructure for different purposes – e-voting and email. You can achieve provable anonymity and 'un-map-ability' of the network. You can guarantee that the network observer can not identify who is communicating with whom [Elijah, LEAP].*

As Conversations, LEAP is "backwards compatible", supporting older protocols and older clients without radical transition towards post-PGP protocols. LEAP is open to support different encryption protocols, however by now the only supported are OpenPGP and S/MIME. By consequence, LEAP-based solutions also lack important security properties, such as forward secrecy. However, this is not inherent to LEAP per se but is more of a problem of the actual state of email encryption ecosystem:

> *OpenPGP has this problem [...] it combines message authenticity with non-repudiation. So people are unhappy about that. It lacks forward secrecy. If anybody ever intercepts the message if they eventually get your key in the future they can read all your messages. But this is baked in the OpenPGP as a protocol itself. And there's nothing in what LEAP's doing that requires us to use OpenPGP. We could discover some new encryption method that are based more like on Axolotl method, that supports forward secrecy, and upgrade that so that both parties supported it and not require user intervention to have to enable everything or worry about everything. So the idea is that once we gonna have a system that's automatic, we can automatically ramp up to things better that what we currently support. [Elijah, LEAP]*

[65] "Mix networks [or mixnets] are protocols that create hard-to-trace communications by using a chain of proxy servers known as *mixes* which take in messages from multiple senders, shuffle them, and send them back out in random order to the next destination (possibly another mix node). [https://en.wikipedia.org/wiki/Mix_network]

LEAP proposes an alternative solution to the lack of forward secrecy in OpenPGP:

> *"We throw away the encryption and signature information once we have obtained the message and we reencrypt it in a new format [...] We throw away this information [key updates and signatures] and we instead take the cleartext, remove all the crypto baggage and then put it into an encrypted database. So once it's in the encrypted database, where the application is just trusting it, if the flag in the database says it was signed, we trust that it was signed. We don't keep any proof about it. [...] It's not like we found a magic way to support forward secrecy or non-repudiation but we have different problems with those than a normal client. Or I would say our problems are less important... because of the unique way in which we reencrypt in a different format". [Elijah, LEAP]*

Forward secrecy being coherent to the specifics of mobile and instant communication, LEAP team argues that the email usage is very different, including a different attitude to time and archiving: people do need archives in email, whereas disappearing messages are suitable and needed for IM communication.

On the client-side, LEAP works with solutions such as Pixelated (a hosted/cloud version of Bitmask), an automated mail encryption client with an easy-to-use interface inspired with the modern email clients. Pixelated aims in bridging the gap in user experience between existing mainstream proprietary solutions and open-source tools based on a federated infrastructure:

> *"We want client side tools that work with this federated infrastructure that work exactly like the tools people are used they have chat that works exactly like existing chat, they have email that works exactly like existing email… But it has very strong encryption both on the server and on the client device. Very strong identity verification which is the key thing that's missing in almost all security tools. So that's the goal. Same user experience". [Elijah, LEAP]*

Pixelated is aimed at a specific audience of users with low technical expertise. As our interviews showed, this tool has been tested with a community of brazilian farmers, a radical collective fighting to keep their land. The name of the collective was never explicitly mentioned for the sake of security. As Pixelated UI/UX designer notices, the client has been developed for conditions with bad Internet connection, and for a user-group that has few interactions with email and online services in general.

In Pixelated, key verification is invisible for users and happens automatically. When we assisted at the Pixelated usability workshop at Chaos Communication Congress in December 2017 (33c3), it was interesting to observe the reaction of its audience (tech-savvy people, frequent GPG users) to Pixelated UI/UX. A number of participants noticed that they did not feel their emails had been encrypted because they had "nothing to do". They wanted to "see encryption happening" and associated security with additional effort and user implication. The question of interoperability/interaction between GPG users and users of new automatic mail encryption services is a specific question (addressed by Autocrypt for example). Pixelated being a hosted/cloud solution, it puts less trust on the client side. This solution turns out to be interesting for specific use-cases, such as journalist work and physical device threat/seizure situation:

> *let's say you're a journalist [...] If you are travelling and doing reporting and crossing borders, your situation changes dramatically. You probably don't have a device that's always in your presence that you may trust, and may be you don't want to have one. So suddenly it makes sense to have a hosted version and to be able to say may be only on a temporary basis or on a permanent basis, I am gonna move my trust, my most sensitive things like my private keys that unlock my universe of communication archive, I'd like to move that to the web. And the unique thing about what we're doing with Pixelated and LEAP is that by moving that trust from your personal device to a server, you don't have to change your trust relationship with your email provider. [LEAP]*

(Self)-governance and advancement of federated projects implies an important community-driven effort and depends on engaging a variety of service providers and clients into accepting new open protocols or new libraries. Communication and consensus among various projects are needed in order to be able to advance in a federated environment. One of the practical examples of it is, for instance, the so-called "backwards compatibility" that makes a harmonious transition from older to more recent protocols possible, without "blocking" or "boycotting" some of the clients. The transition towards next generation encryption protocols within federated ecosystems may be very slow and difficult, however we are currently witnessing the rise of a powerful and diverse community of interested actors involved in a co-production of elements (protocols, packages, libraries…) necessary to prepare the ecosystem for adopting automatic encryption. One of the examples of such community effort is Autocrypt, now collaborating with K9, Enigmail, Mailpile and other important MUAs and service providers. Another federated project that undertakes important community-oriented efforts and changes the ecosystem is Conversations and the OMEMO protocol:

> *I think, the guy who wrote Conversations… he has done a lot to adapt XMPP to get rid of the stupid parts of XMPP and add good parts of fixed finite set of the parts that should be there, and wrote a good demonstration client, and* ***has done a good job of encouraging the servers to support this set.*** *It's a good example of other "can-be-changed" protocols. They kind of like gamified it*[66]*, you could run tests against different service providers to see how well they fit this new standard, and then there's rankings. There's way you can apply pressure to have protocol innovation and you can also have greater coordination. And if you produce high quality common libraries that everyone can use, then it's pretty easy to update the protocol because if you provide them with a new implementation that's worth the new stuff, than they can just upgrade to a new library.*

As we have seen in the first part of this deliverable, the field of end-to-end encrypted instant messaging applications is highly competitive, with important tensions happening among protocol and application developers, implementers and open-source community activists. Due to the very nature of centralized and non-interoperable encrypted IMs that "lock users" (as Elijah, LEAP founder puts it) within a tool with specific interfaces and sets of features, IMs compete for users. Email being an open federated ecosystem, it shows better collaboration and coordination effort. However, some tensions have been observed, namely

[66] Reference to the OMEMO.top project, a ranking of XMPP clients that have adopted OMEMO.

between pEp and Autocrypt projects (for example during the panel on automatic mail encryption at 33c3, December 2016). Both project adopt in-band key transmission, however “Autocrypt is a specification with independent implementations whereas the “pEp engine” is a product and open source library which mail programs are to integrate as an external dependency”. [See D 5.5] Besides the difference in the technical approaches of the two projects, the debates can be explained by the necessity for both initiatives to enrol an important number of email app developers in order to implement and spread their solution.

3. Distributed/peer-to-peer projects and anonymity

High-risk users show interest towards peer-to-peer systems, as they see a coherence between their political and economic models, based on horizontal connections, mutual help, self-governance, participation, and the technical architecture of distributed networks. These users are usually knowledgeable about mesh networks, or have a history of usage. Some heard of or used Ricochet (especially trainers in Ukraine), Tor Messenger or Tox. However, none of these tools is used on a daily basis and trusted as much as centralized applications or a more classical XMPP+OTR.

3.1. The promise of p2p encryption

The “promise” of p2p encryption is frequently cited among the Russian group chats we have observed (Telegram chats of Pirate Party Russia, Cybersecurity chat, internal Rublacklist chat). These users, whom we classify as “high-knowledge” or tech-savvy/tech-enthusiasts, regularly discuss the “re-decentralization” of the Internet(s). Two main aspects are underlined in these debates: the potential of p2p as a circumvention tool in the context of growing surveillance and censorship, and the technical features of p2p in terms of metadata protection. In the context of a state-centered Internet governance[67], Russian Internet activists suggest federation and p2p as a coherent technical answer that can potentially help users to “slip between the cracks” of state filtering and surveillance. The second aspect of p2p is related to privacy aspects and metadata; users believe that these solutions will have less impact on privacy compared to Google or Amazon-based solutions. They also believe metadata can be better protected within distributed or mixnet-based systems. Other discussions on re-decentralization concern the infrastructure level, namely the re-decentralization of DNS; alternatives such as Zero Tier One are mentioned.

In France, as well, discussions about the need to move away from proprietary and closed-source centralized services are spreading across tech-enthusiast communities. A new trend is developing, a “relocalization” of hosting and service providers. With the motto

[67] Ermoshina, K. & Musiani, F. (2017). “Migrating Servers, Elusive Users : Reconfigurations of the Russian Internet in the Post-Snowden Era”, *Media and Communication*, vol. 5, n° 1, p. 42-53.

"host local", a project called Chatons[68] has been launched by the "Degooglise Internet"[69] collective, to map local independent hosting, email and XMPP providers, in comparison with AMAPs (associations connecting consumers and local farmers). This movement suggests that instead of hosting data in a big centralized remote anonymous datacenter, it is more privacy-preserving to host it with someone you personally know. Trust relationships and sometimes even 'IRL' encounters give an additional layer of protection, in addition to TLS and end-to-end encryption.

Mastodon, a federated version of Twitter, is gaining popularity in France (most of the instances are French). Diaspora, a decentralized social network, is also gathering important communities of French privacy enthusiasts, namely through an instance called Framasphere. Our interviewees from the French cryptoparty scene comment on that:

> "*I feel like recently there's a riposte of European services to USA-based ones. I don't really understand why we should give our data to giant datacenters somewhere across the ocean. It's like eating our local food… You like French cheese, French strawberries, why not French hosting? Or even better... you can grow your own strawberries [laughs] or run an instance at your place" [A., informational security trainer, France]*

In this context, p2p solutions become part of a more global trend towards a re-localization that is associated with a more responsible attitude. De-anonymization of service providers paradoxically promises better anonymity and privacy online, which goes hand in hand with new protocol designs often based on IRL contacts and key exchange. Several projects exist that tend to redesign the backbone and propose a more direct and local, sometimes off-the-grid, device-to-device connection, in order to increase anonymity.

3.2. Briar: rethinking anonymization and resilience

Briar is born out of a problem that is activist and academic at once: how to increase anonymity and move communications off the backbone.

> "*I was working on p2p communication networks for my PhD and I reached a point where I realized that the fact to be able to observe the Internet backbone gives you the ability to observe all of the endpoints and their interconnections, that fact really completely shaped the possibilities for having private communications over the Internet. And I think people who'd been looking on completely different structures like Mix Networks came to the same conclusions. You can treat the entire anonymity system as a black box and if you can see the end points you could not get the anonymity" [Briar lead developer].*

The author of Briar, Michael Rogers, was contributing to LimeWire, a peer-to-peer file sharing service. In 2009, they were contacted by Iranian journalists from the Green

[68] https://chatons.org/
[69] https://degooglisons-internet.org

movement; activists were wondering if LimeWire would be suitable for communication among people in Iran:

> *The guy who contacted us worked for BBC Persian service. He had a principal interest to getting news from BBC into Iran but I think generally the question was essentially what can we do to support a movement like this. One part is getting news from the outside world, another part is disseminating news to the outside world, and the third way is kind of internal communication. And those are all things that we kept going as strands within Briar, how do we look at those different use cases. [Michael, Briar]*

At that time, LimeWire was not suitable and secure for high-risk communication, however, Michael suggested to build another, more secure tool. Together with activists, he sketched the rough idea of a network built over social connections, relying as much as possible on local network connections. This technical solution was relevant to the local political context: international connections in Iran were heavily monitored and filtered. Within those circumstances Michael with his team opted for an off-backbone communication: this collective effort ended up as Briar. The team now counts 4 members, with 2 developers, a UX/UI designer and a security/usability researcher who is also responsible for communications.

The name “Briar” refers to an organic metaphor, a distributed, rhizomatic and ramified structure. Behind the seemingly hostile appearance of Briar, it can become a protective environment:

> *I think that’s an american folks story: it’s about a fox that catches a rabbit and says: I am going to tear you into pieces. And the rabbit starts crying: Oh dude, please, do everything you want to me but please don’t throw me into the briar patch! So the fox eventually throws the rabbit into the briar patch, the rabbit runs away in the briar laughing: “I was born and bred in the briar patch, you know?” [...] In order to communicate privately we have to move away from these centralized services and rely on our social networks, and we have to fall back on these much more difficult structures to communicate” [Briar lead dev].*

The Briar Patch is also a specific region of space featured in Star Trek. According to the plot of the movie, Briar Patch emanates a specific "metaphasic radiation" that is concentrated in the planet's rings, continually rejuvenating their genetic structure. It is a region of space that starships usually avoid because of various radiation sources and energy fluctuations that impair communications systems and make it difficult for vessels inside the nebula to make contact with those outside the nebula.

Indeed, this description bears a close resemblance to Briar’s architecture and technical features, being designed for situations where communication with the “outside” Internet is hard to maintain. Briar focuses on a specific context of state-driven blockage and filtering, as well as extreme situations with a total Internet blackout. Connections in Briar are made over bluetooth, wifi and Tor. In this sense, Briar is designed both as a circumvention and an anti-surveillance tool:

> *What we had in mind specifically was how to get information in and out of the country in times of unrest when it might be blockaded, and it might be particularly difficult to reach Facebook [and other international sites]. One of the problems is how do you tunnel information outside or within the country and then let it spread widely outside the narrow tunnel. And that remains a question that people in Briar think of. People need to use it in conjunction with other tools and especially when they need to reach people who are not part of movement or whatever social group it is, and who are not using Briar. We need to think about bridges. So we have an RSS import feature to import a blog from a web. [Michael, Briar lead dev]*

Briar sees its users as people who are aware of their own need for security and who are aware of surveillance-related threats. Briar's main threat-model sees governments as the main threatening group of actors -- and attackers, performing filtering and interferences as well as blackouts – not only reading and intercepting communications and metadata. Briar is also intended to be a solution for crisis mapping and disaster response, and as such is aiming to collaborate with humanitarian organisations. Briar's UX/UI and usability concerns are informed by the experience of lead developer Michael Rogers, who had worked as an informational security trainer for journalists and had previously witnessed

> *the kind of enormous conceptual gap between what the designers of an encryption tools think that everybody knows and needs to know in order to make a system work, and on the other hand what a user actually tries to achieve through the use of it. [Michael, Briar lead dev].*

In this sense, one of Briar's concerns is to make a usable peer-to-peer tool for secure communication: while usability seems to be less of a burden for centralized systems, users have not yet formed "mental models" to embrace distributed secure communication:

> *With a certain technical structure that is more centralized, it is definitely achievable [...] But now the question is: can we also bring decentralization into that picture without breaking all of those mental models that users have and without asking them to learn a lot and make a lot of theoretical effort before they can use that tool. [...] What we're trying to achieve is a balance between asking a user to understand how the system works which is obviously a burden, or having a system do surprising things because it works differently from what they expect.*

Briar's identity management and key discovery models are linked to the structures of social movements and to the offline communication. In this sense, Briar redistributes trust relationship between human and non-human agents:

> *Social networks[70] are the foundation of all powerful social movements, so by emphasizing it we bring the attention back to the fact that all the security relies on the people that you can trust, by bringing those trust relationships to the fore… This very difficult constraint can turn in a strength. And I feel we are in the position of the rabbit,*

[70] This is not a reference to Facebook or Twitter, but about networks of contacts and interpersonal connections.

> *we're thrown in this supposingly hostile environment that actually was the place where we were born and bred.*

While Briar uses Tor as a "very well designed backbone that's designed to know as little as possible on what we do", the team is currently reflecting upon the limitations and security flaws found in Tor -- a concern that mirrors a broader preoccupation in privacy research. Indeed, Briar has been conceived as to be deployable on any kind of transport, and is not, by design, "attached" to Tor; actually, it is currently thinking of "migrating" to a different kind of distributed backbone:

> *That's also one of the reasons why I'm interested in Panoramix project because I think, Tor is starting to show its age. Some of the attacks we heard about as theoretical actually went very practical, and we need to think about anonymity infrastructure, privacy infrastructure that is not operated by someone in your house or on your street. [Michael, Briar lead developer]*

The Tor vulnerability mentioned here concerns the exit nodes and is related to the connection point between the onion network and the "normal" internet. The traces left by the exit nodes can provoke serious problems for the node administrators, as it happened to the Russian Dmitry Bogatov, arrested on April 10, 2017 because his exit node was used to post messages judged by Russian court as "extremist". The critique of Tor vulnerabilities brings the Briar team to think of a separate or disconnected, resilient network, independent from the Internet infrastructure:

> *I was looking for something that would work in a sort of partially disconnected delay tolerant environment, that kind of publish-subscribe when you exchange patches of messages with your peers, whenever the link is available and then you can be offline for an undetermined amount of time, or you can be connected to some peers but not all… The flexibility of that compared to what you have in a mesh-network where you try to have a path to every point available at every instant that seemed like quite practical place to start from. [Briar lead developer]*

Some of these ideas have already been developed within the Pond project -- itself a delay tolerant, mixnet inspired messaging system that introduces noise and latency to increase privacy and hide metadata. Another project that goes in a comparable direction is Scuttlebutt, an off-the-grid peer-to-peer social network or blogging platform.

Briar's personal inspiration comes from Usenet, when it was running on dial-up connections supporting early publish-subscribe systems on top of a "patchwork of different technologies before there was anything like IP address". The sustainability of Briar is supposed to be guaranteed by separating the protocol from the application[71].

> *If it's a piece of infrastructure it's really easy problem within the opensource world, there are a lot of well maintained pieces of infrastructure in the OS world. But if it's a user facing application that's more difficult, and that's partly why we want to make this separation because the user-facing app will probably have to be maintained with crowdfunding from users or hopefully it can be maintained on a volunteer basis*

[71] A similar dynamic was observed with Signal.

> *because most of the difficult technical plumbing will be moved into the infrastructure project, where the users don't have to maintain it. [Briar lead developer]*

The underlying Briar protocol is called Bramble. Briar is moving from the user-facing application to the codebase and infrastructure, and is expecting to guarantee sustainability of the project with no dependency on users -- or, to be more precise, in shifting from end-users to 'reusers' or power-users who can adapt the protocol to their needs and develop other projects on top of it:

> *The idea is that people can build other kinds of resilient networks on top of the same protocol stack and hopefully we can make sort of consulting business for people who need to communicate with devices out in the field or to communicate within teams that deploy in remote areas, that can be interested to use this kind of networking technology. [Briar lead dev]*

However, Briar is not used yet; test builds are available for Android devices on request. We have participated in a usability workshop for Briar at UCL in February 2017 with our UCL-based partner Marios Isaakidis. 11 people took part in the workshop, all of them being UCL PhD or postdoctoral students in computer science or usability. We tested several functionalities, such as key exchange, invitations for a one-to-one chat, group chat creation, blacklisting, changing the "trust level" of contacts.

The key discovery in Briar happens in two different ways: via a direct QR-code scanning and via indirect suggestion or invitation. The first configuration postulates the co-presence of the two users in the same physical space; this use context is considered as the most secure and the "trust level" is thus shown as "green". The second case supposes that two users have one contact in common; trust level is set on "yellow". Yellow can later be transformed into green when the two users meet. The "red" trust level is indicated for participants of a group chat with whom no key exchange has been established.

Briar group chat function provides interesting options for privacy protection: only the creator of the group connections is visible. This structure takes the shape of a "star" offering some degree of metadata protection to the Briar group chat participants.

The Briar project is now experiencing a transition phase as the team is choosing which path to take in the close future. They are working on the separation of the Bramble protocol from the Briar app, and are focused on looking for alternatives to Tor. This project, though unused by now, has been tested in "field" conditions in remote rural areas, where participants could communicate successfully in the Briar mesh on a limited distance. Even though Briar does not yet have an actual user base, it constitutes an interesting example of a project that is driven at the same time by research interests (usable p2p encrypted instant messaging in the context of resilient communications and blackouts) and by activist- and community-based motivations (the team members are frequent participants of Circumvention Tech, now Internet Freedom Festival; collaborating with Guardian project, GNUNet, Unlike Us, Open Internet Tools):

> *The sense of community is really important to have everybody motivated to work on these projects that are very open-ended and somehow against the flow that society in general is taking… where there is less and less privacy and more and more social control. It's nice to be reminded to know that other people are going in the same direction. [Briar lead developer]*

Ultimately, Briar is a socio-technical experiment (alongside other projects from the galaxy of p2p encryption tools, such as the MIT-based Vuvuzela) and as such, it is important to the field as it raises important questions about the limitations and problems of p2p IM, just as it shows its potential.

3.3. Problems of peer-to-peer instant messaging: from contact discovery to battery consumption

When it comes to p2p systems, an important part of the feedback from our respondents highlights the potential issues of distributed messaging systems. One of the most important ones is reputation management and identity management:

> *"Very specifically the things we care about most such as the ability to do social network mapping of social movements, the ability [...] to authenticate whom they are communicating with… and… certain usability properties of identities, are very very difficult to do in a peer to peer decentralized model. And a decentralized model also has issues with Sybil attacks, the question of how you control access, how you establish reputation when there is no barrier to entry. There's essentially no good way for a p2p model to have reputation. And [for] the Internet as we know, bad reputation management is a very big problem in any communication medium cause there's so much trolling". [Elijah, LEAP]*

The problem of reputation is also relevant for the Matrix.org founder who mentions important difficulties with trolling, spam and abuse within Matrix.

Another problem concerns user IDs. Identities are often represented in p2p messaging as "long hashes" (as in Ricochet, that uses Tor "rendez-vous" points). In this sense, these identities are unique but users usually find them hard to memorize:

> *"user IDs are long strings that are hard to remember… There's something that is called Zooko's triangle. For any identity system you get to pick 2 of the following 3 choices: you can have something where the names are globally unique, you can have something where the names are globally memorable, and you can have something where the naming system is decentralized. The problem is that everyone wants to get all three, but you have to pick two [...] some peer to peer, some federated models try to get all 3. But it also has problems" [Elijah, LEAP]*

Another potential problem is that a p2p architecture by design demands the device to be constantly online (as every device is also a "server"), that results in important battery consumption. Improving this aspect is one of the ongoing tasks for Briar:

> *The whole battery life thing made me think about how simple things like.. whether the client can remain connected to an anonymity network without constantly exchanging data, it is absolutely crucial, that's also a design constraint. [Briar lead developer]*

Finally, Briar's peer-to-peer design makes it impossible by now to have any kind of backup. This is a feature that can be seen in a positive way from the point of view of security in high-risk situations. However, it may be bad for some users who prefer to rely on cloud-based solutions:

> *Briar is in worth situation that some tools, by the moment you own account is stored on your device. If you destroy the device or uninstall Briar, you lose all your contacts and messages. [Briar lead developer]*

Seeking solutions to all of these problems is at the heart of current privacy and anonymity research. However, there seems to be a gap between academic research fields and activist needs and questions. Some projects such as LEAP and Briar are trying to work in between the two:

> *In computer science it's one of the greatest unsolved mysteries [...] The computer science problems that activists care about are not necessarily close to the computer science problems that are prestigious to work on in computer science. But for me as an activist working on usable communication this is a great unsolved problem. [Elijah, LEAP]*

NEXTLEAP's effort goes in the same direction, with our focus on activist use-cases (both high and low-risk) and collaboration with open-source developer communities (Autocrypt). We hope that this work will inform the protocol design process by mapping various architectures and licensing choices, UI/UX solutions and privacy/security features, and trying to analyze user mental models and their creative interactions with security and privacy scene.

Other cases of p2p Tor-based IMs are being covered in this fieldwork. We have interviewed Ricochet, Pond and Tor teams, and we will develop a more global overview of distributed messaging systems in the next deliverable [D3.5, M24], also including reflections on NEXTLEAP-related mixnet projects such as Panoramix. Within the ecosystem of distributed identity management, we will also cover a few relevant blockchain-based solutions such as Claimchain.

4. Conclusion

This deliverable has presented the results of the M6-M18 in-depth investigation of three end-to-end encrypted messaging applications, their communities of developers and users, and their 'ecosystem' of competing and allied projects. This analysis will be completed by M24, delving into decentralized applications, their qualities and potential issues[72].

[72] This is the reason for the presence of 'draft' in the title of the deliverable.

As Simondon puts it, technical objects "translate a number of notions and principles into matter"[73]; behind every protocol lies a number of powerful but not purely technical, and often political, design choices, which we have sought to investigate in this deliverable. These include centralization vs. decentralization (which in turn can be divided into federation vs. p2p), free software vs. closed-source vs. more nebulous varieties of licensing, and so on. In order to properly make explicit the notions embedded into privacy-enhancing technologies, we have combined the analysis of existing secure messaging software as technical objects (including analysis of code repositories, white papers and documentation, websites, interfaces) with in-depth semi-structured qualitative interviews with developers and users of such tools. In this conclusion, we elaborate on some directions opened up by the analysis in its current state, and we present the research agenda that will drive our efforts for the M24 and M36 deliverables in WP3.

4.1. On users' choices of e2e applications

> *It's very little about technology today. It's more about "who and where" and how much money you have. Who are people behind the application. [Peter Sunde]*

This deliverable has pointed out many facets of what contributes to users' choice of encrypted messaging applications. Users opt for an IM app either because other users in their reference group have been using it for a long time, because they trust the qualities and the leadership of the app's creator, or because of the technical properties of the tool and the protocol.

This range of choices may be compared to the three ways in which people legitimate their governments, as outlined by Max Weber[74]: the traditional, the charismatic and the rational. At times, users are driven by a willingness to 're-localize' their services, in order to have a better grasp of who their service provider is and what it does with their data. It is the case of projects such as "Degooglisons", "Framasoft" or "Chatons". This dynamic often goes hand in hand with a deanonymization of service providers so as to have a better accountability, leading to a debate on whether it is better to trust a local provider whom you know, or to trust a big company with its 'star power' but possibly unsavory practices.

The choice of an instant messaging tool can also be geographically determined. An example of this is Russians' widespread use of Telegram: as Telegram servers are located in five different countries around the world, outside Russia, its broadcasting function is used by censored media as a way to bypass the blockage, and by bloggers as an alternative to Facebook and traditional blogging platforms. At the same time, Telegram's quick rise on the market of messaging apps tells us a lot about the socio-economic factors that influence the success of an innovation in the field: it was when Facebook bought WhatsApp (followed by a

[73] Simondon, G. (1958), *On the mode of existence of technical objects*, University of Western Ontario, London, Canada.
[74] Weber, M. (1958). The three types of Legitimate Rule, *Berkeley Publications in Society and Institutions* 4(1): 1-11, 1958

several hours blackout for the latter), that the Telegram download rate exploded. As opposed to WhatsApp, Telegram can publicly underline its non-for-profit character and lack of ties with any commercial or governmental services.[75] Another example is provided by users being sceptical about american-based tools after Trump's elections: representations of a political situation can thus influence user's choice and her behavior regardings security.

Thus, to sum up, the reasons why users choose a particular secure messaging app has to do with both technical and non-technical factors. Among the latter is the presence of a community (whether other users are already using it, and what are their profiles); the charisma and reputation of the authors (e.g. Moxie for Signal, or the Durov brothers for Telegram); the geopolitical status of the underlying infrastructures, such as servers, and the specific data protection regulation of the countries in which they are hosted. Among the technical features are the presence of group support features (very important in the Russian case), or anonymity features for group support and the history (the main reason of Cryptocat's popularity in Ukraine). For highly tech-savvy users, the licensing solution is also important (in particular, several communities are keen on open source), as well as federation or peer-to-peer in those cases when it is important to be 'coherent' with specific political choices.[76]

We have observed an interesting tension between the protocol design choices and user behavior/"real-life" usage patterns, both during our interviews and at the cryptoparties. While some developers were focusing on designing the most secure protocol with all the features "baked in", informational security trainers were pointing to the fact that users find unpredictable ways of using the tool, that often go against the initial design ideas. This is a classical situation observed in the STS tradition of user studies[77] where users are considered as "actors of innovation", contributing to the advancement of innovation and research by "hijacking" (*détourner*), opening up, circumventing some of the "by-design" features and functionalities.

In the case of e2e encryption messaging systems, protocol designers need to integrate this user behavior in their understanding of "threat models". They actually need to take into consideration *"how to match what protocol does to what users are trying to achieve and where the possible mismatches can cause misunderstanding? What can cause people to bypass security measures that you put in place to make things more convenient?" [Briar lead developer]*

During our fieldwork, we have encountered manifold situations where, regardless of strong encryption, security flaws and vulnerabilities occurred due to operational security and offline

[75] We describe the Russian situation in more detail in Ermoshina, K. & Musiani, F. (2017). Migrating Servers, Elusive Users: Reconfigurations of the Russian Internet in the Post-Snowden Era", *Media and Communication*, vol. 5, nº 1, p. 42-53.
[76] From March to June 2017, we observed Pirate Party Russia chats and RosKomSvoboda chats, where discussions took place on what encryption tool best suited the needs of these activist groups. Finally, one group stayed on Skype after having launched a vote for migrating to another tool; another group migrated to Matrix.org after two months of discussions, and the reason of migration was related to the federated nature of Matrix, even though encryption is in beta.
[77] Akrich, M. (1998). Les utilisateurs, acteurs de l'innovation, *Education permanente*, 134 : 78-89.

user behavior, such as the material location where the hard-drive was kept, the use of a phone in a crowded place, the modalities of password storage, the strength of passwords, the presence or absence of a two factor authentication, the habit of users to keep their accounts logged in by default and so on: "*Making sure that you don't have a little screenshot of the app in the "recent apps' menu on Android, is almost as important practically to maintain the privacy of information as all the work you do on crypto, on the network protocol… It's a very good rebalancing exercise." [Briar lead dev]*

This issue of "rebalancing" the strong protocol design and user education, or a "behavioral change", appears as crucial. From developers' perspective, this behavioral change may be triggered with the development of some instructions, educational elements, explanations, tutorials or quiz in the UI of the app. Autocrypt, for example, is considering to implement a pedagogical feature to show "what your provider sees", while OONIprobe implements a quiz that users have to pass before they can start using the tool.

Another effort to reach out to the users and build bridges between users and developers is the "Cryptoparty" movement. This international movement was born in Australia in 2012 and is spreading all around the world. It has recently been institutionalized with the publishing of several instructions published on "how to run a cryptoparty". Other efforts are done by international organizations such as EFF or Tactical Tech collective, with developing guides for users, and designing classifications of encryption tools.

4.2. "Ordering" the mess of messengers: classifications of encryption tools, their limits and consequences

The "in the making" quality of the field of e2e encryption in secure messaging has led us to include in our inquiry a 'meta'-research question on how the cryptography community in a broad sense (including these different categories of stakeholders) is currently reflecting upon sets of criteria that would allow to account for the quality of tools and protocols, and classify those tools according to the presence/absence of specific properties. Focusing in particular on the pioneer effort led by the Electronic Frontier Foundation, by means of the Secure Messaging Scorecard, we have sought[78] to examine the role of attempts of categorization and classification of secure messaging tools and we have observed how, by challenging, re-examining and re-shaping the categories that are meaningful to define the quality of secure messaging tools, actors "carrying some weight", such as the EFF, are able to spark a "global crypto discussion" that currently contributes to shape what constitutes "good" security and privacy in the field of encrypted messaging.

Classifications and categorizations, Bowker and Star point out, are "powerful technologies" in themselves, whose architecture is simultaneously informatic and moral[79]. Due to their

[78] Ermoshina, K. and Musiani, F. (under review). What is a *good* secure messaging tool? The EFF Secure Messaging Scorecard and the shaping of digital (usable) security. *Westminster Papers in Communication and Culture*.

[79] Bowker, G. & Star, S. L. (1999). *Sorting Things Out: Classification and Its Consequences*. Cambridge, MA: The MIT Press.

embeddedness in working infrastructures, they can become relatively invisible as they progressively stabilize, without losing their power. Thus, categorization systems should be acknowledged as a significant site of political, ethical and cultural work -- three aspects that our analysis of the SMS has examined. In these three respects, categories are performative: far from being “enshrined [...] in procedures and stabilized conventional principles that one merely needs to follow in order to succeed”[80], they actively participate in the construction of the relation between the different actors that have a stake, or a role, in the categorized environment; categories, in this view, are one of the components of a complex network of actors and technologies.

In the case of the SMS’s pioneer effort, categorizing encryption tools seem to contribute to the “opportunistic turn” in encryption[81] that gained momentum in 2014 after the Snowden revelations, and consists in a progressive move of the crypto community towards making encryption “seamless”, with almost no efforts required from users. In terms of design choices, this entails a “blackboxing” of quite a few operations that used to be visible to users, and needed to be actively controlled by users [e.g. key exchange and verification, choice of encrypted/unencrypted status etc.]. The opportunistic turn calls for an “encryption by design”, and constructs a new user profile, one who “does not have to” have any specific knowledge about cryptographic concepts and does not have to undertake any additional operations to guarantee a secure communication. At the same time, users are at the core of the most recent categorization systems, and thus, they are entrusted with an important decision-making responsibility. Users have to question their threat models, increase their awareness of them, and have to know how to make technological choices according to their particular situation -- a tool is “good” if pertinent to the context of use.

4.3. Towards a governance of encryption

A number of authors, some of them with STS sensibilities but overall coming from a broader disciplinary spectrum, have examined in recent years how the concept and the practice of governance may be reconsidered in light of an increasing number of informal uses, practices, norms that affect the distribution and the exercise of power on the Internet. While Michel van Eeten and Milton Mueller argue that the definition of governance should include “environments with low formalization, heterogeneous organizational forms, large number of actors and massively distributed authority and decision-making power”[82], Sandra Braman suggests that the definition of governance may go as far as including “decision making with constitutive (structural) effect whether it takes place within the public or private sectors, and

[80] Denis, J. (2006). “Les nouveaux visages de la performativité”, *Études de communication*, 29: 8-24.
[81] Internet Engineering Task Force (2014) Request for Comments 7435, Opportunistic Security: Some Protection Most of the Time, https://tools.ietf.org/html/rfc7435
[82] Van Eeten, M. J., & Mueller, M. (2013). Where is the governance in Internet governance?. *New Media & Society, 15*(5), 720-736

formally or informally"[83]. Governance may even be just a side effect of actions with non-governance-related aims"[84].

Throughout this deliverable, we have been able to witness several dynamics that speak to 'Internet governance' in this sense -- e.g., interoperability and de facto standardization processes; tensions between centralization, federation and decentralization of technical architectures -- and of communities; concentration of leadership, and controversies between prominent albeit informal 'leaders'; and last but not least, the openness of code, which is linked to both geographical differences and to the variety of user threat models, and is a concern of different importance for different actors. While keeping the source code more or less open is a heavily debated issue among developers, open-source and licensing choices are less covered in contexts of appropriation and education of users, even those who live and operate in high-risk contexts as high-risk users do not always associate open-source with security.

As the field of end-to-end secure messaging apps is still very much "in-the-making," the analysis of interfaces and underlying protocols and architectures, coupled with in-depth interviewing, helps us to address important questions related to Internet governance. For example, an analysis of the technical design choices made by developers can both provoke new questions in the cryptographic research community, and lead to the revisiting of previous design choices that the secure messaging developer community may have made that are at odds with user expectations. Other important questions include long-term changes in infrastructure via corporate deployment and standardization (possibly via traditional avenues such as the IETF), which requires inspecting the business models (or lack thereof) and attitudes of developers towards adoption. Furthermore, the choice of centralization, decentralization, federation can be qualified as a tentative of "governance by infrastructure"[85] -- an attempt to stabilize a governance model, both for the technology and the communities managing it, through the technology itself.

[83] Hofmann, J., Katzenbach, C., & Gollatz, K. (2016). Between coordination and regulation: Finding the governance in Internet governance. *New Media & Society*. http://doi.org/10.1177/1461444816639975

[84] Braman, S. (2009). *Change of state: Information, policy, and power*. Cambridge, MA: MIT Press.

[85] Musiani, F., Cogburn, D. L., DeNardis, L. & Levinson, N. S. (2016, eds.). *The Turn to Infrastructure in Internet Governance*. New York: Palgrave/Macmillan.

Appendix. Question templates for semi-structured interviews

All the interviews we conducted are semi-structured, with open questions. In every case, a set of context-specific questions was also added. Thus, for developers, based on our preliminary reading of the documentation and analysis of the website and UI of the tool, we formulated specific questions regarding the tool. For users, country-specific questions were added. For developers, we ask the following technical questions:

How have you chosen the name?
How have you come up with the technical solution?
Were you inspired by some other projects or was it created from scratch?
Do you have a threat model? List of properties?
How do you come to decide what components of the software secured using a cryptographic or privacy-enhancing protocol and what's not?
How do you come to decide what kinds of user data you need to store or use?
Have you chosen centralized architecture, and are you thinking of moving towards decentralization?
Do you support the transfer of large files?
Do you support repudiation? Do you let users archive or search their messages?
What kinds of groups does your protocol support?
What kinds of metadata do you collect, and why?
Do you use tools (ranging from programming languages and cryptographic libraries to development environments) in the same field as the one you are developing? Which ones and why?
Every developer is also a user. You, as a user, what kind of difficulties do you experience with technologies you depend on, for example, with cryptographic libraries?

We also ask a number of social questions to developers:

How many people are in your team?
How do you share responsibilities and tasks?
Who's allowed to make changes?
In addition to software development, is there an operational component to your work that includes security (such as hosting servers)?
What's your choice of licensing? Is the protocol you use standardized, working towards a standardization or do you prefer not to standardize the protocol?
How do you sustain yourself financially?
What is your business model of running the any infrastructure, such as servers?
Which other projects from the field are you collaborating with?
How do you communicate with these projects?

Has your protocols ever been reimplemented by other projects?
What is your opinion on the existing academic work in the field?
Do you collaborate with researchers?
If you use research, what do you read? Blogs, papers?
What conferences or gatherings do you organize or convene with developers and/or users in the field?
How do you explain your politics (such as data collection) to your users?
Do you get in touch with your users or informational security trainers? If yes, how do you gather feedback? Do you know who your users are?

For users, the following questions were asked:

Can you tell us about the moment when you installed your first encrypting tool?
What was this tool? Why have you decided to use it?
Were you satisfied with this tool? Has it helped?
Did you install it by yourself or did someone helped you?
Was it easy to use? Since then, which other privacy-enhancing technologies have you tried, if any?
If you stopped using some of these tools, can you explain when and why have you abandoned them?
Have you tried PGP? If yes, have you installed it by yourself or has someone helped you?
Was it easy to install? Where did you learn to install it?
What is your “privacy kit" for today? Describe it, of which tools it consists?
Why have you chosen these tools?
How do you use it in your profession/activism?
Can you define “who is your enemy"?
What would happen to you if your enemy got your messages?
Do you worry about any data these tools store, and what data?
Do you know if these tools store your list of contacts on their servers? Do you worry these servers could be monitored, or seized?
What do you worry about more, your device being seized or the server?
Do you want the ability to be able to move your data between servers?
Are you more concerned over your old messages being read or new messages being read?
Do you want to search through or archive your old messages?
How often do you send large files as attachments?
Do you want your messages to disappear? Do you know if they disappear on your device or on the server?
What features are missing on secure messaging application?