



SCHOOL OF
ECONOMICS AND
MANAGEMENT

Protecting Derivative Software: A Study of EU Copyright Law and Open-Source Licensing

Sarvenaz Soortejie

DEPARTMENT OF BUSINESS LAW

Master's Thesis in European and International Trade Law

15 ECTS

HARN63

Spring 2024

Contents

Abstract	5
Abbreviations	7
1 Introduction	9
1.1 Background.....	9
1.2 Purpose and research questions	11
1.3 Method and materials	11
1.4 Outline	13
2 Open Source Software.....	14
2.1 Introduction	14
2.2. Software.....	15
2.2 Development of the Open Source Software Industry.....	16
2.5 Conclusions	18
3 Copyright Protection of Software	20
3.1 Introduction	20
3.2 Copyright Protection	20
3.3 Directive 2009/24/EC on the legal protection of computer programs	22
3.3.1 Case Law.....	23
3.3.2 Criteria of originality in copyright.....	25
3.3.3 The Author's Rights.....	26
3.4 Derivative works in copyright.....	27
3.4.1 Derivative works in Member States Jurisdictions	28
3.5 Summary and conclusions.....	29
4 Open Source Software Licensing	31
4.1 Introduction	31
4.2 Licenses	31
4.3 Derivative works under OSS Copyleft licenses	33
4.3.1 GNU:s General Public License	34
4.4 Conclusion.....	35
5 Summary and conclusions	36
5.1 When can a computer program that was derived from the source code of another computer program be protected by copyright under the Computer Program Directive?	36

5.2	How can a creator of such a derivative work exercise his/her rights in relation to the rights-holder(s) of the original source code program under the Computer Program Directive?.....	36
5.3	To what extent does the use of a copyleft open-source software license, such as GNU General Public License, impact questions (1) and (2) under EU law?	37
5.4	Conclusions	38
	References.....	40

Abstract

Software applications are integral to various sectors, including manufacturing, commerce, transportation, healthcare, education, arts, and entertainment, with technological advancements, new applications continue to emerge, expanding the scope of software use. Open-source software (OSS) plays a crucial role in ensuring competition, promoting inclusiveness, and improving the digital environment, as supported by the European Commission. Moreover, it fosters macroeconomic benefits by preventing monopolies.

The purpose of this study is to describe and analyse the legal protection given to a computer program that was derived from the source code of another computer program by copyright under Directive 2009/24/EC on the legal protection of computer programs (Computer Program Directive). Firstly, this study aims to illustrate the rights of a derivative work creator in relation to the rightsholder of the original source code creator under the Computer Program Directive. Secondly, this study aims to clarify the impact that the use of a copyleft open-source software license, such as GNU General Public License has on the protection of derivative computer programs as well as the impact on the author's rights in relation to the rights-holder(s) of the original source code program under the Computer Program Directive.

Article 1 of the Computer Programs Directive states that computer programs shall be protected under the Berne Convention if the criteria of originality is met, in the sense that it is the author's own intellectual creation.

The use of a copyleft open source software license, like the GNU General Public License, impacts the first and second questions under EU law by enforcing the release of derivative software under terms that ensure access to the source code. The GPL, which is designed to safeguard freedom for software code, allows unrestricted copying and modification of the code but mandates that both copies and derivative works be distributed under the same license, without additional restrictions.

As a result, the Computer Program Directive, combined with the use of copyleft licenses, promotes innovation and collaboration in software development while protecting the rights of creators, on the other hand, it restricts the distribution rights of the author.

Keywords: Copyright, Computer Programs Directive, Software, Open Source Software Licensing, Originality, Derivative Works.

Abbreviations

CJEU	Court of Justice of the European Union
CVS	Concurrent Versions System
EU	European Union
EUIPO	European Union Intellectual Property Office
EULA	End User License Agreement
FSF	Free Software Foundation
GNU	GNU's Not UNIX
GPL	General Public License
GUIs	Graphical User Interfaces
OSD	Open Source Definition
OSI	Open Source Initiative
OSS	Open Source Software
TRIPS	Trade-Related Aspects of Intellectual Property Rights
WIPO	World Intellectual Property Organization
WTO	World Trade Organisation
WWW	World Wide Web

1 Introduction

1.1 Background

Software applications are an integral part of our daily lives and are widely used in various sectors such as manufacturing, commerce, transportation, healthcare, education, arts, and entertainment. As technology develops, new applications for computer programs continue to emerge, expanding their scope of use.¹ According to the European Commission open-source software has a role to play in ensuring competition and promoting an open approach as well as inclusiveness for an improved digital environment.²

Many modern products and services now rely on software for their operation, including e-readers and smartphones. Additionally, there's a growing trend towards the widespread adoption of cloud computing services. These services, hosted on the internet, cover a wide range of software solutions, from remote access to computing resources and data storage to messaging applications. Consequently, this development has also meant that the hardware is increasingly becoming a shell for the software.³

Open Source software has many benefits such as usually being free of cost, better capability of portability, legally able to modify through license, typically more secure and reliable products in comparison to proprietary developed systems and even possible macroeconomic benefits because of its antimonopolistic character.⁴ The European Union Intellectual Property Office (EUIPO) found in its report from surveys an indication that open-source software (OSS) does not constrain but rather expands the business prospects for software companies. Many of these firms view OSS as a chance for growth rather than a challenge to their business model.⁵

The Free Software Foundation (FSF) and the Open Source Initiative (OSI) represent two distinct branches of the free and open-source movement. The FSF advocates for "free software," emphasizing user and developer freedoms. The FSF's General Public License (GPL) ensures that software licensed under

¹Wolk, S., 'Datorprogramalster I Upphovsrätten: Skyddet För Datorprogram, Datorspel, Bildskärmsuttryck, Design, Filformat Och Algoritmer, M.M', (1st edn, Iustus, 2016), page 15.

² European Commission, COMMUNICATION TO THE COMMISSION OPEN SOURCE SOFTWARE STRATEGY 2020-2023, Think Open, Brussels, 21.10.2020, page 2.

³Wolk, S., (2016), page 15.

⁴Deek FP and MacHugh JA, Open Source: Technology and Policy (Cambridge University Press 2010) <<https://www.cambridge.org/core/books/open-source/16BC3434DD89136889E7A1F6FD1AC686#fndtn-information>> accessed 10 April 2024, page 3.

⁵EUIPO, OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION, June 2020, page 10.

it cannot be redistributed in closed, proprietary distributions, embodying a "share and share alike" philosophy. The FSF acknowledges other licenses as free if they grant users the rights to run, access, modify, and redistribute the code and modifications. In contrast, the OSI defines criteria for "open source" licenses, which do not necessarily require free redistribution of modifications, allowing for proprietary distributions.⁶

Open source software is public code, meaning it's a wise investment of public funds, promoting freedom of choice, it facilitates the use and reuse of software solutions, enabling collaboration across borders to develop interoperable services and enhance efficiency. Open-source software allows for easy and efficient addition of features, which can be openly shared with others, ensuring widespread benefits for all. Gradual innovation is promoted through the exchange of knowledge and expertise, by ensuring transparency, open-source software enhances trust in public services. Additionally, it aids in resolving technological challenges by encouraging contributions from various sources, often leading to unforeseen solutions.⁷

Intellectual property rights legal roots can be found in ancient Roman times where the principle of *accessio* was relevant, which stated that one could gain ownership of a canvas if that person had painted it.⁸ What characterizes intellectual property rights is that it is a negative right where the rightsholder may solely prevent others from using the work.⁹ Both the TRIPS Agreement and the WIPO Copyright Treaty explicitly designate computer programs as literary works deserving copyright protection under the Berne Convention.¹⁰

Through mainly licensing can the rightsholder gain remuneration for its work and distribute it. As the main rule, according to the principle of freedom of contract the contracting parties are free to decide the terms of their agreement.¹¹ However, there are exceptions to that rule when the law is binding and other than what the law says cannot be agreed upon. Copyright licenses are agreements where the owner of the copyright gives someone else the right to use or distribute the work.

The issue arises when programmers develop their own code and create computer programs, raising questions about copyright, authors' distribution rights, and the extent of code alteration required to classify it as a derivative

⁶ Ibid., page 2.

⁷ European Commission, COMMUNICATION TO THE COMMISSION OPEN SOURCE SOFTWARE STRATEGY 2020-2023, Think Open, Brussels, 21.10.2020, page 2.

⁸ Pila J., Torremans P., "European Intellectual Property Law", second edition, Oxford University Press, 2019, page 33.

⁹ Ibid., page 5.

¹⁰ EUIPO, OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION, June 2020, page 16-17.

¹¹ Communication from the Commission — Guidance on the Commission's enforcement priorities in applying Article 82 of the EC Treaty to abusive exclusionary conduct by dominant undertakings (The Guidance Paper), 2009/C 45/0, para. 75.

work. This study examines these issues from an EU law perspective and provides a brief overview of the individual copyright laws of Member States.

1.2 Purpose and research questions

The purpose of this study is to describe and analyse the legal protection given to a computer program that was derived from the source code of another computer program by copyright under the Computer Program Directive. Firstly, this study aims to illustrate the rights of a derivative work creator in relation to the rightsholder of the original source code creator under the Computer Program Directive. Secondly, this study aims to clarify the impact that the use of a copyleft open-source software license, such as GNU General Public License has on the protection of derivative computer programs as well as the impact on the author's rights in relation to the rights-holder(s) of the original source code program under the Computer Program Directive.

For the aim of this study to be achieved, the following question must be answered:

1. When can a computer program that was derived from the source code of another computer program be protected by copyright under the Computer Program Directive?
2. How can a creator of such a derivative work exercise his/her rights in relation to the rights-holder(s) of the original source code program under the Computer Program Directive?
3. To what extent does the use of a copyleft open-source software license, such as GNU General Public License, impact questions (1) and (2) under EU law?

1.3 Method and materials

This study adopts the legal dogmatic approach to review and analyse the current state of legislation, which consists of answering a legal question based on legal sources in a hierarchical order. This method relies on the law, court decisions, preparatory work, and doctrine to address legal issues. The named legal sources follow a hierarchy structure and must be taken into account during the interpretation process. Primary sources such as EU law and binding national constitutions hold the highest authority, followed by statutes and national laws. Court decisions and preparatory work should be examined,

while legal literature and doctrine provide additional insights and therefore may be used.¹²

The research question is answered from an EU law perspective where EU directives are in focus and relevant international agreements such as the Bern Convention, Trade-Related Aspects of Intellectual Property Rights (TRIPS) and WTC are considered. To analyse the interpretations of the directives, relevant case law from the EU Court of Justice (CJEU) will be assessed. Legal doctrine and literature will be used to describe some concepts, notions and background history of copyright law. The EU legal method is used when assessing EU law, official EU documents and CJEU case law.

The CJEU developed the EU legal method to be used as a tool for the interpretation of European Union law.¹³ The core of the EU legal methodology is teleological interpretation, which means purpose oriented. EU legislation should be read with the context and underlying purpose in mind.¹⁴ The words or concepts of national law must be interpreted independently and separately from the concepts of other Member State legislation.¹⁵

The CJEU made it clear in Case C-26/62 that CJEU Treaty Provisions have the capacity of direct effect, as well as that the right of sovereignty of Member States is not unconditional as regards international law.¹⁶ The supremacy of Community laws over national law was established in Case C-6/64.¹⁷ According to Case C-11/70, the EU Community law includes principles such as non-discrimination, legitimate expectations, proportionality, legal certainty and fundamental rights, which should be taken into account.¹⁸

Article 216(2) Treaty on the Functioning of the European Union (TFEU) states that international law becomes part of the Union law through agreements where the EU is a participating party. When there are different ways of interpreting EU law, the interpretation should comply with international agreements.¹⁹

¹² Peczenik A, 'Juridikens Teori Och Metod: En Introduktion till Allmän Rättslära', (1st edn, Fritzes 1995), page 33.

¹³ Hettne J., Eriksson I., "EU-rättslig metod, teori och genomslag i svensk rättstillämpning", Norstedts Juridik, second edition, 2011, page.34.

¹⁴ Ibid. Page 36.

¹⁵ Ibid. Page 161.

¹⁶ Case C-26/62 NV Algemene Transporten Expeditie Onderneming van Gen en Loos v Nederlandse Administratie der Belastingen (1963) ECR.

¹⁷ Case C-6/64 Costa v ENEL (1964) ECR I-1141.

¹⁸ Case C-11/70 Internationale Handelsgesellschaft mbH v Einfuhr- und Vorratsstelle für Getreide und Futtermittel (1970) ECR 1125 (Solange I).

¹⁹ Lenaerts, K., & Gutiérrez-Fons, J. A. (2014). To Say What the Law of the EU is: Methods of Interpretation and the European Court of Justice. In *Columbia Journal of European Law* (Vol. 20, Issue 2, pp. 3–61). Parker School of Foreign and Comparative Law, page 33-34.

Finally, in the second chapter scientific literature, articles and reports are used for describing computer programs technicality and the history of open-source licensing. In the fourth chapter the GNUs General Public License and its terms are analysed.

1.4 Outline

This study consists of five chapters which will be outlined in this section, beginning with the first chapter on the background of the study, purpose and research question, method and material.

Next, the second chapter provides an overview of computer programs and the development of the software industry. The second chapter aims to provide a base knowledge of what computer programs are and how open source software industry started, as well as a better understanding of its underlying business model.

The third chapter examines the relevant legal framework including directives, international treaties and case law for software copyright protection. The third chapter aims to provide a better understanding of the criteria of originality regarding computer programs and analyze what qualifies as a derivative work across the EU and Union Member States jurisdictions. Lastly, this chapter examines the author's rights according to EU law.

The fourth chapter begins with a categorisation of open source licenses and the difference between nonreciprocal and reciprocal as well as proprietary and free or open licenses. Next, the key terms of GNU General Public License, which includes a copyleft clause and its impacts on software distribution and alteration, are assessed. Lastly, the conclusions are presented in the part of this chapter.

Finally, the study's questions are answered in chapter five, where the conclusions are stated.

2 Open Source Software

2.1 Introduction

The predecessors of modern computers emerged as mathematical devices during the 1930s and 1940s. They were initially seen as scientific tools and were used primarily in research work and military applications. Key milestones in their technological evolution included the introduction of the transistor, the integrated circuit, and the microprocessor.²⁰

Today the processor and memory circuits are the main important components of the hardware, while other functionalities of the hardware have been transferred into the software.²¹ Free software, also known as open-source software, forms the foundation of today's information technology, it is used in our smartphones and laptops to household appliances and industrial machinery in the background and unnoticed. Open-source software is also essential in activities such as emailing, web browsing, and accessing information on smartphones.²² The Internet relies heavily on open-source software, important functions like domain name assignment, mail routing, and critical components of the TCP/IP network stack utilize open-source software.²³

Contrary to more common business models, the Open Source movement aims to promote collaboration and transparency for the development of software. Its goal is to make scientific advancements in software accessible to everyone for further enhancement and understanding. Its products and processes are openly available for scrutiny and improvement, facilitated by the Internet. The source code is freely accessible for viewing, study, modification, and redistribution, without discrimination or restrictive licensing. In contrast, proprietary software development occurs behind closed doors, with limited transparency and strict control over code distribution and modification. The two approaches represent opposites in software development methodologies, but there are instances where collaboration between commercial and open-source communities occurs.²⁴ In contrast, alternatives to open-source

²⁰ Ibid., page 16.

²¹ Ibid., page 16.

²² Fogel K, (2023), page 1.

²³ Kavanagh P, Open Source Software: Implementation and Management (Software Development) (1st edn, Digital Press 2004) <<https://www.sciencedirect.com/book/9781555583200/open-source-software#book-info>> accessed 12 April 2024, page 29-30.

²⁴ Deek FP & MacHugh JA, (2010) accessed 10 April 2024, page 1.

software are often referred to as proprietary, commercial, or closed software.²⁵

In this chapter, an overview of computer programs and the development of the software industry is provided. This chapter aims to provide a base knowledge of what computer programs are and how open source software industry started, as well as a better understanding of its underlying business model.

2.2. Software

The concept of the World Wide Web (WWW) was conceived by physicist Berners-Lee while working at the CERN physics laboratory in Switzerland. The development of a global hypertext system was proposed by him in 1989. In 1991, Berners-Lee created the first web server and a text-based web browser, which he released for the NeXT operating system. He dubbed his browser "WorldWideWeb" and his web server "http".²⁶

A computer program is essentially a manufactured product comprising a sequence of instructions that enable a computer to execute various tasks.²⁷ Software is constructed using lines of code that humans can read, known as source code, which essentially defines what the software can do, such as specific tasks it can perform.²⁸

In many ways, software operates similarly to a machine: it interacts with hardware and other software components to achieve its goals. It's made up of multiple components and assembly parts, although they are not physical, they must cooperate effectively for the software to function correctly.²⁹

Ensuring compatibility with other software is a crucial strategic consideration during the development process. Compatibility allows users to run files generated by different software programs or to run software on specific operating systems. Achieving compatibility relies on standardization, which involves formal or informal agreements to develop and implement technical standards. These standards facilitate the interoperability of various software and hardware components. Standardization fosters collaboration, encouraging the development of new functionalities while ensuring compatibility with existing technical solutions.

Thanks to CVS (Concurrent Versions System) multiple developers can make changes to the code simultaneously and facilitate the merging of these

²⁵ Kavanagh P., (2004) accessed 12 April 2024, page 2.

²⁶ Deek FP & MacHugh JA, (2010) accessed 10 April 2024, page 23-24.

²⁷ COM (88) 172 final s.170.

²⁸ EUIPO, OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION, June 2020, page 12.

²⁹ EUIPO, OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION, June 2020, page 12.

changes into an updated version of the project while monitoring for potential conflicts. Notably, CVS also enables easy access to previous versions of the project, serving as a comprehensive record of its development history, often likened to a "time machine". It is distributed as open source under the General Public License (GPL). And stores software projects in a central repository on a server, which is accessed by remote clients.³⁰

Decompiling is one type of reverse engineering, which involves undoing the process of compiling source code into object code. This can be done by remodelling the object code back into a higher-level language, making it easier to understand. A countermeasure to decompiling is obfuscation, which entails rearranging code and modifying symbolic names to keep the code's functionality while making it harder to read and therefore harder to reverse engineer.³¹

'Compilation' is the process of converting text code (high-level) into machine code in the form of a binary (low-level) and may also be the act of creating a code based partly on already copyrighted works.³²

2.2 Development of the Open Source Software Industry

The first modern business computers emerged in the 1950s, building on advancements of World War II. These early machines were unique and ran custom programs written in assembler language or binary codes.³³

During the 1960s, commercial mainframe systems were tailored for specific applications, often programmed in COBOL or FORTRAN. System software, including crucial programs like CICS and IMS, was developed in assembler language. IBM's System/360, introduced in 1964, marked a significant milestone as it allowed the same assembly language to run across every related system, enabling system software compatibility within a group of computers.³⁴

Before the 1970s, software was considered "open" or "free" by default, stemming from its origins in scientific research where information sharing was common practice. During the 1970s the new concept of "closed-source software companies" emerged.³⁵ Bill Gates representing Microsoft, wrote an

³⁰ Deek FP & MacHugh JA, (2010) < accessed 10 April 2024, page 120-123.

³¹ Ibid., page 241.

³² Ibid., page 225.

³³ Kavanagh P., (2004) accessed 12 April 2024, page 6.

³⁴Ibid., page 6.

³⁵Ibid., page 6.

open letter to hobbyists in 1976, emphasizing that software should be regarded as private property rather than a public good.³⁶

In 1984, GNU standing for "GNU's Not UNIX," was initiated by Richard Stallman to create a complete operating system akin to UNIX but devoid of any UNIX code, thus avoiding potential legal constraints associated with corporate ownership. Important UNIX-like utilities were developed by this project, such as compilers, debuggers and editors independently from UNIX, leading to the creation of strong and efficient tools accessible across various systems.³⁷ GNU aimed to craft a completely liberated and transparent computer operating system along with a suite of application software, ensuring users would always retain the freedom to modify and share.³⁸ Richard Stallman published in 1984 the underlying concept and arguments of "free software", which is that software can be free as in making changes to it and analysing it as needed.³⁹

The "four freedoms" outlined by the Free Software Foundation (FSF) in 1985 was a result of dissatisfaction with the End User License Agreement (EULA) terms. By using copyright law and licensing terms they were able to promote access to source code and ensure the four following freedoms:

1. The freedom to use software for any purpose.
2. The freedom to study and modify the program, provided access to the source code is available.
3. The freedom to distribute copies to others.
4. The freedom to distribute modified versions, allowing the community to benefit from improvements, provided access to the source code is available.⁴⁰

In 1989 the GNU Public License was introduced, reflecting on the Free Software Foundation (FSF), established by Stallman. The purpose and vision of the FSF are to create an environment where programmers can openly display their work without commercial intervention.⁴¹

The phrase "open source" came in 1997 by a group of people, mainly because the term "free software" was not as attractive to companies.⁴² The phrase "free software" could be mistaken for free-of-cost software and falsely assumed that it didn't have anything to do with the right to modify and redistribute the codes.⁴³

³⁶Ibid., page 7.

³⁷Ibid., page 9-10.

³⁸ Fogel K., (2023), page 5.

³⁹ Kavanagh P., (2004), page 2.

⁴⁰ EUIPO, OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION, June 2020, page 23-24.

⁴¹ Ibid., page 10.

⁴² Ibid., page 2.

⁴³ Fogel K., (2023), page 7.

In technical terms, the distinction lies in the types of licenses endorsed by two different organizations. "Free software" refers to licenses that comply with the "four freedoms" outlined in the Free Software Definition by the Free Software Foundation. Conversely, "open source" pertains to licenses that align with the Open Source Definition approved by the Open Source Initiative.

Linus Torvalds, a student at Helsinki University, embarked on developing a free UNIX kernel tailored for Intel-based PC systems in 1991. At first drawing inspiration from Andrew Tanenbaum's Minix system, which served as a teaching tool based on early UNIX source code, Torvalds set out to create a kernel that would be freely accessible and adaptable for PC users.⁴⁴ Linus named this kernel Linux, which also marked the first occasion where users could boot up their computers and perform tasks without relying on any proprietary software.⁴⁵

Torvalds adopted an unconventional "hacker" model of software development from the outset, although the term was not formally recognised then. Although this approach is unconventional, it has proven to be very successful in fostering collaborative development and innovation within the software community.⁴⁶ The Open Source Definition (OSD) was at first drafted by Bruce Perens for Debian Linux in 1997.⁴⁷

2.5 Conclusions

In conclusion, software is a computer program that by receiving a sequence of instructions it executes various tasks.⁴⁸ These instructions consist of lines of code that humans can read, known as source code.⁴⁹

In 1984, GNU standing for "GNU's Not UNIX," was initiated by Richard Stallman to create a complete operating system akin to UNIX but devoid of any UNIX code, thus avoiding potential legal constraints associated with corporate ownership.⁵⁰ GNU aimed to craft a completely liberated and transparent computer operating system along with a suite of application software, ensuring users would always retain the freedom to modify and share.⁵¹

⁴⁴ Kavanagh P., (2004), page 10.

⁴⁵ Fogel K., (2023), page 6.

⁴⁶ Kavanagh P., (2004), page 10.

⁴⁷ Ibid., page 1.

⁴⁸ Mazziotti G., EU Digital Copyright Law and the End-User, (1 ed. 2008), Springer Berlin, page 58.

⁴⁹ EUIPO, OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION, June 2020, page 12.

⁵⁰ Ibid., page 9-10.

⁵¹ Fogel K., (2023), page 5.

By using copyright law and licensing terms they were able to promote access to source code and ensure the four following freedoms outlined by the Free Software Foundation (FSF) in 1985:

1. The freedom to use software for any purpose.
2. The freedom to study and modify the program, provided access to the source code is available.
3. The freedom to distribute copies to others.
4. The freedom to distribute modified versions, allowing the community to benefit from improvements, provided access to the source code is available.⁵²

⁵² EUIPO, OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION, June 2020, page 23-24.

3 Copyright Protection of Software

3.1 Introduction

Article 17 of the Charter of Fundamental Rights of the European Union, states that a person has the right to own, dispose and use his or her lawfully acquired possessions, as well as that intellectual property shall be protected.⁵³ Some general justifications of why copyright is important are firstly the natural right a person has to own her creation, secondly the need to reward the creation to promote new creations and thirdly aspects of incentives such as public interest. Recital 3 of the Computer Programs Directive notes that computer program technology is considered to be of crucial concern for the European Union's industrial progress.

Copyright protects scientific, artistic or literary expressions and not ideas, this is stated in Article 2 of WIPO Copyright Treaty (WCT)⁵⁴, Article 9(2) TRIPS.⁵⁵ The qualifications for protection are that the work is an original creation which is not a copy of another's work. The copyright holder has exclusive rights to reproduce the work which will make it available to the public and the moral rights.

Software as a literary work is connected to the technical description of the program and is related to the parts that enable the computer to fulfil a task or a special function.⁵⁶⁵⁷

This chapter assesses the relevant legal framework including directives, international treaties and case law for software copyright protection. This chapter aims to provide a better understanding of the criteria of originality regarding computer programs and analyze what qualifies as a derivative work across the EU and Union Member States jurisdictions. Lastly, this chapter examines the author's rights according to EU law.

3.2 Copyright Protection

The notion of granting copyright protection to computer programs wasn't at first evident but gradually gained recognition over time. Initially, it was

⁵³ Charter of Fundamental Rights of the European Union 2012/C 326/02.

⁵⁴ WIPO Copyright Treaty (WCT), (1996), TRT/WCT/001.

⁵⁵ Agreement on Trade Related Aspects of Intellectual Property Rights (TRIPS) (1994).

⁵⁶ Case-393/09, para 38.

⁵⁷ Wolk, S., (2016), page 49.

unclear whether computer programs could be classified as literary works under the Berne Convention for the Protection of Literary and Artistic Works dating back to 1886. Alternative protection mechanisms, such as *sui generis* protection, were also debated internationally during the 1970s. In 1978, the World Intellectual Property Organization (WIPO) introduced a model law titled "Model Provisions on the Protection of Computer Software," aiming to offer a unique form of protection for computer software. However, this model law only provided a basic level of protection for computer programs and associated materials. Despite discussions at both international and regional levels, the concept of *sui generis* protection for computer programs failed to gain traction.⁵⁸

The idea that computer programs should be protected by copyright as literary works gained popularity in the late 1970s and during the 1980s. At the time, many countries recognized copyright as the primary intellectual property protection for computer programs.⁵⁹

Even if the Bern Convention does not explicitly mention a certain type of work, such as computer programs, new works are protected by the Convention according to a general principle, so-called the Bern Convention plus principle.⁶⁰ Article 2.1 of the Berne Convention states that within the expression literary works are included in all kinds of works in the literary field, regardless of the way and in what form they are expressed. Article 5(2) of the Berne Convention states that no formalities shall be required for copyright.

Article 10.1 of the TRIPS Agreement states that computer programs in the form of source and object code must be protected by copyright and that the Berne Convention on Literary Works provisions must also be applied to them. The 1996 WIPO Copyright Treaty (WCT) states in Article 4 that computer programs are to be protected as literary works, regardless of the manner or form in which they are expressed.

The Berne Convention bans the act of translating a copyrighted work into a different programming language if the owner of the work has not given its permission. As a main rule software that consists of translated or changed source code from an original program is derivative.⁶¹

Council Directive 2009/24/EC, dated April 23, 2009, harmonises copyright protection for computer programs across EU Member States. Article 1 of the Directive mandates that computer programs be protected under copyright as literary works according to the Berne Convention's definition. Additionally,

⁵⁸ Wolk, S., (2016), page 19-20.

⁵⁹ *Ibid.*, page 20.

⁶⁰ *Ibid.*, page 21–22.

⁶¹ Deek FP & MacHugh JA, (2010), page 226.

other aspects of copyright protection, such as safeguarding technological measures and rights-management information for software, are also standardized by the Information Society Directive 2001/29/EC, (InfoSoc).⁶² The duration of copyright protection is determined by Directive 2006/116/EC.⁶³ Important to note that the InfoSoc Directive does not affect the specific rules regarding protection set out in the Computer Programs Directive, according to Recital 50 of the InfoSoc Directive.

Copyright also extends to the protection of Graphical User Interfaces (GUIs), as affirmed by the CJEU in the BSA Ministerstvo kulture case.⁶⁴ However, GUIs can be protected under ordinary copyright law according to Directive 2001/29.⁶⁵

Following the principle of idea-expression dichotomy, copyright protection applies solely to the expression of ideas, not the ideas themselves.⁶⁶ This principle aims to prevent the monopolisation of ideas and ensure fair rewards for authors. Consequently, copyright safeguards the written form of computer programs (source code), but excludes functionalities, programming languages, data formats, and architectural features.⁶⁷

The term computer programs as literary work can be interpreted as including the writing of the program itself.⁶⁸ However, the protection is not affected by the difficulty degree of the function as such.⁶⁹ How the code enables another machine to perform its function is important when assessing computer programs' copyrightability,⁷⁰ which means that copyright protection of the program in all its forms occurs for the code, but not for what is expressed on the screen.⁷¹

3.3 Directive 2009/24/EC on the legal protection of computer programs

In the harmonization work, the European Commission identified in the 1985 White Paper that copyright protection for computer programs was particularly important to investigate in more detail,⁷² and in June 1988, the Commission

⁶² Directive 2001/29/EC of the European Parliament and of the Council of 22 May 2001 on the harmonisation of certain aspects of copyright and related rights in the information society.

⁶³ Directive 2006/116/EC of the European Parliament and of the Council of 12 December 2006 on the term of protection of copyright and certain related rights.

⁶⁴ C-393/09, para 38.

⁶⁵ EUIPO, OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION, June 2020, page 16-17.

⁶⁶ Recital 11 Directive 2009/24/EC on the legal protection of computer programs.

⁶⁷ EUIPO, OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION, June 2020, page 16-17.

⁶⁸ Wolk, S., (2016), page 48.

⁶⁹ Prop.1988/89:85 om upphovsrätt och datorer page 28.

⁷⁰ Wolk, S., (2016), page 49.

⁷¹ C-393/09, para 41.

⁷² COM (85) 310 final, para 149.

presented a special discussion document, a Green Paper.⁷³ It stressed that software development often requires a lot of investment, but the increasingly serious phenomenon of computer software piracy has caused damage to the European software industry.⁷⁴ In 1991, this work resulted in Directive 91/250/EEC on the legal protection of computer programs, which is updated now to Directive 2009/24/EC on the legal protection of computer programs.

Article 1.1 of the Computer Programs Directive states that computer programs shall be protected under the Berne Convention. According to Recital 7 of the Computer Programs Directive, all forms of computer programs shall be protected, including those incorporated into hardware and preparatory design work. Therefore, copyright protection applies in all forms in which computer programs can be embodied including when the program is in machine-readable form and is in written form such as flowcharts or system descriptions.⁷⁵

Article 1.2 of the Computer Programs Directive provides that only the expression of the program is protected regardless of its form. Therefore, ideas and principles that describe elements of the computer program such as its interfaces are not protected. Recital 11 of the Computer Programs Directive outlines that algorithms, programming languages and logic hold ideas and principles, that are not covered by the directive. Accordingly, others can create their own original code built on the principles or ideas of copyrighted software and not infringe on that work, even if both computer programs fulfil the same function.

Furthermore, Article 1.3 of the Computer Programs Directive states that for copyright protection to be applicable, the program needs to be original in the sense that it is the author's own intellectual creation.

Article 2 of the Computer Programs Directive declares that authorship of computer programs can be given to natural persons, a group of natural persons, or legal persons if the legislation of the Member State allows it.

The Member States' national copyright law shall protect computer programs in the same way as other literary works, according to Article 3 of the Computer Programs Directive.

3.3.1 Case Law

Firstly, in the BSA case, the EU Court dealt with the question of whether a program's graphical user interface should be considered part of the computer program protection and fall within the work category computer program or

⁷³ COM (88) 172 final.

⁷⁴ COM (88) 172 final., page 35.

⁷⁵ Prop.1988/89:85 om upphovsrätt och datorer page 8.

whether the display expression can have its own independent copyright protection. The case concerned a request for a preliminary ruling made by the Supreme Administrative Court of the Czech Republic in a dispute about, among other things, whether the concept of a computer program in Article 1.2 of the Computer Program Directive includes a program's graphical user interface. In its rulings, the European Court of Justice held that the graphical user interface of a computer program is not a form of expression for a computer program. However, it was emphasized that the graphical user interface can enjoy independent copyright protection.⁷⁶

Secondly, in the SAS case, the court emphasized that program languages, file formats, and program functions cannot be categorized as software programs, but can be in the same category as other types of works if the criterion for copyright protection is fulfilled.⁷⁷ This case concerned the developer of analytical software SAS Institute, which among other things developed a group of programs consisting of a system in statistical analysis named the SAS-system. The system could be used by others through “scripts” written in the ‘SAS Language’ which is unique for the SAS system. WPL, another company saw this demand and created a competing system, called the WPS system which mimicked the functions of the SAS system and made it possible to run scripts in the SAS language in the WPS system. SAS Institute initiate proceedings on copyright infringement grounds.⁷⁸ However, SAS Institute was not able to show that WPL had access to the SAS system, nor that it copied any part of the source code. According to WPL, they studied how the program worked and then by mimicking the functions of it, wrote their own program.⁷⁹ According to the CJEU ruling, neither the programming language and file formats used nor the function of a program are in the same category as the computer program itself employing Article 1 of the Computer Programs Directive.⁸⁰

Lastly, in the Nintendo case regarding whether the Computer Programs Directive or Information Society Directive 2001/29 (‘Infosoc’) should be applied for technical protection measures, the CJEU held that if a computer program such as a computer game, largely consists of other copyrighted work the Infosoc directive shall be applied to the whole and not the Computer Programs Directive.⁸¹ Nintendo worked with creating video games and marketed those games with two products, firstly the portable system, named ‘DS’ consoles and secondly by a fixed console system named ‘Wii’. These

⁷⁶ C-393/09 Judgment of the Court (Third Chamber) of 22 December 2010. *Bezpečnostní softwarová asociace - Svaz softwarové ochrany v Ministerstvo kultury*.

⁷⁷ Case C-406/10, Judgment of the Court (Grand Chamber), 2 May 2012. *SAS Institute Inc. v World Programming Ltd*, parag 46.

⁷⁸ Case C-406/10, para 23-24.

⁷⁹ Case C-406/10, para 44.

⁸⁰ Case C-406/10, para 46.

⁸¹ Case C-355/12, Judgment of the Court (Fourth Chamber), 23 January 2014. *Nintendo Co. Ltd and Others v PC Box Srl and 9Net Srl*.

products were protected technologically by installing a recognition system in the consoles and a copyright protected encrypted code. As a result, others lacking the encrypted code were unable to use illegal copies of Nintendo's video games.⁸² The PC Box equipment could be installed on the consoles and by that facilitated illegal use of videogames, to which Nintendo initiated legal copyright infringement proceedings.⁸³ According to the CJEU the InfoSoc Directive shall be applied in this case because the videogames consist of complex multimedia products, including sounds, images and a software. In conclusion even if images and sounds coded in programming languages are independently protected by copyright, these cannot be separated and therefore these parts have contributed to the originality of the work as a whole.⁸⁴

3.3.2 Criteria of originality in copyright

For a work to be eligible for copyright, it must originate from the author's creative effort. In other words, it needs to reflect the author's unique expression and individuality. Copyright protection is granted to works that demonstrate subjective novelty, meaning that anyone who independently creates a work through their own personal effort can claim exclusive rights to it.⁸⁵

Computer programs qualify for copyright protection if the author's intellectual creation demonstrates originality. Other criteria should not be implemented for assessing copyright eligibility. The concept of originality is an EU legal concept, according to the European Court of Justice and so must be interpreted independently and uniformly through the Union.⁸⁶

According to the CJEU in the Painer Case C-145/10, a work is protected by copyright if it reflects the author's intellectual creativity and personality. In practice, this means when an author has succeeded in demonstrating his creative ability through making free and creative choices during the creation process.⁸⁷ The CJEU has explained in the Painer case that originality can be expressed in many ways and at different times in the works production period, for example in the preparation phase, in the creation phase, and the selection phase by adding a "personal touch".⁸⁸

In the BSA case, the CJEU held that account shall be taken to the specific arrangement or configuration of all the components that form part of the

⁸² Case C-355/12, para 9-10.

⁸³ Case C-355/12, para 11-16.

⁸⁴ Case C-355/12, para 22-23.

⁸⁵ Levin M, *Lärobok I Immaterialrätt: Upphovsrätt, Patenträtt, Mönsterrätt, Känneteckensrätt i Sverige, EU Och Internationellt* (11th edn, Wolters Kluwer 2017), page 83.

⁸⁶ Case C-5/08, Judgment of the Court (Fourth Chamber) of 16 July 2009, *Infopaq International A/S v Danske Dagblades Forening*, para 35.

⁸⁷ Case C-145/10, *'Eva-Maria Painer v. Standard VerlagsGmbH and Others'*, 2011, para 88-89.

⁸⁸ Case C-145/1, para 90-92.

work.⁸⁹ Furthermore according to the Court in the BSA case the components that are necessary for the technical function shall not be considered, due to the inseparability of the idea and the expression.⁹⁰

In the SAS case the court emphasised originality being based on the sequence, combination, and choice of figures, mathematical concepts, or words that the creator expresses originality by being creative according to paragraph 45 of the previous Infopaq International case.⁹¹ In the case of software, the syntax, keywords, options, commands, and combination of commands as well as iterations consisting of figures, mathematical concepts and figures are not intellectual creations.⁹²

The creator makes a series of intellectual decisions when developing the computer program. The creator's initial individual assessments of the program's characteristics serve as the foundation for his design, mirrored in the program's actual code. The creator's decisions and considerations during the program's design phase are influenced by their understanding of its intended characteristics and functionalities. The various considerations a program creator might take into account during the design process, such as how it should be displayed on different screens, how it should respond to inputs such as keyboard, mouse or speech commands and how it should interact with various hardware components are examples of how broad the scope of identifying decisions that have gone into crafting a computer program.⁹³

Recital 8 of the Computer Programs Directive states that when evaluating originality, “no tests as to the qualitative or aesthetic merits of the program should be applied”. Therefore, solely the intellectual choices made during the creation of the program should be of importance.

In the Infopaq case, it was established that even a small number of words can be eligible for copyright protection, which in practice means that the bar for what qualifies as original is relatively low.⁹⁴

3.3.3 The Author's Rights

Article 4 of the Computer Programs Directive provides the author's rights regarding copyrighted software. The author has the exclusive right to reproduction by any means and in any form, right to translation, any alteration, adaption or arrangement of the software and the reproduction of its

⁸⁹ C-393/09, para 48.

⁹⁰ C-393/09, para 49.

⁹¹ Case C-406/10, para 67.

⁹² Case C-406/10, para 66.

⁹³ Wolk, S., (2016), page 73.

⁹⁴ Case C-5/08, para 47.

results, as well as all forms of distribution to the public including copies of the original work.

According to Recital 15 of the Computer Programs Directive, the permission of the author must be taken before reproducing, adapting, transforming or translating the form of the code. Committing these acts is an infringement of the author's exclusive rights, but under the conditions that such actions are necessary to create independently a new program and attain interoperability.

As mentioned in the second chapter of this study, a computer program consists of instructions in the form of source codes, given to the machine. Having in mind that the source code encompasses the instructions the machine needs to perform, a momentary reproduction of the source code is made every time the program is used.⁹⁵ Article 4 of the Computer Programs Directive therefore evidently gave the rights of any momentary reproduction to the owner.⁹⁶

3.4 Derivative works in copyright

The term "derivative work" is not explicitly defined in EU Legislation. However, the InfoSoc Directive (Directive 2001/29/EC) addresses the protection of works and the rights of the author, such as adaptations and reproduction of an original work.

Article 2 of the InfoSoc Directive states that Member States must ensure that authors have the sole authority to allow or prevent directly or indirectly, permanently or temporarily copying of their works. This applies regardless of whether the entire or just part of the work is reproduced. Article 3 of the InfoSoc Directive, states that the exclusive right to allow or prevent any public communication must be provided by the author of the copyrighted work. Article 4 of the InfoSoc Directive regulates the author's distribution rights and states that the author has the exclusive right to allow or prohibit any form of public distribution of either original work or copies.

It is important to note the difference between an adaption which is the modified version of an existing work and the creation of a new, independent work inspired by something pre-existing. As above the term "derivative work" is not precisely defined by EU copyright law, but the term is included in the legal framework of some Member States, which might give further guidance in answering the research questions.

⁹⁵ Mazziotti G., (2008), page 58.

⁹⁶ Mazziotti G., (2008), page 59.

3.4.1 Derivative works in Member States Jurisdictions

According to Swedish legislation, copyright protection may be applied to computer programs that are adaptations of earlier programs.⁹⁷ An adaptation is the result of processing already existing works and transfers of such to another form of art. It is important to note that copyright protection is limited in that it concerns the structure of the code itself and not the program's functions.⁹⁸ While these adaptations are granted independent protection, they still rely on the original work's copyright for each contribution. This implies that any use or exploitation of the adapted work requires permission from the original author or their rights holders.⁹⁹

Derivative works are included in the economic copyrights according to Bulgarian legislation, under "adaptions" consisting of adjustments and alterations as well as creating a new work.¹⁰⁰ Estonian copyright legal framework, namely 4§ of the Copyright Act includes "derivative works" and defines the term as "translations, adaptations of original works, modifications (arrangements) and other alterations of works."¹⁰¹

In the Italian legal framework, Article 12 of Law No 633 of 1941 for the Protection of Copyright and Neighbouring Rights, it is stated that authors have the exclusive rights to reproduction and publication of their works, as well as exploiting the work in any way, both as an original or derivative.¹⁰² Furthermore Article 7 of the above mentioned law applies that "a derivative work shall be deemed the author of that work within the limits of his own effort".¹⁰³

Lithuanian legal copyright framework, namely Article 4(3), Law of 18 May 1999 No VIII1185 on Copyright and Related Rights, applies that derivative work are "works created on the basis of other literary, scientific or artistic works, such as translations, adaptations, reviews; collections of works or compilations of data; unofficial translations of legal acts, official documents of administrative, legal or regulative nature".¹⁰⁴

In the Romanian copyright legal framework, Article 13 of Ordinary Law No 8 of 14 March 1996 (updated) on Copyright and Neighbouring Rights states that the work's author must permit the making of derivative works.¹⁰⁵

⁹⁷ Lag (1960:729) om upphovsrätt till litterära och konstnärliga verk, 4§ and SOU 1985:51 page 90.

⁹⁸ Wolk, S., (2016), page 48.

⁹⁹ Levin M, *Lärobok I Immaterialrätt: Upphovsrätt, Patenträtt, Mönsterrätt, Känneteckensrätt i Sverige, EU Och Internationellt* (11th edn, Wolters Kluwer 2017), page 78.

¹⁰⁰ European Parliament, "Copyright Law in the EU - Salient features of copyright law across EU Member States", European Parliamentary Research Service, June 2018 - PE 625.126, page 20

¹⁰¹ *Ibid.*, page 98.

¹⁰² *Ibid.*, page 215.

¹⁰³ *Ibid.*, page 216.

¹⁰⁴ *Ibid.*, page 242.

¹⁰⁵ *Ibid.*, page 315.

3.5 Summary and conclusions

In conclusion, according to Article 1 of the Computer Programs Directive copyright protection is given for computer programs, under the condition that the program is the expression of the author's own intellectual creation. The criteria of originality must be fulfilled, which reflects the author's intellectual creativity and personality, by making free and creative choices during the creation process. Originality can be expressed in many ways and at different times in the works production period, for example in the preparation phase, in the creation phase, and the selection phase by adding a "personal touch".¹⁰⁶ Account shall be taken to the specific arrangement or configuration of all the components that form part of the work.¹⁰⁷ The components necessary for the technical function shall not be considered, due to the inseparability of the idea and the expression.¹⁰⁸¹⁰⁹ In the case of software, the syntax, keywords, options, commands, and combination of commands as well as iterations consisting of figures, mathematical concepts and figures are not intellectual creations.¹¹⁰

The Court of Justice of the European Union has consistently ruled that only the expression of ideas, not the ideas themselves, can be protected by copyright. This said, it can be concluded from the case law that it is not an infringement of copyright if another developer creates a computer program that does the same task or fulfils the same function as another developer's code if the codes are different and original as in expressing each author's intellectual creativity.

Authors have exclusive rights to reproduce, translate, adapt, and distribute their software. These rights are crucial for protecting the economic and moral interests of creators. The requirement for authors' consent before making changes to the software ensures respect for the original work while allowing for necessary technological advancements. Therefore, permission from the original author or copyright holder is needed in order to generate and adopt a derived work.

A precise definition of the term "derivative work" is not provided by EU copyright law, but it is generally understood as an original work that is partly grounded on an existing copyrighted work. Computer programs consist of a series of information instructing a device to perform in a specific form.¹¹¹As

¹⁰⁶ Case C-145/10, 'Eva-Maria Painer v. Standard VerlagsGmbH and Others', 2011, para 90-92. 8

¹⁰⁷ C-393/09, para 48.

¹⁰⁸ C-393/09, para 49.

¹⁰⁹ Case C-406/10, para 67.

¹¹⁰ Case C-406/10, para 66.

¹¹¹ Mazziotti G., *EU Digital Copyright Law and the End-User*, (1 ed. 2008), Springer Berlin, page 58.

a main rule software that consists of translated or changed source code from an original program is derivative.¹¹²

It is important to note the difference between an adaption which is the modified version of an existing work and the creation of a new, independent work inspired by something pre-existing. As above the term “derivative work” is not precisely defined by EU copyright law, but the term is included in the legal framework of some Member States.

¹¹² Deck FP & MacHugh JA, (2010) page 226.

4 Open Source Software Licensing

4.1 Introduction

A license is an agreement between the product owner and someone who wants to use the product, which gives the user specific rights to use the product in ways that would otherwise be against the law.¹¹³

The exclusive rights of a copyright owner cannot be exercised without obtaining a license. A license is necessary for the distribution of a copyrighted computer program. The GPL grants licenses for its copyrights, although not ownership of the copyright itself, through bare licenses. The difference between a bare license and a contract is that it doesn't require explicit agreement. The provisions of a general software license do not oblige the owner of the copyrighted product but only oblige the person to whom the license is granted.¹¹⁴

This chapter begins with a categorisation of open source licenses and the difference between nonreciprocal and reciprocal as well as proprietary and free or open licenses. Next, the key terms of GNU General Public License, which includes a copyleft clause and its impacts on software distribution and alteration, are assessed. Lastly, the conclusions are presented in the part of this chapter.

4.2 Licenses

There are two different categories of Open source licenses, firstly the reciprocal licenses, for example, the GNU General Public License (GPL) and secondly the nonreciprocal licenses, such as the Apache and the BSD licenses.¹¹⁵

For reciprocal licenses, it is necessary to incorporate a provision which states that the code needs to be open source in the case of relicensing. There is a similarity in incorporating provisions that restrain distribution as well as relicensing between reciprocal licenses and commercial or closed-source licenses.¹¹⁶ For example, GPL states that the distribution of a work which consists of GPLed code, must be under the GPL and that no other limitations

¹¹³ Deek FP & MacHugh JA, (2010), page 232.

¹¹⁴ Ibid., page 234.

¹¹⁵ Kavanagh P., (2004), page 297.

¹¹⁶ Ibid., page 298.

can be added for redistribution of the modified work as well as the original work.¹¹⁷

On the other hand, for nonreciprocal licenses not such a provision is needed to be incorporated.¹¹⁸ LGPL, which stands for The Lesser General Public License is a nonreciprocal adaption of the GPL.¹¹⁹ Nonreciprocal licenses provide the right to redistribute, use or make changes to the code without the need to hand out the source code.¹²⁰

“Free” or “open” software stands for software that can be distributed and altered freely. “Proprietary” or “closed source” software is the opposite of open software and means that under conditions that can be shared or modified such as a license and usually cannot be seen by others.¹²¹ Examples of open source software are the Apache Web server, the Linux operating system and the GNU C/C++ language suite. These software programs are very widespread and have been used by millions of users for a long time. They have gained a reputation for their accuracy and high levels of customer satisfaction.¹²²

The GNU General Public License is designed to safeguard perpetual freedom for its code. The GPL allows unrestricted copying and modification of the code, stipulating that both copies and derivative works must be distributed under the same license, without additional restrictions. In essence, the GPL employs copyright law in a manner contrary to traditional copyright practices; rather than restricting software distribution, it prohibits anyone, including the author, from imposing such limitations. For Stallman, this approach exceeded placing his code into the public domain, as it prevents proprietary software from exploiting GPLed code fully. The GPL thus acts as a protective shield for free software, inhibiting non-free software from fully capitalizing on GPL-licensed code.¹²³

An important question that comes up is from whom the license should be taken for the original work that is created by a chain of contributions. The answer to this question is based on whether the license is sublicensable. In the case of a sublicensable license then any distributor may grant a license to third parties for the software or its parts.¹²⁴

The CJEU, in the SAS case, recognized that license terms cannot restrict the use of software that doesn't infringe copyright. The court emphasized that

¹¹⁷ Fogel K., (2023), page 198.

¹¹⁸ Kavanagh P., (2004), page 297.

¹¹⁹ Ibid., page 299.

¹²⁰ Ibid., page 300.

¹²¹ Fogel K., (2023), page 196.

¹²² Kavanagh P., page 3.

¹²³ Fogel K., (2023), page 5.

¹²⁴ Deck FP & MacHugh JA, (2010), page 235.

allowing the functionality of software to be copyrighted would weaken technological progress and industrial development by monopolizing ideas.¹²⁵

Copyleft was made known as a free software movement, that contrary to copyright requires anyone who obtains a copy of a software source code gains the right to alter, copy or distribute both the original work and any modifications derived from it, although the same free terms must be kept if distributed. More precisely this means that the same freedoms must be given to the next person who gets the copy.¹²⁶

OSS licenses can be broadly categorized as copyleft licenses, such as GNU GPL, which require derivative code to be shared under the same terms, and permissive licenses, like Apache, BSD, and MIT, which allow for more flexibility in distribution terms, including proprietary licenses. The main distinction lies in the freedom to distribute derivative software: copyleft licenses mandate adherence to the original terms, while permissive licenses afford greater flexibility in licensing.¹²⁷

4.3 Derivative works under OSS Copyleft licenses

As a main rule software that consists of translated or changed source code from an original program is derivative.¹²⁸ According to EUIPO software licenses safeguard open access to source code and at the same time protect ownership rights. The founders of the copyleft concept adapted license terms that would secure the distribution of software and its derivatives without licensing fees and by doing so provided alternative licensing models contrary to proprietary licensing.¹²⁹

Contrary to intellectual property rights, which aim to internalize knowledge externalities, open-source licenses aim to create externalities to prevent others from exclusively benefiting from knowledge.¹³⁰

The GPL leveraged copyright terms to enforce the release of derivative software under terms that ensure access to the source code. Notably, the GPL included the so-called copyleft clause which is viral in mandating that all modifications be distributed under the same terms.¹³¹

¹²⁵ EUIPO, OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION, June 2020, page 22.

¹²⁶ Deek FP & MacHugh JA, (2010) page 253.

¹²⁷ EUIPO, OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION, June 2020, page 23-24.

¹²⁸ Deek FP & MacHugh JA, (2010), page 226.

¹²⁹ EUIPO, OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION, June 2020, page 22.

¹³⁰ EUIPO, OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION, June 2020, page 23-24.

¹³¹ Ibid., page 23-24.

4.3.1 GNU:s General Public License

According to the definitions of version three of the GNU General Public License, modifying means adapting, or copying the computer program as a whole or a part of it to an extent that requires permission from the owner. The outgoing computer program is termed the “modified version” in the GPL. Both the modified version based on the original program and the unmodified version of the program are named “covered work” in the GPL.

According to the second term of GPL, the license gives absolute permission to use the unmodified program. Suppose the licensee uses the covered work and generates an output from it, in that case, the output is only considered to be covered by the same license if it meets the condition of qualifying as a modified version of the original covered work or in other words being derivative. The second term of the GPL recognises the licensees’ rights under fair use or similar principles provided by copyright law, which may under certain circumstances allow the use of copyrighted works without the need for permission from the copyright holder.

Furthermore, the second term of GPL states some permissions and restrictions that the license offers, namely:

- The right to “make, run and propagate covered work”, which means that creating, using and sharing undistributed versions of the covered works is permitted if the terms of the GLP are followed.
- The option of “conveying covered works”, which aims at sharing the covered works with others if the reason is having them modify it exclusively for the licensee, or when using someone else’s services to run the works.
- The obligation of the licensee to have control over the covered works distribution. Therefore, when a third person is running or making the covered works on behalf of the licensee, it must be agreed not to make copies of the licensee’s copyrighted material for other reasons.
- Sublicensing is not permitted and if the licensee wants to distribute the covered work, he must do so under the terms of GPL.¹³²

The third term of the GPL specifies that covered works are not considered part of technological protection measures under Article 11 WCT or similar regulations.

According to the fourth term, when the covered work consists of the exact copies of a program’s source code, the licensee is permitted to distribute the

¹³² GNU General Public License, Version 3, 29 June 2007, < <https://www.gnu.org/licenses/gpl-3.0.html> >, assessed on 14 May 2024.

work in any format, as long as a copyright notice is displayed on each copy that shows that GPL terms apply, and all disclaimers of warranty intact as well as a copy of the GPL is given.¹³³

The fifth term of the GPL outlines the rules for distributing a modified version of a program's source code, namely:

- a. It requires a transparent indication of the changes made to the code, including the relevant date.
- b. The distributed work must openly reference that it is released under the same license as the original program, along with additional conditions specified.
- c. The entire modified work must be distributed to all other recipients under the same license and terms.
- d. Appropriate Legal Notices must be displayed, if the modified work includes interactive interfaces, except for when the original program's interfaces don't.¹³⁴

4.4 Conclusion

In conclusion "free" or "open" software licenses, such as the GNU General Public License, play a crucial role in promoting the sharing and modification of software. These licenses ensure that software is distributed with its source code accessible, fostering an environment where developers can freely use, modify, and distribute software.

The GPL allows unrestricted copying and modification of code, mandating that any distributed copies or derivative works must remain under the same license. This approach contrasts with traditional copyright practices, which typically restrict software distribution. The copyleft principle, a cornerstone of the free software movement, requires that anyone who receives a copy of the software can alter, copy, and distribute it under the same terms, ensuring that these freedoms are preserved in all subsequent distributions. More precisely this means that the same freedoms must be given to the next person who gets the copy.¹³⁵

¹³³ GNU General Public License, Version 3, 29 June 2007, < <https://www.gnu.org/licenses/gpl-3.0.html> >, assessed on 14 May 2024.

¹³⁴ Ibid., assessed on 14 May 2024.

¹³⁵ Ibid., assessed on 14 May 2024.

5 Summary and conclusions

5.1 When can a computer program that was derived from the source code of another computer program be protected by copyright under the Computer Program Directive?

A computer program derived from the source code of another program can be protected by copyright under Article 1, Computer Programs Directive if it fulfils the criteria of originality. The directive states that copyright protection is given for computer programs if the program is the expression of the author's own intellectual creation. Originality in this context refers to the expression of the author's intellectual creativity and personality, demonstrated through free and creative choices during the creation process. Originality can be expressed in many ways and at different times in the works production period, for example in the preparation phase, in the creation phase, and the selection phase by adding a "personal touch".¹³⁶ The specific arrangement or configuration of all the components that form part of the work are considered, while components necessary for the technical function are not considered. Additionally, even a small number of words or code can be eligible for copyright protection, implying that the bar for what qualifies as original may be relatively low in some cases, according to the CJEU case law.

The components that are necessary for the technical function shall not be considered, due to the inseparability of the idea and the expression.¹³⁷¹³⁸ In the case of software, the syntax, keywords, options, commands, and combination of commands as well as iterations consisting of figures, mathematical concepts and figures are not intellectual creations.¹³⁹

5.2 How can a creator of such a derivative work exercise his/her rights in relation to the rights-holder(s) of the original source code program under the Computer Program Directive?

Under the Computer Program Directive, the creator of a derivative work derived from the source code of another program can exercise their rights by obtaining permission from the original rights-holder(s) of the source code program. Article 4 of the directive provides the author's rights regarding

¹³⁶ Case C-145/10, 'Eva-Maria Painer v. Standard VerlagsGmbH and Others', 2011, para 90-92. 8

¹³⁷ C-393/09, para 49.

¹³⁸ Case C-406/10, para 67.

¹³⁹ Case C-406/10, para 66.

copyrighted software, including the exclusive rights to reproduction, translation, adaptation, and distribution to the public.

However, under certain conditions, such as those related to interoperability, actions like reproduction, adaptation, transformation, or translation of the code may be permissible without the author's permission. These actions should be necessary to independently create a new program and achieve interoperability, as stated in Recital 15 of the directive.

5.3 To what extent does the use of a copyleft open-source software license, such as GNU General Public License, impact questions (1) and (2) under EU law?

The use of a copyleft open-source software license, like the GNU General Public License, impacts questions (1) and (2) under EU law by enforcing the release of derivative software under terms that ensure access to the source code. The GPL, which is designed to safeguard freedom for software code, allows unrestricted copying and modification of the code but mandates that both copies and derivative works be distributed under the same license, without additional restrictions.¹⁴⁰ This means that derivative works derived from GPL-licensed software must also be released under the GPL, ensuring that the original rights-holder(s) maintain control over the use and distribution of derivative works. Additionally, copyleft licenses like the GPL aim to prevent proprietary software from exploiting GPL-licensed code fully, thereby promoting the principles of free and open-source software.

According to the definitions of version three of the GNU General Public License, modifying means adapting, or copying the computer program as a whole or a part of it to an extent that requires permission from the owner. The outgoing computer program is termed the “modified version” in the GPL. Both the modified version based on the original program and the unmodified version of the program are named “covered work” in the GPL.

According to the second term of GPL, the license gives absolute permission to use the unmodified program. Suppose the licensee uses the covered work and generates an output from it. In that case, the output is only considered to be covered by the same license if it meets the condition of qualifying as a modified version of the original covered work or in other words being derivative. The second term of the GPL recognises the licensees’ rights under fair use or similar principles provided by copyright law.

The rights of the author are impacted by not being able to license the derivative work, through another “proprietary” or “closed source” license.

¹⁴⁰ Fogel K, Producing Open Source Software: How to Run a Successful Free Software Project (2nd edn, MTM 2023), page 198.

The second term of the GPL strictly forbids sublicensing if the licensee wants to distribute the covered work, he must do so under the terms of the GPL. According to the fifth term of the GPL, the author when distributing a modified version of a program's source code must:

- indicate transparently the changes made to the code,
- distribute the work with an open reference that it is released under the same license as the original program, along with additional conditions specified,
- distribute the entire modified work to all other recipients under the same license and terms, and
- display appropriate legal notices, if the modified work includes interactive interfaces, except for when the original program's interfaces don't.¹⁴¹

To summarise the software developer that has created a code based on another developer's copyrighted code under the GPL is restricted by having to distribute the new original derived code under the GPL. The author of the derivative work owns the copyright or his code, if it is original, as in is being an expression of the author's intellectual creativity and personality, following Article 1 of the Computer Programs Directive.

5.4 Conclusions

The Computer Program Directive gives copyright protection for computer programs that meet the criteria of originality, even if they are derived from the source code of another program. Originality is demonstrated through the expression of the author's intellectual creativity and personality, focusing on the specific arrangement or configuration of program components, which extends to derivative works, ensuring that creators can assert their rights over their creations.

The authors or creators of derivative computer programs must seek permission for the rights of the rightsholder of the original source code under the Computer Program Directive. While the directive grants exclusive rights to the author, it also allows for certain actions related to interoperability without explicit permission, however, in most cases, creators of derivative works need to obtain permission from the initial rightsholder to exercise their rights fully.

¹⁴¹ GNU General Public License, Version 3, 29 June 2007, < <https://www.gnu.org/licenses/gpl-3.0.html> >, assessed on 14 May 2024.

The use of copyleft open source software licenses, such as the GNU GPL significantly impacts copyright protection and the exercise of rights for derivative works. Copyleft licenses state that derivative works be distributed under the same terms as the original work, ensuring access to the source code and preventing proprietary exploitation, which impacts both the protection of derivative works under copyright law and the ability of creators to exercise their rights, as derivative works must adhere to the terms of the copyleft license.

As a result, the Computer Program Directive, combined with the use of copyleft licenses, promotes innovation and collaboration in software development while protecting the rights of creators, and on the other hand, it restricts the distribution rights of the author.

References

Legislation

European Union Primary Legislation

Charter of Fundamental Rights of the European Union OJ 2012 C 326/02.

European Union Secondary Legislation

Directive 2001/29/EC of the European Parliament and of the Council of 22 May 2001 on the harmonisation of certain aspects of copyright and related rights in the information society.

Directive 2006/116/EC of the European Parliament and of the Council of 12 December 2006 on the term of protection of copyright and certain related rights.

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

International Agreements

Berne Convention for the Protection of Literary and Artistic Works, of 28 September 1979.

WIPO Copyright Treaty (WCT), December 20, 1996, TRT/WCT/001.

Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS) (1994).

Sweden

Lag (1960:729) om upphovsrätt till litterära och konstnärliga verk.

Official documents

Commission of the European Communities. (1985). Completing the internal market: White paper from the Commission to the European Council (Milan, 28-29 June 1985). COM (1985) 310.

Commission of the European Communities. (1988). Green paper on copyright and the challenge of technology - Copyright issues requiring immediate action. COM (1988) 172.

European Commission. (2009). Guidance on the Commission's enforcement priorities in applying Article 82 of the EC Treaty to abusive exclusionary conduct by dominant undertakings (The Guidance Paper). 2009/C 45/0.

European Commission. (2012). Proposal for a regulation of the European Parliament and of the Council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation). COM(2012) 11 final.

European Parliament. (2018). Copyright law in the EU - Salient features of copyright law across EU Member States. European Parliamentary Research Service. PE 625.126.

European Commission. (2020). Communication to the Commission: Open source software strategy 2020-2023, Think Open. Brussels, 21.10.2020.

European Union Intellectual Property Office (EUIPO). (2020). Open-source software in the European Union. ISBN 978-92-9156-276-3.

Sweden

Prop.1988/89:85 om upphovsrätt och datorer.

SOU 1985:51 Upphovsrätt.

Literature

Monographs

Deek, Fadi. P., & MacHugh, James. A. (2010). Open Source: Technology and Policy. Cambridge University Press. Retrieved from <https://www.cambridge.org/core/books/open-source/16BC3434DD89136889E7A1F6FD1AC686#fndtn-information>.

Fogel, Karl. (2023). Producing Open Source Software: How to Run a Successful Free Software Project (2nd ed.). MTM.

Hettne, Jörgen., & Ida Otken Eriksson, I. O. (2011). EU-rättslig metod, teori och genomslag i svensk rättstillämpning (2nd ed.). Norstedts Juridik.

Kavanagh, Paul. (2004). Open Source Software: Implementation and Management (Software Development) (1st ed.). Digital Press. Retrieved from <https://www.sciencedirect.com/book/9781555583200/open-source-software#book-info> (Accessed 12 April 2024).

Lenaerts, K., & Gutiérrez-Fons, J. A. (2014). To Say What the Law of the EU is: Methods of Interpretation and the European Court of Justice. In *Columbia Journal of European Law* (Vol. 20, Issue 2, pp. 3–61). Parker School of Foreign and Comparative Law.

Levin, Marianne. (2017). Lärobok I Immaterialrätt: Upphovsrätt, Patenträtt, Mönsterrätt, Känneteckensrätt i Sverige, EU Och Internationellt (11th ed.). Wolters Kluwer.

- Mazziotti Giuseppe, (2008) EU Digital Copyright Law and the End-User, (1 edition), Springer Berlin.
- Peczenik, Aleksander. (1995). Juridikens Teori Och Metod: En Introduktion till Allmän Rättslära (1st ed.). Fritzes.
- Pila, Justine., & Torremans, Paul. (2019). European Intellectual Property Law (2nd ed.). Oxford University Press.
- Wolk, Sanna. (2016). Datorprogramalster I Upphovsrätten: Skyddet För Datorprogram, Datorspel, Bildskärmsuttryck, Design, Filformat Och Algoritmer, M.M (1st ed.). Iustus.

Case law

Court of Justice of the European Union

Case C-26/62 NV Algemene Transporten Expeditie Onderneming van Gen en Loos v Nederlandse Administratie der Belastingen (1963) ECR, ECLI:EU:C:1963:1.

Case C-6/64 Costa v ENEL (1964) ECR I-1141, ECLI:EU:C:1964:66.

Case C-11/70 Internationale Handelsgesellschaft mbH v Einfuhr- und Vorratsstelle für Getreide und Futtermittel (1970) ECR 1125 (Solange I), ECLI:EU:C:1970:114.

Case C-5/08, Judgment of the Court (Fourth Chamber) of 16 July 2009. Infopaq International A/S v Danske Dagblades Forening, ECLI identifier: ECLI:EU:C:2009:465

C-393/09 Judgment of the Court (Third Chamber) of 22 December 2010. Bezpečnostní softwarová asociace - Svaz softwarové ochrany v Ministerstvo kultury, ECLI identifier: ECLI:EU:C:2010:816.

Case C-145/10, Judgment of the Court (Third Chamber) of 1 December 2011, Eva-Maria Painer v Standard VerlagsGmbH and Others, ECLI:EU: C:2011:798.

Case C-406/10, Judgment of the Court (Grand Chamber), 2 May 2012. SAS Institute Inc. v World Programming Ltd, ECLI identifier: ECLI:EU: C:2012:259.

Case C-355/12, Judgment of the Court (Fourth Chamber), 23 January 2014. Nintendo Co. Ltd and Others v PC Box Srl and 9Net Srl, ECLI identifier: ECLI:EU: C:2014:25.

Case C-300/21, Judgment of 4 May 2023, Österreichische Post (non-material damage in connection with the processing of personal data), EU: C:2023:370.