Protecting your Mobile App
Intellectual Property Solutions

Creative Industries Series

WIPO
WORLD INTELLECTUAL PROPERTY ORGANIZATION

Protecting your Mobile App
Intellectual Property Solutions

Creative Industries Series

WIPO
WORLD INTELLECTUAL PROPERTY ORGANIZATION

Except where otherwise indicated, this work is licensed under the Creative Commons Attribution 4.0 International.

The user is allowed to reproduce, distribute, adapt, translate and publicly perform this publication, including for commercial purposes, without explicit permission, provided that the content is accompanied by an acknowledgement that WIPO is the source and that it is clearly indicated if changes were made to the original content.

Suggested citation: World Intellectual Property Organization (WIPO) (2021). *Protecting your Mobile App: Intellectual Property Solutions*. Geneva: WIPO.

Adaptation/translation/derivatives should not carry any official emblem or logo, unless they have been approved and validated by WIPO. Please contact us via the WIPO website to obtain permission.

For any derivative work, please include the following disclaimer: “WIPO assume no liability or responsibility with regard to the transformation or translation of the original content.”

When content published by WIPO, such as images, graphics, trademarks or logos, is attributed to a third party, the user of such content is solely responsible for clearing the rights with the right holder(s).

To view a copy of this license, please visit: *https://creativecommons.org/licenses/by/4.0/*

The designations employed and the presentation of material throughout this publication do not imply the expression of any opinion whatsoever on the part of WIPO concerning the legal status of any country, territory or area or of its authorities, or concerning the delimitation of its frontiers or boundaries.

This publication is not intended to reflect the views of the Member States or the WIPO Secretariat.

The mention of specific companies or products of manufacturers does not imply that they are endorsed or recommended by WIPO in preference to others of a similar nature that are not mentioned.

© WIPO, 2021

World Intellectual Property Organization
34, chemin des Colombettes, P.O. Box 18
CH-1211 Geneva 20, Switzerland

ISBN (Print) 978-92-805-3308-8
ISBN (Online) 978-92-805-3309-5
ISSN (print): 2789-5432
ISSN (online): 2789-5440

Attribution 4.0 International (CC BY 4.0)

Cover: Getty Images

Table of contents

Foreword 5

About the author 6

Acknowledgments 7

Chapter 1
Industry and Legal Ecosystem 9
Industry overview 9
Relevant intellectual property legal ecosystems 10
Summary of relevant legal systems 22

Chapter 2
Code and Architecture Protectability 29
The making of software 29
Intellectual property rights systems and business impact 29
IP protection and the cloud 42
Summary of IP rights and app code and architecture 43

Chapter 3
Legal and Business Aspects of Protecting Interfaces 51
Graphical user interfaces (GUIs) 51
IP protection and impact on GUIs 51
Summary of IP protection for GUIs 62

Chapter 4
Functionality 67
Introduction 67
Copyright law and functionality 68
Functionality and patent law 71
Functionality and laws against unfair competition 71
Practical implications and considerations of protecting functionality 72
Summary of app functionality and legal protections 73

Chapter 5
Non-IP Legal Considerations 77
Introduction 77
End-user license agreements 77
Data protection 78
Privacy 79
Consumer protection 80
Advertising 81
Digital rights management and technical protection measures 82
App developer agreements 83
Summary of non-IP legal considerations for app developers 89

Chapter 6
Global Challenges 97
Copying and emulation risk mapping 98
Conclusion 103

Foreword

Over the past several years, the digital environment has created enormous interest in making a living in various creative industries. Mobile applications, commonly referred to as apps, have become an indispensable part of daily life in the digital world and the digital economy has grown exponentially driven by a huge community of software developers. Indeed, one out of every eight developers is involved in designing the mobile apps used by millions of people and businesses worldwide.

This new publication by the World Intellectual Property Organization (WIPO), released under the *Making a Living in the Creative Industries* series is designed as a tool for app developers and publishers who have not yet given due consideration to the intellectual property (IP) aspects of their work. It seeks to address the growing demand for legal clarity and offers business-oriented guidelines on copyright and IP more broadly to generate additional income for creators and right holders.

The content provides practical advice and insights to allow informed strategic decisions based on the legal ecosystem and IP law. It presents a thorough review of specific app industry areas that relate to IP law and provides an overview of the relevant business issues in the mobile app market.

The publication provides specific recommendations and maps the complex legal IP system. Mobile apps are multilayered products with different features which may be protected by various IP rights and therefore an understanding of this multiplicity and how best to leverage it to drive business growth will be discussed. Finally, the publication reviews the mobile app value chain and offers a checklist of issues to consider when identifying the relevant IP rights, protection options and strategies.

With this publication, WIPO hopes to offer relevant information to the growing community of professionals operating within the mobile app economy.

About the author

Dr Noam Shemtov is a Reader in Intellectual Property and Technology Law and a renowned academic in his field. He is the Deputy Head of the Centre for Commercial Law Studies at Queen Mary University of London. He lectures in the areas of intellectual property, creative industries and technology. His research interests also focus on these fields.

Noam has led research projects and studies funded by the UK Research Council UK and by industry, supranational and commercial organizations and has published extensively in his areas of interest.

He holds visiting appointments at Spanish and Dutch universities where he lectures regularly in areas pertaining to intellectual property and technology. He is a qualified solicitor, both in the United Kingdom and Israel.

Acknowledgments

The author is grateful for the assistance of Ms Diana Renuka Dukhia for all aspects of this project. Diana has always been diligent, methodical and insightful and her contribution has been invaluable. Thanks are also due to WIPO staff for their constructive comments and suggestions, in particular Mr Dimiter Gantchev, whose input was most insightful, helpful and contributed greatly to the production of this work.

Chapter 1

Industry and Legal Ecosystem

Industry overview

The mobile application market has been growing at a rapid pace over the last several years. Commonly called mobile apps, developers have been employing various monetization models such as direct sales, freemium, subscriptions, in-app ads and in-app purchases (the latter two becoming more and more popular in recent years). According to the mobile data and analytics platform App Annie, in 2019, users downloaded 200 billion apps and spent more than USD 120 billion in app stores worldwide. The market is projected to continue to grow significantly.

With the exponential increase in smartphones' processing power, the latter have turned into mobile gaming devices. In terms of income generation, this is where financial gains may be made. App Annie reports that mobile games, which were responsible for about 50 percent of revenue in 2011, are now responsible for upwards of 85 percent of the industry's revenue.

Geographically speaking, the Asia-Pacific (excluding Japan) region is expected to retain its lead position in terms of Compound Annual Growth Rate, with Latin America and Eastern Europe battling for number two position.

2016's Pokémon Go mobile game frenzy saw augmented reality (AR) make significant inroads into the mobile app ecosystem. It is predicted that with the technological advancement of mobile devices, virtual reality (VR) and AR technologies will become the new frontier in the mobile app market. Another new frontier for the industry may be wearables, with Samsung, Apple and a plethora of Asian companies releasing new wearable gear. It is projected that these areas hold enormous growth potential. Not only are mobile devices becoming more powerful and capable of processing heavier applications, but in 2019 there were an estimated 4 billion

smartphone users worldwide. We can expect a significant increase in the number of AR and VR-based devices which are likely to be more powerful and enable heavier apps.

Relevant intellectual property legal ecosystems

A variety of intellectual property (IP) rights may apply to mobile apps. Furthermore, the majority of IP rights could be employed to protect various facets. The following is a brief overview of these rights and their nature, scope and legal mapping.

Copyright

As its name suggests, copyright primarily concerns the right to copy. Obviously, defining copyright in such a manner over-simplifies things, but the origin of the right as well as its current essential function concerns, among other things, the regulation of the reproduction of protected works. Copyright law was originally intended to protect conventional authorial works such as books, musical compositions, paintings and sculptures. At first glance, it may therefore seem odd that copyright plays a part in protecting a functional and technical item such as a mobile application. A computer program stands at the basis of each and every mobile application. Such a computer program is a functional item that may be protected under copyright law.

It is not its functional character that renders a computer program a somewhat odd subject matter for copyright protection. In fact copyright law has been protecting functional works such as geographical maps and directories for almost 200 years. Rather, it is that computer programs, whether in object or source code format, are not designed nor intended to communicate with humans. Unlike traditional works protected under copyright law, whether functional or artistic, computer programs are ultimately intended to instruct a computer. It is this peculiarity that makes computer programs stand out as a protectable subject matter under copyright law and contributes to some of the problematic aspects that become apparent when trying to determine protection boundaries.

The subject matter of copyright protection is diverse and ranges from items such as computer code to paintings and films. As we shall see, many relevant facets of mobile applications apps are eligible for copyright protection, subject to some important caveats and exceptions. However, it is not sufficient to show that a given subject matter is eligible in principle to copyright protection. An important pre-condition for copyright protection is originality. Hence, it is necessary to show that the relevant work is original in a copyright sense. Although, in general, the originality requirement should not pose a particular problem to a work that results from an author's exercise of choices, it may raise some issues in the context of works that are functional in nature.

Various aspects of mobile applications apps have functional features, and the scope of copyright protection should therefore be examined with care in this context. One of the key concepts that places limits on the scope of protection granted under copyright law is known as the "idea/expression dichotomy." This concept essentially states that copyright does not protect mere ideas, but only the specific expression of those ideas. The rationale for this rule is clear: since every first author is also a second author (i.e., every author utilizes old ideas when generating new works), allowing monopolization of ideas may dramatically reduce the overall number of new works.

While the concept of the idea/expression dichotomy is recognized in various international treaties and, either explicitly or implicitly, in many jurisdictions' copyright systems, the devil is in the detail. Where is the line between taking or copying an aspect of one's ideas and simply being inspired by those ideas?

The following discussion examines such questions in the context of mobile apps and provides guidance in identifying the boundaries between permissible and impermissible. This could benefit parties interested in this question from both sides of the divide. It may enable a party to make an initial assessment as to whether or not a competitor copied aspects of its mobile app which are protected under copyright law. Equally important, it may enable a party who wishes to launch a competing product to make an initial assessment as to whether or not it could replicate some of its competitor's app elements.

Once copyright eligibility is established, it is then necessary to examine whether the contested behavior falls within the scope of the exclusive rights enjoyed by the right holder. Every copyright system provides for several exclusive rights which, as their name suggests, are enjoyed by the author to the exclusion of all others. When, a party carries out an act within the scope of an exclusive right of another without authorization, the party may be liable for copyright infringement.

Among the exclusive rights of relevance to the present context are the right of reproduction (copying), the right of adaptation (the right to make derivative works), and the right of making available to the public (the right of distribution). The right of reproduction means the right to make a copy of all or part of the work in question. The right of adaptation encompasses, among others, the right to make a derivation from the work. The right of making available to the public includes the right to make the work available over the Internet in such a way that members of the public may access it where and as they wish. A person may be liable if they infringe directly or indirectly on the exclusive rights of the copyright owner. While direct infringement is a strict liability tort and does not require a particular state of mind (e.g., intention), indirect infringement usually requires a certain knowledge threshold. It is important to bear in mind that even where it appears that copyright infringement could be established, a relevant defense may be applied and possibly excuse the party.

The final issue of copyright to mention is ownership. Like some of the other issues explained above, establishing ownership will vary from one jurisdiction to another. As a general rule, initial ownership will usually reside in the author of the work. This is subject to one main exception – where the work was created in the course of one's employment. Where that is the case, the default rule in common law countries, such as the United States or the United Kingdom, will usually be that ownership lies with the employer rather than the author/employee.

The ownership rule in civil law countries usually provides that whether or not created in the course of employment, initial ownership lies with the author/employee. In such a case, in order for

the employer to have ownership over a work created in the course of employment, a suitable clause should be included in the employment contract. Such a clause, drafted by a local employment lawyer, may provide that works created in the course of employment by the employee are thereby assigned to the employer.

Copyright's term of protection is considerable: protection is provided at least for the duration of the life of the author plus 50 years.

Patents

Patents are traditionally associated with industrial products and processes rather than with software-based items. However, over the last few decades, the legal landscape of the patent sphere has changed. Software-related inventions are now eligible for patent protection as long as they satisfy the requirements of patent law. However, due to certain public policy considerations, they usually face difficulties in forming patent-eligible subject matter.

Unlike copyright, patent rights do not arise automatically upon creation but are because of registration. The application process may take, on average, a few years; however, an application may be contested at various stages thereby prolonging the process. For example, the European Patent Office (EPO) grant procedure takes three to five years from the date an application is filed. It is made up of two main stages. The first comprises a formalities examination: the preparation of the search report and the preliminary opinion on whether the claimed invention and the application meet the requirements of the European Patent Convention (EPC). The second involves substantive examination.

Another major area where the patent system differs from the copyright system is in the associated costs. Such costs may comprise filing fees, prosecution costs, grant fees and renewal fees. To this, one may add the fees charged by a specialized patent attorney. In some jurisdictions, a patent attorney is to be distinguished from lawyers or attorneys at law, as they are technically qualified, with a degree in science or engineering, and have undergone legal training in patent practice.

Like all IP rights, patents are territorial in nature and thus only valid in the jurisdiction in which they were granted. An exception is European patents that are granted by the EPO, based on the EPC. The EPC provides a centralized system for granting patents in any one of the signatory states, using one language and one procedure. Once granted, such patents are subject to the same conditions and have the same effect as national patents in EPC countries. The EPO and the EPC are not a part of the European Union (EU) framework and include signatory states that are not EU member states. It should be noted that, in principle, the EPO does not grant a unitary patent right, but a bundle of national rights for jurisdictions designated by the applicant. At present, the availability of a unitary European patent right is becoming a reality as a 'European patent with unitary effect' could soon be granted by the EPO in relation to the territory of the 25 member states participating in the unitary patent scheme.

Like copyright, patents provide a set of exclusive rights to its owner for a limited period, generally up to 20 years from the filing date. These rights allow a patent owner to control who can use, make, and sell the protected invention. In return, the patentee discloses to the public how the invention works so that a person skilled in the relevant field may make the patented invention. After the patent expires, others may implement aspects of it in their own products or services. Throughout the duration of the patent, others may learn from what is described in the application and use this information to implement different solutions that do not infringe the patent at issue.

The scope of the patent and the benchmark against which novelty and inventive step will be assessed are the claims. These are drafted by a patent specialist and define the scope of the monopoly sought by the applicant. Narrowly drafted claims are likely to prove less useful in fending off competitors, while broadly drafted claims are more susceptible to challenge on the basis of lack of novelty and inventive step. A good patent specialist will strive to draft claims in the broadest possible manner, while ensuring that they can withstand novelty and inventive step challenges.

In general, a patent is to be granted over a subject matter in any field of industry,[1] which is new, not obvious and capable of industrial application. In addition, the subject matter should not fall under one

of the categories excluded from patentability. The requirement of novelty means that patents should only be granted to something that has not existed before. In determining whether an invention is new, it is necessary to compare it to similar items that existed at the time the patent application was filed and assess whether it is different. Such a collection of similar items is usually referred to as "state of the art." Although patents are territorial and enforceable only in the jurisdiction in which they are granted, the "state of the art" is assessed globally. In other words, it does not matter in which jurisdiction a patent application is filed, the invention will be compared to that which existed at the time of filing anywhere globally. Patent examiners in offices around the world have access to databases that enable them to make such assessments.

The requirement of non-obviousness or inventive step (both terms are used interchangeably) is that the invention must establish more than just being new. It must show that the invention involves a considerable leap in comparison to the "state of the art" so that it is not obvious to a person skilled in the relevant technological field. While the novelty requirements ensure that there is a quantitative difference between the invention and the prior art, the inventive step requirement is designed to ensure that there is also a qualitative difference. It encourages people to carry out research. Put another way, the "distance" between prior art and the invention must be significant and must not be a matter of routine activity within the relevant field but requiring an inventive effort.

While this is straightforward to explain in abstract, differentiating between obvious and non-obvious inventions is often challenging. This is because, although there are legal tests designed to assist patent examiners and courts, it is ultimately a factual inquiry that depends on the specifics of a given case. As previously mentioned, one of the most important factors for determining whether an invention is obvious is the level of knowledge held by the "person skilled in the art." Clearly, the more skills and qualifications the skilled individual possesses, the more likely that a given invention will be found to be obvious, and vice versa. Beside skills and qualifications, the resources and equipment that are usually available to such a person are considered. Determining the technical field in question is of great importance in establishing the level of skills and

qualification held by the skilled person. For example, the EPO has previously determined that the "person skilled in the art" in relation to an invention concerning a computer implementation of a business method was a technical expert in data processing rather than merely a businessperson.

Patent law's equivalent of copyright's exclusive rights, namely infringement activities under patent law, commonly encompasses acts such as making or using a patented product or process, including its selling or importing. Unlike copyright, access to and copying from the patented invention is of little relevance to the question of liability. Independent creation is not a defense to a patent infringement action and liability could be established. In essence, most commercially valuable activities are within the bounds of a patentee's exclusive rights. In some patent systems the rights given to the patent owner differ depending on whether the patent was granted in relation to a product, a process, or a product derived from a specific process. As with copyright, a person may be liable if they infringe directly or indirectly on the exclusive rights of the owner. While direct infringement is a strict liability tort and does not require a particular level of state of mind (e.g., intention), indirect infringement usually requires a certain knowledge threshold.

Importantly, although software-related inventions are not excluded from the patent system, certain general exceptions to patentability pose a particular challenge to such inventions. In principle, international treaties require that patents shall be available for any inventions, whether products or processes, in all fields of technology. Therefore, on the face of it, patent systems should not discriminate between different technological fields and bar software-related inventions from patentability. In the United States, the main patent eligibility obstacle to such inventions is the judicially made patentability exception of abstract ideas. At the EPO, it is the requirement of technical character or technical effect. Nevertheless, a non-insignificant number of patents relating to such inventions is granted every year in most developed jurisdictions around the world. Once granted, they may offer a broad legal monopoly that may place its holder in a strategically powerful position in the marketplace. It is therefore often worthwhile for a developer to explore whether a particular software-related development is potentially eligible for

patent protection. Furthermore, a business with a patent portfolio may prove to be more attractive to potential investors.

Initial ownership is often vested in the inventor, or joint inventors, subject to several exceptions, the main one relating to inventions made by employees. Although it is possible to transfer ownership to a third party via a patent assignment, it is prudent for an employer to ensure that inventions created by employees will be owned by the former. An adequately drafted clause in the employment contract may see to that.

Whenever one is considering patents, there is another form of protection – part of the broader patent-like rights family – to be kept in mind: utility models. When available, utility models offer protection that is somewhat similar to patents for a shorter period, typically between seven and 10 years. This shorter term may be suitable to mobile apps, due to its relatively short shelf life. Like patents, utility models must be new to qualify for protection. However, the novelty requirement is less strict and "absolute novelty" is often not required. The legal position on utility models varies greatly from one country to another. While almost every country has a patent system, the same cannot be said for utility models. Utility models are currently available in the following countries: Albania, Angola, Argentina, Armenia, Aruba, Australia, Austria, Azerbaijan, Belarus, Belize, Bolivia (Plurinational State of), Brazil, Bulgaria, Chile, China (including Hong Kong and Macau), Colombia, Costa Rica, Czech Republic, Denmark, Ecuador, Egypt, Estonia, Ethiopia, Finland, France, Georgia, Germany, Greece, Guatemala, Honduras, Hungary, Indonesia, Ireland, Italy, Japan, Kazakhstan, Kuwait, Kyrgyzstan, Lao People's Democratic Republic, Malaysia, Mexico, Peru, Philippines, Poland, Portugal, Republic of Korea, Republic of Moldova, Russian Federation, Slovakia, Spain, Tajikistan, Trinidad and Tobago, Turkey, Ukraine, Uruguay and Uzbekistan. Notably, there are no utility models in the United Kingdom and the United States. In addition, the following organizations offer utility models: the African Regional Intellectual Property Organization (ARIPO) and Organisation Africaine de la Propriété Intellectuelle (OAPI).

A mobile app publisher should be aware that utility models exist and may prove useful as part of their protective arsenal.

Trade dress under trademarks and unfair competition laws

At first glance, trade dress protection may not come to mind when considering mobile app protection. Trade dress generally concerns protection of a product's appearance, such as the product's outer look or packaging. For mobile apps, the main feature applicable for trade dress protection is the graphical user interface (GUI). A mobile app's computer code, software architecture, algorithms, data structures and various other elements, although important to its operation and success, do not relate to trade dress as they are hidden from view. However, the importance of an app's trade dress should not be underestimated. As any user of mobile apps would confirm, the ability to engage and interact with ease through an app's GUI is of considerable importance. It is often one of the main factors considered by users. The ability to protect GUIs against imitation or cloning is key to a developer's success in the marketplace.

The aforementioned refers to trade dress as the subject matter of protection for mobile apps' GUIs, but it does not discuss the legal vehicles for protecting such trade dress. As will be discussed, trade dress can also be protected by design rights (in the EU) and design patents (in the United States). However, this part is concerned with protecting trade dress via the laws of registered trademarks or unregistered signs (the latter through unfair competition laws).

Trademark laws are primarily concerned with signs applied to products or services that serve as indicators of origins. Hence, where a distinct sign applied to one's goods or services is associated in the mind of consumers with a particular source of origin, trademark registration may be successful.

It is these aspects that an applicant will seek to register as a trademark and, once registration is successfully completed, should deter third parties from incorporating them into their work. To be registered, among other things, the subject matter of a trademark application must be clearly defined, must not be descriptive and should possess a certain degree of distinctiveness.

An important exception for registration is functionality. This means that the subject matter must not be functional in nature, with the definition of functionality varying by jurisdiction. The rationale is clear: signs applied to products or services serving a functional objective should be protected under patent law, not under trademark law. Requirements for protection under trademark law are less stringent to those under patent law. However, the duration of trademark protection is perpetual, subject to renewal. Hence, if functional and technical features could be protected under trademark law, the latter could be used to gain protection for subject matter that falls under patent law but nevertheless does not meet it eligibility criteria. This will be likely to frustrate the public policy consideration that stands at the heart of the patent law system.

In many jurisdictions, worldwide trade dress, or a product's visual appearance, could also be protected through unregistered sign systems, such as unfair competition. It is important to note that there is no globally uniform position on the protection of unregistered signs or marks and in some, jurisdictions, such as in China, protection is not afforded.[2] In common law legal systems it is first necessary to show that the contested sign is distinctive and serves as an indication of origin. Once this is established it will be necessary to show that activities resulted in a degree of confusion or deception in the mind of the relevant public.

There are a variety of defenses that may be applicable where a mobile app developer uses a feature of an earlier mobile app that is, at first appearance, protected as a registered or unregistered trademark. This will be discussed in more detail in Chapter 3.

Designs

As previously mentioned, product appearance may be protected as a trademark or unregistered trade dress, but also may also be protected, where appropriate, under an industrial design right.

Intellectual property systems that protect product designs are not uniform. For example, while the EU provides for a two-tier system of unregistered and registered deigns, both the United States

and China have a design patent system where, subject to certain variations, the applicant must establish that the design in question satisfies similar requirements to those under utility patents, that is novelty and inventive step. In comparison, protection criteria for a registered community design (RCD) are novelty and individual character. The latter refers to whether the overall impression of the produced design differs from overall impression of earlier designs made available to the public. Regarding registered designs, the EU uses a two-tier system, whereby one may apply for either a registered community design or a registered member state design. The substantive requirements in both cases are identical.

Both RCD and design patent systems provide protection against infringement by a third party, while, like copyright law, the unregistered design system only provides protection against copying.

As is the case with trademark protection, both for registered designs or design patent systems, a fundamental limitation on eligibility is functionality. More detail will be provided under Chapter 3 as the scope of functionality exceptions varies by jurisdiction.

In terms of protection period, unlike patent (20 years according to the Agreement on Trade-Related Aspects of Intellectual Property Rights), copyright (at least life plus 50 years according to the Berne Convention), or trademarks (potentially perpetual), the term of protection for registered designs or design patents varies from one jurisdiction to another. As examples, the present term for protection of design patents from the filing date is 15 years in the United States, 10 years in China and the EU, while an RCD may be valid for five years and can be renewed to a maximum of 25 years.

Trade secrets

Trade secrets protect information that has commercial value provided steps are taken to keep them secret. Trade secrets are protected under national laws and do not require any formal registration. Almost every IP right originates as a secret. For example, an inventor keeps their inventive concept a secret until they file for a patent. Treating it otherwise will destroy patent novelty,

meaning patent law will not be met and the application will fail, no matter how novel and inventive the invention. Similarly, a writer keeps the detailed theme of their book secret until it is published. Marketing personnel will do the same when planning the launch of a new brand. These early stages of conception often require protection against misappropriation.

On occasion, trade secrets do not only prove beneficial at such preliminary stages, but may prove useful as the primary form of protection throughout most, if not all, of a product's life cycle. This could be the case, for example, where the projected benefit of the technology is of short duration, while as we have seen, the period for obtaining a patent may take a few years. Equally significant is the fact that protection of trade secrets is not limited by time and may last if the subject matter of protection remains a secret.

Unlike other IP rights such as patents, trademarks and designs, trade secrets do not require any procedural formalities such as a precondition for protection. Although eligibility for protection varies by jurisdiction, some general standards for considering the information in question as a protectable secret can be found in Article 39 of the Agreement on Trade-Related Aspects of Intellectual Property Rights, which provides that:

- the information must be secret (i.e., it is not generally known among, or readily accessible to, circles that normally deal with the kind of information in question);
- it must have commercial value because it is a secret; and
- it must have been subject to reasonable steps by the rightful holder of the information to keep it secret (e.g., through confidentiality agreements).

In terms of scope of protection afforded under the trade secrets system, there is one limitation that is of particular importance: information obtained through reverse engineering. Hence, releasing a product bearing such trade secrets to the marketplace makes it available for inspection by competitors with the trade secret being uncovered. This may be mitigated in certain jurisdictions by a carefully drafted license, where such reverse engineering is explicitly prohibited. However, some jurisdictions, such as the EU, may hold such prohibitory contractual provisions as unenforceable.

Summary of relevant legal systems

Copyright

Copyright protects computer programs, which are the basis for mobile apps. Copyright may also protect mobile app screen displays independent from any protection granted to the underlying computer program. It is important to acknowledge that copyright will not protect an idea but only the expression of that idea. It may be difficult at times to distinguish between an idea and an expression, but the distinction lies in the creative choices used. Copyright benefits from existing without formality and, as such, there are few costs associated with acquiring this protection. It also has a considerable length of protection. One area to pay special attention to is who owns copyright. This will typically be the author but in the event the work is created by an employee, it may not necessarily be the case. It is important to be aware of the laws of any relevant jurisdictions. The following is a summary of the basics of copyright as they may apply to mobile apps.

Basics of copyright and mobile apps	
Purpose	Grants rights which allow the author to control reproduction of the work in question
What is protected	Unauthorized copying of an author's work
What is required for protection	To subsist, relevant work must be original Must be a specific expression of an idea
What rights are granted	Right of reproduction (copying) Right of adaptation (right to make derivative works) Right of making available to the public (right of distribution) Rights vary by jurisdiction
How rights are established	Arises automatically once there is creation of a protected work
Duration of the right	Term of protection is the duration of the life of the author plus 50 years
Ownership	Typically owned by the author of the work When the work is created in the course of employment: • Common law: employer may own the copyright • Civil law: employee (author) may own copyright

Patents

Patent rights are different from copyright in that they do not arise automatically. To be granted a patent, there is an application process which requires a fair amount of resources, both in time and finances. Though acquiring a patent may be expensive, a business with a patent portfolio will typically attract more attention and have more appeal to investors. Like with copyright, it is important to be clear as to who owns a particular patent as it is possible to have joint ownership or employer ownership.

Basics of patents and mobile apps	
Purpose	Grants exclusive rights over an invention in return for disclosure of invention in a manner that would allow someone skilled in the art to reproduce it
What it protects	Invention can be a product, process or method
What is required for protection	Invention must be new, not obvious (requires an inventive step) and capable of industrial application
What rights are granted	Right to prevent others from making, using, or selling the patented invention without the owner's consent Rights granted are limited to the territory of the granting jurisdiction Exception are patents granted by the EPO • This may grant a bundle of national rights for the jurisdictions chosen by the applicant • There may soon be the introduction of a unitary right in EU member states
How the rights are established	To be granted, it must be registered The process of registration requires • Formalities examination • Substantive examination Process of registration may take a few years
Duration of the right	Typically, up to 20 years
Ownership	Typically, the owner of the patent is the inventor or joint inventors, unless created in the course of employment. Laws may vary as to how inventions created during employment are treated

Utility models

Functioning similarly to patents, utility models generally require less financial outlay and have less stringent criteria. However, utility models are not available in all countries and may not be available for software-related inventions in a given country. These rights have a usually shorter duration than patents. Utility model schemes are specific by jurisdiction.

Basics of utility models and mobile apps	
Purpose	Protects minor or incremental innovation
What it protects	Primarily, products
What is required for protection	Less strict novelty requirement than patents depending upon jurisdiction "Absolute" novelty not often required Inventive step may not be necessary but if required, threshold is lower
What rights are granted	Like patents, rights may include preventing others from making, using, or selling the utility model without the owner's consent In some countries, these rights may require substantive examination before becoming enforceable
How these rights are established	Needs to be registered In many cases, no substantive examination prior to registration
Duration of the right	Typically, seven to 10 years

Trade dress under trademarks and unfair competition laws

Trade dress is typically used to protect the appearance of a product and may not be instinctively considered when dealing with mobile apps. However, the GUI may warrant this type of protection and has a great impact on its acceptance by users. There is both the option to obtain protection via registration of trade dress or through unregistered protection via unfair competition laws. There is an additional cost to consider when registering the mark and there is no uniform position regarding unfair competition and consequently there is greater uncertainty in relying upon this type of protection.

Basics of trade dress and mobile apps	
Purpose	Protect "get up" features of products or services that serve as indicators of origin
What it protects	The product's appearance, e.g., GUI
What is required for protection	For a registered trademark: subject matter of the application must be clearly defined, not descriptive and should have a degree of distinctiveness Cannot be functional in nature For an unregistered mark: necessary to show that the sign applied to a product or service is distinctive and serves as an indication of origin. It is also necessary to show that a defendant's activities in relation to the signs in question resulted in a degree of confusion or deception in the mind of the relevant public
What rights are granted	Exclusive right to use the trademark or unregistered sign in the context of the relevant goods or services and to prevent infringement
How these rights are established	Can be registered or unregistered
Duration of the right	Perpetual subject to renewal

Designs

The laws in place to protect designs are not internationally uniform and there may be variations in terms of the type of protection afforded. Design laws mainly protect external appearance as seen with trade dress. The design of a mobile app's GUI may be crucial for its success, providing greater appeal to its consumer. There are jurisdictions which have design patents and, consequently, they undergo registration which requires certain standards be met. In other jurisdictions, there is the possibility for a registered design outside of the patent law framework. A design cannot be functional and what is considered functional is a matter of domestic legislation. Also, the term of protection granted to a design varies by jurisdiction.

Basics of design and mobile apps	
Purpose	The creative activity of designing aesthetic or ornamental features of a mass-produced item
What it protects	Original ornamental and non-functional features of an industrial article or product
What is required for protection	Differs by jurisdiction In the United States and China, applicant must show that design satisfies novelty and inventive step Under an RCD, requirements are novelty and individual character Design must not be considered functional
What rights are granted	Protection against third-party infringement
How these rights are established	Design protection may be either registered or unregistered Varies by jurisdiction
Duration of the right	Varies by jurisdiction

Trade secrets

Trade secrets protect information that has commercial value when reasonable steps are taken to keep it secret. This protected information may be technical but could also relate to other important business details such as business plans or financial projections. Trade secrets do not require any formal registration, so these rights can be attractive as there is little upfront cost. However, costs may arise when ensuring appropriate business practices are in place to keep critical information secret. Trade secrets are not protected against independent creation or reverse engineering.

Basics of trade secrets and mobile apps	
Purpose	Provides a level playing field by preventing competitors from gaining unfair advantages through unfair business practices
What it protects	Information that has commercial value by virtue of its secrecy Formula, practice, process, design, instrument, pattern, commercial method or compilation of information
What is required for protection	Varies by jurisdiction International standards state that: • The information must be a secret • It must have commercial value because it is a secret • It must have been subject to reasonable steps by the rightful holder of the information to keep it secret
What rights are granted	Protection against the unauthorized disclosure or use of information deemed trade secrets
How these rights are established	Trade secrets do not require any procedural formalities for protection
Duration of the right	Potentially indefinite if kept a secret

Notes

1 *See* Article 27 of the Agreement on Trade-Related Aspects of Intellectual Property Rights.

2 Except for well-known trademarks.

Chapter 2

Code and Architecture Protectability

The making of software

There are various software development models such as the Waterfall model, V model, Incremental model, RAD model, Agile model, to name but a few. Each model must complete different tests and meet specific time frames. Although these are significant to development teams, they do not pertain to the IP rights for a mobile app's software. There exist some excellent development tools some of which are "no code" or "low code" tools. Such tools enable developers to create mobile apps, usually business centered, within a very short period while involving no, or very little, coding. From an IP law perspective, use of such tools may impact upon some copyright claims over portions of code and architecture.

Intellectual property rights systems and business impact

Code and architecture

Computer code is the most basic building block of software. Whether the mobile app is available for download or use over the Internet or is cloud based, the most basic building block is computer code. Computer programs are created using source code, which is a program language readable and understood by software developers. An example of such a language is Java, used by millions of active developers worldwide. Google's Android operating system uses Java as the basis for all apps written for Android (Android Java is not the same as regular Java but is close). Another example is Swift, which is an open source programming language developed by Apple for iOS and OS X. But when considering computer code in the context of IP protection, attention should not be limited to code available in source code format. To be read by computers, programs in source code format must be "translated" into executable code, also known

as object code. Both source and object codes can be copied or used, in whole or in part, and therefore IP rights may apply.

A computer code's primary form of protection is copyright law. Before the 1990s, it was unclear to what extent copyright law could protect source code or object code formats; however, they are now protected as a literary work within the Berne Convention – the main international agreement governing copyright.

In principle this has the effect of treating computer programs in a similar way as to other, more traditional, copyrighted works such as novels or musical compositions. However, while a book is primarily intended to be read and understood by humans, a computer program's aim is to be understood by a machine. In this way, a computer program and the code comprising it is more akin to an industrial process than to a novel or even a historical textbook. Nevertheless, if parts are copied, they could be subject to copyright infringement. As discussed, copyright does not protect ideas but only an expression of ideas. And although historical facts are situated on the "idea" side of the "idea/expression" divide, copying text from a history book cannot be excused under the idea/expression principle.

Applying the same rationale to computer programs however is less straightforward. There may be public policy reasons for allowing the use or copy of an author's form of expression, such as to achieve a particular function. Functionality per se is not protectable under copyright law and falls into the idea side of the idea/expression divide. Where a particular function or idea may only be expressed in one way or in a limited number of ways, the expression is regarded as incidental to the unprotectable idea or function and is therefore not eligible for copyright protection. Another example of a type of expression that might be ineligible for copyright protection is one dictated by external factors. Although in most cases it is program elements of a higher level of abstraction that more easily fit into this category, it is nevertheless possible that certain forms of expression are dictated by hardware requirements, compatibility and interoperability restraints etc. This will be discussed in further detail.

Once established that a computer code is eligible for copyright protection, the remaining step is to confirm whether the part taken is significant enough to constitute copyright infringement. This would rarely prove problematic. In the EU, for example, to copy a part of an author's own intellectual creation may result in infringement. Such part could be very small in quantitative terms and form a fraction of the overall work from which that part was copied.

All this is to say that a mobile app developer or publisher must be vigilant and ensure that in developing their app no portion of proprietary code, no matter how small, is copied. The same applies to any portion of proprietary code that may be regarded by an app developer as mundane, uncreative or rudimentary. Mobile app developers should avoid using any portion of code written by another to avoid copyright infringement. Determining that a specific portion of code is not eligible for copyright protection involves a highly complex assessment and should be carried out by an expert.

Patent law is not an obvious candidate for protecting code. As discussed, patent law may protect a product, process, or sometimes a product derived from a specific process. Computer code, on its own, does not conform to any such category. A programmer writes computer code to execute various processes. Where available, patents may be used to protect the associated functionality, which may encompass relevant code. The question of patent protection for mobile apps will be discussed in detail in the following chapters on interfaces and functionality.

In the context of computer code, trade secrecy could prove to be an effective form of protection under certain circumstances and may even be the preferable form of protection over all other IP rights. Interestingly, both the United States and EU have found it necessary to update their legislative framework on the protection of trade secrecy.[1]

As explained in Chapter 1, software publishers often use trade secrets to protect their business assets and innovation. With regards to downloadable mobile apps, trade secrets could act as a supplementary layer of intellectual property protection. First, trade secrets could be employed to protect the code, algorithms and

structure of new apps prior to the mobile app's launch. Clauses in employment agreements and non-disclosure agreements with relevant third parties could ensure that confidentiality is maintained until an app's release date. Second, trade secrets law could protect aspects of mobile apps that could not be uncovered through reverse engineering or decompilation once an app has been launched, such as source code commentary and specifications, or new methods for delivering content. Third, trade secrets could protect features that could be uncovered through reverse engineering, such as new algorithms or data structures.

Reverse engineering is a time-consuming and demanding practice often producing uncertain results. Forcing one's competitors to go down the reverse engineering route, rather than poaching an employee that may disclose the sought-after information, could potentially have a few consequences. The length of time and expense involved might be sufficient to deter a competitor from reverse engineering with the effect of such competitor either licensing the relevant information where possible, or abandoning plans to access such information altogether. Where a competitor nevertheless decides to practice reverse engineering and eventually successfully does so, a mobile app publisher may still enjoy a lead-time advantage due to the time it takes to go through the reverse engineering process. Whether or not trade secrets law could be used to prohibit reverse engineering altogether is a question to discuss.

Some mobile apps are not available for download but are offered over the Internet as a service. Since the code of such apps stays out of reach for competitors, its internal architecture might not be accessed through reverse engineering and decompilation. In such cases, trade secrets may serve as the main vehicle for protection against misappropriation of innovative architectural features. Although patent protection may be an option for software-implemented inventions, it may not be a worthwhile option for apps offered over the Internet. As previously mentioned, patents provide protection against independent creation. However, if a patent is obtained, the innovative aspect of the app's architecture must be disclosed within the patent's documentation. This should be considered when comparing the scope and benefit offered by patent law versus trade secrets law.

Trade secrets law can protect mobile app developers from former employees. While labor laws vary from one country to another, a non-disclosure or secrecy agreement could prove more valuable, under the right circumstances, than a post-employment, non-competition agreement. Courts generally construe the latter narrowly and, to be effective, are expected to contain time, geographic and/or industry limitation. On the other hand, a non-disclosure agreement is not necessarily subject to narrow interpretation and does not have to include any of the above limitations.

Understanding decompilation and interoperability

Most computer programs released onto the market are in object code format (i.e., in a format that is incomprehensible to a human). However, a software developer may engage in a process of analyzing a system to create representations of that system at a higher level of abstraction while going backwards through the development cycle. The main method of conducting such a process and gleaning into the source code architecture while recreating the source code that was used by the original developer is known as decompilation or disassembly.[2] The process of working a software product backwards to uncover the original components used is known as reverse engineering. Reverse engineering usually consists of the following stages:

- analysis of the product;
- generation of an intermediate product description;
- human analysis of product description to produce a specification; and
- generation of a new product using the specification.

An app developer may wish to engage in reverse engineering of computer programs, whether they be other apps or operating systems, for two main reasons: to understand the internal workings; and to understand any performance failure. Further to this, understanding the internal working of a computer program has four primary objectives which include to:

1. produce a functional equivalent or a better app (i.e., competition);[3]

2. produce an app that operates with the studied program (i.e., compatibility or interoperability);
3. analyse solutions adopted by the studied program for research purposes;[4] and
4. provide security auditing.[5]

Understanding the performance failures of a computer program is done for diagnostic purposes to understand why a program fails to perform in a desired manner.

Thus, the ability to investigate a program's internal organs may be of significance to app developers. But to what extent should it be possible for an app developer to review another piece of software, be it a competitor's product or software with which an app is intended to interact?

At first glance, the question may appear odd. Why should copyright law restrict a party from uncovering the building blocks that were used in constructing a work protected under our IP law system?

When a composer wishes to understand the structure of an existing musical composition they can listen or read and its building blocks become readily apparent. Recalling his early steps as a songwriter, Bob Dylan described his fascination with the song 'Pirate Jenny,' by Kurt Weill and Bertolt Brecht:

I found myself taking the song apart, trying to find out what made it tick … I took the song apart and unzipped it – it was the form, the free verse association, the structure and disregard for the known certainty of melodic patterns to make it seriously matter, give it its cutting edge. It also had the ideal chorus for the lyrics. I wanted to figure out how to manipulate and control this particular structure and form which I knew was the key that gave 'Pirate Jenny' its resilience and outrageous power.[6]

Dylan's wished to "unzip" what he recognized to be a successful, if not unique, work so he could understand "what made it tick" and become a better songwriter was only natural. What Dylan did to 'Pirate Jenny' could be described as reverse engineering.
The legal position for software protection is different. Unlike a

musical composition, the processes of reverse engineering and decompilation of software may be regulated under copyright law. This is because, for one to study and understand the software, it must be converted from object code (understood by computers) to source code (understood by humans). This conversion implicates copyright law in two distinct ways.

First, the target program needs to be uploaded multiple times as part of the decompilation process. Each time it is uploaded, a copy is created. Where there is no authorization from the right holder to such copies in the context of decompilation, a right holder's reproduction is violated. Second, the pseudo-source code resulting from decompilation may amount to an infringing derivative work. Either way, it appears that decompilation without authorization from the copyright holder can amount to copyright infringement unless exempted under a copyright law exception.

At first glance, the legal position in the United States appears to be satisfactory. The U.S. Copyright Act acknowledges the idea/expression dichotomy, but does not provide for an explicit exception from liability in the case of decompilation. We recall that under the principle of the idea/expression dichotomy, it is the expression of ideas rather than ideas themselves that are eligible for copyright protection. Hence, copyright law recognizes the importance of excluding from protection, the ideas, methods, processes, concepts etc. used by others for their own endeavours. On that basis, one may argue that reverse engineering and decompilation practices should be permitted. After all, if certain ideas and concepts are hidden behind software's technical veil of executable code, it would make sense to allow a minimal degree of infringement to enable access to such elements.

In fact, that is exactly what the Institute of Electrical and Electronics Engineers (IEEE) in the United States has maintained for the last two decades. In its Board of Directors Position Statement on reverse engineering from June 2008, after acknowledging the importance of granting copyright protection to expressive elements in a computer program, the IEEE emphasized the importance of learning and studying non-protectable elements:

Congress did not, under the Copyright Act, protect the ideas contained in that expression. Rather, Congress desired that the ideas contained in works, including computer programs, should be available for use by and to reach others. Accordingly, we consider it appropriate to perceive and learn those ideas by lawful means of 'reverse engineering.'[7] *More than appropriate, the IEEE makes it clear that the ability to study the program through reverse engineering is of paramount importance; they stated: 'We further believe that lawful reverse engineering of computer programs is fundamental to the development of programs and software-related technology'.*[8] *It is of such fundamental importance since it may assist engineers in 'designing competing products that are not substantially similar in expression, as well as to discover patentable subject matter and ideas not otherwise disclosed in the literature provided with the product by the originator'.*[9]

This remains the approach adopted by the United States courts. In three separate decisions (*Atari*, *Sega* and *Sony*), different U.S. circuit courts of appeal have ruled that decompilation done for the purpose of gaining access to unprotectable elements of computer programs may amount to "fair use" under U.S. copyright law.[10] Notwithstanding the potential legitimacy of such practice under copyright law, it is mutually exclusive of any contractual prohibitions. Hence, such practices may be prohibited in the United States even if permitted under copyright's "fair use" doctrine.

Enabling the access and study of such elements through decompilation should be distinguished from reproduction of such elements, an activity that is separately assessed under copyright rules on infringement. As previously discussed, various aspects of a computer program such as algorithms and data structures may not amount on occasions to copyrightable subject matter and can therefore be reproduced.[11] Often the only effective way to gain access to such elements is through decompilation.

Finally, the most fundamental reason for a mobile app developer to carry out decompilation is interoperability. This is defined as "the logical and, where appropriate, physical interconnection […] to permit all elements of software and hardware to work with other software and hardware and with users."[12] To allow application

software to interact with an operation system, an app developer must gain access to the application programming interfaces (APIs). If such information is not made available by the proprietor of the operation system, the only effective method of obtaining it may be through decompilation.

There are two types of interoperability scenario, both of significance to the software market. The first is horizontal interoperability. It refers to the information that enables the reverse engineer to develop their own operating system platform to be compatible with existing applications, which, in turn, are compatible with the target operation system platform. This type of interoperability may lead to the creation of a competing platform that is compatible with other applications but not necessarily with the target software itself.

The second type of interoperability is vertical interoperability, which is of more relevance to the present discussion. Here decompilation of an operating system platform may take place enabling the creation of downstream compatible app software. For obvious reasons, most software vendors will be more reluctant to disclose horizontal interoperability information than vertical interoperability information as the latter may be in their economic interest. Hence, such a vendor may have an interest in having their vertical interoperability information available. After all, the more apps written to such a platform, the more popular it may become. However, other factors may come into play, including where a vendor may choose not to license its interoperability information to a particular app developer. For example, it is reported that Apple refused to authorize an iPhone app that measured mobile phone radiation, an app that purported to provide diagnostic information that might protect one from hackers, and the “I Am Rich” app.

Regardless of the type of interoperability concerned, having access to interface information is of importance to the mobile app market. As a representative of the U.S. Government stated, “To control the interface specifications is to control the industry.”[13] It is for this reason that the EU Software Directive[14] provides for a limited exception to copyright infringement in the case of decompilation to achieve interoperability.[15] Unlike the position under U.S. law, which is based on judge-made exceptions, the EU has a clear statutory

exception for decompilation to enable interoperability. It is noteworthy that the Directive falls short of imposing a positive obligation to disclose interoperability information. At best EU law does not enable a copyright holder to rely on their copyright nor prohibit others from uncovering such information through decompilation. As we have discussed, decompilation is a technically complex, costly and time-consuming practice that is best avoided where possible.

The legal position in the EU regarding decompilation of computer programs is straightforward, as it is codified under Article 6 of the Software Directive. It states that translation from machine readable code to human readable code does not require the authorization of the right holder where such translation is, "indispensable to obtain the necessary information to achieve the interoperability of an independently created computer program with other programs". Thus, it is only decompilation with a view to achieving interoperability that might be excused under Article 6. Three additional conditions must be fulfilled for decompilation to be legitimate under Article 6. First, the act must be performed by a licensee or a lawful user of the program. Second, the information sought must not be available to the party carrying out the act through any other means. Finally, the act of decompilation must be confined to those parts of the program necessary to achieve interoperability.

Obtaining information necessary for interoperability does not serve the idea/expression dichotomy principle or the public's interest. Hence, the scope of permissible decompilation under Article 6 could be contrasted with the rest of the Directive's language. The Directive explicitly states that ideas embodied in a computer program remain outside the scope of copyright protection, while at the same time prohibiting decompilation except for the limited purpose of achieving interoperability. Unlike the legal position in the United States, where decompilation may be permissible to gain access to non-protectable elements under copyright law, EU copyright law provides that only decompilation to achieve interoperability is permissible. All other instances, whether or not done to gain access to unprotectable elements, are likely to give rise to copyright infringement.

A mobile app developer engaged in reverse engineering by decompilation should ensure that the output of a decompilation

process does not harm a newly developed software product with unauthorized copyrighted material. For this reason, developers often employ a technique known as “clean room.” For example, if one wishes to reverse engineer a computer program to create a competing program that emulates the first program, it is a risk that its newly created program may constitute a copyright infringement. Thus, a reverse engineer may employ a “clean room” technique in the following fashion. First, a team of engineers would study and analyze the code of the emulated program; if the program is available only in object code format, it would be decompiled and reconverted to source code format.[16] Studying the program in its comprehensible format, engineers would then describe all aspects of the program without using or referencing any actual code.[17] At this stage, an additional team of programmers would step in and, without prior knowledge of the reverse engineered system or access to code, write a new program to operate as specified.[18] In this way, the resulting code program will be different from the emulated program although for the most part it may operate identically. Using the clean room approach avoids copyright infringement as software functionality is generally not eligible to copyright protection.[19]

Unlike a book, a musical CD or most other works protected by copyright, software products are rarely sold to the public. This has considerable restrictions for the end user. In the case of a sale, the purchaser is entitled to do whatever they wish, so long it does not involve copyright infringement. However, should the transaction be classified as license rather than sale, the licensor might be able to retain the ability to control the type of use through various contractual provisions. As most app publishers will acknowledge, those same advantages that may attract them to the licensing model are the ones they should bear in mind when contemplating reverse engineering and decompilation of a competitor’s product. In principle, a licensing model allows the licensor to place restrictions that go beyond any restriction under copyright law. A licensor may attempt to prohibit decompilation that is otherwise permitted under copyright law. It is therefore necessary to examine what licensing provisions are likely to prove valid.

As with decompilation, there are jurisdictional discrepancies in attitudes towards licensing. For example, in the EU, the

Software Directive makes it clear that the restricted scope for legitimate decompilation can not be curtailed through contractual mechanisms. Namely, any contractual provision that seeks to ban decompilation as permitted under the Directive is null and void. The EU maintains that the scope for decompilation permissible under the Directive is supported by clear public policy considerations. It is not possible to circumvent these public policy considerations via mere contractual arrangements. To contrast, the position in the United States is that an app developer may engage in decompilation in a variety of circumstances. Although from copyright law perspective decompilation is permissible in the United States, in a wider scope of circumstances, this could easily be addressed by the copyright holder who, as a licensor, may restrict the possibility of decompilation, if not ban it altogether, via suitable licensing provisions. Since almost without exception, most types of software are made available subject to license, the U.S. approach may result in rendering the space for decompilation effectively very narrow.

The final issue to be considered for copyright protection is technological protection measures. Except for attempting to prohibit reverse engineering by using contractual provisions, a right holder may seek to rely on the application of technological protection measures and a variety of anti-circumvention provisions to achieve a similar goal. How can the anti-circumvention provisions restrict reverse engineering? Anti-circumvention laws prohibit the cracking of technical measures that are applied to digital works such as access control or copy control mechanisms. Contrary to common belief, such laws are not limited to a prohibition on cracking of digital right management systems, such as copy protection mechanisms applied to DVDs.

In both the United States and EU, the language of such anti-circumvention provisions is broad enough to encompass techniques such as authentication handshakes, code signing, code obfuscation and protocol encryption, which may all qualify as technological protection measures covered by anti-circumvention provisions.[20]

In Europe, the Information Society Directive provides that it shall have no effect on the protection of computer programs.[21] Thus, the elaborate anti-circumvention system that was set under the

Information Society Directive has no application to computer program code and architecture protection.[22] It is the Software Directive that regulates the European anti-circumvention system with respect to computer programs. Unlike the Information Society Directive, the act of circumvention itself is not restricted under the Software Directive and it is only the act of trafficking in circumvention tools that is prohibited. Even in Article 7(1)(c) of the Software Directive, which concerns circumvention tools designed to facilitate circumvention or removal of any technical device protecting a computer program, it is explicitly stated to be without prejudice to Articles 5 and 6, which deal with reverse engineering and decompilation respectively. The EU-limited exception of decompilation for the purpose of achieving interoperability cannot be overridden by the application of technological protection measures. Where a right holder applies technological protection measures to a computer program, circumvention of such measures is not restricted under the Software Directive.

In the United States, the position is somewhat different. First, according to the U.S. Digital Millennium Copyright Act (DMCA), circumvention means, "to descramble a scrambled work, to decrypt an encrypted work, or otherwise to avoid, bypass, remove, deactivate, or impair a technological measure, without the authority of the copyright owner." For example, a computer program on the market with its code, or parts of it encrypted, may necessitate decryption to be decompiled. Under the DMCA such decryption may amount to circumvention. The same applies to acts such as descrambling or to deobfuscate. As previously noted, decompilation to gain access to unprotectable elements of a computer program may be permitted under the broad exception of "fair use." Permissible decompilation is not necessarily limited to cases of interoperability.[23] The introduction of the DMCA has effectively changed this position. Unlike the position in Europe, the DMCA does not distinguish between computer programs and other copyrighted works in a digital format; the same system applies to all.

An exception authorizes circumvention and development to identify program elements necessary for achieving interoperability. But what about decompilation for other purposes? Although prior to the DMCA decompilation may have been permissible under copyright

law for purposes other than achieving interoperability, applying technological protection measures may have the effect of limiting the ability of a competitor to decompile a computer program only to scenarios involving interoperability.

The legal position on decompilation for purposes other than achieving interoperability is, however, not certain. Another provision of the DMCA provides that rights, remedies, limitations or defenses to copyright infringement under the DMCA should not be affected under the anti-circumvention provisions, including "fair use." It may be argued that since prior to the introduction of DMCA decompilation was permissible under the "fair use" defense for a wider range of purposes and should remain permissible. Unfortunately, there is no uniformity on judicial approach as to whether circumvention of technological protection measures amounts to copyright infringement, and not intended to achieve interoperability, is listed under Section 1201(c)(1).

IP protection and the cloud

With software being accessible exclusively in cloud platforms and this option becoming ever more common, one must consider the relevant IP rights available to protect valuable aspects of mobile apps. Mobile cloud apps and mobile web apps are similar in that in both cases the app is run on an external and independent server. In both cases the app is available by access over the Internet. While all cloud mobile apps are web mobile apps, the opposite is not necessarily true: not all web mobile apps are cloud apps as some of the former are written to be run and stored on a physical server. However, for this discussion we will not distinguish between the two as the following information is equally applicable to any type of web mobile apps.

What distinguishes cloud-based mobile apps from what may be referred to as native mobile apps is that the latter are downloaded and installed on a mobile device. With native mobile apps, various versions are written as if the app was to run on various operating systems such as Windows, Android and iOS. Cloud-based mobile apps can be written in only one version and any device with an

Internet browser can access and use it, no matter what type of operating system. This is a clear benefit to the app developer.

If the app is not downloaded over the Internet, no one has access to the software code and architecture except for the developers or other employees. Therefore, as far as the code and internal architecture of the app are concerned, copyright law becomes less relevant.

Under patent law, elements of the code and architecture may be protected. However, trade secrecy may provide the most appropriate form of protection to code and architecture under these circumstances. The development team and other employees with access to the code and architecture could and should be subjected to a well-drafted confidentiality agreement. This could help ensure that such aspects of the mobile app remain secret if the app publisher wishes them to remain so. From an IP rights perspective, other valuable aspects of mobile apps, such as its functionality and GUIs, are hardly affected by the app's availability solely within a cloud environment. Therefore, the following discussion regarding protectability of GUIs and functionality does not distinguish between native mobile apps and cloud-based mobile apps.

Summary of IP rights and app code and architecture

When designing software there are various models and methodologies that can be employed. In addition, tools have been created which enable quick mobile app development requiring little coding. However, the use of these tools may impact the extent of copyright protection over portions of code and architecture. The following provides a summary on IP rights of protection for a software's internal organs.

Intellectual property and code

At the core of mobile apps is computer code. Computer programs are created using source code which are instructions written in programming language readable by humans. For an intellectual

property protection, the focus should not be limited to source code. A computer does not '"execute" source code and must be translated to executable code via a compiler. This executable code is also known as object code and IP protection does not differentiate between source code and object code.

Types of IP protection for computer code	
Copyright	Copyright is the primary form of protection for computer code. Source code and object code are considered literary works under the main international treaty governing copyright (Berne Convention). The difference between most literary works and computer code is that, while literary works are created to be consumed by humans, computer code is not. It is a series of statements created to instruct a machine and manipulate data. In more traditional works of authorship, verbatim copying is not permitted even when discussing facts. However, for computer programs, there can be public policy considerations that support copying of a specific form of an expression created by the initial author. This can be, for example, when the expression is dictated by hardware requirements, compatibility or interoperability constraints. If it is concluded that a piece of software warrants copyright protection, then to establish infringement, it must be determined whether it was of significance. For example, in the EU, copying of a part that constitutes the author's own intellectual creation may result in infringement. Therefore, for a developer, it is essential not to copy any portion of proprietary code. An app developer should avoid using a portion of code written by someone else regardless of size, function or nature unless they are certain that it is not eligible for copyright protection. Any uncertainty could be cleared by an IP expert.
Patent	Patents typically protect products, processes and sometimes a product derived by a particular process. They are granted for software-related inventions in some countries. Such software-related inventions may cover aspects of a computer program where the invention is not abstract or contain non-technical subject matter. If only a portion of software-related invention code has been used by an unauthorized party, it may not necessarily and automatically lead to patent infringement. This may be determined by a local legal expert.
Trade secrets	In certain circumstances, trade secrets may be an effective form of protection for mobile app developers. These rights can be used to protect code, algorithms and the structure of an app prior to its release. It may also protect aspects of the app that may not be uncovered by reverse engineering or decompilation. Conversely, it may also be beneficial to protect aspects of the app which can be reversed engineered such as algorithms or data structures, but this is costly and time consuming. For mobile apps that are not downloadable, trade secrets can serve as a main source of protection as it may prevent third parties from misappropriating the innovative components of the architectural features.

Decompilation and interoperability

Most computer programs are released in object code format. To examine the subject system and to gather an understanding of the source code and its internal architecture, one would partake in decompilation or disassembly. Reverse engineering is the process of working a software product backwards to uncover the original components. This process can be broken down into the following stages:

- analysis of the product;
- generation of an intermediate produce description;
- analysis of product description to produce a specification; and
- generation of a new product using this specification.

Reverse engineering of software may be placed under the scrutiny of copyright law due to its technical particularities. The focus of this section is on decompilation. This allows for software release in object code to be converted back to source code to be examined in a format understandable to humans. This conversion runs against copyright law in two aspects:

- the program must be uploaded multiple times as part of the decompilation process. Each time it is uploaded, an unauthorized copy is created; and
- this pseudo-source code may constitute an infringement.

United States legal position

Under U.S. copyright law, there is no specific exception from liability for decompilation. However, under the idea/expression dichotomy, the law values the importance of ideas, methods, processes and concepts. Copyright does not protect ideas which are contained in the expression of software code. United States' courts have ruled that decompilation for the purposes of gaining access to unprotectable elements of a computer program may amount to "fair use." A distinction should be made between enabling access and study of software elements through decompilation and reproduction of these elements created as a result of reverse engineering.

One of the reasons decompilation may be carried out is due to interoperability. Interoperability is the connection permitting software and hardware elements to work together with other software and hardware elements. There are two types of interoperability, namely horizontal and vertical. For the purposes of mobile app development, it is the second type which is of significance. In this instance, decompilation of an operating system's platform may facilitate development of compatible application software. It is necessary to have access to interface information however, as mentioned, the United States does not have a specific exception for decompilation for interoperability and instead relies upon the general "fair use" defense. It is advisable to seek professional legal advice before decompiling a third party's program.

EU legal position

The legal position surrounding decompilation is codified in Article 6 of the Software Directive. Article 6 allows for translation from machine readable code to human readable code without the authorization of the right holder where it is found to be indispensable to achieving interoperability. There are three criteria required to meet decompilation under Article 6. These are that the:

- act must be performed by a licensee or a lawful user of the program;
- information sought must not be available to the party carrying out the act through any other means; and
- act of decompilation must be confined to those parts of the program necessary to achieve interoperability. In the EU, only decompilation for the purposes of interoperability are legally permitted. It is advisable to seek professional legal advice before decompiling a third party's program.

Utilization of decompilation findings

There should be a distinction between the process of decompilation and the use of the output of that process. Regardless if the process is legally permissible, the use of the uncovered information may

not be. A developer should separate the decompilation output from the process of developing a new mobile app so as not include any unauthorized copyright material therein. To achieve this, a "clean room" approach should be taken. This allows for separation between decompilation findings and creation of new software. Such procedures should be carried out after obtaining professional legal advice.

End-user license agreements

It is rare that a software product is sold to the public outright; it is typically distributed via licensing agreements. A license may allow the licensor to retain control of software usage through the contract. A chosen licensing model can allow the licensor to place restrictions which go beyond the scope of copyright law including preventing decompilation. In the EU, the Software Directive makes it clear that decompilation for legitimate purposes cannot be waived via contractual provisions. In the United States, the defense of "fair use" may be of less benefit where decompilation is restricted by contract.

Technological protection measures

A right holder may rely on the use of technological protection measures and anti-circumvention provisions to prohibit reverse engineering. Anti-circumvention laws prohibit the cracking of technical protection measures which are applied to digital works. In Europe, the Software Directive regulates the anti-circumvention system related to computer programs. Under this Directive, the EU-limited exception to decompilation for the purpose of interoperability cannot be superseded by technological protection measures since it is permissible to circumvent technological protection measures. In the United States, under the Digital Millennium Copyright Act, decryption could amount to circumvention of technical protection measures. However, there is an exception for circumvention that serves to identify program elements necessary for interoperability. It is permissible to circumvent technological protection measures to enable decompilation that is carried out for the purposes of interoperability.

The cloud effect

Mobile cloud apps and mobile web apps are similar in that both run on a server external to the mobile device. As opposed to a cloud-based app, a native mobile app is downloaded and installed on a mobile device. A cloud-based app benefits from only being written in one version so that anyone with an Internet connection can access and use it. Also, with the app not being downloadable, no one may gain access to the software code and architecture except developers and other key employees. Copyright protection may therefore be less crucial against competitors as access to code and architecture is not available. Copyright protection may still prove effective against a developer or employee who appears to use the code. In this instance, trade secrets may be the most effective form of IP protection for code and architecture. Relying on this form of protection requires taking reasonable steps to ensure the mobile app remains a secret, such as through confidentiality agreements and robust security protocols with the cloud supplier.

Notes

1 Defend Trade Secrets Act of 2016, 18 U.S.C. § 1835(b) and Directive of the European Parliament and of the Council on the Protection of Undisclosed Know-How and Business Information (Trade Secrets) Against their Unlawful Acquisition, Use and Disclosure. Directive 2016/943.

2 The original source code can never be fully recovered. Since a great deal of the original programmer's instructions, including commentary, notations, and specifications, are not included in the translation from source to object code, such information cannot be recovered through decompilation. Because the recreated source code forms only a part of the original source code, it is sometimes referred to as pseudo-source code.

3 One may wish to create a functionally equivalent program for non-competitive purposes. The most common reason is where an app's source code, written a long time ago for a legacy system, is not available and it needs to be transported to a new platform. Under these circumstances, the app may need to be written from scratch or, alternatively, decompilation may be used to understand the internal working of the program, before it can be written again.

4 Which may be done either with commercial or non-commercial objectives in mind.

5 Identifying viruses, malware or spyware code in an installed program.

6 Dylan, B. (2004). *Bob Dylan Chronicles: Volume One*. Simon & Schuster, 275.

7 IEEE-U.S. (2008), 1. This statement was developed by the IEEE-USA's Intellectual Property Committee. IEEE-USA advances the public good and promotes the careers and public policy interests of more than 215,000 engineers, scientists and allied professionals who are U.S. members of the IEEE.

8 Ibid.

9 Ibid., 2.

10 Section 107, U.S. Copyright Act 1976.

11 See Chapter 2, The idea/expression dichotomy and decompilation.

12 Directive 2009/24/EC of the European Parliament and of the Council of 23 April 2009 on the legal protection of computer programs, recital 10.

13 *United States v. Microsoft Corp.*, 87 F. Supp. 2d 30 (D.D.C 2000).

14 Directive 2009/24/EC of the European Parliament and of the Council of 23 April 2009 on the legal protection of computer programs – known as the EU Software Directive.

15 Directive 2009/24/EC, Article 6.

16 This broadly corresponds to stages (1) and (2) within the Decompilation and interoperability section.

17 Corresponding to stage (3) within the Decompilation and interoperability section.

18 Corresponding to stage (4) within the Decompilation and interoperability section.

19 Of course, where the system in question is protected by a patent, emulating its functionality may result in infringement. However, under such circumstances reverse engineering is most likely to be redundant as the patent's specifications should disclose the program's functionality.

20 As discussed, this problem has little relevance to computer programs in the European context for reasons other than the statutory definition of "an effective technological protection measure."

21 Article 1(2)(a).

22 Not all aspects of computer programs are governed under the Software Directive. For example, GUIs are governed by the Information Society Directive. See Chapter 3 on Legal and Business Aspects of Protecting Interfaces.

23 See the broad language employed by the circuit courts in *Atari Games Corp. v. Nintendo of America Inc*., 975 F.2d 832 (Fed. Cir. 1992), *Sega Enters. td. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992) and *Sony Computer Entertainment, Inc. v. Connectix Corp.,* 203 F.3d 596 (9th Cir. 2000), cert. denied.

Chapter 3

Legal and Business Aspects of Protecting Interfaces

Graphical user interfaces (GUIs)

Graphical user interfaces are the point of contact between the device and the user and include graphic elements such as icons, menus, text boxes, scroll bars, as well as animated features. On computers, user-friendly GUIs have replaced the need for most text-based commands. Given the small size and proportion of mobile devices, text-based commands are impractical.

As GUIs are user-friendly, use of smartphones has increased and they are used by all generations. Additionally, a well-thought-out GUI does not require a user to remember many text commands; and indeed can contribute to the overall success of a product or service. Users do not have access to, nor are they interested in, the underlying code and algorithms that form the basis of mobile apps. If a GUI is user-friendly, intuitive, easy to navigate and, in the case of some, entertaining to use, then the app will succeed. Successfully protecting one's developed GUI against imitation and emulation can give app developers and publishers a clear competitive edge.

IP protection and impact on GUIs

Mobile apps are multifaceted products which may be protected by a variety of IP rights. After examining IP protection for mobile apps' internal organs, such as software code and architecture, we now move to examine an "outer" facet of mobile apps, namely its GUI. *Britannica* defines a GUI as a graphical interface facilitating human interaction with an electronic device. For the purpose of our examination, think about a GUI as comprising three categories:

1. desktop or overall interface outlay;
2. individual components included in the desktop; and
3. ephemeral and animated features, often an outcome of a user's interactions with categories (1) and (2).

Desktop refers to the overall screen display, including everything presented on it. Components include items such as icons, pointers, menus, scroll bars and any other notable component forming part of the desktop. Ephemeral and animated effects refer to features such as the genie effect in Mac OS and minimizing or maximizing desktop components.

That a GUI is visible to mobile app users is significant from an IP rights perspective. Protection of trade secrets is of little relevance when considering the protection of a GUI once a mobile app is available, whether in a native or cloud-based form. The item is available to inspect and study and any trade secret that may have existed in relation to such visual appearance at the GUI development stages loses its secrecy. However, trade secret protection may be important prior to the mobile app's launch.

Copyright protection of GUI

Computer code or the portion of a computer program that represents the GUI is subject to copyright protection. But sometimes protecting your GUI at a code level might not prove adequate. Consider a situation where a competitor wishes to replicate aspects of your GUI within their own mobile app. Rather than making a verbatim copy of the code that stands as the basis of your GUI, they leave the code altogether and concentrate only on its visuals. For example, the competitor might show your GUI to a programmer and ask them to write a new a computer program that generates the exact same or very similar GUI. In such a case, copyright protection to the GUI code is of little use. The computer code created was not copied, or even accessed. What your competitor did was to generate the same outcome in GUI terms without copying the code. Therefore, copyright protection in such cases will need to be assessed at a different level: the screen display itself. Could copyright protection be established in relation to the GUI's visual aspect regardless of the code type used to generate it? Could copyright law protect a desktop's appearance and the look and feel generated by the organization and sequencing of icons and items? The short answer is that it depends.

Starting with the static appearance of the overall desktop, in principle there is nothing to preclude it from copyright protection. The difficulty may be in proving that the display is both original and non-functional within the meaning of copyright law. This means that many elements of a desktop may be commonplace and would therefore fail to satisfy the originality requirement under many copyright systems around the world. Furthermore, even desktop designs that are not commonplace may not be eligible for copyright protection if they are considered to be dictated by functional considerations.

The Agreement on Trade-Related Aspects of Intellectual Property Rights agreement, of which most countries in the world are signatories, provides that copyright protection does not extend to ideas, procedures and methods of operation. It is in this context that the functionality of a GUI may be assessed for copyright purposes.

What is less clear and may vary from one jurisdiction to another is the meaning of "functional." For example, does functional refer to the GUI's functionality whereby the developer had no design alternatives available to achieve the same function in a successful manner? Perhaps it would suffice to show that the main objective of the developer was to serve a functional purpose, no matter how many alternatives were available? Finally, maybe it means that if the GUI is intended to interact with a device it is considered functional. If it is the latter, the vast majority mobile app GUIs, no matter how creative they may be, are not likely to be eligible for copyright protection.

As is often the case, the truth lies somewhere in the middle. Most jurisdictions would offer, in principle, copyright protection to GUIs that are creative and not commonplace and which resulted from developers exercising choice not dictated by functional considerations.

Furthermore, even where copyright protection is available, it may only protect against verbatim copying. For example, in one UK case, the High Court of England and Wales found that the web interface of a flight booking system, which was the main feature contributing to the system's look and feel, was not eligible for copyright protection, no matter how valuable it was deemed.

Copyrights for individual components may be more straightforward. For pointers, for example, they should contain an artistic embellishment to be considered original and non-functional within the general rules of copyright law discussed in Chapter 1. The same applies to icons. For example, an image of a camera for a photography function is not likely to attract copyright protection, while the ghost silhouette of Snapchat does. Menus, which are usually a collection of words representing commands and instructions, may also be protected when they are original and non-functional. Although it is highly unlikely that the actual words chosen to represent such commands will attract copyright protection in isolation, their selection and organization in a particular menu may represent arbitrary choices made by the developer. Therefore, it is easier to argue for originality in the selection, arrangement and organization of menu commands rather than in the individual commands.

However, clearing the originality hurdle may not be sufficient, as a combination of commands may be deemed to be functional since it may be regarded as a method of operation. One circuit court of appeal in the United States concluded that this was indeed the case regardless of whether the software developer had other alternatives in designing its menu command hierarchy. Such a generous interpretation of what constitutes unprotectable “system,” “process” or “method of operation” did not gain widespread acceptance in other circuits in the United States and neither was it followed in most jurisdictions around the world.

Therefore, an arbitrary selection of command names for mobile apps desktops may be eligible for copyright protection when it is not commonplace and not dictated by functionality.

Animated effects often represent some form of an idea and require artistic embellishment and, to be considered for copyright protection, must be considered as original and non-functional.

This represents an overview of copyright eligibility considerations. It does not touch upon the question of infringement and applicable defenses. Although elements of an interface may be eligible for

copyright protection, it does not automatically follow that any copying of such elements is actionable. While eligibility for copyright protection is mainly assessed based on the subject matter, infringement and the applicability of a defense is usually based on the circumstances in which the elements were copied.

Designs and GUI

Ornamental or aesthetic aspects of GUIs may be protected in some jurisdictions under an IP system. For example, in the United States and Japan they may be protected as design patents while in the EU they may be protected as a registered community design (RCD). The overall imagery and other visual effects of GUIs have been protected under such dedicated IP systems for the past few decades. Unlike copyright, but like patents, such systems require registration and involve an up-front cost. However, once design patents are registered they give a presumption of validity and may deter a competitor.

To qualify as a U.S. design patent, a design must be new, non-obvious, and ornamental. While the first two requirements are not much different from their utility patent equivalents, it is the requirement of ornamentation that may appear problematic as GUIs are intended to achieve a utilitarian rather than ornamental purpose. Therefore, functional designs, as opposed to ornamental designs, are not subject to design patent protection. In practice however, the functionality exclusion is interpreted narrowly in U.S. design patents. Only designs that are purely functional are excluded. In fact, in most cases, a GUI will be eligible for design patent protection even where it simultaneously ornamental and functional. U.S. versions of design patents are distinguishable from utility patents in their scope of protection. An infringement occurs where an observer observes the two designs to be substantially the same.

The EU RCD system is similarly attractive to GUI designers of mobile apps. It offers protection to designs that are novel and have “individual character.” There is no substantive examination prior to registration and therefore registration will become effective

once the application form is completed and application fees paid. Any substantive opposition that a GUI's RCD may meet would be post-registration. In such a case, a defendant is likely to have two different defense options. First, it may be argued that the design has not been infringed upon because the defendant's design produces a different overall impression, the latter being the test for infringement under the EU-registered design system. Second, it may be argued that in any event, the design should not have been registered in the first place and should be invalidated.

A main obstacle to a mobile app's GUI, is the explicit technical function exclusion. Here, the EU RCD Directive explicitly states that, "A Community design shall not subsist in features of appearance of a product which are solely dictated by its technical function."[1] The rationale is like the functionality exclusion under U.S. design patent law: to obtain protection over technical functional aspects of a product, a person may turn to the more onerous system of utility patent law; otherwise, the result may hamper competition and technological advancement. The functionality exclusion for RCD essentially means that a court should establish whether there was any factor other than technical function, such as aesthetic appeal, which led to the choices made by the designer. Where that is not the case, the design at issue is likely to be found to be "functional" and thus excluded from protection.

Whether it be the United States' or Japanese design patent systems, or an EU-registered design system, eligibility for protection hinges on two substantive requirements. While in the United States, it is novelty and obviousness, under the EU RCD system it is novelty and overall expression. Obviousness under the United States' system is conceptually like the obviousness requirement under its better known utility sibling, although less onerous. Overall impression refers to whether the design's impression on an informed user differs from an overall impression of an earlier design publicly available. In both the United States and EU, this test involves measuring the "distance" between that which has gone before (i.e., the "state of the art" in patent law parlance) and the subject matter of the application. The objective is to guarantee that mundane deviations from the state of the art, irrespective of being novel, will not be granted protection.

GUIs and trademark protection and unfair competition

Trademark protection is primarily about protecting signs applied to goods or services, as an indication of commercial origin. Unfair competition will be discussed in brief, although its potential scope may be much wider. As mentioned in Chapter 1, the former involves registration in order to be to be effective, while the latter does not.

To what extent is a GUI capable of identifying the source or origin of a mobile app? A non-commonplace GUI to one's mobile app is not sufficient. One must demonstrate that the public perceives it as an indication of origin. It means treating a mobile app's user interface as a particular source of origin without any reference to other signs such as a logo or brand name (if such logo or brand name do not constitute part of the screen layout).

Trademark law

As mentioned, trademark protection depends upon prior registration. There is no global consensus regarding entitlement to register. Some trademark systems, such as in China, Japan, the Russian Federation and the EU, operate a "first-to-file" system, where entitlement and priority does not depend upon use, and prior use is not a prerequisite for registration. In these systems, applicants must recognize that registration entitles protection but use does not give priority to trademark rights. There may be literally a race to the registry office where two conflicting marks are involved, with the first to register a trademark application prevailing. Other trademark systems such as those in the United States, Canada and India operate on the basis of a "first-to-use" principle, where trademark rights are based on adoption and use rather than on registration. Nevertheless, in these countries, registration is also highly desirable as it strengthens the proprietor's use-based rights.

Registrability requirements may also vary between jurisdictions, although there are general criteria to meet. As with any registration, the subject matter must be clearly and precisely defined. While this might not pose problems to static GUI features, such as a

complete user-interface or elements such as icons, it may prove more challenging for animated features. Some jurisdictions enable the registration of animated features, which are sometimes referred to as “motion marks.” These may be represented in application documents as a series of frames, capturing the movement sequence. Both the United States’ and EU trademark offices allow registration of such marks, which are expected to satisfy trademark law requirements in the same way as conventional trademarks.

Additionally, a GUI trademark application must show that the subject matter has a distinctive character and may serve as a designation of origin. It is likely an applicant will have to show that the GUI or its features have a distinctive character through use, as it is less likely to be inherently distinctive. Consequently, an applicant may have to rely on the period of prior use where consumers view the GUI as indicating a source rather than merely enabling interaction with a device. Even if an applicant manages to meet the distinctiveness threshold, a GUI registration must also not fall foul of the functionality exception.

As mentioned in Chapter 1, functionality exceptions exist under most trademark systems, such as in the United States, EU and China, and state that product features whose objective is functional or technical are not eligible for trademark protection. Therefore, for mobile app GUIs, if the feature to be protected is essential to the function or to achieving a technical result, it may not be eligible for trademark protection. Such features may be protected under patent law, should they satisfy the strict set of requirements under that branch of IP law.

This distinction is particularly relevant in the context of GUIs, the majority of which are ultimately intended to enable a user to interact with a mobile device. Where the feature over which trademark protection is sought could be characterized as essential to the function of the mobile app at issue or as necessary to achieve a technical result, it may be considered as technical or functional and thus not eligible for trademark protection. While most people would agree with the general principle, according to which trademark law is not designed to reward technical or function-related innovation, there is a variety of views as to what constitutes technical or functional for the purpose of trademark law; each jurisdictions

has its own test and draws its boundaries as to the scope of the functionality exception in a slightly different manner. Nevertheless, it is safe to say that in most cases the technicality or functionality exception under trademark law is somewhat broader under that present under design patents or registered designs regime.

For example, in the United States the Samsung and Apple litigation over Apple's GUI features made it clear that they were considered to be functional and therefore non-protectable under trademark law. However, they were not considered to be "functional" under United States' design patent law and therefore Samsung's appropriation of such features resulted in liability under United States' design patent regime. This illustrates the benefit in having a broad portfolio of intellectual property rights protecting one's trade dress: even where the defendant managed to successfully challenge protection under one regime, the contested use was covered by another intellectual property right, which ultimately led to infringement.

Even though GUIs or their features are not designed to achieve a technical result, they could nevertheless fall foul of what could be described as an offshoot of the functionality principle. This is the case where a feature may not perform a technical function but does enhance the desirability of the product due to its aesthetic appeal. Hence, the rationale here is that trademark law is not intended to reward aesthetic creations that render a product more commercially desirable due to its visual appeal; otherwise, trademark law could impede competition by preventing competitors from adopting aesthetic features that could enhance the commercial desirability of their product in a non-origin indicating manner. Where that is the case, registered designs or design patents would be the most suitable vehicles for protection.

Marks that meet the aforementioned criteria and are potentially registerable may still be refused registration where they conflict with earlier marks or rights. This could be the case where the subject matter is identical or like subject matter protected under earlier rights such as trademarks, copyright or design rights. This is something to be examined by a trademark attorney prior to filing a trademark application. Where a potential conflict is identified,

alternative GUIs designs should be explored to avoid potential registration challenges.

Once a trademark for a GUI or its features are registered, a trademark proprietor must ensure there is continuous use of the trademark. Non-use for a period of a few years (e.g., five years in the case of a Community Trademark) may lead to revocation and loss of registration. Furthermore, a trademark proprietor has an ongoing obligation to supervise and police not only its own use but also the use by authorized licensees and by unaffiliated third parties as they relate to identical or confusingly similar marks. Failure to do so may also lead to loss of the trademark. A proprietor must be vigilant in relation to the use made of its mark to avoid the loss of trademark rights.

Unfair competition

Where trademark registration was not obtained, redress may be possible under unfair competition laws against a party who appropriated a mobile app GUI or its features.

Most countries have laws against unfair competition. In most cases, they can be grouped into two categories: common law and civil law models. Countries that follow the common law model do not have unfair competition laws in a broad sense. Under the common law model of unfair competition, which protects unregistered signs, the plaintiff is usually required to establish that the defendant's actions led to consumer deception. These systems may also be referred to as unfair competition by misrepresentation rather than merely unfair competition.

Laws against unfair competition in civil law systems are usually broader in application and are intended to shield competition against misappropriation of reputation and trade values, distortion, misrepresentation and unfair practices in the interests of the competitors, consumers and other operators. Often liability could be established even without any evidence of consumer confusion.

Regardless of the availability of potential redress under the laws of unfair competition, the nature of these systems renders it less predictable in terms of the scope within which one may act without fearing appropriation by competitors. It is always prudent and desirable to register a GUI or its features as a trademark, where possible. Registration provides a presumption of validity and serves as a notice to the world of proprietor's claimed legal monopoly.

Patents and GUI

It is the GUI that plays a pivotal role in creating and enhancing a user's experience. Design patents in the United States or registered designs in the EU may be available to GUIs that are not solely dictated by functionality. Both in the case of trademarks and copyright laws, features that may be defined as solely functional may not be eligible for protection. However, where the relevant GUI or aspects are functional and are designed to achieve a technical result, a mobile app developer or publisher may choose patent law.

Under current United States patent law regarding software-related inventions, obtaining a utility patent for a GUI design may prove difficult. Under the current legal position, a court is likely to enquire whether the GUI improves the functioning of the computer or improves an existing technological process. If the answer is no, as may be the case when the GUI for a mobile app is involved, the invention would require further elements to render it eligible for patent law. It is likely to prove difficult to establish that extra elements are present where the patent application relates to features of a mobile app's GUIs. A U.S. patent specialist should be consulted on the matter.

European patents granted by the EPO are subject to several exclusions, which include the presentation of information and programs for computers, both being relevant to GUI patent protection. To avoid the exclusions, one must show that the GUI features possess a technical character. Although numerous GUI patents have been granted by the EPO, it is currently questionable how many of these patents would be granted at present.

While in the past, the EPO was prepared to accept GUI aspects that lower the cognitive burden on the user as potentially possessing a technical character, it appears that this lax approach has been abandoned and that lowering of the cognitive burden as a result of choices as to what or how to present information is no longer sufficient to constitute technical character. In addition, it appears that the color, size and shape of items on the screen do not usually amount to a technical aspect of a GUI.

In conclusion, in many cases GUI features are not likely to prove eligible for patent protection under most utility patent systems around the world. This makes it even more crucial for mobile app developers and publishers to have at their disposal a tapestry of intellectual property rights, such as copyright, trademarks and design patents/ registered designs, each of which potentially protecting a different aspect of a GUI.

Summary of IP protection for GUIs

One of the key components to mobile apps are the graphic user interfaces with which the user interacts. An app's ease of use is crucial to its success as consumers have little interest in code or text commands.

The GUI can be broken down into three parts:

1. overall interface outlay;
2. individual components comprising the desktop;
3. animated features that are a result of a user interacting with either numbers 1 or 2.

Because GUIs are critical to the success of an application, it is essential to have them legally protected. The following is a guide to GUI-relevant IP protection and their application processes.

Copyright

Copyright protects software code including the portion of code that pertains to GUIs. But this type of IP protection may not be enough. It is possible to have different coding which results in the same, or similar, GUIs. Therefore, a GUI's copyright protection may need to be assessed based on what is on screen versus its coding. There is no clear answer to whether copyright protection extends to the appearance of the desktop, as it depends on the GUI feature and the overall circumstances.

Summary of copyright and GUIs	
Overall interface	For the overall appearance of a desktop to have copyright protection, it must be original and non-functional. Though many elements of a desktop may be commonplace, it is possible to have aspects which are original. Keep in mind that even where all relevant aspects are considered as commonplace, their selection and arrangement may warrant copyright protection. In addition, it must be established that the interface is not dictated by functionality and hence eligible for copyright protection.
Individual components	For an icon to warrant protection, it should contain an embellishment. For example, using an arrow to represent a pointer will likely not provide protection. Menus may also be protected if they are original and non-functional. If there is originality in the selection, arrangement and organization of the menu commands it will be easier to argue that there is copyright protection.
Animated effects	These effects represent some form of idea and consequently require a level of arbitrary embellishment to overcome the originality and non-functional requirement necessary for copyright protection.

Designs

In certain jurisdictions, a GUI's aesthetic or ornamental elements may be protected by design law. For example, in the U.S. and Japan they are protected via a design patent while in the EU they are protected via an RCD. Design rights need to be registered and consequently have an associated cost. In the EU the cost is moderate as registration is not conditional upon substantive examination. Although the EU also has a parallel unregistered design system, its protection term is only three years, which may make it less suitable for GUIs, while its breadth of protection is narrower (it protects only against copying)

GUI design protection in the United States and EU	
United States	In the United States, to obtain a design patent the design must be new, non-obvious, and ornamental rather than functional. For a design to be non-obvious the duration between the current design and those previous is assessed. What may be a challenge for GUIs is that they are designed for user interaction and functional designs are not eligible for design protection. In the United States, however, this functionality challenge is interpreted narrowly and only "purely" functional designs are excluded. A design patent is infringed upon when a user cannot differentiate between two designs which could lead to the user to purchase one product thinking it was the other.
EU	An RCD system in the EU protects designs that are novel and have "individual character." In the EU there is no substantive examination prior to registration and protection takes effect once registration is completed. If there an allegation of infringement after registration, an alleged infringer may rely upon two lines of defense. First, the defendant's design must produce a different overall impression. Second, one could try to invalidate the design. For a GUI, this invalidation would be argued on the grounds of technical function exclusion. The functionality exclusion for RCD essentially means that a court should establish whether there was any factor other than technical function, such as aesthetic appeal, which led to the choices made by the designer. Where that is not the case, the design at issue is likely to be found to be "functional" and thus excluded from protection.

Trademarks and unfair competition

Trademarks protect signs that act as an indication of origin. For a GUI to be considered as a source indicator it is necessary to demonstrate, among others, that it is distinctive and its character must be capable of being perceived by the relevant public as an indication of origin. This threshold is fairly high as it means that a person will be able to recognize the source of origin of a product by looking at the GUI and no other branding related signs.

Trademark law

The following are issues that may arise when examining and establishing trademark protection for GUIs.

Summary of copyright and GUIs	
Distinctive character	It is necessary to show that subject matter of a trademark application has some distinctive character and is not merely descriptive. The developer needs to show that the GUI has a degree of distinctiveness capable of serving as a designation of origin. Though an interface may be unique, it is difficult to foresee a consumer treating the interface as an indication of source of origin and therefore it would probably be necessary that a distinctive character be acquired through the use of the mark in the marketplace.
Functionality	A GUI may be required to overcome a functionality exception when seeking trademark protection. If the GUI is considered essential for function or necessary to achieve a technical result, it may not be eligible for trademark protection. The definition of what amounts to technical or functional varies among jurisdictions.
Priority of registration	Regardless of whether a GUI trademark application meets the aforementioned criteria, there remains a chance that registration is refused due to a conflict with an earlier filing. In this case, it is advisable to alter the GUI's design to prevent future litigation.
Maintenance of registered mark	If a GUI trademark is registered then the owner of the mark needs to ensure that it continues to be used, as non-use of a trademark could lead to revocation or loss of registration. The owner must also pay attention to the use of the mark by authorized licensees or associated third parties and ensure that the mark is used "as registered." Renewal fees must be paid periodically, or the mark's registration may lapse.

Unfair competition

Under unfair competition law it may be possible to seek a remedy against a party who appropriated the GUI of a mobile app. This requires no registration. In a common law jurisdiction, the plaintiff would have to establish that the defendant's actions amounted to misrepresentation and led to consumer deception. In civil law jurisdictions, unfair competition laws are often broader and establishing liability may be possible without evidence of consumer confusion. However, unfair competition laws tend to be less predictable and it is of greater benefit to register a trademark where possible.

Patents

Where the relevant aspects of a GUI are functional and designed to achieve a technical purpose, it may be possible for a developer to seek protection via patent law.

Currently, in the United States it may be difficult to obtain patent protection for a GUI. To obtain a patent, the applicant would have to show that the patent improves the functioning of the computer itself or improves an existing technological process. Typically, this is not the case. An applicant would need to show that the GUI has extra elements to be awarded patent protection. It is prudent to seek legal advice on this matter, but for the most part it is challenging to establish the existence of these additional elements.

In the EU, the "presentation of information" and programs for computers are both excluded from being the subject of patent protection. In order to overcome these exclusions, the features of a GUI must possess a technical character. Although in the past the EPO may have granted patents for features of a GUI that lowered the user's cognitive burden (e.g., by being intuitive), it appears that this is no longer sufficient for establishing technical character.

Notes

1 Council Regulation (EC) No 6/2002 of 12 December 2001 on Community designs, Article 8(1).

Chapter 4

Functionality

Introduction

The term *functio* in Latin means "to perform." Mobile app functionality covers both the functionalities that a given mobile app delivers and the form in which it is delivered. For example, among Snapchat's functionalities, one can send a photo or video (available for a few seconds) or add filters or lenses to photos. Could functionalities be protected on this generic level against imitation, assuming it was Snapchat that came up with them in the first place? (In fact, Snapchat did not; both filters and lens functions were already present on the Instagram app.) If not, at what level of specificity, if at all, might IP protection be available to functionalities?

For mobile apps, regardless of their functionalities, the simpler the delivery, the more likely they will be successful. The significance of well-sorted product functionality is illustrated by the failure of the Betamax player, the revolutionary video player/recorder Sony introduced in the 1970s. Although both its image and build quality were clearly superior to its competitor, JVC's VHS player, it nevertheless lost the race to the VHS format. This can be attributed to two main factors. First, the early Betamax tapes played for only one hour, while most film length was an hour and a half. Second, Betamax players had a top-loading feature which was a problem to users who wished to fit the player into a tight space. The Betamax failed because it did not support the media and user-base it was targeting. These factors remain true when designing mobile apps.

When something is done successfully, it is likely that competitors will seek to emulate it. The following concerns freedom to engage in such imitation from an IP law perspective.

When a mobile app is at initial stages, its functions or functionality need to be determined and agreed upon. After determining the app's functionality and writing it into its specifications, the design stage takes place, ensuring as much as possible that the mobile

app contains all desired functionalities. Quite often, after the design stage is complete, functionality testing verifies the app performs and functions in accordance with its design specifications.

While independently some functionality aspects such as GPS, camera or language support may be too generic for any form of protection, combining them with other aspects in a particular way may result in IP protection.

Copyright law and functionality

The idea of protecting something as formless as functionality under copyright law may appear odd at first. After all, copyright is mainly intended to protect literary, visual or aural works, which are expressed with a high degree of specificity, such as texts, photos, paintings or musical compositions. For software-related works, copyright may also protect the actual code or expressive user interfaces.

The idea/expression dichotomy

Under the idea/expression dichotomy, copyright protects only expression of ideas, not ideas themselves. This division between protectable and non-protectable subject matter is widely acknowledged. What is meant by ideas and, in particular, what do ideas mean in the context of functionalities?

Continuing the Snapchat example, let us assume that upon releasing Snapchat, a mobile app developer recognizes its appeal and wishes to offer their own version. What level of imitation may be acceptable to avoid copyright infringement? Can one offer a mobile app having one of the core functionalities of Snapchat: the ability to send photos and videos only available for a few seconds? This question is posed in relation to the function itself regardless of the means through which it is achieved. The technical outcome of a self-destructing photo or video could be realized in a few ways. The technical process itself may be subject to patent protection.

As mentioned, many functions may be realized through a variety of technical courses. Can copyright law be used to stop a competitor from imitating what a mobile app does? Can for, example, copyright law grant the first entrant into the market a monopoly over the actual concept of self-destructing photos or videos once sent? The answer is no as the concepts of self-destructing photos or videos, filters or lenses falls into the idea side of the idea/expression divide and are therefore not protectable under copyright law. But as will be explored, the more details which surround general concepts one chooses to copy, the more likely it is that copyright infringement will be established. For example, the concept of using filters for altering one's image can be considered an idea and therefore not protectable under copyright law. What about a filter that allows photo editing so that the subject of the photo appears to be puking a rainbow? Or the addition of a particular type of bunny ears? If both are still too generic to be considered as a protectable expression, how about their combination? Assuming the concept behind each of the filters individually cannot be protected, it means that the whole or a significant part of the filters on Snapchat, if copied, may amount to a protectable compilation.

The merger doctrine

It is possible for an expression of an idea rather than the idea itself to be considered as non-protectable under the idea/expression dichotomy due to the operation of the merger principle. Although the general rule is that once an idea is expressed, it is eligible for copyright protection under the idea/expression dichotomy, such expression may nevertheless be considered non-protectable where it is found that it has merged with the idea.

Copyright law considers an idea and its expression to have merged where a non-protectable idea is expressed in a very limited number of ways, so the expression becomes inseparable from the idea. Where there is one way to express an idea, protecting that expression means that no other person would be able to utilize that idea and hence would effectively amount to granting copyright protection to the idea itself.

The implications of the merger rule on functionality are significant. Although generic functionalities previously described may not be eligible for copyright protection at a concept level, some of them may be realized technically in a software's structure, sequence and organization. While aspects of software are expressive and may be eligible for copyright protection, where there are no other ways to bring about the requisite outcome, such expressive elements merge with the unprotectable concept and are considered themselves non-protectable. The operation of the merger rule is designed to ensure that one may not gain a legal monopoly over a non-protectable concept.

Judicial treatment

In the United States, the idea/expression dichotomy was first mentioned in a Supreme Court case in the late nineteen century. In that case, the court was asked to decide whether one could have copyright protection over an original bookkeeping system. The defendant read the plaintiff's book to understand the bookkeeping system and wrote, using their own words, to describe the system. The question was whether the system itself was eligible for copyright protection. The Supreme Court decided that a bookkeeping system is not a proper subject matter for copyright protection. If at all, such a system may be protected under patent law, provided it satisfies the strict set of requirements therein. In this context the Supreme Court stated in the aforementioned case, "[t]o give the author of the book an exclusive property in the art described therein, when no examination of its novelty has ever been officially made, would be a surprise and a fraud upon the public."

The conceptual similarity between a book describing a booking system and the system itself and a computer program and the logic and functionality embodied in it was indeed highlighted in later cases where the rationale was applied to software functionalities.

Functionality and patent law

Is it possible to obtain patent protection for the function of mobile apps? Securing protection for a feature of an app can be difficult. However, a specific way of implementing a feature may be patentable. In the United States, the algorithm for executing the feature must be described. Protection is limited to what is explained. At the EPO, the relevant algorithm does not always need to be disclosed. It may be possible to define an invention by describing the relevant steps in enough detail so that a skilled person could implement the feature, resulting in the possibility for protection.

Most jurisdictions that allow software patents take a similar approach. As a result, patents related to a functionality are limited in scope to aspects which are fully described in the patent applications. It may be possible that a competitor could work around such protection to develop apps with similar functionality without infringing on a patent owner's rights. While these assets have strategic value, care should be taken to understand the scope of the associated rights. Such analysis is prudent both for the owner of a mobile app and the developer who wishes to include existing functionality in their own app.

Functionality and laws against unfair competition

Registered trademark law cannot protect the pure functionality of mobile apps, as trademark registration requires more description than available for mobile apps regardless of details provided.

Unfair competition laws are, in principle, capable of providing protection against an imitator that emulates functional or behavioral elements of a mobile app to the extent that the imitating application could lead to confusion as to its origin or affiliation. Take a hypothetical example of a functional feature such as a calendar mobile app, where a user can "drag" a date listed in an email directly into the calendar app to create an event. If a developer decided to imitate this feature, say to create a feature where a user could "drag" an activity listed in an email to an "organizer" app, it may

lead to violation of unfair competition laws where the said functional feature is characterized by a unique action. The extent to which such liability may be established depends on parameters such as the actions of the emulator, public perception and the jurisdiction concerned. These parameters will be further discussed. In addition, the emulated feature must not fall within the functionality exception present under many unfair competition/trade dress laws.

Regarding the actions of the emulator and the scope of that which is being emulated are of key significance. In the hypothetical example given above, it is not mentioned what exactly the emulator sought to imitate in relation to the 'drag' function. It is pretty clear that the 'idea' or concept' of being able to drag an item from an email to another application is not likely to prove protectable under trade dress laws. To start with, the concept itself is likely to fall within the functionality exception that most trade dress laws contain (if at all, such concept might have been protected when it was first conceived under patent laws). If, however, the emulator in our example seeks not only to use the same concept, but also the manner in which it is being put into effect – namely, the exact input required from the user and the consequent visual output produced by the application – trade dress laws might come into play.

As is the case with most trade dress actions, it is necessary to show that the concept and use is distinctive enough to be associated with one source of origin. This would usually require a period of extensive use by the original developer so that now the relevant public "learned" to associate it with one particular source of origin. In most cases the input and output would be reflected in the app's GUI. Refer to Chapter 3 for more detail.

Practical implications and considerations of protecting functionality

Conceptually, protecting functionality may prove difficult. Unless protected under patent law, an abstract functionality may be imitated without attracting liability. This, however, may change where functionality is connected with visual signposts such as static or dynamic graphical features.

Signposts could potentially be protected under copyright, trademark and trade dress laws, as well as laws protecting designs. Therefore, where possible, tying a working environment to protectable signposts could help a developer obtain a certain level of protection for the functionalities of their mobile app. The protectability of such signposts should be considered both individually and in combination. Regardless of the protectability of an individual sign, a particular way in which signs are selected and arranged in a mobile app may be eligible for protection independently.

Having mobile app users accustomed to such signposts so that they become an integral part of the app's 'working' environment may seriously compromise the potential success of imitative competitors. Since such signposts could not be copied without giving rise to liability under IP laws, any attempts by an emulator to create substitutable imitation are less likely to prove successful as users of the original version are less likely to migrate to a version that does not offer the visual signposts to which they have got accustomed.

Summary of app functionality and legal protections

Mobile app functionality refers to the app's purpose and the way it responds to a user's input. Though some mobile app functions such as the GPS, camera or language support may be too generic to warrant protection, there may be some functional aspects that may be eligible for protection under IP laws.

We have seen that copyright may protect software code and expressive user interfaces; however, these components of an app are definitive. With respect to functionality, examining the relevant areas of copyright law may help establish whether and to what extent mobile app functionality may be eligible for copyright law protection.

Functionality and copyright law		
Concept	Principle	Relevance to mobile applications
Idea/expression dichotomy	Copyright protects expressions of ideas but not ideas themselves	The more details surrounding a concept, the increased likelihood for copyright protection A basic element of an app, e.g., a filter, may not be protectable; but if unique filters are created and put together, there may be copyright
Merger doctrine	An expression of an idea may not be protectable if it has merged with the idea The idea becomes inseparable from the expression	Aspects of software may be expressive and suitable for copyright protection; but if there is no alternative way to express this idea, then the expression will not be protected This prevents granting of a monopoly in instances where an idea may only have a single or limited form of expression

Generally, a patent can only protect the mobile app's function if it is disclosed in the application. Protecting functionality via patent law was examined by reviewing two specific jurisdictions selected because of their developed IP legal systems, as well as their mobile app industry and market.

Unfair competition law may provide protection against an imitator who tries to reproduce a mobile app's functional components, particularly if such reproduction results in confusion as to its origin or affiliation.

To determine whether an imitator's actions fall foul of unfair competition law two considerations are examined: the actions of the imitator, and public perception. The emulated features must not fall into any functionality exception under unfair competition law in certain jurisdictions.

Considerations for functionality protection under unfair competition law	
Actions of the imitator	In many instances, a concept will fall within a functionality exception In order for developers to safeguard against falling into the functionality exception, they would have to show that the manner chosen to carry out a certain function was arbitrary rather than dictated by the requisite outcome
Public perception	The public must perceive the concept/function, and the means by which it is put into effect, associated with the original developer The result of this perception is the confusion caused by the imitator's app as to origin or affiliation

Chapter 5

Non-IP Legal Considerations

Introduction

Mobile apps have become a modern necessity and, because of their far-reaching use, their breath and scope as widened significantly. Though IP may be of most relevance, there are other legal considerations for mobile app developers and purchasers to consider. Most of the legalities concern use by consumers. This involves contract law and applies to end-user licenses. These agreements include issues such as data protection, privacy and consumer protection, to name but a few. Digital rights management is an additional legal avenue available to protect content. The scope of this chapter is centred on the legal position in the United States and EU; however, there will be specific domestic legal considerations depending on where the app is developed and marketed.

End-user license agreements

End-user License Agreements (EULA) are contracts which set out the terms on which a consumer may use a mobile application and as a way for software designers to protect their economic investment. An EULA defines the relationship between the developer and the user, outlining each party's rights. These contracts detail several issues but explicitly state that a EULA is a licensing agreement stipulating the terms of the license, that is, whether it is non-exclusive, revocable or not subject to transfer.

The EULA will typically set forth any restrictions that the owner wants to place on the app's use and may also bind the user to other agreements such as additional terms and conditions and privacy policy agreements. As these contracts are fundamentally concerned with the IP, a term will cover infringement and the right to terminate the license under specific circumstances. Another main concern of such contracts are clauses limiting liability. Two of the most important limitation clauses include the limitation of warranties and

of liability. The former advises the user that the app is licensed on an "as is" basis and prevents the licensor from being obliged to modify the software to suit the user's needs. The latter limits liability to any damage to hardware caused by downloading and/or use.

In addition to governing the specific relationship between the developer and user, an EULA may address other legal considerations necessary to comply with data protection, privacy, consumer protection and advertising legislation.

Data protection

In 1995 the EU adopted the Data Protection Directive which set out the framework for data protection.[1] This directive sought to protect the fundamental right of the personal data and to ensure the free flow of personal data in the internal market.[2] The directive was implemented by member states to regulate the process by which data is collected, used, stored, disclosed and destroyed. There were also additional data directives applicable to those operating online services.[3] Understanding the relevant provisions of the Data Protection Directive and domestic legislation is essential for mobile app developers operating within the EU. Personal data,[4] among other things, are provided during a mobile app's use and requires the app owner to comply with certain requirements.[5] It is essential to collect a minimum amount of data for the app's tasks and this data must not be stored for longer than required to carry out its specific tasks. Users must be informed as to what will happen to their personal data. Users must also have a way to contact developers to request any personal data collected.

In the United States, there is no single comprehensive law governing the collection and use of personal data and therefore no single reference point for data protection for mobile apps. There is an array of federal and state laws and regulations that govern data in a patchwork fashion and many guidelines which may not be legally enforceable but form part of self-regulation.[6] There are federal privacy laws which regulate the use and collection of personal data, namely the Federal Trade Commission Act (FTC).[7] There are also specific laws which govern data in various sectors such as financial

services and those handling medical records.[8] The FTC prohibits unfair or deceptive acts or practices involving practices that fail to safeguard consumers' personal information and can challenge a company that fails to protect a consumer's personal data.[9] The FTC is cognizant that the different apps have different protection needs, however, they have set some criteria to consider when setting out data protection policies. These include appointing an individual responsible for security who oversees how data is collected and retained and considers obscuring data collected through encryption measures, among others.[10]

Whether active in the United States, EU or many other legally developed jurisdictions, it is imperative that mobile app publishers ensure their business model is compliant with local data protection laws. Failure to do so may result in significant monetary and criminal sanctions.

Privacy

A mobile app developer should consider key areas of privacy law. In many instances, privacy operates in tandem with data protection. In 2013, the FTC created guidelines for marketing mobile apps which included guidance on certain privacy aspects. It provided advice rooted in building privacy considerations from the onset. These included limiting the amount of information collected, providing secure storage, and safe disposal of information no longer needed.[11] In addition, it advised that mobile apps should have privacy setting choices available for users to ensure privacy promises are honored and additional efforts are made to protect children's privacy. There are also state laws to follow depending on where the app is used. For example in 2013, California published privacy recommendations for the mobile "ecosystem."[12] Though very similar issues were raised by the FTC, the recommendations for developers included creating a clear, accurate and accessible privacy policy and enhanced measures such as special notices or short privacy statements.

For mobile apps in the EU, privacy considerations are largely intertwined with data protection as previously addressed. In addition to the Data Protection Directive, there are also relevant

aspects within the e-Privacy Directive.[13] For instance, Article 5(3) of this directive outlines aspects of storing and gaining access to information which is critical to the functioning of most apps.[14]

Consumer protection

Beyond data protection and privacy issues, there are consumer protection concerns to be discussed when developing a mobile app.

Directive 2011/83/EU on consumer rights (Consumer Directive) applies where a person purchases a mobile app. This purchase is considered a "distance contract" between the developer and consumer and sets out rules regarding required information and cancellation.[15] An app publisher must provide the following information in a clear and comprehensible manner to the consumer before the contract is signed:

- app's main characteristics;
- developer's identity and contact details;
- total price and any additional charges;
- payment arrangements;
- where a right of withdrawal exists, conditions for exercising that right and the model cancellation form;
- where a right of withdrawal does not exist, information on circumstances under which the consumer loses right of withdrawal;
- contract duration and, in case of an indeterminate duration, conditions for terminating;
- where applicable, the minimum duration of contractual obligations;
- functionality of digital content, including applicable technical protection measures;
- relevant interoperability of digital content with hardware and software that the trader must be aware of; and
- details of any relevant codes of conduct.

Consumer protection in the United States is addressed at both the federal and state level. At the federal level, the FTC aims to prohibit "unfair and deceptive acts or practices in or affecting commerce".[16] The FTC considers deception has occurred when

there is an omission within presented material, or omission of a practice that may mislead a consumer. Unfair practices are those which cause, or are likely to cause, avoidable and substantial injury to consumers without any offsetting benefits.[17] At the state level, consumer protection is a matter for the State Attorney Generals who can initiate consumer protection litigation and set consumer protection policy.

It is essential to ensure that mobile apps within a given jurisdiction are in compliance with consumer protection laws.

Advertising

Mobile apps foster the same legal advertising as for websites, email and other networked communication.[18] It is common that an app developer is provided with code from an advertising network or third party to facilitate advertising or app analytics. It is possible that the developers are not aware of the function of this code which allows advertisers to collect data without developers ensuring end users are aware of such practices.[19] It is imperative that developers and advertisers have an open dialogue so that consumers are provided with accurate information.[20]

In the United States, advertising must comply with the Telephone Consumer Protection Act, as well as Federal Communications Commission rules. The FTC guidelines regarding mobile app marketing are based on the principles governing the U.S. Privacy Bill of Rights which include transparency, control, respect for context.[21] The FTC also administers The Truth in Advertising provisions which emphasize that online advertisers should provide clear and conspicuous disclosures of the information that consumers need to make informed online purchasing decisions.[22] In addition to legislation, there are also certain industry standards that set parameters for mobile app advertising. The Digital Advertising Alliance has published guidelines entitled *Application of Self-Regulatory Principles to the Mobile Environment*.[23] Furthermore, the National Advertising Initiative released a mobile application code of conduct.[24]

In the EU, third parties collecting information from an app in order to supply additional services of their own, personalized advertising for example, become data controllers and are therefore subject to data protection laws. If online behavioral advertising is carried out, there are e-privacy consent requirements to which one must adhere. The e-Privacy Directive which came into force in 2002[25] was replaced by the e-Privacy Regulation which came into force in 2018. For business to consumer communication, this regulation seeks to require consent of the consumer for direct e-marketing purposes.[26]

Digital rights management and technical protection measures

There is another way a mobile app developer can safeguard copyright so as to protect their investment. Digital rights management (DRM) is a broad term used to refer to several technologies employed to impose predetermined limitations on the use and transfer of copyright protected digital content.[27]

There are two levels to the DRM approach. The first aims to control copying while the second aims to control viewing, printing, modifying etc., of digital content. Technical protection measures (TPMs) are mechanisms that a developer can adopt to control and/or restrict access to protected works. In the EU, TPMs are defined under Article 6(3) Directive 2001/29/EC which is sometimes referred to as the "Infosoc Directive." A TPM may be any technology, whether software or hardware, which limits access to materials protected by copyright without the consent of the right holder. One may be liable if circumventing these measures without consent regardless of any liability arising from copyright infringement (e.g., circumventing an access control mechanism for unauthorized copying).

In the United States, provisions regarding anti-circumvention of TPMs are found in the Digital Millennium Copyright Act (DMCA). The DMCA aims to ban acts that circumvent TPMs including any device, service and technology whose primary function is to circumvent TPMs.[28] These measures add a layer of protection to the legal protection already granted by copyright. TPMs may take a variety of forms, a few of which are as follows:

- use of a dongle which is a piece of hardware containing an electronic serial number that must be plugged into the computer to run the software;
- use of a registration key which is series of letters and numbers that is requested when installing or running the program. The software refuses to run if the registration key is not typed in correctly and multiple use applications (e.g., multiplayer games) will refuse to run if the same registration key is typed in more than once;
- use of Internet-product activation which requires the user to connect to the Internet and type in a serial number which notifies the software manufacturer and prevents other users from installing the software if they attempt to se the same serial number;
- use of encryption, such as the Content Scrambling System (CSS), to make copying more difficult. In these schemes, the work is encrypted using a key included in the firmware of authorized players, allowing only legitimate uses of the work (usually restricted forms of playback, but no modification or conversion); and
- use of digital watermarks which is a digital signal or pattern inserted into a digital image. A given watermark may be unique to each copy (e.g., to identify the intended recipient), or be common to multiple copies (e.g., to identify the document source).[29]

There have been various concerns surrounding TPMs with respect to their scope and effectiveness. It has been viewed that circumventing TPMs is unlawful and may protect content that does not warrant copyright protection.

App developer agreements

When developing a mobile app, it is important to bear in mind the legal relationship between the app developer and the platforms on which it may run. The agreements governing these relationships tend to include boilerplate clauses, meaning there is little to no scope for developers to negotiate the terms. Many of these clauses are common to all the various distribution platforms. The focus here is

on common clauses that feature in the developer agreements of the three major platform providers, namely Apple, Google and Microsoft. However, it should be noted that Apple's agreement prohibits a developer from making any public statements about the terms of the agreement (although the agreement itself does not define the "Apple Confidential Information").

The application of various branches of law, other than IP law, may be crucial to the design, functioning and marketing of mobile apps. However, it is not the intent here to engage in a detailed discussion of these legal considerations. The following summary is meant to introduce some of the issues governed by license agreements. Should an app developer seek to enter into an agreement with the abovementioned three commercial entities or any other platform provider, they would be well advised to seek independent legal guidance in the relevant jurisdiction.

Licenses

When using these platforms, developers may have to grant a license to platform providers. Google's agreement provides that developers should grant a non-exclusive royalty-free license to host, link, copy, translate, publicly perform, publicly display, test, distribute and otherwise use the app in question. Google also specifies that by using its platform the developer grants a non-exclusive, worldwide, perpetual license to users. Developers may include an additional end-user license agreement that regulates the use of their app. Google clearly states that apart from the stipulations of the agreement, the developer retains all rights to the app and that both parties retain the rights they would have individually held regardless of the agreement. These include rights granted via copyright.

Microsoft's agreement also specifies that a developer does not transfer app ownership to the company but does grant the right to host, install, use, reproduce, publicly perform and display via any digital transmission technology, format, or make available to customers.

Apple's agreement does not specifically dwell on the legal relationship between the content provided by a developer and the

Apple platform. It does however, stipulate that nothing restricts Apple's right to develop, acquire, license, market, promote or distribute products which perform the same or similar functions that may compete with what has been developed, produced or marketed by a developer. Apple also affirms that in the absence of a separate agreement, it is free to use any information, suggestion or recommendation provided by the developer for any purpose. However, it expressly states that this is subject to any applicable copyright or patents. Hence, such a license does not deprive the developer of the copyright or patent (to the extent that there is one) for the app.[30]

Amendments

Should these agreements require amendment, Apple reserves the right to modify its agreement with developers at its discretion and this may also pertain to rules and policies. It then becomes the responsibility of the developer to review and become familiar with any changes. Apple will deem continued use of its site as an agreement to the additional or amended terms.

Google also states that it may make changes to its agreement from time to time. Should this happen, a copy of the new agreement and a notification of the changes introduced will be posted on their site and deemed accepted by the developer seven days after posting the notice and if the developer continues to use the platform.

Microsoft states that it may modify its agreement at any time at its sole discretion. It does not state how developers may be notified of such changes but does confirm that the last modification will appear at the top of the agreement. Similarly, it does not provide information as to how developers should accept or be deemed to have accepted such amendments.

Termination

The three platforms have slight variations on the grounds and means for terminating developer agreements.

Microsoft allows termination by either party at any time, with or without reason, after providing at least 60 days' written notice. If a material breach occurs that cannot be remedied, the agreement will lapse 30 days after the party alleged to have committed the breach receives written notice.

Apple can terminate or suspend a registered Apple developer at any time at its sole discretion. However, a developer may also terminate the agreement for any reason by notifying Apple in writing of its intention to do so.

Google states that it will terminate an agreement if: (1) there has been a breach of the provisions of the agreement by the developer; (2) it is required to do so by law; and (3) it decides to no longer provide Google Play, which is its app store. A developer can terminate an agreement with Google after giving 30 days' written notice.

It should be noted that all three platforms can remotely disable apps, even after they have been installed by users. The license agreements and termination clauses give them these legal rights.

Liability limitation

All the platforms contain a clause limiting their liability for any damage a developer may incur from the platform's use. Such a clause is like stating that developers use the platform at their own risk.

Apple declares that it is not liable for any damage resulting from a delay in delivery, loss of profits, data, business or goodwill, business interruption or any other commercial damages or losses relating to its agreement with developers. It also limits the amount recoverable for damages under its agreement to USD 50 except where required by law to act otherwise such as in the case of personal injury.

Microsoft limits the amount of recoverable damages to USD 1. It also limits the ability of the developer to recover any losses or damages except where prohibited by the laws in the developer's state or country.

Google states that it is not liable for any damages and expressly uses the example of not being liable for loss of data. It goes one step further to include an indemnification clause which states that the developer is to indemnify Google for any third-party claim and/or other associated costs arising from use of Google Play or where the app infringes any copyright, trademark, trade secret, trade dress, patent or other IP right which defames any person or violates their rights of publicity or privacy.

Warranty disclaimer

In addition to limiting liability, the platform agreements also include a warranty disclaimer. A warranty disclaimer refers to any damage that a developer may incur from the platform not performing the way the developer assumed. This may include: loss of the developer's data caused by the platform; virus-infected content downloaded by the developer from the platform; or unavailability of the platform due to technical problems. Consequently, this means that the developer uses the platform as is and without any warranty. The risk of using the platform rests solely on the developer who should carry out all relevant checks before relying on any platform characteristic or function.

Apple also disclaims all warranties of accuracy, non-infringement, merchantability and fitness for a particular purpose. Apple's sole remedy for a developer who is not happy with its service is to suggest they stop using the service. In addition to the elements outlined by Microsoft and Apple, Google emphasizes that it is not responsible for damage caused to a developer's computer or any loss of data that may occur from platform use or the material downloaded. Domestic laws often restrict the ability of one of the parties to an agreement to disclaim all warranties and prescribe some basic safeguards to survive contractual attempts to exclude all responsibility for the state and operation of products and services. This is especially relevant if the contract parties have unequal bargaining powers. If necessary, the relevant domestic law implications should be discussed with a local legal expert.

Governing laws

All platform agreements attempt to subject the agreement to laws of a specific jurisdiction. For example, Apple and Google state that the laws governing the relationship between their platforms and developers are those of the State of California. Both companies expressly exclude the State's conflict of law provisions. Most countries have rules which assert that the laws in the country where the cause of action arose should govern the matter. By expressly excluding the provisions within a country's conflict of laws, these companies ensure that matters are handled in accordance with the agreement provisions.

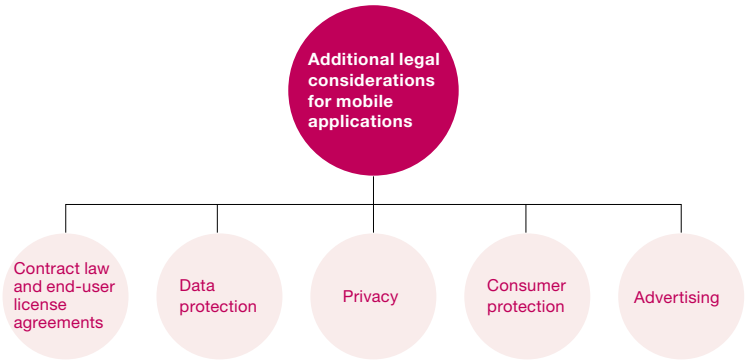
Notwithstanding, Google states in its Google Play agreement that it can seek injunctive relief in any jurisdiction. Apple, although it does not state a sole county, does require that the developer should not object to the conduct of legal proceedings in the U.S. District Court of Northern California, California Superior Court for Santa Clara County, Santa Clara County Municipal Court or any other forum in the county of Santa Clara.

In Microsoft's developer agreement, the governing law is the State of Washington and the State's conflict of laws principles are also excluded. Furthermore, the developer is required to consent to the exclusive jurisdiction and court location in King County, Washington. There is, however, one exception to the exclusive use of Washington State laws: if the developer's primary headquarters are in New Zealand, the agreement is governed by the laws of Singapore.

As in the case of limitation of liability and exclusion of warranty, the ability to use such a contract to impose the laws of one jurisdiction is often limited by domestic laws. This is especially so when parties do not enjoy equal bargaining powers. The enforceability of such a choice of clauses should be examined in detail with respect to the domestic laws of various jurisdictions.

Summary of non-IP legal considerations for app developers

In addition to the many legal implications for mobile apps that arise from IP laws, there are other legal aspects a developer should be mindful of when bringing an app to the market. Some of the areas that should be addressed include contract law with a focus on end-user license agreements, data protection, privacy, consumer protection and advertising. There is also the element of technical protection measures which may provide a developer with additional security against unauthorized copying or use.



Contract law and end-user license agreements

These agreements set out the terms in which a consumer may use the mobile app. These agreements allow a developer to protect their economic investment by providing them with the ability to set restrictions on the exploitation of the app. An agreement can include what amounts to infringement of the IP rights and may also limit the liability of the app developer and publisher.

Data protection

When an application is utilized, it will require the user to provide personal information which must be protected. Different jurisdictions have different legal mechanisms in place to protect consumer data that should be complied.

In the EU, there is a directive that protects the fundamental rights of personal data and ensures the free flow of such data in the internal market. There are specific provisions that govern the way data is collected, used, stored, disclosed and destroyed. When creating and marketing an app, it is essential that a developer collect the minimum amount of data necessary to carry out a specific task. This data must not be stored for longer than necessary and those providing the data must be informed as to what may happen to their data and be able to request any personal data that a developer may hold.

Unlike the EU, the United States does not have a single piece of legislation that deals with data protection. There is a variety of federal and state laws which govern the use of personal data. There are also guidelines in various states which act as a reference for data use. At the Federal level, the U.S. FTC oversees personal data matters and aims to prohibit unfair or deceptive acts or acts involving practices that fail to safeguard a user's personal information.

Privacy

In most instances, privacy and data protection go together. The U.S. FTC published guidelines on marketing mobile apps which addressed privacy issues. It stated that privacy considerations should be addressed at the onset of development. These guidelines include information on limiting, securely storing and destroying information collected. In the EU, privacy and data protection are very closely intertwined. However, in addition to the Data Protection Directive there is also an e-Privacy Directive which provides guidance on issues pertaining to information storage and access.

Consumer protection

Consumer protection rights are available for the purchaser of a mobile application. In the EU, such rights are protected by the Consumer Directive and in the United States, among others, by the FTC.

Information to be provided by an app owner/developer to a user in the EU		
Characteristics of the app	If right to withdraw exists, indicate means of exercising this right	Functionality of digital content, including applicable technical protection measures
Identity of developer and contact details	If the right to withdraw does not exist, indicate this information	Any relevant interoperability content
Price and any additional charges for app	Duration of the contract and conditions for termination	Details of any relevant codes of conduct
Payment arrangements	User's obligation under the contract	

In the FTC issues guidelines aimed at addressing misleading users. It prohibits any unfair and deceptive acts or practices in or affecting commerce.

Advertising

Mobile app advertising carries the same legal concerns as in website, email or other networked communication. In the United States, advertising needs to comply with the Telephone Consumer Protection Act and also Federal Communications Commission rules. The FTC administers the Truth in Advertising which outlines that online advertisers should provide clear disclosures of information that users need in order to make informed purchasing decisions. In the EU, mobile app developers and publishers must be mindful of the Misleading and Comparative Advertising Directive and the Unfair Commercial Practices Directive, in relation to advertisement appearing on their apps.

Digital rights management and technical protection measures

Mobile app developers may be able to protect their content further using digital rights management (DRM). Technical protection measures (TPMs) are mechanisms that are used to protected copyrighted content. In Europe, the United States and many other jurisdictions, subverting TPMs without consent could lead to liability independent of any copyright infringement. TPMs may take a variety of forms which include, among others, use of:

1. dongles
2. registration keys
3. Internet product registrations
4. encryptions
5. digital watermarks.

In contrast, DRMs are seen to be any of several technologies employed not only to protect content, but also to facilitate payment and regulate user behavior.

App developer agreement clauses

Common clauses in app developer agreements			
Clause	Apple	Google	Microsoft
Licenses granted under the agreement	Stipulates that no provision in the agreement between Apple and the developer restricts Apple's right to develop, acquire, license, market, promote or distribute products which perform the same or similar functions that may compete with what has been developed, produced or In the absence of a separate agreement, Apple is free to use any information, suggestion or recommendation provided by the developer for any purpose subject to any applicable copyright or patents	Developers grant a non-exclusive, royalty-free license to host, link, copy, translate, publicly perform, publicly display, test, distribute and use the app Developers grant a non-exclusive, worldwide, perpetual license to platform and app users Developers may include an additional end-user license agreement	Developers do not transfer ownership of app but grant Microsoft rights as an agent or commissionaire These include the right to host, install, use, reproduce, publicly perform and display via any digital transmission technology, format, and make available to customers for the purposes of fulfilling Microsoft's obligations
Amendments to the agreement	Reserves the right to amend the agreement at its discretion	May modify its agreement at any time and at its discretion	May modify its agreement at any time and at its discretion
Termination of the agreement	Apple may terminate or suspend a registered developer at any time at its sole discretion A developer may terminate the agreement for any reason after giving Apple notice of its intention	May terminate an agreement if: (1) there is a breach of the provisions by the developer; (2) required to do so by law; and 3) decides to no longer provide its app store service. A developer may terminate an agreement with Google after giving 30 days' written notice	Either party may terminate the agreement at any time, with or without reason, after giving at least 60 days' written notice

Common clauses in app developer agreements (cont.)			
Clause	Apple	Google	Microsoft
Limitation of liability	Not liable for any damages arising from a delay in delivery, for loss of profit, data, business or goodwill, for business interruption or any other commercial damages It limits the amount recoverable for damages under its agreement to USD 50	States that Google is not liable for any direct, indirect, incidental, special consequential or exemplary damages	Limits or waives the ability of the developer to recover any losses or damages except where prohibited by the laws of the developer's state or country The amount of recoverable damages is capped at USD 1
Disclaimer of warranty	Disclaims all warranties that its site, content or services will be accurate, reliable, timely, secure, error-free or uninterrupted or that any defect will be corrected The platform is provided on an "as is" and "as available" basis Cannot guarantee that any content downloaded will be free of viruses, contamination or destructive features Developers assume total responsibility and all risks Sole remedy for dissatisfaction with the service is to stop using it	Developers use the platform at their own risk and what is provided is "as is" and "as available" Developers are responsible for any damage to their computer system or other device or loss of data that could result from platform usage Disclaims all warranties regarding, but not limited to, conditions of merchantability, fitness for a particular purpose and non-infringement	Developers use the platform "as is," "with all faults "and "as available" The risk of using the platform is assumed by the developer and recourse is left to local laws
Governing laws of the agreement	Governed by the laws of the State of California	Governed by the laws of the State of California	Governed by the laws of the State of Washington unless the developer's primary headquarters are based in New Zealand, in which case the governing law would be in Singapore

Notes

1 Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data (Data Protection Directive), 1995 O.J. (L 281) 31.

2 Ibid.

3 Directive 2002/58/EC; Directive 2006/24/EC amending Directive 2002/58/EC; Directive 2009/136/EC amending Directive 2002/22/EC; Directive 2002/58/EC.

4 Personal Data is defined as "any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person" Data Protection Directive Article 2(a).

5 As potential "controllers or processors", mobile app publishers may be required to comply with the General Data Protection Regulation (GDPR), Regulation (EU) 2016/679).

6 Jolly, I. (2017). *Data Protection in the United States: Overview*. Thomson Reuters Practical Law. Available at https://uk.practicallaw.thomsonreuters.com/6-502-0467?transitionType=Default&contextData=(sc.Default)&firstPage=true&bhcp=1.

7 Federal Trade Commission Act, 15 U.S.C. §§ 41–58.

8 The Financial Services Modernization Act (Gramm–Leach–Bliley Act (GLB)) (15 U.S.C. §§6801–6827); The Health Insurance Portability and Accountability Act (HIPAA) (42 U.S.C. §1301 et seq.); The Electronic Communications Privacy Act (18 U.S.C. §2510) and the Computer Fraud and Abuse Act (18 U.S.C. §1030).

9 FTC, 15 U.S.C. § 41.

10 *App Developers: Start with Security*. Federal Trade Commission. Available at https://www.ftc.gov/tips-advice/business-center/guidance/app-developers-start-security.

11 Federal Trade Commission (2013). *Marketing Your Mobile App: Get it Right from the Start*. Available at https://www.ftc.gov/system/files/documents/plain-language/pdf-0140_marketing-your-mobile-app.pdf.

12 Harris, K.D. (2013). *Privacy on the Go: Recommendations for the Mobile Ecosystem*. California Department of Justice. Available at https://oag.ca.gov/sites/all/files/agweb/pdfs/privacy/privacy_on_the_go.pdf.

13 Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications).

14 Directive on privacy and electronic communications, Article 5(3).

15 Consumer Directive, Art. 20.

16 FTC, 15 U.S.C. § 5(a).

17 Ibid.

18 Practical Law Intellectual Property & Technology (2017). *Mobile App Development: Key Legal Considerations*. Thomas Reuters Practical Law. Available at https://uk.practicallaw.thomsonreuters.com/7-525-8637?transitionType=Default&contextData=(sc.Default)&firstPage=true&bhcp=1.

19 FTC Staff Report (2013). *Mobile Privacy Disclosures: Building Trust Through Transparency*. Available at https://www.ftc.gov/sites/default/files/documents/reports/mobile-privacy-disclosures-building-trust-through-transparency-federal-trade-commission-staff-report/130201mobileprivacyreport.pdf.

20 Ibid.

21 Federal Trade Commission (2013). *Marketing Your Mobile App: Get it Right from the Start*. Available at https://www.ftc.gov/system/files/documents/plain-language/pdf-0140_marketing-your-mobile-app.pdf.

22 Federal Trade Commission. *FTC Seeks Input for Revising Its Guidance to Business about Disclosures in Online Advertising* (May 26, 2011). Available at https://www.ftc.gov/news-events/press-releases/2011/05/ftc-seeks-input-revising-its-guidance-businesses-about.

23 The Guidance is available at https://digitaladvertisingalliance.org/sites/aboutads/files/DAA_files/DAA_Mobile_Guidance.pdf.

24 The NAI Code of Conduct can be found at http://www.networkadvertising.org/mobile/NAI_Mobile_Application_Code.pdf.

25 Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications).

26 Boardman, R. and Sampedro, G. (2017). *European Commission published the proposed text for the new e-Privacy Regulation*. Bird & Bird. Available at https://www.twobirds.com/en/news/articles/2017/global/eprivacy-regulation-alert.

27 Office of the Privacy Commissioner of Canada (2006). Digital Rights Management and Technical Protection Measures. Available at https://www.priv.gc.ca/en/privacy-topics/technology-and-privacy/02_05_d_32.

28 Hinze, G. *Technological Protection Measures in the draft FTAA*. Electronic Frontier Foundation. https://www.eff.org/pages/technological-protection-measures-draft-ftaa.

29 Office of the Privacy Commissioner of Canada (2006). Digital Rights Management and Technical Protection Measures. Available at https://www.priv.gc.ca/en/privacy-topics/technology-and-privacy/02_05_d_32.

30 Some licenses may impact the ability of a licensor to enforce its patents, either directly or indirectly. Before entering an app developer agreement, adequate legal advice should be sought.

Chapter 6

Global Challenges

As regards traditional piracy, it is really detection and enforcement that matters. Hence, in the case of unauthorized reproduction for sale of a mobile app, there is little to consider in terms of whether or not such practice triggers copyright infringement. Quite often, this will be accompanied by trade mark infringement, as an authorized copy will also bear the trade name or logo of the copied mobile app. A developer may also consider adding superfluous and extraneous portions of code, the existence of which in an allegedly infringing article could only be explained by copying. This would demonstrate copyright infringement.

On July 2017, Forbes reported that mobile app developers are losing USD 3 to 4 billion annually due to pirated apps. Apparently, up to 14 billion pirated apps are installed globally each year, that is, stolen from their original creators. How does mobile app theft take place? In many cases, it is invisible to consumers and to the mobile app developers themselves, who may not be fully aware that someone is siphoning off some of the revenue stream. It takes place in the following manner. A pirate developer downloads an app from a legitimate source, such as Google Play. They then deconstruct the app, embedding in it their own monetization methodology, and upload it back to some of the hundreds of alternative app stores. When such apps are downloaded and used, and consequently when adds are viewed, it is the pirate developer that enjoys the advertising revenue.

Such apps are more popular in China and other developing countries, while in the United States and Europe most apps are downloaded from Google Play and iOS App Store. The main solutions lie with detection which can be done through technological means, such as monitoring (e.g., monitoring the 20 to 30 leading alternative app stores), and enforcement.

Moving away from piracy, we have seen that a mobile app developer faces several important decisions regarding the level of an app's

protection. It requires understanding of the relevant systems that regulate our IP ecosystem, their differences and their advantages and disadvantages, all the while bearing in mind variations within jurisdictions.

Building on the discussions and examinations from earlier chapters, this final chapter examines the key milestones leading to such decisions. It is designed to assist the reader in understanding and identifying these milestones in order to enable informed decisions.

The following examination comprises an overview of protectability-related issues for mobile apps.

Copying and emulation risk mapping

This part of our overview concerns mobile app developer's ability and likelihood of success in fending off competition by relying on IP laws that protect the app.

Code

The actual code, whether in object code or source code format, is protected as a literary work under copyright law. Any part of the code which results from the developer's choice, rather than being dictated by external factors such as functionality considerations, is protectable. In principle, even a portion of the code that is less than 1 percent of the whole program could prove to be protectable under copyright law and a competing program that reproduces that portion may infringe including all the accompanying consequences (e.g., injunctions, delivery up and destruction orders, damages and account of profits). Also, in the case of source code, the presence of even a small portion of code in a competing product gives rise to the question of accessibility. How did the competitor obtain of this portion of code for copying? Since a mobile app is likely to be released into the market in an object code format, copying a portion of the source code will usually indicate either illegitimate code decompilation or potential violation of trade secrecy laws.

This analysis states “could” rather than “would” prove protectable. In most jurisdictions copying of what would otherwise amount to protected code may be excused under certain circumstances. For example, we have seen that reproducing code for the purpose of achieving interoperability between the target program and another program could be excused from copyright protection due to public policy considerations which encourage interoperability.

Notwithstanding the above, the presence of identical code in a competing program usually spells trouble for the latter. Establishing that the code in question was dictated by functionality (hence, un-protectable), or necessary to achieve interoperability (hence, reproduction may be exempted from copyright infringement) is burdensome, costly and uncertain. It follows that a mobile app develop that identifies such portion of code in a competing program will usually find himself in an advantageous position, having a leverage in any negotiations that may ensue with the proprietor of the competing, allegedly infringing, program.

Internal architecture

Some elements of a mobile app’s internal architecture may be described in its technical specifications document or system architecture paper. Mobile app developers may decide whether to have such documents or elements available.

Internal architecture elements include file formats and algorithms and more general concepts such as structure, sequence and organization. Internal software features in a competing app can raise questions of access. Unless independently created, the competing app developer may have had access to the target app’s internal architecture to copy it. Often such access is only possible because of reverse engineering and decompilation. We have seen that the latter is highly restricted under copyright law and, if at all, will only be possible for interoperability-related purposes. As mentioned, the onus of establishing that both decompilation and reproduction of elements of internal architecture are necessary for interoperability-related purposes is not easily determined. Alternatively, such access

may have been possible due to violation of trade secrets law, such as in the case of an ex-employee now working for the competing app.

The above analysis is subject to one main caveat: it is far from certain whether some architectural elements are eligible for copyright protection in the first place. Where this is the case, there is no need to establish that their reproduction was necessary for interoperability-related purposes as this is not protected under copyright to start with, and does not need to be excused from copyright protection. For example, it is far from certain to what extent, if at all, data file formats are protected under copyright law in the EU. However, although such questions may be of great interest to legal scholars, it is of less significance to the purpose of the present discussion. This is due to the fact that in order for such elements to be copied, they first need to be accessed. As mentioned, such access usually requires decompilation, which is often permissible only for interoperability-related purposes. This being so, even in the case of an element that is not protected under copyright law and may therefore be reproduced in this context, the initial access to it may nevertheless have to be justified on interoperability grounds.

Some elements of internal architecture may be subject to patent law. For example, some data files encode data using algorithms that may be subject to patent protection. Where this is the case, replicating such algorithms may require a license. Operating without one may result in patent infringement.

Where internal architecture elements are accessible with decompilation, there is a clear benefit in maintaining such elements under the veil of trade secrecy. First, it may be worthwhile to define the information as trade secrets in one's licensing agreement. This may help bolster, depending on the jurisdiction, the protection granted to such elements under trade secrets law. Even where such elements may not be eligible for copyright protection or may be excused from copyright infringement under certain circumstances, their classification as trade secrets may entitle them to trade secrets protection. Second, defining such elements as trade secrets may help a developer regulate the actions of its employees after their employment. Where properly defined and maintained as trade

secrets during the employment period, a developer may prevent a former employee from disclosing information to a competitor without time or geographic restrictions. While non-compete clauses or restrictive covenants are usually enforced where they are considered as fair, reasonable and justified, confidentiality clauses pertaining to trade secrets are not subject to these restrictions. And while non-compete clauses or restrictive covenants are restricted in relation to their term and geographical scope, these restrictions do not have to accompany confidentiality clauses to be enforceable. In conclusion, trade secrecy for elements that are not open to public inspection should be maintained wherever possible.

Finally, for mobile apps that are available only through the Internet and not available to download, trade secrecy is a key vehicle for protection. Apart of patent law, which grants protection even against independent creation, it is mainly through violation of trade secrets that a competitor may access and copy elements of a mobile app that relate to its internal "organs."

User interfaces

We have seen that with all other things being equal (or even similar), it is the GUI (graphical user interface) that has a significant impact on a mobile app's usability and, hence, its popularity. It is this feature that comprises a significant part of what is sometimes referred to as software's "look and feel," with non-graphical features such as command inline interfaces as well as APIs contributing to the overall "look and feel." GUIs may also be useful in establishing mobile apps' branding, by accustoming the users to a particular type of working environment associated with a specific source of origin.

Although potentially possible, patent protection for GUIs is difficult to obtain. In most cases it may be useful to focus on alternative forms of protection. Copyright, design protection and trademarks/trade dress protection appear to be most relevant. In essence, neither copyright, trademarks nor design laws grant protection to GUI features dictated by functionality, as protecting functionality is the province of patent law. This being said, the scope of the functionality exclusion under each of these IP rights is different. Arbitrary or

aesthetic-driven choices made during the design process are more likely to survive a functionality challenge. Therefore imprinting a GUI with an individual and unique character that will help to differentiate the developer's offering from other alternatives may make it simpler to fend off competition.

While both trademark and design laws require registration, copyright law does not. Consequently, in the case of potential copyright infringement, developers may act against competition even in jurisdictions where no preparatory protective steps are taken. In the case of imitation of unregistered trade dress, such as a mobile app's "look and feel," this too may enable a developer to act against imitative competition depending on the jurisdiction at hand, particularly in territories where protection of unregistered marks is weak. Hence, a developer designing a distinctive GUI for their mobile app would be well advised to attempt to register it. Failure to do so may mean that they may not be able to fall back on rules pertaining to unfair competition should a competitor chose to emulate the distinctive features of their GUI.

Logic and behavioral aspects

As mentioned, in addition to GUIs, non-graphical elements such as command inline interfaces as well as APIs may contribute to the overall "look and feel" of mobile apps.

These elements are protectable under different IP rights, including patent, copyright and trade dress laws. Patent protection of one's mobile app is difficult to obtain and may require considerable upfront costs. However, again, it is mainly copyright law and trade dress protection that may assist a mobile app developer. For copyright, although functionality elements such as behavioral features may not constitute copyrightable subject matter, elements such as APIs may be eligible for copyright protection in some jurisdictions.

Overall "look and feel" may be relied on when associated with a particular source of origin. When it is sufficiently distinctive so as to show a particular source or origin, trade dress laws may be used

to prevent appropriation. Where possible, such eventuality would be borne in mind by mobile app developers during the design and development stage. Where a mobile app could be designed so that that its operation is sufficiently distinct from industry norms, such departure could be viewed by consumers as designating a particular source of origin. To successfully rely on such indication of origin perception by consumers against imitators, it should be ensured that the behavioral aspects of the app that set it apart are not attributable solely to the functional objectives. Once it is not attributable solely to a functional objective, the developer is more likely to successfully claim trade dress protection.

Unlike laws governing registered trademarks, trade dress protection is less harmonized internationally and its scope varies significantly from one jurisdiction to another.

As regards registered trademarks, overall “look and feel” usually may not constitute a registerable sign due to the requirement of specificity in trade mark applications. However, we have seen that elements of it, whether static or dynamic, may be so protected. Such protection could prove vital when attempting to stop imitative competition.

Conclusion

Various IP rights may protect various aspects of mobile apps. The extent to which such protection is available depends on the app’s elements and the jurisdiction. Reliance on some IP rights does not require upfront costs associated with registration, while others exist only when registered.

As a general rule, where it is clear to a mobile app developer that a certain market would be central to its marketing efforts, it is advisable to consider registering IP rights, such as trademarks, designs, and patents – where possible. A variety of registered and unregistered rights could prove essential when fending off imitative competition. It gives the owner the flexibility to use one IP right when another is successfully challenged.

World Intellectual Property Organization
34, chemin des Colombettes
P.O. Box 18
CH-1211 Geneva 20
Switzerland

Tel: +41 22 338 91 11
Fax: +41 22 733 54 28

For contact details of WIPO's
External Offices visit:
www.wipo.int/about-wipo/en/offices

WIPO Publication No. 1071E
ISBN (Print) 978-92-805-3308-8
ISBN (Online) 978-92-805-3309-5
ISSN (print): 2789-5432
ISSN (online): 2789-5440