Privacy and data protection in mobile applications

A study on the app development ecosystem and the technical implementation of GDPR

NOVEMBER 2017

www.enisa.europa.eu

European Union Agency For Network and Information Security

About ENISA

The European Union Agency for Network and Information Security (ENISA) is a centre of network and information security expertise for the EU, its member states, the private sector and EU citizens. ENISA works with these groups to develop advice and recommendations on good practice in information security. It assists member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. More information about ENISA and its work can be found at www.enisa.europa.eu.

Contact

For queries in relation to this paper, please use isdp@enisa.europa.eu
For media enquires about this paper, please use press@enisa.europa.eu.

Legal notice

Notice must be taken that this publication represents the views and interpretations of ENISA, unless stated otherwise. This publication should not be construed to be a legal action of ENISA or the ENISA bodies unless adopted pursuant to the Regulation (EU) No 526/2013. This publication does not necessarily represent state-of the-art and ENISA may update it from time to time.

Third-party sources are quoted as appropriate. ENISA is not responsible for the content of the external sources including external websites referenced in this publication.

This publication is intended for information purposes only. It must be accessible free of charge. Neither ENISA nor any person acting on its behalf is responsible for the use that might be made of the information contained in this publication.

Copyright Notice

ISBN: 978-92-9204-242-4, DOI: 10.2824/114584

Table of Contents

Executive Summary

While online users increasingly rely on the use of mobile applications (apps) for their everyday activities and needs, the processing of personal data through such tools poses significant risks to users' security and privacy. Such risks stem mainly from the variety of data and sensors held in mobile devices, the use of different types of identifiers and extended possibility of users' tracking, the complex mobile app ecosystem and limitations of app developers, as well as the extended use of third-party software and services. For these reasons, the implementation of the core data protection principles, as stipulated by the General Data Protection Regulation (GDPR), faces serious challenges in mobile apps. This may hinder compliance of mobile app developers and providers with specific rules of GDPR, e.g. with regard to transparency and consent, data protection by design and by default, as well as security of processing.

Against this background, the scope of the present document is to provide a meta-study on privacy and data protection in mobile apps by analysing the features of the app development environment that impact privacy and security, as well as defining relevant best-practices, open issues and gaps in the field.

To this end, the document explains the basics of the app development lifecycle and takes a look at different depictions of mobile app ecosystems (development versus deployment). While the ecosystem is complex, an app developer centric approach is taken, while also addressing app providers and other actors in the ecosystem (OS providers, device manufactures, market operators, ad libraries, etc.).

Specifically, roles and responsibilities are analysed and aspects of software development are discussed as they can be leveraged as privacy and security action points. A presentation of idealized app lifecycles (data versus development lifecycles) is performed, as well as their potentials for implementing privacy by design. Particular attention is paid to the Agile Secure Development Lifecycle and possible ways of extending it to also cover privacy and data protection requirements. The permission model of apps is used as an example for a more detailed analysis of data protection challenges in the current mobile app development and deployment practices.

Moreover, the document focuses on the concept of privacy by design and tries to make it more clear, especially for mobile app developers. Approaches to privacy and data protection by design and by default are presented that help translate the legal requirements into more tangible engineering goals that developers are more comfortable with. In particular, the concepts of data protection goals and privacy design strategies are discussed in general terms, while providing concrete examples from the mobile app development perspective.

The main conclusions and recommendations of the overall study are as follows:

Providing guidance to app developers

In the area of mobile apps and privacy there is still a serious gap between legal requirements and the translation of these requirements into practical solutions. Existing recommendations to app developers usually provide insights only into *what* the developers are required to do, without further guidance on *how* they will be able to fulfil these requirements.

Policy makers, regulators (e.g. Data Protection Authorities) and the research community should provide recommendations to app developers that shed more light on how privacy requirements can be fulfilled in practice.

Professional organisations and the industry of app developers should provide more insight on current development practices and key privacy and security issues.

The European Commission and EU institutions in the area of privacy and security should promote the formulation of workshops and working groups with all relevant stakeholders in the field.

Policy makers and regulators should invest in awareness raising for app developers and other actors in the app development ecosystem, in co-operation with professional organisations and the industry of app developers.

Need for scalable methodologies and best practices

The preference for agile development methods makes it difficult for developers to apply heavy-weight methods developed for big design up front approaches (in which the design is fully completed before implementation is started). It is, thus, important to provide scalable methodologies for security and data protection by design, taking into account the different positions that app developers may take in the app ecosystem and the possibilities and constraints associated with these.

The research community should continue work in privacy and security engineering for apps, focusing in particular in the integration of privacy requirements as part of app development and SDL like methods.

The European Commission and EU institutions in the field of privacy and security should support relevant initiatives and research projects in this area.

Regulators (e.g. Data Protection Authorities) should provide an assessment of developed systems and methodologies and point out best available techniques, as well as state-of-the-art solutions that can be adopted by data controllers or app developers.

A DPIA framework for mobile apps

One place where the GDPR spells out an activity that aligns well with development rather than data operations is with data protection impact assessments (DPIAs). Still, existing methods for applying DPIAs tend to assume a big design up front model which clashes with the agile practices.

The European Commission and regulators (e.g. Data Protection Authorities) should closely co-operate with professional organisations, the industry of app developers and researchers with the aim of developing a DPIA framework for mobile apps.

Policy makers and regulators should promote the use of DPIAs as a means for threat modelling and building data protection by design and by default in their tools.

Improving privacy and usability in the developer ecosystem

Further studies are needed to understand the state of the art in the privacy opportunities and obstacles inherent to IDEs, APIs and OSs, as well work on providing APIs for PETs and data protection relevant functionality. Further, there is a need for tools to test, verify and audit existing libraries, services, APIs etc.

The research community should continue work in enhancing privacy and usability in the developer ecosystem, e.g. by empirical evaluation of IDEs, APIs and ad libraries.

The European Commission and EU institutions in the field of privacy and security should support relevant initiatives and research projects in this area.

Addressing the entire mobile app ecosystem

The design and the functionality of an app is not only dependent on development methods, but on the operation of the entire mobile app ecosystem. For a comprehensive approach in protecting privacy and data protection this overarching issue of governance must not be neglected.

The European Commission, policy makers and regulators (e.g. Data Protection Authorities) should address the governance issues of today's mobile app ecosystems with respect to maintaining privacy, data protection and other fundamental rights. Interdisciplinary research groups and practitioners should support this endeavour.

1. Introduction

1.1 Background

The mobile application (app) ecosystem has emerged to become one of the biggest industries in the world, encapsulating millions of app developers[1] and billions of smartphone owners[2] who use mobile apps in their daily life. Recent statistics provide for a forecast of 197 billion mobile app downloads in 2017[3], with most popular app categories being those of games, social networking, entertainment, news, as well as health fitness and lifestyle.

However, as indicated in several studies[4], while online users increasingly rely on smart mobile devices (e.g. smartphones, tablets) for their everyday activities and needs, the processing of personal data through such tools is not always transparent to or controllable by the users. Moreover, the understanding of how the apps practically operate is often complex, due to their dynamic environment, reuse of software libraries and interconnection with different networks and systems, thus making it even more difficult to assess their privacy and security characteristics. Although few app developers might maliciously oversee data protection obligations, in many cases poor data protection and security practices are due to lack of awareness, knowledge or understanding (from app developers) on how to practically organize for and engineer privacy and security requirements into their tools.

The processing of personal data through apps is regulated by the General Data Protection Regulation (EU) 679/2016 (GDPR) [1], which will be, as of 25 May 2018, the main data protection legal framework in the EU directly applicable in all Member States, repealing the current Data Protection Directive 95/46/EC [2]. While reinforcing all the data protection principles, obligations and rights enshrined in the Directive, the GDPR includes additional protection mechanisms to allow individuals to better control their personal data, which is especially a challenge in the online and mobile environment[5]. On top of GDPR provisions, in the area of mobile apps privacy and data protection requirements also stem from the Directive on privacy and electronic communications 2002/58/EC (ePrivacy Directive) [3] , which is currently under review in order to be updated and aligned with GDPR. In January 2017 a proposal for a new ePrivacy Regulation was made by the European Commission and is currently being debated in the European Parliament and the Council.[6] This proposal sets out important provisions for privacy and data protection in mobile apps, with regard to confidentiality of communications and related metadata, the placement of software and data (e.g. cookies) on user devices and the regulation of privacy settings in relation to tracking.

Taking into account the mobile apps privacy considerations and the underlying legal framework, it is essential to define specific recommendations for mobile app developers and other parties of this ecosystem,

[1] http://www.businessofapps.com/12-million-mobile-developers-worldwide-nearly-half-develop-android-first/

[2] https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

[3] https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/

[4] See for example results from Mobilitics project conducted by CNIL and INRIA, https://www.cnil.fr/fr/node/15750

[5] Recital 9 of the GDPR states: "The objectives and principles of Directive 95/46/EC remain sound, but it has not prevented fragmentation in the implementation of data protection across the Union, legal uncertainty or a widespread public perception that there are significant risks to the protection of natural persons, in particular with regard to online activity."

[6] See European Commission, Proposal for an ePrivacy Regulation, available at https://ec.europa.eu/digital-single-market/en/proposal-eprivacy-regulation. See also European Parliament, Legislative Train Schedule, Proposal for a Regulation on Privacy and Electronic Communications, http://www.europarl.europa.eu/legislative-train/theme-connected-digital-single-market/file-e-privacy-reform.

which would help them 'translate' the legal requirements into practical implementation controls and appropriately safeguard users' security and privacy. In order to do so, however, it is essential to have a clear and detailed understanding of the current state-of-play of mobile apps design and behaviour with regard to data protection. Such analysis should provide more insight on the current practices and tools used for the development and maintenance of apps, including the way that data protection requirements are considered, as well as the extent to which privacy enhancing technologies (PETs) are embedded.

Against this background, ENISA initiated a project under its 2017 work-programme in the area of privacy and data protection of mobile apps, with the aim of establishing the current state-of-the art and providing guidance for further work in the field. The present study is the result of this work.

1.2 Scope and objectives

The scope of this document is to provide a meta-study on privacy and data protection in mobile apps by analysing the features of the app development environment that impact privacy and security, as well as defining relevant best-practices, open issues and gaps in the field.

The focus of the study is on apps running on personal mobile devices (such as smartphones and tablets). IoT devices or other types of smart devices (e.g. smart cars) fall outside the scope of the document. Moreover, the study is especially centred around the app developers, i.e. the entities responsible for building the app, as these entities can greatly influence the technical design of apps and make relevant decisions for implementing privacy and security requirements. Also, it addresses the app providers, i.e. the entities that offer the app to end-users or otherwise the 'owners' of the app. The app providers usually fall under the definition of 'data controller' in European data protection law. App developers may also be the app providers, but this is not always the case and this raises important issues about their relative role in ensuring data protection rules are complied with and privacy and security requirements are implemented in practice. Besides focusing on app developers and providers, the study takes account of the broader environment of mobile app development and other relevant parties/stakeholders, including the app market (app stores), OS providers, device manufacturers and the providers of third-party libraries.

In terms of objectives, this study specifically aims at:

- Analysing the privacy and data protection risks arising from the use of mobile apps, as well as the relevant key legal and regulatory issues (GDPR and ePrivacy Regulation).
- Exploring the current software development practices and challenges specific to app developers (and the entire app ecosystem) with regard to the processing of personal data.
- Presenting privacy and data protection by design examples and paradigms for mobile app development.
- Making proposals for future steps in the field, in particular with regard to research activities, as well as practical guidance and recommendations that can be provided to app developers and other entities in the app ecosystem.

The target audience of the study includes app developers, providers and other parties in the app ecosystem, as well policy makers, regulators and the research community in the areas of security, privacy and data protection.

1.3 Methodology

The study was supported by an expert group, comprising of: Claude Castelluccia (INRIA), Seda Guerses (KU Leuven), Marit Hansen (ULD), Jaap-Henk Hoepman (Radboud University Nijmegen), Joris van Hoboken

(University of Amsterdam), Barbara Vieira (SIG). It was mainly based on desk research, including literature reviews on the state of the art in mobile app security and privacy, regulatory documents, comparison of mobile app ecosystem models to identify different actors and associated data protection issues, empirical studies and interviews with developers, as well as the expertise of the established expert group in privacy and security engineering. The preliminary output of the study was presented at a closed ENISA workshop in June 2017 with external stakeholders (especially app developers, regulators and researchers in the field). The feedback from the audience and discussion with other partners was then used to finalize the results of the study.

1.4 Structure

The remaining of this document is structured as follows:

- Chapter 2 provides an overview of privacy and data protection risks of mobile apps, followed by an analysis of the key legal requirements (stemming from GDPR and ePrivacy Regulation). It also refers to privacy risk management and Data Protection Impact Assessments (DPIAs).
- Chapter 3 examines the app development ecosystems and current development practices, focusing especially on the actors involved, the role of the app developers, as well as if and how security and privacy requirements can be integrated in the apps development process.
- Chapter 4 takes a deep dive into the permission architectures in mobile apps as an example of addressing privacy and security in the development process and relevant challenges.
- Chapter 5 provides an abstract description and examples of the notions of 'data protection goals' and 'privacy design strategies' as key elements of embedding data protection by design and by default in mobile apps.
- Chapter 6 derives conclusions from the previously conducted analysis and makes relevant recommendations for further actions in the field.

The study complements previous ENISA's work in the fields of privacy by design [4] and security of mobile devices [5].

2. Privacy and data protection requirements in mobile apps

In this chapter we first analyse the specific privacy and data protection risks of mobile apps. We then present the key legal and regulatory issues, arising from the application of the requirements from the GDPR and ePrivacy Regulation in the mobile app environment. Attention is paid to the practical implementation of the aforementioned requirements and the challenges that they bring. Finally, we discuss risk management and the DPIA (Data Protection Impact Assessment) process in the context of mobile applications.

2.1 Privacy and data protection risks

The privacy and data protection risks of mobile apps stem mainly from two dimensions: a) their nature, as software running on private mobile user devices (handheld devices), and b) the particularities of the mobile development and distribution environment as such. In the next sections we provide a more detailed analysis of the relevant risks and risk factors.

1. Variety of data & multiple sensors

Mobile devices can typically have access to various types of personal/sensitive data (such as wellbeing, health, medical data) provided by users via various mobile apps. Furthermore, standard handheld devices embed many and various sensors (microphone, camera, accelerometer, GPS, Wifi, etc.) that generate very personal and various data and metadata (location, time, temperature), that can have unexpected privacy impacts. For example, it has been shown that users can easily be identified and authenticated from smartphone-acquired motion signals, such as accelerometer and gyroscope (inertial) signals provided by most commercial smartphones [6]. Similarly, it has been demonstrated that mobile devices can sometimes be tracked from the capacity of their battery [7].

2. Personal device, always 'on'

Users often see a smartphone or a tablet as an extension of themselves, and tend to consider it as a trusted, very personal, device, which they will not share with anyone. Furthermore, these devices are almost always activated, carried around by their user almost everywhere and connected to a network. They usually store a lot of personal data for a long period of time. This makes them perfect targets for data brokers, advertisers or trackers in general, and can lead to pervasive and continuous monitoring of users. This is the notion of 'liquid surveillance', where the smallest details of our daily lives are tracked and traced [8]. Also, users are increasingly getting used to the possibility of voice control, supported by voice analysis agents such as Siri, Google Now, or Cortana. However, users are less aware of the fact that the voice control feature is realized by a device that is always listening in – at least to react on the defined control terms such as "Hey Siri", "Okay Google", or "Hey Cortana" – and therefore has access to all spoken communication[7].

3. Different types of identifiers

Mobile devices contain many different types of identifiers [9] (device hardware ID, stored files and metadata) or fingerprints (configuration [10] or behavioural fingerprints [11]) that can be used by mobile apps to fingerprint and track them [12]. For example, *De Monjoye and al*. showed that four spatio-temporal points, possibly obtained from a smartphone, are enough to fingerprint, i.e. to uniquely identify, 95% of

[7] https://www.computerworld.com/article/3106863/microsoft-windows/cortana-the-spy-in-windows-10.html.

the individuals [13]. Similarly, *Achara and al*. showed that any 4 apps installed by a user on his or her device are enough to fingerprint him or her with a probability of 95% [14]. Most of these identifiers, such as behavioural fingerprints, are often permanent and difficult (if not impossible) to reset.

4. Mobile and connected

Mobile devices can be geo-localized, and physically tracked. This characteristic can result in significant privacy damage. In fact, a lot of potentially sensitive personal information (such as religion, disease) could be inferred about an individual from his/her mobility trace [15]. Furthermore, since they are mobile, they connect to different, potentially malicious, networks which introduces new security and privacy risks [16].

5. Possibility of tracking

As shown previously, handheld devices can be physically tracked via their wireless interfaces by third-parties to generate physical “profiles”. They can also be tracked by third-parties on the Internet when they go online. Many third-parties are performing cross-domain tracking, i.e. to combine the physical and online profiles of mobile users [17]. This cross-domain tracking can provide a more complete view into user’s behaviour, and introduces new privacy and security risks. Cross-device tracking [18], [19], where third parties try to link together the devices that belong to a user, or cross-app tracking [14], where an app tries to identify/track the other installed apps on the device, are also expanding practices that introduce new and severe privacy concerns. For example, it was shown that user traits, such as religion, relationship status, spoken languages, countries of interest, and whether or not the user is a parent of small children, can be predicted from a subset of the list of his or her mobile apps [20].

6. Limited physical security

Handheld devices are often small physical devices, that are difficult to secure. They can easily be stolen or broken, which might have an impact on confidentiality, but also data availability. Furthermore, many risk sources (companions, spouses, relatives) can have physical access to them, their sensors or associated services.

7. Limited user interfaces

Handheld devices have usually limited User Interfaces (UI). This, of course, affects privacy, transparency, security[8]. For example, *Melicher et al.* showed that passwords created on mobile devices are weaker [21]. Privacy policies and notices are more difficult to read on a smartphone and require special attention. As a result, privacy policies should be built using a 'layered' approach where the most important points are summarized, with more detail easily available if the user wants to see it. Additionally, good graphical design, including use of colours and symbols, can help users understand better [22].

8. Limitations of app developers

Mobile apps are often developed by a single individual or a small group of persons, with limited resources and security/privacy expertise. It is therefore difficult for developers to adopt the latest privacy-protection technical solutions and measures.

[8] Still it is important to note that, despite the limitations, handheld devices such a smartphones do have a UI (e.g. in comparison with some IoT devices).

9. Use of third-party software

Most mobile apps are written by combining various functions, developed by other companies (and not the app developer). These third-party libraries help developers, for example, track user engagement (analytics), connect to social networks and generate revenues by displaying ads. However, in addition to the provided services, libraries may also collect personal data for their own use. The owners of the libraries can use this information to build detailed digital profiles of users by combining the data they collect from different mobile apps. For example, a user might give one app permission to collect his/her location, and another app access to his/her contacts. If both apps used the same third-party library, the library's developer could link these two pieces of data together [9]. Furthermore, these libraries are often proprietary and closed-source, and cannot be easily analysed. As a result, it is common that a mobile app developer does not fully understand what data these services are actually collecting [23]. While not a security risk as such, combining sources of data can lay the ground for an attack.

10. App market

Apps are often distributed via specific app markets (app stores), which may play a critical role in the security and privacy of apps. An app store usually does not only provide access to apps, but gives information on apps and collects and shows user ratings. Also, an app store may perform security checks on each provided app, in order to prevent distributing malicious or fake apps. Due to the important role for distributing apps, app store providers could (or, on the basis of government requests, be made to) filter out apps on featuring apparent security risks, however the filtering practice may also be grounded on market or political reasons or otherwise. As long as app stores remain unregulated, accessing their distribution capability by apps suppliers and developers will remain elusive and the criteria against which apps qualify may remain opaque. The availability of apps in app stores as well as the way they are presented may influence the distribution of the apps.

Moreover, from a privacy perspective it has to be noted that the knowledge level of a users' choice of an app could constitute personal data or even sensitive personal data, at times (e.g. if an ehealth app is installed, thereby revealing personal health preference or data). Currently users may not be sufficiently informed at all times about potential personal data collection by app store operators or providers of value added services they make use of, thus exposing themselves to cybersecurity risks.

11. Cloud storage

Mobile apps often store personal information in the cloud. This service should be secure and should protect against data leakage. As a matter of fact, it has been demonstrated that most quantified-self apps[9] exclusively store users' data in the cloud [24]. This introduces a new risk source and requires the user to trust the service provider without having regard to objective criteria upon which a trust decision can be based.

12. Online Social Networks

Many apps give the option to a user to share his or her data (aggregated or not) with other (selected) users for comparison or statistical purposes (as in a social network). This feature bears the risk of leaking personal data to other users and introduces new privacy and security risks, to be considered.

[9] See some examples of such apps in http://quantifiedself.com/guide/

2.2 Key legal and regulatory issues

The key legal and regulatory issues relevant to the processing of personal data by mobile apps stem from the GDPR [1], as well as certain aspects of the ePrivacy Directive [3] and the upcoming ePrivacy Regulation[10]. Our aim in this section is to identify and present selected legal and organisational considerations that pose challenges which may result in risks in mobile apps.

Since the emergence of smartphone application ecosystems, privacy regulators have addressed data protection and privacy risks that stem from the collection and use of personal data. The Article 29 Working Party adopted relevant opinions in 2011 on "Geolocation services on smart mobile devices" [25] and in 2013 on "Apps on smart devices" [26]**.** National Data Protection Authorities (DPAs) have their own initiatives, such as the United Kingdom's Information Commissioner's Office guidance for app developers [22] and the CNIL's project 'Mobilitics' together with technical researchers at Inria[11]. Of special relevance are the activities of the Global Privacy Enforcement Network (GPEN)[12], which in 2014 focused its global privacy sweep on mobile apps and regulators sent a joint open letter to app marketplaces (app stores) urging them to provide for links to privacy policies.[13] In the United States, the Federal Trade Commission published a relevant report on mobile privacy disclosures in 2013 [27], providing for recommendations for platforms, developers, ad networks, analytics companies and trade associations. The US National Telecommunications & Information Administration published a report in 2013 on mobile app transparency [28], including a Code of Conduct for mobile application short notices[14]. Finally, the California Attorney General published "Privacy on the Go: Recommendation for the Mobile Ecosystem" [29].

Notwithstanding the predominant focus of this report on the role of app developers, this section addresses in particular the processing of personal data by app providers. App providers are entities responsible to make available apps to users, and as such, usually qualify as *data controllers* under EU data protection law (see also discussion in section 2.2.2). An app provider can be the same entity as the app developer, but this is not necessarily the case at all times. Apart from app developers and providers, we also refer to other relevant entities in the app development ecosystem.

2.2.1 Personal data processed by apps

The application of the European data protection framework starts with the question of whether there is any processing of 'personal data'. The definition of personal data is broad, i.e. "any information relating to an identified or identifiable natural person ('data subject')" [1] [2]. The question whether a specific data type qualifies as personal data may require an elaborate analysis including a legal one, that can only be answered adequately when considering the specific context in which the processing of the information takes place. In

[10] In the rest of this document we refer exclusively to the European Commission's Proposal for an ePrivacy Regulation, which will finally replace the ePrivacy Directive. See relevant proposal at https://ec.europa.eu/digital-single-market/en/proposal-eprivacy-regulation. In the rest of this document we refer exclusively to the

[11] For more information, see: https://team.inria.fr/privatics/mobilitics/

[12] GPEN was established in 2010 by a number of privacy authorities, including European Data Protection Authorities and the Federal Trade Commission. The network now has 27 members, and seeks to promote co-operation in cross-border enforcement of privacy laws. Since 2013, GPEN has been co-ordinating global privacy sweeps. For more information, see: https://privacyenforcement.net/

[13] See: "Joint Open Letter to App Marketplaces", December 2014, available at https://autoriteitpersoonsgegevens.nl/sites/default/files/atoms/files/joint_open_letter_to_app_marketplaces_en.pdf

[14] For more information, see https://www.ntia.doc.gov/files/ntia/publications/july_25_code_draft.pdf

a mobile app environment when data is collected about or from a mobile device, the personal nature of mobile device usage implies that such data has to be considered as personal data as in the meaning of the GDPR. Thus, not only data on the device that is personal and private by nature, such as pictures, messages, emails, agenda items, etc. qualifies as personal data, but also data that is related to the device, such as device identifiers, environmental aspects, such as the device's location, and data related to its use, including logs containing usage data related to specific apps.

Once an app developer collects (and further processes) data on and from the device and its user, including metadata related to the device and the user's behaviour, all the key data protection requirements in the GDPR are triggered. If personal data is fully anonymised, the data protection framework does not apply, because anonymised personal data is indistinguishable from any other type of data.

Pseudonymous data is a new subset of personal data introduced in a legal sense in the GDPR. Under the GDPR, 'pseudonymisation' means "*the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person*". Notably, pseudonymous data is still personal data and the process of pseudonymisation is merely a measure that the GDPR incentivizes in view of its benefits for data privacy and security.

When accessing sensitive (special categories of) data, stricter requirements apply on the basis of Article 9 GDPR. Relevant special categories of data include data related to racial or ethnic origin, political opinions, religious or philosophical beliefs, genetic biometric and health data or data concerning a natural person's sex life or sexual orientation. When the use of an app results in the processing of such sensitive data, for instance in health or dating apps, typically, the controller has to ensure that there is explicit consent of the user for the processing of these data for particular specified purposes. It is possible that certain information processed in the mobile context, such as images, messages, user inputs, contain data that have to be qualified as sensitive data under Article 9 GDPR. Images are not generally considered sensitive data but images of persons can reveal their racial or ethnic origin, people's private messages can reveal people's beliefs and attitudes and health status. The same can be true for metadata, which collected over time can provide a uniquely invasive profile of users. While location data are not included in the list of special data categories, the processing of location data is generally considered to require special attention to the question of necessity and proportionality and major mobile OSs have implemented special mechanisms for location data transparency and user consent. In the case of over-the-top service providers of interpersonal communication services, the ePrivacy regulation proposal contains new rules for the lawfulness of the processing of content and metadata that are stricter than the general GDPR rules for personal data. The precise scope of the new rules is still unclear, including the question of whether the rules will only apply to communications in transit or also to communications in a stored format.[15]

It should also be noted that, like in US privacy law (COPPA)[16], the GDPR contains several stricter safeguards for the processing of children's data. When apps are specifically targeted at children and minors, most of

[15] For an in-depth discussion of the proposals, see Frederik Zuiderveen Borgesius, Joris van Hoboken, Ronan Fahy, Kristina Irion, and Max Rozendaal, 'An assessment of the Commission's Proposal on Privacy and Electronic Communications', Study for the the European Parliament LIBE Committee, 2017 [90].

[16] Children's Online Privacy Protection Rule, see: https://www.ftc.gov/enforcement/rules/rulemaking-regulatory-reform-proceedings/childrens-online-privacy-protection-rule

the obligations and issues discussed above gain additional legal weight and are likely to be enforced more strictly.

2.2.2 Different actors and their varying responsibilities

From a legal perspective, there are two key roles that imply specific data protection obligations, namely the data controller and the data processor. The data controller is the most important entity to consider as the data controller has to comply with the central legal obligations in the GDPR as regards the lawful, fair and transparent processing of personal data and the respect for data subject rights. Typically, the app provider will be qualified as the primary controller for the processing of personal data to the extent that the app processes users' personal data for its own purposes. In many cases, the app developer may be the same as the app provider, thus acting as data controller. This is not necessarily the case, however. For instance, when an organisation without any relevant expertise decides that it wants to offer its own app, it will likely contract for development outside of the organisation.[17]

In some cases, there might be more than one controller involved in the processing of personal data in the context of an app. This will be the case when an app integrates other data-driven functionality into the app, such as a third-party service provider for authentication of users or advertisement networks for monetization. In addition, the operating system may be gathering data when apps are used.

Furthermore, it is likely that one or more entities are involved that have to be qualified as data processors. Processors are entities that process personal data not for their own purposes but under the authority of the controller (e.g. the app provider). For instance, when app providers and app developers are different entities, it is possible that the app provider contracts the app developer for deploying the app and offering it in the marketplace. The app developer organisation would merely have a technical role, and the legal responsibility would remain on the app provider. Various forms of cloud services used by app developers will be considered data processors, if the service involves the processing of personal data. Under GDPR, data processors are also subject to specific obligations; however, most privacy guarantees still need to be implemented through the data controllers.

Under the GDPR, the producers of products, services and applications that are based on the processing of personal data or process personal data should be encouraged to take into account the right to data protection in their development and design processes (Recital 78 GDPR). Even though these entities may not be directly regulated under the GDPR, they are encouraged to make sure that controllers and processors are able to fulfil their data protection obligations. In the mobile context, this recital clearly resonates in relation to app developers, app stores, OS providers, library providers and hardware manufacturers. At least, app developers have to make the app in such a way that compliance with the GDPR is ensured. Another example is app stores, which can enforce reasonable guidelines for apps in terms of their security and privacy behaviour. OS providers can design the APIs, user experience and permission architectures in such a way that they facilitate compliance and trustworthy interactions between users and apps.

It should be noted that both GDPR and the proposed ePrivacy Regulation extend the obligations of data controllers to companies operating from outside of the EU, as long as they are targeting the European

[17] The EDPS has issued guidelines for the commissioning of apps by European institutions that provide a good example of how organisations contracting for development can ensure respect for data privacy, see [73].

market. They also require that apps operating from outside of the EU appoint a representative in the EU in view of their data protection obligations under EU law.[18]

2.2.3 Data protection principles

As discussed in section 2.1, mobile apps pose specific risks to security and privacy, both due to their nature, as well as the overall context of mobile app development. This is why the core data protection principles, as stipulated in Article 5 GDPR, can face serious challenges in the area of mobile apps. The core principles are as follows:

- '**Lawfulness, fairness and transparency**': Personal data shall be processed observing fairness and transparency towards the data subject and fulfilling the requirement of a legitimate ground for the processing of personal data (see section 2.2.4).
- **'Purpose limitation'**: When an app processes personal data, the app needs to have a specific lawful purpose for doing so and must inform the data subject accordingly. Further processing for other purposes is only allowed on the basis of a specific set of criteria in the GDPR (Article 6(4)). It is still common that application developers collect data on the basis of broadly construed general purpose, which is not sufficient to comply with the GDPR's obligations. So for instance, processing personal data "to provide the service and develop new features and future services" is too general to be able to comply with the legal requirement of purpose specification.
- **'Data minimisation'**: Personal data need to be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed. See Chapter 5 for a discussion of technical approaches to achieve data minimisation in practice, in the mobile app context.
- **'Accuracy'**: Personal data shall be accurate and, where necessary, kept up to date. In addition, every reasonable step must be taken to ensure that personal data that are inaccurate are erased or rectified without delay, having regard to the purposes for which they are processed.
- **'Storage limitation'**: Personal data shall be kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed. Personal data may be stored for longer periods insofar for public interest archiving purposes or for statistical purposes (Article 89 GDPR).
- **'Integrity and confidentiality'**: Personal data shall be processed in a manner that ensures appropriate security, including protection against unauthorised or unlawful processing and against accidental loss, destruction or damage. In view of this, data controllers shall implement appropriate technical or organisational measures.

2.2.4 Consent and other legitimate grounds for the processing of personal data

To process personal data via an app, the app provider or app developer (in their role as data controllers) need to have a legal basis. For private sector app providers, this legal basis stems from one or more of the following provisions:

- The processing is based on the **freely given and informed consent by the user.** The legal requirements for valid consent are stipulated in Article 7 GDPR. Notably, consent may not be valid if the application asks consent for the processing of personal data that is disproportionate. Furthermore, consent for the processing of personal data should not be mixed up with consent and or statements of agreement with other aspects of the service that is provided.

[18] See Dutch ruling with respect to WhatsApp end of 2016 in [30].

- The processing is **necessary for the service/functionality** that is provided to the user with the application. Specifically, Article 6(2) GDPR provides that processing is lawful if the "processing is necessary for the performance of a contract to which the data subject is party or in order to take steps at the request of the data subject prior to entering into a contract." For instance, a camera application will likely need to get access to the camera and be able to store images in the device's memory. It is important to note that Data Protection Authorities have interpreted the necessity criterion here objectively. In addition, when an app processes personal data for monetization purposes (e.g. for advertising), this is generally not considered strictly necessary and therefore needs to be based on a different legal ground.
- The processing of the personal data is "**necessary for the purposes of the legitimate interests pursued by the controller or by a third party**, except where such interests are overridden by the interests or rights and freedoms of the data subject which require protection of personal data, in particular where the data subject is a child". This final ground for processing of personal data involves a balancing exercise. When an app provider relies on this ground, it should be ready to clarify on what basis it determined that this balancing exercise provides a proper legal basis for its processing operations, including in individual cases. In practice, app providers may be able to use this legitimate ground for processing of personal data in view of security and marketing, as long as such processing operations are pursued fairly, transparently and proportionately.

As many apps will need to rely on users' consent for the processing of certain personal data, the requirement of consent deserves special attention, in particular as it relates to the issue of permissions (See Chapter 4 for a more detailed discussion on the matter). When accessing personal data from the device and/or using sensors, app providers likely want to rely on the permission architecture to get the consent of the app's users for the processing of relevant personal data.

Accessing personal data on the device can take place in different ways with different technical conditions attached to them: (i) using the permission architecture implemented by the OS, (ii) through direct interaction with the user, and/or (iii) by observing app and user behaviour (metadata). When using the available APIs in the OS, the app developer has to comply with the conditions (in the OS) for getting access to the specific data. Ideally, the permission architecture in the OS will contribute to compliance with the GDPR. For example, when the operating system allows access to personal data, such as location data, the OS would ideally provide for a mechanism that properly informs the user about the reasons for processing such data (transparency, purpose specification) and assists in the negotiation of permission to access such data with the user (consent). However, app developers should be aware that merely following the technical conditions to get access to data by the operating system and the contractual requirements to get accepted into the app store, is unlikely to be sufficient to bring them in compliance with the full range of obligations under the European data protection framework. First, the mechanisms that are implemented to negotiate permissions with users do not easily facilitate a proper explanation of the reasons for which access to certain data or sensors is required. However, the permission mechanisms, when used adequately, can be an important first step in complying with transparency and purpose specification principles. Second, in certain situations, the permissions lack the granularity to facilitate consent as permissions are grouped together in the OS. Finally, the permissions do not facilitate the negotiation of permissions for the functioning of the app and for any third-party functionality that is integrated into the app, which causes problems from a legal perspective.

App providers should be aware that mobile devices also tend to have personal data related to other individuals (than the owner of the device) stored on them, such as contact details, private communications and pictures. Therefore, app providers cannot simply rely on the potential willingness of their users to share third-party personal data with them for further processing. Since app providers do not have direct communication channels to the device owner, the processing of these third-party data can bring complications with respect to the legitimate grounds for its processing, the compliance with transparency obligations and data subject rights. As an example, in 2013 the Dutch DPA concluded that WhatsApp was violating the Dutch data protection act, by processing the contact details of third parties without their unambiguous consent [30].

Depending on the techniques used for the gathering of personal data from devices and about users by apps, such tracking can also bring apps into the scope of the legal framework of the so-called cookie provision in the ePrivacy Directive, which requires informed consent for such app behaviour. The new ePrivacy proposals are set to have an impact on this legal regime, including on the default settings in browsers and OSs. Article 10 of the European Commission's ePrivacy proposal stipulates requirements for browsers and other software permitting electronic communications "*to offer the option to prevent third parties from storing information on the terminal equipment of an end- user or processing information already stored on that equipment*". The provision covers mobile operating system software as well as apps that facilitate electronic communications inside of them. There is discussion on whether the requirement stipulated above should be brought in line with the GDPR's data protection by default obligation. This would mean that third-party data processing is not possible under the default settings and it would require further consideration of the user.[19]

2.2.5 Transparency and information

While transparency about the collection and further use of personal data is a key data protection requirement Articles 12-14 GDPR stipulate the types of information that should be disclosed to data subjects. Both the Article 29 Working Party and the Federal Trade Commission have concluded that transparency is a key challenge in mobile apps and provided specific guidance on the obligations for app providers to inform users e.g. in terms of a privacy policy etc. [26] [27].

In practice, app developers will rely on providing information through the following complimentary available channels:

- The app's privacy policy, which should contain the information that is required under the GDPR and be sufficiently accessible to data subjects.
- A link to the privacy policy of the app in the relevant app stores where users can review permissions and privacy statements. Both the Apple Store and Google Play store now require applications to provide a privacy policy and have stepped up enforcement of this requirement recently. In February 2017, Google announced that it would start removing apps that did not provide a privacy policy from the Google Play store [31]. More recently Google announced that it has developed machine learning tools to evaluate whether apps offering certain functionality are asking for too many permissions [32].
- Information through permissions that the app seeks to obtain from the user as regards the processing of personal data. Having proper information when permissions are asked, is important for users to be able to make informed decisions, but such information tends to be very minimal and cannot generally be a basis for fully complying with the transparency requirements under the GDPR.
- Information provided directly through the app in the context of user inputs and special notifications and explanations of the apps functionality in interaction with the user.

[19] For a detailed discussion on the ePrivacy proposals see in [90].

2.2.6 Data subjects' rights

The GDPR provides for similar data subject rights to access and to correct one's personal data and to object to the processing of one's personal data (in specific situations) as those rights that already exist under the Data Protection Directive. In addition, the GDPR provides for a more explicit right to erasure, also called the right to be forgotten (Article 17), the right to restriction of processing personal data (Article 18), a new right to data portability (Article 20) and a right not to be subject to automated decision making and profiling without appropriate safeguards (Article 22).

App providers need to provide proper information about the availability of data subject rights. Information about data subject rights shall be included in the privacy policy. In practice, app developers should ensure that the app architecture facilitates the exercise of data subject rights, including the right to access and correct personal data or to obtain erasure of personal data. The exercise of the right to erasure is of particular interest in the case of users deleting the apps from the mobile devices.

2.2.7 Data transfers and processing by third parties

As mentioned in section 2.1, mobile apps often rely on the integration of different types of third-party functionality, including measurement and developer tools and advertising, and the processing of personal data by such third parties as a result. A key issue in this regard is that there is lack of granularity as regards permission architecture in mobile Operating Systems (Oss), which raises the question of what is the responsibility of the app (first party) in ensuring the legality of the third parties' personal data processing.

European data protection regulators have interpreted the current rules as to imply that the app provider or developer (in the role of the data controller) has a legal responsibility in relation to the third-party data processing it helps to facilitate. The Court of Justice of the European Union is expected to rule on this question in the near future, which will likely result in further guidance on the precise responsibility of the first party to ensure that third parties are processing personal data lawfully, fairly and transparently.[20]

One of the key problems in the mobile environment is that the permission architectures do not provide for the possibility to grant permission to the app and integrated third parties separately. Sometimes, an app may therefore request access to a particular type of data on the device, only because a third party wants or needs to gain access to such data. At the least, the transparency obligations discussed above require that app developers are fully transparent about the third-party processing that the app facilitates and those processing operations need to have a sufficient legal basis.

2.2.8 Accountability, data protection by design and by default, security, and data protection impact assessments

The GDPR provides for a new requirement on data controllers to be able to demonstrate compliance with the GDPR and to assess the risks of personal data processing operations to the rights and freedoms of natural persons (Article 24). In practice, this means that data controllers, including application providers, need to have proper record keeping in place with respect to their personal data processing operations and conduct Data Protection Impact Assessments when required (principle of accountability)

[20] See here for a discussion of the reference to the CJEU, https://www.paulhastings.com/publications-items/details/?id=9f1aec69-2334-6428-811c-ff00004cbded

Data protection by design has been turned into an explicit obligation under the GDPR (Article 25(1)) and is an important obligation for app developers, as the design of the app in terms of the data flows it will generate can be scrutinized for using appropriate privacy by design strategies that minimize the impact on the rights and freedoms of the data subjects. Chapter 5 discusses in more detail privacy by design strategies and best practices in the area of mobile apps.

Data protection by default is another new obligation under the GDPR (Article 25(2)), highlighting the practical force and importance of the default privacy settings for the end-users' enjoyment of privacy. Whenever default settings are pre-configured, they have to be carefully chosen so that only personal data which are necessary for each specific purpose of the processing are processed.

Articles 32-34 GDPR stipulate the obligations on data controllers to provide for security of the processing of personal data as well as their obligations in the case of a data breach. The general obligation on app providers is to "*implement appropriate technical and organisational measures to ensure a level of security appropriate to the risk*". Controllers are expected to take into account "*the state of the art, the costs of implementation and the nature, scope, context and purposes of processing as well as the risk of varying likelihood and severity for the rights and freedoms of natural persons, the controller and the processor*".

Finally, a Data Protection Impact Assessment (DPIA) is required under Article 35 GDPR when processing of personal data is "*likely to result in a high risk to the rights and freedoms of natural persons*". In the following section we discuss in more detail privacy risk management and DPIAs in the context of mobile apps.

2.3 Privacy Risk Management

Privacy risk management is one of the major elements of GDPR and the ePrivacy legislation in general. In particular, Recital 75 GDPR spells out that risks to the rights and freedoms of natural persons may result from personal data processing which could lead to physical, material or non-material damage, e.g. discrimination, identity theft or fraud, financial loss, damage to the reputation, loss of confidentiality of personal data protected by professional secrecy, unauthorised reversal of pseudonymisation, or any other significant economic or social disadvantage. Also, Recital 75 explicitly mentions as a risk cases where data subjects might be deprived of their rights and freedoms or prevented from exercising control over their personal data or where personal aspects are evaluated in order to create or use personal profiles. Further examples are processing of sensitive personal data, of children's data or of large amounts of data or a large number of data subjects.

As shown in sections 2.1. and 2.2., mobile apps may hide risks (and accountability obligations), depending on the overall context of the processing of personal data. In fact, even a very small company, for example a single developer providing an app can trigger strict accountability obligations, in case of pervasive processing of large amounts or sensitive data. In that sense, privacy risk management is essential for mobile app providers and developers to understand the different privacy risks (as presented in section 2.1) that are specific to their processing of personal data and accordingly map them to the legal requirements of GDPR (as discussed in section 2.2). A possible example of such a mapping is shown in Table 2.1, where the core data protection principles are associated to relevant risks and risk factors and subsequent data protection requirements. This table is only indicative of how such a mapping could be performed and does not aim to provide an exhaustive analysis or methodology for privacy risk management.

GDPR PRINCIPLES	INDICATIVE PRIVACY RISKS	INDICATIVE REQUIREMENTS
Lawfulness, fairness and transparency Art.5(1)(a)	Unlawful, excessive and incorrect processing (e.g. due to permissions to unauthorised parties to access personal data through the app).	App providers/developers should make sure that they have a legal basis for the processing of personal data. App providers/developers should inform the data subjects properly about their data processing activities. This may help the users to understand what personal data is collected by them and why. App providers/developers should be aware of data subject rights such as rights to access, rectification, erasure, data portability. They should implement appropriate processes to support these rights.
Purpose limitation Art.5(1)(b)	Excessive collection and sharing of data (e.g. due to multiple sensors of mobile devices that are activated without need).	App providers/developers should use the data for a specific purpose that the data subjects have been made aware of and no other, without further consent. If the personal data is used for purposes other than the initial, they should be anonymised or the data subjects must be notified and their consent must be re-obtained.
Data minimisation Art.5(1)(c)	Excessive processing (e.g. due to use of third party libraries).	The minimum amount of data for specific processing should be processed by app providers/developers. For instance, they should not store the exact location point when a generic location area is sufficient for their app functionalities.
Accuracy Art.5(1)(d)	Outdated data pose identity theft risks.	Rectification processes into data management should be embedded in the app design.
Storage limitation Art.5(1)(e)	Undue data disclosure (e.g. due to cloud storage services used by mobile app developers).	Personal data must not be stored longer than necessary. App providers/developers should provide the "right to be forgotten" to the data subjects. This data must be kept only for a certain period of time for non-active users.
Integrity and confidentiality Art.5(1)(f)	Unlawful data processing, data loss, data breach, data destruction or damage .	App providers/developers should ensure that the security requirements of the personal data and the processing systems are met. This encompasses integrity and confidentiality as well as availability and resilience (Art. 35(1)(b) GDPR). For instance, the appropriate control access mechanisms should be embedded into the apps infrastructure in order to detect or monitor unauthorized access to the data.

Table 2.1: An indicative example of assessing risks with regard to GDPR compliance

The indicative privacy risks mentioned in Table 2.1 can infringe the rights and freedoms of natural persons (e.g. discrimination may stem from unlawful, excessive or incorrect data processing) and thereby have to be mitigated. In order to manage the risks to the rights and freedoms of natural persons, those risks have to be identified and treated. Unlike the classic approach of risk management, it would not be sufficient for substantial risks to ignore them or that data controllers try to escape their responsibility by covering such risks under insurance policies. Data Protection Impact Assessment (DPIA) is a tool that can greatly support this process.

2.3.1 Data Protection Impact Assessment (DPIA)

A DPIA involves identifying privacy threats resulting from the collection or processing of personal data. As opposed to regular security risk analysis that considers the risks for the organisation, a DPIA considers the privacy risks for the users of the service/application. Although the methodology might be similar, the perspectives are different. Providing security (confidentiality, integrity, availability) of personal data is, of course, essential to privacy but it is not sufficient. A DPIA should consider the risks resulting from the accidental or unlawful destruction, loss, alteration, unauthorized disclosure of, or access of personal data, but also the risks resulting from its collection by the data controller. In other words, a DPIA should also consider the data controller as a potential adversary or risk source.

GDPR poses for the first time an obligation to data controllers to perform a DPIA when the processing is "*likely to result in a high risk to the rights and freedoms of natural persons*" (Article 35(1)). Since mobile apps often collect personal, and often sensitive, data, a DPIA for processing performed through such apps will probably be required in several cases. When the DPIA concludes that the residual risks to the data subjects are high, despite mitigation measures, the data controller must notify the supervisory authority and obtain its view on how to proceed (Article 36 GDPR). The supervisory authority shall provide written advice and may use its powers as laid down in Article 58. The Article 29 Data Protection Working Party has recently defined a list of operations for which a DPIA is mandatory [33]

As already mentioned, in the area of apps data controllers are usually the app providers, which are not necessarily the developers of these apps. Although the legal obligation for DPIA lies only with the controllers, a DPIA of a software, library or piece of code can also be useful (and requested by the data controller) to assess their privacy impact. **Therefore, DPIAs can be essential to app developers to assess the risks of their tools and embed privacy and data protection requirements by design and by default.**

A DPIA should be carried out prior to the processing of the data, and start as early as possible since it can impact the system design. It is a continual process, not a one-time exercise that needs to be reviewed regularly and updated whenever there is a change in technology or data processing that might impact the risks. Although different DPIA methodologies have been proposed (see for example [34], [35], [36], [37] or [38]), most on them consist in defining the following common elements:

- **Respect of fundamental principles for data protection.** The data protection principles should be covered for ensuring compliance with the GDPR (see section 2.2.3). For example, in the context of mobile apps, app developers are responsible to demonstrate that the data processed are specific to the purpose and have the relevant documentation in place. Also, the app providers and developers have to make sure that the data subjects can exercise their right of access (Article 15 GDPR), the right to rectification (Article 16 GDPR) and the right to erasure (Article 17 GDPR).
- **Context description of the application/services/system**. A context is defined by its processing system and its purpose (description of the service architecture), the collected, generated and further processed personal data (defined by their attributes), the supporting assets and the risk sources (related to data controller, to data subject, to third parties or generic). Each risk source is also defined by its capabilities in terms of access to the data and assets, resource and background knowledge.
- **Feared events and threats evaluation**. A feared event defines an event that can impact the privacy of users. The event can result from inappropriate data collection or processing, in particular from accidental or unlawful destruction, loss, alteration, unauthorized disclosure of, or access of personal data. A feared event is defined by several attributes (motivation of the risk source, list of privacy impacts, list of associated threat, likelihood that a user might be affected by it) and performed by a specific risk source. It results from one or several threats (attack tree), which are defined by different attributes (feasibility/difficulty, likelihood, scale). Section 2.1 provides a discussion on privacy risks, risk factors and potential feared events in the case of mobile apps.
- **Risk Assessment**. The risk assessment is performed by computing for each feared event its likelihood and the severity of its potential impacts. In a "standard" security risk assessment process,

the risks are estimated based on their potential impacts to the organisation. In the case of a DPIA, the impacts are considered with regard to the freedoms and rights of individuals. This *"switches the analysis of impacts towards possible adverse effects that an individual may suffer, including for example identity theft or fraud, financial loss, physical or psychological harm, humiliation, damage to reputation or even threat to life*" [36]. Although the number of potential affected individuals is important to consider in the analysis, the impact might still be considered high even if the privacy of only a single person can be severely affected.

- **Risk Management**
  - Risk treatment: Once risks have been identified and analysed, they need to be addressed by considering technical or organisational options that may remove, minimize or mitigate them [36]. Technical options are various and may consist, for example, in minimizing data collection and processing (necessity and proportionality), implementing transparency and accountability measures (i.e., via privacy dashboards and privacy policies), providing control mechanisms to users over their personal data (consent). Other technical options may consist in adopting a privacy-by-design approach and using privacy enhancing technologies (such as data anonymisation). Finally, organisational measures, such as access control, or staff training should also be considered.
  - Risk acceptance and communication: Not all risks can always be completely mitigated. These residual risks need to be discussed and possibly accepted by a management decision. Furthermore, all involved stakeholders (including users) need to be informed about the counter-measures and accepted residual risks.

In the area of mobile apps, an **example case for DPIA** could be as follows:

- An example of *personal data* could be location data, defined by its accuracy and collection frequency (*attributes*).
- A *risk source* could be a third-party tracker, such as an advertiser, collecting this information without users' consent.
- An *associated feared* event could be that the third-party uses the location data for profiling and targeting, without the users' consent. This could easily be performed via a third-party library that the mobile app developer would use to get revenue. It could be argued that the *motivation* for the advertiser is high, the *difficulty of the attack* is low, and many users could be affected (the *scale* is large).
- The associated *privacy risk* is therefore significant. The *privacy impacts* for users could be manifold, such as a mere feeling of privacy invasion to more severe impacts, such as disclosure of private information that the individual wanted to keep confidential (e.g. pregnancy, health conditions, etc.).

In cases such as the one described above, since the privacy risk is significant, it needs to be managed. Some solutions could consist in asking users' consent, degrading the accuracy or frequency of the location information that the third party gets or using privacy-preserving solutions, such as geo-fencing [39]. Since the risk will probably not be completely mitigated, the mobile app provider or developer (or the data controller) could either decide to document it (in the DPIA) and to explicitly inform users of the risks and associated counter-measures in the mobile app's privacy policy, or not to include the third-party library in the code. In this case, it is questionable whether merely informing users would be sufficient if the app's stated purpose does not encompass personalized or targeted advertising: the legal obligations of data protection by design and by default (Article 25 GDPR) as well as the principle of data minimisation (Article 5(1)) demand to limit the processing of personal data to the necessary extent for the purpose.

3. Mobile application ecosystems and current development practices

Following the analysis of the privacy risks and key legal challenges, this Chapter explores the mobile app ecosystem and development practices with the aim of analysing the causes of these risks and challenges, defining relevant gaps and proposing potential solutions.

To organize concerns of production versus deployment, we describe the basics of the app development lifecycle and take a look at different depictions of mobile app ecosystems in the literature. While the ecosystem is complex, as mentioned earlier, we take an app developer centric approach, while also addressing app providers and other actors in the ecosystem. Specifically, we find it important to address:

- Who in the ecosystem is responsible for a privacy or security problem?
- Who has the capability to address the problem?
- Who is liable from a data protection perspective?

Aspects of software development are discussed as they can be leveraged as privacy action points. To this end, we take a look at idealized app lifecycles (data versus development lifecycles) and their potentials for implementing privacy by design.

With the generic terms app developers or developer teams, we refer to all parties involved in the technical development and deployment of an app. Our exploration seeks to address challenges independent of assumptions about how teams may be organized. We are aware that how software development is organized may be key to successfully engaging in privacy engineering activities, something we shortly touch on in the following pages.

Recent reports indicate that at the end of 2017, Android and iOS made up the operating systems of 99.6% of new smart phones sold [40]. While Windows phones continue to occupy almost half a percent of the world market, all other operating systems are sold in negligible numbers. Assuming that most developers will be developing for either Android or iOS, we focus on these two platforms in the study of the technical ecosystem.

While we try to generalize, we predominantly refer to the Android ecosystem. The insights we provide here are slanted towards Android, since the open source operating system makes it easier to execute studies using this environment[21]. Google Android and Apple iOS platforms do have divergences that can, for example, be traced back to the business models of the parent companies [41]. Google's business model is focused on providing free services that are also wired to collect and optimize data for their advertising business, whereas Apple's business model depends predominantly on the sale of physical devices. When possible, we mention details from the iOS platform and highlight such important differences. Nevertheless, the detailed application of some of the analysis in greater depth in iOS and other platforms should be a priority for future research and guidance in this field.

[21] We note, however, that formal recommendations for app developers should speak to the needs of developers working on either popular OS, or even less prominent OS like Windows Mobile.

3.1 Mobile app ecosystems

To address the complexity of the mobile apps ecosystem we take two different viewpoints: the development and the deployment view. The different views allow us to highlight the actors, stakeholders and processes relevant to addressing privacy and data protection compliance.

3.1.1 The App Ecosystem: Development view

3.1.1.1 Actors involved

A host of actors are involved in the development ecosystem of apps. These actors tend to be distinct from the actual development team (where in the context of agile development, for instance, each actor in a development team has its own specific role). Figure 3.1 provides an overview of these actors in the Android ecosystem [42]. Note that although the actors involved are derived from the Android ecosystem, these can be generalized to ecosystems of other market operators. Moreover, the figure presenting the development view only depicts app developers. In the rest of the text, we make a distinction between app providers and app developers (see also in 2.2.2). Since this Chapter focuses on actions that can be taken by the actual technical team engineering the software, we stick to the representation here and, when appropriate, mention responsibilities of app providers in the text. We assume that the responsibility to be compliant with the GDPR lies mainly on the shoulders of the app providers. We make no further presumptions about how to best organize the relationship and responsibilities between these two entities in order to fulfil requirements of the GDPR.

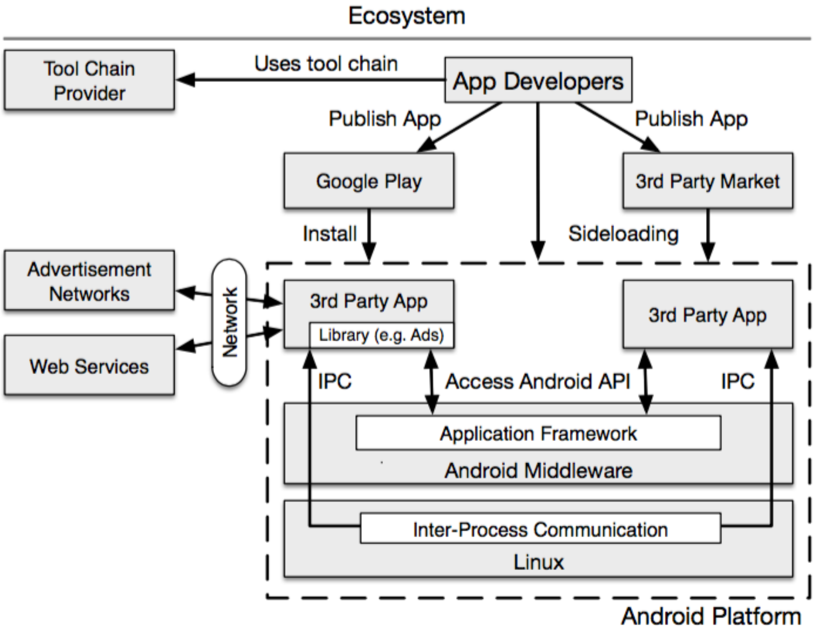


Figure 3.1: Android ecosystem [42]

The following actors are included in this view of the ecosystem:

- **Operating System (OS)/ platform providers.** These are the actors that develop the platform where the app is going to run/live. E.g.: Google provides the Android SDK/OS platform. These actors make an initial/basic decision on the security and privacy of the apps that are going to be built on top of it, for example, by determining which permissions require explicit user notification. There are inherent privacy and security threats that derive from the OS/platform providers [42]: (1) the permission model chosen by OS developers and offered in the mobile OS; (2) the way apps can access personal information and how such data is handled by the app itself; (3) and the app update model employed by the OS developers that sometimes grants permissions to apps based on general permissions enforced at the OS level (e.g.: when a user choses to not being asked all the time to set app permissions, the default ones are set instead).

- **Hardware/device manufacturers**. These actors provide and develop the hardware device where the app is going to be installed. In Android, they are allowed to implement some of their own security decisions. Usually these actors adapt and customize the OS/platform for different needs, which may have a substantial impact on the privacy decisions. While they have some independence, the manufacturers may still need to follow certain guidelines imposed by the OS/platform providers. E.g.: Most smartphones manufacturers that rely on Android OS have their own version of the OS tailored to their specific needs, but the manufacturer still needs to follow Google guidelines. Serious privacy and security concerns may arise when these vendors stop supporting older versions of devices and known vulnerabilities can be exploited at scale by malicious others.
- **App developers**. These are the actors that are responsible for the development of the app itself, i.e., for coding the app functionalities and requirements. They provide the app to the app providers or the end users, depending on the business model (see section 2.2.2). The app developers implement the apps using SDK APIs and libraries. The design decisions on privacy and security directly influence the compliance of the mobile app with GDPR. Moreover, misuse of the SDK APIs and libraries by the developers might have an impact on the protections implemented in the app.
- **Tool-chain providers**. These provide IDEs used to develop the mobile app (e.g.: Android SDK, Eclipse, etc.). These can directly influence the implementation of the privacy and security requirements by helping detect API misuse and fix weaknesses introduced by the SDK providers and device manufacturers. Ideally, the tool-chain providers could also provide Privacy APIs.
- **Library providers**. These parties build their own APIs to offer new features such as ad services or provide easier to use APIs that replace platform APIs. Library providers have the power to improve/break the security of the apps. Although their decisions cannot directly affect the platform security, they do affect the security and privacy of the mobile apps.
- **App publishers**. These are professional service providers that help developers publish their apps. When the app signing process is delegated to them, they have the ability to insert ads and, if they are malicious, inject malware into the apps. Therefore, they can negatively influence the compliance of the mobile app with regard to data protection.
- **App markets and stores**. These are the operators that provide a gateway to make apps available to the end user (e.g.: Google play, AppStore, etc.). They are responsible for managing the app markets (e.g.: AppStore or Google Play) and are able to find malware, bugs and unsafe code in the apps. Some app marketers, such as Google, validate, for instance, incorrect usage of SDK APIs.
- **End users**. These are the actors that have the ability to install and use the app on the mobile device. They can make use of privacy controls, like permissions, but these actions are limited by the privacy and security decisions made by all the other actors in the ecosystem. Misinformed end-users may install apps or enable permissions that access sensitive data in a way that is in violation of privacy or data protection requirements.

Our focus in this section is on app developers, and specifically on the ways in which they may be responsible for privacy problems and capable of taking steps to mitigate these. The complexity in the development view of the mobile app ecosystem demonstrates that app developers are one of many parties making design decisions that affect the collection and processing of personal data through a mobile app. When the organisation that provides the app is not the same as the one that develops it, there is interdependency between requirements and implementation in terms of GDPR compliance. However, even in this mesh of dependencies, the app developers have much discretion in design decisions, development processes and actual outcomes. Understanding these dependencies and the discretion available to app developers are key to providing usable recommendations to app developers and increasing the feasibility of applying security and privacy by design effectively.

3.1.1.2 Security and privacy in the development view

In [42] a list of research issues based on prominent security and privacy problems in the Android ecosystem is provided. These include:

- **Permission evolution and revolution**. The Android ecosystem does not provide a proper separation of privileges. As a result, third-party libraries inherit the privileges from the host app. Ad libraries tend to exploit these privileges and compromise the security and privacy of the hosting app (e.g.: by collecting users' personal data). At the same time, Android does not support permission models for anti-virus, where higher privileges are required. Although some researchers have proposed alternative access control models for Android, the platform still lacks support for more advanced access control mechanisms (e.g.: mandatory access control) which brings additional challenges to app developers when implementing privacy requirements.
- **Webification**. Integration of web content in mobiles apps through technologies as WebView are becoming a trend, due to the associated advantages to the developers (e.g.: app portability). However, the two-way interaction between the hosting app and the web content requires relaxation of the WebView sandboxing, leading to attacks that can be initiated either from the app or from the web content (e.g., the web content can use malicious JavaScript content to bridge the host app and perform elevation of privileges and leak personal data of the user).
- **Programming induced leaks**. Personal data leakage can also result from errors introduced by the development team due to, for instance, misuse of the platform APIs. A study [43] concluded that the APIs provided by the Android platform are too complex for the majority of the app developers. Moreover, it is not uncommon for the examples provided in programming forums to be insecure, leading to API misuse (e.g.: badly implemented cryptography). AndroidLeaks [44] and FlowDroid [45]are examples of static analysis tools proposed in the literature that try to detect privacy leaks in Android apps. However, many developers already struggle with the implementation of the basic functionalities of a given app. This leaves little room for privacy considerations.
- **Software distribution**. Android users can download the Android package file (APK) from the internet and install the app themselves, or select any app market that provides the app they desire. This decentralised way of distributing Android apps (that is prevented in iOS, for instance) can lead to malware installation on the users' device. Studies [46] show that malware introduced through mobile apps may collect users' personal data and send premium-rate SMS messages. Although, app markets find themselves in very powerful position with regards to the overall security and privacy of the provided apps (as these can control which apps are directly available to the end user), they do very little to address known privacy issues.
- **Vendor (Hardware and Operator) Customization and Fragmentation**. Studies [47] [have shown that vendor-specific customisations of the Android OS significantly increase the phone's attack surface and result in a large number of over-privileged apps. The fragmentation of the OS that derives from these customisations has a huge impact in the enforcement of privacy and security requirements.
- **Software Update Mechanisms.** The increased variability in the time taken by the manufactures and network operators to deliver OS security updates leads to a huge percentage of vulnerable devices

for an undefined time. Also, because some security updates are patched with other feature updates, users may sometimes have little incentive to immediately install them, as they may incur performance problems with their devices.

It is not untypical in security engineering papers for authors to equate privacy with security of personal data, and specifically with confidentiality requirements. This is also the case with the survey paper we used to depict the development view of the mobile app ecosystem. It is true that many of the security problems depicted in this survey, if fixed, would protect users against many types of privacy violations. However, many of the data protection challenges identified in Chapter 2 would not be covered at all. Prominent examples of such challenges are associated to purpose limitation, data minimisation, accountability, transparency and data subjects' rights. On the other hand, security evaluations in companies rarely consider the integration of privacy enhancing technologies (and the overall concepts of data protection by design and by default).

3.1.2 The App Ecosystem: Deployment view

In our second viewpoint on the app ecosystem, we take the analysis inspired by the review performed in the context of the proposed ePrivacy Regulation [48], that aims to identify the weaknesses of mobile apps regarding privacy regulations/recommendations. The focus is mainly on the interactions between the different components of a mobile infrastructure, thus emphasizing the deployment view of the app ecosystem, bringing forth other actors, information flows, and associated privacy issues.

3.1.2.1 Actors involved

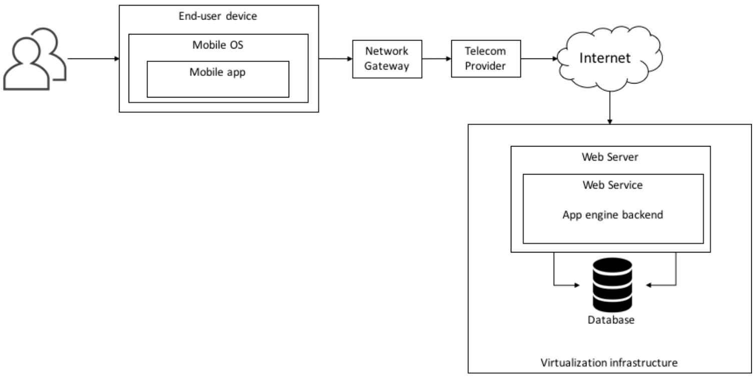


Figure 3.2: The deployment view on the app ecosystem

As shown in Figure 3.2, this view of the ecosystem includes the following main components[22]:

- **End-user device**. This corresponds to the mobile phone hosting the mobile OS and the mobile apps. The devices have capabilities such as wireless, mobile telecommunications technology (3G/4G/GSM), NFC, Bluetooth, camera, GPS, etc. The extensive number of the device capabilities and the fact that they can be used by the mobile apps (or even external devices directly communicating with the device) to collect even more personal information about the user, makes it hard to identify all the privacy issues inherent

[22] The device may connect to other IoT devices (e.g. smart-watch) that may interact with the user through an app installed in the mobile device or through an IoT Hub device. The analysis that extends beyond apps to IoT is not in the scope of this document.

to its usage. For instance, the privacy issues that can be introduced by the FaceID functionality recently introduced by Apple in the new devices are yet to be discovered[23].

- **Network gateway device**. This provides the interface for the different components, making the translation/mapping between different protocols and interconnecting the different networks. Network gateways, such as routers, sometimes placed at easily accessible locations, may include unpatched vulnerabilities or even might not be properly configured (e.g.: manufacture admin passwords left unchanged). All of these issues may enable the attackers to control the network gateways and intercept device communications. When apps are not designed with security and privacy in mind, this certainly leads to severe data protection violations.
- **Telecom service provider**. This provides access to telephone and related communications services. The telecom service provider has the ability to influence users' privacy as these entities are not always transparent regarding the information collected about the users and do not always provide the user the right to opt-out or data erasure.
- **Server**. The server corresponds to the application that hosts the backend of the mobile app (usually a web-service) providing the app engine. This application serves the purpose of providing the mobile app service, but also centralising the users' data at a specific location that can be used by other services that are related with the mobile app (e.g.: bank mobile apps can connect to the same backend engine of the online bank platform). In most of the cases the app providers are not clear about the information stored and for which purposes that information may be processed.
- **Database**. This corresponds to the data store used by the backend to store application data; this database may also store users' data, depending on the architecture of the underlying/connected app. In some cases, other services may directly connect to the database in order to collect data for different purposes (e.g.: ETL processes that integrate data for multiple applications). Most of these services do not make any distinction between application data and users' data, which may introduce several privacy and data protection violations.
- **Virtualization infrastructure**. This implements the virtualization mechanisms used to host the server that contains the app engine backend. This can be hosted/provided by a third party or by the app provider (e.g.: the server can be deployed in a cloud environment which relies on a software defined infrastructure to provide the instances where the server is running). Vulnerabilities in the virtualisation infrastructure, as well as having different entities (other than the app provider, which acts as the data controller) hosting the virtualised infrastructure (e.g.: public cloud provider), may bring up further privacy and security issues.

Without proper protection of the data in transit any of the actors described above are able to perform a man-in-the-middle attack. In the next section we summarize the possible privacy and security issues resultant from the interaction between the actors of the deployment view.

3.1.2.2 Security and privacy in the deployment view

All actors that are part of the deployment view may influence the enforcement of the privacy requirements by the mobile app. For instance, network gateways can be exploited by attackers that will redirect all the traffic exchanged between the mobile app and the backend engine to the attackers’ location. Moreover, attackers can use the network gateways to locate a certain user, using a certain app. Implementation of the data protection mechanisms may thus be a challenge for the app developers in this context. End-to-end security between the mobile app and the backend engine is one way to protect the content of the transactions, but nevertheless the app's metadata is always exposed to the intermediate actors that are part

[23] For more information, see: https://images.apple.com/business/docs/FaceID_Security_Guide.pdf

of the transaction. The telecom provider, for instance, can collect the metadata of the transactions between the app and the backend service. Given that the telecom provider knows which user owns the device, it can always link the metadata of the transactions to the users' identity. Even when these parties are co-operative, the sum of their activities may gravely impact any attempt to inform and provide subject access rights to data subjects.

It the review performed in the context of the proposed ePrivacy Regulation [48] it is stated that in the ecosystem as it stands today, users are given no free will with respect to the collection and processing of their personal data through their mobile phones. The characteristics inherent to the mobile app ecosystems (either due to the deployment models used or the development approaches followed) do not give any real choice to the user regarding the type of information that is collected and how it is going to be used. Identification of a new and more efficient mechanism that gives the control of personal data back to the user is therefore needed. To this end, some other propositions have been made to enforce privacy and data protection on mobile apps, such as:

- **Development of privacy aware mobile coding frameworks and operation systems**: specification and implementation of coding frameworks that easily enable enforcing privacy in mobile apps, as well as privacy aware mobile OSs that manage permissions and access to personal data in a privacy friendly way are still under development.
- **Development of privacy aware coding guidelines (for developers of mobile apps)**: there are no privacy specific coding guidelines that can help the developers to improve the privacy of the apps.
- **Enabling the certification and labelling of privacy compliant apps**: some entities (e.g.: ePrivacySeal) provide certifications for privacy compliant mobile apps. However, these certifications tend to be very expensive and not achievable by all mobile app providers.
- **Promoting the use of DPIAs:** Making DPIAs effective and mandatory (see also section 2.3).

Having the aforementioned considerations in mind, the next sections explore the development practices of apps and the way privacy requirements can be addressed.

3.2 Development of Apps: who, what, how?

3.2.1 Mobile app developers

Industry expects over 250 billion app downloads in 2017: this is a rapidly growing market[24]. But how do apps come to be? App development has a low threshold of entry, meaning almost anyone with some coding skills can string together an app. Apps are also developed by larger teams employed by multinational companies with support from management and legal teams. In addition to the in-house IT department, larger companies may outsource development of new apps to third parties, for some of which the development of apps are part of the "lift and shift" of an enterprise legacy system to the cloud. Apps are also developed by start-ups, which may be producing novel end-user facing functionality from scratch, and by individuals just tinkering at home. A recent study in the US [49] suggests that 82% of apps are developed by small companies (revenue less that $38 million and/or less than 250 employees).

Developers from any of these (non-)organisational backgrounds may develop apps, at times with high security and privacy requirements, and used by millions if not billions of users (e.g., Signal and Whatsapp

[24] For a number of mobile app downloads worldwide in 2016, 2017 and 2021 by region (in billions), see https://www.statista.com/statistics/266488/forecast-of-mobile-app-downloads/

come from two very different organisations, partially share a code-base, and have overlapping privacy and security requirements).

Industry estimates suggest that, globally there are 8.7 million mobile app developers in the world. Geographically, app developers are distributed across the globe, with approximately a third situated in Europe. On average, developers are around their late 20s and early 30s[25] with less than 10% expected to be women.

Data protection compliance for someone who is brewing apps in their kitchen may differ from that of a larger organisation with potential support from a legal or technical team with data protection expertise. An individual developer may neither have the time, money, nor the expertise to wrap their heads around complicated legal concepts, yet they may be the prime target for data protection recommendations.

In organisations, data protection requirements are likely to transform the relationship between management, the legal or compliance office and development teams. Typically, organisations have treated data protection as a legal matter. Organisations may be shy of spending funds beyond what it costs to have a legal team to address compliance issues. They may also not be aware of technical approaches. Concrete technical recommendations may help companies conceive what is required by data protection by design. Similarly, understandable examples of privacy by design approaches and solutions may aid developers to convince the legal team and management that data protection is not just a legal matter. The cost of implementing and maintaining technical mechanisms for data protection compliance may also incentivize a preference for legal solutions.

Our search returned no studies that surveyed types of app developers and their working conditions that would be useful with regard to data protection compliance for app developers. For the sake of the discussion that follows, we imagine a mobile app development team in a small sized company.

3.2.2 Types of apps

A mobile app is a self-contained program, running on a mobile device that has limited functionality. There are different ways of implementing mobile apps, as shown in the following:

- **Native apps**. These apps are installed through an app marketplace (app store), for instance, Google play, and developed for a specific platform (e.g.: Android, Windows phone or iOS). Apps of this kind can make use of all device capabilities, including the GPS, accelerometer, camera, the notification system, etc. and
- **Web apps**. The web apps usually run on a browser and are usually written in HTML5. They can be accessed in the same way as the web pages on the browser and do not rely on app stores, therefore they are free from any type of updates triggered at the user device. Some of these apps have the option to add an icon to the home screen by creating a bookmark to the page.
- **Hybrid apps**. These apps result from a combination of a native app and a web app. They can run inside a native container and leverage the devices' browsing engine and include a web-to-native abstraction layer that enables access to the device capabilities. Given their flexibility, these apps allow cross-platform development (not targeting any specific device platform) and are usually built as wrappers for existing web-pages.

[25] Statistics averaged from the statistics portal Statista https://www.statista.com/statistics/256629/share-of-android-app-developers-worldwide-by-country/ and from the blog article “7 Surprising Statistics about the World of App Development” by Simon Lee published at https://thisisglance.com/7-surprising-statistics-about-the-world-of-app-development/

The typology of the app may directly influence how the privacy and security requirements are implemented. Previous studies have shown that currently apps are more likely to send requests to tracking domains than mobile web browsers and can easily obtain data points through normal permissions that can be used to fingerprint devices [50]. Moreover, there is an increasing number of apps that are automatically generated. Further studies on the privacy and data protection implications of different kinds of apps will provide an important basis for developing precise recommendations for app developers.

3.2.3 Pricing model

Mobile apps have different types of pricing models. Some apps require on-time payment for their usage, others have a revenue stream through ads (developers collaborate with ad networks, display ads on the app, and depending on views and clicks receive shares) and yet others rely on in-app purchases (paying for additional features). Currently, games are the primary revenue generator (with around 80% of the $88 billion revenue in 2016) with other domains growing more rapidly (social networks, messaging and sports), and Asian markets becoming the leader in generating income[26].

Regardless of the selected pricing model, app developers and app providers are required to be transparent about how personal data is handled by the app. Those apps that rely on advertisement revenue are more likely to integrate ad libraries which tend to hoard user data and, when possible, link this data across multiple apps that a user has installed. App developers may not always be aware or transparent about information flowing to third parties through the integration of third party libraries that include ads.

3.3 Addressing privacy in different lifecycles: data versus development lifecycle

At first sight, it seems that data protection requirements better align with the data lifecycle. Predominantly, provisions of personal data protection engage with what data is collected, justification and transparency of such collection, processing and retention. This parallel is also often reflected in privacy research that is concerned with data protection compliance.

In the technical literature, there are more or less detailed versions of data-lifecycles. We rely here on the data-lifecycle considered by [41] to map the different privacy decisions (by users and developers) associated with each (non-linear) phase:

- Conception phase: decisions made during platform creation, app store creation, phone manufacturing, and app development.
- Creation phase: decisions that are made at the time data are first generated and/or captured by the smartphone
- Transmission phase: decisions made about how and with what protections data can be transmitted to service providers, storage, or shared with other users or companies.
- Storage phase: decisions made about data at rest in a repository (e.g., where the data will be stored, expiration and access control)
- Exploitation phase: decisions made with respect to how the data is analysed and used (also in fusion with other data sets). Exploitation may generate new data which initiates further data lifecycles.
- Backup and deletion phase: Either backing up data or removing it according to different retention requirements.

[26] For more information, see App Revenue 2017, http://www.businessofapps.com/data/app-revenues/

However, this list of phases contrasts with the different activities of the (secure) development-lifecycle: data-lifecycle decisions may impact development activities and vice versa, however they are not one and the same thing. Keeping track of where data flows is an essential part of the transparency requirements for data protection. Yet, data lifecycle management supports but does not determine how the app is developed, what functionalities are prioritized, how data is leveraged to fulfil these functional requirements, and how the different privacy and security requirements related to data protection are implemented or evaluated. Hence, we argue, that there is a missing step of relating data protection requirements to how the software is actually developed. Better connecting the two parts, the development and data lifecycle, by understanding how development decisions affect the data lifecycle may also be pertinent. In this section we try to address these gaps.

3.3.1 Addressing data protection by design through the development lifecycle

As already mentioned, data protection by design and by default are legal obligations for data controllers under GDPR (see section 2.2.8). However, critics argue that GDPR lacks clear guidance on the parameters and methodologies for achieving data protection by design and by default [51]. From the point of view of app developers, GDPR in fact provides little guidance on how the different development activities in the development lifecycle can contribute to fulfilling these and other requirements.

There are few studies that explicitly consider what privacy activities may be appropriate in the different phases of the development life-cycle (see e.g. in: [52], [53], [54], [55], [56]). These studies show activities in different phases of the development lifecycle which can be pertinent to integrating existing privacy solutions into mobile apps. For example, data minimisation can be achieved by not collecting unnecessary data, a data-lifecycle decision. But, it is also possible to minimize data by using cryptographic protocols and decentralized architectures. These technical mechanisms for data minimisation (from now on "computational data minimisation") can make it possible to collect the data necessary for the desired functionality without having the data flow to the service providers, something a data-lifecycle may or may not reflect. In fact, we posit that most computational data minimisation mechanisms are better addressed in the development-lifecycle, e.g., requirements, design, implementation, testing and evaluation. While the methods proposed in these papers contribute to addressing privacy as part of the development lifecycle, they abstract away from the development methodology used. This shows that further research is needed on the feasibility of applying these methods in practice and specifically an agile environment.

Researchers have also proposed Privacy Enhancing Technologies (PETs) with strong privacy protections, e.g., homomorphic encryption, private information retrieval (PIR), or multi-party computation [4]. While PETs do not provide methodological guidance, some of the PETs can be turned into development tools or even services. As it currently stands, mechanisms introduced in PETs are hard to implement by app developers without crypto expertise. Wide scale adoption in agile environments requires at least the development of standards, libraries, and usable APIs, as well as research on how to engineer such systems in an efficient and scalable manner.

In the mobile app ecosystem, a data lifecycle internal to an organisation may not be sufficient to capture the cross-service data-lifecycle. The move from shrink-wrap software to services has allowed for increased modularity in software development [57]. This means developers can easily integrate functionality from third parties, instead of developing their own functionality from the ground up. In the case of app developers, modularity manifests itself in the integration of third-party libraries, use of IDE and SDK's provided as services, etc. By including code and other resources from third parties in their app, the developers co-produce a cross-service data-lifecycle. No single organisation has overview over this cross-service data-lifecycle, especially since third-parties may include further third parties, making transparency a challenge. Contractual agreements may address some of the challenges here, however, they are better combined with technical mechanisms that provide either privacy guarantees or techniques to enforce and verify compliance with privacy policies.

A number of mechanisms for enforcing data protection policies across services have also been proposed [58], [59]. However, enforcement and verification require oversight. Potentially, some enforcement across apps can be done by platform providers. This may be considered a win: platforms like Apple, Google or Facebook can play a key role in verifying data protection compliance across the app ecosystem. This, however, also has a direct influence on the political economy of platforms and developers, and how much control the prior can assert on the latter. How regulatory power and data protection responsibilities can be balanced in the case of platforms is a serious issue and beyond the scope of this document.

3.3.2 Mobile app development phases

In the previous section, we discussed that existing recommendations for app developers do not distinguish between lifecycles. However, they also do not address the different development roles and responsibilities. Especially in larger development teams, it is necessary to translate privacy and security decisions made with respect to the data-lifecycle into development activities and vice versa.

In this section, we focus on the literature on developing software applications. Software engineering approaches work in distinct phases, usually described as part of the development process lifecycle, and follow a specific methodology. For instance, Xamarin[27] mobile app specifies development lifecycle phases that can be summarised in five different steps:

- Inception: this is mainly the idea that leads to starting building the app.
- Design: defining the user experience and the app functional and non-functional.
- Development: corresponds to the actually building of the app and associated testing.
- Stabilization: after testing and quality assurance (QA), the app goes into a special phase (e.g.: beta phase) where the user can provide feedback and suggest changes.
- Deployment: the app is distributed by the market operators (e.g.: Google Play, Apple store, etc.).

The difference between these development lifecycle approaches does not lie so much in the phases, but more on how these phases are tackled during the development of the application. For instance, in a waterfall methodology development approach, the phases are followed sequentially while in an Agile approach an iterative development methodology is addressed. Agile software development has been gaining more and more supporters and therefore adopted as the default approach for most of mobile developers. We discuss this approach next.

3.3.3 The Agile software development approach and associated roles

The Agile development workflow is depicted in Figure 3.3[28]. In a nutshell, the agile process defines a set of values and principles for developing software. Three of the most widely used Agile frameworks are Scrum, Lean and Kanban[29] software development methodologies. These rely on development cycles, called sprints,

[27] https://developer.xamarin.com/guides/cross-platform/getting_started/requirements
[28] https://commons.wikimedia.org/wiki/File:Scrum_process.svg#/media/File:Scrum_process.svg
[29] See for example in: https://leankit.com/learn/kanban/kanban-software/

which correspond to basic units of development, time-boxed within specific period of time. The application requirements are maintained in what is called the product backlog. The backlog is prioritized with the help of the product owner (see below), and contains not just the functional requirements, but also bug fixes and non-functional requirements. Items added to the backlog are written in story format, called user stories.

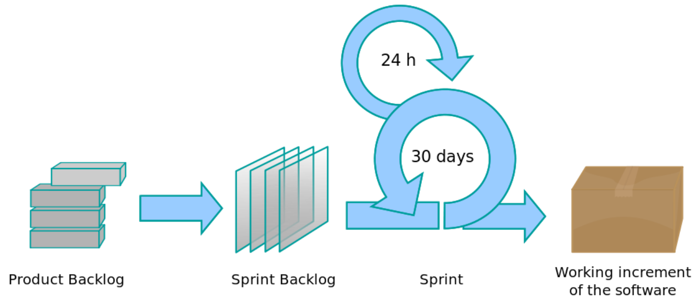


Figure 3.3: Scrum development process

When following an Agile development methodology, the app developer may take one or more of the following roles[30]:

- Product owner: represents the stakeholders and is responsible for ensuring that the team delivers value to business; maintains and prioritizes the project backlog.
- Application and solution architects: lead or technical manager that is specialized in the application built and technologies used; responsible for translating requirements into the architecture for that solution; are mostly involved in the design phase of the project.
- Scrum master or team lead: acts as a buffer between the team and the stakeholders.
- Team member or developer: responsible for building the app.
- Tester (independent or part of the team): validates that the development team has implemented the requirements initially defined.
- Project manager: responsible for defining a strategy, roadmap and feature definition for the mobile app.

This list is only representative and not exhaustive. There are further actors involved in a mobile app development, such as the stakeholders or business analyst, etc., but these are not always directly involved in building the app, or do not always actively participate in the definition of the app requirements and functionalities. Increasingly important is the role of UX designers[31] that can play a key role in interfacing privacy related activities, controls and transparency efforts to end users.

All these actors can play an important role in the enforcement of privacy and security requirements. Notice that when considering the GDPR, for instance, the responsibilities of each role are not well defined. However, some general recommendations that can be considered are as follows:

- The project owner needs to be responsible for making sure that the privacy and security requirements take priority in the backlog and that they are enforced in the definition of the user stories.

[30] Of course different roles may be assigned to different persons within the same organisation (app developer).

[31] User experience (UX) design is the process of enhancing user satisfaction with a product by paying particular attention to issues such as usability, accessibility and overall pleasure of user interaction. For more information, see: https://www.interaction-design.org/literature/topics/ux-design

- The testers must be responsible to test that the privacy and security requirements have been properly implemented and any control or transparency efforts are usable.
- The developers or team members must implement the pre-defined privacy and security requirements and make sure that the libraries used in the app development are privacy-compliant.
- The scrum master needs to guarantee that the privacy and security requirements are being implemented through the sprints, so that upon each sprint the app that is ready to be released is privacy compliant.
- The application and solution architect need to take into account the privacy and security requirements and validate if the proposed architecture is actually compliant.

Beware that when the team demonstrates that it does not have the expertise to correctly and efficiently implement privacy requirements, it is still required that some privacy expertise is injected anyway. It is the responsibility of the project manager to make sure that the team has the correct means and tools to develop a compliant app.

The challenge that most teams face is on how to integrate all these recommendations into the development lifecycle of the app. In the following sections we address this issue in the context of Agile development, by considering the secure development lifecycle approach proposed by Microsoft and providing some suggestions on how to extend it to privacy and data protection.

3.3.4 Agile Secure Development Lifecycle

The Agile Secure Development Lifecycle (SDL)[32] is a framework proposed by Microsoft to address the security practices in the context of agile development. Since it is used in practice, we propose SDL as a productive starting point to explore development methodologies for data protection compliant mobile apps. SDL extends the Microsoft Secure Development Lifecycle approach, depicted in Figure 3.4, to the specific Agile development workflow.

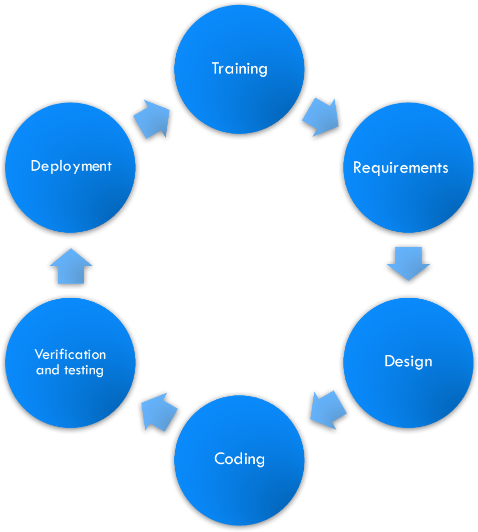
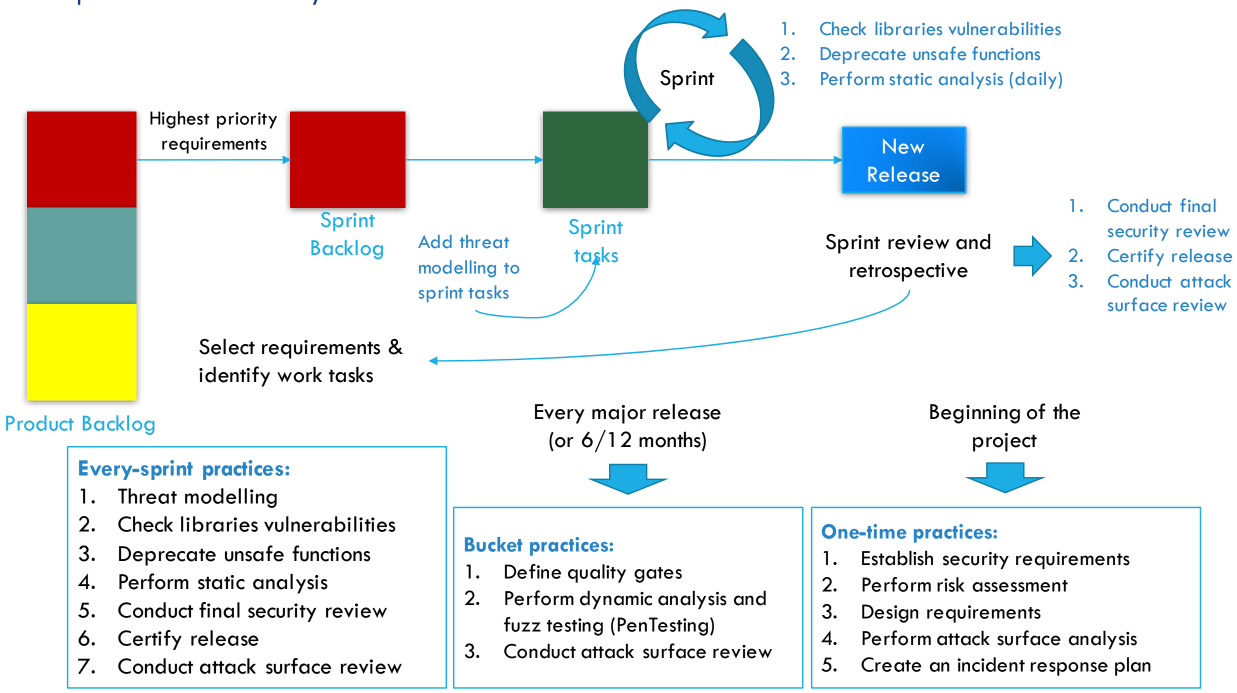


Figure 3.4: Secure Development Lifecycle

[32] For more information, see: https://www.microsoft.com/en-us/SDL/Discover/sdlagile.aspx

The Agile SDL places the security practices into three main categories or buckets[33] (Figure 3.5):

- One-time practices done before starting development of the system
- Every-sprint practices that are performed on every sprint
- Bucket practices done every 3-6 months



Sprint
1. Check libraries vulnerabilities
2. Deprecate unsafe functions
3. Perform static analysis (daily)
Highest priority requirements
New Release
Sprint Backlog
Add threat modelling to sprint tasks
Sprint tasks
Sprint review and retrospective
1. Conduct final security review
2. Certify release
3. Conduct attack surface review
Select requirements & identify work tasks
Product Backlog
Every major release (or 6/12 months)
Beginning of the project

Every-sprint practices:
1. Threat modelling
2. Check libraries vulnerabilities
3. Deprecate unsafe functions
4. Perform static analysis
5. Conduct final security review
6. Certify release
7. Conduct attack surface review

Bucket practices:
1. Define quality gates
2. Perform dynamic analysis and fuzz testing (PenTesting)
3. Conduct attack surface review

One-time practices:
1. Establish security requirements
2. Perform risk assessment
3. Design requirements
4. Perform attack surface analysis
5. Create an incident response plan

Figure 3.5: Agile Secure Development Lifecycle (SDL)

Whilst some may consider security and privacy requirements orthogonal to the functional requirements, this is not always the case. For instance, a mobile app that gathers weather information of different end-points and shows it to the end user may not need to encrypt the data in transit (integrity is important), while an e-banking app has different security requirements on data transfer.

3.3.5 Extending the Agile SDL to address privacy requirements

Technical enforcement of privacy requirements is in practice not that different from enforcing security. Functional requirements that GDPR may deem intrusive or that initiate some unacceptable personal data collection need to be designed and implemented in a compliant manner. Still, no development practices that integrate privacy by design into the Agile development philosophy have been proposed or implemented yet.

Any proposals for integrating privacy into existing security processes may, however, have to take the incentive structures into account: over time security has become a concern for the app developer (the developers pay if they don't do it (right)), privacy however remains a concern for the data subject (the data subject pays if the app developer doesn't do it (right)). The latter is expected to shift with the implementation of the GDPR.

As already mentioned, SDL is focused on security. Apart from the requirements and design phase, many of the security activities focus on things that should not happen, i.e., if an attacker can manipulate existing functionality to do unintended and undesirable things. To expand SDL to include privacy controls requires

[33] https://www.microsoft.com/en-us/SDL/Discover/sdlagile.aspx

defining requirements from a data protection perspective. This includes things that should not happen (e.g., linkability of users actions across apps or devices) but also things that should happen (e.g., transparency and intervenability as discussed in Chapter 5).

LINDDUN and "abuser stories" are examples of methodologies and techniques that take this approach and that may be reasonable to include in recommendations to app developers. Further ideas on how to extend Agile SDL to include the privacy requirements on top of the security ones are as follows:

1. Privacy quality gates

Definition of the privacy-enforcing quality gates will enable the development of privacy-aware apps. Quality gates are important to define the criteria to which the application should adhere to, in order to pass the quality checks. Therefore, the specification of design patterns (see Chapter 5) and a privacy-aware architecture are the first steps in implementing a privacy-compliant mobile app. Moreover, these steps can also include the definition of the privacy policies and associated defaults, so that they can be implemented and validated later on. It should be noted that mobile app development, as well as the maintenance of the app itself, is often outsourced. When the data controller (app provider/developer) does not have quality gates in place to enforce privacy compliance, it becomes very challenging to identify the privacy issues.

2. Privacy requirements

In the agile development context, the requirements are usually defined by the team members, and because of the agile free-format, there are no clear roles on who should be assigned to perform a particular task of the secure development lifecycle. Therefore, a specific team member must be responsible to define the privacy requirements. The implementation and testing of those requirements is normally assigned to the developers and testers, respectively.

In the case of data protection and privacy, high level goals can be derived from the GDPR. This can be executed by the app provider in collaboration with app developers. However, the challenge is in refining these high-level goals into privacy requirements. Executing a DPIA (see section 2.3) fulfils one of these high-level goals, and can also be used to determine the level of privacy protection that needs to be implemented in the specific context of the developed app. When specifically considering mobile apps, data protection requirements should be refined by the different development team members, taking into account the specific functionalities of the application and its intended use. The lack of general knowledge of more refined privacy requirements and architecture design patterns for privacy (some of which are discussed in Chapter 5), as is the case for security, makes this a very challenging task that not all mobile app developers may initially consider.

3. Privacy risk assessment

After defining the privacy requirements for the mobile app one has to derive the risks in case, for instance, of data leakage. DPIAs can also be useful here to assess the privacy risks of mobile apps (see section 2.3).

4. Attack surface analysis

The analysis of the attack surface of the application enables describing all the points where an attacker could get into the application and leak personal data. Therefore, a suggestion of the type of questions one could ask/answer in order to perform the analysis of the attack surface are:

- What personal data is going to be collected by the application?
- How is the personal data transferred and where to?
- Can any information be deduced when personal data is transferred to third parties? Can that transfer and the information lead to data protection or privacy violations?
- How is the data stored: locally or in the cloud? If in the cloud, is the cloud provider trusted to store the personal data or is the data protected? Who has access to the data stores?
- Is the maintenance of the system outsourced to an external party? What is the impact of that if sensitive data is leaked?

5. Privacy threat modelling

Threat modelling is a process that enables the identification of application threats from the attackers' point of view. LINDDUN is a framework that provides a systematic way of deriving privacy related threats, therefore could be used as a methodology to derive privacy threats in mobile apps. Whether and how it works in an agile environment is a question for future research.

6. Analysis of third party libraries

As already mentioned, third party libraries can introduce new privacy issues in the apps. Evaluation of the third party library or framework used to implement the mobile app is therefore of utmost importance in this context. Although there is no standard way to perform this type of assessment, some possible questions that one could try to answer when evaluating a third party dependency are as follows:

- Which information is going to be transferred to the third party components?
- Is there any personal information that is going to be transferred to the third party that needs to be part of the DPIA analysis and following requirements?
- What are the default permissions defined by third party component? Are they data protection compliant?
- Does the usage of the third party component imply that personal information is going to be transferred from the mobile app to an external third party?

Since many app developers will be using the same third-party libraries, it would be reasonable to consider making their DPIAs and data protection compliance activities public especially so that app developers do not have to repeat the same cumbersome process of assessing third party libraries repeatedly. This is a point in the app ecosystem where certification can also play a constructive role.

7. Deprecate privacy-invasive constructions

When performing secure coding, deprecating unsafe constructions helps reduce the security associated bugs with a very small engineering cost. This kind of approach can also help the developers when considering privacy. For instance, if the developers are aware of the possible constructions that can cause data leakage or that may store sensitive data that should not be stored, they should deprecate and mark them as privacy-invasive constructions. It should be noted that the deprecation of privacy-invasive constructions is known to be in its infancy, due to the lack of knowledge and research in this specific area.

8. Static and dynamic analysis for privacy

Although the static analysis tools available in the market provide low insight on the privacy issues, enhancing the existing tools to enable the detection of privacy problems would provide a huge help to the development of privacy-aware mobile apps. The development team would benefit even more from the utilization of such tools when these would be integrated in the development pipeline and steered by the pre-defined quality gates (e.g.: mobile app can only be distributed if no high-risk privacy related issues are found by the tool).

Over the last years, a number of research projects have provided tools for end users and developers of mobile apps. For instance, TaintDroid [60], AppFence [61], AppIntent [62] dynamically evaluate the information flows of the Android applications by trying to detect privacy-sensitive data and exposing potential privacy leaks; Saint [63], Aurasium [64] and Apex [65] enable specification of run-time constraints on privacy-sensitive data; FlowDroid [45], ComDroid [66] are static analysis tools for android apps that try to detect potential privacy sensitive data leaks and TrustDroid [67] provides a framework to isolate applications that are divided into corporate and private applications. Such tools, if implemented well, may make it easier for developers to integrate privacy activities into the development lifecycle.

9. Validation or verification

Validating that the privacy requirements initially specified are fully implemented is one of the most important steps of the development lifecycle. Not testing if the requirements are implemented increases the risk of having privacy issues when releasing the application. Therefore, the creation of unit and regression tests, for instance, to validate privacy requirements may help reduce the risk of data breaches. Moreover, performing privacy-aware code reviews, validating that the privacy policies are met by the app and that the defaults of the privacy policies are, for instance, GDPR compliant, also helps understand the status of the mobile app and fix the issues before releasing it to the market.

Addressing privacy by design throughout the development lifecycle could greatly enhance data protection compliance of mobile apps. Extending agile SDL to include privacy requirements could be an interesting area to explore further. Based on the aforementioned ideas, Chapter 4 looks at challenges to providing recommendations that may be useful for developers building apps for the mobile app ecosystem using the example of app permissions. Chapter 5 builds on the concept of privacy and data protection by design, aiming at bringing forward technical requirements that correspond to legal privacy provisions (privacy goals and design strategies).

4. Data protection challenges in the mobile app ecosystem: the example of permissions

Following the discussion in Chapter 3 on the app ecosystems and the difficulties in addressing privacy requirements into the development process, in this Chapter we take a deep dive into analysing permissions, a key data protection challenge in mobile apps. As already discussed in Chapter 2, the management of permissions is directly related to consent management and is one the most critical aspects regarding legitimacy of a data processing operation in the mobile app context. Our analysis aims to highlight the ecosystem-related problems that are affiliated with this data protection challenge, as well as to propose relevant ways forward.

4.1 Permission models and impact on privacy

Permission models differ depending on the operating system and device. The Android architecture distinguishes between *normal* and *dangerous* permissions. “Dangerous permissions” are those that Google has decided may pose a risk to a user’s privacy or the device’s operations (e.g., phone, location, sensors), compared to “normal permissions” (e.g., access the Internet, vibration, setting time zone). In contrast to normal permissions, dangerous permissions require user approval prior to installation or first use, and may be revoked at any time afterwards.

Android offers a centralized interface for privacy settings, as well as the possibility to change permissions for a specific app. Furthermore, the operating system allows users to decide on default apps (e.g., for sending SMS) for which users may tailor the permissions to fit their needs. A permission explanation dialog allows developers to provide users with more information about a permission. However, studies show that the usability of these explanations can still be improved [68].

IOS, on the other hand, uses “entitlements” and “permissions”. Entitlements are set in an application’s configuration and define capabilities unavailable by default and necessary for the app to function (e.g., iCloud access). Entitlements are submitted to Apple in the app bundle and cannot be modified after the application is submitted to the App Store. Permissions in iOS are only approved at run-time and are used to prompt the user for the use of restricted resources, where access is only granted if the user agrees. Agreement to a permission means access to the same resource is subsequently approved. However, users may revoke permissions at any time using iOS’s privacy and security settings [69].

Instead of install time permissions, iPhone's iOS has a one stop setting option for all apps plus runtime permissions. The one stop setting allows users to see all apps that request a certain permission, e.g., location, or all the permissions each app requests. If users want to understand why a permission is needed, they are forwarded to the privacy policy. With the centralized settings, iOS goes against privacy usability principle of making permissions contextual, for the sake of another usability principle, simplicity. Whether the centralized controls are usable when a user has dozens of apps is an interesting question worthy of further investigation.

There are further types of permissions that may be set by app developers. Abstracting away from platform specifics, these can be defined as follows:

- Static permissions: these are permissions set by app developers and managed by the users upon app installation (e.g.: app is allowed to send notifications, etc.).

- Dynamic permissions: these are permissions that are set by developers and are prompted during the app runtime (e.g.: when in the course of use an app needs access to the camera, the user will be prompted).
- Custom permissions: these are managed by the developers or different teams in an organisation which may be responsible for the development of multiple apps. They relate to permissions that can be set between different apps belonging to the same organisation. Since apps of the same developers/organisations may interact and exchange information between each other, these permissions are set during development phase and allow apps of the same manufacturer to exchange data or services between each other.
- Third-Party Library permissions: these are managed by the app libraries used by the developers. Third-party libraries (e.g., ad libraries) can set their own permissions. In Android, there is no isolation between components, and these permissions are propagated to the apps relying on these libraries. This may have the effect that apps request permissions that are not necessary for their core functionality. How it works for iOS is more complicated and we were unable to find research papers analysing the iOS framework for third-party library permissions.

Each permission type has varying impact on privacy and security. Depending on the platform, a number of static and dynamic permissions are managed by the user. Either upon installation or during runtime, the app requires the user to consent to data collection or some functionality of the device. Custom and ad permissions, on the other hand, do not explicitly require user consent. Developers set custom permissions at the development phase -- they may be mentioned in the privacy policies, but this is not always the case.

In most of the cases, app developers or app providers are not transparent about the user data or services required by the app. Ad permissions are even more critical, as the developers might not even be aware that these permissions are needed or foresee the potential impact these may have on user privacy.

4.2 Complexities and problems of permission management

The permission model is central to informing users and obtaining consent but come with many complexities. As shown in [48] and other similar studies, some known issues of permission management, are as follows:

- Pre-installed or OEM apps in Android are automatically granted all the required permissions.
- Most of the times the end-user, in order to be able to use a certain app, is forced to give all the necessary permissions, otherwise it is not guaranteed that the app can still work properly.
- Permissions are not a one-to-one mapping with the actual methods exposed by the API to manage the permissions (e.g.: access to the camera, may also grant access to the photos automatically).
- Permission revocation does not provide any guarantees to the user that the app still works as intended (e.g.: app may fail after removing the permissions).
- Certain apps may require more permissions than actually needed to functioning properly.
- Certain APIs may introduce security flaws by not providing full control over the resources of the mobile device, therefore exposing certain personal information stored in the mobile device to all the apps.

These design weaknesses and exceptions have led to over-collecting data as in the case of CarrierIQ, a pre-installed analytics app that was automatically granted permission to access phone logs which included login credentials [70], a prayer app that disclosed the user's location to all the other apps [71], and apps that practically prompt all possible permissions as in the case of Goluk, a dashcam app [72].

The above issues clearly violate a number of data protection rules (as discussed in Chapter 2), in particular the principle of data minimisation, purpose limitation, information and control, as well as security of personal data.

From a privacy perspective, there is also a fundamental challenge with the scope of dangerous permissions, as defined by Google. The division between the two is not reflective of the potential privacy leaks possible from normal permissions. In an experiment conducted at MIT, researchers wrote a scraper script that scanned the public Android API references to identify API calls that require no permissions. They found 36,000 unique API calls and, after manual analysis, confirmed that some of these can be used for "fingerprinting the phone, identifying vulnerable apps, and identifying location of private data for exploitation" [41]. This is an issue that needs to be solved at the OS level and has consequences across the mobile app ecosystem.

Inspired by [42]. on the Android ecosystem, some of the main problems associated with the permission models can be summarized as follows:

Problem 1: Permission comprehension and attention by users

Users have limited understanding of the associated risks of enabling permissions (or access to) in certain apps; app developers may use the Android explanations to express why a permission is needed, however, studies show that explanations are often not informative. Furthermore, asking users for permissions can lead to habituation, undoing any effective control or consent provided by the user participating in the permission model.

Problem 2: Permission comprehension and attention by app developers

Studies have shown that developers have difficulties in comprehending and appropriately handling permissions. In some cases, developers amplify bad security decisions at the platform level, e.g., allowing differently privileged apps to communicate with each other. In other cases, they may demand more permissions than needed and/or the permissions may make it possible to infer other private information e.g., device fingerprinting for tracking. This would violate, among other, the principle of "data protection by default" that is laid down in the GDPR. However, often it is not fully clear what the best (risk-minimizing) default is for the chosen purpose and the user base.

App developers have low insight into how to use APIs to correctly manage permissions; APIs are more than often insufficiently documented and do not identify all permission protected APIs; some documentation even has errors (e.g.: describe wrong permissions required for an API function).

Improper implementation of a permission model results in apps stalling during run time; this could have the unwanted effect that users give permissions in order to have the functionality.

Problem 3: Permission comprehension and attention by IDE and OS developers

Android makes a distinction between normal and dangerous permissions. However, as mentioned earlier, normal permissions can be combined to fingerprint and track users against their will. These different privileged apps, which can communicate with each other, open the platform to privilege escalation attacks. Android groups permissions are such that granting access to a single permission in the group gives access to all the permissions in that same group.

4.3 Crafting recommendations regarding permissions

The aforementioned problems with permissions show that app developers are often responsible for privacy problems but not always. For example, Android developers act within an app ecosystem that has loopholes and exceptions for different actors when it comes to requesting permission; that gives third-party libraries an upper hand in driving permission requests; and provides insufficient or misleading documentation of APIs and permission models. In order to empower app developers to implement privacy and data protection by design, these conditions need to change. Until such change occurs, it is important to give recommendations to developers on how to address privacy in such an ecosystem, highlighting bad practices and encouraging good ones. Most importantly, recommendations should distinguish (1) where and why privacy problems arise, (2) who is capable, and (3) who is liable for making the necessary changes, emphasizing the responsibilities of the data controller.

Regardless of the problems in the app ecosystem, recommendations must address those aspects that the developers can improve and have control over. Studies mentioned earlier show that developers suffer from the same comprehension problems that users have when it comes to understanding how permissions (and APIs) work and assessing the associated privacy risks. Studies also show that for app developers the difference between security and privacy is not clear. Developers often believe that protecting privacy is equivalent to securing user data at rest or in transit, lacking any in depth understanding of data protection requirements. It is hence important that recommendations include an introduction to how privacy and data protection differ and overlap with security (see Chapter 5 for some ways to do this).

We next drill down to explore how granular and specific recommendations should be in the context of the current permission models. Specifically, we discuss the following questions:

- How far should recommendations consider the privacy problems inherent to the ecosystem?
- At what granularity and technical detail should recommendations regarding permissions be? Specifically,
    - What is the best way to provide technical recommendations that will persevere over time and be generally useful?
    - Should recommendations go beyond data protection requirements, e.g., include recommendations on usable privacy, if they are valuable to supporting it technically?

To further discuss these questions, we need to first take a look at the existing recommendations to app developers provided especially by regulators in the field of privacy and data protection (see for example in [73] [27] [22]). A general finding is that such recommendations are usually reasonable vis-à-vis the high level data protection goals but do not always consider the underlying app development landscape and its complexities, as described in Chapter 3 of this document.

For example, it is often stated as a high level data protection goal that users should be adequately informed and consent should be obtained for any personal data processing before installing an application on users' smart mobile device. It is also usually stressed that the mobile app must ask for granular consent for every category of personal data it processes and every relevant use. Although this is reasonable in terms of legal obligations and GDPR, we observe that, vis-à-vis the developers, such high level recommendations:

- Seem to diverge from the reality of the permission models in the two OS, as well as from studies showing the impossibility of requesting permissions for all accessed data points [74]. Moreover, Android only prompts for "dangerous" permissions and iOS users find out what permissions will be requested at run time. The idea of informing users of all information categories prior to installation would be difficult if not impossible for the app developers to implement in practice.

- Imply that if the ecosystem fails to provide the necessary granularity, then the developers should themselves implement the necessary model to get consent for each category of data and relevant use. Aside from the burden posed to developers in this context, guidance is lacking on some of the hard, technical decisions that show up in literature, e.g., when to ask for permissions, habituation, transparency, etc. Usable privacy and security research shows that over prompting users leads to habituation (e.g., clicking away without considering the contents of a prompt) which may beat the purpose of informed consent.

This triggers a number of questions that, if we could point to answers, would help provide better guidance to developers for implementing the high-level goals of the GDPR. In particular:

- How reasonable is it to expect from app developers to make up for how the platform providers have preconfigured the type of permissions? Or, what are the possible constructive ways of bringing ecosystem problems into recommendations for app developers?
- How effective would such an approach be? Experience from mid-range systems indicates that app developers' design decisions could probably be circumvented by those of other parties in the ecosystem.
- How likely is it that developers will get it wrong when layering another permission model on top of the existing one?
- How should developers deal with balancing informed consent and user habituation?
- What should (temporary) permission revocation entail (with respect to past data and future functionality)?
- Are there other supporting principles that should be bundled with revocation of permissions, e.g., revocation of permissions should not lead to the dysfunction of the app?

Finally, in an ideal world, recommendations to app developers would also include security and privacy best practices, methodologies or activities that can be fitted into app development in an agile environment, and tools to evaluate own apps, third party libraries, integrate PETs and manage data protection compliance. Keeping such resources up to date may be a daunting task, which may benefit from institutional support and continuous work at a future Data Protection and Software Observatory.

5. Privacy and data protection by design in mobile apps

Privacy by design is a system engineering philosophy – initially formulated by Ann Cavoukian [75] – that stresses the fact that privacy considerations need to be taken into account throughout the system development lifecycle, from the inception of a system, through the design, implementation, and use all the way to the decommissioning of the system. In more technical, software development, terms, privacy is a *system quality attribute* [76]. Data protection by design has become mandatory with the GDPR coming into force, but many organisations still struggle with the concept, both in terms of what it exactly means, and how to implement it within the organisation.

This Chapter aims to make the concept of privacy by design more clear, especially for mobile app developers. We will discuss approaches to privacy and data protection by design and by default that help translate the legal requirements into more tangible engineering goals that developers are more comfortable with. In particular, we will explain the concept of *data protection goals* (section 5.1) and make this more concrete using *privacy design strategies* (section 5.2). Both approaches are especially suitable to steer the privacy and data protection by design process early in the system development lifecycle, i.e. in the early concept development and analysis phases, where the initial information architecture is typically defined. We will describe both approaches in general terms, but will explain them and make them more concrete using examples from the mobile app development perspective.

We note that the approaches described in this chapter are (by necessity) quite abstract and high level in nature, due to the wide variety of application scenarios, even in a limited setting like mobile app development. For this reason, the approaches do not specifically address the complexities of the mobile app ecosystem as described in Chapters 3 and 4. In fact, the approaches outlined in this Chapter can be applied at different layers in the software development process (giving more concrete recommendations as the approach is applied further down in the process), and can be relevant to different components in the app development ecosystem. Each of the stakeholders in this ecosystem can only address the privacy-friendliness of the systems they are responsible for themselves, and depend (sometimes crucially) on the opportunities for and limitations to privacy friendly design imposed by the other components. In particular, app developers are constrained by e.g. the properties of the telecommunications network, the permission system and the absence/presence of tracking facilities offered by the mobile operating system, as well as the way the app store works, as outlined in Chapter 3.

5.1 Data protection goals

5.1.1 Introduction to data protection goals

For ICT security, the classic triad of the protection goals confidentiality, integrity, and availability is well known and accepted. Developers are used to the model of security protection goals and have at least some understanding of why it makes sense to consider those goals in the development process and that, depending on the purpose or scenario of an app, the importance of each protection goal may vary. Different communities have derived further protection goals; still those three core protection goals form a stable foundation for evaluating ICT security properties, identifying risks and selecting appropriate measures to counter these risks.

Since 2009, three protection goals focusing on data protection have been proposed to complement the three security protection goals: unlinkability, transparency, and intervenability [77]. They extend the well-known triad of security protection goals, but shift the perspective to the individual whose privacy may be at stake and whose personal data are processed. Those protection goals were not chosen accidentally, but stem from data protection law: each of the protection goals addresses data protection principles as they are laid down in the GDPR, in particular in Article 5 GDPR. A more detailed description follows, quoted from the ENISA study on "Privacy and Data Protection by Design" [4].

Unlinkability (including data minimisation[34])

- Unlinkability ensures that privacy-relevant data cannot be linked across domains that are constituted by a common purpose and context, and that means that processes have to be operated in such a way that the privacy-relevant data are unlinkable to any other set of privacy-relevant data outside of the domain.
- Unlinkability is related to the principles of necessity and data minimisation as well as purpose binding. Mechanisms to achieve or support unlinkability comprise of data avoidance from the very beginning (data minimisation), separation of contexts (physical separation, encryption, usage of different identifiers, access control), anonymisation (including aggregation or adding noise for ensuring that the data cannot be linked to a person and that persons cannot be singled out) and pseudonymisation, and early erasure of data.

Transparency

- Transparency ensures that all privacy-relevant data processing including the legal, technical and organisational setting can be understood and reconstructed at any time. The information has to be available before, during and after the processing takes place. Thus, transparency has to cover not only the actual processing, but also the planned processing (ex-ante transparency) and the time after the processing has taken place to know what exactly happened (ex-post transparency). The level of how much information to provide and in which way it should be communicated best has to be tailored according to the capabilities of the target audience, e.g. the data controller, the user, an internal auditor or the supervisory authority.
- Transparency is related to the principles concerning openness and it is a prerequisite for accountability. Mechanisms for achieving or supporting transparency comprise logging and re-porting, an understandable documentation covering technology, organisation, responsibilities, the source code, privacy policies, notifications, information of and communication with the persons whose data are being processed.

Intervenability

- Intervenability ensures intervention is possible concerning all ongoing or planned privacy-relevant data processing, in particular by those persons whose data are processed. The objective of intervenability is the application of corrective measures and counter-balances where necessary.
- Intervenability is related to the principles concerning individuals' rights, e.g. the rights to rectification and erasure of data, the right to withdraw consent or the right to lodge a claim or to raise a dispute to achieve remedy. Moreover, intervenability is important for other stakeholders, e.g. for data controllers

[34] In several publications, "data minimisation" is regarded as an own protection goal that has to be taken into account before applying the other protection goals. It is argued that otherwise the prominent role of data minimisation for designing data processing systems may not be sufficiently apparent. However, the strong relationship to Unlinkability allows an approach that integrates the important concept of data minimisation as part of the Unlinkability protection goal.

to effectively control the data processor and the used IT systems to influence or stop the data processing at any time. Mechanisms for achieving or supporting intervenability comprise established processes for influencing or stopping the data processing fully or partially, manually overturning an automated decision, data portability precautions to prevent lock-in at a data processor, breaking glass policies, single points of contact for individuals' intervention requests, switches for users to change a setting (e.g. changing to a non-personalised, empty-profile configuration), or deactivating an auto pilot or a monitoring system for some time. Note that these mechanisms often need the cooperation of the service provider (often referred as honest-but-curious-attacker model).

Working with protection goals means to balance the requirements derived from the six protection goals (ICT security as well as privacy and data protection goals) concerning data, technical and organisational processes. Considerations on lawfulness, fairness and accountability provide guidance for balancing the requirements and deciding on design choices and appropriate safeguards.

For all roles and actors in a system, the protection goals can be employed. This is also necessary to identify potential conflicts. For instance, if the integrity of an e-mail archive is achieved by using hash values calculated on the basis of the previous e-mail, no single e-mail can be erased without compromising the integrity. However, in a case where a data subject may legally demand the erasure of a certain e-mail, his or her right to erasure – a manifestation of the data protection goal "intervenability" - should not be overlooked because of the choice of the integrity mechanism. Instead, the choice of mechanisms has to take into account all protection goals to the necessary extent. This effect is well known also in the context of information security, e.g. when it has to be figured out under which conditions data should be erased (to prevent a breach of confidentiality) or be kept (to achieve availability).

Definitions of the protection goals and a description of the Standard Data Protection Model have been proposed by the German Conference of Data Protection Authorities as a means for data protection audits [78]. The Standard Data Protection Model can not only be used for evaluating existing data processing systems, but is helpful in the design process both for developers and controllers or processors. The technological and organisational measures address the entire workflow and the full lifecycle of personal data.

5.1.2 Addressing data protection goals in mobile apps

In the following the focus is set on the three data-protection-specific protection goals for the case of mobile apps. For each one of them examples of possible implementation are provided, which correspond to different privacy by design strategies (see also section 5.2). Please note that these non-exhaustive lists of examples contain design recommendations that have to be interpreted with respect to the legal ground and the purpose of the data processing. For instance, the protection goal of unlinkability may be less demanded in a situation where linkage across apps is necessary for the purpose or is at least allowed, e.g., when a user chooses to share data from one app with his/her contacts. Some examples in the following list may be legally demanded in the context of the GDPR (e.g. several of the transparency issues), while for others the implementation would at least be recommended as best practice:

- Unlinkability:
  - For each app the purpose(s) should be determined and stated beforehand. Only personal data necessary for the purpose(s) should be processed.
  - The app should be isolated from the platform as much as possible; usage data concerning the app should not be communicated to the platform provider.

- Different apps, i.e. processing personal data for different purposes, should be isolated by default; a data exchange should be prevented unless explicitly specified or otherwise chosen by the user.
- Unique identifiers should not be used for different purposes; the unwanted linkage between identifiers should be prevented.
- Personal data should be erased as soon as possible.
- If the personal data cannot be erased, they should be anonymised or, if this is not possible, pseudonymised as soon as possible.
- In case the app offers different ways of configuration, the default configuration should ensure that only personal data which are necessary for the purpose are processed, i.e. minimum amount of personal data collected, minimum extent of their processing, minimum period of their storage, minimum accessibility.
- Functionality that may infringe the privacy and security of users should not be activated before the user voluntarily and in the knowledge of the related risks gives his/her consent. This comprises functionality of transferring audio or video data from the user side (e.g. an activated microphone or an activated video functionality), location data or data from the address book of the user.
- Users should be able to use an app without any network connection and potential data flow to other parties (e.g. using an offline map), as far as the purpose of the app allows it.
- The developed app should work together with privacy tools for self - data protection.

- Transparency:
  - Users should be informed about the privacy-relevant information that is collected and analysed. This comprises both the app and related code provided by other parties.
  - Users should be informed about any data flow with respect to privacy and data protection.
  - Users should be informed on how to exercise their data subject rights: access, rectification, erasure, giving and withdrawing consent, portability.
  - Users should be able to understand where to get help for their questions or problems (help desk). Even in complex systems, the way to get help or to achieve remedy should be clear. In case of different organisations offering help, the respective responsibilities should be clarified.
  - Users should be made aware which data controller is responsible for which app or for which data flow.
  - The necessary information should be communicated to users in a way that it can be easily understood. This could comprise multi-layer policies with the most important information on the first layer, support by visual icons, audio information and machine-readable approaches.
  - A thorough documentation of the developed system should be available. The documentation specifically should contain which personal data are processed and which data flow may or will occur.
  - The interfaces of the app and the possibilities of combining the app usage with self - data protection tools should be documented.
  - Activities by administrators and/or changes of the IT system should be logged to the necessary extent, e.g. to support the integrity of the system and to prove that the data were correctly processed.
  - Risk management, as well as the compliance with the requirements of the GDPR should be documented, e.g. in a Data Protection Impact Assessment.
  - Best practice solutions and specific privacy-enhancing achievements should be communicated so that others can learn from those achievements and experiences.

- Intervenability:
  - Users should be enabled to exercise their data subject rights: access, rectification, erasure, giving and withdrawing consent, portability.
  - Users should be provided with a central communication point for potential complaints (e.g. a help desk).
  - Users should be able to change the pre-configured setting to their needs.
  - When changing a pre-configured “data protection by default” setting, it should be done with the appropriate granularity, thereby preventing that the users’ protection is fully lost at once.
  - After having changed the pre-configured “data protection by default” setting, users should be able to go back to the default setting.
  - All involvement of users should take into account usability requirements.
  - It should be possible to provide necessary updates.
  - It should be possible to stop any data flow immediately.
  - It should be possible to exchange components (such as third-party libraries or clouds).
  - Proper reactions to changes or events that influence the data protection functionality of the system should be ensured. Among others, patches against vulnerabilities have to be installed; a data breach notification has to be sent to the Data Protection Authority and potentially to users; in the case of legal changes, proper adaptions have to be realized (this is part of a functioning data protection management system).

In the next sections the concept of privacy design strategies is explored and linked to data protection goals.

5.2 Privacy design strategies

As described by *Colesky, Hoepman and Hillen* [79] a privacy design strategy specifies a distinct architectural goal in privacy by design to achieve a certain level of privacy protection. It is noted that this is different from what is understood to be an architectural strategy within the software engineering domain. Instead our strategies can be seen as the goals of the privacy protection quality attribute.

In the description of the privacy design strategies we frequently refer to processing of personal data. Engineers should be aware that the legal concept of processing is broader than what a typical engineer understands processing to mean. In what follows we use the legal interpretation of processing, which includes creating, collecting, storing, sharing and deleting personal data.

We now proceed to briefly describe the eight privacy design strategies [80], and give examples on how they could apply when developing a mobile app.

5.2.1 Minimise

Definition: Limit the processing of personal data as much as possible.
Associated tactics:

- Exclude: refrain from processing a data subject’s personal data, partly or entirely, akin to blacklisting or opt out.
- Select: decide on a case by case basis on the full or partial usage of personal data, akin to whitelisting or opt-in.
- Strip: removing unnecessary personal data fields from the system’s representation of each user.
- Destroy: completely removing a data subject’s personal data.

Example: Apps should limit their access to sensors (location, motion, camera, microphone) and locally stored data (pictures, contacts) to the bare minimum, and only when clearly relevant for the proper functioning of the app. The proverbial flashlight app asking permission to access all of these is the perfect anti-example of this case. A weather app asking for the location, to give local weather information, can be acceptable, though. Note, however, that for a weather app to deliver a localised weather forecast it does not need to know the *exact* location. Instead a coarse indication of the location (e.g. the name of the city, or indication of the area in several square kilometres) will do.

Other examples are using explicit whitelists (of people and or attributes) of data to collect, or blacklists (of people and or attributes) of data that should not be collected.

5.2.2 Separate

Definition: Prevent correlation of personal data by separating the processing logically or physically
Associated tactics:

- Distribute: partitioning personal data so that more access is required to process it.
- Isolate: processing parts of personal data independently, without access or correlation to related parts.

Example: Mobile devices are incredibly powerful, in terms of processing, bandwidth and storage, and can therefore perform many tasks locally that were unthinkable several years ago. For example, image recognition within pictures can be done on the smart phone, so that uploading pictures to a central server is no longer necessary. Also, the increased networking capabilities (and data usage allowance) make peer-to-peer applications, where users directly share or communicate without the help of a central server, a possibility. In particular, peer-to-peer social networks are, technically speaking, a possibility.

5.2.3 Abstract

Definition: Limit as much as possible the amount of detail of personal data being processed.
Associated tactics:

- Summarise: extracting commonalities in personal data by finding and processing correlations instead of the data itself.
- Group: inducing less detail from personal data prior to processing, by allocating into common categories.
- Perturb: add noise or approximate the real value of a data item.

Example: Location based services typically only need an approximate indication of the current user's location to offer an overview of services near this location. So instead of using the precise GPS coordinates offered by the smart phone, a location based app could compute a coarser location before querying the associated services from a central server. In fact, mobile devices could make several levels of granularity of location available as OS API calls, and even allow users to grant or deny access to more precise location data on a per-app basis.

A more generic approach to privacy friendly authentication, is when people are allowed access to data based on more general attributes (e.g. whether the user has a subscription), instead of using the identity of the person to make that access decision. So called Attribute Based Credentials (ABCs) support this is in a privacy friendly and unlinkable manner (which make ABCs also fall under the Hide strategy described below).

5.2.4 Hide

Definition: protect personal data, or make them unlinkable or unobservable. Prevent personal data becoming public. Prevent exposure of personal data by restricting access, or hiding its very existence.

Associated tactics:

- Restrict: preventing unauthorized access to personal data.
- Mix: processing personal data randomly within a large enough group to reduce correlation.
- Encrypt: encrypt data (in transit or at rest).
- Obfuscate: preventing illegibility of personal data to those without the ability to decipher it.
- Dissociate: removing the correlation between different pieces of personal data.

Example: At the very minimum apps should encrypt all their communications, and use certificate pinning (or preinstalled keys) to prevent man-in-the middle attacks when adversaries are able to compromise the TLS certificate infrastructure. More advanced applications will try to hide the metadata by using mixing techniques, or deploying an onion routing network (e.g. Tor).

5.2.5 Inform

Definition: provide data subjects with adequate information about which personal data is processed, how it is processed, and for what purpose.

Associated tactics:

- Supply: making available extensive resources on the processing of personal data, including policies, processes, and potential risks.
- Notify: alerting data subjects to any new information about processing of their personal data in a timely manner.
- Explain: detailing information on personal data processing in a concise and understandable form.

Example 1: A possible method to intuitively convey the way an app handles personal data, especially given the small screen of a smart phone, is to use privacy icons (similar in spirit to those used for the creative commons). No standardized approach currently exists, and a few competing proposals exist. Uptake of these has been slow, partly because their usefulness is questioned by some. Use of such icons in an app store could create a de-factor standard for such icons.

Example 2: Also, access to sensors or locally stored data can be signalled to the user in less obtrusive ways than a “pop-up-and-say-ok-or-cancel”, i.e. modal, dialog. For example, by using special icons in a status bar that light up when certain types of sensitive data are accessed and that, when clicked, provide more information about the specific access and give the user the option to change the settings to address any concerns associated with the access. As a concrete example, we can consider the use of a small arrow in the status bar on the top of the screen of iOS devices to signal the (recent) use of location services.

5.2.6 Control

Definition: provide data subjects mechanisms to control the processing of their personal data.

Associated tactics

- Consent: only processing the personal data for which explicit, freely-given, and informed consent is received.
- Choose: allowing for the selection or exclusion of personal data, partly or wholly, from any processing.
- Update: providing data subjects with the means to keep their personal data accurate and up to date.

- Retract: honouring the data subject’s right to the complete removal of any personal data in a timely fashion.

Example: When asking for permissions to access sensors (location, motion, camera, microphone) and locally stored data (pictures, contacts), mobile apps should still work (perhaps offering limited functionality) when that access is not provided.

5.2.7 Enforce

Definition: commit to a privacy friendly way of processing personal data and enforce this.
Associated tactics:

- Create: acknowledging the value of privacy and deciding upon policies which enable it and processes which respect personal data.
- Maintain: considering privacy when designing or modifying features, and updating policies and processes to better protect personal data.
- Uphold: ensuring that policies are adhered to by treating personal data as an asset, and privacy as a goal to incentivize as a critical feature.

Example: First and foremost, this strategy requires the app developer to specify and enforce a privacy policy. Another approach is to set up a privacy management system similar to the Information Security Management System (ISMS) defined in ISO 27001.

5.2.8 Demonstrate

Definition: provide evidence that you process personal data in a privacy friendly way.
Associated tactics:

- Log: tracking all processing of data, without revealing personal data, securing and reviewing the information gathered for any risks.
- Audit: examining all day to day activities for any risks to personal data, and responding to any discrepancies seriously.
- Report: analysing collected information on tests, audits, and logs periodically to review improvements to the protection of personal data.

Example: Apart from the self-explanatory tactics cited above, where logging can be done both centrally and on the smart phone (and as a consequence auditing can be done both centrally and on the user device, perhaps by an independently developed and provided tool), the “demonstrate” strategy also puts burden on the app developer to carefully select the libraries provided by third parties that it includes within the app to implement certain functionality. In particular, it must be checked (and verifiably documented) that the library does not violate the privacy policy.

Also performing a proper Data Protection Impact Assessment and documenting its result is a key contributor to this strategy**.**

5.3 Relating privacy design strategies and data protection goals

The privacy design strategies aim to refine the data protection goals, especially through the definition of associated tactics, and in some cases broaden the scope beyond that of the user (aka data subject) to also include the perspective of the data controller. Let us make the relationship between data protection goals and privacy design strategies a little clearer.

The 'Unlinkability' protection goal encapsulates both the 'Hide' and the 'Separate' strategy. Separation ensures that data collected for different purposes in different domains do not get mixed. Hiding, e.g the use of pseudonyms, ensures that the link between a data subject and his/her associated data is removed or only accessible to a restricted set of employees within the data controller organisation. Also the 'Minimise' and 'Abstract' strategies contribute to the stated objectives of the 'Unlinkability' protection goal, even though the name of the protection goal does not immediately suggest that data minimisation is part of it.

The 'Transparency' protection goal corresponds one to one with the 'Inform' strategy. This strategy naturally depends on the 'Enforce' strategy for the definition of a privacy policy. We do note however that the 'Enforce' strategy goes way beyond the 'Transparency' goal, in that it requires a data controller to also maintain and uphold this privacy policy within the data controller organisation. This entails, among many other things, setting up a privacy management system (comparable to an information security management system [81], and assigning responsibilities and resources within the organisation.

The 'Intervenability' protection goal similarly corresponds to the 'Control' strategy. Again the 'Demonstrate' strategy is not fully part of this protection goal, as it mostly targets the data controller perspective and its dealings with the data protection authority; it is rather part of the 'Transparency' protection goal.

We see that the 'Enforce' and, to a lesser extent, 'Demonstrate' strategy are not really part of the data protection goals outlined above. This can be explained by the fact that the privacy design strategies aim to refine and extend the privacy protection goals in the field of data protection management.

5.4 Towards a privacy by design methodology for apps

Privacy engineering is a topic of active and ongoing research[35], but, as also shown in Chapters 3 and 4, a concrete, simple to use methodology to apply privacy by design for app developers is missing at the moment. We believe a lightweight methodology based on, for example, the privacy design strategies combined with the data protection goals is feasible, as long as it takes account of the current app development practices and complexities. The main advantage of this approach is that it clearly separates the legal requirements from the more concrete engineering goals. This removes the current unreasonable expectation that engineers need to think like lawyers or social scientists. While this approach does not waive the necessity of checking the conformity with detailed legal requirements that may be laid down in sector-specific regulation before employment of the app, it still will be a huge step forward to build in privacy and data protection principles (as in Article 5 GDPR) into the ICT systems. Thereby this approach can provide much better solutions than the current status where privacy and data protection guarantees are rarely built in. What is more, both data protection goals and privacy design strategies provide a means to express requirements and options in a language that can be comprehended across disciplines.

A first step towards such a methodology is providing an overview of best practices, design patterns, building blocks and concrete code, as well as mistakes to learn from. From these examples it can be made clear that even seemingly small architectural decisions may have a huge influence on privacy and data protection:

[35] The Privacy Enhancing Technologies Symposium, the International Workshop on Privacy Engineering and Symposium on Usable Privacy and Security (SOUPS) are some of the venues where this research can be found.

- For instance, if linkage between processing operations for different purposes should be prevented, this should be reflected by avoiding the typical re-use of existing unique identifiers (e.g. hardware IDs or already set globally unique identifiers). Further, exact time or location information can be used for linking personal data across purposes and thereby should only be processed if necessary for the purpose.
- Another decision to be made is the location of the data storage and processing. This affects who may access the personal data (e.g. if the data are stored in a remote cloud, potentially under a jurisdiction that allows governmental access) and how well they can be secured (e.g. against hacking or in case the user loses his or her phone).
- App developers should choose third party components carefully because their behaviour may pose privacy and security risks to users, e.g. by collecting user data on their own without a legal ground and transparency for the users. The architecture should facilitate an exchange of third party components if it turns out that they don't behave as promised and infringe data protection law.
- The decision on what functionality is wired-in (e.g. always encrypted communication) and what is configurable by the user influences the degree of privacy and data protection. For all configurable settings the developer has to decide whether there is a pre-setting or whether the user is asked at installation time. The GDPR demands for any pre-setting to follow the principle of "data protection by default" (Article 25(2)), so that the amount of personal data collected, the extent of their processing, the period of their storage and their accessibility is limited to what is necessary for each specific purpose. If, e.g., tracking or personalization is not necessary for the purpose, the default setting should ensure that the respective data are not processed unless the user actively changes the configuration.

App developers are used to architectural decisions for their apps, but in the past privacy and data protection demands were rarely prioritized so that the app ecosystem often preferred privacy-infringing standards instead of built-in privacy and data protection. The GDPR may function as a game changer by demanding data protection by design and by default [82].

Usually, neither information security nor privacy and data protection are primary goals when developing an app; instead the development focuses on the app's functionality. The necessary risk assessment must not be limited to the perspective of the data controller and the assets to be protected, but has to consider also the data subjects' interests and their rights and freedoms. This encompasses that the data controller has to be treated as a risk source as well (see section 2.3). Therefore, the requirements derived from the data protection goals should take into account the perspectives of all roles and actors including the users and their rights according to the data protection law.

It is desirable to provide well documented best practice solutions, concepts or even the code of implementations to developers, e.g. privacy-enhancing or transparency-enhancing technologies. This could be done via a repository containing reference solutions that can be analysed concerning their merits and potential disadvantages, as well as copied and adapted for own purposes. Since the degree of readiness is varying and the solutions offer different levels of protection of different requirements, this information should be added for each entry in the repository [76]. This would facilitate developers as well as supervisory authorities to distinguish state-of-the-art solutions from others, and funding initiatives could identify those ideas and concepts that need a little more support to become ready for the market [83].

Further research is necessary to determine the best medium by which to provide such an overview (instruction videos, discussion fora, interactive website, etc.). In any case, information about developer privacy decision points (when should they be thinking about privacy and data protection, where do they go if they are thinking about privacy and data protection to find more resources) should be taken into account.

6. Conclusions and Recommendations

As discussed in the previous Chapters, privacy and data protection face some serious challenges in the area of mobile apps, especially due to the complexity of the current app ecosystems and development practices. App developers in particular, even if they are aware of the underlying legal requirements (from GDPR and ePrivacy regulation), often struggle to embed them into their products and meet several constraints, also due to limitations of other parties in the ecosystem (OS providers, third party software, etc.).

Following the analysis in the previous Chapters, in this Chapter we derive some key conclusions and make relevant proposals and recommendations towards embedding privacy and data protection by design in the app development process. As in the rest of the document our focus mainly lies on app developers, as these entities may play a central role in the privacy and security properties of mobile apps *by design*.

6.1 Providing guidance to app developers

As our research earlier shows, one important issue in the area of mobile apps and privacy is the gap between legal requirements and the translation of these requirements into practical solutions that app developers can implement. Indeed, existing recommendations to app developers usually provide insights only into *what* the developers are required to do, without further guidance on *how* they will be able to fulfil these requirements.

For example, in the area of permissions (see Chapter 4), we observe that data protection guidance is focused on how app providers/developers should organize consent with respect to the personal information they will be collecting and processing. In a sense, they are answering a *"what"* question: 'What should the app provider/developer do? He/she should ask for consent'. Yet, the recommendations contain less or no guidance for *"how"* to ask for consent, which is central to privacy by design. The "how, when, in what way" are questions that app developers have to ask when it comes to translating some of this into the permission models provided by the different OS. Hence, ideally, recommendations that are actionable for app developers should include answers to some basic questions like:

- When should developers be asking for the permission (e.g., at install, runtime, when an app is updated)?
- How frequently should the developers prompt the users for permissions and how should they deal with user habituation?
- Are there ways to re-design app functionality (or the OS platform) so that the number of permissions necessary can be minimized?
- How much margin should be left for user preferences? What is the acceptable margin of user choice with respect to permissions and notification?

Whether these questions are the right ones, and how to best answer them so that they really are actionable for developers are important matters that should be part and parcel of any initiative to provide recommendations to app developers. A key challenge is to provide recommendations that are general enough, so that they do not lose their relevance with updates to operating systems and associated privacy functionality and are consistent with current constraints on the ecosystem.

Moreover, some other important matters that need to be addressed in order for such recommendations to be useful and effective are as follows:

- **Awareness and education.** In many cases the developers may not be the ones liable or may not be seen as the ones responsible for data protection, however, they may be in place to make a difference. Recommendations should be coupled with an awareness campaign that allows the developer to identify what role they can play in addressing data protection requirements. They should also make apparent the urgency for a developer to address these issues given the numerous other responsibilities that they are tasked with. Education of developers on privacy and security is crucial to this end.
- **Specific to the conditions of the app developers.** There are few scientific studies evaluating developers and the conditions under which they develop mobile apps from a data protection by design perspective. A few projects have recently been launched to consider how to best give recommendations to individual developers[36]. It is an interesting challenge and a topic worthy of further study whether it is possible to provide generic data protection recommendations to all app developers or whether these need to be tailored to the different constraints like the size of an organisation, structure of the team, development methodology, app domain and type of app.
- **Dissemination**. It needs to be explored whether a single PDF document, a top 10 list, a portal, or other options are the best way to make the recommendations available, up-to-date and usable to the developers. The answer may also depend on the type of recommendations (e.g., code snippets may be better displayed on StackOverflow, while legal requirements better explained in a longer PDF) or the pain points that drive developers to the recommendations. Figure 6.1 shows recent statistics for resources that android developers turn to when they seek technical assistance or have security related questions. These statistics stem from studies showing that the most popular resource might not be the one that returns correct or effective results (i.e., books may serve developers better than landing on a random page on StackOverflow [84]).

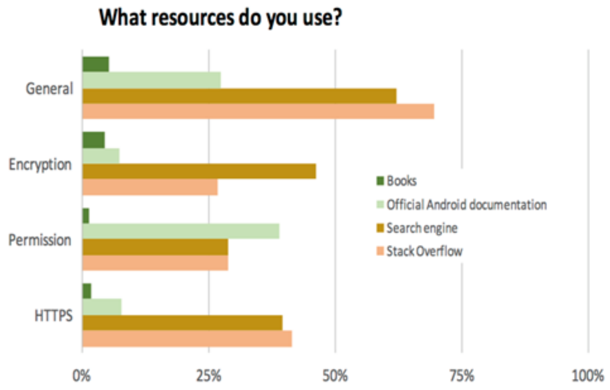


Figure 6.1: Resources that developers use for technical questions

- **Speaking to developers**. Recommendations should be easy to comprehend and not expect developers to think like lawyers, academics, or policy makers. They should ideally speak to their working conditions and also relate to the management stress or the feelings of being overwhelmed that they may be encountered when they want to address topics like data protection and privacy.
- **Identify typical data protection pain points:** It is important to identify and capture those moments when developers notice they have a data protection responsibility. Empirical studies could identify the typical questions that may lead developers to tackle (or avoid) data protection challenges. From “Am I collecting personal data?” to “Am I done if I anonymise user data?” to “What do I have to do to make sure I am not the one paying the large data protection fine?” are potential questions that may lead developers to data protection recommendations.
- **Learning from past mistakes**. It is well documented that privacy policies are not effective in informing end users and over-asking permissions may lead to habituation. It is easy to implement mechanisms for data protection that beat the purpose of their execution in the first place. Recommendations should not

[36] See for example the project of University of Hannover on Android's Developer Documentation 2.0: Helping Developers Write More Secure Android Applications, https://www.sec.uni-hannover.de/projekte00.html?&tx_tkforschungsberichte_pi1%5BshowUid%5D=498&tx_tkforschungsberichte_pi1%5Bbackpid%5D=373&cHash=cd99092dc8dcbd835ca4f62283cd6ca7

blindly propose applying the law but should ideally distil lessons learned from applying data protection law and privacy technologies in the last 20 years.

Ideally, recommendations should be developed iteratively. Frequent evaluation of effectiveness and quality of the recommendations, as well as updates in the rapidly changing ecosystem, can be used to provide input into the next iteration of recommendations. Even if the recommendations are a one shot simple document, evaluation of the usability and effectiveness should be completed and published for use in future initiatives.

Policy makers, regulators (e.g. Data Protection Authorities) and the research community should provide recommendations to app developers that shed more light on how privacy requirements can be fulfilled in practice.

Professional organisations and the industry of app developers should provide more insight on current development practices and key privacy and security issues.

The European Commission and EU institutions in the area of privacy and security should promote the formulation of workshops and working groups with all relevant stakeholders in the field.

Policy makers and regulators should invest in awareness raising for app developers and other actors in the app development ecosystem, in co-operation with professional organisations and industry of app developers.

6.2 Need for scalable methodologies and best practices

The turn to agile development methods makes it difficult for developers to apply heavy-weight methods developed for big design up front approaches. What are some simple techniques or best practices that an agile app developer can take up? Given the prominence of agile methodologies among app developers, we highly support research and development of scalable methodologies for data protection by design. These methodologies should take into account the different positions that app developers may take in the app ecosystem (OS vs. library vs. app developers) and the possibilities and constraints associated with these. Best practices, strategies and patterns can also be used to augment proposed methodologies.

In order to develop such scalable methodologies, a grounded analysis of the state of the art is necessary. Such a state of the art should include:

- **Survey of app development methodologies used in practice.** The challenges for developers who do and do not use structured (agile) approaches are likely to be different and should be analysed in a survey on current development approaches. The results of this survey are valuable for understanding whether SDL like methods are in use and whether it is possible to piggy-back on existing methodologies (like SDL) with respect to matters of data protection.
- **Assessment of whether it is reasonable, desirable and feasible to integrate data protection issues into SDL**. Privacy in the mobile app ecosystem is usually considered as part of the security research. However, the security engineering view on privacy often falls short of addressing the application of data protection principles in mobile apps that are not well aligned with security concerns. One would expect to find studies that focus on how privacy and data protection problems arise when developing of apps in the current ecosystem, but these are rare. Such studies would be very valuable in identifying bottlenecks as well as points for applying regulatory, socio-technical, market or technical solutions in the ecosystem. Further research is particularly needed to evaluate whether SDL would be a good place for data protection issues to also be brought to action. Another possible avenue of research is to consider identifying ways to reduce developers' burden by identifying potential alignments between data

protection and other quality requirements that developers are expected to fulfil (e.g., data minimisation may align with security, performance and resilience requirements).

- **Empirical studies for identifying and implementing privacy action points**. A complimentary approach is to identify potential points in the development life cycle that can be used to insert privacy actions. Empirical studies may be used to discover potential privacy action points or to evaluate their use in practice. For example, "story/scenario analysis and development" may be a key point to plug-in activities that surface or fulfil privacy requirements. Some have proposed the "abuser stories" to surface security requirements: these can be adapted to privacy and data protection. Building a "data protection testing" suite is another opportunity worth pursuing in this line of thinking.
- **Best practices, privacy patterns and code snippets**. Further work is needed on understanding what are effective and good responses that could be used to respond to privacy and data protection pain points. In addition to methodologies, developers appreciate specific recommendations, which may be a list of best practices, privacy patterns, tools that can be integrated into the development environment, or even at times code snippets. Recommendations for best practices can be enriched with examples of typical mistakes or "privacy dark patterns".

The research community should continue work in privacy and security engineering for apps, focusing in particular in the integration of privacy requirements as part of app development and SDL like methods.

The European Commission and EU institutions in the field of privacy and security should support relevant initiatives and research projects in this area.

Regulators (e.g. Data Protection Authorities) should provide an assessment of developed systems and methodologies and point out best available techniques, as well as state-of-the-art solutions that can be adopted by data controllers or app developers.

6.3 A DPIA framework for mobile apps

One place where the GDPR spells out an activity that aligns well with development rather than data operations is with data protection impact assessments (DPIAs). DPIAs require an analysis of impact and risks that can serve as a reference for threat modelling and may guide future development activities (see also section 2.3). However, as mentioned previously, existing methods for applying DPIAs tend to assume a big design up front model which clashes with the agile practices dominant in app development. This goes to show that further research is necessary on if and how DPIAs can be made effective and actionable in an agile environment.

Some further points that require attention in this regard are as follows:

- **Homogenization of DPIAs**. Different DPIA methodologies have been proposed by various parties and different DPAs [34], [85], [36], [35], [86]. Although many of them share a similar methodology, they are different. Most developers (especially the ones that do not have access to a Data Privacy Officer - DPO) are confused and do not know how to process, and which methodology to use. It would be recommended to homogenize these proposals to come up with a single methodology or a DPIA framework that would be accepted across Europe. Furthermore, most developers do not apprehend the difference between a security risk analysis and privacy impact assessment. This should also be clarified.
- **Specific DPIA for mobile apps.** Many mobile apps collect similar personal data and perform similar processing. It might be advisable to develop a specific DPIA methodology/framework for mobile apps, as it was done for RFID technologies [87]. Alternatively, a DPIA methodology for different types of mobile

apps (game, quantified-self, news, social networks, etc.), libraries or SDKs (Software Development Kits) could be considered.

- **DPIA support tools**. The different parts composing a DPIA are well defined. It is therefore recommended to develop supporting tools to help a data controller performing a DPIA by guiding the controller though these different phases. Similarly, testing and auditing tools would also be very useful. Many developers use third-parties without knowing what they are actually doing and which personal data they are collecting and processing. Analysis and transparency tools, such as Lumen [88], App-census [89] or MobileApps scrutinator [9] need to be developed and promoted. Note that these tools could also be used by developers to analyse their own mobile apps. It is also recommended to promote initiatives, such as the Data Transparency Lab[37], that aims at developing tools to improve transparency.

  The European Commission and regulators (e.g. Data Protection Authorities) should closely co-operate with professional organisations, industry of app developers and researchers with the aim of developing a DPIA framework for mobile apps.

  Policy makers and regulators should promote the use of DPIAs as a means for threat modelling and building data protection by design and by default in their tools.

6.4 Improving privacy and usability in the developer ecosystem

Recent research has turned to improve the usability of developer tools. This may include IDEs, APIs, and Operating Systems. Much of this research focuses on security rather than privacy and data protection. Further studies are needed to understand the state of the art in the privacy opportunities and obstacles inherent to IDEs, APIs and OSs, as well work on providing APIs for PETs and data protection relevant functionality. Further, there is a need for tools to test, verify and audit existing libraries, services, APIs etc. To address these matters, following studies are recommended:

- **Empirical evaluation of IDE's, API's, build environments.** Studying privacy and data protection relevant APIs; privacy and security defaults in APIs; APIs specific for data protection (e.g., privacy policy generators), as well as the evaluation of the usability of APIs with respect to privacy and data protection.
- **In depth analysis of ad libraries and their APIs.** Developers need greater transparency and control over ad libraries if they want to fulfil their data protection obligations. We are missing an overview of the options that are or should be available to app developers (and implemented by OS or third-party library providers) so that they can fulfil their data protection requirements in ad libraries.
- **Building libraries for Privacy Enhancing Technologies.** If development is more and more about integrating different functionality from third party services, it seems like a great opportunity to encourage the development of PET functionality in the form of libraries with usable APIs. It is important to promote some (simple) practical privacy preserving techniques and solutions, possibly with examples and code. It is also important to stimulate research and development in this area. Many issues still require some research work (such as privacy-preserving analytics, privacy dashboard, data anonymisation). It would, for example, be very useful to develop and publish libraries that perform privacy-preserving analytics or that implement a privacy dashboard. Developers could then use them without having to code them from scratch or turn to data-hungry versions of services.
- **Evaluation of privacy tools available to app developers.** A number of research tools exist to evaluate privacy sensitive information flows in apps. Developers can use these to test their own apps, or other

[37] For more information, see: http://datatransparencylab.org

apps and libraries that they want to integrate into their code base. Similarly, it is possible for DPAs to use such tools or make them available to a greater public. However, currently no studies exist on the uptake of such tools and their effective use for data protection. An evaluation of existing tools ought to also consider their uptake, usability, and effectiveness.

The research community should continue work in enhancing privacy and usability in the developer ecosystem, e.g. by empirical evaluation of IDEs, APIs and ad libraries.

The European Commission and EU institutions in the field of privacy and security should support relevant initiatives and research projects in this area.

6.5 Addressing the entire mobile app ecosystem

The design and the functionality of an app is not only dependent on app development methods, but on the entire mobile app ecosystem. This ecosystem relies on hardware, software, operating systems, protocols, APIs, infrastructures, contracts etc. As shown in the course of our analysis, the influence of app developers is often limited and, to a great extent, the rules of the ecosystem are determined by industry stakeholders, such as platform providers. For a comprehensive approach in protecting privacy and data protection for mobile app users, these overarching issues of governance must not be neglected. This big issue should be followed up in future work:

- **Building upon the knowledge of app developers, data controllers, regulators and researchers.** Those stakeholders who are aware of the privacy needs of users, the risks and the requirements from data protection legislation should document the current difficulties and deficiencies in implementing privacy and data protection in apps as far as these stem from the mobile app ecosystem. This hands-on experience from various perspectives is a valid input for identifying necessary or reasonable changes.
- **Defining and standardizing appropriate interfaces and protocols.** Standardization organisations as well as industry initiatives should take into account privacy and data protection requirements that influence the mobile app development, e.g. for the specification of interfaces or protocols.
- **Interdisciplinary approach for redesigning the mobile app ecosystem.** The current status of how the mobile app ecosystem works must not be taken for granted. For instance, today’s approach of offering “free” services in exchange of collecting personal data may be challenged: the principle of data protection by design and by default (art. 25 of the GDPR) can – if taken seriously –restrict data processing to the necessary extent and for specific purposes. Such possible changes may significantly affect the mobile app ecosystem with respect to governance (as well as the internet and digitisation in general). Researchers of multiple disciplines, practitioners, policy makers, and regulators should, thus, work on alternative approaches that enable privacy and data protection, as well as the implementation of other fundamental rights, in the system design for the future society. This work should encompass ways for improving the current digital ecosystem, as well as presenting supplementing approaches and it should propose migration paths towards envisioned options.

The European Commission, policy makers and regulators (e.g. Data Protection Authorities) should address the governance issues of today’s mobile app ecosystems with respect to maintaining privacy, data protection and other fundamental rights. Interdisciplinary research groups and practitioners should support this endeavour.

Annex A: Bibliography

[1] European Commission, *Regulation (EU) 679/2016 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,* 2016.

[2] European Commission, *Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data,* 1995.

[3] European Commission, *Directive 2002/58/EC concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications),* 2002.

[4] ENISA, “Privacy and data protection by design,” 2014.

[5] ENISA, “Smartphone secure development guidelines,” 2016.

[6] M. Gadaleta and M. Rossi, “IDNET: Smartphone-based Gait Recognition with Convolutional Neural Networks,” 2016.

[7] L. Olejnik, G. Acar, C. Castelluccia and C. Diaz, “The leaking battery: A privacy analysis of the HTML5 Battery Status API,” 2015.

[8] Z. Bauman and D. Lyon, “Liquid Surveillance: A Conversation,” 2013.

[9] J. Achara, V. Roca, C. Castelluccia and A. Francillon, “MobileAppScrutinator: A Simple yet Efficient Dynamic Analysis Approach for Detecting Privacy Leaks across Mobile OSs,” 2016.

[10] A. Kurtz, H. Gascon, T. Becker, G. Freiling and K. Rieck, “Fingerprinting Mobile Devices Using Personalized Configurations,” in *Proceedings on Privacy Enhancing Technologies*, 2016.

[11] J. Achara, M. Cunche, V. Roca and A. Francillon, “WifiLeaks: Underestimated Privacy Implications of the ACCESS_WIFI_STATE Android Permission,” in *7th ACM Conference on Security and Privacy in Wireless and Mobile Networks (ACM WiSec)*, 2014.

[12] OPEN EFFECT, “Every Step You Fake. A comparative Analysis of Fitness Tracker Privacy and Security,” 2016. [Online]. Available: https://citizenlab.org/2016/04/every-step-you-fake-final-report/. [Accessed 9 2017].

[13] Y.-A. de Montjoye, C. Hidalgo, M. Verleysen and V. Blondel, “Unique in the Crowd: The Privacy Bounds of Human Mobility,” 2013.

[14] J. Achara, G. Acs and C. Castelluccia, “On the Unicity of Smartphone Applications,” in *14th ACM CCS Workshop on Privacy in Electronic Society (ACM WPES)*, 2015.

[15] A. Blumberg and P. Eckersley, “On locational privacy and how to avoid losing it forever,” 2009.

[16] Y. Zou, J. Zhu and L. Hanzo, “A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends,” in *IEEE*, 2016.

[17] D. Arp, E. Quiring and C. Wressneger, “Privacy Threats through Ultrasonic SideChannels on Mobile Devices,” *IEEE Security and Privacy,* 2017.

[18] J. Brookman, P. Rouge and A. e. a. Alva, “Cross-Device Tracking: Measurement and Disclosures,” in *Proceedings on Privacy Enhancing Technologies (PETS2017)*, 2017.

[19] V. Mavroudis, S. Hao, Y. Fratantonio and e. al, “On the Privacy and Security of the Ultrasound Ecosystem.,” in *Proceedings on Privacy Enhancing Technologies*, 2017.

[20] S. Seneviratne, A. Seneviratne, P. Mohapatra and A. Mahanti, “Predicting user traits from a snapshot of apps installed on a smartphone,” *ACM SIGMOBILE Mobile Computing and Communications Review,* vol. 18, no. 2, p. 1–8, 2014.

[21] W. Melicher, M. L. Mazurek, D. Kurilova, S. M. Segreti, P. Kalvani, R. Shay, B. Ur, L. Bauer, N. Christin and L. Cranor, “Usability and security of text passwords on mobile devices,” in *34th Annual ACM Conference on Human Factors in Computing Systems*, 2016.

[22] UK Information Commissioner's Office, “Privacy in mobile apps: guidance for developers,” 2013.

[23] N. Vallina-Rodriguez, S. Sundaresan, A. Razaghpanah, R. Nithyanand, M. Allman, C. Kreibich and P. Gill, “Tracking the Trackers: Towards Understanding the Mobile Advertising and Tracking Ecosystem,” 2016.

[24] D. Leibenger, F. Möllers, A. Petrlic, R. Petrlic and C. Sorge, “Privacy Challenges in the Quantified Self Movement – An EU Perspective,” in *Privacy Enhancing Technologies*, 2016.

[25] Article 29 Data Protection Working Party, *Opinion 13/2011: Geolocation services on smart mobile devices,* 2011.

[26] Article 29 Data Protection Working Party, “Opinion 2/2013: Apps on smart devices,” 2013.

[27] Federal Trade Commission, “Mobile privacy disclosures,” 2013.

[28] US National Telecommunications and Information Administration, “Privacy Multistakeholder Process: Mobile Application Transparency,” 2013.

[29] California Attorney General's Office, “Privacy on the Go: Recommendations for the mobile ecosystem,” 2013.

[30] Dutch Data Protection Authority, “Investigation into the processing of personal data for the ‘whatsapp’ mobile application by Whatsapp Inc,” 2013.

[31] C. Osborne., “Google plans purge of Play Store apps without privacy policies,” ZDNet, 2017.

[32] M. Pelikan, G. Hogben and U. Erlingsson, “Identifying Intrusive Mobile Apps using Peer Group Analysis,” 2017.

[33] Article 29 Data Protection Working Party, *Guidelines on Data Protection Impact Assessment (DPIA) and determining whether processing is “likely to result in a high risk” for the purposes of Regulation 2016/679,* 2017.

[34] CNIL, “Privacy Impact Assessment,” 2015.

[35] F. Bieker, M. Friedewald, M. Hansen, H. Obersteller and M. Rost, “A Process for Data Protection Impact Assessment under the European General Data Protection Regulation. In: Privacy Technologies and Policy,” in *4th Annual Privacy Forum (APF 2016)*, 2016.

[36] ENISA, “Guidelines for SMEs on the Security of Personal Data Processing,” 2016.

[37] S. J. De and D. L. Metayer, “A Privacy Risk Analysis Methodology,” Grenoble Rhone-Alpes, 2016.

[38] NIST, “An Introduction to Privacy Engineering and Risk Management in Federal Systems,” 2017.

[39] A. Sheth, S. Seshan and D. Wetherall, “Geo-fencing: Confining Wi-Fi Coverage to Physical Boundaries,” in *7th International Conference on Pervasive Computing (Pervasive '09)*, 2009.

[40] J. Vincent, “99.6 percent of new smartphones run Android or iOS,” 16 February 2017. [Online]. Available: https://www.theverge.com/2017/2/16/14634656/android-ios-market-share-blackberry-2016. [Accessed 14 October 2017].

[41] A. Yerukhimovich, R. Balebako, A. E. Boustead, R. K. Cunningham, W. Welser and R. Housley, “Can Smartphones and Privacy Coexist?,” RAND Corporation, 2016.

[42] Y. Acar, M. Backes, S. Bugiel and M. Smith, “SoK: Lessons Learned from Android Security Research for Appified Software Platforms,” in *IEEE Symposium on Security and Privacy (SP)*, 2016.

[43] S. Fahl, M. Harbach, H. Perl, M. Koetter and M. Smith, “Rethinking SSL development in an appified world,” in *CCS '13 Proceedings of the 2013 ACM SIGSAC conference on computer & communications security*, 2013.

[44] C. Gibler, J. Crussell, J. Erickson and H. Chen, “AndroidLeaks: automatically detecting potential privacy leaks in android applications on a large scale,” in *TRUST'12: Proceedings of the 5th international conference on Trust and Trustworthy Computing*, 2012.

[45] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau and P. McDaniel, “FlowDroid: precise context, flow, field, object-sensitive and lifecycle-aware taint

analysis for Android apps," in *35th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI '14*, 2014.

[46] A. Porter, M. Finifter, E. Chin, S. Hanna and D. Wagner, "A survey of mobile malware in the wild," in *SPSM '11 Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile device*, 2011.

[47] G. Michael, Z. Yajin, W. Zhi and J. Xuxian, "Systematic Detection of Capability Leaks in Stock Android Smartphones," in *NDSS Symposium*, 2012.

[48] European Commission, *Commission staff working document impact assessment accompanying the document proposal for Regulation of the European Parliament and the Council concerning the respect for private life and the protection of personal data in electronic communications,* 2017.

[49] J. Godfrey, B. Courtney and N. Miller, "State of the App Economy," 2016.

[50] E. Papadopoulos, M. Diamantaris, P. Papadopoulos, T. Petsas, S. Iannadis and E. Markatos, "The long-standing privacy debate: Mobile Websites vs. Mobile Apps," in *Proceedings of IW3C2*, 2017.

[51] L. A. Bygrave, "Data Protection by Design and by Default: Deciphering the EU's Legislative Requirements," vol. 4, no. 2, 20 June 2017.

[52] S. Gürses, C. Troncoso and C. Diaz, "Engineering privacy by design," in *Computers, Privacy and Data Protection Conference (CPDP)*, 2011.

[53] S. Gurses, C. Troncoso and C. Diaz, "Engineering privacy by design reloaded," in *Amsterdam Privacy Conference*, 2015.

[54] C. Kalloniatis, E. Kavakli and S. Gritzalis, "Addressing privacy requirements in system design: the PriS method," *Requirements Engineering,* vol. 13, no. 3, pp. 241-255, 2008.

[55] J.-H. Hoepman, "Privacy design strategies," in *IFIP International Information Security Conference*, 2014.

[56] M. Deng, K. Wuyts, R. Scandariato, B. Preneel and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements," *Requirements Engineering,* vol. 16, no. 1, pp. 3-32, 2011.

[57] S. Gürses and J. V. Hoboken, "Privacy After the Agile Turn," 2017.

[58] T. D. Breaux, H. Hibshi and A. Rao, "Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements," *Requirements Engineering,* vol. 19, no. 3, pp. 281-307, 2014.

[59] S. Sen, S. Guha, A. Datta, S. K. Rajamani, J. Tsai and J. M. Wing, "Bootstrapping privacy compliance in big data systems," in *IEEE Symposium on Security and Privacy (SP)*, 2014.

[60] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel and A. N. Sheth, “TaintDroid: An Information Flow Tracking System for Real-Time Privacy Monitoring on Smartphones,” *Communications,* vol. 57, no. 3, pp. 99-106, March 2014.

[61] P. Hornyack, S. Han, J. Jung, S. Schechter and D. Wetherall, “These aren't the droids you're looking for: retrofitting android to protect data from imperious applications,” in *18th ACM conference on Computer and communications security* , 2015.

[62] Z. Yang, M. Yang, Y. Zhang, G. Gu, P. Ning and X. S. Wang, “AppIntent: analyzing sensitive data transmission in android for privacy leakage detection,” in *SIGSAC conference on Computer & communications security* , 2013.

[63] M. Ongtang, S. McLaughlin, W. Enck and P. McDaniel, “Semantically rich application-centric security in Android,” in *Annual Computer Security Applications Conference, ACSAC '09*, 2009.

[64] R. Xu, H. Saidi and R. Anderson, “Aurasium: Practical policy enforcement for Android applications,” in *21st USENIX Security Symposium*, 2012.

[65] M. Nauman, S. Khan and X. Zhang, “Apex: extending Android permission model and enforcement with user-defined runtime constraints,” in *5th ACM Symposium on Information, Computer and Communications Security, ASIACCS '10*, 2010.

[66] “ComDroid,” [Online]. Available: http://comdroid.net/. [Accessed 9 2017].

[67] Z. O. F. Zhao, “Trustdroid: Preventing the use of smartphones for information leaking in corporate networks through the use of static analysis taint tracking.,” *MALWARE, IEEE* , p. 135–143, 2012.

[68] A. Porter Felt, E. Ha, S. Egelman, A. Haney, E. Chin and D. Wagner, “Android Permissions: User Attention, Comprehension, and Behavior,” in *Symposium on Usable Privacy and Security (SOUPS)* , 2012.

[69] C. Spensky, J. Stewart, A. Yerukhimovich, R. Shay, A. Trachtenberg, R. Housley and R. Cunningham, “SoK: Privacy on Mobile Devices–It’s Complicated,” in *Privacy Enhancing Technologies (PETs)*, 2016.

[70] Y. Lancet, “What is CarrierIQ and how do I know if I have it?,” 2011. [Online]. Available: http://www.makeuseof.com/tag/carrier-iq/.

[71] K. Waddell, “When apps secretly team up to steal your data,” 7 April 2017. [Online]. Available: https://www.theatlantic.com/technology/archive/2017/04/when-apps-collude-to-steal-your-data/522177/. [Accessed October 2017].

[72] “Goluk app,” Beijing Mobnote Technology Co., Ltd, 9 2017. [Online]. Available: https://play.google.com/store/apps/details?id=com.mobnote.golukmobile&hl=en. [Accessed October 2017].

[73] European Data Protection Supervisor, “Guidelines on the protection of personal data processed by mobile applications by European Union institutions,” 2016.

[74] P. Wijesekera, A. Baokar, A. Hosseini, S. Egelman, D. Wagner and K. Beznosov, “Android Permissions Remystified: A Field Study on Contextual Integrity,” in *USENIX Security Symposium*, 2015.

[75] A. Cavoukian, “Privacy by design – the 7 foundational principles (revised version),” 2011.

[76] M. Hansen, J.-H. Hoepman and M. Jensen, “Towards Measuring Maturity of Privacy-Enhancing Technologies,” in *Annual Privacy Forum (APF 2015)*, 2016.

[77] M. Hansen, M. Jensen and M. Rost, “Protection Goals for Privacy Engineering,” in *International Workshop on Privacy Engineering (IWPE), Security and Privacy Workshops (SPW)*, 2015.

[78] German DPA, “Standard Data Protection Model,” in *German Conference of Data Protection Authorities*, 2017.

[79] M. Colesky, J.-H. Hoepman and C. Hillen, “A Critical Analysis of Privacy Design Strategies,” in *International Workshop on Privacy Engineering – IWPE'16*, San Jose, CA, USA, 2016.

[80] J. H. Hoepman., “Privacy Design Strategies,” in *IFIP TC11 29th Int. Conf. on Information Security (IFIP SEC 2014)*, 2014.

[81] ISO, “ISO/IEC 27001:2013 Information technology-Security techniques - Information security management systems -- Requirements,” 2013.

[82] M. Hansen, “Data Protection by Design and by Default à la European General Data Protection Regulation,” in *11th IFIP Summer School on Privacy and Identity Management*, Heidelberg , 2017.

[83] M. Hansen, J.-H. Hoepman and M. Jensen, “Readiness analysis for the adoption and evolution of privacy enhancing technologies,” ENISA, 2015.

[84] A. Yasemin, M. Backes, S. Fahl, D. Kim, M. L. Mazurek and C. Stransky, “You Get Where You're Looking For: The Impact Of Information Sources on Code Security,” in *Security and Privacy (SP), 2016 IEEE Symposium*, 2016.

[85] UK Information Commissioner's Office, “Conducting privacy impact assessment code of practice,” 2015.

[86] Australian Government, Office of the Australian Information Commissioner, “Guide to Understanding Privacy Impact Assessments,” 2014.

[87] Article 29 Data Protection Working Party, “Opinion 9/2011 Privacy and Data Protection Impact Assessment Framework for RFID Applications,” 2011.

[88] “The Lumen tool,” 2017. [Online]. Available: https://haystack.mobi. [Accessed 9 2017].

[89] “The Appcensus tool,” 9 2017. [Online]. Available: https://www.appcensus.mobi. [Accessed 9 2017].

[90] F. Z. Borgesius, J. v. Hoboken, R. Fahy, K. Irion and M. Rozendaal, “An assessment of the Commission’s Proposal on Privacy and Electronic Communications, Study for the the European Parliament LIBE Committee,” 2017.

[91] M. Ikram, N. V. Rodriguez, S. Seneviratne, M. Kaafar and V. Paxon, “An analysis of the Privacy and Security Risks of Android VPN Permission-enabled Apps,” in *ACM IMC*, 2016.

ENISA
European Union Agency for Network
and Information Security
Science and Technology Park of Crete (ITE)
Vassilika Vouton, 700 13, Heraklion, Greece

Athens Office
1 Vasilissis Sofias
Marousi 151 24, Attiki, Greece

TP-07-17-067-EN-N

PO Box 1309, 710 01 Heraklion, Greece
Tel: +30 28 14 40 9710
info@enisa.europa.eu
www.enisa.europa.eu

ISBN: 978-92-9204-242-4,
DOI: 10.2824/114584