WIPO
WORLD
INTELLECTUAL PROPERTY
ORGANIZATION

WIPO HANDBOOK ON KEY CONTRACTS FOR MOBILE APPLICATIONS - A DEVELOPER'S PERSPECTIVE

OCTOBER 2020

TABLE OF CONTENTS

INTRODUCTION

Mobile apps have become an integral part of people's lives, their popularity having increased exponentially with the uptake of smartphones. The market value of the mobile app economy has also grown exponentially, driven by a huge community of software developers[1].

Statistics abound on how mobile apps pervade every walk of life, from health, entertainment, banking and financial services to just about anything the developer can imagine. Mobile apps have become an essential tool through which a company's customers access products and services. Creative app development is a global phenomenon, found in most countries around the world. It is therefore essential for companies to remain within the legal frameworks, which differ, of all the jurisdictions in which they operate.

This new handbook of the World Intellectual Property Organization (WIPO) is intended to complement other WIPO materials that pertain to mobile applications and IP. It provides a practical tool to help one particular segment of the mobile app market sector: app developer organizations, alerting them to the issues that need to be addressed at the various stages of the mobile app life cycle. It will highlight potential problem areas that can be solved through effective contracts and other legal structures.

This handbook will give developer organizations a thorough grounding in the basics of IP contracts, before they resort to specialized professionals.

It is much more effective in both time and cost to protect, from the outset, all intellectual property created during app development. Clarifying, at an early stage, the framework for interaction between the parties reduces the risk of potential misunderstandings, freeing up both to focus on the revenue-enhancing aspects of their business relationship.

Where disputes do arise, parties can turn to WIPO's Alternative Dispute Resolution service, available at [Insert link], designed to provide a streamlined, cost-effective resolution process.

This handbook is based on a review of the literature as well as the authors' own practical experience in the field. It takes a pragmatic approach, highlighting issues to be considered in relation to key elements of a contract or other legal requirements. It then provides examples of typical key contract clauses and concludes with a checklist of key issues to consider and steps to take.

1 WIPO – Intellectual Property and Mobile Applications - January 2018 - https://www.wipo.int/export/sites/www/ip-development/en/agenda/pdf/ip_and_mobile_applications_study.pdf

Who this Handbook is for and what it won't provide?

This handbook is designed to help small or medium organizations engaged in the development of mobile apps negotiate a hassle-free path through the IP and other legal issues they can expect to encounter, pinpointing the issues to be considered when embarking on app development and at each stage thereafter.

From the time they first conceive of a new app up to the time it is made available to end users, the app's developers will interact with many individuals and organizations, including software and coding service agencies, app stores, providers of hosting services and providers of third-party services, such as payment processing and data feeds. Developers interact most importantly with the app's users. Contracts need to be put in place with all these organizations and individuals.

In addition to contracts, the development and deployment of apps also involves issues around the creation, acquisition, protection and licensing of various forms of intellectual property. Questions also arise out of regulations and legal liabilities.

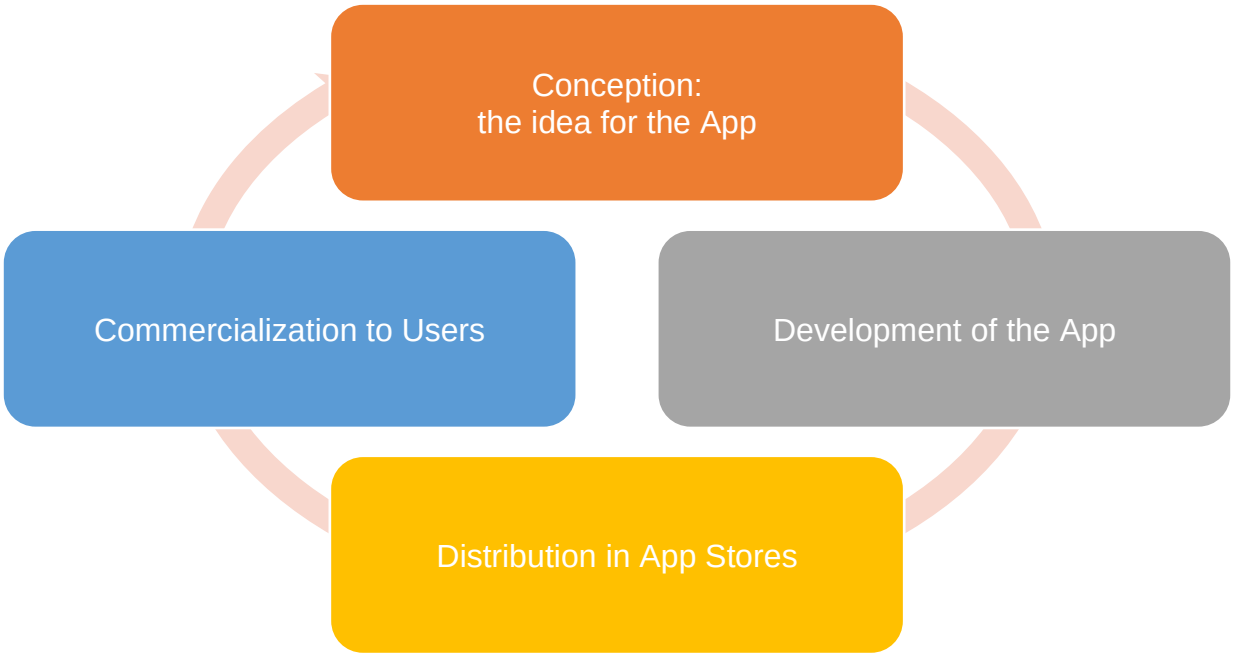
This handbook will provide a high-level overview of all such issues, highlighting practical aspects for consideration throughout an app's life cycle, from conception and development through launch and implementation.

This handbook cannot provide country-specific legal advice or advice relating to specific circumstances. Local specialized guidance should be sought for any such particular concerns.

Life cycle of a mobile app from the perspective of a developer/owner

The mobile app ecosystem is complex. We have set out below a typical life cycle for a high-level mobile app – from conception and development through distribution via app stores. We examine subsequently the contracts needed at each stage.

Figure 1 - Life cycle of a mobile app



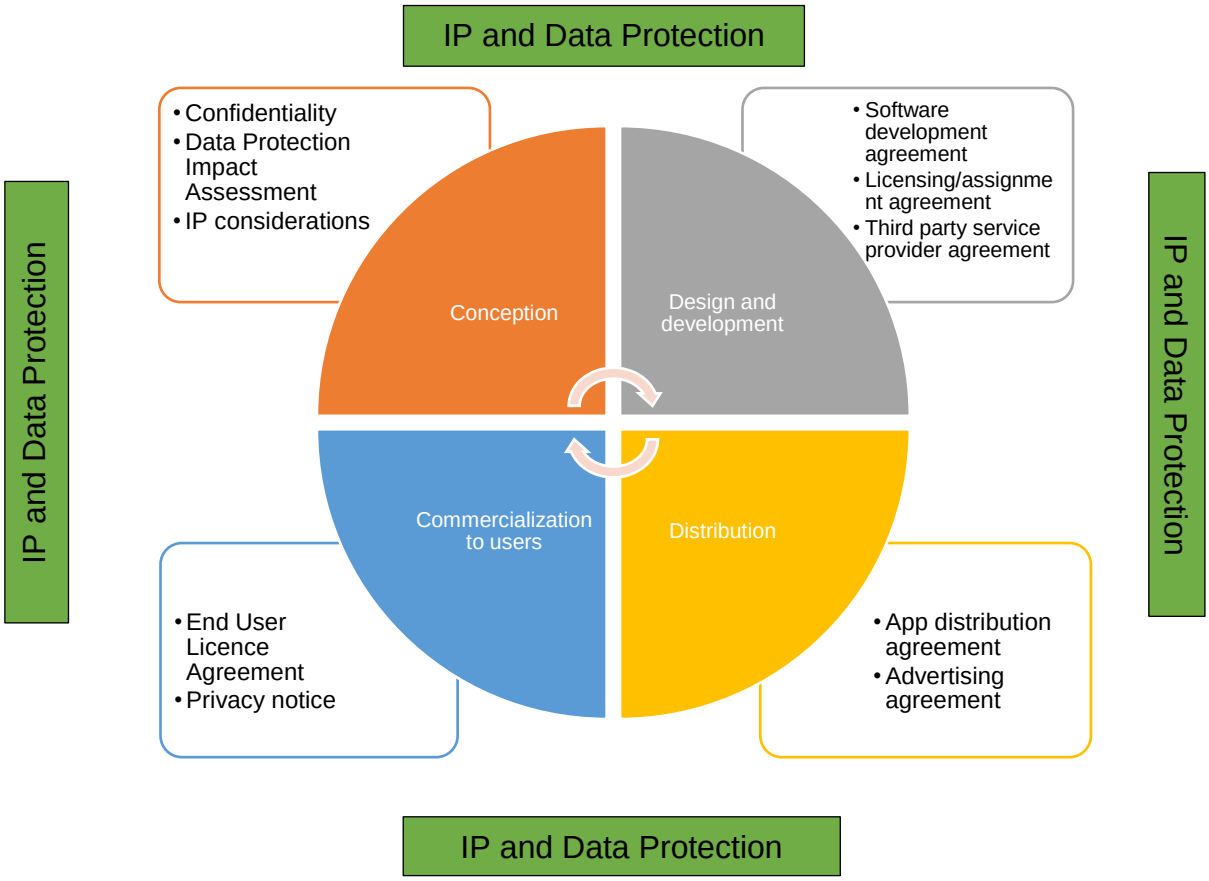
Relevant key contracts and issues for mobile apps

Listed below are some of the key contracts and issues developers may have to deal with during the mobile app life cycle. Key points to bear in mind when dealing with each are covered in the sections below.

- Intellectual property rights of relevance throughout an app's life cycle
- Non-Disclosure Agreements
- Developer contracts with programmers and agencies
- Software development agreements
- Third-party service provider agreements (including hosting)
- App store hosting and distribution agreements
- Advertising agreements and use of third-party IP (e.g., image rights, characters)
- End user license agreements and terms of use (including consumer law requirements)
- Data protection and privacy policies

An app consists of software code, associated content (such as text, graphics and music) and data, which may include personal data. These materials are regarded at law as intellectual property and are also (in the case of personal data) regulated under the data protection regimes of various countries. This is illustrated in the diagram below, summarizing key contracts and legal issues at each stage of the mobile app life cycle.

Figure 2 - Key contracts and documents flowchart



As shown, IP and data protection issues arise, and must therefore be considered and continually reviewed, at each stage of the mobile app life cycle. The concept behind a new app can be expressed through coding, interface design, added creative elements and connection to a database. Various IP rights may arise at any stage and need to be considered when they do.

During the development phase, various parties may be involved in creating the different components, including coders and designers. The question of who owns the IP becomes important from this point on, with developers usually wanting to own, or at least have a generous license to, the newly created IP. IP is also important as the app is commercialized and becomes popular. Paying close attention to the IP situation will facilitate eventual sales or investor decisions to inject more capital into the business, as detailed further in the chapters to follow.

Data protection is an issue throughout a mobile app's life cycle, generating obligations that need to be understood and reflected in the app's design. Waiting to address data protection issues at a later stage may necessitate extensive reworking of the product and unnecessary costs. By understanding the requirements of data protection, a developer can build compliant working practices with third-party providers into the app. Personal data will almost certainly be passed on to a party's suppliers for services

such as hosting or payments. By respecting the requirements of data protection legislation, developers can make the rights of data subjects an integral part of the app from its earliest design stage and avoid the substantial regulatory fines and brand damage that can result from a breach of such laws. These topics will be dealt with in greater detail in Chapter 9.

CHAPTER 1 - GENERAL OVERVIEW OF IP RIGHTS THROUGHOUT THE LIFE CYCLE OF A MOBILE APP

Introduction

In this chapter we will assume a basic understanding of what Intellectual Property (IP) rights are and the various such rights available. A short checklist is provided below as a recap. For further reading, please consult the WIPO "Intellectual Property for Business" series, which provides a number of introductory guides on the use and benefits of the various forms of IP. WIPO has also published a study on IP and mobile apps, which gives an in-depth analysis of the nature, scope, complexity and costs of the IP legal ecosystem for mobile apps[2].

Various IP rights come into play in the development and use of mobile apps – and various risks that developers should be aware of. Developers should think about IP in terms of the computer program but also other aspects of the app, such as images, text, sound, video and design of the app icon, all of which may be protected separately. It is therefore essential to keep IP in mind throughout the app's life cycle. IP must be an ongoing, hands-on concern since IP rights can arise at different times.

Internationally recognized IP rights

Copyright

Copyright is a fundamental IP right. It protects a newly created work – such as text, images (including photos, artwork and moving images), music and computer software – from being copied or exploited by anyone other than the copyright holder without the latter's permission.

This means that, unless an exemption applies, an existing work can only be copied, or a derivative of it created, by its owner or licensee.

At international level the framework for copyright protection is established by several instruments, the Berne Convention being the most widely-recognized.

But at local level the extent to which a particular work (or part of a work) can be protected by copyright varies from country to country. In this context, works consist mainly of software or computer programs but can also include such other aspects of an app as music, images or text. Computer code, whether in source code or object code, is protected by the Berne Convention as a literary work.

[2] WIPO - Intellectual Property and Mobile Applications - WIPO - January 2018 - https://www.wipo.int/export/sites/www/ip-development/en/agenda/pdf/ip_and_mobile_applications_study.pdf

Under the Berne Convention, copyright comes into existence automatically upon the creation of a work (in countries to which the Convention applies, which in practice is almost all countries). Registration is not needed for copyright to arise.

A few countries, however, do have copyright registration systems, since registration of a work typically makes it easier for copyright owners to pursue infringement claims in those countries. While works often display copyright notices (for example, "copyright © 2020 XYZ Corporation"), these are not legally required under the Berne Convention. And even where copyright notices or registration are not required, it is helpful for developers to date and put copyright notices on their materials so as to establish an audit trail for potential infringement claims. For this reason, copyright notices should be placed in the relevant "read-me" file and in the documentation accompanying an app. It is also helpful to encode them in the software.

Copyright may protect screen displays, designs, icons, on-screen text and other creative elements of an app.

Alongside copyright, authors also have specific "moral rights", such as the right to attribution and the right to prevent a work from being treated in a way that is prejudicial to the author. Unlike copyright, such rights are not transferable and remain with the author even if the copyright is transferred (in some countries, such moral rights will transfer to the author's heirs on death). This is considered in greater depth in Chapter 3 below.

Some jurisdictions, most notably within the EU, have a separate intellectual property regime (also applying automatically) for databases. Determining the applicability of database rights, however, is complex, and local advice should be sought.

Patent

Developers should check to determine if aspects of the app being developed, such as some novel and inventive functionality, can be patented. It is often the functionality that can be patented and not the software itself.

Functionality has a specific meaning in this context: it must add a truly new element to the end user's experience with the app, which usually means more than just causing a computer to execute a number of software commands. In many jurisdictions the functionality must have some technical effect in the real world beyond what happens on the screen of the device.

In some countries, patents can be issued for software/business methods that provide solutions to technical problems and are novel and inventive.

Patent notices should also appear in the relevant "read-me" file, and user manuals, if provided, should be encoded in the software (in a help/about menu, for example).

The extent to which software-implemented inventions can be protected by patent varies greatly from jurisdiction to jurisdiction, so the harmonization of substantive provisions of patent law is challenging. Effective patent protection worldwide has been greatly simplified, however, by WIPO's Patent Cooperation Treaty (PCT), which can help applicants obtain international patent protection in numerous countries through a single international patent application.

Unlike copyrights, which are usually deemed to be infringed only if an existing work has been copied or a derivative thereof created, patents can be infringed even if the infringer does not know of its existence and does not copy anything. This is a complex area, so in case of doubt a qualified patent lawyer should be consulted.

Trademarks

Another method of protecting the IP in an app is to register its name, logo, icons, distinctive images, slogans and characters as trademarks. The rules governing what can and cannot be registered are fairly complex. For example, a name that is purely descriptive, such as "The Accounting App", cannot usually be registered. A qualified trademark lawyer can help to determine whether a proposed name or logo can be trademarked and perform searches to find potentially conflicting marks.

A trademark lawyer can also help to register marks in territories around the world where protection is being sought. WIPO's Madrid System allows a single application to be used to obtain protection in many countries at once. This often proves to be a cost-effective trademark strategy.

Developers should ideally consider registering trademarks before launching their apps. An initial trademark search, followed by a trademark application strategy, will help to highlight any protection issues (e.g., conflicting marks or registration issues) at an early stage. In some countries, especially those with common law systems, unregistered trademarks can be protected through legal action, usually known as "passing off". Such action provides only limited protection, however. Bear in mind also that a registered trademark, when associated with a successful product, can be a valuable asset in itself.

The market for new apps is highly dynamic. Even developers who prefer not to apply for a trademark should at least conduct searches to make sure that the proposed name for their app, or something similar, has not already been trademarked. This needs to be done separately in all countries where the app is to be made available.

Like trademarks, domain names can also be protected. While not in itself a form of intellectual property, an appropriate domain name can provide protection by preventing others from taking the domain in question. Domain name registration should therefore form part of a brand protection policy.

Design Rights

Design rights (some countries call them "design patents") are bundled intellectual property rights offering an intermediate form of protection, somewhere between copyrights and patents. In the context of app development, these typically cover graphic works such as logos and type-faces. But the scope of design rights is not uniform worldwide. Some countries require a novel feature or unique character – a concept that mirrors patent law. Local laws may also provide automatic protection for unregistered designs, although proving infringement may be more difficult. As a general rule, developers should register such design elements where allowed in particular countries.

Due diligence as to suppliers

Whenever engaging suppliers or others to work on an app, it is important to ensure that they can deliver on the following key questions. Are they qualified to perform the work required? Do they have the appropriate resources and skills? Do they understand the importance of obtaining appropriate IP on behalf of the developer?

Due diligence in screening suppliers is thus an important part of the selection process.

Suppliers need to understand what IP the developer requires. Is the IP to be held under license or will it be assigned? Do the suppliers hold IP rights themselves that they can pass on to the developer? Do the suppliers have appropriate contracts with their own contractors and are they aware about the importance of IP?

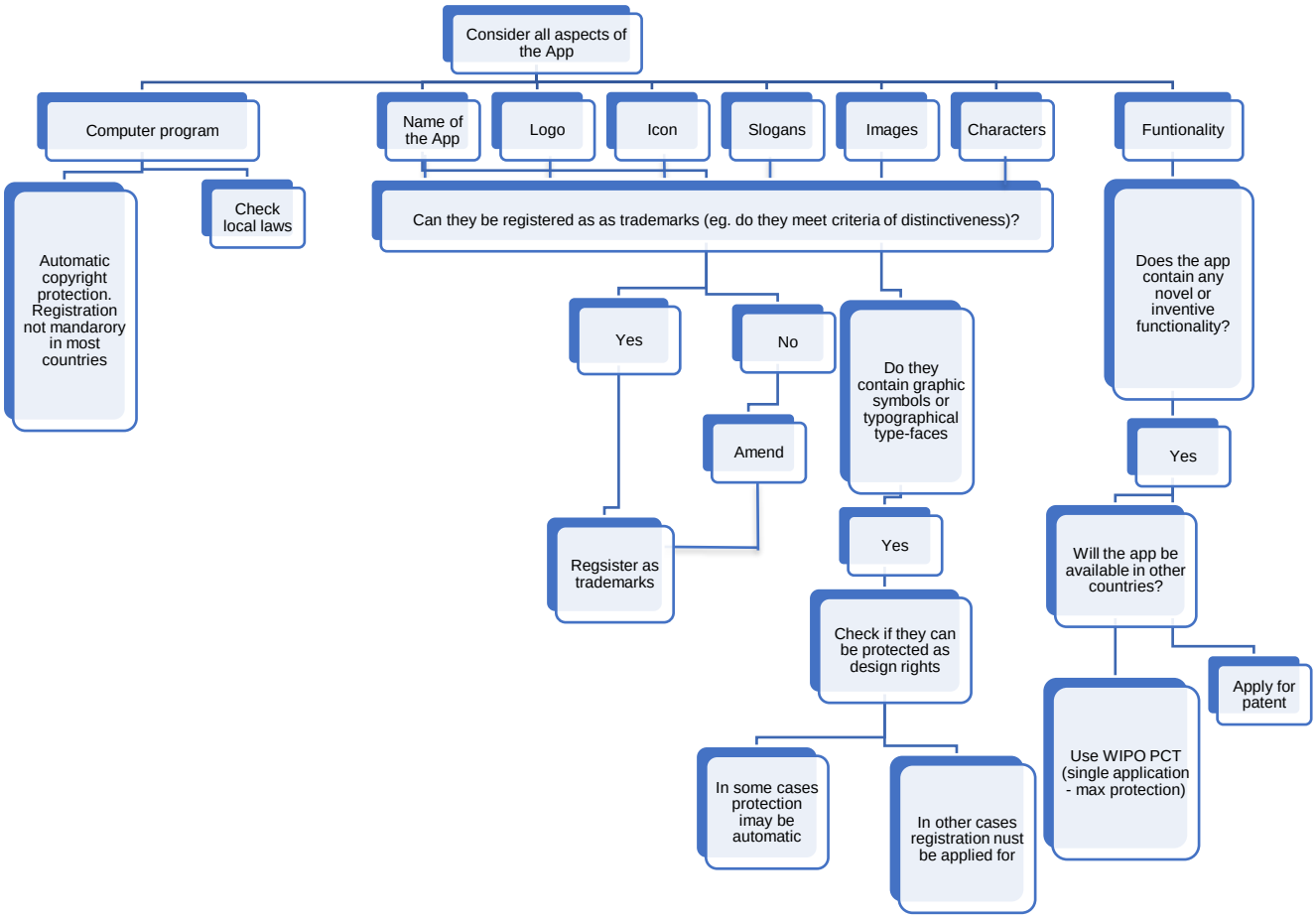
Such due diligence should be performed before entering into contract negotiations, enabling the developer to weed out inappropriate suppliers before major problems occur. See chapter 4 for more information about due diligence.

Checklist

- Identify the various elements of the app.
- Identify which of these elements can be protected by IP rights under local laws. Protections include copyright, trademarks, design rights and database rights. Complete such formalities as required.
- Consider registering trademarks, trade names and logos for the app, preferably before launch. If this is not feasible, for financial or other reasons, consider at least carrying out searches. Also consider obtaining domain names.
- Ask whether a database has been created. Has work been done to select and arrange the data in a unique way, potentially generating database rights?
- Check if such rights arise automatically and keep appropriate documentation as an audit trail. Prioritize the rights that require registration and have the most value.

- Perform due diligence on suppliers and the IP they provide to minimize the risk of infringement and potentially expensive infringement claims.

Figure 3 – IP flowchart



CHAPTER 2 - NON-DISCLOSURE AGREEMENTS

Introduction

Planning and launching new software is often a sensitive activity. Accordingly, confidentiality or non-disclosure agreements (NDAs) are widely used in the software development industry. NDAs can be useful for the following purposes:

- **To protect trade secrets** and thus a developer's business from competitors. Such secrets may pertain to processes and procedures, marketing schemes or development strategies.

- **To protect confidential information** shared between parties during development of the app, such as passwords, customer lists, blueprints, prototypes, source code, software products, business plans, analytical data, etc.

- **To minimize risks when working with external specialists (contractors and third-party agencies),** who may also be involved with the developer's competitors.

- **To operate in "stealth mode",** keeping a product secret from the public while under development.

Considerations

Consider whether information requires protection, and if so, enter into an NDA before it is shared.

Key clauses

- Scope of confidential information

This clause defines what information is to be considered confidential for the purpose of the NDA, identifying as well information to be excluded from confidential treatment.

- Purpose of project

This clause sets out the purpose for which confidential information may be disclosed by either party.

- Permitted disclosures
-

This clause identifies companies or individuals to whom disclosures are permitted, giving both sides greater control over the protection of confidential information. Such clauses often permit disclosure to a professional adviser, provided that the adviser is

aware of the confidentiality obligations and does not transfer confidential information to others.

- Mutuality of obligations

Where both parties share confidential information, it is better to have a two-way, reciprocal NDA, assigning to both parties the same obligations in respect of each other's confidential information.

- Duration

A duration or term clause defines how long the NDA remains valid (typically 1-6 years). It also details the requirement for both parties to return or destroy confidential information upon termination of the agreement.

Checklist

- Consider whether an NDA should be signed, as would be advisable, for instance, when:
  - there is confidential information to protect
  - the project is to remain confidential
  - contractor developers or third-party agencies are concerned
- NDAs are not usually necessary:
  - during the developer's search for a contractor
  - when the information concerned is in the public domain

CHAPTER 3 - CONTRACTS, ASSIGNMENT AGREEMENTS, IP AND THIRD-PARTY DEVELOPERS

Introduction

Developers often use their own in-house employees to work on apps but may also have to outsource elements of app development to specialized individuals or agencies – perhaps to cover needs for additional staffing or technical or creative inputs.

In most jurisdictions, when an agency develops a software element for a mobile app on behalf of a developer, the agency, not the developer, owns the IP in that piece of work – by default.

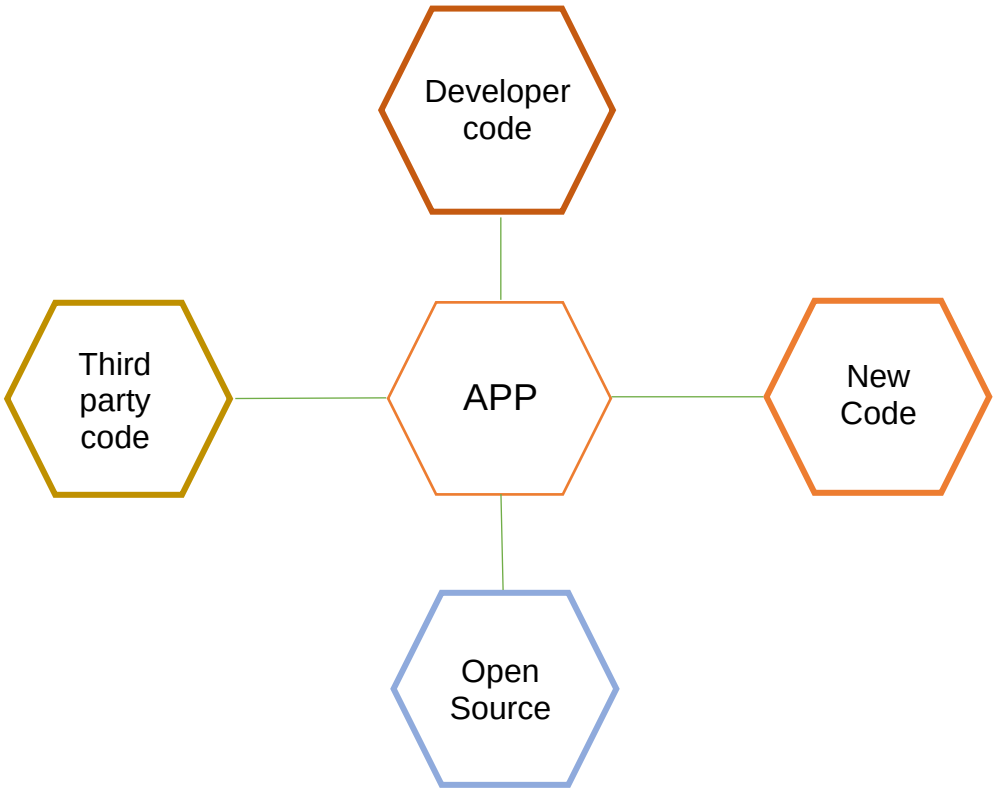
This usually means that developers need to do one of two things:

(i) either have the newly created IP assigned to them; or
(ii) obtain a license for the IP.

Analyzing the differences between assignment and licensing

To better analyze these differences here is a quick recap of what an app typically consists of. The IP rights associated with each type of software need to be legally established.

Figure 4 - Composition of a typical app



So, how do assignments and license agreements differ?

In simple terms, "assignment" is a transfer of IP ownership. Where an agency writes the code for an app, the agency owns the rights to that code as soon as it is created, with complete freedom to use the software as it wishes. The developer at that point has no rights to the code and might therefore insist on an assignment or license.

If the agency assigns the code, it relinquishes ownership to the developer, which then owns all rights to the code (the agency retaining none).

By contrast, "licensing" IP means granting the right to use it (with the agency retaining ownership). Such rights to use IP are often subject to restrictions and limits.

Where the ownership and licensing of IP is concerned, the interests of developers and agencies often differ.

Agencies usually prefer not to assign IP rights to developers but to license the software to them. By retaining IP rights agencies can reuse the same code and build up a bank of standard code components for other customers.

The ideal for developers is assignment of the IP, so that they "own" the software outright, having paid for its development. If the IP is properly assigned, the developer becomes its new owner and can exploit it, making the app available to users or selling or licensing the IP behind it as it sees fit. Crucially, owners of IP can make changes to the app without being in breach of any IP laws. Under license arrangements, the copyright owner's consent is often required to make changes. Recurring license fees may also be payable.

Developers should therefore seek assignment of the IP in new code.

An agency may say it cannot assign IP rights, because the app created for a customer incorporates some of the agency's standard code bank, which has already been used for other customers. It may also want to add the new code developed in creating an app to its code bank. A compromise solution is to assign "foreground rights" and "background rights". Foreground rights are the rights to code and other materials that are specific to a particular app. They give the developer a special market-differentiating advantage. They include graphics displaying the developer's logo and trademarks, graphics developed especially for the app, and code providing the app with special functionality. "Background rights" are rights to generally used code and materials not specific to the app in question.

Developers sometimes negotiate to make certain items subject to foreground rather than background rights, at extra cost, since the agency cannot then reuse them.

For developers entering into license agreements, important issues must be negotiated and agreed under the license contract. To permit the widest usage rights possible, the license terms should:

- not have a time limit;
- grant worldwide rights;
- allow the developer to amend and modify the software;
- allow the developer to sub-license to another developer or agency (for example where the other developer or agency specializes in a different platform or app store);
- be fully transferable; and
- allow the developer to pursue claims against third parties who infringe the former's copyright.

Third-party code

Agencies frequently use third-party code in their software (for examples, games are often built on a "game engine" which provides the framework on which a game can be built). In addition to entering into a license agreement with the agency they have commissioned, the developer must also ensure that further usage licenses are obtained for third-party code.

It should be noted that the concept of assignment as discussed here refers to assignment of the IP in newly created code. Where third-party code or open source software are involved, assignment is unlikely to be relevant.

Key contract terms which need to be agreed in a development agreement

The following are some key considerations to bear in mind, ideally before the development process starts. If there is a development agreement in place with a third-party contractor or agency, assignment can be addressed there together with other aspects of the projects such as the deliverables or price. Please refer to Chapter 4 on software development agreements for more details.

Considerations

- Are the developer's staff employees or contractors?

In some countries it matters from an IP perspective whether the individual who creates the app is an employee of the developer or a third-party contractor. In many jurisdictions, the IP in an app which has been created by a developer's employees would automatically belong to their employer: the developer. Despite this blanket protection, employment contracts often cover this point expressly, which is advisable in cases where the laws could change, or where questions arise as to whether an employee created the code in the course of employment. In such cases, the developer organization would still be protected.

Assignment is often not automatic where apps are developed by contractors working on their own account. In this case, in order to secure ownership of the IP concerned, the commissioning developer should obtain assignment of the IP from the coders. This is usually done by means of a written agreement, typically called an assignment (sale) agreement. Alternatively, assignment can be dealt with as part of the software

development agreement (see next chapter). The considerations and key clauses below would be relevant in both cases – whether assignment is dealt with through a stand-alone assignment agreement or the software development agreement.

– Consideration for IP assignment

In some jurisdictions the price for IP is charged separately from that for the development work. The developer is considered to be buying two different "products". In other jurisdictions a single price is charged, which includes assignment. Care must be taken to clarify which of these regimes applies to a specific contract or agreement to avoid unexpected claims for additional compensation, a point that may need to be clarified in the wording of the agreement. Tax issues may also arise, so specialized tax advice should be considered.

- Use of third-party agencies and the transfer of IP

It is important for developers resorting to third-party agencies to deal with IP assignment at an early stage, so that unexpected costs do not arise later – particularly where third-party agencies have strong bargaining power. A clause is used in some jurisdictions to ensure that copyrights for code and other materials belong automatically, upon creation, to the commissioning developer. In others, the clause establishes an agreement for future assignment of such copyrights, which the developer can then enforce later.

- Use and incorporation of open source (OS) code and compliance with OS licenses.

OS code (sometimes called free software) is software that is available for use, modification or distribution by anyone free of charge, enabling companies to share the development burden of foundational technologies. Much of the software currently under development around the world falls in this category, to such an extent that most software cannot be developed without some OS content.

While freely available, open source software must still be used with care, for it remains subject to licenses and thus conditions which, if breached, can result in developers losing the right to use it. Developers may also be subject to an injunction preventing them from distributing their apps at all, and possibly a claim for damages. Typical license conditions include the requirement to retain any copyright notices, license notices and disclaimers – all of which are fairly easy to comply with. More complex are the obligations to provide source code for the code or possibly the entire app and to license all of it on the same open source terms. This latter obligation can be disastrous if not properly managed, as it can drain all value from the developer's app and expose it to legal claims.

There are also potential issues with "license compatibility", where licenses for components of the same project are incompatible. This is a complex topic beyond the scope of this paper, but it highlights the importance for developers of ensuring a process for determining which OS code is being used in their apps, so they can ensure compliance with the OS licenses.

Ideally, any agency developing code will have both a framework demonstrating that they understand open source obligations and a mechanism for fulfilling them and documenting such fulfillment[3].

– Provision of compliance materials for open source software

Developers using open source software must take care to ensure a process not only for disclosing the use of OS code but also for providing, on delivery, a complete list of all components used and, crucially, a set of compliance materials, including all text and other information required for the licenses. The list of software components is called a "bill of materials"; the compliance materials (notice files, license texts, attribution notices, disclaimer and so on) are called "compliance artifacts".

Many app store agreements (see below) have provisions which restrict the extent to which open source components can be used. It is therefore important to ensure that the components used in a developer's app do not contain code licensed under terms that could cause problems under the app store agreement or be incompatible with the license the developer is using under its customer agreement.

Key clauses in an assignment agreement

– Assignment

This key clause specifies the work for which IP is to be assigned. The assignment should be without any restrictions and with full title passing to the developer organization. It should be written in clear, legally binding language. All IP rights in the work should be assigned. The applicable legislation may require that it be signed (electronic signatures are acceptable in some countries but not all).

– Moral rights

Aside from IP rights, moral rights may need to be dealt with depending on local laws.

Moral rights are rights authors retain over their work even if ownership of the work is transferred to someone else. For example, in the case of a piece of text created for an app, the moral rights of the text's author include the right to be identified and recognized as such and possibly the right to object to derogatory treatment of the work.

Moral rights generally do not apply to computer programs but may apply to other elements of an app, such as graphics, text or music. They do not always arise where the work's creator is an employee.

Copyright allows authors (as copyright owners) to prevent commercial exploitation of their work, such as copying or distributing a piece of software without the author's consent. As we have seen, copyright can be assigned and licensed. The Berne

[3] One such framework is the Linux Foundation's OpenChain compliance program.

Convention, in Article 6bis, requires its members to grant authors the following rights: (i) the right to claim authorship of a work (sometimes called the right of paternity or the right of attribution); and (ii) the right to object to any distortion or modification of a work, or other derogatory action in relation to a work, which would be prejudicial to the author's honor or reputation (sometimes called the right of integrity)[4]. These and other similar rights granted in national laws are generally known as the moral rights of authors. The Berne Convention requires these rights to be independent of authors' economic rights.

Moral rights, therefore, are designed to protect the integrity of a work and the author's connection to it. They are personal to the author. They are not property rights and cannot be assigned, even when the work itself is assigned. Moral rights stay with the author.

Moral rights can, however, be waived in many jurisdictions[5]. To waive a right means to give it up and agree not to enforce it (for example, against the party or parties to whom the economic rights have been assigned). An example might be where the author of the instructions page of an app agrees not to require that he be recognized as the author of the page.

Since works give rise to moral rights as well as copyrights, parties to the development of apps, when considering IP, should be ready to discuss moral rights and reflect the results of such discussions in the assignment agreement.

For example, developers should be ready to recognize the original creators of materials such as text, graphics or music within the app – or might alternatively seek a waiver of moral rights where permitted. If granted, the waiver would allow the developer to freely commercialize the software without further reference to the authors. It would also absolve them of any obligation to identify and seek permission from the writer or developer to alter the software.

It is particularly important to check local laws in these circumstances, since the scope of moral rights varies significantly from jurisdiction to jurisdiction. In many countries, as mentioned, moral rights do not attach to software, and in some, the author's moral rights cannot be waived.

– Warranties for the developer (ownership, third party claims of IP infringement)

Developers need to ensure that their agreements with agencies contain appropriate warranties. A warranty is a promise, enforceable under local laws, that an app or the process of developing it has certain characteristics. A key warranty is that the app will not infringe or contravene the rights of any other person who may claim it, or who has rights over the app, or parts of it. In effect this means that agencies cannot sell any part of an app if someone else already owns the IP (unless they have an appropriate license).

[4] WIPO. Understanding copyright and related rights, 2016, available at https://www.wipo.int/edocs/pubdocs/en/wipo_pub_909_2016.pdf
[5] For example, in the UK, U.S, Canada, South Africa. Please see https://en.wikipedia.org/wiki/Moral_rights#cite_note-kwall-5

Why should a developer seek warranties?

1. A warranty allows a commissioning developer to obtain greater disclosure from an agency about the IP rights under discussion. For example, it can force an agency to reveal whether other customers are already using and have already been granted licenses to use any of the IP. It can also ensure the agency's disclosure of any open source software used. See sections below on use of third-party and open source software.

2. A warranty also entitles a commissioning developer to seek financial compensation and claim damages if breached. Local advice should be sought as to remedies available for breach of a warranty.

– Liability and indemnities

The biggest issue that can arise from outsourcing development of an app is the possibility of third parties claiming that the app is infringing their rights. In the case of open source software, claims may arise if it is used or distributed in breach of license terms. Such claims can be very costly and time-consuming – devastating, from a business perspective, if a court issues an injunction preventing sale of the app until the issue is resolved.

The obligation to defend any claims that do arise, and indemnities that protect the commissioning developer, should be obtained from the agency as the party assigning or licensing the rights. If indemnities are not available under local law, other available remedies, such as relevant liability clauses, should be carefully drafted to protect the developer. It needs to be ensured that the agency developing code has the right to transfer or license it to the commissioning developer, and if there are claims of the app infringing someone's rights, that the agency will cover the costs and handle the claims.

– Warranties, indemnities and third-party code (including open source)

Agencies will usually be reluctant or even refuse to provide warranties and indemnities with regard to open source components, which they have not written. But at the very least developers should seek a warranty that agencies have carefully selected the code concerned to ensure that it comes from a reputable source. The warranty should also guarantee that use of the code is compliant with the terms of the license under which it is supplied. Code selection should follow a set procedure; OpenChain Specification, for example, provides a good framework, as discussed in Chapter 4.

– Further assurances

Contractual assurances must also be obtained that the agency or contractor will assist in "perfecting" the assignment to the developer. In this context perfecting means completing all the legal formalities necessary to ensure that assignment is valid. If allowed by local law it is better to obtain a power-of-attorney clause or document enabling the developer, as party benefiting from the assignment, to execute any

necessary documentation itself, enabling it to enjoy the benefit of assignment without needing the assignor's assistance, in case the assignor refuses to assist or is not easily traceable. There may be further formalities if the assignment document entails power of attorney, such as requiring the document to be made as a deed.

Where the code is licensed, it may be necessary in some jurisdictions for the copyright owner to grant the developer a separate contractual right to sue infringers of copyright in the code (e.g., for pirating the software). This is another reason assignment of the code is preferable for the developer: it is much easier for the owner of a copyright than for a licensee to sue infringers.

– Other local considerations

Local laws may require confirmatory assignments specific to the country, such as translation of the assignment agreement into local language and other formalities and documentation. Care must be taken to comply with these requirements.

– Boilerplate clauses

The standard legal clauses typically found in most contracts (notices, dispute resolution procedure, governing law and jurisdiction, etc.) should also be included in accordance with local laws, a point usually addressed by the lawyers who draft the assignment agreement.

Checklist

- Establish whether the developers are employees, contractors or agencies.
- If they are employees, ensure that your employment contracts contain adequate clauses. If they are contractors or agencies, address the issue in an assignment agreement or within the software development agreement.
- Ensure that the assignment covers rights to all past, present and future IP included in the work.
- Negotiate and complete any separate payment for the assignment as early as possible. Some jurisdictions do not specifically require this to be dealt with separately.
- Stipulate in the development agreement what moral rights belong to the developer.
- Obtain warranties about the use of third-party code and open source software in compliance with their license terms, adopting procedures to verify what is going into the app.
- Make sure the agency has provided adequate warranties and indemnities in case of third-party claims
- Obtain further assurances from the agency that it will assist in perfecting the assignment, for example by requiring an additional formality or document. Obtain a power-of-attorney clause if possible.

- Seek legal advice on local laws and consider having the agreement drafted or reviewed by a lawyer.

Figure 5 – IP in App flowchart

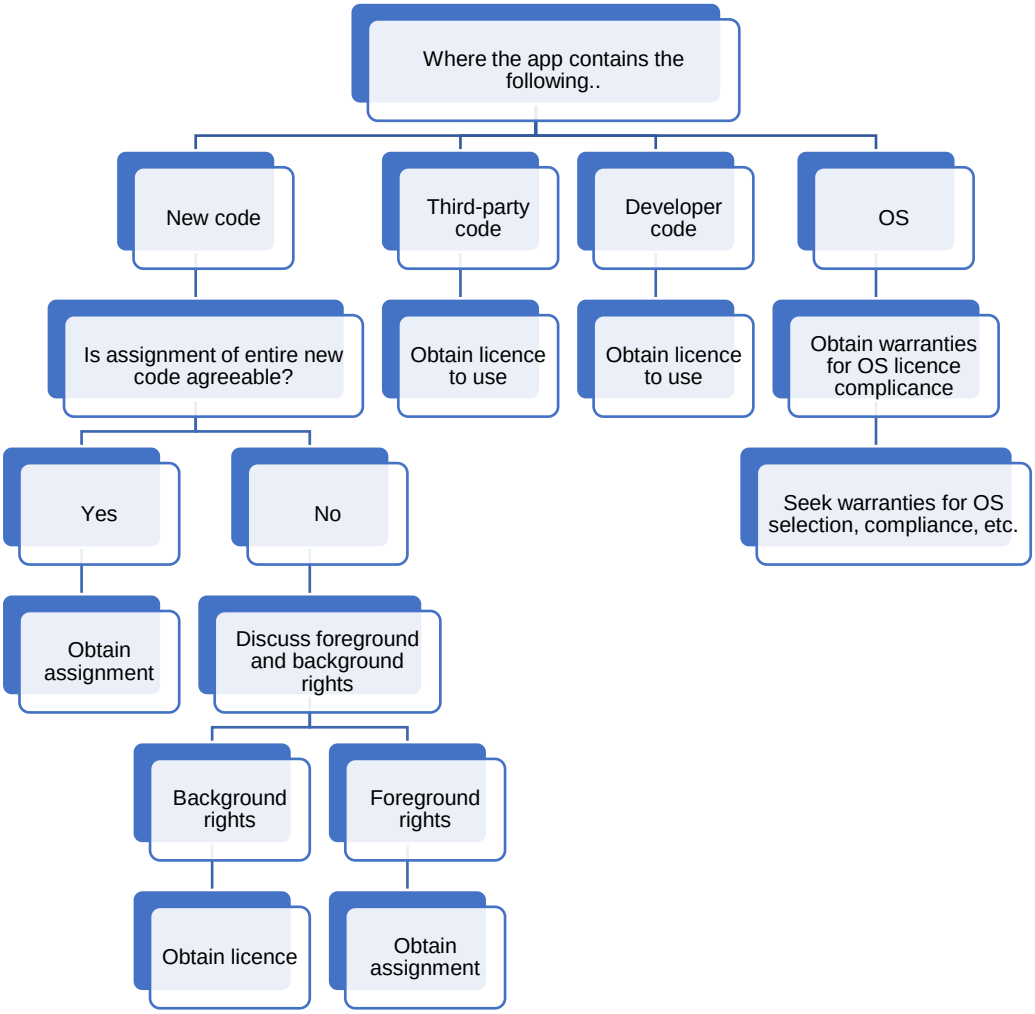
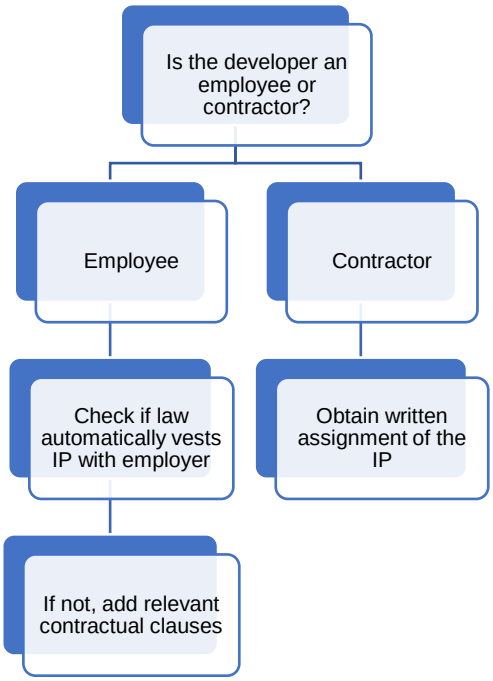


Figure 6 – Developer employee or contractor flowchart



CHAPTER 4 - SOFTWARE/MOBILE APP DEVELOPMENT AGREEMENT

Introduction

A mobile app development agreement performs several essential functions in the creation of new software.

It is principally used by developers commissioning the creation of mobile apps and associated materials by software development agencies.

More than one agency may work on a specific project – one to do the coding work and another to deal with graphic design, for instance. The development agreement clarifies how such agencies will work together and with the developer.

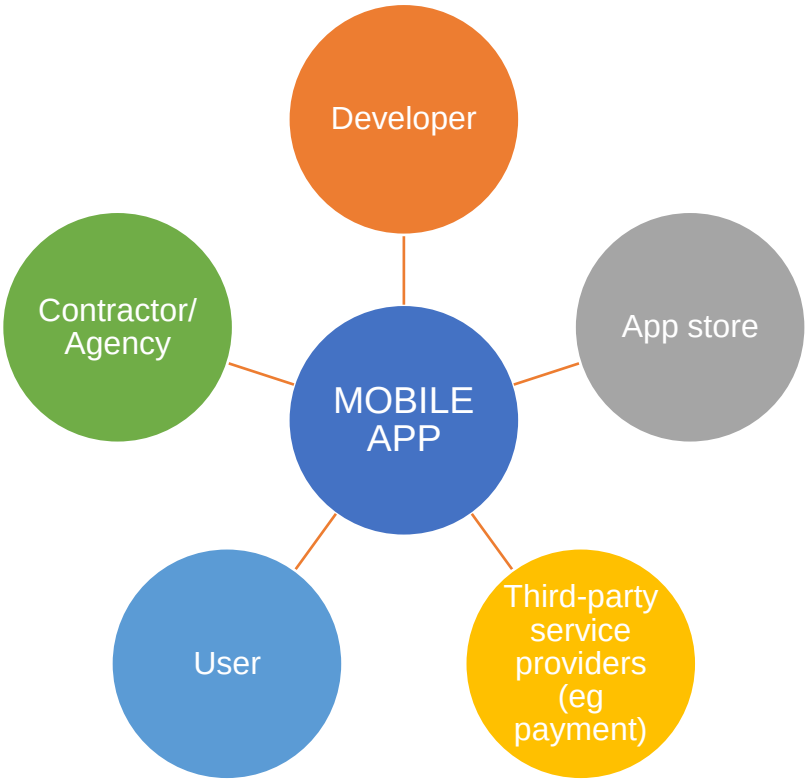
The development agreement also deals with the assignment and/or licensing of IP rights to the commissioning developer and sometimes provides for maintenance of the mobile app. It may also provide for distribution of the app through one or more app stores.

Typical stakeholders and elements involved in the software development process include:

- The developer: the party creating the app or commissioning its creation by another business
- The software development agency, which may have employees or suppliers under contract.
- The app, which increasingly performs a frontend function for users to connect to the developer's backend servers, which provide the app's functionality.
- The app store: the operator of a distribution platform through which the app can be downloaded and the end user or customer can be connected to the developer's service. Some of the best-known app stores include Apple (through iOS), Google (Android) and Microsoft (Microsoft Store), but platform providers, including smartphone manufacturers and some mobile network operators, also run app stores.
- Users/customers of the app.
- The payment service provider for app purchases by users/customers, including:
    - Purchases of the app itself for delivery through an app store, with payment taken by the app store itself or by a third party, such as Stripe or Pay Pal.
    - In-app purchases of add-ons to the app to increase its functionality. Many app stores require payments for in-app purchases to be made through the app store itself (which usually takes a sizable commission).
    - User service fees: developers can use their own payment function or, as more commonly, that of the app store or a third party. Note that mobile app service can be charged to users/customers or made available free of charge. Additional functionality may be provided by having the user purchase tokens or vouchers through the developer (or another web

site), but app stores may not permit it, as it will reduce their own revenues. Care must therefore be taken in designing such mechanisms to avoid compliance problems with app store rules or terms.

Figure 7 - Actors in the mobile app life cycle



Considerations

– Selecting an agency

When selecting an agency, it is important to carry out basic due diligence. Questions should be asked about previous client work, and client references should be obtained. It is also wise to talk to former customers. Local public registries should be searched for details about the company, its officers, financial standing and accreditation (e.g. OpenChain compliance).

Other questions for prospective agencies might include:

- Are they used to working on the developer's preferred platform?
- Are they willing to work alongside other agencies?
- Are they familiar with the interfaces and specifications of any third-party services the app will use, such as payment platforms?

It is quite common for several agencies to work on the same project, which makes it important to divide labor so as to avoid confusion and disputes and ensure proper project management. It is always worth asking each agency whether it has worked with others in the past, and if so, to characterize the working relationships.

– Complete bill of materials

It is rare for a developer to write a piece of software from scratch. In creating an app, as we have seen, developers often use some of their own code, some third-party code or some open source software.

The developer organization should therefore require the agency to provide a complete software bill of materials. In the context of app development, a *software bill of materials* is a comprehensive list of an app's components. Details must also be supplied for each component on all applicable licenses, together with a text file containing all the compliance artifacts that are distributed with the software as required under open source and third-party licenses. Compliance artifacts include notice files, copyright notices, the relevant open source license text and disclaimers. A standard format for defining a software bill of materials, called SPDX, can be found at www.spdx.org.

– Agency software

Agencies may be unwilling to assign copyrights to their codes, which would prevent them from reusing them for other clients. There will probably be clauses, as we saw above, separating background materials (the agency's own code, designed for reuse) and foreground materials (materials written specifically for the project concerned or containing the developer's own IP, and assigned to the developer). This information should also be included in the bill of materials.

– Third party and open source code

Care must be taken with third party code to ensure that the agency delivers the licenses needed for the developer to use the app for its own business purposes. This should include the right to license or sub-license the software to end users. All open source licenses, by definition, already permit this, possibly subject to conditions.

The agency should disclose in the bill of materials any third-party or open source software used.

– Avoiding open source compliance issues (OpenChain)

The use of open source software should be pre-approved, and the developer must warrant that it has complied with the license terms. OpenChain is a compliance framework for incorporating open source code into software projects. It defines the set of policies and practices that an organization must have to develop code using open source in a robust and risk-controlled way. For more information see www.openchainproject.org.

If the agency uses the OpenChain framework, the development agreement can be simplified by combining multiple warranties into a single warranty. Such warranties are usually required in relation to the source of the code and the provision of materials. This single warranty certifies that the agency has developed the app in compliance with OpenChain specifications.

Third-party non-open source software is usually licensed and supported under the third party's standard license terms, which need to be carefully reviewed to avoid conflict with those in the agreement between the developer and the supplier of the app.

– Bespoke code

This is code specifically written for the developer of an app. The assignment of bespoke code should be negotiated so that the developer is entitled to full benefits as an owner (see Chapter 1 on assignment agreements).

– Ongoing maintenance and updates

A developer should think about the support services it will need from an agency over time.

These usually include supplying new releases or updates to the software to correct defects or add new functionality. A distinction is often made between new versions and new releases. New versions add functionality to the software and require payment of a separate fee. New releases address faults, make modifications or enhancements, and install revisions or updates. These obligations should be considered over the lifecycle of the app.

In addition to new releases or updates, there may be a change in the functionality of the underlying platform that entails significant re-tooling of the software, such as when Apple releases new versions of iOS with different capabilities. Alternatively, an entirely new platform may be introduced that is supported by the same app store, such as Google hypothetically introducing a device that projects a heads-up display in your car. Would the developer want its app to support such a feature?

Finally, the requirements of the app store may also change.

The contract between the developer and the agency should allow for all these contingencies to be dealt with.

The developer should also ensure that appropriate service levels and a service credits mechanism are put in place to make sure that support services are delivered to the developer's satisfaction.

– Responsibility for complying with app store terms and conditions

It is vital for software under development to comply with the relevant app store terms and conditions. Such compliance needs to be built into the development lifecycle from the very beginning. The commissioning developer needs to clarify in the development agreement that it is the agency's not the developer's responsibility to ensure compliance, even if the developer ultimately maintains its relationship with the app store.

– End of life and transition to a different agency

"End of life" or transition might arise if the agency fails to perform, becomes insolvent or loses the right to upload to the app store. It is important to prepare for such circumstances by building an exit plan into the development agreement.

Ideally the developer will receive a complete set of source code for the app as well as build instructions detailing how all components are combined in the app and uploaded to the app store. The developer should have an account with the app store that enables it to smoothly take over the uploading of apps from the agency or transfer it to another agency.

One way to do so is to give the developer access to the agency's repository. A repository is a software development version control system for all of an app's source code, continuously updated as it is developed[6].

Developers that are not permitted access to the repository at the outset have the option to use a third-party service, with login access, to keep copies of all source code. The developer gains access only if a specified event makes the agency unable to continue providing the service. This is called an *escrow* arrangement.

In the absence of direct access to the source code, escrow arrangements provide a tool for use by the developer to continue maintaining the app, either on its own or through a third party. An older form of escrow arrangement identifies a third party who agrees to hold the source provided by the agency and to release it to the developer on the occurrence of specified events. The escrow agreement usually has three parties: the developer, the agency and the escrow agent. Agreements between an escrow agent and an agency only are also not uncommon. In such cases the developer must ensure that it is a named beneficiary under the agreement, so that it can enforce it. The terms of the escrow should be reviewed carefully to ensure that the events defined for release of the source code are in accordance with the developer's wishes. The developer should also ensure that the agency is placed under contractual obligation to regularly update the source code in escrow.

[6] GitHub and Gitlab are two popular repository services.

Key clauses

- Scope

The scope of the agreement should be clearly set out. Scope in this context might include the terms under which the agency will supply the app together with its:

– Documentation
– Software bill of materials
– Compliance artifacts
– Copies of licenses for any third-party components incorporated in the app

If applicable, training, professional, support and maintenance services – as well as relations with the relevant app stores (see below) – might also be included within the agreement's scope.

The agreement should also specify which party will be responsible for ensuring that the app complies with regulations and possibly local legislation on age-ratings, ensuring that:

– it contains no unlawful (obscene, defamatory or inappropriate) content;
– regulations relating to the supply of services to children are addressed; and
– privacy and data protection issues are covered.

The developer should link the agency's obligations to its own business requirements, which must be defined in the agreement, to ensure that the agency delivers software that meets the developer's technical specifications as well as business needs.

The agency will commonly work against its own set of technical specifications, drafted in close consultation with the developer. The developer will specify its own business requirements.

- Test Cycle

In mobile app development, the process of acceptance testing is key. During testing, the app and all its components must demonstrate that they serve the business priorities identified by the developer in the original development agreement and interoperate seamlessly with third-party services.

The testing and acceptance process must be defined in the contract. It will probably involve making the app available to the developer on a beta basis, either on a simulator or on the actual device. For iOS in particular this is quite an elaborate process, with the developer downloading the app onto a limited number of devices for testing purposes.

- Delivery of the App

It is important to specify in detail what the agency will deliver to the commissioning developer upon completion of the app.

The agreement must describe the app's components, which may include the agency's own software, bespoke software created by the agency for the developer, approved third-party software, or open source software. These will sometimes be listed explicitly (for example, if the app is a game and is intended to be built on a specific game engine) but sometimes be more generic (the app may contain open source code, for example, so long as that code is available under a specific whitelisted license).

The agreement needs to identify what documentation will be delivered with the software to explain the functionality of the app, and also list any other work or services expected from the agency.

The agency can deliver the app either by transferring it to the developer, who is then responsible for uploading it to the relevant app store, or uploading it to the app store directly, on the developer's behalf.

Bespoke software should be delivered both in source and object code. Any other code not delivered in source code should be placed under escrow as set out in the section above on end of life and transition to a different agency. The only exception to this would be if the component forms part of a respected third-party framework which the developer trusts will remain available indefinitely. In any event, build instructions will be required, and if the agency is handling the relationship with the app store, then the developer should have details on the credentials used to access the agency's account with the app store. Arrangements should also be made for dealing with any cryptographic keys used to sign the code, a complex issue outside the scope of this guide.

- Documentation

Documentation should include information on how the app works and is written, including details about the frameworks, technologies and interfaces used. It should also include information on how to build the app from the source code (see above).

- Work and Services

Other work and services might include maintenance and support, training, and any configuration work. The agreement should specify how the agency would interact with the app store(s), and deal with issues such as an app store's rejection of the app. Please see the section below on ongoing maintenance and updates.

- Licenses

The agreement should require the agency to ensure that it has provided all necessary licenses, including the software licenses contained in the bill of materials but also licenses to any third-party content, such as text, music and images. Access licenses may be needed for the app to interface with third party services (such as a mapping service to provide geographical functionality). Finally, some technology, such as H265 video codecs, may be covered by patents in certain jurisdictions, so an additional license agreement with the patent holders may be required. For common technologies such as H265, another option is to buy a license to a bundle of patents covering the technology from a patent pool set up explicitly to license those patents.

Aside from the licenses required from specific rightsholders, it may be necessary to obtain licenses from collecting societies, which are organizations set up to handle rights on behalf of their members, such as musicians. Music publishing is a particularly complex area, with rules that vary from country to country. In the case of apps using or delivering music, developers should seek advice from specialists in music licensing.

The contract will also need to provide details on the materials needed to comply with open source licenses, as described above.

- Software development methodology

The parties have to agree whether the software will be developed using the traditional "waterfall" methodology or a more modern agile delivery method.

Under the waterfall methodology, the requirements for the app must be specified by the developer at the outset, together with relevant milestones, acceptance tests and delivery dates. The advantage of the waterfall process is that the outcomes are defined at the start, making it easier to determine a fixed cost for the software development activities as a whole and to check deliverables for conformity with the specifications. The disadvantage is that the process is less flexible and that defining specifications at the outset is a lengthier and more complex process.

The agile delivery method is a more iterative process, allowing the specifications to change as the app is developed. For it to function successfully the developer and agency need to work well together and share an overall vision of how the app is to look, feel and operate at the end of the process.

Assignment

An assignment clause can be included in the software development agreement for bespoke software. For various reasons, parties may prefer to keep the assignment agreement separate from the commercial software development agreement. For more details on dealing with assignment, please refer to Chapter 3.

- Ongoing maintenance and updates

Any arrangement requiring the agency to support the app after delivery of the software must be specified in detail, together with the required service levels and service credit mechanisms. This may also include first-line support (support to end-users) and the handling of queries from the app store.

- Obligation to comply with app store terms and conditions

As discussed above, the agreement must contain an obligation ensuring that the app is compliant for publication on the app store(s). Crucially, if the app is rejected for any reason, the agreement must cover what happens, who is responsible and who will address the issue and liaise with the app stores. App stores can be very quick to remove access to an app, even after its launch, if they feel it affects their overall integrity. Grounds for removal could include violation of a site's rules or norms, such as inappropriate material, or infringement of third-party copyrights. A plan must be in place to deal with such contingencies, since removal of an app can be devastating for a developer's business.

- Obligation to comply with third-party service terms and conditions

Payment and other third-party services can likewise have rigid terms and conditions. Unusual activity on an account may lead to withdrawal or suspension of service, interrupting the developer's income, so it is important to be aware of and comply with terms of service, and to assign responsibility for this function in the contract. Where a third-party service is critical to an app's functionality, it is also wise to consider providing an alternative third-party service with similar functionality. A back-up service provider can be kept on standby in case the original supplier ceases or suspends service for any reason. This applies particularly to payment services, but it can also apply to geographic information, mapping, data feeds or other services.

- Warranties, indemnities & limitation of liability

Some components of an app may be under the developer's direct control (bespoke software), and some not (third-party IP).

Warranties are essentially promises by the agency that its services will meet certain standards, that the app will have certain characteristics or certain levels of functionality

and so on. A breach of such promises can lead to claims by the developer against the agency.

Warranties for bespoke software need to be negotiated but are a well-recognized element in software development.

The developer should try to obtain several types of warranty, guaranteeing for instance that:

- the bespoke code it has commissioned and paid for has been developed specifically for the developer, with the agency guaranteeing that no third party has rights to the code that conflict with those granted to the developer (warranty of title);
- the software and related materials and services will not infringe any third-party rights; and
- the app will perform according to technical specifications and will meet defined business requirements.

Where an app incorporates proprietary third-party code (as opposed to open source code), including game engines or video or audio codecs, the agency will typically have a contract with the code's provider that contains some warranties itself. At the very least, the developer should require the agency to pass such warranties on to it, on the same terms.

For open source code, the developer should seek warranties that the agency has selected the code carefully in compliance with license terms. Such a warranty should also guarantee that the code has been screened for security issues and has been selected following a reasonable selection process (as it should if the agency is OpenChain conformant).

An indemnity goes further than a warranty. It is a contract term under which the party giving the indemnity essentially takes all the risk of a particular occurrence.

In the case of bespoke code created by the agency, the commissioning developer may insist that the agency takes all the risk of a third party claiming that the code breaches intellectual property rights. This is a reasonable expectation, since the agency is completely in control of the code development process. In these circumstances the indemnity clause would make the agency responsible for dealing with such third-party claims against the developer, settling them as appropriate at no cost to the developer. The agency would have such responsibility even if the underlying claim ultimately proves groundless. So, indemnities can represent significant protection for the developer but a significant obligation for the agency. It is therefore not surprising that indemnities are some of the most fiercely negotiated clauses in any contract.

The developer should at least consider seeking indemnities to cover such third-party claims, while the agency will naturally seek to exclude them. This inherent tension needs to be addressed in the development agreement.

The agreement should also cover the extent to which the agency may sub-contract part of its activities. While agencies commonly deal with elements such as artwork and

graphic design through third-party contracts, that should be irrelevant from the developer's perspective: the agency should be equally responsible for the quality and intellectual property rights connected with such work whether developed internally or not. There should also be a warranty that in subcontracting any activity the agency will take reasonable steps to verify that the subcontractor can perform the work appropriately.

In addition, applicable legislation must be consulted to guard against any attempt to exclude liability in cases where not legally permissible, such as death, personal injury and other areas where liability may be limited only to the extent considered reasonable. Applicable legislation varies significantly from jurisdiction to jurisdiction, and local legal advice should be obtained.

Each party will want to limit its liability exposure to the other. The agency's exposure will generally be greater since it will be undertaking work and providing materials subject to failure, malfunction, intellectual property issues and so on. The negotiation of liability limits is frequently contentious: the developer will want to claim losses without limitation should they arise from a breach by the agency; the agency will want comfort that its overall risk is limited. It is common to have an overall cap on liability: some multiple of the total sum paid to the agency by the developer. The multiple may be as low as one, where the most the developer can recover is the money it has paid over. In the case of an app's total failure, the developer's losses may be greater, so the overall cap is sometimes higher. The cap is often calculated differently depending on the type of loss. Data protection and privacy breaches can be significant sources of liability, with claims potentially coming from end users as well as regulators, and losses potentially overshadowing development costs. Such liability may be unlimited and far exceed amounts paid to the agency, with multipliers in the range of five, for example.

- Termination

If an agreement has to be terminated because of a supplier's breach or insolvency the customer faces a difficult decision. In cases where the project has not been completed it will have to be taken back and either completed in-house or commissioned to someone else for completion.

Carefully drafted provisions on termination and its consequences must therefore be included in the development agreement. There may be particular issues around the ability of another agency to pick up development of the code where the first agency left off. It may also be complicated to hand over the agency's relationships with the app store and third-party providers, such as payment providers.

From a practical perspective, it is important to consider how the work in progress could be handed over to a new agency. This is where the escrow discussed above would be triggered and the source code delivered to the new agency. But in addition to the code itself, the details of how the code's components fit together and how the app is built from them must be considered as well. It is also important to keep the source code and other materials provided to the developer up to date, whether directly or through an escrow arrangement.

Special Considerations

Special care has to be taken to ensure the app does not accidentally facilitate unlawful activity, such as grooming children, money laundering or breaches of export control laws. Specialized advice may be required if the app ventures into highly regulated areas, such as medical, legal or financial services, telecommunications or e-money, such as cryptocurrency. Such regulations tend to vary significantly from jurisdiction to jurisdiction.

Checklist

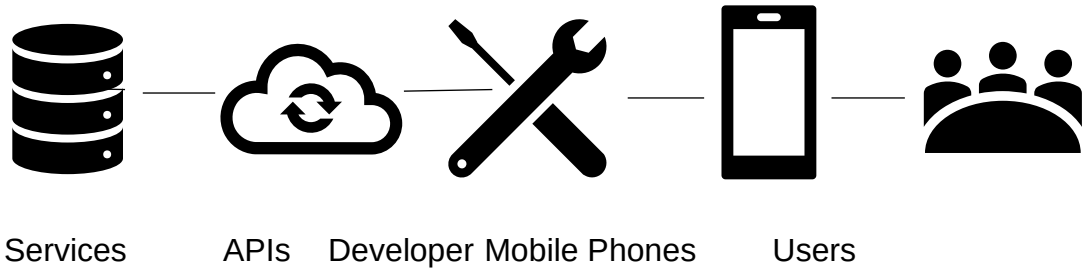
- Conduct basic due diligence on the agency or agencies of interest.
- Once an agency has been selected and serious discussions are about to start, it is recommended to have an NDA in place before confidential information is exchanged.
- The developer should have detailed information on the app's architecture and structure, with the agreement clarifying what types of software will be included – developer, bespoke, third-party or open source – and what third-party services are required to operate it.
- Once this has been established, the developer should ensure that it has the right to approve or reject the components.
- The developer should require the agency to provide a compete software bill of materials as well as details on other required materials (images, sound, etc.).
- The developer should make sure to obtain relevant assignment, licenses, warranties and indemnities in relation to the various components.
- Consideration should be given to the most appropriate software development methodology (waterfall or agile).
- It should be clear who is responsible for other work and services, such as support services.

CHAPTER 5 – THIRD-PARTY SERVICE PROVIDER AGREEMENTS

Introduction

Apps may require integration with other services to enrich their functionality, as opposed to building such services from scratch. Common examples include payment services, third-party data feeds and social media, such as Facebook and Google Maps. Such integration is achieved via application programming interfaces ("APIs"). An API is a defined method of communication between various software components with a set of subroutine definitions, tools and protocols. These give a mobile app borrowing functionality and allow it to use data from other apps or services.

Figure 8 - How mobile APIs work



Use of third-party APIs is beneficial in the development of an app for the following reasons:

- **Cost**. By using the functions of other applications, developers can spend less time building such functions from scratch, reducing the cost of development.
- **Speed.** Less time spent on development means faster marketing.
- **Value**. Advanced features help to differentiate an app from other apps.
- **Convenience**. APIs make an app easily accessible via the channels that users already interact with. APIs with Facebook for social login, with Google Maps for determining geolocation and with Stripe or PayPal for payments can make an app more convenient and thus more widely used.
- **Business development**. A third-party API providing analytics can make for better decisions about an app's functionality in terms of usage patterns, device types and geographical locations. Such information can help developers better understand user behavior and improve the app accordingly.
- **Functionality**. Over time, APIs can provide a simple path for upgrading an app without significant additional development effort by the agency or developer.

Many third-party services are provided on standard terms and conditions, which can make specific terms difficult to negotiate. Developers should carefully review such standard terms and conditions in respect of key issues, as examined below. This will help to keep a developer's agreement with users in line with such terms and conditions and avoid inadvertent over-commitment to things it cannot deliver. Below are a few considerations when choosing a third-party provider.

Considerations

- A potential provider's reputation, experience, financial stability and (where relevant) cybersecurity and data protection credentials should be investigated prior to engagement. Consideration should also be given to the warranties offered by a provider. Developers should carry out their own research and obtain references directly from other users rather than rely on a vendor's website information.
- Some third-party providers can play a key role in determining the kind of mobile app ecosystem a developer can put in place. A third-party payment provider, for instance, will specify the payment options and functionality it can offer to end users. It is good practice to line up a back-up service provider for such critical functions.
- Choosing the right service provider is a crucial decision that can have long-term consequences. To avoid the need to migrate to another provider, disrupting service, developers should make sure at the beginning of the project that their choice is able to scale up as the business grows. Contracts should clearly make the provider responsible for purchasing additional equipment and upgrading software as needed to achieve consistent service levels and cope with growth in the business.
- Developers need to understand the terms on which a third-party provider can terminate the service, and also anticipate how they will respond if the service is withdrawn or becomes unsustainable. As suggested above, it may be sensible to have an alternative on standby for rapid transition if necessary.

Key clauses

- Uptime and availability of Service Level Agreements ("SLAs")

Under standard third-party agreements, providers do not usually enter into service-level commitments, but SLAs may be negotiable at additional cost. This is the time for developers to look closely at service levels and commitments concerning uptime and availability, which will have a direct impact on the quality and level of service offered. The following questions need to be clarified and addressed contractually:

- What is the level of service uptime to be agreed (e.g., 99.9%, 99.99%, 99.999%, etc.)?
- Is the uptime percentage on a daily, weekly, monthly or yearly basis? What are the practical implications of that level?

- Are there exclusions from the uptime percentage for scheduled maintenance or other reasons?
- When and at what times of day will scheduled maintenance take place?
- Is maintenance scheduled on a predictable cycle? If not, how much notice is given?
- How are outages exceeding the agreed level to be compensated?
- Does the provider have the resources to provide full compensation in the event of a major outage?

Apart from specifying response and resolution times, SLAs must be backed by service credits and appropriate contractual remedies – including ultimately the right for the developer to terminate the agreement – if the supplier fails to meet required performance standards.

Service credits provide a mechanism for deducting money from payments to suppliers in the event of such failures. If they are badly structured, they may open the option for the supplier to simply move the developer to a lower tier of service (albeit receiving less money). This makes it imperative to have an ultimate remedy, such as termination, for persistent performance failures. The developer should generally avoid the insertion of financial penalties, which are legally unenforceable in many jurisdictions. Payments or credits so required must in any case be no more than a reasonable calculation of the loss suffered for the provider's failure. Specialized legal advice should be sought to balance these requirements.

- Flow-through issues around privacy and data protection

This section is primarily written from the perspective of the European Union's data protection law: The General Data Protection Regulation (GDPR). GDPR is one of the world's most comprehensive privacy and data protection regimes. Wherever they might be located in the world, organizations engaged in processing the data of EU data subjects are required to comply with it. While an increasing number of countries are following the GDPR model and adopting its terminology – e.g., for such terms as data controller, personal data, data processor and pseudonymization (see the glossary at the end of this handbook) – data protection legislation outside the EU varies significantly, which makes it critical for developers to seek advice in their own jurisdictions.

Developers should ask the following questions:

- Will the provider comply with applicable data protection provisions? Where will the hosting servers be located?
- Given that providers receiving personal data will in most cases be considered data processors, are the provider's "technical and organizational measures" appropriate to comply with relevant data protection laws and regulatory requirements? Particular attention should be given to:

- pseudonymization and data encryption;
- confidentiality, integrity, availability and resilience of processing systems and services;
- restoring availability and access to personal data in a timely manner in the event of an incident; and
- Regular testing, assessment and evaluation of the effectiveness of technical and organizational measures.

- Changes to the service (specifications and roadmap)

Third-party providers should be able to adjust to changes the developer might make to its specifications and roadmap of services. Such providers may include the agency, if it has agreed to provide hosting services, as well as the app store. The reality is that many data processors will be significantly larger and have far greater bargaining power than the developer. This will generally be true of app stores and large hosting organizations, which are likely to operate essentially on a "take-it-or-leave-it" basis.

An app's data protection and privacy features must therefore be tailored so as to comply with the provider's standard terms without breaching data protection and privacy laws. It is therefore always a good idea for developers to make data protection and privacy an integral part of an app's design, as in fact required under GDPR as discussed in Chapter 4 above. Details on privacy by design are covered in Chapter 9 below.

- Warranties & liability

Most providers either exclude or limit their warranties and liability for data loss or corruption and service failure. Historically, on the other hand, most of these service agreements have provided for indemnities that providers have demanded from customers to cover third-party claims and breaches of acceptable use policies or data protection laws. This is driven by the direct obligations data processors must assume under GDPR.

Increased competition among such providers, however, has led to a shift in this customer/supplier relationship. To attract customers, providers are beginning to offer SLAs, warranties and indemnities. For services that are reasonably generic, it is therefore worth examining not only who provides the best service at the lowest cost but also whose contractual terms are most favorable.

Checklist

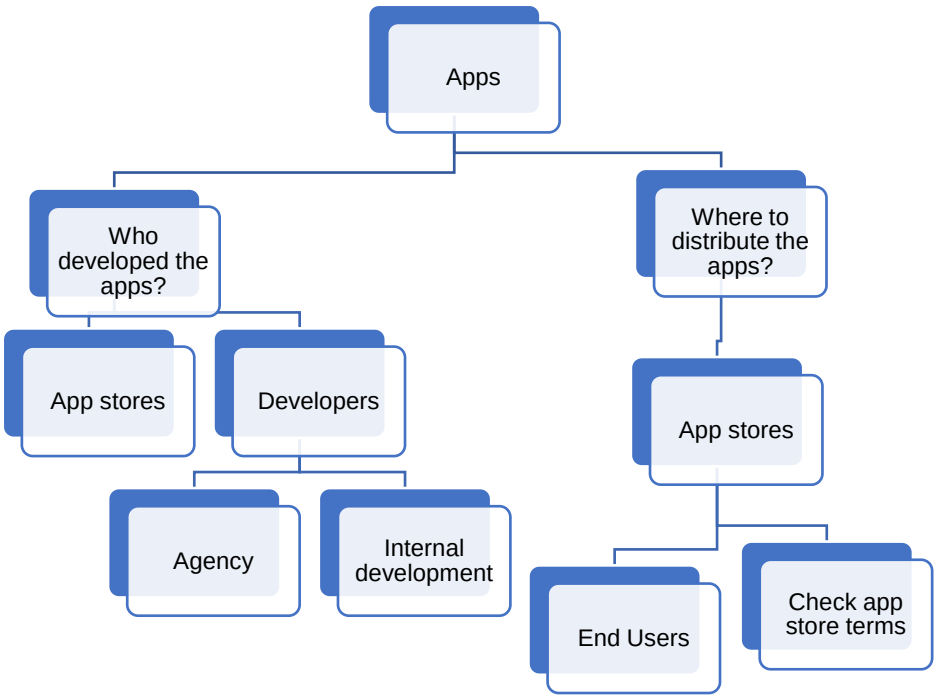
- Conduct basic due diligence on third-party service providers.
- Have a backup service provider if the service is critical (e.g., payment).
- Check that the third party is capable of scaling up as necessary.

CHAPTER 6 - DISTRIBUTION AGREEMENTS WITH AN APP STORE

Introduction

Mobile apps are distributed to end users mostly through app stores. App stores own some of the apps they distribute but also distribute them on behalf of third parties, which may in turn be distributors of other third-party apps – or app developers themselves, having either developed their apps internally or had them developed on their behalf. As we have seen before, the developer may own all the intellectual property rights in an app or have a license to exploit them.

Figure 9 – Typical distribution model



The app store will be responsible for delivering the app to the end user and will generally also take payment for it, and for any upgrades or in-app purchases. App stores generally choose to characterize themselves as agents, making sales on behalf of the app provider in order to avoid being subject to certain consumer law requirements. This means that the developer will essentially be selling its products on the app store's standard terms, while in reality, the sale process will be entirely controlled by the app store.

App stores usually impose their own terms on their customers (the developer in this case) in relation to their use of the app store site through what is typically referred to as a distribution agreement.

The distribution agreement may also include terms on which the app store's own apps are licensed (essentially as a default set of terms) as well as other terms to protect the app store as a software distributor. App stores often give providers a choice between licensing their apps to end users on the app store's default terms or introducing their own terms. An End User License Agreement (EULA) can be used by the developer in place of an app store's default terms, or if there are no default terms.

Most app stores' distribution agreements are publicly available on their websites. Key contractual considerations and clauses when publishing a mobile app through an app store are discussed below.

Considerations

Who has the primary relationship with the app store: the developer or the agency? Life is easier for the developer when the agency does, but serious issues can then arise if the agency contract is terminated. It is also more difficult in that case for the developer to negotiate directly with the app store and deal with particular issues. Assurance should be sought at the very least that if the relationship between agency and app store breaks down, the developer can step in immediately to pick it up, before any damage is done, such as removal of the app from the app store.

Key clauses

Different app stores use different agreements, and there is rarely scope for negotiation. When choosing an app store, key contractual clauses include the following:

– License versus transfer of ownership

It is important to consider what rights to an app a developer would be conveying to the app store: a license to distribute the app or a transfer of ownership? The latter rarely occurs, but developers should be aware of the possibility and pay close attention to clauses of this kind. Developers should in most cases grant a license only. But licenses also require close attention: the ones granted under some distribution agreements are elaborate, granting the app store perpetual, royalty-free rights to the app.

Unless explicitly negotiated otherwise the license should not be exclusive, which would prevent the developer from placing the app with other app stores. Exclusive licenses are not common but do arise with highly popular apps, which app stores and platforms are keen to exploit as promotional tools. Such licenses are likely to be individually negotiated.

– App store development of competing apps

Developers should review these contracts carefully to see if the app store will develop competing apps or allow competing third-party apps, and if so, determine whether this is acceptable. There will generally be no opportunity for developers to restrict such competition.

– Warranties from and to app stores

App stores expect certain warranties from developers in relation to the IP in their apps. This includes specifying whether the organization owns the IP itself or whether any third-party materials have been used. Other warranties may be expected in relation to compliance with applicable laws, and with the developer's obligations as principal to the end user, in which case the app store's legal position is likely to be that of an agent.

App stores may have provisions limiting the deployment of certain open source components. These must be considered carefully.

App stores generally provide very limited warranties. Use of their platforms is on an 'as is' basis. Developers need to be aware that damages, including data loss through use of the platforms, are not usually covered.

App store platforms are also generally free, at their own discretion, to add, amend, or remove their services. Developers must be aware that their apps are subject to removal without notice – and without compensation in most cases. Developers are therefore well advised to maintain good relations with the app stores distributing their products. They should deal with queries or complaints and provide services promptly and effectively. This is particularly true where services are paid for, whether on initial purchase of the app or as later in-app purchases.

With respect to sensitive content, a source of frequent concern, app stores are very mindful about maintaining their reputations in all the territories where they operate – bearing in mind that what appears completely harmless in one jurisdiction may be objectionable in another.

In the same vein, the developer should ascertain an app store's attitudes towards age ratings and ensure that the content of its app is always appropriate for the age rating it receives.

– Advertising: control over creative content, licensing of images, characters etc.

Some app stores demand wide-ranging rights to include visual elements in the app, including characters, videos of game play or the developer's branding images, such as logos and trademarks. This right extends to use of the app on any device, in the app store itself or in its advertising.

A potential concern is that the app store's default terms may enable it to use visual elements from within an app for its own purposes – without the developer's permission. The store could use them to advertise its own services in a manner potentially at odds with the developer's advertising strategy, in terms of brand values and exposure. More significantly, the app store could use such visual elements outside the scope of rights granted to the developer by a third-party owner of IP.

For example, if an app uses a well-known cartoon character – its provider having been granted very limited rights to do so by the studio concerned – and if the app store then

attempts to use the character's likeness to advertise its own services, the app provider will be in breach of its license.

– Termination

The developer should consider how easily the distribution agreement can be terminated and its app removed from the app store. It also needs to identify post-termination consequences in advance.

Issues

- Who has the relationship with the app store: the developer or the agency?

Is the app being licensed directly by the developer, or is the developer acting as a distributor for a third-party app?

Some app stores have one set of terms for individual developers and another for developer organizations; developers should verify that the correct terms are being considered.

- How to respond quickly to app store takedowns?

App stores can and do remove apps for various legal failures. These include infringements of the IP of any third party (or a threat of such infringement), any other breach of third-party rights and failure to comply with applicable laws. Their approach is often "withdraw first, deal with the issue later". This places the developer in a potentially risky position, in terms of lost revenue and reputation.

Most app stores have a procedure for receiving complaints of legal infringements but no process for investigating them or conducting additional vetting.

It is up to developers, therefore, to respond quickly to complaints, seek legal advice as needed, and comply with the app store's requirements.

– Cost of in-app purchases

App stores charge service fees to process in-app purchases by end users. Developers need to clarify the cost and scale of such fees.

These charges are not uniform across all app stores. Close attention must be paid to the specific app store's terms.

Checklist

- Review the terms of the distribution agreement.
- Pay close attention to what is being given away to the app store, (ownership rights, a perpetual license?).

- Determine how easy it is to terminate the agreement and remove the app from the app store.
- Pay attention to in-app purchase fees and other commissions to be charged.

CHAPTER 7 – ADVERTISING AGREEMENTS

Introduction

When developers decide to display ads in their apps (and there are plenty of ways and formats to do so), they need to find and work with an advertiser who will provide an appropriate advertising feed. This process takes place in mobile ad networks that bring developers and advertisers together. There are many such network providers with a range of different offerings, and importantly, many prospective advertisers to place ads with a developer's apps. Interested app developers need first to join one or more such networks and provide details about their apps to the member advertisers, who can then decide whether to place ads.

It is a fast-moving sector, with issues still to be resolved or yet to emerge. Developers are advised to review an ad network's terms before signing up.

Developers should be aware that most ad networks operate on their own standard terms, leaving very little room for negotiation. They should not be approached until their terms – which differ from one network to the next – have been examined in detail. Once developers decide to go ahead with an ad network, their primary commercial interests would be: (a) the network provider's portfolio of advertisers; and (b) the size and reputation of the network provider (the more customers it has, the more consistent and predicable its business strategy is likely to remain).

It is critical that the advertising network offer ads that are appropriate for the developer's app, and particularly the age ranges and jurisdictions targeted. Inappropriate ads will not only affect revenue and potentially generate bad PR, they may also cause an app store to remove the app concerned.

From a technical standpoint, the app's performance must not be impaired if the required SDK (software development kit) is used or if the network goes down. It is common to have a default ad (which may be for the developer's own products) for such eventualities.

Key issues encountered by developers in considering the terms under which ad networks operate are outlined below.

Key Clauses

– Are the terms applicable to developers whether acting on their own behalf or publishing apps on behalf of others?

Terms are mostly applicable in both cases. Developers wishing to distribute the app directly through app stores should focus on the latter.

– What prerequisites has the network provider established?

The network provider requires the developer to incorporate the network provider's SDK. If the app is developed by an agency it should be aware of this requirement.

– What is the scope of the license granted by the network provider?

The network provider grants to the developer or agency a license to use its service/IP. There is no transfer of ownership.

Separate End User License Agreement for the respective software

– What access does the network provider have to the developer's content?

Developers typically grant network providers the right to access, index and cache requests for their content.

– How are potential advertisers selected?

The network provider has the sole right to enter into agreements with advertisers. The developer has no say in who will advertise on its app. The only possible exception is a setting that limits ads to those suitable for certain age ranges. More information about ads suitable for minors is provided below.

The network provider can modify advertiser portfolios without notifying the developer.

– Who is responsible for complying with local laws and policies?

The developer is ultimately responsible for complying with applicable laws and regulations, including privacy regulations.

Content/apps can only be made available to minors in most cases with written notification to the network provider. Some ad networks prohibit advertising to minors altogether.

Developers will usually be held to strict additional data protection requirements, as well as obligations to include certain pre-set clauses in the app terms.

– Who owns data generated by an ad?

Each party owns the data it collects, including data about end users.

– Understanding revenue share and payment terms

The developer must make sure that it understands how the revenue share model works and how revenue is calculated (pay per click/pay per impression/pay per action).

The developer must also understand the payment terms offered (to be made within 30 or 60 days) and the minimum payment thresholds that may apply.

A developer is typically prohibited from making payments to other individuals or organizations out of funds received under this arrangement. This is to prevent a developer acting as an intermediary for other developers/organizations.

– Who bears the costs?

The developer will be solely liable for its incidental costs in respect of advertising.

– How can contracts with ad network providers be terminated?

Ad networks will often insist on the right to terminate service unilaterally – at their own discretion. They can thus terminate the ad agreement with a simple notice and without having to provide any reason. Notice periods are usually 30 days but can be as short as 48 hours in some cases.

Ad networks also often have the right, at their own discretion, to unilaterally modify, suspend or discontinue services or to suspend access to an account.

There are also automatic suspension provisions in some cases if the developer's account remains inactive, e.g., for 12 consecutive months, with any remaining balance settled on termination.

– What warranties are available?

The standard terms of most ad networks provide a service with limited warranties, on an "as is" and "as available" basis, often with no obligation to provide support or updates.

Ad networks generally do not accept liability for the quality, accuracy, reliability, integrity or legality of their services.

– What indemnities are in place?

Under their standardized contracts, ad networks generally have extensive indemnities to their own advantage, with few indemnities that are mutual.

– Limitation of liability

Liability limits operate in favor of the ad networks and are designed to be as wide as legally permissible. They often exclude liability to the maximum extent allowed by law and limit the ad network's liability to a contractual sum. Liability payments may include a fixed sum or a percentage of the net annual amount payable by the network provider.

– Click and impression fraud and other prohibited actions

The developer will be prohibited from engaging in click or impression fraud or other mechanisms that produce similar results. There is also a blanket prohibition against using the ad network in an unauthorized way.

– How will the developer be notified of changes or variations in contract terms?

Any change or variation of terms will simply be posted on the network provider's website. A developer's continued use of the ad network thereafter constitutes acceptance of the updated terms.

– Which country's law governs the contract?

Ad networks will naturally choose the jurisdiction they believe offers them the greatest advantage.

Checklist

- Review the ad network's proposed contract in detail – even if there is no room for negotiation – to be aware of onerous clauses.

CHAPTER 8 - END USER LICENSE AGREEMENT /CONSUMER LAWS

Introduction

As discussed in Chapter 1, an app is protected by copyright in most jurisdictions under local legislation. Where this is the case, users need a license to install and use the software on their mobile devices. Where the developer owns the copyright, the license is granted by the developer directly. Where the rights are licensed to the developer (by an agency, for example) the developer grants a sub-license. In practice, from the user's perspective, there is little difference between these two scenarios, and licenses are often a mixture of the two.

These licenses are generally referred to as end user license agreements (EULA).

The aim of a EULA is to allow developers to protect their investments by setting restrictions on the use of their apps. A EULA acts as both a copyright license (under which users can be sued for breach of copyright law) and a contract (under which both the user and developer can be sued for breach of contract). It is also an opportunity for the developer to exclude its liabilities. The right to exclude liabilities will often be limited under consumer law: many jurisdictions have legislation protecting consumers by restricting the extent to which liability can be contractually limited. They may also have plain language requirements, under which terms have to be drafted in a way that is clear to non-technical readers.

Considerations

– App store requirements

App stores impose their own terms for use of their sites, including the terms under which the app store's own apps are licensed.

Where the app store operates according to a "platform" or "agency" model, developers may have the option of using the app store's default terms. They may also have an opportunity to introduce their own terms, in which case the developer should consider applying its own EULA.

App stores usually require a developer's EULA to address some of the app store's own concerns, passing responsibility for them on to the developer. This is usually specified within their distribution agreements and needs to be carefully noted.

Some of their key concerns are to ensure that:

- The developer's restrictions on use of the app do not conflict with restrictions in the app store's own terms of service.
- App stores will have no responsibility to provide maintenance and support services for the app.

- As far as the law allows, the app store will not be liable for claims relating to the app, including third-party claims for infringement of intellectual property rights. While some consumer laws may prevent them from imposing such terms on end users, app stores usually have agreements with app owners that make the latter liable for such claims.
- The terms of any relevant third party (such as a content provider) will be adhered to by the end user.
- The app store will not be held responsible for an end user's use of any third-party sites to which the app or service may be linked, or for services accessible through such sites.
- The app store will be granted third-party beneficiary rights to enforce licenses (meaning that even if it is not a contracting party to a EULA, the app store can enforce the license concerned as if it were (although to an extent that varies from one country to the next).

– Consumer laws

Developers should seek advice about other consumer laws that may apply to users of their apps. Depending on the jurisdiction, contracts may have to be presented in all of a country's official languages, or include provisions allowing the user to terminate the agreement within a certain cooling off period (although possibly exempting digital goods delivered for use immediately, which include apps).

It is also important to be aware that many countries have legislation requiring services to make provision for users with disabilities. This could entail requiring service providers to adhere to particular standards or make reasonable adjustments for disabled individuals.

Key clauses

- Acceptance

This clause makes the terms of a EULA contractually binding, usually in one of two ways:

- *By providing a link to the EULA on the app's purchase page*, to be reviewed prior to purchase. Developers should verify that the app store provides an option for users to click acceptance of the EULA terms at this point in the transaction. Local laws should also be checked: many require a positive action (clicking a box) to signify user acceptance of such terms, thereby affecting their enforceability.

- *By requiring end users to click acceptance of the EULA before installing an app.* It is generally not considered good practice to make the EULA available only

after payment and download, since refunds for users changing their minds could lead to complications.

A developer's privacy policy, explaining how and for what purpose a user's personal data will be processed, can be handled the same way. Such policies clarify what rights end users have over their personal data and how they can exercise them (see Chapter 9 for discussion of data protection matters).

The same applies for the privacy policies of any third parties concerned, with links provided as above.

App stores may in addition require developers to refer within their apps to the app stores' own rules and policies.

With so many different terms applying to any app store contract, there is a real likelihood of internal conflicts that must be resolved. As discussed in Chapter 6, an app store's distribution agreements may require their terms to prevail over those of the corresponding EULAs, clearly placing the app store's rights above the developer's. To mitigate or avoid such potential conflicts, developers should identify conflicting positions and rewrite their EULAs to reflect a position acceptable to them.

- Restrictions

Developers need to include several types of restriction in any contract with an app store. These include:

1) A clause limiting user rights to download, install and use the software. This commonly refers to prohibitions against users copying, modifying, renting, disassembling or using an app to create competing products. Certain uses cannot be prohibited under some laws, however, so applicable legislation should be checked.

   App stores may require apps to be sharable with other users in a household, under family sharing rules. Such requirements should be detailed in the restrictions clause.

2) Age requirements. Developers can restrict the use of an app to persons meeting age requirements. Many countries restrict the ability of persons under a certain age (typically but not exclusively 18) to enter into contracts (see the "special considerations" section below for further information).

3) Transferability. Restrictions may apply to the transfer of an app to someone else (other than under the sharing rules). Users may be required to remove an app from their mobile devices before selling or giving it to another person, whether or not the transfer involves payment.

4) Acceptable use restrictions, e.g., against:

   a) using an app or service in any unlawful manner;

b) transmitting material that is defamatory, offensive or otherwise objectionable in connection with the use of an app or service;
c) using an app or service in a way that could damage, disable or overburden systems or interfere with other users;
d) infringing intellectual property rights, including those of third parties, in connection with the use of an app or service.

Further restrictions may be appropriate if the app or service allows licensees to submit content.

- Support services

As mentioned in Chapter 6, app stores require developers to take responsibility for quality issues through their support services, with contact details provided to end users. Such services may in some cases be outsourced to an agency (see above). Applicable legislation should also be checked to ensure that other required information, such as a toll-free number or email address, has been properly published with an app.

- Limits to the liability of an app developer

Users might be unable to access or use an app or service for various reasons, including errors, defects, interruptions or delays. Common causes include:

- Reduced service levels attributable to third-party services, including software providers and mobile operators.
- Installation of the app on damaged, corrupted or non-functioning devices, hardware or software.
- Loss or inaccessibility of data stored on the device.
- Services not working as the user expects, not meeting the user's requirements or containing errors or defects that the app owner fails to correct.
- Removal of certain information or content, or refusal to process it, by the app store.
- Reliance by the user on financial information or location data displayed through the app, or on a service or linked third-party site or service.

Some of these may be under the developer's control but others not. In their role as app providers, developers may not always be able to cap their financial liability to consumers, and local laws may prevent or limit their ability to do so.

Minimizing the developer's liability for any such loss or damage can be equally challenging, particularly in the face of consumer protection and other legislation. While it may not be possible to disclaim all liability, exposure can be reduced by educating the user about the uses and limitations of an app. An app that tailors meal recipes to one's preferences and lifestyle, for instance, could feature a disclaimer: "this app is designed for entertainment only. It is not intended to give medical or health advice, and you should always consult a qualified medical expert or nutritionist". Such

language is more likely to be effective than "We accept no liability for loss or injury caused by this app". The rules in this regard vary significantly, however, from jurisdiction to jurisdiction.

- Other considerations

Developers should check if any further information is required by consumer laws or other applicable rules.

- Alternative dispute resolution

Alternative dispute resolution (ADR) is a process where an independent body considers the facts of a dispute and seeks to resolve it, without the parties having to go to court.

Increasingly in many countries, consumer laws require app providers to resort to a recognized ADR entity. The provider must reference the entity's name and website address in their terms and conditions. Users not satisfied with the outcome of ADR can still go to court. Applicable legislation needs to be checked to see if such requirements apply.

Even where not legally mandated, ADR can be a useful way to resolve disputes, particularly with users. The WIPO ADR service provides one such option (see Conclusion for more details).

Special considerations

Local laws may impose restrictions or obligations in the case of contracts with children. Obligations arise in the areas of privacy, contracting, advertising etc.

Checklist

- Since software is protected by copyright, users need to enter into a EULA (end-user license agreement) before they can use the software.
- If third party software or content is used in the app, check if any terms need to be added to the EULA.
- If open source is used in the app, then relevant open source license terms need to be included in the EULA.
- Check key concerns that the app store will want to include in the EULA.
- If the app is complicated, the developer may wish to have its own EULA.
- Consider how loss or damage could be incurred and how mitigated, especially if local laws do not cap the developer's liability.
- Specific requirements may apply where children are concerned.

CHAPTER 9 - PRIVACY AND DATA PROTECTION

Introduction

Although the use of mobile devices and apps is pervasive in day-to-day life, their use in processing data is not always transparent or manageable for users. This is mainly due to the complexity of the mobile app ecosystem, composed of many different elements: the mobile devices themselves (see below), their operating systems and the apps provided for them by app stores. It also includes third-party providers of such diverse services as mapping, advertising and network service monitoring.

The features and characteristics of mobile devices that make them vulnerable to data and privacy breaches include:

- They are easy to carry around, which increases risks to privacy and security.
- They are often kept in 'always on' mode.
- They can collect and store several types of data: emails, messages, photographs, voice memos, health data, contact details, etc.
- They incorporate many different types of sensor – microphone, camera GPS receiver, etc. – that an app can potentially access.
- They use identifiers that can assist in tracking users.
- They likely contain many other apps, which could access data stored by the developer's app (or make use of vulnerabilities in it to read the data).
- They use and interact with third-party systems and services.
- They tend to have small screens, which makes communication with app users difficult.

In many cases, app developers and agencies are not aware of – or lack adequate training in – the requirements of data protection legislation. They are therefore unable to properly prioritize data protection and privacy in designing their apps.

The purpose of data protection legislation is to protect individual privacy and ensure people retain control over the use and accuracy of their data. Failure to carefully attend to such legislation when developing an app can have far reaching consequences. For example, the European General Data Protection Regulation 679/2016 (GDPR) establishes the "privacy by design" principle, under which privacy is considered and addressed from the earliest stages of product development. In the area of mobile apps, other legislation in addition to GDPR may also apply with regard to the confidentiality of communications and related metadata.

GDPR is one of the world's most advanced data protection regimes. It has been adopted throughout the European Economic Area (the European Union plus Norway, Liechtenstein and Iceland), as well as such countries/states as Brazil, Australia, Japan, South Korea, Thailand, California and many more[7]. In other countries – and

[7] https://insights.comforte.com/6-countries-with-gdpr-like-data-privacy-laws

even within the EEA, where each Member State is allowed to adopt its own domestic legislation – data protection laws vary.

It is reported that more than 75 % of all countries now have data protection legislation either in place (in most cases) or under discussion. The need seems to have arisen mostly from the rise in online activity[8]. GDPR has global reach in that it protects the data of the citizens subject to it not only within their own countries but around the world. So, it applies even to developers and app stores based outside the EEA if a single end user qualifies as an EU data subject. Developers may in that case have to register as data controllers with the authorities of an EU country.

Overall, data protection laws remain far from being harmonized worldwide, although several organizations have attempted to devise tools in this area for use by their member countries,[9] and these can be consulted for preliminary information when considering the laws of such countries. It is strongly recommended that developers seek local advice when dealing with any specific country. Here are some key questions when considering data protection legislation:

- What constitutes personal data?
- What obligations are established for controllers, processors and similar establishments?
- Have relevant principles been established in current legislation (e.g., transparency to data subjects, privacy by design, data minimization, etc.)?
- What security-related obligations may necessitate technical and organizational measures?
- What data subject rights should be recognized?
- Are there specific notification procedures for data breaches?
- Are there fines?

The personal data to be protected by legislation in this area is usually defined very broadly, as any information relating to an identified or identifiable living person, including name, address, email address, financial information and medical information. Less obvious examples of personal data include the identifier of a user's mobile device or any information which when used with other information can help identify a user (for example, "the red-haired woman who lives at 43 High Street").

Non-compliance with privacy and data protection legislation can carry substantial fines, but perhaps more importantly, the data breaches that can result from non-compliance could compromise the app's and its developer's credibility and brand value in the eyes of end users.

As we have seen, apps are not usually developed by their owners (those who deploy the app) but by third-party agencies, which makes it even more crucial to clarify the

[8] https://unctad.org/en/Pages/DTL/STI_and_ICTs/ICT4D-Legislation/eCom-Data-Protection-Laws.aspx

[9] For example, see https://unctad.org/en/Pages/DTL/STI_and_ICTs/ICT4D-Legislation/eCom-Data-Protection-Laws.aspx;

role such agencies play in complying with data protection, privacy and security requirements.

Key questions around personal data to bear in mind when developing and deploying an app are outlined below.

Considerations

The first question is whether the app collects and uses personal data. The answer depends on how personal data is defined, which needs to be determined and considered while the app is still under development. It must be assumed that any data – even technical information received back from a user, or a user's IP address – will very likely be used in a way that triggers data protection legislation.

Once it has been established that data to be collected is personal, per the applicable definition, the developer should analyze and decide whether its collection is actually necessary. Under GDPR, for example, personal data cannot be legally collected unless it is genuinely necessary, not merely useful for some unspecified purpose at some future time. This is commonly referred to as the "data minimization principle".

Once the nature and extent of personal data has been ascertained, the app's design should ensure that privacy concerns are properly addressed. As with all aspects of software, privacy is much easier to consider while an app is under development than later. This approach is often referred to as 'privacy by design'.

Where creation of an app is outsourced to an agency, the developer must clearly instruct it to comply with data protection legislation, building compliance into the app from the earliest design stage.

App stores impose security and privacy guidelines, which also must be carefully followed from the outset of app development.

Transparency is another key concept in GDPR and other data protection legislation, under which end users have the right to know in detail how their data will be exploited. Given the complex nature of the mobile ecosystem, however, it is very difficult to obtain a complete overview of data flow, so transparency is challenging. Third parties are also involved, and can in turn involve other third parties. Technical privacy measures must therefore be supplemented with contractual protections.

- Highlighting good practice

Even if not required by legislation the following approach can be adopted as good practice:

- Starting at the design stage, consider what data an app might collect and how its collection might affect the app's users. Then decide whether to collect it, or not.
- Collect and process the minimum data necessary for the app to function. Avoid collecting data "just in case" it might be needed in future. It is not permitted to store

or process data unnecessarily or if no longer needed, and this will automatically reduce the risk of accidentally losing or mishandling it. It is also consistent with the data minimization principle.
- Define how long data should be retained, and retain it no longer than necessary.
- Conduct a data protection impact assessment before starting a project.
- Consider the technical and organizational measures devised by the agency to protect the data collected: for example, designing good security practices (passwords, encryptions) into the app and adopting them for the servers that communicate with the app.
- Determine who has overall responsibility for complying with data protection legislation

GDPR and other data protection legislation identify two key roles in the protection of data: those of the data controller and the data processor.

Data controllers make decisions about the data to be collected and how it is processed. They have the greatest obligations, including responsibility for ensuring that all personal data is processed in a legal, fair and transparent manner.

Data processors process data based on instructions from data controllers, and following protocols for compliance with applicable legislation. They are in a sense servants of the data controllers, carrying out their instructions.

Example

If a developer is distributing an app whose code simply runs on mobile devices, without collecting data from or transferring it to others, and whose output to users contains nothing from the developer's servers or from the internet, then the developer is unlikely to be a data controller with regard to that app.

If, however, the app sends a user's personal data elsewhere for processing, then the data controller has to be clear and transparent about where it is sent and who will be in control of it.

- Determining who is the controller and who is the processor

The developer, as app owner, will typically be identified as the primary controller for the processing of personal data.

But not always. If the developer outsources creation of the app to a third-party agency then there are two possibilities. If the agency has a merely technical role, the developer will likely be considered the sole data controller. But if the agency is more intimately involved in providing the service, and particularly deciding what data to process, and how, either on its own or jointly with the developer, then it becomes a data controller.

In any event, an agreement should be entered into between the data controller and data processor that clarifies the role of each, including with respect to decision making about the data to be processed and about security requirements.

In some cases, there may be more than one controller. For example, where an app integrates with software from third-party service providers – for payment, authentication or advertisements – those third parties often become data controllers in their own right.

More than one entity often processes data on behalf of an app owner. When a user makes a purchase through an app, for example, personal data may be collected by an outsourced distribution center for purposes of fulfilling the sale. Meanwhile, the same data may be processed by a payment service provider. The data controller may collect the same data for its own marketing purposes, if legally allowed (it usually requires the user's consent, which can be withdrawn at any time).

In contrast, where the developer has no involvement in processing the data, which is done directly based on the user's instructions – e.g., if the app simply provides an index of files on the phone, which may contain personal data such as pictures taken by the user or notes containing personal memos – then the user rather than the developer may be data controller.

There are also rules that apply where data is collected and processed inside the EEA and then transferred outside. In a nutshell, any EU data subject must have protection in such cases equivalent to that provided by GDPR, wherever the data is processed. This will mean either application of GDPR or application of another country's law deemed by the EU to provide an equivalent level of protection – or protection through some other mechanism, such as a contract.

This is a highly complex area, but the upshot is that where a developer's users are in the EEA and their data is transferred outside, the developer will be required to ensure that mechanisms are in place to provide equal protections.

Where the app is intended for use in other territories, the local legislation also matters. It is crucial to consider it before the development phase, so the app can be designed accordingly.

Key documents

Where it is determined that personal data will be collected, a privacy policy is required to ensure compliance with data protection and privacy laws.

To comply with transparency obligations under these laws certain information must be provided to consumers about how their personal data is being processed; best practice is to cover transparency in a stand-alone privacy policy prominently flagged for the user.

Important points for providing notices or information in a mobile app:

- Use plain language appropriate for the audience.

- Be transparent about the purpose of collecting the data. Explain why the data is required.

- Be aware that operating system permissions are unlikely to be sufficient (although this could change with future mobile operating systems).

- Make relevant privacy information available as soon as practicable, ideally before the user downloads the app. This could be done via an app store or via a link to the developer's privacy policy.

- If privacy information is provided after an app is downloaded and installed, the developer should make sure that this is done before the app processes the relevant personal data.

- If appropriate, use a "layered" approach, summarizing the most important points while making more detail easily available if users want it.

Special considerations

For sensitive data stricter requirements apply under most legislation. "Special categories" of data include data related to racial or ethnic origin, political opinion, religious or philosophical belief, genetic, biometric or health information and a natural person's sex life or sexual orientation. When the use of a health app, for instance, results in the processing of such sensitive data, the controller typically has to ensure that the user's explicit consent has been given for the data to be processed for particular specified purposes.

Many laws, including GDPR and US data protection legislation, impose several stricter safeguards for the processing of children's and minor's data.

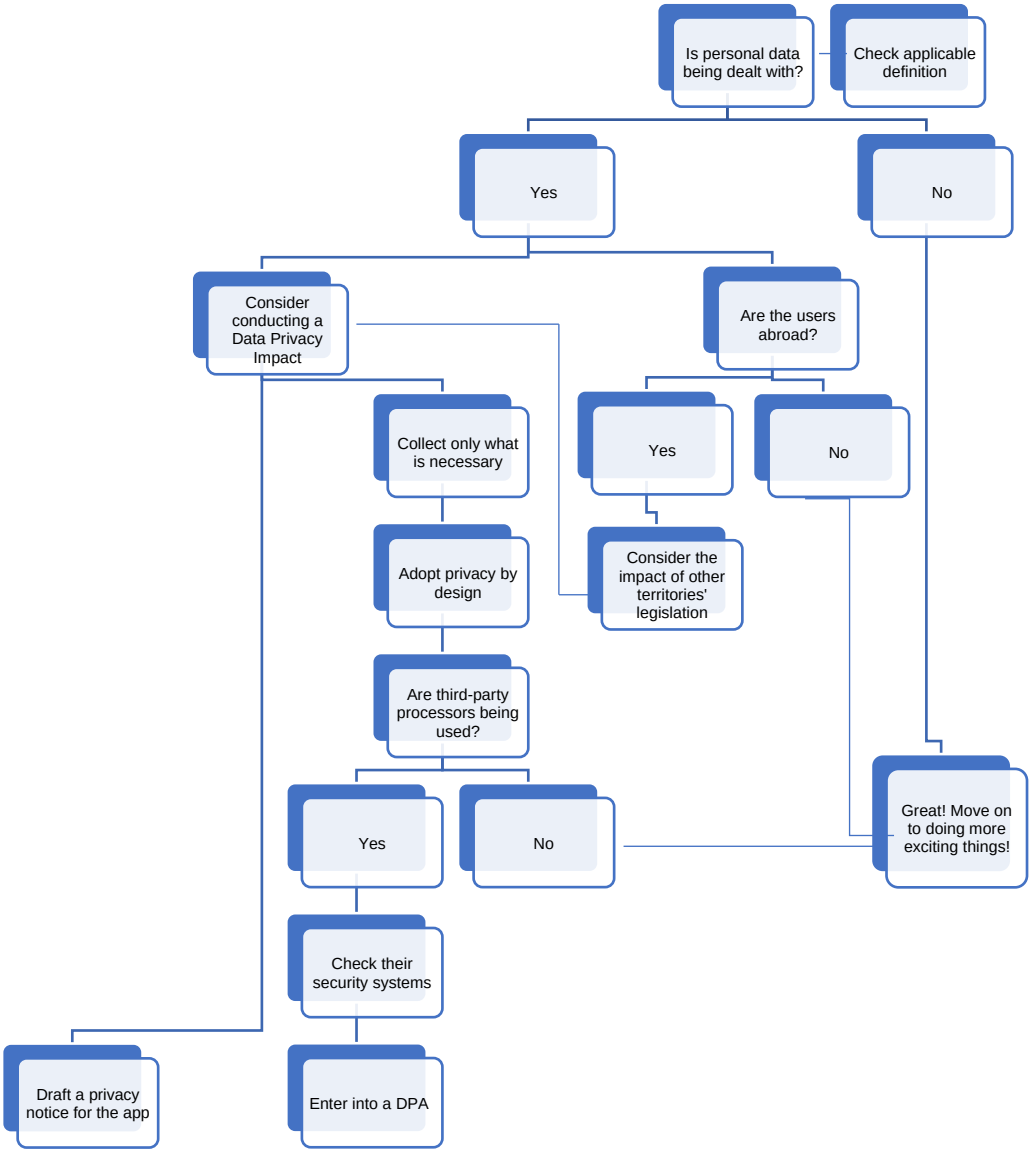
Special consideration must be given to the collection of children's data. The potential harm caused by the inappropriate collection of personal data will be all the greater for children not old enough to fully understand the significance of providing it.

Summary

- Determine if you are collecting or likely to collect personal data. Avoid collecting unnecessary personal data.
- Adopt "privacy by design" during the app development process
- Deploy documents such as privacy notices to comply with law.
- If the scope of the app is not limited to local territory, consider foreign laws as well.
- GDPR must be complied with if there is any possibility that users will include EU data subjects.

- If the app will collect or process children's data or other sensitive personal data, review the relevant requirements and adopt necessary measures.
- Adopt good practices in designing and deploying the app.
- Remember that regulators in some countries can impose heavy fines for failure to comply with data protection laws.

Figure 10 - Data Protection flowchart



CONCLUSION

As we have seen Intellectual Property rights arise throughout the life cycle of a mobile app, from its earliest conception through distribution. IP rights can come into force even after the app is launched, continues to grow and becomes a valuable asset for a business. Such issues include:

- confidentiality at the early stages of discussion about an app;
- IP ownership or licensing;
- data protection throughout the app's life cycle;
- the right choices in selecting providers, including third-party processors and app stores;
- compliance with consumer legislation for the protection of end users.

Ultimately, businesses need to make apps available in a way that minimizes their risks and liabilities to the extent permissible by the relevant laws.

The key contracts and other documents discussed in this handbook, if put in place, can help address these issues, and if done properly, will help give clarity and peace of mind to the parties involved.

Several of these contracts follow the stages of the mobile app life cycle, but some, such as those pertaining to IP, data protection and surrounding issues, pervade the entire life cycle and require consideration at every stage.

As we have seen, third-party providers usually use contracts with their own standard terms and conditions, where there is little room to negotiate. It is hoped that increasing competition in this field will push service providers towards better service-level commitments.

Putting the right contractual documents in place will forestall many issues down the line but is not a guarantee that disputes will never occur.

Since the mobile app ecosystem is such a complex environment with so many actors interacting, differences and disputes are bound to occur with respect to legal, technical, jurisdictional and other issues.

Although mobile applications-related disputes can be resolved through court litigation, in today's fast-paced digital environment, court proceedings for resolving disputes can be slow and expensive. Parties are, with increasing frequency, submitting disputes to mediation, arbitration or other alternative dispute resolution (ADR) procedures. Referral to ADR procedures is consensual. The use of model ADR clauses in mobile applications contracts is encouraged to ensure that the important elements of a dispute resolution clause are provided for and to avoid ambiguity which may later lead to difficulties and delays in the dispute resolution process.

WIPO has put in place an alternative dispute resolution process which can be used for dispute resolution in countries where such mechanisms are either not available or lack the expertise required. For example, the model clauses offered by the WIPO Arbitration and Mediation Center[10] (WIPO Center) have been carefully drafted by experts to allow parties to define the principal elements such as place of arbitration or mediation, applicable law, language of proceedings, appointment of mediator or arbitrator. However, it is recognized that the model clauses may require adapting to suit individual needs. To assist the parties in adapting model clauses to their needs, the WIPO Center makes available an online Clause Generator that allows users to "build" dispute resolution clauses tailored for parties' needs. The freely available Clause Generator allows of the inclusion of a number of core and additional elements based on the WIPO Center's case experience[11]. Developers are encouraged to use this tool within their contractual arrangements to facilitate dispute resolution.

As a concluding note, we would recommend that developers:

- Familiarize themselves with the IP rights that are specific to mobile apps and with the requirements of applicable data protection laws.
- Thoroughly study (at the very least) the key issues that will arise throughout an app's life cycle. Seek legal advice at the earliest stages.
- Protect their IP.
- Conduct basic due diligence when dealing with third parties.
- Following serious consideration, put in place the contracts needed for the arrangements to be clear, meet your expectations and minimize the risk of time-consuming and costly disputes.

Several resources are now available to help put a good contractual framework in place. Those made available by WIPO have been designed to address specific issues. We hope that the practical guidance provided in this handbook will be useful and can easily be adapted to specific national circumstances.

[10] For WIPO model clauses and submission agreements see: http://www.wipo.int/amc/en/clauses. When appropriate, the WIPO Center can also assist parties in adapting the model clauses to the circumstances of their contractual relationship. Parties can contact the WIPO Center at: arbiter.mail@wipo.int

[11] The WIPO Clause Generator is available at: https://wipo.int/amc-apps/clause-generator

ABOUT THE AUTHORS

Andrew Katz

Andrew Katz is the CEO of Moorcrofts LLP and Orcro Limited. Moorcrofts LLP was founded in 2000 as a boutique law firm focused on tech business. Orcro Limited is a sister company to Moorcrofts that provides consulting services, particularly for the issues surrounding software supply chain compliance and the Linux Foundation's OpenChain open source compliance program.

Andrew, who also heads the Moorcrofts Technology Department, has been practicing technology law for over 20 years, having previously been a programmer and accredited NeXT developer. He has focused in particular on cloud computing and free and open source software, including the intellectual property issues arising from the incorporation of open source software into software released through the Apple and Google (Android) app stores. He studied science and law at Cambridge University, qualified as a barrister, and subsequently re-qualified as a solicitor in England and Wales. He is also a solicitor (non-practicing) in Ireland and lectures and works extensively worldwide. His commentary, on issues such as the interface between intellectual property rights and software development, has been published by Oxford University Press, Edinburgh University Press and others. He is a visiting researcher at the University of Skövde, Sweden, where he has co-authored several papers, one used as the basis for the Swedish government's procurement policy.

Usha Guness

Usha Guness is a dual-qualified barrister and solicitor working in the technology department at Moorcrofts. She has over fifteen years' experience in the commercial and technology field, including private practice, as well as international experience working for several companies, including a major global telecommunications company. She holds a Master's degree and recently obtained a certificate in US Copyright Law from the Berkman Center at Harvard University.

ACKNOWLEDGMENTS

We would like to thank Mr. Dimiter Gantchev, of WIPO, for his meaningful insights on the subject of this handbook and its intended audience and for his comments and suggestions throughout the drafting process.

GLOSSARY OF TERMS

Intellectual Property (IP) rights – can include patents, utility models, rights to inventions, copyright and related rights, trademarks, business names, domain names, rights in get-up, goodwill, the right to sue for passing off, rights in designs, database rights, rights to use and protect the confidentiality of confidential information (including know-how and trade secrets) and all other intellectual property rights, whether registered or unregistered, and including all applications and rights to apply for and be granted renewals or extensions of, and rights to claim priority from, such rights and all similar or equivalent rights or forms of protection which subsist or will subsist now or in the future in any part of the world.

Open source software – software provided on terms allowing it to be used, modified and distributed freely. It may be subject to conditions – e.g., requiring that attribution notices, disclaimers, notice files, and copies of license text are retained. There is sometimes the additional requirement that the source code, including that of any other linked software, be made available to any recipient of the code. Technically, open source software is any software licensed in compliance with the Open Source Definition administered by the Open Source Initiative (see opensource.org). The Free Software Foundation (fsf.org) administers a definition of free software that is very similar in practice (if software qualifies as open source it almost certainly also qualifies as free software, and vice versa).

Data Protection Impact Assessment (DPIA) – the process by which an organization determines the impact that proposed data processing activity may have on the rights of data subjects.

Data Processing Agreement (DPA) – the agreement required between a data controller and data processor working on its behalf.

Processing – almost any activity you can perform with personal data, including obtaining, manipulating, correcting, sorting, displaying, printing, storing, transferring and deleting it.

Personal Data – any information relating to an individual and permitting the individual to be identified.

Controller –decision-maker in relation to personal data (what will be collected, how, for what purpose, etc.).

Processor – follower of instructions in relation to personal data, such as those received from controllers concerning whose and which data to collect.

ABBREVIATIONS

IP – Intellectual Property

DPIA - Data Protection Impact Assessment

DPA- Data Processing Agreement

GDPR – the General Data Protection Regulation

BIBLIOGRAPHY

Shemtov, Noam. Reader in Intellectual Property and Technology Law at the Centre for Commercial Law Studies of Queen Mary University of London. Scoping study on availability and use of intellectual property tools to protect mobile applications in three beneficiary countries, namely Kenya, Trinidad and Tobago and the Philippines, available at https://www.wipo.int/export/sites/www/ip-development/en/agenda/pdf/scoping_study_mobile_apps.pdf.

The "Intellectual Property for Business" series, available at https://www.wipo.int/publications/en/series/index.jsp?id=181

Shemtov, N. "Intellectual Property and Mobile Applications" WIPO, available at https://www.wipo.int/export/sites/www/ip-development/en/agenda/pdf/ip_and_mobile_applications_study.pdf

WIPO. Intellectual Property Handbook 2008, available at https://www.wipo.int/edocs/pubdocs/en/intproperty/489/wipo_pub_489.pdf

Privacy and data protection in mobile applications - A study on the app development ecosystem and the technical implementation of GDPR, November 2017, available at https://www.enisa.europa.eu/publications/privacy-and-data-protection-in-mobile-applications/at_download/fullReport.

European Union Agency for Network and Information Security (ENISA). Privacy and data protection in mobile applications - A study on the app development ecosystem and the technical implementation of GDPR, November 2017

Information Commissioner's Office (UK). Privacy in Mobile Apps - Guidance for App Developers, 2013 v 1, available at https://ico.org.uk/media/for-organisations/documents/1596/privacy-in-mobile-apps-dp-guidance.pdf
- Note: this document refers to older legislation that has been superseded. Reference is made from a more general perspective.

Antitrust: Commission opens investigations into Apple's App Store rules - Press Release 16 June 2020, available at https://ec.europa.eu/commission/presscorner/detail/en/ip_20_1073

WIPO. The International Patent System (https://www.wipo.int/pct/en/)

WIPO. The International Trademark System (https://www.wipo.int/madrid/en/)

United Nations Conference on Trade and Development (UNCTAD) Data Protection and Privacy Legislation Worldwide, available at https://unctad.org/en/Pages/DTL/STI_and_ICTs/ICT4D-Legislation/eCom-Data-Protection-Laws.aspx

DLA Piper Data Protection Laws around the world Handbook, available at https://www.dlapiperdataprotection.com/

WIPO: Understanding copyright and related rights, 2016, available at https://www.wipo.int/edocs/pubdocs/en/wipo_pub_909_2016.pdf

WIPO: Guide to Copyright and Related Treaties administered by WIPO and Glossary of Copyright and Related Rights Terms https://www.wipo.int/edocs/pubdocs/en/copyright/891/wipo_pub_891.pdf

Practical law guidance notes

Legal protection of software

Children and the Law

Digital marketing - an overview

App store terms and conditions

Apple terms and conditions for developers: https://developer.apple.com/terms/

Google Play developer agreement: https://play.google.com/about/developer-distribution-agreement.html

Google Play developer policies: https://play.google.com/about/developer-content-policy/

Huawei App Gallery Connect Service Agreement: https://developer.huawei.com/consumer/en/doc/distribution/app/10132

Ad network terms and conditions

AdMob (by Google)
https://www.google.com/adsense/new/localized-terms

Flurry (by Yahoo)
https://developer.yahoo.com/flurry/legal-privacy/terms-service/

MoPub (by Twitter)
https://www.mopub.com/en/legal/tos

Appodeal
https://www.appodeal.com/home/terms-of-service/

StartApp
https://www.startapp.com/policy/publisher-terms/

Smaato
https://www.smaato.com/terms/

Textbooks

Rex o. Nwakodo, Tolley's Commercial Contracts, Transactions and Precedents - 2nd Edition, 2015

David W. Tollen, The Tech Contracts Handbook - 2nd Edition