DATA PROTECTION BY DESIGN

these digital systems can help Humanitarian Organizations connect with, assist and deliver services to people

digital systems from external service providers are increasingly embedded in humanitarian programming

these same digital systems can also create risks for the people that Humanitarian Organizations assist

CHALLENGES

Third Party digital systems often fulfill multiple purposes by allowing the collection or processing of unnecessary amounts of data

regulatory compliance alone may not reduce all risks to people in a humanitarian crisis

RISK MITIGATION

identify precisely what a system needs to achieve

require a technical design that only achives this specific purpose

CHAPTER 6

DESIGNING FOR DATA PROTECTION

Carmela Troncoso and Wouter Lueks

6.1 INTRODUCTION

Humanitarian Organizations assist the most vulnerable populations in extremely challenging circumstances. For reasons of efficiency, accountability, and out of a desire to help as many people as possible, Humanitarian Organizations increasingly rely on digital technology in their programmes. The livelihood and safety of vulnerable populations often relies on the assistance provided by these organizations. As a result, individuals have very little agency in whether to accept the assistance and whether to participate in these digital systems if they wish to accept the assistance. Digital systems bring data protection and privacy risks. Especially for vulnerable populations, these risks might be significant. Therefore, humanitarian organizations have an obligation not just to safeguard individuals' livelihood in the short term, but also to uphold data protection as well as privacy rights and the dignity of the people they help.

This relevance of digital systems is not limited to the humanitarian sector. As these systems gained prominence, in the early 2010s policymakers and researchers redoubled their efforts to ensure that the design of these digital systems ensured strong privacy protection. On the policy side, regulatory efforts aimed to set a legal basis for respectful and privacy-preserving digital services.[1] On the research side, a vast number of privacy-enhancing technologies and building blocks have also been produced for privacy-friendly systems, in addition to end-to-end privacy-preserving systems for a wide range of particular use cases such as electronic voting, document Searches for investigative journalists, and gun registration databases. There have also been efforts to articulate specific strategies to design privacy-friendly systems.[2]

Despite these advances, the process of designing and engineering systems with strong privacy and data protection remains a challenge. One of the main reasons is that privacy-preserving properties of technological outputs are often difficult to map onto data protection regulations, policies and principles.

Typically, Humanitarian Organizations do not design their own systems but instead provide requirements to potential service providers. As a result, they need to assess

[1] See, for example, efforts such as: EU Regulation 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of Personal Data and on the free movement of such data, and repealing Directive 95/46/EC (EU General Data Protection Regulation), [2016] OJ L119/1; California Consumer Privacy Act, 2018 Cal. Legis. Serv. Ch. 55 (A.B. 375) (WEST); and Ann Cavoukian, "Privacy by Design: The 7 Foundational Principles", Information and Privacy Commissioner of Ontario, January 2011: www.ipc.on.ca/wp-content/uploads/resources/7foundationalprinciples.pdf.

[2] Jaap-Henk Hoepman, "Privacy design strategies (Extended abstract)", *IFIP Advances in Information and Communication Technology*, No. 428, 1 January 2014, pp. 446–459.

the solutions provided to them by asking the right questions and requesting better analyses. In this chapter, we aim to provide the reader with means to question about the privacy and data protection provided by digital systems. We do so via a privacy-engineering methodology that can be used to produce designs that provide strong privacy protection. Systems that adhere to the privacy-engineering principles we present will, by design, fulfil data minimization and limit the purpose for which the data that are collected can be used. By providing technical means to enforce these data protection principles, systems engineered according to our methodology provide strong protection of individuals, their dignity and rights.

More concretely, this chapter provides guidance on how to determine the purpose of a system and shows how purpose limitation can guide the system designer into creating systems with strong privacy and data protection by design. Finally, this chapter provides concrete guidance on how to analyse a system to determine whether it implements technical means to enforce purpose limitation and therefore provides strong protection for its users, beyond those that could be achieved via data usage policies.

We finally note that the methods and technologies introduced in this chapter address the need for data minimization, purpose limitation and data security included in data protection. However, it does not address other data protection requirements, e.g. accountability. Yet, the design principles introduced in the chapter will enable Humanitarian Organizations to assess whether the mechanisms to be added to fulfil all data protection requirements are detrimental to the technical protection of individuals and their rights.

6.1.1 WHAT IS A SYSTEM?

This chapter often refers to a “system”. We define a system as “a combination of interacting elements organized to achieve one or more stated purposes”.[3] That is, the system encompasses all the parts (or elements) that are necessary to achieve a purpose. Following this definition, a system is composed of more parts than just a central server. Typically, a system includes at least user devices.

6.2 CASE STUDY: PRIVACY-PRESERVING CONTACT-TRACING APPS

To illustrate how starting from the purpose of an application and using technology to enforce purpose limitation leads to strong privacy guarantees, this chapter uses the

[3] Joint Task Force Interagency Working Group, *Security and Privacy Controls for Information Systems and Organizations*, NIST Special Publication 800-53, National Institute of Standards and Technology, September 2020: https://doi.org/10.6028/NIST.SP.800-53r5.

example of a privacy-preserving system that has been successfully deployed at large scale: a privacy-preserving contact-tracing system based on mobile apps.

In the beginning of the COVID-19 pandemic, contact-tracing apps were introduced as a public health intervention to help break infection chains. Contact tracing aims to identify close contacts of people infected with COVID-19 so that these contacts – who are likely to have been exposed to the SARS-CoV-2 virus through their proximity to a COVID-19-positive person – can take action (e.g. quarantine) to avoid spreading the disease in case they contract COVID-19 themselves.

To be effective, contact tracing must be timely and reach as many contacts as possible. Traditionally, tracing is done manually. However, due to its reliance on trained personnel, manual contact tracing cannot scale when diseases, such as COVID-19, spread to many people. Manual contact tracing is time-consuming because contact tracers have to manually interview *index cases*, meaning the people that contracted the disease. The index cases have to identify their contacts, and then the tracers have to reach out to these contacts one by one. Furthermore, when dealing with airborne pathogens, index cases may not be able to identify all contacts because the contacts' identities may in fact be unknown to the index case (e.g. passengers on a bus and people waiting in line at the supermarket).

In the initial months of the COVID-19 pandemic, several digital solutions were proposed to address the limitations of manual contact tracing. In this chapter, the focus is on solutions that use Bluetooth technology to measure proximity between people and then use close-proximity events with index cases (people with COVID-19) to automatically notify users of their risk of having been exposed. Such digital systems scale better, because they do not have to rely on manual interactions with index cases or contacts. They also can have better coverage, as they do not require people to know who they came into contact with, nor to have their contact information. At the same time, these digital solutions are inherently limited to only finding close contacts that also use the contact-tracing app.

Contact-tracing applications have the potential to expose Personal Data, including sensitive Personal Data. This type of information has historically been abused to profile, manipulate and control individuals and populations.[4] Thus, privacy-preserving contact-tracing applications were created in a way that ensures that those Sensitive Data are not available, and therefore cannot be abused for purposes other than notifying users of danger of infection.

[4] See e.g.: Balthasar Staehelin and Cécile Aptel, "COVID-19 and Contact Tracing: A Call for Digital Diligence", Humanitarian Law & Policy Blog (blog), 13 May 2020: https://blogs.icrc.org/law-and-policy/2020/05/13/covid-19-contact-tracing-digital-diligence.

The next section explains the design of privacy-preserving contact-tracing applications. Readers familiar with these applications may skip to Section 6.3 – Protection of individuals and their dignity and rights through purpose limitation.

6.2.1 DECENTRALIZED PRIVACY-PRESERVING PROXIMITY TRACING

This chapter uses the example of privacy-preserving contact-tracing applications based on the Decentralized Privacy-Preserving Proximity Tracing (DP3T) protocol.[5] This protocol enables the creation of a *decentralized* system with strong protection by design. Sensitive data, such as information about social interactions between users, are stored and processed on users' devices rather than in a central entity. Phones locally compute exposure scores and notify users if their exposure to COVID-positive users is too high. Some data are exchanged via a server, but on their own these contain no sensitive information about users and cannot be abused or misused.

Every user of the digital contact-tracing system installs an app on their phone. At a predetermined interval (around 15 minutes), apps generate a fresh random number. Apps broadcast the random numbers via Bluetooth Low Energy (BLE) beacons. Nearby phones record received numbers in a list of seen numbers. Devices in close proximity receive the transmitted Bluetooth beacon with high signal strength, and those further away either receive it with low signal strength or do not receive the beacon at all. Low–strength beacons are not recorded as they indicate the devices are not close enough to indicate risk of infection.

When a user tests positive, the health authority authorizes this user to upload to a central server the random numbers that their phone transmitted during their contagious period. The central server periodically publishes a list of all random numbers transmitted by COVID-positive users. All devices in the system download this list, and check locally whether any entries on their list of seen random numbers (e.g. corresponding to people that were physically close to them) appears in the list of random numbers that they downloaded (corresponding to people that were contagious). Overlap between these lists indicates proximity of the user to index cases, and potential exposure to the SARS-CoV-2 virus. If this exposure – determined by the length of proximity as well as the relative signal strength – passes the threshold, the phone notifies the user.

In this system, very little information leaves the user device. The central server receives only the random numbers transmitted by COVID-positive users. These numbers are randomly generated by the user device and have no relationship to the user's identity or location. These random numbers are also independent of how

[5] Carmela Troncoso et al., "Decentralized privacy-preserving proximity tracing", *ArXiv:2005.12273 [Cs]*, 25 May 2020: http://arxiv.org/abs/2005.12273.

many people a user has met or the frequency and duration of those meetings. However, if the random numbers of a positive user are published, this user may become easier to track and identify for attackers that can receive Bluetooth beacons (and thus these random numbers) at many locations.[6] In summary, the server holds very little information that could potentially be used to harm users.

Returning to the definition of a system as a combination of components that are organized to achieve a stated purpose, users' phones in the DP3T system collaborate with a central server to fulfil the purpose of notifying users that have been exposed to the SARS-CoV-2 virus. Therefore, both phones and the central server are part of this system. In fact, because the public health authorities must be able to authorize the upload by users that tested positive in a digital manner, the public health authorities (or at least the servers they operate) are part of the contact-tracing system as well.

6.3 PROTECTION OF INDIVIDUALS AND THEIR DIGNITY AND RIGHTS THROUGH PURPOSE LIMITATION

There exist several methodologies and principles that guide the design and analysis of systems to achieve strong data protection and provide strong privacy. This chapter uses a methodology based on guaranteeing **purpose limitation by design**, through the careful introduction of techniques in the design of digital systems to enforce this data protection principle. This methodology is comparatively easy to use, leads to systems with strong privacy guarantees and automatically shows the limits of privacy-friendly designs.

The reader might be most familiar with *purpose limitation* as a data protection requirement, which requires that data are collected for a specific purpose and it forbids these data to be used for any other purposes. Traditionally, purpose limitation is enforced through processes and procedures. This chapter, however, uses purpose limitation in a *technical* sense:

> A system that implements *technical purpose limitation* ensures, through its technical design, that the system as a whole can only be used for the stated purpose. Such systems make pieces of data accessible to adversarial entities only when doing so is part of the stated purpose.

As a result, systems designed to achieve technical purpose limitation minimize the potential harms stemming from how and which data are collected and processed in these systems. In a system with purpose limitation, data cannot be used for anything

[6] The DP-3T Project, *Decentralized Privacy-Preserving Proximity Tracing: Overview of Data Protection and Security*, GitHub, 2020: https://github.com/DP-3T/documents/blob/master/DP3T%20-%20Data%20Protection%20and%20Security.pdf.

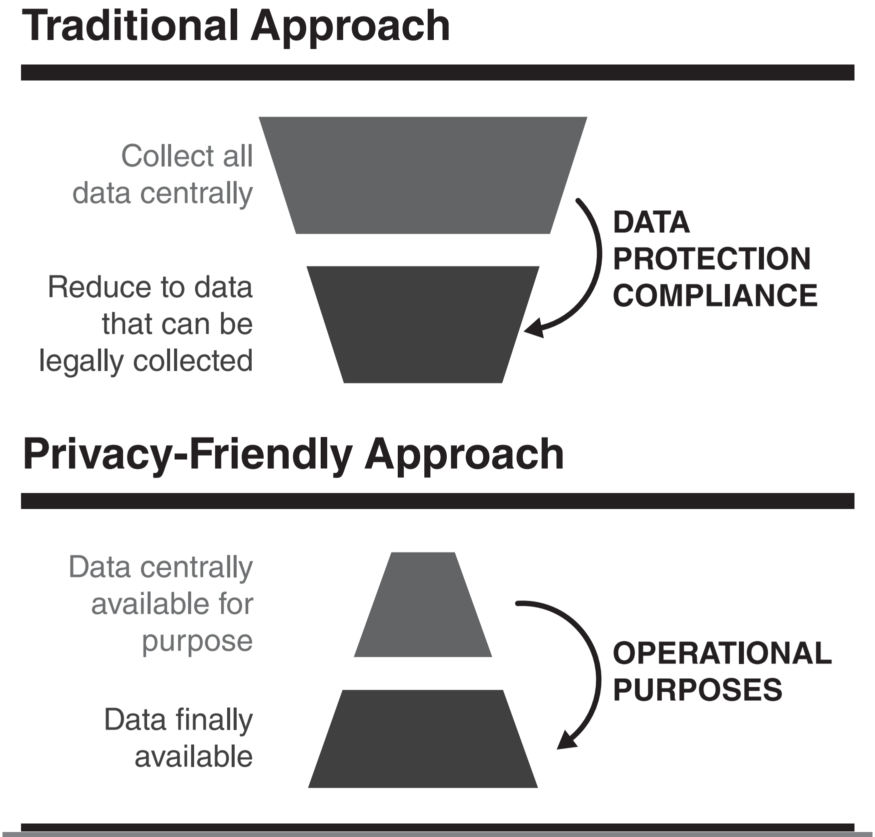
Traditional Approach
Collect all data centrally
Reduce to data that can be legally collected
DATA PROTECTION COMPLIANCE
Privacy-Friendly Approach
Data centrally available for purpose
Data finally available
OPERATIONAL PURPOSES

Figure 6.1. Data collected centrally (at entities untrusted by the user) as a result of starting from purpose limitation are strictly less than when minimizing data through compliance mechanisms.

but the purpose of the system. Therefore, users do not need to trust that other actors in the system are going to behave appropriately, or that they will not violate the data protection policy. The protection against abuse holds even if these other actors intentionally try to do harm.

Thinking in terms of technical purpose limitation has strong implications for the amount of data collected by entities that are out of the control of, and therefore not trusted by, the user. The top diagram in Figure 6.1 shows the approach followed most often when deciding what data should be collected when building a digital system. Typically, designers start by creating systems that collect as much data as possible (with the idea that these data will become useful in the future). Then, regulatory compliance – mainly data protection compliance, or operational constraints, such as storage or processing capabilities – limit how much of these data are finally collected (see Figure 6.1 – Data collected centrally).

By contrast, when designers reason about data collection in terms of the purpose of the system, the starting point is a system that, as a whole, collects and processes only those data that are necessary *to fulfil the purpose of the system*. Collecting any additional data would violate purpose limitation. Sometimes, operational constraints force the collection or processing of additional data. However, even then, the amount of data collected is strictly less than would be allowed by looking at regulatory compliance alone.

The remainder of this section will focus on how to evaluate a system using the mechanism of purpose limitation by design. This involves two key steps. First, the evaluator must establish the purpose of the system. Second, the evaluator must assess whether a system implements technical purpose limitation given this specific purpose.

6.3.1 WHY DETERMINING PURPOSE MATTERS

The first step in designing a privacy-preserving system is to determine the purpose of the system. Narrowing the purpose to the essential goal for which the system is to be deployed is essential. Should the purpose be too broad or ill-defined, it may become very difficult, or even impossible, to design a system with strict purpose limitation and hence strong rights and dignity protection guarantees.

Broad purposes are harmful for privacy and limit data protection. To see why broad or ill-defined purposes are harmful, consider contact-tracing apps. Suppose that instead of the narrow purpose “notify contacts of index cases”, the much broader purpose had been “perform contact tracing”. This latter purpose is so broad that it may be understood as performing all steps associated with the manual contact-tracing process, including epidemiological surveillance, backward tracing (to identify sources of infection rather than potential new cases), monitoring notified patients, and enforcing their quarantine. Satisfying such a broad purpose may require making all kinds of data available to public health authorities, including identities of users, contact information, location, etc. These data could subsequently be abused. None of these data, however, have to be available centrally when the sole and specific purpose of the system is to notify contacts of index cases.

Even narrower purposes can be harmful when they force extra data to be made available. For example, the purpose of the German Luca contact-tracing system was to make available to public health authorities the names and phone numbers of visitors to locations with contagious individuals. By requiring that such information is available, the ability of data protection compliant designs to limit purpose is severely affected, even if data are only available to others under some conditions.

The system has to be trusted, and trust may be violated, e.g. the German police did use the Luca system to access visitor information despite some protections being in place.[7]

Designing privacy-preserving systems for multiple purposes is challenging. Defining multiple purposes can also reduce the privacy guarantees that systems can satisfy, even if they implement purpose limitation. A common example in commercial applications is to include "improving customer experience" among the purposes to motivate central collection of data, regardless of whether the system has a very concrete goal (e.g. a mobile flash light app) or a very broad one (e.g. an app to manage financial assets). When such an "improvement" purpose appears, the amount of data that is collected and made available centrally can increase considerably: from application-related data (e.g. how long the was torch on) to other data that are not strictly about the application but are very related to customer experience (e.g. the battery status of the phone when the app is opened, or the number of apps installed that are running at the same time as the application).

When a purpose makes additional data available centrally, privacy becomes difficult to protect. The relations and correlations among pieces of data, especially when those data are related to humans and their behaviour directly or via their devices, make it extremely difficult to predict the amount of inferences that can be done on these data, the amount of predictions that they can enable, and therefore the amount of uses that they can have in the future. The difficulty in determining the inferences that can be made from different types of data is similar to the problems encountered in the search for robust Anonymization mechanisms:[8] the curse of dimensionality. The fact that there are too many data fields correlated in unpredictable ways prevents the Anonymization algorithm designer from identifying all possible pseudo identifiers. Hence it becomes close to impossible to design robust Anonymization mechanisms without destroying the utility of the data.

Having multiple purposes also constrains the privacy-preserving ways that systems can be designed. For instance, it is the fact that contact-tracing apps are only aimed at notifying that enabled the deployment of a design in which only uninformative random identifiers need to be exchanged through the server. Any extra purpose

[7] Rachel Pannett, "German police used a tracing app to scout crime witnesses. Some fear that's fuel for covid conspiracists", *Washington Post*, 13 January 2022: www.washingtonpost.com/world/2022/01/13/german-covid-contact-tracing-app-luca.

[8] Arvind Narayanan and Vitaly Shmatikov, "Myths and fallacies of 'personally identifiable information'", *Communications of the ACM*, Vol. 53, No. 6, 1 June 2010, pp. 24–26: https://doi.org/10.1145/1743546.1743558; Theresa Stadler and Carmela Troncoso, "Why the search for a privacy-preserving data sharing mechanism is failing", *Nature Computational Science*, Vol. 4, 21 April 2022, pp. 208–210: https://doi.org/10.1038/s43588–022-00236-x.

(e.g. quarantine enforcement or epidemiological surveillance) would probably have forced designers to make more data available centrally. This would then have made it much more difficult to constrain data to be used only for one purpose. For example, enforcement typically requires location information, or at least knowledge of whether a user is home or not; and epidemiological surveillance requires revealing chains of infection, and therefore revealing relationships.

Even when the multiple purposes of a system do not inherently create privacy vulnerabilities, it might be difficult to create practical purpose-limited systems that provide purpose limitation. Building solutions that implement purpose limitation is difficult, especially because these systems must be optimized to be deployable in practice. For example, systems for privacy-preserving medical analysis based on homomorphic encryption require very careful domain-specific optimization to perform well enough.[9] It is difficult to take such systems and use them for different purposes without having to repeat the challenging optimization process to accommodate new constraints.

The temptation of purpose creep. Finally, once a system or infrastructure is built, there is the temptation to add purposes to take advantage of the existing components. For instance, in the contact-tracing applications ecosystem many extensions were suggested, ranging from epidemiological monitoring, to quarantine enforcement, to collecting data on notified users. Ultimately, these were not implemented, but others were. In the second half of 2020, researchers discovered that COVID-19 does not just spread via droplets to close-proximity contacts, but also via aerosols in ill-ventilated rooms. Many countries thereafter adopted check-in solutions applying contact tracing to visitors of shared indoor spaces in addition to the existing proximity-based systems. As soon as this functionality was added, it was immediately suggested that it should also be used to monitor and enforce regulations about maximum capacity in bars and restaurants. However, adding these enforcement mechanisms would mean exchanging more information between users' devices and central servers, making it much harder to implement purpose limitation.

Humanitarian Organizations can expect similar desires and pressures in the humanitarian sector. Systems that are built to prevent double dipping in aid distribution can be seen as opportunities to optimize resource allocation. And systems that are built for authentication of beneficiaries can be seen as opportunities to monitor usage of resources. While these purposes may be perfectly legitimate, and even desirable, it is important to understand that aiming to include all of them simultaneously may make it impossible to design a system that offers strong privacy guarantees, and that

[9] David Froelicher et al., "Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption", *Nature Communications*, Vol. 12, No. 1, 11 October 2021, 5910: https://doi.org/10.1038/s41467–021-25972-y.

enforces purpose limitation via technology. The system could still be built, but privacy and data protection may need to solely rely on policy and regulatory protections. Such protections may not be sufficient, depending on the environment where the system is to be deployed.

6.3.2 DETERMINING PURPOSE

Defining the purpose of a system is not an easy task. As a general rule, the narrower the purpose, the easier it is to find technological means to engineer the system in such a way that it ensures purpose limitation and hence provides strong protection for individuals and their dignity and rights. Typically, determining the purpose requires discussions with stakeholders to determine the main goal of the system. In these conversations, many purposes may arise, often as a consequence of the fact that the power of technology as a means to solve problems is often overestimated.

At that point, it is important to isolate these purposes and identify what is the underlying problem that the system should address, and what are additional desirable functionalities that could address other problems or increase the efficiency of the system, or the organization commissioning it. Once purposes are set apart, the designer must decide which of them can be implemented while providing purpose limitation, thereby avoiding information leakage that could lead to abuse. In this step, the designer may discover that the purpose of the system itself induces risks, for example because the purpose requires making Sensitive Data available to untrusted parties.

Risks may also be introduced by design or implementation choices. This is typical for privacy-preserving designs; see Figure 6.2. Designers may aim for designs that only have risks that are inherent to the purpose, but then might end up with a design that has slightly more risks. Often this is because they either do not know how to build a system that fully mitigates these risks, or they know how to but cannot make such a

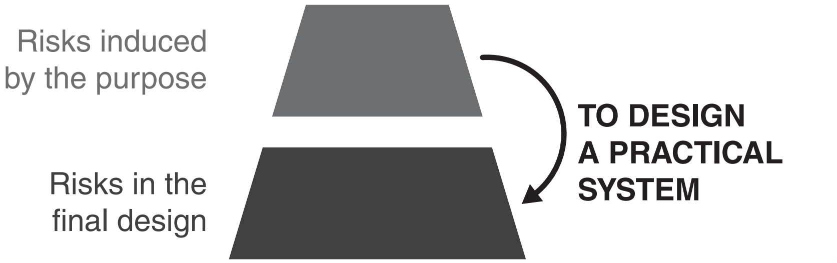
Risks induced by the purpose
Risks in the final design
TO DESIGN A PRACTICAL SYSTEM

Figure 6.2. Practical and deployable systems might have somewhat higher risks than those induced by the purpose alone.

system efficient enough. At this point, there are two paths forward: decide not to build the system as the risks are too large, or build the system, accepting that the harms to individuals' rights and dignity might be larger than what is induced by the purpose and cannot be controlled other than via policy (see Figure 6.1.)

The example of privacy-preserving apps illustrates both inherent and implementation-specific risks. First, consider a risk that is inherent to the purpose of notifying contacts of index cases. When users receive a notification, they may be able to identify the index case that triggered it (e.g. if they were only with one person on the day when the reported contact took place); and thus learn medical information. This leakage is inherent to the functionality of the system: the system *must* notify the user, so that the user can take appropriate measures. Second, consider an implementation-specific risk. The server only receives uploads from positive users. The server, or any observer of the communication, can thus determine pseudo identifiers (e.g. IP addresses) of users that tested positive for COVID-19. In this case, the processing of pseudo identifiers also means that the system processes more data than would be strictly needed to fulfil its purpose (see Figure 6.1).

Limiting the purpose of the system enables privacy engineering at its best (and not limiting may result in solutions providing no privacy protection and therefore risk harming individuals). However, this decision may have implications on the efficiency and cost incurred by an organization. Adopting purpose limitation by design may require building one or more privacy-preserving system for each desired purpose. It is not the goal of this chapter to determine which option is best, as it may depend on the resources available and the conditions in which systems are to be deployed. Instead, this chapter aims to provide guidance to conduct a risk–benefit analysis that will enable Humanitarian Organizations to make informed decisions about the trade-offs between data protection and other operational constraints.

6.3.3 ANALYSING PURPOSE LIMITATION

After having determined the purpose of a system, the next step is to assess whether a particular technical implementation of a system provides (technical) purpose limitation. This is not a straightforward process, as it may be hard to determine whether a system cannot be used for any other purpose than the one stated by the stakeholders.

The following two-step approach can be used to tackle this complex process. First, identify potential *privacy risks* in the system that can result in harms for individuals and their rights and dignity by analysing *all* the data that are produced, stored or processed in the system (regardless of where these operations happen) as well as an exploration of potential harms that could be caused by the system in general. Second, for each identified risk, determine whether this risk is inherent to the system's purpose (in which case there is nothing to be done; see above) or whether the technical privacy-preserving protections implemented in the system mitigate this risk.

As mentioned in the introduction, this chapter deliberately excludes the design of systems because typically Humanitarian Organizations do not design their own systems.[10]

6.3.3.1 IDENTIFYING POTENTIAL RISKS

Identifying potential risks is a complicated process because risks might not be obvious. Humanitarian Organizations may take a combined bottom-up and top-down approach. For the bottom-up approach, start by looking at all the data that are processed and available in the system. Risks should be derived based on who could be harmed when such data would be made accessible, either directly or indirectly. To identify potential risks, specific implementation details such as whether data are processed on users' devices only, distributed between central servers, or available on a single central server should be ignored.

Because the bottom-up approach might fail to identify some risks, it should be combined with a top-down approach that instead starts by identifying potential harms of deploying the system and derives risks from them. When reasoning about who could be harmed, it is important to remember that the subjects of harms include not only individual users, but also groups or communities. These groups or communities may be significantly affected as a whole even though the harm to individual members may be considered acceptable. In fact, this harm may happen even if they do not actively participate in the system.

From data to risks. The digital contact-tracing system described in this chapter also serves as an example of the data-driven bottom-up approach. The data processed in contact-tracing systems must reflect *social interaction data* (e.g. who meets whom, when and for how long) so as to enable the calculation of exposure risk. Any digital contact-tracing system therefore runs the risk of *leaking social contact information*. Additionally, the system may risk *leaking location data* (and consequently risk becoming a *tracking infrastructure*) and also *leaking users' identities*. Indeed, digital contact-tracing solutions can reveal location data. In Germany, the Luca apps focused on tracing visitors to locations with contagious individuals. The police leveraged contact-tracing information stored in the Luca app to request and obtain contact data of visitors to specific venues.[11] Digital contact-tracing systems also process medical

[10] For more details about how to design and implement a privacy-preserving design once purpose has been identified, refer to: Seda Gürses, Carmela Troncoso and Claudia Diaz, "Engineering privacy by design reloaded", in *Amsterdam Privacy Conference*, 2015: http://carmelatroncoso.com/papers/Gurses-APC15.pdf.

[11] Rachel Pannett, "German police used a tracing app to scout crime witnesses. Some fear that's fuel for covid conspiracists", *Washington Post*, 13 January 2022: www.washingtonpost.com/world/2022/01/13/german-covid-contact-tracing-app-luca.

data: who tested positive and who was possibly exposed to this user. Digital contact-tracing systems therefore risk leaking sensitive medical status such as *leaking who tested positive* and *risk leaking who is exposed*.

Risks can also relate to data that do not, in any way, correspond to individuals. For example, contact-tracing solutions such as Luca that focus on tracing visitors to locations with contagious individuals potentially risk *leaking data about locations* because the system must keep track of which locations exist. While this risk is probably low when such a system is only used for bars and restaurants (whose locations are probably already public), this is not necessarily the case for other locations. When such a system is deployed more broadly, the risk of *leaking data about locations* is definitely present. In fact, when the database of a comparable system in Australia leaked, it revealed the location of defence sites and domestic violence shelters.[12]

From harms to risks. The above risks mostly relate to Personal Data (social contacts, location data, medical status). However, risks can also relate to groups of people. To identify these risks, Humanitarian Organizations may apply the top-down approach. In the case of contact-tracing applications, civil society groups identified the harm of stigmatization. Stigmatization can manifest in different risks. First, there is the risk that the system can be used to create *heat maps of medical data*. Such heat maps could then result in stigmatization of particular venues or neighbourhoods, for example, when it turns out that immigrant neighbourhoods have a higher incidence of COVID-19 cases or contacts. Similarly, there is the risk of revealing *demographic information about index cases*. This could result in stigmatization of particular minorities, for example, if it turns out that the prevalence of a certain disease is higher among gay men.

Finally, privacy and other human rights are not always related to keeping data secret or minimizing their disclosure. Instead, they can relate to other rights such as freedom of movement. The way to elicit such risks is to reflect on what the system is or could be used for. For example, the consequence of being notified via a contact-tracing app is that users self-quarantine. This is a serious restriction of movement. Potentially, such a mechanism could be abused, leading to a risk of *population control*.

6.3.3.2 ASSESSING THE PRESENCE OF RISKS

Once risks are identified, Humanitarian Organizations can use them to either drive the design of new systems[13] or to assess the design of existing systems. This chapter

[12] Jonathan Kearsley and Clair Weaver, “Sensitive business addresses among 500,000 published in COVID data breach”, *Sydney Morning Herald*, 14 February 2022: www.smh.com.au/politics/federal/sensitive-business-addresses-among-500-000-published-in-covid-data-breach-20220214-p59wal.html.

[13] See for example: Gürses, Troncoso and Diaz, “Engineering privacy by design reloaded”; Hoepman, “Privacy design strategies (Extended abstract)”.

describes the latter process, linking the mitigation of risks to the technical enforcement of purpose limitation.

Assessing whether the technological design of a system mitigates all risks is a challenging process which requires specialized technical knowledge. This knowledge might not be available at Humanitarian Organizations. Therefore, Humanitarian Organizations should identify potential risks (see the previous section) and then ask for assistance to determine how and why the proposed system mitigates the identified risks. This can be done by asking the designers of the system to explain how the risks are addressed, or by contacting experts (e.g. academics) that can provide an external assessment of the technology given their knowledge about developing and deploying privacy attacks.

Why the privacy-preserving contact-tracing system implements purpose limitation. Recall the risks that are listed above for the contact-tracing applications. Regarding the risk of leaking social contact information, all information related to social contacts (the lists of received random numbers) is stored only on individual user devices, and never leaves these devices. Therefore, the design mitigates this risk.

Regarding the risk of leaking location data, apps do not collect *any* location data. Thus, there are no location data in the system to be leaked or abused. However, not all data protection risks materialize directly. Recall that devices broadcast random numbers in Bluetooth beacons. These random numbers could, potentially, be used to *track* users if there exists an eavesdropping infrastructure *external* to the system. Because phones rotate their numbers every 15 minutes, users that do not test positive cannot be tracked.

At the same time, the DP3T design cannot fully mitigate the risk of tracking for positive users. Recall that positive users will upload all random numbers that they broadcasted to the server to enable exposure computations at other devices. Because of a performance optimization, all random numbers broadcasted on the same day by the same positive user are linked to each other. This makes it possible to track a positive user, given enough Bluetooth coverage. Notice that here the need to design a deployable system increased the risks (see also: Figure 6.2).

Regarding sensitive medical data, phones determine locally whether a user has been exposed and should be notified. No data about this notification are ever communicated to any other party. This ensures that neither data about individual exposures nor group exposures (heat maps) leak. As discussed above, any contact-tracing system that makes notifications enables the potential identification of users that tested positive. This is also true in this design.

Finally, the decentralized design limits the possibility of population control by making it difficult to falsely trigger a notification. Recall that a phone shows a notification when it (1) received a random number and interpreted it as coming from a close-by device, and (2) this same random number later appears on the list it downloads from the server of numbers transmitted by contagious users. To trigger a false notification, an attacker must be close enough to the target to transmit a random number via Bluetooth, and then trick the server into accepting an upload. Neither of these is strictly speaking impossible, but it seems that performing this attack is difficult at scale.

Even though some residual risks remain, using the data *within* the system none of these risks can be materialized: the system by design ensures that the purpose can only be notification.

A contact-tracing system that does not implement purpose limitation. As a counter-example, consider an alternative contact-tracing system also aimed at notification that does not ensure purpose limitation: the NeedToKnow (NTK) system that was proposed in Germany.[14] Like the decentralized design, in NTK phones exchange numbers and store them locally. But in NTK: (1) these numbers are not random, the server knows which numbers every user transmits, and (2) users that test positive upload the list of numbers *that they received.* Because the server can link numbers to people, such a system could potentially be used to track users. For example, law enforcement could request the list of numbers corresponding to a suspect and then use Bluetooth receivers to track that suspect. Indeed, both in Singapore[15] and Germany,[16] contact-tracing systems have been used to track people.

Additionally, because users that test positive upload the list of numbers that they received to a server and that server can relate numbers to people, the server can learn social interactions of positive users. Finally, the system can also know which users test positive and which have been notified.

Given this analysis, it is clear that data created and collected *within* the system can be used for purposes beyond notifying users. Thus, this system does not technically enforce purpose limitation.

[14] Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT), *High Level Overview*, 2022: https://github.com/pepp-pt/pepp-pt-documentation/blob/8ba05287c349318a03837fe374fd949e60d4eaf8/PEPP-PT-high-level-overview.pdf.

[15] Mia Sato, "Singapore's police now have access to contact tracing data", *MIT Technology Review*, 5 January 2021: www.technologyreview.com/2021/01/05/1015734/singapore-contact-tracing-police-data-covid.

[16] Rachel Pannett, "German police used a tracing app to scout crime witnesses. Some fear that's fuel for covid conspiracists", *Washington Post*, 13 January 2022: www.washingtonpost.com/world/2022/01/13/german-covid-contact-tracing-app-luca.

6.4 THE ROLE OF DATA MINIMIZATION

Regulatory frameworks and researchers have proposed many principles to guide privacy and data protection practices. One of the key principles used to judge the privacy and data protection guarantees of a design is the data minimization principle.[17] It requires that data controllers collect data only when those data are necessary for the stated purpose. These data should be retained only for as long as is necessary to fulfil that purpose. This principle is also reflected in technical literature.[18]

Data minimization is indeed a necessary condition for a privacy-preserving design. Not collecting data that are unnecessary for the operation of the system, and deleting data that are no longer needed, reduces privacy risks for users of the system. Reducing the amount of information stored, for instance via aggregation or using privacy-preserving cryptography, also reduces the likelihood that users' privacy is breached and thus reduces the risk that their rights and dignity are affected.

At the same time, it is not always easy to apply the data minimization principle to assess the level of protection offered by a system. Consider a naive (non-private) contact-tracing system where apps send detailed information about contacts to a central server. The server then uses these data to identify and notify contacts of infected people. In this case, the server stores sensitive social interactions data. To minimize data collection and processing *at the server*, privacy-friendly designs let apps compute a user's exposure to the virus *locally* on the users' devices based on locally stored interaction records.

Here, a data minimization can quickly fall short. Surprisingly, when examining the privacy-friendly system as a whole – including users' devices and server – there is no data minimization. Both centralized and decentralized systems, when seen as a system, collect, process and store contact data. The difference is where these data are stored: centralized systems store most of these data at a central server, whereas the privacy-preserving systems distribute these data across user devices and the server.

As a result, applying the data minimization principle does not let an analyst distinguish between these two designs. Even though they obviously have very different privacy and data protection properties. The purpose limitation by design approach

[17] See Section 2.5.4 — The principle of data minimization.

[18] Hoepman mentions data minimization as one of the key techniques for creating privacy-friendly designs, and Gürses et al. argue that creating privacy-friendly designs requires thinking in terms of data minimization. See: Hoepman, "Privacy design strategies (Extended abstract)"; Gürses, Troncoso and Diaz, "Engineering privacy by design reloaded".

does not suffer from this problem. Moreover, any system that provides purpose limitation by design, also provides data minimization by design. Only data that are explicitly allowed by the purpose can be available at entities that are outside the user's control. If more data are available, purpose limitation is violated because these data could be used for other purposes.

6.5 CHALLENGES TO PURPOSE LIMITATION

The previous sections have shown how the principles of purpose limitation, and data minimization to a certain extent, can be used as guidance to design and evaluate whether a particular system design offers strong privacy protection and therefore can guarantee that rights and dignity are preserved. This section highlights aspects related to the design that may limit the designer's ability to implement purpose limitation, and therefore what an evaluator should look for to understand the level of protection offered by the system.

- **Lack of requirements or evolution of requirements.** In modern software development cycles, the requirements of the system are not fully fleshed out at the beginning of the design process. Instead, the designers augment and modify them in an *agile* manner. While this may be very desirable from a development and deployments perspective, the use of such development techniques greatly limits the privacy guarantees that a system can provide. If requirements are not clear, it is hard to identify the purpose and therefore design for purpose limitation. If the purpose has to remain flexible, then there is little that the designer can do to guarantee strong privacy and ultimately the protection of individuals' rights and dignity.
- **Reliance on Third Party services.** A second characteristic of modern software is that designers and developers do not program all modules in their system. Instead, they rely on tools, libraries or services programmed and executed by others. While this speeds up the development and ensures high-quality dedicated modules that offer very good performance at low cost, the use of these elements hinders the application of the purpose limitation system. These Third Party elements constrain, via their interfaces, what data the application can use, and in which format. This in turn limits the number of privacy-preserving technologies that the designer can use, as most will not be compatible with the requirements of the Third Party service.
- **All system layers play a role in data protection.** Humanitarian Organizations typically reason about privacy protection from the point of view of the application: what its is purpose, what data it requires, where these data are stored and processed, etc. In reality, the data of the application are a small portion of the overall (meta)data existent in the system that can lead to a breach of a Data Subject's rights and freedoms. In this sense, it is important to think about privacy as a *weakest link* property: either protection is ensured at all layers, or the users' protection is limited to the protection provided by the weakest of the layers.

- For example, one of the risks in contact-tracing systems is that attackers learn which users tested positive (see Section 6.2.1 – Decentralized privacy-preserving proximity tracing). The weakest link here is not the application (which hides which users are positive from everyone but the server), but the network layer. Recall that only users that test positive upload data to the central server. Any network observer could thus conclude, based on the existence of this network traffic alone, that the user tested positive. Therefore, deployed systems use countermeasures against such network attackers.[19]
- **What is technologically viable.** Deciding which technologies to use in order to implement the strongest purpose limitation can be challenging. In many cases, the most constrained implementation requires the use of non-mainstream techniques, or the development of new technologies – as in the case of contact-tracing apps. Such knowledge may not be available to Humanitarian Organizations, and in many cases also not to the developers of the products they commission. Similar to the evaluation, Humanitarian Organizations may partner with academic institutions to gain knowledge on the possible technologies and designs. Even when those designs are not economically or operationally viable, knowledge of what would be the ideal situation may help the organization to be able to make better decisions as to whether a system is desirable or not.

[19] See also: Section 5.2.5 – Identify risks.