



**MONTCLAIR STATE**  
UNIVERSITY

Montclair State University  
**Montclair State University Digital  
Commons**

---

Theses, Dissertations and Culminating Projects

---

5-2021

## Code Ownership : Plagiarism and Use

Alexis Nicole Amore  
*Montclair State University*

Follow this and additional works at: <https://digitalcommons.montclair.edu/etd>



Part of the [Intellectual Property Law Commons](#)

---

### Recommended Citation

Amore, Alexis Nicole, "Code Ownership : Plagiarism and Use" (2021). *Theses, Dissertations and Culminating Projects*. 724.

<https://digitalcommons.montclair.edu/etd/724>

This Thesis is brought to you for free and open access by Montclair State University Digital Commons. It has been accepted for inclusion in Theses, Dissertations and Culminating Projects by an authorized administrator of Montclair State University Digital Commons. For more information, please contact [digitalcommons@montclair.edu](mailto:digitalcommons@montclair.edu).

## Abstract

Technology is moving at unmeasurable rates to that of law. Ownership rights and legality become harder to grasp distant theories. With community code-sharing and limiting language structures, when does code become plagiarized or entity-owned? The disciplines of Cyberlaw and computer science are used to provide a better understanding.

The Cyberlaw discipline explores how jurisdiction views cyberspace, source code, and source code's placement within legislation. Due to cyberspace's ever-evolving nature, litigation struggles to encompass the possibilities within it. Computer science delves into theory-based excursions that define the law's shape in the cyber realm. It bolsters the possibility of implementing progressive legislation that could produce a more flexible, structured law.

This research involves theories for legislation around source code ownership, record-keeping databases, and a reformed legal system with an understanding of technology. This work aspires to discover new ways of ownership rights and inquire about ownership when there is a limiting language structure with which to work. Flexibility is imperative for the law to function in cyberspace which is rather new to a legal system in this sector. The research serves as a gateway to a newly reformed cyberlaw system in terms of ownership regarding source code and limiting computer language structures.

*Keywords:* technology, cyberlaw, code, source code, source code ownership, code ownership, intellectual property, patent, copyright law, software development.

MONTCLAIR STATE UNIVERSITY

Code Ownership: Plagiarism and Use

by

Alexis Amore

A Master's Thesis Submitted to the Faculty of

Montclair State University

In Partial Fulfillment of the Requirements

For the Degree of

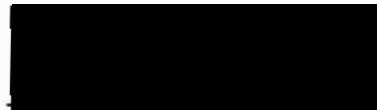
Master of Arts

May 2021

College of Humanities and Social Sciences

Department of Political Science and Law

Thesis Committee:



Dr. Ian Drake

Thesis Sponsor



Dr. Avram Segall

Committee Member



Dr. Ariel Alvarez

Committee Member

CODE OWNERSHIP: PLAGIARISM AND USE

A THESIS

Submitted in partial fulfillment of the requirements

For the degree of Master of Arts

by

Alexis Nicole Amore

Montclair State University

Montclair, NJ

2021

Copyright © 2021 by Alexis Amore. All rights reserved.

## Acknowledgements

Throughout the writing of this master's thesis, I have received a great deal of support and assistance.

I would like to thank my Thesis Sponsor, Dr. Ian Drake, as well as my Thesis Committee, Dr. Avram Segall and Dr. Ariel Alvarez. I could not have asked for a more helpful group of intelligent legal scholars who have aided me in bringing my research to life during such a difficult time.

I would also like to thank my grandmother, Jacqueline Hatch, for her never-ending support, pep-talks, and the confidence along with the drive for education that she has instilled in me since a young age alongside my grandfather, Norman Hatch. As well as them, thank you to my family.

In addition, I would like to thank Sarah Roman for assisting and guiding me through my higher education and for all the papers she has had to proofread.

## Contents

<b>1. INTRODUCTION – A LOOK INTO CODE AND ITS IMPORTANCE .....</b>	<b>6</b>
<b>1.1 AN EXAMPLE OF CONGRESS’ UNDERSTANDING OF THE INTERNET .....</b>	<b>8</b>
<b>2. CURRENT LAW ON TECHNOLOGY AND CODE .....</b>	<b>10</b>
<b>2.1 INTELLECTUAL PROPERTY RIGHTS OVERVIEW .....</b>	<b>11</b>
<b>2.2 COPYRIGHT LAW AND SOFTWARE DEVELOPMENT: A CLOSER LOOK.....</b>	<b>16</b>
<b>2.3 SOURCE CODE OWNERSHIP – A CLOSER LOOK .....</b>	<b>19</b>
<b>2.4 UNITED STATES V. ALEJNIKOV, 676 F.3D 71 (2012).....</b>	<b>22</b>
<b>2.5. TECHNOLOGY ORGANIZATIONS AND THEIR ROLE IN CODE OWNERSHIP .....</b>	<b>24</b>
<i>United States Digital Service.....</i>	<i>24</i>
<i>National Institute of Standards and Technology.....</i>	<i>25</i>
<i>National Technical Information Service.....</i>	<i>26</i>
<i>Intelligence Advanced Research Projects Activity.....</i>	<i>27</i>
<b>3. WHAT THE PREVIOUS TECHNOLOGY ORGANIZATIONS DISCUSSED LACK .....</b>	<b>28</b>
<b>4. ADVANCEMENTS IN SOURCE CODE OWNERSHIP DETECTION.....</b>	<b>28</b>
<b>5. FUTURE INNOVATION ANALYSIS .....</b>	<b>31</b>
<b>6. CONCLUSION.....</b>	<b>36</b>
<b>BIBLIOGRAPHY .....</b>	<b>38</b>

## 1. Introduction – A Look into Code and Its Importance

In an era that has become mostly virtual, especially during the COVID-19 pandemic, it is clear that society exists in both a physical and virtual realm. In terms of the virtual realm, also known as cyberspace, there are many components which make the existence feasible. In order for a computer to work, it must receive and process data based off of the instructions given to it. These instructions, aforementioned, are sequences written in a certain computer language. After the information and data have been processed, the data is sent to an output device like a monitor, cellphone screen, or the answer emitted from the Amazon Alexa in your home. The access individuals have to their devices is due to the sequencing of binary, a numeric coding system, that allows output one can physically see and work with. In terms of accessibility, devices can connect to each other through means of the Internet, also known as cyberspace.

As best put by James Grimmelmann in the book, Internet Law: Cases & Problems, the word “cyberspace” was derived out of the idea that the cyber realm is a brand-new world and should be treated as such.<sup>1</sup> Without code, cyberspace, the Internet, and all of its uses would be non-existent due to the fact code builds the technological universe. The Merriam-Webster dictionary defines code as, “Instructions for a computer (as within a piece of software)”.<sup>2</sup> The reason that code must be understood, alongside accessibility, is that everything reachable on the Internet was and is owned by someone. With how infinite, immediate, and accessible things are on the Internet, individuals are collaborating endlessly in order to continually update and transform technology. As mentioned, to make anything on a device possible, code must be written in order

---

<sup>1</sup> 9 James Grimmelmann Professor of Law, Internet Law: Cases & Problems 53 (9th ed. 2019).

<sup>2</sup> “Code.” (2019). In: Merriam-Webster Online Dictionary. 2019. Available at: <https://www.merriam-webster.com/dictionary/code> (24 Nov. 2019).



to produce the app, program, website, etc. The basic definition for a programming language (code) is as follows:

...a set of commands, instructions, and other syntax use to create a software program. Languages that programmers use to write code are called ‘high-level languages.’ This code can be compiled into a ‘low-level language,’ which is recognized directly by the computer hardware.<sup>3</sup>

With the understanding of what code and programming languages do, it is clear that jurisdiction is imperative to have a functioning cyberspace where individuals work, communicate, and conduct business. The ownership of code must be addressed and established to ensure ownership rights within cyberspace. Ownership of code originates with source code<sup>4</sup> that a programmer makes for the first time of their own creation. With source code, functionality on devices becomes possible. A programmer’s work must be recognized and protected for security, ownership, and data collection.

This topic is imperative to discuss, especially in terms of ownership and politics, because the longer the law remains behind technology itself, it is susceptible to continual error. In most recent events, the COVID-19 pandemic has surged and increased the reliance upon virtual interaction. Individuals on a global spectrum have become homebound, re-adjusted to virtual learning, and remote working more so than ever before. Even though the outcome of the current pandemic is uncertain, with more reliance on virtual interaction, our current regulatory framework is less than ideal, as well as, not sustainable for the future. A focus on legislation around technology will have to be prioritized. In time, it is likely that more individuals/entities will push for

---

<sup>3</sup> “Programming Language.” (2019). In: Tech Terms Online Dictionary. Available at: [https://techterms.com/definition/programming\\_language](https://techterms.com/definition/programming_language) (24 Nov. 2019).

<sup>4</sup> Source Code is defined as, “... the set of instructions and statements written by a programmer using a computer programming language. This code is later translated into machine language by a compiler. The translated code is referred to as object code.” “Source Code.” (2019). In: techopedia Online Dictionary. Available at: <https://www.techopedia.com/definition/547/source-code> (24 Nov. 2019).

compensation, usage, and ownership rights in regard to their programming. Glimmers of this have already been discussed in terms of virtual ownership in video games, virtual platforms, and creative virtual workspaces. This thesis explores the criteria for code ownership, including the rights of programmers, the debates amongst scholars regarding competing theories and justifications for ownership rights, and the effects upon further innovation.

### 1.1 An Example of Congress' Understanding of the Internet

In recent events, a perfect example exists to show Congress' understanding of the Internet. On April 10, 2018, a Congressional hearing was held for the testimony of Mark Zuckerberg, the creator and owner of Facebook. This hearing took place due to the events of the disinformation campaign of the 2016 United States presidential election which heavily involved Facebook and Cambridge Analytica in regard to Russian astroturfing. For background, political astroturfing is a coordinated disinformation campaign in which participants pretend to be ordinary citizens acting independently. This has the potential to influence electoral outcomes and other political behavior.<sup>5</sup> In 2018, it became public knowledge that millions of Facebook users' data had been mined and harvested without the users' consent.<sup>6</sup> Cambridge Analytica was to blame with its partnership Cambridge researcher, Aleksandr Kogan, who harvested data from millions off of Facebook. Micro-targeting was pioneered by data scientists at Cambridge Analytica. Cambridge Analytica worked to develop "psychographic" profiles for every voter in the United States and began experimenting with the information to produce paranoia, bigotry, and certain personality traits.<sup>7</sup> As a result, Cambridge Analytica had a database of millions of United States voters which then

---

<sup>5</sup> Franziska B. Keller et al., Political Astroturfing on Twitter: How to Coordinate a Disinformation Campaign. *POLITICAL COMMUNICATION*. Available at: <https://doi.org/10.1080/10584609.2019.1661888> (2019).

<sup>6</sup> Ikhlaz ur Rehman, Facebook-Cambridge Analytica data harvesting: What you need to know. *Library Philosophy and Practice (e-journal)*, 1-10 (2019).

<sup>7</sup> McKay Coppins, The 2020 Disinformation War. *Atlantic*, 325(2), 28–39 (2020).

had its own algorithm to scan and identify political beliefs as well as personality characteristics.<sup>8</sup> With all the information that Cambridge Analytica collected, they could then decide who to target and craft their political messages in a way that would attract a user which would influence their actions, thoughts, and beliefs. This is also known, as mentioned before, “micro-targeting”. This data that Cambridge Analytica and Facebook collected from these various Facebook accounts alone played a major role in President Trump’s campaign against Hillary Clinton in the 2016 presidential election. While the Internet itself is technically not a threat to the political atmosphere, it is the companies, agencies, and data hungry political parties that make the Internet a vulnerable place. The Cambridge Analytica issue presents hard proof that the Internet had a hand in changing the way politics functioned in previous years. If the Internet were not such a powerhouse of communication and use in users’ everyday lives, many theorists that were previously mentioned believe that President Trump would not have won the 2016 election.

With the background information the Cambridge Analytica incident, the Congressional hearing and testimony of Mark Zuckerberg is what defines the issue between law and technology. During the testimony, Congress asked Zuckerberg simplistic questions about the application WhatsApp, a free instant messaging service, Twitter, Facebook, email, etc. The point being the questions were not targeted towards the issue at hand due to the fact that most members of Congress did not understand the simplistic applications most Americans use to connect on a global scale. This is important to highlight because if Congress cannot understand how emailing works, basic social media platforms, instant messaging systems, etc., legislators certainly have less understanding in the mechanics and micro workings that make these platforms and devices

---

<sup>8</sup> Id.

possible. This is alarming because Congress cannot adequately make legislation and jurisdiction for code ownership, cyberspace, the Internet, devices, etc., with minimal understanding.

## 2. Current Law on Technology and Code

Due to the Internet, two different realities exist which is that of physical and virtual.<sup>9</sup> Cyberspace cannot be governed the way physical places are because individuals, interactions, countries, and businesses are everywhere and nowhere all at once. Continually, lawyers apply legal rule to facts to then apply the law to an understanding of reality. The physical network can bring people to virtual places to carry out deeds, business, free time, and in that sense, “[W]hen we need to apply law to the act of visiting a website, we can apply that law to two different sets of facts, which can produce two different outcomes”<sup>10</sup>. In these two forms of reality, our jurisdiction is far behind in governing something so vast and legislature certainly does recognize that it is behind. The Internet itself has become so colossal that companies, banks, and legislators are worried that we will see a giant transfer in economic wealth.<sup>11</sup> This is mentioned because it implies that most currency interactions will become international and virtual. A whole new level of security is needed in order to move currency immediately from anywhere in the world. Legislation and jurisdiction are needed for cyberspace to run smoothly because democracy imperative. Otherwise, this utopian world will crash and bring many aspects of government and people down with it.<sup>12</sup> This section will discuss current legislation on technology and code in order to understand criteria, ownership rights, and future innovation.

---

<sup>9</sup> James Grimmelmann Professor of Law, Internet Law: Cases & Problems 60 (9th ed. 2019).

<sup>10</sup> Id. at 57.

<sup>11</sup> Id. at 566.

<sup>12</sup> Id. at 61.

## 2.1 Intellectual Property Rights Overview

In terms of software development and ownership, there are four types of intellectual property (IP) rights that are relevant to software development which include: copyright, trademark, trade secret, and patents. For a brief overview, copyright is in charge of protecting work of authorship such as code itself, data planning, charts, etc. Trademarks, in terms of software, help to protect brands such as the visual identity of a platform, software title, or logo. Trademarks do not necessarily protect code but rather protect the way that it is branded. Trade secrets protect secret information that should not be public knowledge such as business models in relation to code, monetization plans, and so on. Lastly, patents protect the function of the code such as configurations and framework.

As simply put by Todavhich, it is imperative to make sure that intellectual property rights, business ideas, source code, documentation, and content remain under sole ownership of a client when outsourcing software development.<sup>13</sup> Todavhich goes on to say, in terms of outsourcing on software development, that there are many harmful factors that may occur when the management of IP rights are neglected such as: harming the business, it can be costly, it might hurt the clients customers, and it might harm collaboration with investors.<sup>14</sup> For example, vendor lock could happen, changes in projects may require high costs, a service provider may make clients dependent on them, and for investors, all IP including source code should be owned by the client.<sup>15</sup> The four types of IP previously mentioned, trademark being the least of concern, has different types of legal protection as described. When working for a company, it is important

---

<sup>13</sup> Moqod, WHO OWNS YOUR APP? HOW TO AVOID IP RIGHTS TRAPS IN SOFTWARE DEVELOPMENT MEDIUM, Available at: <https://moqod-software.medium.com/who-owns-your-app-how-to-avoid-ip-rights-traps-in-software-development-b12304028fdb> (2019).

<sup>14</sup> Ibid.

<sup>15</sup> Ibid.

to note that everyone who is involved in a certain project can overstep IP.<sup>16</sup> Employees, while under employment, have the right to own the IP being developed, but usually they are restricted by their own employment contract.<sup>17</sup> What usually occurs is that anything created at the time of employment and compensation is owned by the company once an employee leaves or changes roles. Employees certainly cannot personally profit off of IP created under employment, but they also cannot take it with them when they go. Continuing on, consultants and individual contracts, unless written in contract, may own the IP that they create.<sup>18</sup> Lastly, vendor companies that develop software, even though the client paid, it does not mean the client owns the IP that the vendor company creates unless it is contracted.<sup>19</sup>

Types of code that can be encountered when dealing with IP rights are existing/third party-code, open-source code, and unique code.<sup>20</sup> To elaborate, existing/third-party code are already used by a certain developing company in other projects and a vendor may use the code that they have created or already used for another client to develop the software.<sup>21</sup> In most cases, a vendor is not interested in giving up ownership of this type of code because they might want to use it for future projects with other clients.<sup>22</sup> A way to manage this and avoid problematic issues, is to pay for a license for the code that is compatible with a clients project in order to protect the IP rights.<sup>23</sup>

Another type of code, and one of the main topics of this thesis, is open-source code. As described before, it is publicly available for use and is a main component for development that

---

<sup>16</sup> Ibid.

<sup>17</sup> Ibid.

<sup>18</sup> Ibid.

<sup>19</sup> Ibid.

<sup>20</sup> Ibid.

<sup>21</sup> Ibid.

<sup>22</sup> Ibid.

<sup>23</sup> Ibid.

can be modified and shared. In continuation with out-sourcing software development as an example to show IP rights, again it should be noted that open-source code cannot be owned by a “client” in this example or maintain exclusive control over it. It is imperative for the client out-sourcing software development to know what open-source code was exactly used in their project because the client should be sure that their developers can use it and that compliance requirements are met.<sup>24</sup> When using open-source code, clients could face restrictions that they should be aware of such as violations of open-source licenses which can tank a project.<sup>25</sup> Also, under a General Public License, also known as a free software license, one must publish the source code of the complete project since open-source code was used.<sup>26</sup> Other types of agreements include BSD licenses, MIT licenses, and APACHE licenses.

Lastly, in terms of types of code, is unique code. Unique code is considered a type of code that was specifically made for a particular business and at times can be of limited use to the developer.<sup>27</sup> In conjunction, with this type of code one might be restricted by license terms related to existing code IP and third-party software IP that might be entangled in the end product.<sup>28</sup>

To continue on the IP conversation, there are a few more agreements to discuss in regard to code, source code, and IP to further knowledge before future innovation can be discussed. First to discuss is end-user license agreements. Of course, copyrighted software is the way to go currently in terms of protection of IP rights related to code, but an end-user license agreement can do a good job at protecting ownership as well. An end-user license agreement is a way to

---

<sup>24</sup> Ibid.

<sup>25</sup> Ibid.

<sup>26</sup> Ibid.

<sup>27</sup> Ibid.

<sup>28</sup> Ibid.

prohibit the sale or transfer of software by the end user.<sup>29</sup> This agreement requires the user to purchase separate licenses for each user or computer and at the same time limit liability for damages if the software proves to be faulty.<sup>30</sup> End-user license agreements come in two forms which are click-wrap agreements or shrink-wrap agreements. Click-wrap agreements or ‘click-through’) agreements, require users to click an “I agree” box after being presented with a list of terms and conditions of use, or “browsewrap” agreements, which generally post terms and conditions on a website via a hyper- link at the bottom of the screen.<sup>31</sup> This can be enticing and dangerous due to the fact that many users do not read the terms and conditions before entering a program, application, video game, etc. In conjunction, similar yet different, shrink-wrap agreements are legal agreements printed and usually put in an actual software package before using it.

In terms of selling software to another company, one must legally transfer the copyright. A copyright assignment agreement documents that a certain copyright has been assigned to another party and lets you retain specific rights that you specify.<sup>32</sup> These rights can include trade secrets and information on the technology being transferred. This agreement protects you from future liability related to the software and releases one from any responsibility for training or instruction regarding the technologies use.<sup>33</sup> In terms of copyright, the question stands on how one copyrights work if software is built for someone else. If one is paid to create software for an individual/company, they are under an arrangement known as “work for hire”<sup>34</sup> and there are

---

<sup>29</sup> TechInsurance, KNOW YOUR INTELLECTUAL PROPERTY RIGHTS ON WORK-FOR-HIRE SOFTWARE TECHINSURANCE, <https://www.techinsurance.com/resources/intellectual-property-rights-software> (last visited February 13, 2021).

<sup>30</sup> Ibid.

<sup>31</sup> James Grimmelmann Professor of Law, Internet Law: Cases & Problems 53 (9th ed. 2019).

<sup>32</sup> TechInsurance, KNOW YOUR INTELLECTUAL PROPERTY RIGHTS ON WORK-FOR-HIRE SOFTWARE TECHINSURANCE, <https://www.techinsurance.com/resources/intellectual-property-rights-software> (last visited February 13, 2021).

<sup>33</sup> Ibid.

<sup>34</sup> In order to understand what is considered work for hire, please see the U.S. Copyright Act of 1976, 17 U.S.C. § 101.



different IP laws that apply.<sup>35</sup> It is an exception to the recognized standard that a person who creates a work is the owner of said work. According to copyright law, if one is hired by another to create software, the individual/business is the legal author of that software.<sup>36</sup> Lastly, if the one wants to modify the software that a client already owns, it is preferred that a software customization agreement which defines the scope of the work that you will do in modification for the software.

In terms of IP rights and different scenarios, it is important to tie together the aforementioned into an understandable scenario. According to Sagastume, a multinational corporation would most likely start developing its own source code and software from scratch, while a smaller business would take advantage of open-source code that already exists.<sup>37</sup> With the possibility of buying source code from a platform or using free open-source code, it is imperative to clarify what legal rights exist over the purchased or developed programming code.<sup>38</sup> Legal experts experience difficulties defining software IP rights under current legal categories due to the fact that the various components of the software make it fall under different legal categories.<sup>39</sup> As said by Sagastume, the World Intellectual Property Organization Copyright Treaty, the World Trade Organizations TRIPS Agreement, and the European Council Directive 91/250/EEC on the Legal Protection of Computer Programs assume that code is a literary work that is subject to copyright protection.<sup>40</sup>

---

<sup>35</sup> TechInsurance, KNOW YOUR INTELLECTUAL PROPERTY RIGHTS ON WORK-FOR-HIRE SOFTWARE TECHINSURANCE, <https://www.techinsurance.com/resources/intellectual-property-rights-software> (last visited February 13, 2021).

<sup>36</sup> Ibid.

<sup>37</sup> Jorge Sagastume, CUSTOM BUILT SOFTWARE AND THE IP LAW - WHAT YOU NEED TO KNOW INTELLECTUAL PROPERTY WATCH (2018), <https://www.ip-watch.org/2017/11/07/custom-built-software-ip-law-need-know/> (last visited Feb 2, 2021).

<sup>38</sup> Ibid.

<sup>39</sup> Ibid.

<sup>40</sup> Ibid.

It is important to note that with source code, you can have two distinctive source code that result in the same functionality.<sup>41</sup> In simpler terms, there can be two software programs with unique source code, but they produce the same result and same functionality. With that in mind, different legislations, on federal, state, and international level have contradicting interpretations on whether copying functional elements of computer software is an infringement on copyright.<sup>42</sup> Sagastume gives a great example by explain to imagine dozens of project management apps that provide the same functionality and are effectively all the same. As a result, it is hard to define whether one software infringes on the copyright of another from a legal point.<sup>43</sup> With that in mind, it is important to note that international legislation does not allow patenting of general ideas or objects. Copyright as a literary work does not fully protect software and many patent their products to encompass more.<sup>44</sup>

## 2.2 Copyright Law and Software Development: A Closer Look

One way ownership is currently regulated is through copyright law and the discussion around it should be expanded for further understanding. Here are some rights that are applicable to software code provided by Chris Shiplett and Erik M. Pelton:

- I. The right to produce the code;
- II. The right to create “derivative works” based on the code, such as the screen display that the code degenerates, future versions of the software, or other software programs into which the code is integrated;
- III. The right to distribute copies of the code;
- IV. The right to “display” the code, for example by posting to a website (17 U.S.C. § 106).

Copyright law has an underwhelming support system from scholars. As Shiplett and Pelton explain,

---

<sup>41</sup> Ibid.

<sup>42</sup> Ibid.

<sup>43</sup> Ibid.

<sup>44</sup> Ibid.

“Applying the basic law of copyright to software development, if you personally write a class or module, you own the copyright to that class or module. If you write a website in html, or a website display script in a scripting language like PHP or ASP.NET, you own the copyright to those lines of code you wrote. You are free to re-use that code in any way you like, and no other person or entity can legally use that code without your permission.”<sup>45</sup>

The rule explained above is subjected to several exceptions in regard to copyright ownership which include:

1. The work made for hire rule – doctrine that generally defines the relationship between a software developer and client;<sup>46</sup>
2. The License or Assignment clause in a development contract – a grant of permission to use the code without giving up ownership of code;<sup>47</sup>
  1. A contract clause is usually created in both the work made for hire rule and the license or assignment clause;
3. The situation encountered when developing on an “open source” platform;<sup>48</sup>
  1. Development is impossible without encountering some form of open-source code either as a platform to develop on or a component of the work.

It should be noted that re-usable code is a major component to developers and programmers.

Creating and owning re-able code/source code is a critical step in the process of creating profitable software. Under copyright law, the author of a line of code is the owner of the copyright in that code. The key to understanding the implications of source code software on development, best described by Shiplett and Pelton, is the understanding that while open-source software is free, it is not in the ‘public domain’.<sup>49</sup> As the aforementioned authors continue to explain, open-source software, is in fact, copyrighted software where the proper use is mandated through a particular terms of license.<sup>50</sup> Continually, most important for developers, derivative

---

<sup>45</sup> Association of Software Professionals, *Who Owns The Code?* Available at: [https://asp-software.org/www/misv\\_resources/business-articles/who-owns-the-code/](https://asp-software.org/www/misv_resources/business-articles/who-owns-the-code/) (2008).

<sup>46</sup> Ibid.

<sup>47</sup> Ibid.

<sup>48</sup> Ibid.

<sup>49</sup> Ibid.

<sup>50</sup> Ibid.

software that is based upon open-source software must and has loosely conform to the original terms of the open-source license agreement. Although, software written to be on an open-source platform does not have to be.<sup>51</sup>

With the understanding of writing code that an individual owns, copyright law becomes difficult to enforce with such a mailable space in which intellectual property changes. The intellectual property, such as code, changes because of editing, updates, and creating adaptations for a new creation. Referencing rule number three mentioned previously, open-source code, while mentioned in copyright law, is not necessarily protected by it. In conjunction, all code stems from open-source code to create new lines of code, new programs, scripts, etc. When multiple individuals work on code, the original owner of the code becomes lost after countless revisions. This is what makes copyright law extremely difficult on the Internet, especially, regarding coding. Software and source code can be classified so narrowly that it can fall into many different categories of intellectual property, or, it can be so broadly classified that it does not fit into a category at all.<sup>52</sup>

Federal and state legislation has supported this in terms of its own shifting boundaries of intellectual property protection, but it is not quite enough for in depth nature of this problem.<sup>53</sup> Cyberspace and code that creates it has obviously birthed a nebulous atmosphere that continually morphs out of the confined constructs of law and international law. There can be minimalistic glimmers of ownership, but with the vast nature of cyberspace, ability, and programmers, computer science reiterates that law must catch up in order to maintain a balance. The future of

---

<sup>51</sup> Ibid.

<sup>52</sup> Sonia K. Katyal, The Paradox of Source Code Secrecy, 104 Cornell L. Rev. 103 (2019).

<sup>53</sup> Ibid.

cyberspace is unpredictable and computer scientists as well as theorists have shown that this is difficult for law to grasp. Open-source code is also an issue when it comes to the designation of ownership. The term “open source” must be looked at for its flexibility and should be studied beyond software.<sup>54</sup>

### 2.3 Source Code Ownership – A Closer Look

*“Software source code is a precious, unique form of knowledge. It can be readily translated into a form executable by a machine, and yet it is human readable...”<sup>55</sup>*

Programmers must be protected in order to maintain ownership rights and further development. Much like other works, code is not always owned by one entity. Programmers create and distill new material into files which can be amongst many owners.<sup>56</sup> At that point, the code should be organized and tracked for clear ownership evaluation. The owner of source code is most likely the creator and inventor of that specific code. Although, with files consisting of larger code artifacts, they usually are created and contributed to by multiple programmers over time creating multiple changes.<sup>57</sup> In this case, the programmer with the highest contribution to the code changes is defined as the owner of the code and takes responsibility for the file.<sup>58</sup> This shows that ownership is determined by the knowledge a programmer has about the code he or she is creating.<sup>59</sup> If a programmer’s knowledge or responsibility is lacking, it can reduce the quality of the code being written or modified.<sup>60</sup>

---

<sup>54</sup> Theodora Vardouli & Leah Buechley, Open Source Architecture: An Exploration of Source Code and Access in Architectural Design, 47 Leonardo 51 (2014).

<sup>55</sup> Jean-François Abramatic et al., Building the Universal Archive of Source Code: A global collaborative project for the benefit of all, 61 viewpoints 29 (2018).

<sup>56</sup> Michaela Greiler et al., Code Ownership and Software Quality: A Replication Study, 1 IEEE 2 (2015).

<sup>57</sup> Ibid.

<sup>58</sup> Ibid.

<sup>59</sup> Ibid.

<sup>60</sup> Ibid.

It is important to know and reinforce that knowledge lies within the programmer who is creating code. Each individual's knowledge, ability, and experience are slightly or majorly different than another individual. Therefore, their work must be recognized and preserved for the sake of ownership rights. With ownership also comes responsibility, which is imperative for a governable scape, cyberspace, that can maintain the qualities of society with right and wrong. If there is a lack of ownership with code, then there is a lack of responsibility that follows suit.<sup>61</sup> If there is a lack of responsibility, then there is not a responsible part to maintain and test the code itself.<sup>62</sup> With such flexibility in collaborative creation, tracking ownership is not easy in regard to code as it is with other works such as literature, music, and art. When code is modified for enhancement, updates are made, pushing new lines of code for better performance, programmers work starts to become muddle, mixed up, and altered. Weak ownership can easily occur and is the way of coding at the moment,

There are several reasons why weak ownership occurs. Some of those are because ownership is currently transferred from one person to another or from one team to another. Another can be ongoing refactoring which is performed by another team than the original team that contributed the code, or because of bug fixing. Also, crosscutting concerns or architectural smells can be a reason why several teams have to edit and work on artifacts together.<sup>63</sup>

Weak ownership not only occurs in business format, such as companies creating software or machinery, but also individuals at home. The wild west nature of the Internet comes from the fact that it is accessible to mostly everyone. Programmers creating at home by themselves must also have their work be protected when it is introduced to individuals via the Internet, applications, or devices.

---

<sup>61</sup> Ibid.

<sup>62</sup> Ibid.

<sup>63</sup> Ibid.

With such disarray in the virtual realm of code ownership, there have been many theories portrayed in order to reform the process known about code as well as ownership. Lawrence Lessig, as previously cited, provides a great theory on criteria for code creation. When discussing source-code and defining what it is, it is common to refer to it as ‘open’ and ‘closed’ source code which will suggest a critically important debate on how software should be developed.<sup>64</sup> As referenced before, the open-source software movement, which allows for endless development and innovation argues, according to Lessig and Richard Stallman, that there are fundamental values of freedom that demand software be developed as free and accessible.<sup>65</sup> The opposite of open-source software development is called proprietary software where the developer “hides the functionality of the software by distributing digital objects that are opaque about the underlying design.”<sup>66</sup>

In terms of the idea “open” and “closed” code, it clearly could change the way ownership regarding code is conducted. “Open” code can continually produce the nature of technology with its rapid movements and growth. Individuals are allowed to create and craft off of the steppingstones of other programmers’ certain lines of code. On the other hand, “closed” code can shut off the accessibility to many by putting monetary gates, subscription blockers, or other regulations one must follow before gaining access. While keeping ownership in mind and understanding how difficult it is to track due to its malleable nature, there are some suggestions into creating a database that can do all that work for individuals internationally.<sup>67</sup> Source code is shared and spread throughout a variety of platforms which has caused computer scientists and theorists to widely agree that there is no universal method to tracking source code. They believe it

---

<sup>64</sup> 2 Lawrence Lessig, Code 138 (2nd ed. 2006).

<sup>65</sup> Ibid.

<sup>66</sup> Ibid.

<sup>67</sup> Jean-François Abramatic et al., Building the Universal Archive of Source Code: A global collaborative project for the benefit of all, 61 viewpoints 29 (2018).

is their responsibility to build a shared universal catalog that solves this problem.<sup>68</sup> If there were to be a universal catalog of source code, it could narrow down and give logs of when code was created, modified, and used elsewhere. This way, individuals and entities could be compensated and credited for their creation/work. Although, to build a database so large would take a long time in creating it due to the amount of information that would have to be compiled. So, this proves to be a downside to this criteria for code creation and further innovation, but in the long run, proves majorly successful to allow legislation to catch up and settle. With the building of such a database and ownership established, theories that exist currently could change into fact with how source code is produced and how programmers/engineers get compensated for their work. For such a database to exist, “The technical challenge is to build crawlers for each code-hosting platform, as there is no common protocol available, and to develop adapters for all version-control systems and package formats.”<sup>69</sup>

#### **2.4 United States v. Aleynikov, 676 F.3d 71 (2012)**

Continuing on the path of copyright and ownership, the case United States v. Aleynikov shows the difficulty and complexity of copyright law in terms of code. In this particular case, a defendant won on appeal after he was convicted of stealing and transferring proprietary computer source code used in his employer’s high frequency trading system.<sup>70</sup> The defendant, Sergey Aleynikov, held the title of vice president for Goldman’s Equities Division for two years. He was in a group of computer programmers who dealt with developing and improving portions of the code for the company’s high frequency trading platform.<sup>71</sup> The defendant was offered a new

---

<sup>68</sup> Ibid.

<sup>69</sup> Ibid.

<sup>70</sup> United States v. Aleynikov, 676 F.3d 71 (2012).

<sup>71</sup> Ibid.



title of Executive Vice President, Platform Engineering, for triple the salary he was currently making in order to develop his own high frequency trading platform.<sup>72</sup> After the defendant had left Goldman's for the new position, he had copied thousands of lines of code from the code repository of Goldman's and uploaded them to a repository for later use.<sup>73</sup> The defendant then transferred the data to multiple computers and a flash drive.<sup>74</sup>

The legal issue in this case was whether the defendant's conduct violated the National Stolen Property Act (NSPA)<sup>75</sup> and the Economic Espionage Act of 1996<sup>76</sup>. The court ruled that the defendant's conduct did not violate these acts because open-source code was not a "stolen good" within the meaning of the NSPA. In continuation, the source code was not "related to or included in a product that is produced for or placed in interstate or foreign commerce" within the meaning of the Economic Espionage Act of 1996. The Appellate Court agreed with the appeal of the defendant and the indictment was legally insufficient. The Appellate Court then reversed the judgement of the district court.<sup>77</sup>

The type of intellectual property the defendant stole, source code, was considered a trade secret. With what the defendant was being charged with, he was being prosecuted criminally and criminal statutes were interpreted narrowly in favor of the defendant. In reference to the decision made by the court, Congress neglected to define the exact material under the act. Therefore, neither statute encompassed the material that was stolen. Just because Congress left gaps in the law due

---

<sup>72</sup> Ibid.

<sup>73</sup> Ibid.

<sup>74</sup> Ibid.

<sup>75</sup> 1312. National Stolen Property Act -- Goods, Wares, Merchandise, THE UNITED STATES DEPARTMENT OF JUSTICE (2020), Available at: <https://www.justice.gov/archives/jm/criminal-resource-manual-1312-national-stolen-property-act-goods-wares-merchandise> (Oct 10, 2020).

<sup>76</sup> 1122. Introduction to the Economic Espionage Act, THE UNITED STATES DEPARTMENT OF JUSTICE (2020), <https://www.justice.gov/archives/jm/criminal-resource-manual-1122-introduction-economic-espionage-act> (Oct 10, 2020).

<sup>77</sup> United States v. Aleynikov, 676 F.3d 71 (2012).

to not considering that type of a material, a company lost millions of dollars and their own trade secret. It is not so much the fault of the company, but rather an example on how legislation is still not efficient up against governing virtual property.

To connect the thought pattern to show that current legislation is lacking and leaving a critical part of ownership in a vulnerable state, remember Mark Zuckerberg's two-day trial with Congress. If legislators cannot envision the kind of repercussions individuals may face when creating online, there is no adequate and efficient way they can establish jurisdiction. While, of course, there are copyrights, contracts, and statutes that exist currently, they do not encompass all possibilities. This becomes important when code owners become territorial over their property and want to profit off of it. There is no current way to establish and maintain ownership since most of code creating relies on source-code to become established.

## **2.5. Technology Organizations and Their Role in Code Ownership**

Before diving into different technology departments that are within the United States, it is important to understand that a few of them will not aid in the issue of code ownership. Continually, this is to highlight the current specialized areas of technology, but how overall the crux of the issue is overlooked and neglected.

### **United States Digital Service**

The first to look into the United States Digital Service (USDS) to understand what it truly does. The USDS is a technology unit within the Executive Office of the President of the United States.<sup>78</sup> It is not a regulatory body. The function of the USDS provides consultation services to federal agencies regarding information technology which seeks to simplify access and improve

---

<sup>78</sup> UNITED STATES DIGITAL SERVICE, Our Mission, Available at: <https://www.usds.gov/mission> (2021).

federal websites.<sup>79</sup> Within the USDS a group of technologists, such as designers, engineers, product managers, and bureaucracy specialists work with civil servants in order to provide best practices and new approaches in order to modernize the government.<sup>80</sup> This is due to the fact that government websites are so difficult to navigate and are extremely confusing to users. According to the USDS itself, their objectives include:

- I. Transforming critical, public-facing services;
- II. Expanding the use of common platforms, services, and tools;
- III. Rethinking how the Government buys digital services;
- IV. Bringing top technical talent into civic service.<sup>81</sup>

With understanding the USDS's objectives, it should be noted that it was formed in 2014 – only six years ago. Undergoing a modernization of federal websites is only the steppingstone in reforming governmental lag with technology. The fact that the USDS is rather new is concerning due to the fact that the issues they resolve are rather simple compared to the complexity of code ownership.

### **National Institute of Standards and Technology**

The National Institute of Standards and Technology (NIST) is a physical sciences laboratory and non-regulatory agency of the United States Department of Commerce.<sup>82</sup> The NIST was founded in 1901 and is one of the oldest physical science laboratories. Congress established the NIST to “...remove a major challenge to U.S. industrial competitiveness at the time—a second-rate measurement infrastructure that lagged behind the capabilities of the United

---

<sup>79</sup> Ibid.

<sup>80</sup> Ibid.

<sup>81</sup> Ibid.

<sup>82</sup> NIST, About NIST (2017), Available at: <https://www.nist.gov/about-nist> (Dec 1, 2020).

Kingdom, Germany, and other economic rivals.”<sup>83</sup> Currently, the NIST measurements support both the smallest technologies to the largest complex ones.<sup>84</sup>

The current mission of the NIST is the promotion of United States innovation and industrial competitiveness by advancing multiple areas of technology measurement such as: “...science, standards, and technology in ways that enhance economic security and improve our quality of life.”<sup>85</sup> An important reason as to the discussion of the NIST is because it focuses on measuring science, conducts rigorous traceability, and development and use of standards. In conjunction with this thesis, in order for ownership to work with code and conduct criteria for creating code, there must be rigorous traceability which does not exist yet. The NIST has its own, just like other government entities, companies, and individuals, source code management across its own programs. This is to ensure rules and boundaries when dealing with critical source code which can reveal trade secrets, works-in-progress, and other forms of advancement technology. Most times, there is an open-source software (OSS) is used in companies, agencies, administrations, and so on in order to promote productivity and efficiency.

### **National Technical Information Service**

Continuing on some of the technology branches that are currently established in the United States, the National Technical Information Service (NTIS) is a great topic to discuss future innovation. The NTIS is also another agency within the United States Department of Commerce and is part of the America’s Data Agency. The purpose of the National Technical Information Service is to help federal agencies make better decisions about data, with data.<sup>86</sup> The main function of NTIS is to store, analyze, sort, and aggregate data in new ways which allows

---

<sup>83</sup> Ibid.

<sup>84</sup> Ibid.

<sup>85</sup> Ibid.

<sup>86</sup> NIST, About NIST (2017), Available at: <https://www.nist.gov/about-nist> (Dec 1, 2020).

partners to solve problems with their vast database.<sup>87</sup> This is important due to the fact that NTIS has stated they have changed to meet the demands of a data driven government. Under the provisions of the National Technical Information Act of 1988<sup>88</sup>, NTIS is authorized to establish and maintain a permanent repository of non-classified scientific, technical and engineering information in different areas of expertise.<sup>89</sup> The reason that this is brought up is for the future innovation portion of this thesis. If such a repository can be built to keep log and track of data, a similar repository can be built in order to track source code ownership to make it easier for owners, attorneys, and legislators.

### **Intelligence Advanced Research Projects Activity**

The Intelligence Advanced Research Projects Activity (IARPA) is an organization within the Office of the Director of National Intelligence which is responsible for leading certain research to overcome overwhelming intelligence difficulties.<sup>90</sup> The IARPA's mission is to do research and create on technical areas to bridge the gap between advancing technologies in all areas. The reason that this organization is brought up is because it very well take on the issue of source code ownership and help legislators narrow down the overwhelming vast nature of it all. The IARPA addresses challenges and transitions for the partners that they then work with which is a perfect steppingstone in understanding and innovating a better criteria for source code ownership.

---

<sup>87</sup> Ibid.

<sup>88</sup> 15 U.S.C. 3704b.

<sup>89</sup> Ibid.

<sup>90</sup> IARPA BE THE FUTURE, Home, Available at: <https://www.iarpa.gov/index.php/about-iarpa> (last visited Feb 15, 2021).

### 3. What the Previous Technology Organizations Discussed Lack

With only a few organizations mentioned previously, they are all innovative and progressive in terms of technology in their own aspects. Although, the crux of the internet and what breathes life into the technology that those organizations are refining, is that of code and its ownership. Copyright law does not quite fix the whole issue of code ownership due to the advancement of technology and source code. With all the information provided so far, the more code is shared online and distributed, ownership becomes blurred and hard to maintain. Eventually ownership will not be able to be tracked once technology advances. Some of these organizations mentioned could aid in the innovation of legislation and help with the criteria of code ownership.

### 4. Advancements in Source Code Ownership Detection

It is interesting to note that there is a wide discussion amongst computer science theorists on the issue of plagiarism regarding code. Many identify that there is an issue with “plagiarism” amongst computer science students within assignments, although, it does not just occur amongst students. When plagiarism is spoken about, most people think about schoolwork, stealing the words of another person’s paper, literature, etc. In regard to code, it can be thought about in the same way. Code is made up of terms and instances that give life to programs, applications, and the ability to modify everything there is to see in cyberspace and technology. Source code plagiarism according to Cosma and Joy, includes the definition of source code plagiarism as,

SOURCE-CODE plagiarism detection in programming assignments is a task many higher education academics carry out. Source-code plagiarism occurs when students reuse source-code authored by someone else, either intentionally or unintentionally, and fail to adequately acknowledge the fact that the particular source-code is not their own [1].<sup>91</sup>

---

<sup>91</sup> Georgina Cosma and Mike Joy, An Approach to Source-Code Plagiarism Detection and Investigation Using Latent Semantic Analysis, 61 IEEE Transactions on Computers 1 (2012).

It can be concluded that source code or code plagiarism does not just occur amongst computer science students. Individuals from the academia have created a system that is trying to spot plagiarism within code so that they can better control students work.<sup>92</sup> This would not only help academia, but it would help with code ownership criteria and legislation around source code ownership. With this in place, avenues of protection and compensation open up.

Not only is the idea of ownership and plagiarism important to the law, but that of responsibility as well. When one steals someone else's code, it can be used for a majority of things such as committing crimes, cyber-attacks, stealing identity, the options are endless due to the endless nature of cyberspace. With possibilities such as the ones listed previously, we often lack convincing direct evidence that would allow us to attribute responsibility for cyber-attacks in terms of international law. Even then, international actors still attribute responsibility in politic terms with political consequences.<sup>93</sup> With the reference before with the system that could detect plagiarism in source-code, something in that nature is needed in order to maintain governance in cyberspace.<sup>94</sup> Cyberspace, as one needs to remember, is international and therefore the code programmers create is also internationally available and accessible.

There is a new data-driven economy that is built off of the hard work that programmers have put in to create the "new world" we have over time.<sup>95</sup> <sup>96</sup> In terms of ownership,

---

<sup>92</sup> Ibid.

<sup>93</sup> Lucie Kadlecova, State Responsibility in the Cyber Age: The Course towards Indirect Evidence, 53 Mezinárodní Vztahy 35 (2018).

<sup>94</sup> Georgina Cosma and Mike Joy, An Approach to Source-Code Plagiarism Detection and Investigation Using Latent Semantic Analysis, 61 IEEE Transactions on Computers 1 (2012).

<sup>95</sup> Nikolas Ott and Hugo Zylberberg, A European Perspective on the Protection of Personal Data in Cyberspace: Explaining How the European Union Is Redefining Ownership and Policies of Personal Data beyond National Borders, 16 KSR 69 (2016).

<sup>96</sup> 2 Lawrence Lessig, Code, (2nd ed. 2006).

We argue that as the quantity and sensitivity of data increase, users should have more information about ‘ownership’ of the data in question. Indeed, if users generate the data and companies provide the tools to capture and analyze it, who should own that data?<sup>97</sup>

As it is shown both in the United States and internationally, there is not enough regulation in the determination of many ownership rights, privacy rights, intellectual property rights, regulation of code, etc. Both the law and computer science make the valid point of reformation and creation of the law. Although, the law does not know how to get there in time. While many areas need regulation in cyberspace, the plagiarism of source code and ownership of code has more of an effect than one thinks. Code gives individuals the ability to corner the market. As shown in, U.S. v. Microsoft, Microsoft’s actions threaten the future of business in the cyber realm. Through its actions toward Netscape, IBM, Compaq, Intel, and others, Microsoft has demonstrated that it will use its standing, market power, and immense profits to harm any company that has similar initiatives that bolster competition against Microsoft’s main products.<sup>98</sup> In lieu of Microsoft’s powerful standing and immense profits, it mirrors a firm that has transformed into a virtual monopoly. With this nature of conduct, this deters “investment in technologies and businesses that exhibit the potential to threaten Microsoft”.<sup>99</sup> While monopolies are not necessarily illegal, they can cause issues when trying to create progressive legislature for an unpredictable and international space. In terms of code, which makes all of this possible, there must be a declaration and better understanding of ownership between data, code, and individuals. While there are legal implementation and agreements at work, the current legal solutions are not enough for these cyber issues as the legal system realizes.

---

<sup>97</sup> Nikolas Ott and Hugo Zylberberg, A European Perspective on the Protection of Personal Data in Cyberspace: Explaining How the European Union Is Redefining Ownership and Policies of Personal Data beyond National Borders, 16 KSR 69 (2016).

<sup>98</sup> U.S. v. Microsoft 253 F.3d 34 (DC Cir, 2001) en banc.

<sup>99</sup> Ibid.



Cyberspace has little regulation, jurisdiction, and governance at this point in time. As technology continues to grow rapidly, it can cause frustration and lag for law to catch up. Not only does the Internet create and destroy new technologies, it causes a plethora of areas to lag behind. This encompasses code and source code entirely because cyberspace is made up of code. Imagine the Big Bang Theory in terms of the creation of our physical universe, it is not really clear how the Big Bang started which thus created the universe we have come to know on a small scale. Imagine source code and code being the *reason* for the creation of a new universe, cyberspace, which can be used over and over again to create more life. A power like that must be regulated and it seems as though the law does not retain all the answers just yet. That is why we turn to computer science for more in-depth understanding, possibilities, and theories to create a virtual jurisdiction regarding these matters.

## 5. Future Innovation Analysis

With the ideas around cyberspace, code ownership, and rapid shape-shifting nature of technology, it continually proves the need for ownership to be better defined and protected. Code can very easily be modified and taken by another entity. In this current state of understanding, plagiarism or modification in regard to source code is being taken care of in an out-of-date type of way. For example, all of academia is familiar with plagiarism and what happens when it is committed. Action is taken against the student, the work is invalid, and there are steep consequences. In the corporate world, lawsuits are the solution in order to receive justice and compensation. With code being a rather new concept for law, they do not have the exact tools to create the right justifiable path. In turn, a new structure of legal analysis and jurisdiction is need for ownership rights to continue forth in this new “space”. The idea of a single author whose work is preserved through property rights seems to be surpassed by authorship via current corporate

agents.<sup>100</sup> There must be clearer representations of flexible legal understanding and analysis that current property regimes are outdated, too specific, and saturated with politics.<sup>101</sup>

When thinking about plagiarism in regard to source code and cyberspace, one must also recognize and instill that the cyberspace/Internet is everywhere and nowhere. Almost the whole world, roughly 4.54 billion people, use the Internet. This means that ways of jurisdiction and legality must be thought about internationally on a very large shape-shifting scale. Zhang Shasha, from the Beijing Review, has interesting input about foreign affairs and the future of the Internet and ownership. He states that bridging the digital divide has become a new challenge in an age where integration and innovation of information and communication of technologies are expanding.<sup>102</sup> These technologies are changing the economy, deepening Internet cultural exchange, discussing new security risks and creating international governance through cyberspace. People from different countries have different values, rules, and regulation, so it is hard to create a unified virtual system.<sup>103</sup>

Individuals and programmers are entitled to rights, regulations, protection for themselves, and protection for their property because the Internet is accessible to all (who can afford it). It is no longer a luxury to be online. Being online is a necessity in order to be connected to the 21st century and without the Internet, many individuals would have a difficult time conducting their everyday lives as they know it. Lives on the Internet are created by others and therefore their work is the foundational steppingstone into the cyberworld. In order to have ownership in regard to source code and cyberspace, there needs to be flexibility and a progressive nature towards the law

---

<sup>100</sup> T.L. Taylor, "Whose Game Is This Anyway?": Negotiating Corporate Ownership in a Virtual World, 1 CGDC Conf. 227 (2002).

<sup>101</sup> Ibid.

<sup>102</sup> Zhang Shasha, One Community in Cyberspace, 62 Beijing Rev. 26 (2019).

<sup>103</sup> Ibid.

online. Computer science continually shows its protentional and growth yet does not have a clear solution for executable governance. Computer theorists, for now, suggest that contract law should be used in virtual worlds due to it being the most flexible in nature.<sup>104</sup>

To further bridge the gap between law and technology, there needs to be a special arbitration panel who understands the scope and complexity of cyberspace and technology. There are specialists called intro trials in order to speak more to the field in which they work. The same needs to be done for plagiarism and ownership regarding source code. There have been source code plagiarism detectors for students and programmers alike, but sometimes the consequences of plagiarism or modification are unclear. To remind again, source code can be unique, but the functionality can be the same. In that scenario, who is to say an entity has committed copyright infringement. Source code ownership is important because ownership is essential for compensation, data collection, and future development. Not only is source code plagiarism important in regard to the previous listed, but it will also be more prominent when technology advances to a stage that is unfathomable at the moment. While there may be ways in which to deal with source code ownership already like contracts, clickwrap agreements<sup>105</sup>, educational contracts, etc., it does not encompass the possibilities stealing source code creates. With current legislators' lack of knowledge, it is impossible to create and pass a law based on a topic they know very little about. Legislation needs progressive, forward-thinking specialists in order to encapsulate what is

---

<sup>104</sup> Christopher J. Cifrino, Virtual Property, Virtual Rights: Why Contract Law, Not Property Law, Must Be the Governing Paradigm in the Law of Virtual Worlds, 55 B.C.L. Rev. 235 (2014).

<sup>105</sup> “[C]lick- wrap’ (or ‘click-through’) agreements, require users to click an “I agree” box after being presented with a list of terms and conditions of use, or “browsewrap” agreements, which generally post terms and conditions on a website via a hyper- link at the bottom of the screen” James Grimmelmann Professor of Law, Internet Law: Cases & Problems 53 (9th ed. 2019).

currently going on with source code ownership and to shape the future of the Internet. There must be solid precedent that does not wavier too much in terms of validity.

For introspection on whether or not a special arbitration panel and legislators refined in the knowledge of technology, take the hearings in Congress of Mark Zuckerberg in regard to Facebook data collection. Zuckerberg was able to dodge many questions pertaining the illegal data collection of individuals without their permission due to the simple fact that the Congressmen and Congresswomen did not hold the knowledge of technology. While Zuckerberg did answer Congress' questions, he was able to not answer the questions fully and withhold information about the data collection that was happening behind the scenes. This is important to source code plagiarism because individuals and programmers will start arguing for their rights to their intellectual property as Artificial Intelligence (AI) progresses as well.

In regard to AI, virtual property is also built off of source code and created by programmers. As mentioned in the hypothesis, various languages of computer code have very limiting language structures at this current moment. Limiting language structures means the scope in which the code is written and produced. Many lines of code are built of others and there are many different ways to write it- the same as language. There is a structure to code and programmers use each other's work in order to produce further updates or new technology. Commerce is very much part of virtual property which is created by programmers and source code. As said by Justice Fahey in the case of United States v. Aleynikov, 676 F.3d 71 (2012)., "The changes made to a hard drive or disc when information is copied onto it are physical in nature. The representation occupies space".<sup>106</sup> Justice Fahey shows that source code becomes complicated when transferred to the physical world,

---

<sup>106</sup> United States v. Aleynikov, 676 F.3d 71 (2012).

stolen, or put in different areas than it was originally. Source code ownership is still in its early stages and is very vulnerable to poaching as well as being not fully protected by legislation.

A good take on what code means to the courts in the future is taken by Lessig as he has been featured throughout the previous chapters as he states,

Code is technical; courts aren't well positioned to evaluate such technicality. But even so, the failure is not even to try. The formalism in American law, which puts beyond review these structures of control, is a third pathology that inhibits choice. Courts are disabled, legislatures pathetic, and code untouchable. That is our present condition. It is a combination that is deadly for action—a mix that guarantees that little good gets done.<sup>107</sup>

It is clear that various legal systems around the world do not have a good handle on code and code seems to be a mutating entity that cannot be contained or governed. It is imperative that they know where they stand in order to create law around source code ownership that is hard to encapsulate in legislature currently. There needs to be a push for a repository and database in order to track ownership, as well as a specialized team that can bolster innovation for legislation surrounding code ownership.

---

<sup>107</sup> 2 Lawrence Lessig, Code, (2nd ed. 2006).

## 6. Conclusion

In terms of the research presented forth in this thesis, it is clear that law lags significantly behind technology. Legislators and government actors, both nationally and internationally, are at a deficit in terms of creating accurate domestic and international law to bridge the gap between law and code ownership. As discussed, code usage, function, and ownership are a main component of virtual life and commerce adaptation. Due to the overwhelming shift of online careers, buying, trading, investments, machinery, and everyday use, code ownership is more of an issue now than it originally was during the first iterations of the internet itself. This is because of the massive shift in virtual interaction, work, and compensation.

As of right now, there does not seem to be a clear definitive answer on creating a more sustainable way for code ownership to be recognized and adhered to. The issue still presents itself to be extremely complex for its time even with the current legislation set in place. While current legislation fixes various issues and keeps other issues at bay, it is not sustainable for the near future. This is due to the need for ownership to be even more present when large international companies and individuals alike need to declare their software, goods, and ideas in order to be properly protected, compensated, and to assume responsibility appropriately. Given the research that has been presented forth so far, even though there is a plethora of intellectual property legislation to still go over, it is a fact that there needs to be a system implementation in place. A solution, to which the research in this thesis provides, calls for an intellectual data base in order to track and disclose ownership both nationally and internationally. If there are technology agencies and companies that create a data base as vast, that the government also uses, it would be possible to create a data base to track source code ownership as well. In conjunction with the data base, academia has already implemented iterations of source code plagiarism detection

amongst Computer science students which would help in terms of implementing usage rights and ownership authority. This would allow for refined detail on original source code ownership even when various programmers work on the code thereafter. It would allow for the different governances to assign responsibility either in terms of ownership regarding compensation and credit or responsibility in terms of a criminal matter.

With this being said, this data base, which would data scrape information and allow for ownership submission internationally, would help put issues at rest regarding who owns what and who to assign responsibility to. With the example of the deficit of knowledge that even the United States Congress had regarding the rather extensive internet terminology and implementation, this database could help clear some uncertainty as well as inform various government employees around the world to the beginnings of law understanding the intricacies of code and source code ownership. As well as that, while there are experts to call upon during trials, testimonies, and the like, there needs to be an additional effort made to inform and teach current legislators, lawyers, judges, etc. so that they may respond and act accordingly and efficiently to technology in order to start maintaining and creating adequate law. On this point, while there is no definitive answer, the aforementioned provides an efficient and needed steppingstone in order for the law, both nationally and internationally, to become adequate and adapt as fast and as efficiently as technology does itself.

## BIBLIOGRAPHY

U.S. v. Microsoft 253 F.3d 34 (DC Cir, 2001) en banc.

United States v. Aleynikov, 676 F.3d 71 (2012).

15 U.S.C. 3704b.

1122. Introduction to the Economic Espionage Act, THE UNITED STATES DEPARTMENT OF JUSTICE (2020), <https://www.justice.gov/archives/jm/criminal-resource-manual-1122-introduction-economic-espionage-act> (Oct 10, 2020).

1312. National Stolen Property Act -- Goods, Wares, Merchandise, THE UNITED STATES DEPARTMENT OF JUSTICE (2020), Available at: <https://www.justice.gov/archives/jm/criminal-resource-manual-1312-national-stolen-property-act-goods-wares-merchandise> (Oct 10, 2020).

Andrej Zwitter, Big Data ethics, 1 Big Data & Society 1, 5 (2014).

Association of Software Professionals, Who Owns The Code? Available at: [https://asp-software.org/www/misv\\_resources/business-articles/who-owns-the-code/](https://asp-software.org/www/misv_resources/business-articles/who-owns-the-code/) (2008).

Christopher J. Cifrino, Virtual Property, Virtual Rights: Why Contract Law, Not Property Law, Must Be the Governing Paradigm in the Law of Virtual Worlds, 55 B.C.L. Rev. 235 (2014).

“Code.” (2019). In: Merriam-Webster Online Dictionary. 2019. Available at: <https://www.merriam-webster.com/dictionary/code> (24 Nov. 2019).

Franziska B. Keller et al., Political Astroturfing on Twitter: How to Coordinate a Disinformation Campaign. POLITICAL COMMUNICATION. Available at: <https://doi.org/10.1080/10584609.2019.1661888> (2019).

Georgina Cosma and Mike Joy, An Approach to Source-Code Plagiarism Detection and Investigation Using Latent Semantic Analysis, 61 IEEE Transactions on Computers 1 (2012).

Ikhlaq ur Rehman, Facebook-Cambridge Analytica data harvesting: What you need to know. Library Philosophy and Practice (e-journal), 1-10 (2019).

IARPA BE THE FUTURE, Home, Available at: <https://www.iarpa.gov/index.php/about-iarpa> (last visited Feb 15, 2021).

9 James Grimmelmann Professor of Law, Internet Law: Cases & Problems 53 (9th ed. 2019).

Jean-François Abramatic et al., Building the Universal Archive of Source Code: A global collaborative project for the benefit of all, 61 Viewpoints 29 (2018).



- Jorge Sagastume, CUSTOM BUILT SOFTWARE AND THE IP LAW - WHAT YOU NEED TO KNOW INTELLECTUAL PROPERTY WATCH (2018), <https://www.ip-watch.org/2017/11/07/custom-built-software-ip-law-need-know/> (last visited Feb 2, 2021).
- 2 Lawrence Lessig, Code 79 (2nd ed. 2006).
- Lucie Kadlecova, State Responsibility in the Cyber Age: The Course towards Indirect Evidence, 53 Mezinárodní Vztahy 35 (2018).
- McKay Coppins, The 2020 Disinformation War. *Atlantic*, 325(2), 28–39 (2020).
- Michaela Greiler et al., Code Ownership and Software Quality: A Replication Study, 1 IEEE 2 (2015).
- Moqod, WHO OWNS YOUR APP? HOW TO AVOID IP RIGHTS TRAPS IN SOFTWARE DEVELOPMENT MEDIUM, Available at: <https://moqod-software.medium.com/who-owns-your-app-how-to-avoid-ip-rights-traps-in-software-development-b12304028fdb> (2019).
- Nikolas Ott and Hugo Zylberberg, A European Perspective on the Protection of Personal Data in Cyberspace: Explaining How the European Union Is Redefining Ownership and Policies of Personal Data beyond National Borders, 16 KSR 69 (2016).
- NIST, About NIST (2017), Available at: <https://www.nist.gov/about-nist> (Dec 1, 2020).
- “Programming Language.” (2019). In: Tech Terms Online Dictionary. Available at: [https://techterms.com/definition/programming\\_language](https://techterms.com/definition/programming_language) (24 Nov. 2019).
- Sonia K. Katyal, The Paradox of Source Code Secrecy, 104 Cornell L. Rev. 103 (2019).
- “Source Code.” (2019). In: techopedia Online Dictionary. Available at: <https://www.techopedia.com/definition/547/source-code> (24 Nov. 2019).
- TechInsurance, KNOW YOUR INTELLECTUAL PROPERTY RIGHTS ON WORK-FOR-HIRE SOFTWARE TECHINSURANCE, <https://www.techinsurance.com/resources/intellectual-property-rights-software> (last visited February 13, 2021).
- UNITED STATES DIGITAL SERVICE, Our Mission, Available at: <https://www.usds.gov/mission> (2021).
- T.L. Taylor, “Whose Game Is This Anyway?”: Negotiating Corporate Ownership in a Virtual World, 1 CGDC Conf. 227 (2002).
- Theodora Vardouli & Leah Buechley, Open Source Architecture: An Exploration of Source Code and Access in Architectural Design, 47 Leonardo 51 (2014).
- Zhang Shasha, One Community in Cyberspace, 62 Beijing Rev. 26 (2019).