

UNIVERSITY OF CAPE TOWN

FACULTY OF LAW

SCHOOL FOR ADVANCED LEGAL STUDIES



**Application Program Interfaces and the Law of Copyright in South Africa:**

**A Proposal for an Optimal Approach**

By

Azola Siphe Futshane

(FTSAZO001)

Research Dissertation Submitted for the Approval of Senate in fulfilment of part of the requirements for the Master of Law in Intellectual Property Law in approved courses and a minor dissertation. The other part of the fulfilment for this qualification was the completion of a program of courses.

Supervisor:

**Dr. Lee-Ann Tong**

Word Count: 24, 819

**February 2017**

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source.

The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

## Declaration

Research dissertation presented for the approval of Senate in fulfilment of part of the requirements for the degree of Masters of Law in approved courses and a minor dissertation. The other part of the requirement for this qualification is the completion of a programme of courses.

I hereby declare that I have read and understood the regulations governing the submission of LLM dissertations, including those relating to length and plagiarism, as contained in the rules of this University, and that this dissertation conforms to those regulations.

I authorize the University of Cape Town to reproduce for the purpose of research the whole or any portion of the contents in any manner whatsoever.

I, **Azola Futshane**, know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

I have used the footnoting convention for citation and referencing. Each contribution to, and quotation in, this dissertation from the work(s) of other people has been duly attributed, and has been cited and referenced.

This dissertation is my own work.

I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.

Signature:

Date: 14 February 2018

## Abstract

Application Program Interfaces are simple computer programs that enable other pieces of software to call on each other so as to operate in tandem. They are solely designed to perform this singular function, yet they have a significant impact on computer software development. The benefits of Application Program Interfaces include the increased speed and efficiency of computer program development. In essence, Application Program Interfaces enable interoperability between different computer programs that may otherwise have been unable to operate with each other. Developers are saved from the onerous task of having to develop new programs that will be able to work with their own, because the Application Program Interface acts as a catalyst between the different programs, as it were.

Due to the function they perform, Application Program Interfaces are essential to computer software development, and are of high economic value. As Application Program Interfaces are computer programs, they are protected under the law of Copyright. This means that developers of Application Program Interfaces are able to exercise exclusive rights in copyright over these pieces of technology. This is problematic in the programming community, where there is a rapidly growing trend towards using and producing open source software. As such, there is a friction created between programmers who seek to use Application Program Interfaces freely, and the copyright owners who seek to derive an economic benefit.

There is no controversy as to whether the literal elements of an Application Program Interface may be eligible for copyright or not. There is debate, however, as to whether, and to what extent, the non-literal elements of an Application Program Interface may be eligible for copyright. This debate brings to the fore the age-old idea-expression dichotomy, and the question of what elements constitute an idea or an expression.

This paper will investigate the approach to Application Program Interfaces and copyrightability in South Africa, and make proposals as to the approach that ought to be taken. Guidance will be taken from the precedent and legal debates ongoing in other jurisdictions.

# Table of Contents

<b>Abstract</b>	<b>5</b>
<b>Chapter 1: Introduction</b>	<b>10</b>
1.1 Background of APIs in society	10
1.2 APIs in the Context of Copyright	11
1.2.1 Comparative Analysis of Copyright Treatment Using Foreign Law	11
1.2.2 Ideas-Expression Dichotomy	12
1.2.3 Literal and Non-literal Components	13
1.3 Problem	14
1.3.1 Open Source Software	14
1.3.2 Interoperability and Its Importance	15
1.3.3 Conflict Between Interests of Copyright Owner and the Community	16
1.4 Thesis Organisation	16
1.4.1 Outline of Arguments	16
1.4.1.1 Copyrightability of Non-Literal Components	17
1.4.1.2 Exceptions to Liability for Infringement	19
1.4.2 Chapter Focus	20
<b>Chapter 2: Copyright in Context: Prevailing Position in South Africa</b>	<b>21</b>
2.1 Justification for Copyright Protection	21
2.1.1 Introduction	21
2.1.2 Rational for Protection of Author Rights: Utilitarianism	22
2.1.3 Relevance of Justification	23
2.2 Copyright in Context	24

2.2.1 Introduction	24
2.2.2 International Treatment of Copyright	24
2.2.3 Requirements for Copyright Eligibility per South Africa Law	25
2.3 Treatment of APIs in South Africa	26
2.3.1 Introduction	26
2.3.2 Copyrightability	27
2.3.3 Exceptions to Infringement	27
2.3.3.1 Quotations as a Fair Dealing Exception: Potential Applicability of s12(3)	28
<b>Chapter 3: APIs and Copyrightability</b>	<b>30</b>
3.1 Introduction	30
3.2 USA Approach	31
3.2.1 Summary of USA Approach	31
3.2.2 <i>Oracle v Google 1</i>	33
3.2.2.1 Findings of the Court	33
3.2.2.2 Reasoning of the Court	34
3.2.2.3 Analysis of Court Reasoning	38
3.2.2.4 Implications of Judgment	39
3.2.3 <i>Oracle v Google 2</i>	39
3.2.3.1 Findings of the Court	39
3.2.3.2 Reasoning of the Court	40
3.2.3.3 Analysis of Court Reasoning	41
3.2.3.4 Implications of Judgment	43
3.3 UK Approach	44
3.3.1 Summary of UK Approach	44

3.3.2 <i>SAS v World Programming</i>	45
3.3.2.1 Findings of the Court	45
3.3.2.2 Reasoning of Court	46
3.3.2.3 Analysis of Court Reasoning	47
3.3.2.4 Implications of Judgment	48
3.4 South Africa	49
3.4.1 Summary of South African Approach	50
3.4.2 The Constitutional Context	50
3.4.3 Analysis of Current Legal Position and Proposal for Improved Approach	51
<b>Chapter 4: APIs and Exceptions to Infringement</b>	<b>54</b>
4.1 Introduction	54
4.2 Alternative Mechanism: Reverse Engineering	55
4.3 United States of America	57
4.3.1 Summary of USA Approach	58
4.3.2 Fair Use Doctrine	58
4.3.3 <i>Oracle v Google 2</i>	58
4.3.3.1 Summary of Findings	58
4.3.3.2 Case Discussion of Fair Use	58
4.3.4 Implications of Fair Use Treatment	60
4.4 United Kingdom	62
4.4.1 Summary of UK Approach	62
4.4.2 Fair Dealing Doctrine	62
4.4.2.1 Permissible Uses in Fair Dealing	63
4.4.2.2 Factors of Fairness	64

4.4.3 Implications of Fair Dealing Doctrine Treatment	65
4.4 South Africa	66
4.4.1 Summary of South African Approach	67
4.4.2 Constitutional Context	67
4.4.3 Discussion of South African Copyright Bill	68
4.4.4 Proposal for Improved South African Approach	71
<b>Chapter 5: Conclusion</b>	<b>72</b>
<b>Bibliography</b>	<b>76</b>

## CHAPTER 1: INTRODUCTION

### 1.1 Background

Whether Application Programming Interfaces (hereafter 'APIs') should be freely available for use under the laws of copyright or not is an issue worth billions of Dollars.<sup>1</sup> The magnitude of the monies implicated underlies the importance of software in the modern technological age.

Computer software applied in smartphones, laptops, tablets, desktops, super-computers and their ilk is central<sup>2</sup> to endless facets of global development including the worlds of business, healthcare and science; as well as the financial sector and the social landscape. It is of value not only because of its economic uses in global development, but also because of the proprietary value found in its 'intellectual property-dense material like programming codes, technical specifications, graphic displays and user manuals'.<sup>3</sup> Computer programs, in particular, are the driving force of software. Key to their interoperability, which is their ability to work with each other to achieve designated results<sup>4</sup>, is the API.

An API is a public specification which is used as an interface between components of software so that they may communicate with each other.<sup>5</sup> It includes a set of routines, protocols and tools used for building software applications, or in web development. Simply put, an API is comprised of code which is used in computer programming so that different applications may communicate with each other in order to operate in conjunction with each other, even if they are programmed in different languages.

---

<sup>1</sup> This question formed the basis of recently-decided litigation in the USA between Oracle (PTY) Ltd and Google (PTY) Ltd in a claim which amounted to nine billion Dollars. This matter will be discussed below.

<sup>2</sup> Simon Crowfoot 'Five Computer Programs We Use Everyday' available at <http://www.iceni.com/blog/five-computer-programs-we-use-every-day/> accessed on 12 February 2018.

<sup>3</sup> Lee-Ann Tong 'Copyright Protection for Computer Programs in South Africa: Aspects of Sui Generis Categorization' (2009) 12 *The Journal of World Intellectual Property* 266 at 266.

<sup>4</sup> John Abbot 'Reverse Engineering of Software: Copyright and Interoperability' (2003) 14 *J.L & Inf. Sci* 7 at 16.

<sup>5</sup> Uri Sarid 'A Non-Apocalypse: APIs, Copyright and Fair Use' available at <https://www.wired.com/insights/2014/05/non-apocalypse-apis-copyright-fair-use/> accessed on 12 February 2018.

The API consists of packages split into classes which carry out methods. The methods are lines of code which produce required functions. The technical aspects of an API were best explained in *Oracle*<sup>6</sup>:

'An API is like a library. Each package is like a bookshelf in the library. Each class is like a book on the shelf. Each method is like a how-to-do-it chapter in a book. Go to the right shelf, select the right book, and open it to the chapter that covers the work you need. Every method and class is specified to carry out precise desired functions and, thus, the "declaration" (or "header") line of code stating the specifications must be identical to carry out the given function.'<sup>7</sup>

Due to their properties, APIs are prevalent in software development. They are used, for instance, by web applications that seek to use the infrastructure of Google Maps in order to provide GPS services to persons who use the respective web applications. The function that they serve is critical; they enable developers and programmers to create efficient networks of software which operate in tandem.

## 1.2 APIs in the Context of Copyright

### 1.2.1 Comparative Analysis of Copyright Treatment Using Foreign Law

APIs are computer programs.<sup>8</sup> Computer programs are treated differently in various jurisdictions. It is for this reason that it may be difficult to extrapolate principles from foreign law in order to receive guidance as to how our local law ought to be.<sup>9</sup> Nevertheless, in the absence of a rich body of literature concerning the treatment of APIs in South Africa, it is necessary to engage in a comparative analysis with other jurisdictions where judicial and academic debate is rife in relation to APIs and copyright specifically, or computer programs and copyright generally.

It is for this reason that the law as applied in the United States of America (hereafter the USA) and the United Kingdom (hereafter the UK), respectively, has been chosen in order to provide ideas which may be explored and analysed. This is

---

<sup>6</sup> *Oracle Am., Inc. v. Google, Inc.*, 872 F. Supp. 2d 974 (N.D. Cal. 2012) at 5.

<sup>7</sup> This understanding of APIs was also used in the appeal judgment: *Oracle Am., Inc. v. Google Inc.*, No. 13-1021 (Fed. Cir. 2014).

<sup>8</sup> Shireen Smith 'Are APIs Protected by Copyright? – Google v Oracle' available at <https://blogs.mulesoft.com/dev/quiz/api-expert-knowledge-test/> accessed on 12 February 2018.

<sup>9</sup> Caroline B Ncube 'Equitable Intellectual Property Protection of Computer Programs In South Africa: Some Proposals For Reform' (2012) 3 *Stellenbosch Law Review* 438 at 442.

so that a proposition may ultimately be put forward for application in the South African context. The reason that the USA has been chosen for analysis is that the question of the copyrightability<sup>10</sup> of APIs has been directly dealt with in the controversial *Oracle* matter. The reason that analysis of the UK position is appropriate is twofold. Firstly, South African copyright law is largely based on UK copyright law so the two systems of law are not entirely divorced historically<sup>11</sup>; and secondly, the UK judiciary has taken a firm position on the issue of copyrightability and the non-literal elements of a computer program. Granted, this position is not limited to APIs but applies to computer programs generally. Nevertheless, the arguments in these jurisdictions are relevant because certain important themes, such as the ideas-expression dichotomy, are common to all three jurisdictions.

### 1.2.2 Ideas-Expression Dichotomy

Computer programs are unique in that they have characteristics that fit into both the definitions of copyrightable matter as well as patentable matter.<sup>12</sup> These unique characteristics have made it difficult to categorise computer programs<sup>13</sup>, and even though it is widely accepted that they do not constitute inventions for the purpose of patent law, challenges persist as to how they should be treated in the realm of copyright.<sup>14</sup>

Specifically, the ideas-expression dichotomy is a challenge with regards to computer programs. In copyright law, ideas are available to all persons and consequently do not attract protection, whereas expressions are the products of creative effort and are thus worthy of eligibility for copyright protection.<sup>15</sup> The ideas-expression dichotomy encapsulates this view that ideas and the expressions thereof are different.<sup>16</sup> Problems arise in determining which component of a work constitutes

---

<sup>10</sup> In this paper, 'copyrightability' and all variations thereof mean: 'able to be eligible for copyright'.

<sup>11</sup> Ncube op cit note 9 at 442.

<sup>12</sup> See Ncube ibid at 439, who notes that the source code of a computer program is protectable under copyright, whilst the functionality may be protected in patent law.

<sup>13</sup> Charles R. McManis 'Intellectual Property Protection and Reverse Engineering of Computer Programs in the United States and the European Community' (1993) 8 *Berkeley Technology Law Journal* 26 at 43.

<sup>14</sup> Tong op cit note 3 at 268.

<sup>15</sup> See Edward Samuels 'The Idea-Expression Dichotomy in Copyright Law' (1989) 56 *Tennessee Law Review* 321 at 323.

<sup>16</sup> Matt Flinders 'Protecting Computer Software – Analysis and Proposed Alternative' (2007) 7 *Journal of High Technology Law* 71 at 78.

an idea, and which part qualifies as an expression.<sup>17</sup> This issue appears to be focal in the question of whether APIs should be eligible for copyright or not.

### 1.2.3 *Literal and Non-Literal Components*

APIs consist of lines of code which cause computers to perform actions. Like all programs, APIs are created using source code. This is the component written by a programmer in a language legible to humans. It is subsequently translated into object code, which is the instruction in a form which can only be interpreted by computers.<sup>18</sup> This code is treated as a literary work across most jurisdictions party to the Agreement on Trade-Related Aspects of Intellectual Property Rights (hereafter the 'TRIPS Agreement').<sup>19</sup> In South Africa, however, it receives its own *sui generis* protection as a computer program work.<sup>20</sup>

Copyright works across these three jurisdictions have literal and non-literal components. The definition of the latter differs slightly across the three jurisdictions under discussion. The literal part of a computer program is the actual written portion of the code.<sup>21</sup> Copying of the literal component is always unlawful where it is substantial.<sup>22</sup> This is determined with a greater emphasis on the consideration of the quality of the copied portion, rather than the amount.<sup>23</sup> This much is the same in South Africa as it is in the other jurisdictions. Differences appear, however, in the manner in which the non-literal elements of a computer program are treated.

In South Africa, there is little case law which considers copyright and computer programs.<sup>24</sup> Nevertheless, the non-literal components of a computer program have

---

<sup>17</sup> Ncube; Allen Rosen 'Reconsidering the Idea/Expression Dichotomy' (1992) 26 *University of British Columbia Law Review* 263 at 263.

<sup>18</sup> Roux de Villiers 'Computer Programs and Copyright: The South African Perspective' (2006) 123 *SALJ* 315 at 334.

<sup>19</sup> Agreement on the Trade-Related Aspects of Intellectual Property (TRIPS), 15 April 1994, *Marrakesh Agreement Establishing the World Trade Organization, Annex 1C, 1869 U.N.T.S. 299, 33 I.L.M. 1197*.

<sup>20</sup> Tong op cit note 3 at 268.

<sup>21</sup> Arjun Krishnan 'Testing for Copyright Protection and Infringement in Non-Literal Elements of Computer Programs' (2005) 10 *Journal of Intellectual Property Rights* 9 at 9 – 10; Richard Raysman & Peter Brown 'Copyright Infringement of Computer Software and the 'Altai' Test' *New York Law Journal* 9 May 2006.

<sup>22</sup> David I Bainbridge 'The Look and Feel of Computer Programs after Richardson v. Flanders' (1993) 2 *Laws, Computers and Artificial Intelligence* 269 at 274.

<sup>23</sup> *Haupt t/a Softcopy v Brewers Marketing Intelligence (Pty) Ltd. and Others* 2006 (4) SA 458 (SCA) para 45.

<sup>24</sup> Ncube op cit note 9 at 442.

been acknowledged to exist. In *Pastel Software*<sup>25</sup> the court failed to distinguish between the infringement of the literal and non-literal components of software. Notwithstanding its conflation of the two elements, the court did acknowledge that computer programs do have a non-literal part.<sup>26</sup> The non-literal component of a program includes the 'structure, sequence of operations, functions, interfaces and methodologies.'<sup>27</sup> In the USA, the non-literal component of a computer program has been described as its 'structure, sequence and organisation.'<sup>28</sup> In the UK, the non-literal element of a computer program is its 'structure, flow and sequence of operations'.<sup>29</sup> In essence, this describes the functionality of the computer program.<sup>30</sup>

Although the jurisdictions differ slightly in their interpretations of the non-literal components of a computer program, a common thread appears. The non-literal component pertains to the functionality of the program. In relation to APIs, issues arise as to whether this should be copyrightable or not. This is where the ideas-expression dichotomy comes into play. A court must decide whether the non-literal components of an API constitute an idea or an expression.

### 1.3 Problem

#### 1.3.1 Open Source Software

Legal control over new creations is granted to the creators thereof for the purpose of preventing, delaying or increasing the cost of imitation by other persons in order to encourage innovation.<sup>31</sup> Other reasons for legal protection of rights in intellectual property are canvassed below. Suffice it to say, the norm is for exclusive rights of exploitation to attach over a creation in favour of the creator.

---

<sup>25</sup> *Pastel Software (Pty) Ltd V Pink Software (Pty) Ltd and Another* 399 JOC (T), as stated in Ncube *ibid*.

<sup>26</sup> *Ibid*.

<sup>27</sup> Ncube *op cit* note 9 at 442.

<sup>28</sup> *Oracle supra* note 6 at 2.

<sup>29</sup> David I Bainbridge *Introduction to Computer Law* 2 ed (1993) 32.

<sup>30</sup> *SAS Institute Inc v World Programming Ltd* (2013) EWHC 69 (Ch).

<sup>31</sup> Mark A Lemley & Ziv Shafir 'Who chooses Open-Source Software' (2011) 78 *University of Chicago Law Review* 139 at 139.

In spite of this, open source software (hereafter OSS) has become increasingly important in markets that are the traditional domain of large proprietary groups.<sup>32</sup> OSS is source code that any person can modify and share because, by design, it is accessible to the public.<sup>33</sup> The existence of OSS is reliant on intellectual property rights that are contrary to the norm in that they keep software freely and widely accessible. This allows other developers to view the code, copy, alter, share and learn from it.<sup>34</sup> It differs from code which is kept in private by its natural and juristic owners for any number of reasons, chief of which include the commercial exploitation of the developed software as a product. This is called proprietary software.<sup>35</sup>

### 1.3.2 Interoperability and Its Importance

Interoperability is defined as ‘the ability of different information technology systems and software applications to communicate, to exchange data accurately, effectively, and consistently, and to use the information that has been exchanged.’<sup>36</sup>

This is important in the realm of computer programs as it means that different programs are able to work in conjunction with each other to create systems able to perform complex functions. An example that highlights the importance of interoperability is the Internet. It consists of software from innumerable suppliers that works in conjunction to create a multitude of electronic networks that enable the creation of an enormous telecommunications infrastructure.<sup>37</sup>

APIs are unique in relation to other computer programs in that they are very simple programs designed to perform a single function – enabling program interoperability.<sup>38</sup> This is relevant when determinations must be made as to the level

---

<sup>32</sup> Vidya Atal & Kameshwari Shankar ‘Developers’ Incentives and Open Source Software Licensing: GPL vs BSD’ (2015) 15 *B.E Journal of Economic Analysis and Policy* 1381 at 1381.

<sup>33</sup> Lemley & Shafir op cit note 31 at 139.

<sup>34</sup> Ibid.

<sup>35</sup> As referred to in Tong op cit note 3 at 266.

<sup>36</sup> National Alliance for Health Information Technology ‘What Is Interoperability?’ available at [www.nahit.org](http://www.nahit.org) accessed on 12 February 2018.

<sup>37</sup> Kurt Rowley ‘Understanding Software Interoperability in a Technology Supported System of Education’ available at <https://www.educause.edu/ir/library/pdf/cem9535.pdf> accessed on 12 February 2018.

<sup>38</sup> John-Paul Kamath ‘APIs lead the Way to Interoperability’ available at <http://www.computerweekly.com/feature/APIs-lead-the-way-to-better-interoperability> accessed on 12 February 2018.

and kind of creativity that goes into the development of an API, for purposes of separating ideas from expressions.

### *1.3.3 Conflict between Interests of Copyright Owner and the Community*

Due to the fact that APIs are publicly available, issues arise where the creator of the respective API in any given circumstance seeks to exclude others from making use of the API, subject to some form of compensation. This creates a conflict between the rights of the private party who wrote the API, and the interests of the greater community in making use of the APIs.

Broadly speaking, the question arises as to whether, and to what extent, the creator of an API ought to be able to exercise rights over the API. Specifically, the issue is whether APIs are, and should be, of a nature sufficient to warrant copyright protection; and, if so, whether exceptions to copyright infringement are available to users of APIs so that they may be precluded from attracting legal liability.

## 1.4 Thesis Organisation

### *1.4.1 Outline of Arguments*

This thesis will investigate two issues. The first issue concerns whether or not the non-literal aspects of an API should be eligible for copyright protection under South African law. There will be no such investigation of the law pertaining to the literal components of an API as it is not a matter of controversy that source code may be eligible for copyright.<sup>39</sup> It will be found that the non-literal elements of an API ought not to be copyrightable in South Africa.

The second issue is whether, in the alternative event that APIs are copyrightable in South Africa, the available exceptions to infringement offer adequate protection for users<sup>40</sup> of APIs. Where these exceptions do not provide an adequate

---

<sup>39</sup> TRIPS Agreement op cit note 19, Article 10(1).

<sup>40</sup> In this paper, the term 'user' denotes a person who seeks to use a copyright work, even though she does not own any copyright in the work, and does not have the requisite licence from the copyright owner to use the work in any way.

legal avenue for users, proposals will be made in an attempt to resolve their shortcomings. These proposals will call for measures that have already been proposed in the Copyright Bill, subject to minor changes.

These arguments and the reasoning behind them as they will appear in the paper are expanded upon below.

#### *1.4.1.1 Copyrightability of Non-Literal Components*

The Copyright Act of 1978<sup>41</sup> (hereafter the SA Copyright Act) protects computer programs.<sup>42</sup> However, there is no statutory provision, case law or academic writing which concerns the copyrightability of APIs in South Africa. Specifically, there is no South African precedent or written source which gives direction as to whether the non-literal components of an API should be eligible for copyright or not.

In the US, the prevailing, albeit controversial, position is that the structure, sequence and organisation of an API should be copyrightable. This stems from the following reasoning. Firstly, the Copyright Act of 1976<sup>43</sup> (hereafter the US Copyright Act) protects computer programs as literary works, but does not protect mere ideas.<sup>44</sup> In order to determine if the structure, sequence and organisation of an API constitute an idea or an expression, it must be determined whether these non-literal elements are purely functional or whether they are imbued with creativity on the part of the author, sufficient to render them as expressions.<sup>45</sup> On this basis, it is argued, it is possible for an API's structure, sequence and organisation to contain functional and creative elements.<sup>46</sup> Therefore, it is possible for an API to be copyrightable, and its eligibility must be determined on a case by case basis.<sup>47</sup>

A counter position exists in the US, though it has been superseded by the approach described above.<sup>48</sup> This counter position is to the effect that the structure,

---

<sup>41</sup> Copyright Act 98 of 1978.

<sup>42</sup> *Ibid*, s2(1)(i).

<sup>43</sup> Copyright Act of 1976, 17 U.S.C.

<sup>44</sup> *Ibid*, s102(b).

<sup>45</sup> *Oracle supra* note 6 at 30.

<sup>46</sup> *Oracle Appeal supra* note 7 at 48.

<sup>47</sup> *Ibid*.

<sup>48</sup> *Ibid*.

sequence and organisation of an API is functional.<sup>49</sup> All of the inputs used to design it are ultimately utilitarian. Irrespective of the ingenuity required to create the structure, sequence and organisation of the API, all decisions are intended to achieve a functional result.<sup>50</sup> As such, any design of the API is not expressive, but functional in that it is used to improve the efficiency of the API. Thus, for the purposes of copyright law it cannot be copyrightable.<sup>51</sup>

In the UK, there is no specific comment on APIs. However, there is a blanket ban on the copyrightability of the functionality of computer programs.<sup>52</sup> The Copyright, Designs and Designs Act<sup>53</sup> (hereafter the UK Copyright Act) also protects computer programs as literary works, and offers no protection to mere ideas. Like the latter approach in the US, it is argued in the UK that an API is purely functional, with no room for expressive creation.<sup>54</sup> It is viewed as a purely utilitarian work, such that its non-literal components constituting its functionality cannot attract copyright protection. Furthermore, it is argued from a social perspective that granting copyright rights over an API would be prejudicial to the community because APIs are so integral to achieving interoperability between separate programs. If APIs were eligible for copyright, so it is argued, small players will be pushed out of the market.<sup>55</sup>

I argue in favour of the approaches which find that APIs should not be eligible for copyright. I argue that the SA Copyright Act must be interpreted with the values of the Constitution<sup>56</sup> in mind. These values call for fair and equitable balance between the rights of a copyright owner to his property, and the interests of the community in using available technology to boost its socio-economic welfare.

In a similar manner to the copyright statutes in the other jurisdictions, the SA Copyright Act protects computer programs<sup>57</sup> but not ideas. As there is no case law to offer direction as to what the approach in South Africa should be, I argue that the US and UK approaches are instructive and should be followed insofar as they postulate

---

<sup>49</sup> *Oracle* supra note 6 at 34.

<sup>50</sup> *Ibid* at 35.

<sup>51</sup> *Ibid* at 34.

<sup>52</sup> This stems from the decision of the European Court in interpreting its doctrine as discussed in section 3.3 of this paper.

<sup>53</sup> Copyright, Designs and Patents Act 1988.

<sup>54</sup> *SAS* supra note 30 para 40.

<sup>55</sup> *Ibid* para 31.

<sup>56</sup> Constitution of the Republic of South Africa, 1996.

<sup>57</sup> SA Copyright Act op cit note 41, s2(1)(i).

that the true nature of an API is that of a functional work wherein no creative expression exists. I argue that this interpretation is not unfair in relation to the copyright owner of any API. The limitation of the copyright owner's rights is fair when consideration is given to the overall benefit which accrues to the greater community. This, I argue, is constitutionally sound as it constitutes an equitable and proportional balance of interests between affected parties.

#### 1.4.1.2 *Exceptions to Liability for Infringement*

The SA Copyright Act contains a closed list of acceptable uses which would otherwise constitute copyright infringement.<sup>58</sup> Computer programs are included in these exceptions, however, I submit that it is highly doubtful that a user of an API would benefit from these exceptions. This is because the exceptions were originally designed to cater for traditional literary and artistic works. The uses that they permit are far too narrow for an API user who seeks to use the respective API for its interoperability function in order to make two or more separate programs operate in tandem.

Guidance from the US approach shows that fair use is an effective method of catering to the needs of an API user.<sup>59</sup> The fair use doctrine relates to an open list of uses which may be deemed to be fair even though they would otherwise offend a copyright owner's rights in a work.<sup>60</sup> It allows a judge to determine on a case by case basis if a user's activities in relation to an API are excusable or not. The interoperability function of an API is a key factor in favour of a determination that the use is fair.<sup>61</sup> However, other factors such as the prejudice that the use may cause the copyright owner in the market operate against a finding of fair use.<sup>62</sup> This means that the result of this enquiry is not certain or predictable. This will discourage API users who do not

---

<sup>58</sup> *Ibid*, s12.

<sup>59</sup> It is well accepted that fair use offers greater protection than the fair dealing doctrine. See Gluseppina D'Agostino 'Healing Fair Dealing – A Comparative Copyright Analysis of Canada's Fair Dealing to U.K. Fair Dealing and U.S. Fair Use' (2008) 53 *McGill Law Journal* 309 at 314.

<sup>60</sup> D.R Jones 'Commerciality and Fair Use' (2015) 15 *Wake Forest Journal of Business and Intellectual Property Law* 620 at 621; Sean O'Donnell 'Fair Use in the UK, Fair Use in the USA ... No Future, No Future' (2007) 7 *Hibernian Law Journal* 57 at 58.

<sup>61</sup> *Oracle Appeal* supra note 7 at 61.

<sup>62</sup> *Ibid* at 60.

have the means to risk litigation with the copyright owner.<sup>63</sup> This, it is argued, hinders innovation. The US also permits reverse engineering of computer programs<sup>64</sup> but this is not a right of use. As such, though it is advantageous in that it permits the user an opportunity to understand the workings of an API, it is limited in that the user is still forced to construct her own API as opposed to being able to use the existing API. Furthermore, not all users will be skilled at reverse engineering. As such, only the few will benefit.

The fair dealing doctrine as applied in the UK, in comparison, is narrowed to particular approved purposes<sup>65</sup> and does not afford API users an opportunity to benefit from any existing exceptions to copyright infringement. The UK Copyright Act provides a closed list of acceptable uses, but this list does not cater for uses in relation to APIs. As in the US, the UK does permit reverse engineering of computer programs.<sup>66</sup> The same advantages and disadvantages as in the US appear in this regard.

In light of all of the above, I argue that South African law must be less restrictive and allow for uses which better relate to the users of computer programs. I note that there is a new South African Copyright Bill<sup>67</sup> (hereafter the Copyright Bill) which permits reverse engineering of computer programs and creates a right of fair use when the use is related to the interoperability of a computer program. In light of this Copyright Bill, I argue that the law will be adequately developed in a constitutionally sound manner if the Copyright Bill is enacted, subject to slight semantic changes to some of its provisions. I argue that the provisions of the Copyright Bill put into effect an equitable and proportional balance between the rights of the affected parties.

#### *1.4.2 Chapter Focus*

Chapter 2 will concern the justifications for the rights that attach to the creators of works and outline the basics of copyright law from a South African perspective. Chapter 2 will also describe the present South African position in relation to the

---

<sup>63</sup> Timothy B. Lee 'The Court that Created the Patent Troll Mess is Screwing Up Copyright Too' available at <https://www.vox.com/2014/5/9/5699960/this-court-decision-is-a-disaster-for-the-software-industry> accessed on 12 February 2018.

<sup>64</sup> Ncube op cit note 9 at 442.

<sup>65</sup> Catherine Colston 'Fair Dealing: What is Fair?' (1995) 10 *Denning Law Journal* 91 at 103.

<sup>66</sup> Ncube op cit note 9 at 442.

<sup>67</sup> Copyright Amendment Bill 2015 in GN 646 GG 39028 of 27 July 2015.

copyrightability of APIs and the law as it relates to exceptions to copyright infringement and APIs. Chapter 3 will investigate the approach to API copyrightability in the USA and the UK, and suggest that APIs should not be eligible for copyright in South Africa. Chapter 4 will investigate the exceptions to copyright infringement in relation to APIs across the foreign jurisdictions, and suggest that there should be exceptions developed in South African law which better balance the rights of the copyright owner and the community of API users. Chapter 5 will offer the conclusion, including the implications on South Africa of all of the findings provided in the preceding Chapters.

## CHAPTER 2: COPYRIGHT IN CONTEXT: PREVAILING POSITION IN SOUTH AFRICA

### 2.1 Justification for Copyright Protection

#### 2.1.1 Introduction

These issues will call into question the balance of the interests of the creator of a work – in this case the creator of the API – and the interests of the community. As such, to consider the nature of this balance properly it is necessary to be cognisant of the most accepted justification<sup>68</sup> as to the reason that the authors of works are deserving of copyright protection.

Over the years, different rationales have been put forward to provide the basis as to the reason that copyright protection exists.<sup>69</sup> This is in light of the fact that copyright protection has been accused of, at its worst, serving to inhibit additional creation and thereby stifling creativity and development.<sup>70</sup> As such, these rationales exist so that the copyright law does not operate in a void sans logical and sound

---

<sup>68</sup> See Shubha Ghosh 'Duty, Consequences, & Intellectual Property' (2013) 10 *University of St. Thomas Law Journal* 801 at 801 who acknowledges that the incentivisation of creativity and invention of new works is often viewed as the justification for intellectual property rights.

<sup>69</sup> See Jean-Luc Piotraut 'An Authors' Rights-Based Copyright Law: The Fairness and Morality of French and American Law Compared' (2006) 24 *Cardozo Arts and Entertainment* 549 at 553 – 554.

<sup>70</sup> See Katie Sykes 'Towards a Public Justification of Copyright' (2003) 61 *University of Toronto Faculty Law Review* 1 at 6 who notes the view that: 'in some instances, technical infractions [of copyright] are sometimes more in harmony with the spirit of the law than obedience would be.'

reasoning as to its existence. The most common justification<sup>71</sup> of relevance is the utilitarian theory.<sup>72</sup>

### 2.1.2 Rationale behind Protection of Author Rights: Utilitarianism

The prime economic justification for the protection of copyright is to provide authors with the 'necessary incentives to create works'.<sup>73</sup>

In economic terms, the public-good nature of copyright works necessitates the creation of incentives in the form of copyright protection in order for authors to create works; absent said incentives authors will not make works, or make enough of them.<sup>74</sup> Public goods possess the properties of being both non-rival and non-excludable. A good (or service) which is non-rival is one that can be enjoyed by more than one person at a given period.<sup>75</sup> This means, for example, that the contents of a book, like the code of a computer program, can concurrently be used and enjoyed by more than one person without diminishing in value or quality. The term 'non-excludable' denotes, in the case of tangibles, property that cannot be 'physically controlled'.<sup>76</sup> What this means in the context of copyright works is that persons cannot be prevented from the enjoyment of said works.<sup>77</sup>

The issue with public goods, then, lies in their non-excludability. A public good creates positive externalities in that it produces benefits which consumers can enjoy, while there is no way for the producer of the goods to prevent said enjoyment.<sup>78</sup> This creates the problem of free-riding, which, simply put, means that nobody is going to pay for something that they can easily get for free. This results in market failure because the lack of incentive to pay means that, although the demand is being met, there is no concurrent transfer of payment from consumer to producer in exchange for

---

<sup>71</sup> Sadulla Karjiker *Open-Source Software and the Rationale for Copyright Protection of Computer Programs* (published PhD thesis, Stellenbosch University, 2013) 34.

<sup>72</sup> *Ibid* at 32.

<sup>73</sup> Sykes op cit note 70 at 23.

<sup>74</sup> Karjiker op cit note 71 at 108.

<sup>75</sup> Mark A. Lemley 'Ex Ante v Ex Post Justifications for Intellectual Property' (2004) 71 *University of Chicago Law Review* 129 at 129.

<sup>76</sup> Karjiker op cit note 71 at 93.

<sup>77</sup> *Ibid* at 94.

<sup>78</sup> *Ibid* at 95.

the supply. Furthermore, it is inefficient for producers to take measures to prevent free-riding where the cost of supplying the good to one more person is negligible, because people will simply stop consuming the good rather than start paying for it.<sup>79</sup> In this regard, the cost of exclusion is non-negligible; and the producer cannot recover the costs for the benefit she has created.

Market failure occurs as a product of free-riding behaviour which results from the public-good nature of copyright works, and their positive externalities. This failure leads to a 'decrease in social welfare because copyright works are not produced at socially-desired levels'.<sup>80</sup> Simply put, authors will not create works when others are able to freely benefit at the expense of the efforts of the authors. The optimal solution, then, is to rectify this situation by providing protection for authors in the form of property rights. Consequently, it is necessary to incentivise authors to create such works by 'eliminating such free-riding conduct, and enabling them to realise a sufficient return on their investments'.<sup>81</sup>

### *2.1.3 Relevance of Justification*

This paper investigates whether APIs should be eligible for copyright or not in South Africa. It also determines whether the use of APIs by users should constitute an exception to copyright infringement or not. These questions necessarily induce a friction between the interests of those who seek to assert copyright rights over APIs, and those for whom the optimal outcome is that APIs be freely accessible. In navigating through this friction, it is important to remember that copyright protection is justified as it generally provides an incentive for creation.

In sum, the economic rationale is of persuasive value as it definitively shows that there is ample reason in logic for authors<sup>82</sup> to receive copyright protection. This is important because, in instances where it may be tempting to consider only the broader interests of the community, it draws attention to the interests of the person who seeks to have copyright rights in a work. It calls for an equitable balancing of interests.

---

<sup>79</sup> Ibid at 95.

<sup>80</sup> Ibid at 110.

<sup>81</sup> Ibid.

<sup>82</sup> In the event that they own copyright in the respective work.

Notwithstanding the above, however, a fair and balanced result will necessitate the limiting of the rights of the author.

## 2.2 Copyright in Context

### 2.2.1 Introduction

While the rationale for the protection of authorship rights in copyright works has been dispensed with, it is necessary to place the forthcoming analysis into greater context so that the arguments may be considered against an informed background. What follows is a general description of copyright law, as it is treated on the international stage, followed by the basics of copyright law in South Africa. Then the treatment of computer programs in South Africa is investigated, with focus given to the questions of copyrightability and possible exceptions to infringement. The manner in which copyright is dealt with in the USA and the UK – specifically in regard to computer programs - will become apparent in the analyses of the approaches taken by the respective jurisdictions.

### 2.2.2 International Treatment of Copyright

The Berne Convention<sup>83</sup> provides for the minimum protection that is to be given in respect of literary works in states that are party to it.<sup>84</sup> The Berne Convention does not specifically make provisions in respect of computer programs.<sup>85</sup> However, South Africa is a Member of the TRIPS Agreement which sets minimum standards of protection for intellectual property.<sup>86</sup> Article 10(1) of the TRIPS Agreement provides that computer programs are to fall into the ambit of literary works as defined in the Berne Convention. Similarly, the WIPO Copyright Treaty<sup>87</sup> stipulates that:

---

<sup>83</sup> Berne Convention for the Protection of Literary and Artistic Works, 9 September 1886, *as last revised at Paris on 24 July 1971*, 1161 U.N.T.S. 30.

<sup>84</sup> Such states include South Africa, the US and the UK.

<sup>85</sup> Tong op cit note 3 at 267.

<sup>86</sup> Ibid.

<sup>87</sup> WIPO Copyright Treaty (WCT), 20 December 1996, *S. Treaty Doc. No. 105–107 (1997)*, 36 I.L.M. 65 (1997), Article 4.

‘Computer programs are protected as literary works within the meaning of Article 2 of the Berne Convention. Such protection applies to computer programs, whatever may be the mode or form of their expression.’

It is for this reason that most countries which are party to these international legal instruments tend to classify computer programs as literary works. This includes the USA and the UK. In South Africa, however, computer programs are categorised as a *sui generis* work separate and distinct from literary works.<sup>88</sup> Computer programs under the Copyright Act are consequently treated differently, in circumstances, to literary works. As a minimum, however, the rights enumerated in the Berne Convention in relation to literary works must attach to computer programs in our local statute as the TRIPS Agreement makes it so.

What follows is a brief description of the requirements for copyright to subsist under the SA Copyright Act.

### *2.2.3 Requirements for Copyright Eligibility per South African Law*

In order for a creation to be eligible for copyright protection, it must satisfy certain criteria. Chief amongst these is that it must be a work.<sup>89</sup> The SA Copyright Act does not offer a definition for a work, but rather offers nine classes of works.<sup>90</sup> Contrary to the norm in international treatment of literary works<sup>91</sup>, computer programs are expressly excluded from the ambit of this definition in the SA Copyright Act<sup>92</sup> and are separated into their own category.<sup>93</sup>

A creation must meet other requirements as well in order to attract copyright protection. These include that the author is a qualified person<sup>94</sup>, and that the work is

---

<sup>88</sup> See SA Copyright Act op cit note 41, s1(1)(g); De Villiers op cit note 18 at 320.

<sup>89</sup> SA Copyright Act op cit note 41, s2(1).

<sup>90</sup> Owen Dean and Alison Dyer *Dean & Dyer: Introduction to Intellectual Property Law* (2014) 8.

<sup>91</sup> Most parties to the Berne Convention, including the US and the UK, protect computer programs as literary works.

<sup>92</sup> SA Copyright Act op cit note 41, s1(1)(g).

<sup>93</sup> Computer programs are dealt with in greater detail below.

<sup>94</sup> The SA Copyright Act confers copyright upon every eligible work which:

‘the author or, in the case of a work of joint authorship, any one of the authors at the time [that] the work or a substantial part thereof is made, is a qualified person[.] That is:

(a) in the case of an individual, a person who is a South African citizen or is domiciled or resident in the Republic; or

(b) in the case of a juristic person, a body incorporated under the laws of the Republic.’

in material form.<sup>95</sup> When a computer program is written down, either on paper or electronically, it is already in a material form. The reason that a work must be in material form harks back to the ideas-expression dichotomy: there is no copyright in ideas. In *Galago Publishers*<sup>96</sup> the SCA stated:

'Ideas, thoughts and facts merely existing in a man's brain are not 'works', and in that form, are not within the Copyright Act ...'<sup>97</sup>

This is an integral theme which finds emphasis in this paper as it is used to distinguish the components of an API which should be copyrightable from those which shouldn't. Even if a creation fits into the ambit of one of the established categories of works, and it is created by a qualified person, it will not be eligible for copyright if it is deemed to be a mere idea.<sup>98</sup> The requirement of materiality solves this problem in relation to the literal components of a work. It is a more complicated issue, however, when the non-literal components of a work are considered as they cannot be put into material form.

Lastly, once it has been adduced that a creation is a work in a material form, the primary requirement for copyright to subsist is originality of the work.<sup>99</sup> The courts have found that a work will be original where it has not been merely copied, and constitutes an application of the author's own skill and labour.<sup>100</sup>

## 2.3 Approach to APIs in South Africa

### 2.3.1 Introduction

Statutory, judicial and academic writing on APIs in South Africa is scant. One is left to deduce from the existing positions in relation to computer programs in order to postulate as to what the law may be with regards to APIs. In light of this limitation, the current approaches as they may relate to APIs in South Africa are determined, both in

---

<sup>95</sup> The SA Copyright Act op cit note 41, s44(1) provides that a work is deemed to have been made only at the time when it has first been written down, recorded or reduced to a material form.

<sup>96</sup> *Galago Publishers (Pty) Ltd. and Another v Erasmus* 1989 (1) All SA 431 (A).

<sup>97</sup> *Ibid* at 21.

<sup>98</sup> See SA Copyright Act note 41, s2(3).

<sup>99</sup> *Ibid*, s2(1); see *Haupt* supra note 23 para 26; see Dean and Dyer note 90 at 16.

<sup>100</sup> *Ibid*.

relation to their copyrightability and as to whether they may qualify under the exceptions to infringement in the SA Copyright Act.

### 2.3.2 *Copyrightability*

A computer program is defined as a set of instructions which are fixed or stored in any manner and which direct the operations of a computer to bring about a result.<sup>101</sup> As is well documented in this text, South Africa does not protect computer programs as literary works. Unlike the case in both the USA and the UK, in South Africa computer programs are afforded their own protection under a *sui generis* category of the same name.<sup>102</sup>

The SA Copyright Act provides an extensive list of exclusive rights which attach to the owner of the copyright in a computer program. The rights of reproduction<sup>103</sup> and adaptation<sup>104</sup> will be most relevant to developers who seek to establish copyright protection over the APIs in which they own rights. This involves using and replicating programs in part and in their entirety, including their non-literal elements. The SA Copyright Act does not address APIs and South African courts have yet to have occasion to pronounce on the copyrightability of APIs. This means that there is a vacuum in South African law in this regard.

### 2.3.3 *Exceptions to Infringement*

What appears from the SA Copyright Act is that the fair dealing provisions are similar to those available in the UK Copyright Act, even though both statutes treat computer programs as different types of works. The SA Copyright Act does not permit reverse engineering.<sup>105</sup> This denies users the benefit of disassembling the API to discover its workings and use the knowledge to build a program of their own which has the same function. The SA Copyright Act does provide exceptions to infringement:

---

<sup>101</sup> SA Copyright Act op cit note 41, s1(1).

<sup>102</sup> Ibid, s2(1)(i).

<sup>103</sup> Ibid, s11B(a).

<sup>104</sup> Ibid, s11B(f).

<sup>105</sup> Ncube op cit note 9 at 442, which defines reverse engineering as: 'a variety of practices undertaken to understand how a software program is built and how it achieves its functionality'.

'[C]opyright in a computer program shall not be infringed by a person who is in lawful possession of that computer program, or an authorized copy thereof, if—

(a) he makes copies thereof to the extent reasonably necessary for back-up purposes; [or]

(b) a copy so made is intended exclusively for personal or private purposes; and

(c) such copy is destroyed when the possession of the computer program in question, or authorized copy thereof, ceases to be lawful.'<sup>106</sup>

The SA Copyright Act also excepts certain other acts from infringement, including those done for the purposes of criticism or review of a work<sup>107</sup>; or for reporting current events<sup>108</sup>, reporting judicial proceedings<sup>109</sup>, or for teaching.<sup>110</sup> This reflects the provisions in the UK Copyright Act, and creates similar issues. These issues relate to the fact that the narrow ambit of the categories of acceptable uses precludes computer programs, and by extension, APIs. Interestingly, section 12(3) of the SA Copyright Act provides:

'The copyright in a ... work which is lawfully available to the public shall not be infringed by any quotation therefrom ... Provided that the quotation shall be compatible with fair practice, that the extent thereof shall not exceed the extent justified by the purpose and that the source shall be mentioned, as well as the name of the author if it appears on the work.'

This raises the question: what does it mean to 'quote' in the context of computer programs?

### 2.3.3.1 *Quotations as a Fair Dealing Exception: Potential Applicability of s12(3)*

The SA Copyright Act identifies the quotation exception as applicable not only to literary and artistic works, but to computer programs as well.<sup>111</sup> It is a general law of

---

<sup>106</sup> SA Copyright Act op cit note 41, s19(2).

<sup>107</sup> Ibid, S12(1)(b).

<sup>108</sup> Ibid, s12(1)(c).

<sup>109</sup> Ibid, s12(2).

<sup>110</sup> Ibid, s12(4), subject to the use being compatible with fair practice and the source being mentioned with the name of the author (if such name appears on the work).

<sup>111</sup> Section 19(1) of the SA Copyright Act provides: 'the provisions of section 12 (1) (b) and (c), (2), (3), (4), (5), (12) and (13) shall mutatis mutandis apply, in so far as they can be applied, with reference to computer programs.'

interpretation that the legislature did not intend any provision to create an absurdity.<sup>112</sup> The inclusion of this specific exception must reasonably show that the legislature specifically intended for it to be a lawful exception when a user 'quotes' from a computer program.

This brings certain questions to the fore. What does it mean to 'quote' from a computer program? Which part of the program is quotable? What constitutes 'fair practice' in relation to quoting from computer programs; and what kind of purposes are acceptable justifications for 'quoting' the work? Put differently, can a computer program be quoted for any purpose (such as using the 'quote' for functional purposes) or is there a limit to the types of reasons that a computer program may be quoted. Unfortunately, these questions have yet to be explored in South African jurisprudence. The provision remains open to interpretation.

The context of the section 12 exceptions, with the Constitutional mandate for fair and just law in mind, is instructive. The quotations exception in particular was mandated by the Berne Convention which provides:

'It shall be permissible to make quotations from a work which has already been lawfully made available to the public ... including quotations from newspaper articles and periodicals in the form of press summaries.'<sup>113</sup>

The Berne Convention was created specifically to provide minimum standards of protection for literary works. Though the TRIPS Agreement subsequently gave computer programs the same minimum protection as those granted to literary works under the Berne Convention<sup>114</sup>, it is a fact that the Berne Convention was not written with computer programs in mind.<sup>115</sup> It is apparent that this is the reason that the examples provided in the Article are 'newspaper articles and periodicals in the form of press summaries'.<sup>116</sup> Nevertheless, discerning the literal, plain meaning of the key word in the provision may provide guidance as to how computer programs may fit into this exception. The Cambridge Online Dictionary defines the word 'quote' as

---

<sup>112</sup> *Poswa v MEC for Economic Affairs, Environment and Tourism, Eastern Cape* 2001 (3) SA 582 (SCA) para 10 – 14.

<sup>113</sup> Berne Convention op cit note 83, Article 10(1).

<sup>114</sup> See Tong op cit note 3 at 70 for a salient discussion on protection having to be 'the same as' that provided for literary works.

<sup>115</sup> Ibid at 267.

<sup>116</sup> Berne Convention op cit note 83, Article 10(1).

‘repeat[ing] the words that someone else has said or written’.<sup>117</sup> The word is also defined as ‘to repeat words from a text ... written by another person.’<sup>118</sup> The basic, literal meaning of the word means to copy words written by somebody else.

Given the above, it seems that it is the source code of a computer program which a court might find quotable. This is the text which is legible to humans and is translated to the machine-readable text which is known as the object code. Collectively they cause the computer to perform a function. This seems uncontroversial, as it fits neatly within the range of things that one can ‘quote’ even when consideration is given to the plain meaning of the word ‘quote’. This means that the quotation exception may apply to APIs insofar as their literal component is concerned. The same cannot be said, however, for the non-literal components. A purposive interpretation of statute does not equate to an unwieldy bound beyond the confines of reason and logic.<sup>119</sup> The history of the origins of the provision, as well as the impossibility of quoting abstract devices such as structures, sequences, functions or methods make it clear that, at best, the quotation exception could allow users of computer programs to only quote the written code.

It will be open to a court to decide what a user may permissibly quote an API for, as well as the extent to which this quoting may occur. The common purpose of quoting from the work of another is usually for the purpose of commentary. It is difficult, then, to see how this may be of use to a person who seeks to use the API of another programmer in the development of her own software. The position in relation to APIs, then, is not encouraging. Those who seek to use APIs without acquiring a licence from the copyright owner will not find much recourse in the statute as it is.

## CHAPTER 3: API'S AND COPYRIGHTABILITY

### 3.1 Introduction

---

<sup>117</sup> Cambridge Dictionary ‘Quote’ available at <https://dictionary.cambridge.org/dictionary/english/quote> accessed on 12 February 2018.

<sup>118</sup> English Oxford Living Dictionaries ‘Quote’ available at <https://en.oxforddictionaries.com/definition/quote> accessed on 12 February 2018.

<sup>119</sup> See note 112 for a recount as to how an interpretation may not be absurd.

The manner in which APIs are treated in other jurisdictions offers guidance as to what the South African approach ought to be. In the USA, the question of whether APIs should be copyrightable has been determined by the courts in recent times, in the matter involving Oracle and Google.<sup>120</sup> The USA law as applicable to computer programs, insofar as it is relevant, will appear from the case discussions. The reasoning and findings of the court *a quo* are delved into below, before the appeal finding is considered. For purposes of clarity, the matter as handled by the court *a quo* will be referred to as *Oracle 1*, whereas the judgment of the Federal Court of Appeals (hereafter the Federal Court) will be referred to as *Oracle 2*. This is compared to the European approach, with specific focus on the UK.

Finally, an argument is made as to a position appropriate for South Africa. Though there is no case law dealing with APIs in South Africa, consideration of the local treatment of computer programs and copyright, issues of the rights of the copyright owner and the public interest against a constitutional background, as well as analysis of the approaches in the abovementioned foreign jurisdictions are used to support a proposal as to the stance that should be taken in South Africa.

## 3.2 USA Approach

### 3.2.1 Summary of USA Approach

APIs are eligible for copyright in the USA.<sup>121</sup> It has been found, at the level of the Federal Court, that the literal components of an API may be copyrightable; and that the non-literal components of an API may constitute an expression which is worthy of copyright protection. The stance in relation to the latter is as it is for the reason that the structure, sequence and organisation of an API can only fall short of copyright eligibility where it constitutes a mere idea.<sup>122</sup> This will be the case where no creative input on the part of the creator of the work is identified. Creative input requires that the programmer incorporate elements into the structure and organisation of the work which are more than the absolute minimum which is necessary to achieve the intended

---

<sup>120</sup> This includes both of the Oracle cases. See *supra* note 6 and 7.

<sup>121</sup> Due to the finding in *Oracle Appeal* *supra* note 7.

<sup>122</sup> *Ibid* at 43.

function of the work.<sup>123</sup> In other words, the structure of an API will constitute an expression where it contains more than the minimum required to achieve its function.<sup>124</sup>

This position is controversial and has been duly received with much criticism in American legal commentary. The prior position, as is outlined below in *Oracle 1*, has been much preferred by advocates of open access.<sup>125</sup> This position is to the effect that APIs cannot be eligible for copyright insofar as their structure and organisation is to achieve a functional purpose.<sup>126</sup> That means that any changes to the non-literal components of an API are not creative, but rather utilitarian because the reason for the change is to improve the functioning of the API.<sup>127</sup>

What follows below is a discussion of the *Oracle* matter, including the development of the law over time in order to paint the background against which the legal reasoning and conclusions occur. What appears from the case discussion is that the question of what constitutes an idea and what constitutes an expression is the issue at hand.

JAVA was developed by a company named Sun Microsystems. It is a virtual platform which enables developers of software to write programs that are operable on different types of computer hardware without a need for the programs to be re-written for each different type.<sup>128</sup> It consists, amongst other things, of a set of pre-written programs which enable various commands to be carried out. These pre-written programs comprise the JAVA API. Prior to proceedings, the JAVA API consisted of 166 packages, which were divided into 600 classes which could carry out more than 6000 methods.<sup>129</sup>

---

<sup>123</sup> Ibid at 64.

<sup>124</sup> Ibid.

<sup>125</sup> See Peter S. Menell 'API Copyrightability Bleak House: Unraveling and Repairing the Oracle v Google Jurisdictional Mess' (2016) 31 *Berkely Technology Law Journal* 1515 for a full criticism on the *Oracle 2* finding.

<sup>126</sup> *Oracle* supra note 6 at 30.

<sup>127</sup> *Oracle* supra note 6 at 22.

<sup>128</sup> Ibid at 4.

<sup>129</sup> Ibid at 5.

What is relevant is that Google replicated the lines of code, known as 'declarations' or 'headers', which specified the names, functions and parameters of the methods and classes of 37 packages of the JAVA API to use in its own Android Platform.<sup>130</sup> However, Google used its own, different, code to implement the methods and classes of the packages. Oracle, the plaintiff, bought JAVA from Sun Microsystems then made a claim of copyright infringement against Google for its replication of the 37 packages. The Court proceeded with the view that both parties accepted that Google had not literally copied the JAVA software, but rather, it had come up with its own way of implementing the 37 API packages.

The issue that the court had to deal with was whether the 'structure, sequence and organisation' of the 37 elements of the Java API which Google had copied were eligible for copyright, as well as a claim that Google had unlawfully copied the names of the methods and classes and packages. Put differently, the court had to determine whether Google was free to replicate the names, organisation of names, and functionality of the 37 packages of the JAVA API.

### 3.2.2 *Oracle v Google 1*

#### 3.2.2.1 *Findings of the Court*

The court finds that, so long as the code used to implement each method in the JAVA API is original and belongs to its developer, anyone may write code to implement the methods. Copyright law does not extend a right over any of the ways to implement a method or a function. The method specification is the idea, whereas the method implementation is the expression in this case. On this basis, the court holds that there is no copyright in a JAVA method specification.<sup>131</sup>

Furthermore, in order to carry out the specific functions of the JAVA API, it is necessary that the method specification is identical to the one set forth in the declaration of the JAVA API, per the JAVA rules.<sup>132</sup> For this reason, there are no alternative methods to achieve the same result. On this basis, the application of the

---

<sup>130</sup> Ibid.

<sup>131</sup> Ibid at 35.

<sup>132</sup> Ibid at 7.

merger doctrine means that there is no copyright infringement in Google's use of identical declarations to the JAVA API declaration.<sup>133</sup> Due to the name doctrine, the court concludes that there is no copyright infringement caused by Google's use of method names which are identical to those found in the JAVA API.<sup>134</sup> Lastly, the court finds that Google is permitted to organise its methods in the same way as they were grouped in the JAVA API, because the overall set-up constituted a command structure for a method of operation of the API.<sup>135</sup>

In other words, that particular structure carried out the function of enabling the API to operate. It allowed for interoperability because millions of lines of code written by other programmers are written using that command structure, meaning that these lines of code would not be able to run on Android unless Android also made use of that command structure.<sup>136</sup> The fact that it has a functional use of interoperability illustrates, as argued by the court, that the organisation of the methods of the JAVA API are a functional system or method of use, which disqualifies them from copyright eligibility per the US Copyright Act.<sup>137</sup>

### 3.2.2.2 Reasoning of the Court

The judgement began with a consideration of the plaintiff's claim of copyright over the names (and other such short-phrases) of methods, classes and packages. The court noted the rule as espoused<sup>138</sup> by the US Copyright Office and followed in *Sega*.<sup>138</sup> It stipulates that 'copyright law does not protect names, titles [or] short-phrases.'<sup>139</sup>

The court then turned to the development of the law in order to deal with the more complex question of the copyrightability of computer programs and their structure, sequence and organisation. In the seminal case *Baker*<sup>140</sup>, which pre-dated the advent of computers, the Supreme Court found that the copyright in a literary work of mathematical science cannot give the author exclusive rights over the methods

---

<sup>133</sup> Ibid.

<sup>134</sup> Ibid at 35.

<sup>135</sup> Menell op cit note 125 at 1553.

<sup>136</sup> *Oracle* supra note 6 at 5.

<sup>137</sup> See US Copyright Act note 43, s102(b).

<sup>138</sup> *Sega Enterprises. Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992).

<sup>139</sup> *Oracle* supra note 6 at 14.

<sup>140</sup> *Baker v Seldon* 101 U.S. 99 (1879).

contained therein.<sup>141</sup> The US Copyright Act, enacted almost one hundred years after the *Baker* case, provides that computer programs are to be treated as literary works and creates a limitation on computer programs which is similar to that which was found in *Baker*:

‘In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.’<sup>142</sup>

The court then turned to the House Report<sup>143</sup> accompanying this provision for guidance. It is necessary to quote it to its full extent in order to capture the fullness of the reasoning behind the court’s interpretation of the provision:

‘Some concern has been expressed lest copyright in computer programs should extend protection to the methodology or processes adopted by the programmer, rather than merely to the ‘writing’ expressing his ideas. Section 102(b) is intended ... to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law.’<sup>144</sup>

The court found that this served to entrench the well-established dichotomy between ideas and expressions in copyright. Mere ideas are available in the common pool of knowledge to be utilised by whomsoever; whereas actual expressions by a competent author in the correct circumstances attract copyright protection. The court then noted the National Commission on New Technological Uses of Copyrighted Works (CONTU)<sup>145</sup>, which found that copyright does not protect the electromechanical functioning of a machine. Consequently, it concluded that the preclusion from copyright protection of ‘procedures, processes and methods of operation’ found in the US Copyright Act was not at odds with the definition of computer programs in the US Copyright Act.<sup>146</sup>

Finally, the court emphasised the CONTU recognition of the ‘merger rule’. This rule is designed to deal with instances where the idea and expression of a creation

---

<sup>141</sup> Ibid.

<sup>142</sup> See US Copyright Act note 43, s102(b).

<sup>143</sup> House Committee Reports are produced by Congress in the USA.

<sup>144</sup> *Oracle* supra note 6 at 17.

<sup>145</sup> National Commission on New Technological Uses of Copyrighted Works, Final Report (1979).

<sup>146</sup> See US Copyright Act, s101 which states: ‘A “computer program” is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.’

are closely interlinked.<sup>147</sup> It provides that language that is copyrighted may be lawfully copied where there are few alternatives to the language to make an expression of an idea.<sup>148</sup> In other words, where there are limited options in copyrighted language to express something, it will not constitute an infringement for a third party to copy that language.<sup>149</sup> In the realm of computer programs, the effect of this doctrine is that copyrighted specific instructions may be copied if they are the only and essential means of achieving a result.<sup>150</sup> This must be determined on a case-by-case basis. This concept has been built-upon in case law.

In *Whelan*<sup>151</sup>, the Court of Appeals developed a framework to deal with the non-literal copying of computer software. It stated that the purpose or function of a utilitarian work would constitute the work's idea, and everything else unnecessary for the purpose of the work would form a part of the expression of the idea.<sup>152</sup> The structure of the program in this case was found to be copyrightable because there was a multiplicity of ways to structure a program in order to achieve the desired result.<sup>153</sup> Due to the high bar it sets for infringement, the *Whelan* case, though instructive, has not been followed.<sup>154</sup> What has been followed is the abstract-filtration-comparison' test as adopted in *Computer Associates International*.<sup>155</sup> This test focuses on whether there has been substantial copying of the copyright material or not. The abstract-filtration-comparison test operates by removing all elements of the work which are not copyrightable, and comparing the remaining portion using the usual test for substantial similarity.<sup>156</sup>

First, the copyright program is divided into its structural components.<sup>157</sup> Next, all structures which are not copyrightable are excluded from comparison. Such structures are those which are primarily functional; those which were designed to

---

<sup>147</sup> Deba Alam 'Oracle v Google' (2015) 26 *DePaul Journal Art, Technology and IP Law* 39 at 45 – 46.

<sup>148</sup> *Ibid.*

<sup>149</sup> Lerinda Saint Waltrip 'Copyright Law – the Idea/Expression Dichotomy: Where Has It Gone' (1986) 11 *Southern Illinois University Law Journal* 411 at 418.

<sup>150</sup> *Oracle* supra note 6 at 18 -19.

<sup>151</sup> *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222 (3d Cir. 1986).

<sup>152</sup> *Ibid.*

<sup>153</sup> *Oracle* supra note 1 at 20.

<sup>154</sup> Menell op cit note 125 at 1526.

<sup>155</sup> *Computer Associates International, Inc. v. Altai*, 982 F.2d 693 (2d Cir. 1992).

<sup>156</sup> *Ibid* at 705.

<sup>157</sup> *Oracle* supra note 1 at 23.

optimise the speed and efficiency of the program, as well as its simplicity.<sup>158</sup> These structures include not only those the design of which is dictated by efficiency (which encompasses ease of use)<sup>159</sup>; but also, those already in the public domain, such as elements of a computer program that have been freely accessible. As was defined in *Gates Rubber*<sup>160</sup>, structures are ‘a description of how the program operates in terms of its various functions’.<sup>161</sup> In *Lotus*<sup>162</sup> the court found that, because a program’s menu command hierarchy was essential in making use of the program’s functional capabilities, it constituted a ‘method of operation’, which could not be copyrighted.<sup>163</sup> This was so even though the programmer had made ‘expressive’ choices as to what the names of the command terms should be; and how they should be structured in their hierarchy.

Given the above, the court in *Oracle 1* confirmed that non-literal components of a program can be protected under copyright where the structure, sequence and organisation qualify as an expression of an idea, as opposed to actually being an idea.<sup>164</sup> The court also supported the precedent set in *Atari v Nintendo*<sup>165</sup>, where it was found that a program was eligible for copyright if it had organisation and sequencing which were more than what was necessary to achieve the intended function of the program.<sup>166</sup> As they were unnecessary, they were deemed to be creative elements eligible for copyright. This is an indication that non-literal elements of a program that are not necessary for the purposes of utility constitute an expression. The court noted that the judgments in *Sega Enterprises*<sup>167</sup> and *Connectix*<sup>168</sup> which confirmed that interface procedures which enable interoperability between programs are functional aspects which are not copyrightable per the US Copyright Act.

Based on its consideration of the development of the law, the court found that there were principles which were relevant to the issue before it. Firstly, under the

---

<sup>158</sup> Ibid at 24.

<sup>159</sup> Ibid at 22.

<sup>160</sup> *Gates Rubber Co. v. Bando Chemical Industries, Ltd.*, 9F.3d 823 (10<sup>th</sup> Cir. 1993).

<sup>161</sup> Ibid.

<sup>162</sup> *Lotus Development Corp. v. Borland International, Inc.*, 49 F.3d 807 (1<sup>st</sup> Cir. 1995).

<sup>163</sup> Ibid.

<sup>164</sup> *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9<sup>th</sup> Cir. 1989).

<sup>165</sup> *Atari Games Cor. v. Nintendo of America Inc.*, 975 F.2d 832 (Fed. Cir. 1992).

<sup>166</sup> Ibid.

<sup>167</sup> *Sega* supra note 138.

<sup>168</sup> *Sony Computer Entertainment, Inc., v Connectix Corporation*, 203 F.3d 596 (9<sup>th</sup> Cir. 2000).

merger doctrine, there can be no copyright in an expression which lacks no or few alternatives. Secondly, the names doctrine dictates that names and short-phrases may not be eligible for copyright. Thirdly, as provided by the US Copyright Act, copyright protection does not extend to any procedure, process, system or method of operation in any form.<sup>169</sup> Further implications from the case law which the court found persuasive include that, where the non-literal elements of a program are not necessary to perform its function, such elements will be deemed expressions. This has the effect that there is no copyright eligibility for a non-literal purely functional component of a work.

### 3.2.2.3 *Analysis of Court Reasoning*

The reasoning of the court was grounded in the functional attributes of the JAVA API.<sup>170</sup> Judge Alsup, who presided over the matter, acknowledged that the overall structure of the JAVA API was creative; however, its functions nevertheless constituted a command structure designed to carry out specific functions.<sup>171</sup> Following the CONTU guidance as well as his interpretation of the merger doctrine, Alsup J found that there was no creative expression in the structure, sequence and organisation as it was the only and essential means for Google to perform the task.<sup>172</sup>

This reasoning seems sound on a legal basis insofar as its underlying principle seems to be that there can be no creative expression where there is no choice in the manner of creating the work. It cannot be the case that a creation can qualify as an expression where there is little choice in how such creation could be made. The conclusion is the same whether the question of freedom of choice in making the work is viewed from the perspective of the author or from the alleged infringer.

The court's position is also correct to the extent that it finds that aspects of a work which serve no function properly constitute an expression. This is sensible because, if creativity is used to determine expressiveness, then utility must be indicative of a non-expressive component of a work. My opinion is that the question of functionality is indeed central to the task of separating an idea from an expression. As such, I submit that the court is correct in its view that the interoperability of an API is a

---

<sup>169</sup> See US Copyright Act op cit note 43, s102(b).

<sup>170</sup> Mennel op cit note 125 at 1553.

<sup>171</sup> Ibid.

<sup>172</sup> Ibid at 1554.

factor which indicates that the non-literal components of the API are functional. It will become clear forthwith that the Federal Court in this matter differs markedly in its opinion. Nevertheless, it seems that the reasoning employed by Alsup J is sound and instructive as to the direction that ought to be followed in South Africa.

#### 3.2.2.4 *Implications of Judgment*

The immediate implications of the *Oracle* case were that Google, and the community at large, were able to freely write their own implementations for the purpose of carrying out the exact same functions of all methods in the JAVA API, using exactly the same method specifications and names.<sup>173</sup> Effectively, APIs became freely available to the developer community. This is a benefit to innovation as APIs are integral to speedy software development; and it curbs the onset of a market monopoly.

The impact of this judgment on persons who develop APIs is that they would not stand to gain a direct financial benefit from developing APIs, which may call into question whether there is sufficient incentive for the development of APIs. However, this is no issue in relation to incentivisation in the developer community as financial incentives are rarely the prime drivers of software development.<sup>174</sup> This is made evident by the robust existence of the open-source movement which encourages and participates in the free sharing of software resources for development. More was to follow, however, as Oracle appealed the case. In the Federal Court, the ruling of the court a *quo* was overturned wholly.

### 3.2.3 *Oracle v Google 2*

#### 3.2.3.1 *Findings of the Court*

The Federal Court holds that the court a *quo* was incorrect in all of its findings, including with regards to the copyrightability of the declaring source code as well as

---

<sup>173</sup> Ibid.

<sup>174</sup> See Lemley note 31 at 140.

the structure, sequence and organisation of the API packages, and its reasoning in considering interoperability in the context of determining copyrightability.<sup>175</sup>

### 3.2.3.2 Reasoning of the Court

With regards to the issue of the copyrightability of the declaring source code, the court accepted the plaintiff's argument that the court *a quo* misapplied the merger doctrine and incorrectly focused its analysis on the options available to Google at the time of infringement, rather than on the rights of the author at the creation of the JAVA APIs.<sup>176</sup> The court accepted the argument that the merger doctrine bars copyright protection for lines of source code only where the author had just one way, or a very limited number of ways, to write them. It found that the evidence showed that the author had a multitude of ways to select and arrange the lines of code that were copied by Google.<sup>177</sup> Secondly, the court reiterated that the focus of analysis must be on the options available to the author at the time that the API packages were created. Google had not argued on this basis.<sup>178</sup> For these two reasons, the court rejected that Google could succeed on the basis of the merger doctrine.

The court then turned to the court *a quo*'s finding that short-phrases are not copyrightable on the basis that the Copyright Act provides that short-phrases can never be eligible for copyright. The court stated that this finding was wrong because it failed to consider that the real issue which must be considered in this regard is whether those short-phrases are creative or not. The court found that, when viewed in their entirety of over 6000 or 7000 lines of code, the short-phrases did embody an element of creativity.<sup>179</sup> This decision indicates that, even in an instance concerning the copying of names, the ideas-expression dichotomy must still be at the forefront of a courts interpretation.

The court then turned to the main issue concerning the structure, sequence and organisation of the API packages. It found that, because the API packages could have been organised in many ways and still have achieved the same function, and thus

---

<sup>175</sup> *Oracle Appeal* supra note 7 at 5.

<sup>176</sup> *Ibid* at 24.

<sup>177</sup> *Ibid* at 30.

<sup>178</sup> *Ibid* at 33.

<sup>179</sup> *Ibid* at 33 – 34.

were original and creative, the Copyright Act's bar on methods of operation did not exclude them from copyright eligibility.<sup>180</sup> This is because the creative nature of the organisation, as viewed by the court, rendered it an expression and not a mere functional aspect which effectively amounts to an idea.<sup>181</sup> Lastly, the court found that the court a *quo*'s reliance on interoperability to show that the organisation of the packages was a command structure for the purpose of achieving a function was wrong, because determinations of interoperability are related to the question of fair use, and not copyrightability.<sup>182</sup> In any case, it found that Google would still fail because Google's copying of the organisation was not the only and essential means of achieving interoperability; rather, the court found, Google wanted to take advantage of the fact that most programmers were well versed and used to writing according to the JAVA API packages.<sup>183</sup>

For these reasons, the reasoning of the court a *quo* was found to be wrong, and the JAVA API packages were held to be copyrightable in their entirety. This means that, as the law stands in the USA, the non-literal components of APIs can be eligible for copyright even though they are designed to achieve a function.

### 3.2.3.3 *Analysis of Court Reasoning*

The discussion of the Federal Court carried a theme of 'software as creative expression.'<sup>184</sup> It noted the concession of a Sun Microsystems software engineer who was of the opinion that there can be creativity in a single method declaration<sup>185</sup>; and likened a method declaration to a short phrase in a Charles Dickens novel.<sup>186</sup> This view reflected in the way that the Federal Court interpreted the law.

It has been argued, with good reason, that the Federal Court was wrong in law. Specifically, in its judgement it misinterpreted section 102(b) of the US Copyright Act,

---

<sup>180</sup> Ibid at 42.

<sup>181</sup> Ibid at 21.

<sup>182</sup> Ibid at 45.

<sup>183</sup> Ibid at 50 - 51.

<sup>184</sup> Menell op cit note 125 at 1555.

<sup>185</sup> Ibid.

<sup>186</sup> Ibid.

misconstrued the software copyright jurisprudence of the Ninth Circuit<sup>187</sup>, conflated expressive and technological "creativity" and applied an overly rigid approach to the limiting doctrines of USA copyright law.<sup>188</sup> Section 102(b) excludes ideas and processes from copyrightability. The court recognised these limitations but failed to distinguish between the question of what is copyrightable and the question of the scope of actions which constitute infringing activity.<sup>189</sup> The incorrect view of the court was that the section was applicable at the inquiry pertaining to infringement, and not copyrightability.<sup>190</sup> This is clear as Google argued that certain compilations of functions in the JAVA API were methods of operation which the provision excluded from copyright, but the Federal Court stated that the section merely restated the basic ideas-expression dichotomy and did not exclude methods.<sup>191</sup> However, the dichotomy in USA law is traceable to the *Baker* case, which dealt with a literary work but set the underlying principle.<sup>192</sup> Here, the court in that matter categorically excluded any method of accounting from copyrightability, though it found that the actual accounting book describing the method was eligible for copyright protection.<sup>193</sup>

Secondly, the Federal Court embraced principals that the Ninth Circuit had rejected. Its decision to find that interoperability may be protectable under copyright contradicts the position of the Ninth Circuit which states that copyright law does not function to preclude interoperability.<sup>194</sup> Thirdly, the Federal Court's view that the creativity needed in the design of an API constitutes an expression conflates expressive technology as applicable in copyright law and technological 'creativity'.<sup>195</sup> Programmers need to reproduce the package, class and method names of a particular API in order to ensure that their programs 'respond to the same inputs and produce the same outputs' of the software they are operating with.<sup>196</sup> That is so because the construction of those package, class and method names are purely functional:

---

<sup>187</sup> The jurisprudence as set out in previous matters of the Ninth Circuit Federal Court of Appeals.

<sup>188</sup> Menell op cit note 125 at 1562.

<sup>189</sup> Ibid at 1563.

<sup>190</sup> Ibid at 1562.

<sup>191</sup> Ibid at 22.

<sup>192</sup> See development of law in section 3.2.2.2 of this paper.

<sup>193</sup> Menell op cit note 125 at 1563; David R. Owen 'Interfaces and Interoperability in *Lotus v. Borland*: A Market-Oriented Approach to the Fair Use Doctrine' (1996) 64 *Fordham Law Review* 2381 at 2386.

<sup>194</sup> The decisions of the Ninth Circuit's *Sega* supra note 138 and *Sony* supra note 168 conclusively held that the code necessary for interoperability is uncopyrightable; Menell op cit note 125 at 1565.

<sup>195</sup> Menell op cit note 125 at 1567.

<sup>196</sup> Ibid.

everything in their composition is designed to produce a particular result. To consider the ingenuity that goes into an APIs functionality design as a 'creative expression' serves only to conflate expressions with ideas.<sup>197</sup> It cannot be correct.

South Africa is not bound by foreign law. Nevertheless, the fact that the decision of the Federal Court was based on obtuse reasoning undermines the persuasiveness of the decision. It would not be wise to derive guidance from a result borne of illogical thinking. Furthermore, the implications of the judgment erode its appeal.

#### 3.2.3.4 *Implications of Judgment*

This judgment makes it possible that there are aspects of the non-literal elements of APIs which may be eligible for copyright, even though these arcane pieces of technology operate primarily to complete a simple function. Lawsuits may be brought against technology start-up companies and open source projects which have copied APIs for the purpose of ensuring that their software is compatible with other products. Defending an infringement case is expensive, so the practical implications of this judgement are prejudicial to players who lack financial strength. This in turn will curb use of APIs, as a preventative measure against possible lawsuits.

It is reasonably foreseeable that this ruling has created a setting for the entrenchment of copyright monopolies in the software industry. This market dominance is similar to that which is created by patent laws, but endures for many more years under copyright than under the law of patents.<sup>198</sup> This is so even though it is much easier to gain this market dominance through copyright than it is through patent law (which requires stringent criteria to be met). Menell argues that 'copyright's low threshold for protection, complex scope of protection ... and long duration, [overprotect] software and thereby undermin[e] technological innovation and competition.'<sup>199</sup> This is relevant because computer software, unlike other literary works,

---

<sup>197</sup> Ibid.

<sup>198</sup> Patents typically last for 20 years, after which they must be renewed at cost to the rights holder – TRIPS Agreement op cit note 19, Article 33; whereas copyright subsists for at least as long as the lifetime of the copyright owner – Berne Convention op cit note 83, Article 7.

<sup>199</sup> Menell op cite note 125 at 1520.

serves a functional purpose akin to patentable inventions. The implication created by this ruling is like using the law of copyright to govern inventions: creating monopolies over technology becomes easier, and said monopolies are more robust under copyright than they would be under the law of patents.

### 3.3 UK Approach

#### 3.3.1 *Summary of UK Approach*

The position in the United Kingdom contrasts to the American approach in that it has been definitively held in the UK that the functionality of a computer program (which includes APIs) cannot be eligible for copyright.

Copyright law in the UK is governed by the UK Copyright Act. It provides that computer programs are included within the definition of a literary work.<sup>200</sup> As a literary work, any computer program must be original in order to attract copyright protection. As in South Africa, the question of originality turns on the skill, labour and application of judgment in creating the work. Where the expression of a computer program is the inevitable result of its function, it will not satisfy originality.<sup>201</sup> As will be evident in the case discussion, the Information Society Directive<sup>202</sup> guides the court's treatment of computer software in law. Reference is made by the court to the Software Directive<sup>203</sup> where its provisions match those of the Council Directive.<sup>204</sup> This finding is on the basis that the structures and methods that are developed in order to enable the operations of a program cannot be deemed to be creative expressions, but rather utilitarian measures which constitute ideas in the copyright law.<sup>205</sup> The impact on society of permitting the copyrightability of the functionality of computer programs is also noted as a reason for the deviation from the approach in the USA: granting rights of copyright over the functionality of computer programs would create unfair monopolies in favour

---

<sup>200</sup> UK Copyright Act op cit note 53, s3(1)(b).

<sup>201</sup> See *Kenrick & Co v Lawrence & Co* (1890) 25 QBD 99 where the court decided that a drawing lacked sufficient 'artistic faculty' to attract copyright protection, because it could only be drawn in that way.

<sup>202</sup> Council Directive 2009/24/EC.

<sup>203</sup> Council Directive 91/250/EEC.

<sup>204</sup> Council Directive 2009/24/EC has replaced the Software Directive; however, the court opts to refer to the Software Directive as their provisions are the same.

<sup>205</sup> SAS supra note 30 para 57.

of the copyright owners to the prejudice of all others.<sup>206</sup> This is similar to the findings of the court *a quo* in the *Oracle* matter.

A greater discussion of the reasoning and findings as applied in the UK is canvassed below in the discussion of *SAS v World Programming*.<sup>207</sup> The matter concerned a dispute between the SAS Institute, the applicant, and its direct competitor which was WPL, the respondent. The SAS Institute was a program developer, and it created the SAS System which was an integrated network of analytical software programs that performed data processing and analysis tasks. The SAS System made use of Base SAS, which was a component of the SAS System that allowed its users to write programs using the SAS Language to manipulate data. The defendant developed alternative software called the World Programming System (WPS) which could interpret inputs written in the SAS language and produce the same results as the SAS System. In this way, it had developed software which could execute application programs that were written in the SAS language.<sup>208</sup> In order to develop the WPS, the defendant had to copy the functionality of the Base SAS. It did this by closely replicating a substantial amount of the information provided in the SAS Manuals, which gave technical specifications as to the external workings (but not the internal workings, such as those relating to the code structure, or the object and source codes).<sup>209</sup>

One of the claims of the applicant was that the defendant had copied its program.<sup>210</sup> The underlying issue of the matter was the extent to which a developer may lawfully replicate the functions of another computer program.

### 3.3.2 *SAS v World Programming*

#### 3.3.2.1 *Findings of the Court*

The Appeal Court notes the distinction between ideas and the expression thereof, and supports the conclusion reached by the European Court<sup>211</sup> (hereafter the CJEU) that

---

<sup>206</sup> Ibid para 76.

<sup>207</sup> Ibid.

<sup>208</sup> Ibid para 1 -3.

<sup>209</sup> Ibid para 15. The court states that the SAS Manuals are 'only a very detailed description of what the software does'.

<sup>210</sup> Ibid para 4.

<sup>211</sup> European Court of Justice.

it is the form of the expression which is worthy of copyright. This echoes the reasoning of the judgment of the court *a quo* in the *Oracle 1* case. The functionality of a computer program is not deemed to be an expression; rather, it is found that it constitutes an idea. The CJEU states that:

'Article 1(2) of Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs must be interpreted as meaning that neither the functionality of a computer program nor the programming language and the format of data files used in a computer program in order to exploit certain of its functions constitute a form of expression of that program and, as such, are not protected by copyright in computer programs ...'<sup>212</sup>

### 3.3.2.2 Reasoning of Court

The court *a quo* resolved this question using guidance from the CJEU. Though the court and both of the contesting parties accepted the finding of the court *a quo*, which stated that the copying of a computer program's functionality is not unlawful, it elected to relay the reasoning behind the finding.

The court stated that the issue boils down to the ideas-expression dichotomy. It noted the TRIPS Agreement, which is incorporated into the legal order of the European Union and thus applies in the UK, which provides that copyright protection extends to expressions and not to ideas.<sup>213</sup> The problem related to determining what, in the realm of computer programs, constituted an expression and what constituted an idea. This is the same issue that the *Oracle* judgements had to contend with, though the consideration there was specifically focused on APIs. The court considered the interpretations of the CJEU of its Directives. The Software Directive provides that:

'Whereas, in accordance with this principle of copyright, to the extent that logic, algorithms and programming languages comprise ideas and principles, those ideas and principles are not protected under this Directive.'<sup>214</sup>

Furthermore, the Council Directive states that the ideas and principles which form the foundation of any element of a computer program, including those upon which

---

<sup>212</sup> SAS supra note 30 para 70(1).

<sup>213</sup> TRIPS op cit note 19, Article 9(2); see also WIPO Copyright Treaty op cit note 87, Article 2 which also applies in the UK.

<sup>214</sup> SAS supra note 30 para 22.

its interfaces are based, are not eligible for copyright protection.<sup>215</sup> The court supported the argument made by the CJEU, however, that the object of protection is the expression of one's intellectual creation.<sup>216</sup> It based this reasoning on the finding of the CJEU in *Infopaq*<sup>217</sup> where it is stated that works such as computer programs are only worthy of copyright protection if they are the intellectual creations of their authors. An intellectual creation refers to a work which the author has made expressive and creative choices in order to produce.<sup>218</sup> This expression, it must be noted, is not limited to the object and source code. It includes all elements expressing the creativity of the author. Where the choices available to the author are restricted, it is unlikely that the work will constitute an intellectual creation (or the expression thereof).<sup>219</sup>

In lieu of the above, the court concluded that there can be no originality in the functionality of a computer program, since the range of methods to implement an idea is so limited.<sup>220</sup> The court was also motivated by policy concerns, as it stated that a contrary finding would have the effect of conferring a monopoly to a few companies. This would stymie market creation and innovation and run contrary to the goals of the Council Directive.<sup>221</sup> For this reason, the court found that:

'... if expression is dictated by technical function then the criterion of originality is not satisfied ... where that is the case, the product is not an intellectual creation of the author at all.'<sup>222</sup>

### 3.3.2.3 *Analysis of Court Reasoning*

The UK position mirrors the approach taken in the USA, which is to consider only the expressive component of a work when considering if there has been infringement. The UK approach differs from the prevailing position as espoused in *Oracle 2* with a strict stance that there is a limited array of ways to implement a method, and thus the implementations of methods (or, the functionality of a computer program) can never be eligible for copyright. This view is persuasive in that it takes into consideration the

---

<sup>215</sup> Software Directive op cit note 203, Article 1(2).

<sup>216</sup> *SAS* supra note 30 para 29.

<sup>217</sup> *Infopaq International A/S v Danske Dagblades Forening* (2009) ECR I-6569.

<sup>218</sup> *SAS* supra note 30 para 31.

<sup>219</sup> *Ibid.*

<sup>220</sup> *Ibid.*

<sup>221</sup> *Ibid.*

<sup>222</sup> *Ibid* para 33.

reality that there can be no creativity in a creation where alternative options as to its constitution were minimal. I submit that the focus on the author's 'intellectual creation' is limited to the produce of the author's mind which properly fits into the ambit of that which may be protected in copyright. That is to say, the creation of a technological tool would not constitute an 'intellectual creation' insofar as its contents are functional for these purposes. Rather, it would be better fitted within the bounds of patents law.

It is also sensible that the court did not want to grant rights in copyright over a functional, technological element of an API – its interoperability - when the law of patents is the proper domain for such aspects. Michael Morgan sets the position succinctly: 'The ability to interoperate is ... a valuable characteristic of computer products. While strong copyright protection serves the goal of encouraging diversity in literary expression, it does not serve the goal of encouraging standardization and interoperation of computer products.'<sup>223</sup> It is easier to gain exclusive rights over a copyright work than it is for a patentable invention, and the protection under copyright is much longer.<sup>224</sup> This is also the case in South Africa. It makes no sense to grant exclusive rights in copyright over the functional aspect of an APIs.

#### 3.3.2.4 *Implications of Judgement*

The implication of this ruling is that software developers in the UK will struggle to hold any competitor who copies their computer program's functionality liable for infringement unless there has been literal copying of the underlying code of the program as well.

The findings in relation to computer programs in general are likely to apply to APIs in particular. APIs are computer programs which are, per the UK approach, completely utilitarian in how they operate. Their non-literal components will be deemed to merely constitute an idea, which is not worthy of copyright protection, insofar as they have a functional purpose. This echoes the basic sentiment of the court a *quo* in

---

<sup>223</sup> Michael F. Morgan 'Trash Talking: The Protection of Intellectual Property Rights in Computer Software' (1994) 26 *Ottawa Law Review* 425 at 435.

<sup>224</sup> See John Cady 'Copyrighting Computer Programs: Distinguishing Expression from Ideas' (2003) 22 *Temple Environmental Law and Technology Journal* 15 at 19 where he writes: 'Copyright should not be available to extend the equivalent of patent protection, because the safeguards built into the patent system are absent from the copyright system.'

the *Oracle* matter. It seems like a logical conclusion which follows from the fact that copyright is intended to protect creative expression. In making this decision, this court has shown that it is not confused by the dual nature of computer programs – that is, their having a functional technological nature which might fit more comfortably under patent law, as well as their literary aspect which is moulded to fit into the law of copyright. They have considered this purely as a literary work, which is what their law dictates, and found that its functional aspects cannot be protected under copyright as they lack the necessary creative element.

It is also interesting that the court sees fit to pay consideration to the view that permitting the copyrightability of the functionality of computer programs would be detrimental to competition in business. This is a concern that is rife amongst proponents of freely available APIs. This is so because APIs, due to their function of allowing different programs to call on each other and thereby facilitating the creation and functioning of interconnected networks of applications, are simple but core components of all programs. The effects on the greater programming community would be disastrous if a single person could have exclusive rights of copyright over an API. These considerations are relevant in the South African context as well, though the copyright law in South Africa differs slightly.

Like the USA and the UK, South Africa protects computer programs under the law of copyright.<sup>225</sup> The implications of the UK finding, were it to be applied in South Africa and expressly so in relation to APIs, would be highly beneficial to all persons seeking to develop interoperable software. As will be shown, though, the South African context demands that there be fair and just treatment of all parties. This includes the actual authors of the APIs. The UK position does not wholly benefit the authors of computer programs because it does not protect their economic interests in the programs that they have developed, in a direct way. In considering what the best solution which ought to be taken in South Africa is, it is important to take note of these warring interests.

### 3.4 South Africa

---

<sup>225</sup> Where computer programs are incorporated into an invention, the law of patents may be applicable, but discussion of that possibility is beyond the scope of this thesis.

### 3.4.1 *Summary of South African Approach*

The SA Copyright Act provides *sui generis* protection for computer programs. This includes the literal components of a computer program.<sup>226</sup> There have been signs in South African case law of an acknowledgment of the copyrightability of the non-literal components of a computer program<sup>227</sup>; however, there is yet to be a definite decision by any court. In relation to APIs, this question is yet to be posed at all. It is for this reason that this paper seeks to determine what the position in relation ought to be, with consideration to the USA and UK approaches as described above, and in light of the legal and policy considerations relevant in the South African landscape.

### 3.4.2 *The Constitutional Context*

The starting point of any discussion that concerns legislation, its interpretation and any potential development thereof is the Constitution.<sup>228</sup> This is evident in its own text, as well as the dicta set in prior cases.

One of the founding values of South Africa is the supremacy of the Constitution.<sup>229</sup> Of all the different types of laws at play in South Africa – statutory, customary and common laws – the Constitution reigns supreme.<sup>230</sup> This means that it is imperative that the obligations imposed by the Constitution be fulfilled; any law or actions which contradict it are invalid.<sup>231</sup>

The state is obligated to respect, protect, promote and fulfil the rights set out in the Bill of Rights, as it is the ‘cornerstone of democracy’ in South Africa.<sup>232</sup> The Bill of Rights applies to all law and binds all organs and subjects of the state.<sup>233</sup> It is trite that any interpretation and development of the law in any of its forms must promote the spirit, purport and object of the Bill of Rights.<sup>234</sup> This entrenches the status of the Constitution as the supreme law of the land, and the Bill of Rights as a focal point of

---

<sup>226</sup> See SA Copyright Act note 41, s(1)(1).

<sup>227</sup> See Ncube op cit note 9 at 442; *Pastel* supra note 25.

<sup>228</sup> Constitution op cit note 56.

<sup>229</sup> *Ibid*, s1(c).

<sup>230</sup> *Ibid*, s2.

<sup>231</sup> *Ibid*.

<sup>232</sup> *Ibid*, s7(1) – (2).

<sup>233</sup> *Ibid*, s8(1).

<sup>234</sup> *Ibid*, s39(2).

reference when the law is considered. Various cases have developed precedent over time which supports these assertions which cannot be derogated from. In the seminal case *S v Makwanyane*<sup>235</sup> Mokgoro J stated that:

'The Constitution makes it particularly imperative for the courts to develop the entrenched fundamental rights in terms of a cohesive set of values, ideal to an open and democratic society. To this end, common values of human rights protection the world over and foreign precedent may be instructive.'<sup>236</sup>

It was Rabie CJ in *University of Cape Town v Cape Bar Council*<sup>237</sup> who held that, when interpreting legislation, the court is compelled to examine the entire context in ascertaining the intention of the legislature, regardless of whether the text of the legislation is ambiguous or not. It is a truth of the modern constitutional dispensation that the Constitution has 'changed the context of legal thought and decision-making in South Africa'.<sup>238</sup> That is to say, when interpreting any legislation (or developing the law), the first point which an adjudicator is compelled to think carefully about is the background of rights and considerations contained in the Constitution.

### 3.4.3 Analysis of Current Legal Position and Proposal for Improved Approach

The body of case law in South Africa is scant. In *Haupt v Brewer*<sup>239</sup>, the court dealt with the general criteria which render a computer program eligible for copyright, and distinguished the computer program from other works involved in the overall make-up of the computer software (including preparatory materials which were correctly found to be literary works). However, consideration of the effects of the non-literal aspects of a computer program on its eligibility for copyright are lacking. Even though South Africa's *sui generis* categorisation of computer programs may limit the persuasive value of the foreign jurisdictions, they are still instructive as there is no other guidance available locally.<sup>240</sup>

---

<sup>235</sup> *S v Makwanyane and Another* 1995 (3) SA 391 (CC).

<sup>236</sup> *Ibid* para 302.

<sup>237</sup> *University of Cape Town v Cape Bar Council* 1986 (2) ALL SA 619 (A).

<sup>238</sup> As stated by Cameron J in *Holomisa v Argus Newspapers Ltd* 1996 (2) SA 588 (W) at 618.

<sup>239</sup> *Haupt* op cit note 23.

<sup>240</sup> Ncube op cit note 9 at 442.

The running theme throughout this paper is the ideas-expression dichotomy and the need to differentiate between ideas and expressions when dealing with the non-literal elements of a computer program – specifically the API. The decision in *Oracle 1* mirrors the finding in *SAS* to the extent that both judgments held that functionality cannot be copyrightable. While the *SAS* position is of particularly wide ambit as it blandly covers all computer programs, the court in *Oracle 1* was more nuanced in that it specifically addressed APIs in holding that tweaks and improvements to the structure and workings of an API have the effect of improving efficiency. As such, the space for creative expression in APIs is very limited because all developments and modifications in structure are primarily functional and not expressive.

Even the Federal Court in *Oracle 2* is in agreement with the court *a quo* that functional aspects of an API will not be eligible for copyright. The difference, of course, is that the judge in *Oracle 2* is not of the view that the non-literal elements of an API will be purely functional in all instances. The possibility of expression in the API render it copyrightable. At first glance, this latter view may be appropriate in the South African context. The Constitution demands fair and equal treatment of all persons.<sup>241</sup> It also provides that the owners of property may not be arbitrarily deprived thereof.<sup>242</sup> This means that the law may not operate to deprive persons of their property. It has already been established that intellectual property falls into the ambit of this constitutional protection.<sup>243</sup> Furthermore, the justifications discussed in Chapter 2 compel us to give due consideration to the economic interests of the author of the work, notwithstanding the fact that the copyright owner of a computer program is not always the author.<sup>244</sup> Of course, it is also true that the Constitution limits the very rights that it protects, in order to put into effect a proportional and just balance between the interests of affected

---

<sup>241</sup> Constitution op cit note 56, s8(1).

<sup>242</sup> Ibid, s25(1).

<sup>243</sup> AJ Van der Walt & RM Shay 'Constitutional Analysis of Intellectual Property' (2014) 17 *Potchefstroom Electronic Law Journal* 54 at 56 -57.

<sup>244</sup> Given that computer programs are usually, but not always, created by developers under a contract of employment.

parties.<sup>245</sup> This means that copyright owners in APIs are not due an unfettered right to their works.<sup>246</sup>

The protection of copyright is not justified where its effect is to stunt innovation and creativity in the community. It makes no sense, therefore, to provide exclusive rights in a function, method or structure in a manner that produces this undesirable consequence. This background must be kept in mind in the consideration of whether the non-literal components of an API should be copyrightable. What is significant about APIs in the context of copyrightability is that they are computer programs which are primarily designed to achieve the simple function of interoperability. It is the case that in the USA interoperability is a factor in the question of fair use; but South Africa is not hamstrung by the precedent of the USA. As such, I submit that the fact that these programs are singularly designed to perform this sole function is an important consideration to note.

This consideration leads, as it did in *SAS* and *Oracle 1*, to a firm conclusion that there is little space for creative expression in the design of an API. All alterations, additions and variations to the structure and overall organisation of an API are ultimately utilitarian. Indeed, it may take a creative and insightful developer to find the best structural arrangement to achieve the desired result, but nevertheless, her decisions will be based on the need to improve the program's efficiency and effectiveness. It has been repeated many times in this paper that copyright law does not protect that which is functional. The realms of patent, trademark<sup>247</sup> and, to an extent, designs law<sup>248</sup> protect functional creations. From a legal standpoint, I submit that there is no basis for copyright to attach to functional elements of an API.

The economic and social impacts are also important. The effect on the public if functionality could, in circumstances, be eligible for copyright, is discussed in the UK.<sup>249</sup> It is found that the possibility of exclusive ownership in non-literal components would be detrimental to innovation.<sup>250</sup> This is a powerful observation. If a person is

---

<sup>245</sup> Constitution op cit note 56, s36.

<sup>246</sup> This position is the same in the USA. John Cady op cit note 224 at 18, states: '... it is necessary to balance a complex system of needs: the author's need for incentives to creativity and the public's needs both for access to the fruits of the author's endeavour and for protection against monopoly.'

<sup>247</sup> Trademarks Act 194 of 1993.

<sup>248</sup> Designs Act 195 of 1993.

<sup>249</sup> *SAS* supra note 30.

<sup>250</sup> *Ibid* para 31.

seeking protection in the market at the level offered by patent rights, the person should take the avenues provided by patent protection. These are much more difficult to satisfy than the paths of eligibility for copyright. This is fair, considering the fact that the exclusive power sought creates an extremely high bar to market entry for competitors and casual persons seeking to innovate. APIs allow for the efficient functioning of networks of computer programs. All of the benefits that computer software brings to society are made possible by APIs. They are critical pieces of software, without which modern advancements would slow to a crippling rate. The negative effect is similar if only a core group of developers have exclusive rights over APIs. On this basis, I submit their use to society as a whole is of greater importance than the interests that a few copyright owners may have in filling their pockets.

It is not fair to grant the minority excessive and exclusive rights which are disproportionately easy to obtain in comparison to the level of control and potential market share that they grant the rights owner, at the expense of the community at large. Rather, functional tools such as APIs should not be capable of copyright eligibility, on grounds of policy and practicality; and with due regard to the legal analysis.

## CHAPTER 4: API'S AND EXCEPTIONS TO INFRINGEMENT

### 4.1 Introduction

There are exceptions to copyright infringement. This means that, in certain cases defined or designated in each jurisdiction, a person who would otherwise be liable for copyright infringement is exempted from such liability on the basis of a lawful exception. The USA differs from South Africa and the UK which employ fair dealing. In the USA, courts determine whether otherwise infringing conduct may be exempted from liability by considering whether the conduct constitutes a fair use of the copyrighted work or not.<sup>251</sup>

---

<sup>251</sup> See O'Donnell op cit note 60 at 621.

Fair use acts as a defence where the infringing act is deemed to satisfy certain criteria which render it a lawful use of the copyright work.<sup>252</sup> The Federal Court in *Oracle 2*<sup>253</sup> had an opportunity to consider some points relating to fair use and API's. Though it did not come to a decision, opting rather to send the question of fair use back to the court a *quo* pending further evidence from both parties, its explanation and interpretation of the criteria in relation to APIs offers some insight. In South Africa and the UK, on the other hand, there is a closed list of exceptions to infringement. This is known as fair dealing. The fair dealing doctrine has been described as:

'a key part of the social bargain at the heart of copyright law, in which as a society we concede certain limited individual property rights to ensure the benefits of creativity to a living culture ... [and] is more important today than ever before.'<sup>254</sup>

This Chapter will analyse the USA and UK position on exceptions in relation to computer programs, with a specific focus on the implications on APIs, and present the South African a proposal on what the South African position ought to be with regards to APIs.

#### 4.2 Alternative Mechanism: Reverse Engineering

Reverse engineering includes the attempt to reverse the steps in making a program in order to uncover the source code version of a program from its object code.<sup>255</sup> This is so because users who need to study the structure and technical parameters of an existing program can only do so once they have accessed the program in its source code form.<sup>256</sup> A person may make use of various methods of reverse engineering which include analysing screen displays of object code, or decompilation or disassembly of the program.<sup>257</sup>

---

<sup>252</sup> Jacqueline Litpon 'The Law of Unintended Consequences: The Digital Millenium Copyright Act and Interoperability' (2005) 62 *Washington and Lee Law Review* 487 at 532.

<sup>253</sup> *Oracle Appeal* supra note 7.

<sup>254</sup> Association of Independent Video and Filmmakers (ANF) *et al.*, 'Documentary Filmmakers' Statement of Best Practices in Fair Use' available at <http://centerforsocialmedia.org/files/pdf/fair-use-final.pdf> accessed on 12 February 2018.

<sup>255</sup> McManis op cit note 13 at 29.

<sup>256</sup> *Ibid.*

<sup>257</sup> *Ibid* at 29.

Section 107 of the US Copyright Act permits the fair use of person's copyrighted work.<sup>258</sup> This permits all forms of reverse engineering, so long as they are deemed to be fair, to be a means of 'obtaining access to the unprotected ideas'<sup>259</sup> found in a publicly distributed computer program.<sup>260</sup> One may also engage in reverse analysis of a computer program where it is an essential step in the utilisation of that program.<sup>261</sup> Reverse engineering is provided for in the European Union. The EC Directive provides that:

'[A] person having a right to use a copy of a computer program shall be entitled, without the authorization of the [copyrightholder, to observe, study or test the functioning of the program in order to determine the ideas and principles which underlie any element of the program if he does so while performing any of the acts of loading, displaying, running, transmitting or storing the program which he is entitled to do.'<sup>262</sup>

Furthermore, the EC Directive permits decompilation or disassembly where they are necessary to 'achieve the interoperability of an independently created computer program with other programs', subject to conditions.<sup>263</sup>

As mentioned above, reverse engineering is permitted in both the USA and the UK. The advantage of reverse engineering is that it allows a person to figure out the workings of a program by creating her own source code after disassembling the object code of the respective program. It is limited, however, in that it puts a person in the position where she is still forced to create her own API, which is inefficient and time consuming. It does little to subvert the disadvantage one faces in having to create one's own API 'from the ground up', as it were. The user remains restricted from using the API.<sup>264</sup> Nevertheless, to permit reverse engineering is a progressive step and one that provides developers with an opportunity to innovate. It would certainly find useful application in the South African context.

---

<sup>258</sup> Ibid at 45.

<sup>259</sup> Ibid.

<sup>260</sup> Ibid, as found in *Games Corp. v. Nintendo of America* supra note 165, and *Sega Enterprises, Ltd. v. Accolade, Inc.* supra note 138.

<sup>261</sup> US Copyright Act op cit note 43, s117, as interpreted per McManis op cit note 13 at 45.

<sup>262</sup> EC, *Directive 2001/29/EC of the European Parliament and of the Council of 22 May 2001 on the harmonization of certain aspects of copyright and related rights in the information society*, [2001] O.J.L. 167, Article 5(3).

<sup>263</sup> Ibid, Article 6.

<sup>264</sup> John T. Soma, Gus Weinfeld & Letty Freinsein 'Software Interoperability and Reverse Engineering' (1994) 20 *Rutgers Technology and Law Journal* 189 at 239.

What follows is an analysis of the general fair use and fair dealing doctrines as used in the USA and the UK respectively.

### 4.3 United States of America

#### 4.3.1 *Summary of USA Approach*

It is possible for a defendant to argue that her use of an API over which she lacks the requisite ownership rights or permission from the copyright owner is excepted from constituting a copyright infringement. Said defendant must prove that the use qualifies as a fair use.

As the law stands after the finding in *Oracle 2*, courts will consider the interoperability function of APIs, amongst other factors, as being favourable towards a determination that the use is fair.<sup>265</sup> This is so because interoperability is a core component of programming, necessary to allow different programs to work in sync without individual programmers having to develop entire lines of code to perform tasks which existing programs already perform. Fair use has its disadvantages, however, as it is unpredictable: if a copyright owner can prove that the defendant's use reduces the market share of the copyright owner, a court may find that the use is not fair.<sup>266</sup>

Ultimately, the decision of each court will differ on a case-by-case basis. The same facts in one matter may foreseeably yield different results in another matter. Such uncertainty offers little relief for parties that may seek to use APIs but lack the financial security to entertain the threat of litigation. As such, the practical effects of the fair use treatment of APIs may result in reduced innovation as small players seek to avoid threats against their business by removing existing APIs that they may be using in the development of their programs, and resisting future use of the same altogether.

The fair use doctrine is explained in greater detail below, as well as the manner in which it may affect APIs.

---

<sup>265</sup> *Oracle Appeal* supra note 7 at 45.

<sup>266</sup> See comments on unpredictable nature of Fair Use in section 4.3.5 of this paper.

### 4.3.2 *Fair Use Doctrine*

The USA is hailed as the most progressive jurisdiction for being flexible in its approach to exceptions to infringement, as it makes use of the fair use doctrine which is arguably the ideal model for copyright law.<sup>267</sup> Fair use is simply an open list of uses which may be permissible if found to be fair.<sup>268</sup> The factors considered to determine fairness will appear from the case discussion below of the Federal Court's judgment in *Oracle 2*.

### 4.3.3 *Oracle v Google 2*

#### 4.3.3.1 *Summary of Findings*

Given that the Federal Court found that the JAVA API was eligible for copyright, it turned its focus to the question of whether Google's use could be excepted from infringement on the basis of fair use. Ultimately, the court stayed from making a decision as it found that insufficient evidence had been tendered.<sup>269</sup> Nevertheless, its discussion of the law is of aid in the considerations to come relating to the South African context.

#### 4.3.3.2 *Case Discussion of Fair Use*

The court began its discussion on fair use with reference to the US Copyright Act. It provides that works that are protected under copyright may be used for criticism, comment, news reporting, teaching, scholarship or research.<sup>270</sup> As has been made clear, this does not constitute a *numerous clausus* of categories of uses which may be fair.<sup>271</sup>

---

<sup>267</sup> See D'Agostino note 59 at 344.

<sup>268</sup> *Ibid.*

<sup>269</sup> A jury for the district court did find that there was fair use during a new trial initiated after Google's application for an appeal against the judgment of the Federal Court had been rejected by the Supreme Court. The views on fair use as expressed by the judge in *Oracle 2* are considered in this paper as they shed more light on the treatment of fair use than the vote of the jury in the subsequent district court trial.

<sup>270</sup> US Copyright Act op cit note 43, s107.

<sup>271</sup> For instance, fair use for reverse engineering is favoured in the USA – see McManis op cit note 13 at 27.

In order to determine if one of these uses is fair or not, four factors must be considered on a case-by-case basis.<sup>272</sup> These factors are instructive as to the considerations that South African courts may make when determining if a permissible use is fair or not. The court listed the factors as they appear in the US Copyright Act:

'(1) "the purpose and character of the use, including whether such use is of a commercial nature or is for non-profit educational purposes;" (2) "the nature of the copyrighted work;" (3) "the amount and substantiality of the portion used in relation to the copyrighted work as a whole;" and (4) "the effect of the use upon the potential market for or value of the copyrighted work.'<sup>273</sup>

The court noted that all of the factors must be considered and weighed against each other, and reflected upon in light of the purpose of copyright in the USA which is to promote the '[P]rogress of Science and useful Arts'.<sup>274</sup>

It noted the general principles of the fair use doctrine in relation to the factors. The first factor revolves around the purpose and character of the use. This includes whether the use is of a commercial nature or whether it is for non-profit educational purposes. This factor is split into two considerations: whether, and the extent that, the work is transformative and whether its use is commercial or not. The court supported the view that a work will only be transformative where the copyrighted work is changed, or where the copyrighted work is used in a new context such that the copyrighted work is effectively a new creation.<sup>275</sup> A work will not be transformative if it makes no change to the *expressive content* [emphasised by the court] of the original work.<sup>276</sup> As regards the question of commercial use, such use weakens an argument for fair use.<sup>277</sup>

The court found that the second factor recognises that certain works are 'closer to the core of intended copyright protection'.<sup>278</sup> This means that there are certain works for which copyright protection is robust and limitations thereof are scarcely applicable. The court stated that the question of whether fair use has been established or not will

---

<sup>272</sup> *Harper & Row Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 549 (1985).

<sup>273</sup> *Oracle Appeal* supra note 7 at 54, quoting US Copyright Act op cit note 43, s107.

<sup>274</sup> United States of America: Constitution, 17 September 1787, Article 1(8)(8); *Ibid*, as found in *Campbell v. Acuff-Rose Music, Inc.*, 510 U.S. 569 (1994) at 575, 578.

<sup>275</sup> *Perfect 10, Inc. v. Amazon.com, Inc.*, 508 F.3d 1146 (9<sup>th</sup> Cir. 2007) at 1164 where the court states: 'A work is "transformative" when the new work does not "merely supersede the objects of the original creation" but rather "adds something new, with a further purpose or different character, altering the first with new expression, meaning or message.'

<sup>276</sup> *Oracle Appeal* supra note 7 at 57.

<sup>277</sup> *Ibid*.

<sup>278</sup> *Ibid* at 57 – 58.

turn on whether the work is of an informational or creative nature. It is creative expression which falls centrally into the protective aims of copyright law.<sup>279</sup> This is appropriate to computer programs in general, and APIs specifically, because fair use is likely to be applicable where purely functional elements exist in the work and the expressive elements must be copied in order to perform the function.<sup>280</sup>

The third factor calls for a consideration of the amount of the original work which was copied; considered from the perspective of the original work, and not of how much of the infringing copy consists of the original work. Complete copying does not, however, automatically defeat fair use. If the person who copies only copies inasmuch as is necessary to achieve the intended function, then the copying will not render the use unfair.<sup>281</sup>

Finally, fair use may not materially impair the potential market for the original copy. The importance of this factor is great, but varies with consideration of the other factors weighing on its significance.<sup>282</sup> As stated by the Supreme Court in *Campbell*<sup>283</sup>, courts must not only consider the extent of the market harm caused by the infringement, but also whether or not unrestricted conduct like that engaged in by the defendant would substantially prejudice the potential market of the original copyright owner.<sup>284</sup>

The court found that interoperability is a relevant consideration with regards to the second and third factors. It stated that, in relation to the core packages of the JAVA API, the functional necessity implicated by interoperability is in favour of fair use.<sup>285</sup>

#### 4.3.4 Implications of Fair Use Treatment

Though the Federal Court withheld from making a final decision as to fair use in this case, its considerations are telling. Its discussion confirms that it is possible for a defendant to use fair use as a defence for copyright infringement of an API. It also

---

<sup>279</sup> Ibid.

<sup>280</sup> Ibid.

<sup>281</sup> *Kelly v. Arriba Soft Corp.*, 336 F.3d 811, 820-21 (9th Cir. 2003).

<sup>282</sup> *Oracle Appeal* supra note 7 at 59.

<sup>283</sup> See *Campbell* supra note 274.

<sup>284</sup> Ibid at 59.

<sup>285</sup> Ibid at 61.

confirms the importance of the interoperability function of APIs in support of an argument that the use of APIs should qualify as fair use. No single factor or sub-factor is likely to be a deciding factor on its own; however, the function that APIs serve is so crucial and cannot be divorced from their expressive aspects. It is likely, then, that exploiting APIs will, in most circumstances, qualify as fair use. It is also likely that the detrimental effects on the market of the owner of the APIs in any given circumstance may lead the court to conclude that use of the APIs cannot be fair. This will differ per the varying facts of each case. Nevertheless, the *Oracle* case gives rise to the conclusion that it is likely that persons who make use of the APIs of others may not attract legal liability.

The manner in which courts in the USA have dealt with fair use seems steeped in a capitalism-centric incentivisation justification for copyright protection, though its doctrine also places emphasis on creativity.<sup>286</sup> So long as a user of a copyrighted work does not prejudice the market share of the owner of the copyright, and the use is minimal (in relation to its importance to the original work), then the use may be fair. Even so, a court may always rule in the opposite way.

The implications of this ruling also attract policy considerations which demand attention in any discussion of the approach which ought to be taken in South Africa. The possibility that developers may deal with APIs using the doctrine of fair use does not offer the kind of legal certainty that allows business – especially small enterprises with scant financial clout – to act without concern.<sup>287</sup> Applications and programs which are already in existence and make use of the APIs of third parties may have to be redesigned in order to ensure that their developers are protected from legal liability.<sup>288</sup> This is so because not everyone can afford to engage in judicial matters in order to argue that their use of any respective API is fair. The irony which attaches to the uncertainty of fair use is that innovation will inevitably be hindered.

---

<sup>286</sup> See USA Constitution op cit note 166.

<sup>287</sup> See Lloyd L. Weinreb 'Fair Use' (1999) 67 *Fordham Law Review* 1291 at 1291 where he quotes from *Folsom v Marsh* 9 F. Cas. 342 (C.C.D. Mass. 1841) (No. 4901). Which states that fair use is: one of those intricate and embarrassing questions... in which it is not... easy to arrive at any satisfactory conclusion, or to lay down any general principles applicable to all cases.; O'Donnell note 60 at 58.

<sup>288</sup> See section 4.3.1 of this paper where the manner in which the fear of litigation may dispel companies from using APIs is noted.

On the opposite end of the spectrum, a person with copyright ownership of an API may welcome the fact that they have rights in a work over which they have expended their efforts and intellect to create. Due to the fact that fairness is determined on a case-by-case basis, it is arguable that a court will necessarily make considerations which have a balanced effect on both the plaintiff and the defendant in such matters. This does not resolve the issue that the uncertainty may have a deterring effect on third-parties that would otherwise seek to use APIs to spur their own innovation. Nevertheless, it seems to be the best option in the event where APIs are found to be eligible for copyright.

#### 4.4 United Kingdom

##### *4.4.1 Summary of UK Approach*

The fair dealing doctrine is applied in the United Kingdom. It is a closed list of uses which are eligible for determination as to whether they are fair or not.

There has been no decision regarding APIs and their applicability to this exception, as is made clear below, but the doctrine as applied in the UK Copyright Act does not support a case for use of APIs being fair. This is contrary to the USA approach, and it is much less flexible and capable of keeping pace with the rapidly changing times. The fair dealing doctrine as applied in the UK is canvassed in greater detail below.

##### *4.4.2 Fair Dealing Doctrine*

There is no case which considers fair dealing in relation to APIs, which is sensible as the only matters to deal with the copyrightability of computer programs have already established that the functional aspects of a computer program are not copyrightable.<sup>289</sup> Nevertheless, computer programs are treated as a specie of literary work in the UK, and the UK Copyright Act provides for fair dealing in respect of such works, among others.

---

<sup>289</sup> This includes SAS supra note 30.

Due to the latter, and because the UK system pertaining to the fair dealing doctrine is similar to that which is employed in South Africa, the treatment of fair dealing in the UK is illustrative and of guidance.

In order for a use to be considered fair dealing, it must fall into one of the categories set in the UK Copyright Act for fair dealing, as well as be fair and, in some cases, include acknowledgement of the author of the original work.<sup>290</sup> The UK Copyright Act provides that use for the purposes of research or private study does not infringe copyright<sup>291</sup>; neither does it infringe for the purposes of criticism and review, nor when reporting current events.<sup>292</sup> Exceptions also include for education<sup>293</sup> as well as anything done pursuant to parliamentary or judicial proceedings.<sup>294</sup> The South African law of copyright operates with a similar limited set of permissible uses, barring the exception on reverse engineering, so the manner in which the UK courts approach this challenge is relevant as it sheds light on the possibility of the availability of opportunities to use computer programs lawfully.

#### 4.4.2.1 *Permissible Uses in Fair Dealing*

Due to the fact that the list of categories of acceptable dealing is closed, the judiciary has opted to take a liberal approach in determining which uses are apt for consideration under the fair dealing doctrine.<sup>295</sup> This means that the courts do not, where possible, follow a strict interpretation of the provisions relating to fair dealing. They interpret the provisions in such a way so as to allow as many different kinds of uses of copyrighted works to be included in the fair dealing exceptions, so long as the respective courts consider these uses to be acceptable. Nevertheless, the scope of their interpretive powers is severely limited.

The closed list of categories of acceptable uses presents a monumental problem for the person seeking to make use of a computer program such as an API in order to enable, for instance, interoperability between different programs. Such a

---

<sup>290</sup> These including criticism and review; and for the reporting of current events.

<sup>291</sup> UK Copyright Act op cit note 53, s29(1).

<sup>292</sup> Ibid, s30(1) – (2).

<sup>293</sup> Ibid, s32.

<sup>294</sup> Ibid, s45(1).

<sup>295</sup> See Healing note 59 at 333.

person will need to frame their use of the API in a manner which can be categorised as one of the acceptable uses. This is a problem commonly raised amongst those who criticise the fair dealing doctrine; it is a problem not encountered in the US system because the fair use doctrine allows an infinite number of uses to qualify under the protection of its exceptions, so long as they are found to be fair.<sup>296</sup> As South Africa also employs the fair dealing system, similar problems are foreseeable in relation to computer programs. This is the case even more so because computer programs are not treated as literary works in South Africa, and consequently are able to only benefit from a shorter list of exceptions than literary works.

If the respective work does happen to fall under one of the enumerated categories, the court will determine its fairness using considerations as introduced in *Hubbard*.<sup>297</sup> In this matter, where the court determined whether the use of a book which relied extensively on the book of another author was fair or not, the court stated:

'It is impossible to define what is 'fair dealing'. It must be a question of degree. You must consider first the number and extent of the quotations and extracts. Are they altogether too many and too long to be fair? Then you must consider the use made of them. ... [i]f they are used to convey the same information as the author for a rival purpose, that may be unfair. Next, you must consider the proportions ... after all is said and done, it must be a matter of impression.'<sup>298</sup>

#### 4.4.2.2 *Factors of Fairness*

Judicial application of these principles has yielded factors which bare a marked resemblance to the considerations made under the doctrine of fair use in the USA. These include not only the nature of the work, but also the manner in which the work was obtained; the amount taken; the degree of transformation of the use; whether the new work has been used for a commercial benefit or not; the motives behind and consequences of the dealing; and whether alternatives existed.<sup>299</sup>

---

<sup>296</sup> Ibid at 345.

<sup>297</sup> *Hubbard v Vosper* (1971) 1 All ER 1023 CA.

<sup>298</sup> Ibid at 1027.

<sup>299</sup> See Healing note 59 at 342 – 343.

In *Ashdown v Telegraph*<sup>300</sup>, where a daily newspaper published confidential political material which related to the pending formation of a new government in the UK and claimed that it was not liable for copyright infringement on the basis of fair dealing, Aldous J provided that the standard to be applied in weighing the factors of fairness is that of a fair minded and honest person.

The range of options that were available to the author in creating the work is also used to determine whether the work can be lawfully used or not. This appears in both the UK and the USA approaches. If the developer of an API creates a structure and organisation where she had multiple alternative options that she could have exploited to yield her intended result, it is highly unlikely use of her API will be fair.

The UK approach, not unlike the USA approach, also places great weight on the mindset or intentions of the user. This is evident in that the court must interpret the situation through the lens of a 'fair minded and honest' person. Nevertheless, the fair dealing doctrine does not create a favourable legal avenue for users of APIs.

#### 4.4.3 Implications of Fair Dealing Doctrine

The implications of the fair dealing doctrine as applied in the UK on those who would seek to benefit from advances in technology do not appear to be entirely beneficial. Indeed, one of the biggest criticisms of fair dealing is its apparent inability to evolve and adopt solutions to copyright problems caused as a result of technological progress.<sup>301</sup>

For example, in Australia it was not considered fair dealing to use a VCR in order to record television at one's home because the Australian fair dealing categories did not accommodate this technology. It was only in 2006 that the legislature enacted

---

<sup>300</sup> *Ashdown v. Telegraph Group Ltd.*, (2001) EWCA Civ 1142; O'Donnell op cit note 60 at 63.

<sup>301</sup> Nicolas Suzor 'Fair Use v Fair Dealing: How Australian Copyright Law Differs' available at <https://www.lifehacker.com.au/2017/07/fair-use-vs-fair-dealing-how-australian-copyright-law-differs/> accessed on 12 February 2018.

the Copyright Amendment Act<sup>302</sup> which cured this issue. By this time, of course, VCRs were well on their way to being relics of the past.<sup>303</sup>

The inability to accommodate technological progress is made further evident due to the fact that the available categories for fair dealing in the UK are of little aid to users of computer programs. The categories cater to those dealing in traditional literary works, such as books and works of art. At its most progressive, the list includes parodies, which could implicate cinematographic works or other modern creations.

It seems clear, then, that the intention of the legislature in the UK was to ensure that the balancing of the interests of the copyright owner of a computer program and third-party users was to be put into effect only at the stage of the copyright eligibility enquiry. The only avenue through which users of computer programs could hope to gain entry would be, as discussed above, through the exception which relates to private research or study, or reverse engineering, which is of limited scope as it is a right to decompile for the purpose of understanding, rather than a right to use the computer program. Overall, it seems that fair dealing in the UK does not accommodate those who seek to make use of computer programs without being required to pay the owner of the copyright in the work.

This assessment seems not to bode well for users in the South African context. The SA Copyright Act does, however, contain key differences to the UK Copyright Act. These stem from the fact that computer programs in South Africa constitute their own separate work. This means that the considerations which underlie the UK position will be different in South Africa. Furthermore, the legal contexts in the two jurisdictions make it reasonably possible to foresee that the results reached in South Africa may differ markedly.

What follows is a consideration of the potential South African approach. A proposal is then put forward as to what the approach in South Africa should be in relation to APIs.

#### 4.5 South Africa

---

<sup>302</sup> Copyright Amendment Act 158 of 2006.

<sup>303</sup> Suzor op cit note 301.

#### 4.5.1 *Summary of South African Approach*

It is within the spectrum of possibilities that the judiciary in South Africa will find that APIs may be eligible for copyright, like the Federal Court in *Oracle 2*. This gives rise to the second issue of whether a user may have a defence against an infringement claim by making use of the doctrine of fair dealing.

I have argued that the SA Copyright Act makes use of a fair dealing exception which is not inclusive of computer programs in any way that is practically beneficial to an API user. There is a slim opportunity created by an exception for quotations; however, no ruling yet exists in relation to such an exception and computer programs, and it is unlikely that any such ruling would offer erstwhile freedom to API users. The Copyright Bill, however, seems progressive and almost constitutionally satisfactory. Its introduction in South African law, subject to certain linguistic changes to its content, will be fair to both API users as well as copyright owners.

#### 4.5.2 *Constitutional Context*

The supremacy of the Constitution has already been underlined. All considerations of what the law is, and what it should be, must be coloured by the contents and intents of the Constitution. This is all the more relevant considering that the SA Copyright Bill remains, at the time of writing, under discussion. The UK approach has revealed some of the difficulties of fair dealing. This Copyright Bill creates an opportunity for some of those issues to be resolved, and should include improvements that fulfil the spirit and purport of the Bill of Rights (specifically, for the purposes of this paper, in relation to computer programs). These improvements can be designed to be of benefit to all parties involved with APIs – from the copyright owner to the prospective users.

What follows is a discussion of the Copyright Bill and an analysis of the foreign law treatment of exceptions to copyright infringement. Finally, I will assess the shortfalls and opportunities of the current fair dealing regime in South Africa insofar as it relates to computer programs, specifically with regards to APIs, and I will issue proposals for a fair and balanced solution.

#### 4.5.3 *Discussion of South African Copyright Bill*

The Copyright Bill introduces a 'fair use' clause which is in actual fact a fair dealing clause as the list of permissible uses remains closed.<sup>304</sup> To this end, it has been suggested that the clause should introduce its list of permissible uses with the phrase 'such as', so as to leave the list of uses open and thus make the provision one of fair use.<sup>305</sup> This would go some way to resolve the issues of restrictive application discussed above in relation to the fair dealing clauses. Nevertheless, the Copyright Bill does introduce some protection that the SA Copyright Act does not afford users of computer programs.

It also bears to note that it completely replaces the provisions in section 19B of the SA Copyright Act<sup>306</sup>, which means that computer programs may not be used for criticism, review, judicial proceedings, illustration, demonstration, broadcasting, or quotations.<sup>307</sup> Any of the new fair dealing additions of section 12 of the SA Copyright do not apply to computer programs in the Copyright Bill. Section 19B of the Copyright Bill provides:

'(1) A person having a right to use a copy of a computer program may, without the authorisation of the rights holder, observe, study or test the functioning of the program in order to determine the ideas and principles which underlie any element of the program if that person does so while performing any of the acts of loading, displaying, running, transmitting or storing the program which he or she is entitled to perform.'

This is the introduction of an exception for reverse-engineering as well as the decompilation of software<sup>308</sup> akin to that which is found in the USA and the UK. The Copyright Bill also provides that interoperability may render a use fair, subject to certain conditions:

'(2) The authorisation of the rights holder shall not be required where reproduction of the code and translation of its form are indispensable in order to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs, if ...:

(a) The acts referred to in subsection (1) are performed by the licensee or another person having a right to use a copy of the program, or on their behalf by a person authorised to do so;

---

<sup>304</sup> UCT Intellectual Property Unit *Copyright Amendment Bill 2017: Comments* (2017) at 13.

<sup>305</sup> *Ibid.*

<sup>306</sup> Copyright Bill op cit note 67, s17.

<sup>307</sup> The Anton Mostert Chair of Intellectual Property *Commentary on the Copyright Amendment Bill 2017* (2017) at 32.

<sup>308</sup> *Ibid.*

(b) the information necessary to achieve interoperability has not previously been readily available to the persons referred to in paragraph (a); and

(c) those acts are confined to the parts of the original program which are necessary in order to achieve interoperability.

The Copyright Bill appears to be a saving grace for users of computer programs. It permits specific usage in a manner that is much more inclusive and flexible than the SA Copyright Act. Its provisions mean that, in a case involving APIs in similar circumstances as in the *Oracle* matter, the defendant is likely to find relief on the basis that APIs enable interoperability (an argument which was considered favourably in *Oracle 2*). It does not defeat the rights of the copyright owner, either, as the set limitations which apply to it protect the copyright owner from undue competition.

'(3) The information obtained through the application of the provisions of subsection (2) may not be—

(a) used for goals other than those to achieve the interoperability of the independently created computer program;

(b) given to others except when necessary for the interoperability of the independently created computer program;

(c) used for the development, production or marketing of a computer program substantially similar in its expression to the program contemplated in subsection (1); or

(d) used for any other act which infringes copyright.'

Criticism of this provision has been levelled, though. The word 'interoperability' is defined in the Bill as the 'ability to exchange information and to use the information which has been exchanged'.<sup>309</sup> It has been argued that this definition is counterproductive because the inter-operation of computer programs does not necessarily occur in a reciprocal manner.<sup>310</sup> In other words, at times, computer programs do not exchange information when operating in tandem: interoperability can occur where the flow of information goes in one direction, from one program to the next. It has been suggested that the definition of 'interoperability' as it applies in this

---

<sup>309</sup> Copyright Bill op cit note 67, s19B(4).

<sup>310</sup> CIP op cit note 307 at 32.

provision should be ‘the ability [of a program] to perform its function in concert with another program or programs.’<sup>311</sup>

As it stands, however, the implication of the definition of ‘interoperability’ creates the risk that certain uses of APIs may not qualify as an exception based on interoperability because a program in an inter-operational pair makes use of the API to call on information from the other program, but does not relay information to the other program in reciprocal exchange. This would arbitrarily undermine the positive intention behind section 19B.

Secondly, criticism has been levelled at the decompilation exception which applies only to those parts of the computer program which are necessary to achieve interoperability.<sup>312</sup> This criticism arises because, in reality, decompilation of the entire program must occur before the specific parts which are needed may be identified.<sup>313</sup> This means that the provision as it stands may contradict itself in that it permits decompilation but the manner in which the exception it creates has been framed makes its application practically impossible.

The Copyright Bill removes uses that are tied to literary works and artistic works in favour of creating a specific exception applicable to computer programs. This exception is progressive in that it places great weight in the functionality of a computer program in determining whether use of it would be fair or not. This is particularly useful in the case of APIs because of the fact that the functionality of an API is deeply integral to its existence, as well as to the efficient functioning of software in general. It creates opportunities for third-party users to use APIs in the same manner that Google did in the *Oracle* case. It means that the user of an API is likely to find relief through this exception.

Barring its definitional issues, and the fact that it renders the rest of the section 12 exceptions inapplicable in relation to computer programs, it seems that section 19B of the Copyright Bill has real potential to effectively create a balance between copyright exceptions and limitations thereof.

---

<sup>311</sup> Ibid.

<sup>312</sup> Ibid at 32; Copyright Bill op cite note 67, s2(c).

<sup>313</sup> Ibid.

#### *4.5.4 Proposal for Improved South African Approach*

Much has been said in this paper of the approaches in the USA and the UK. They have been used to show the benefits and shortcomings of the fair use and fair dealing doctrines respectively.

What is to be taken from the analysis of the positions taken in foreign law? The answer is that fair dealing as applied in the UK is an inadequate solution to problems in the ever-developing world of computer technology. It operates to protect the interests of the copyright owner to the detriment of everyone else. It harms innovation in that it provides developers with little options to work with, other than submitting to the demands of the copyright owner. The legal certainty created by the predictability of fair dealing is grossly outweighed by its imbalanced, restrictive application. The position in the SA Copyright Act is no different. It offers little opportunity for APIs to qualify as a permissible use. Until a judicial determination is made regarding the quotations exception and APIs, it merely constitutes a flimsy and narrow bridge to cross over the abyss that is infringement. It is unlikely that it is of any use to API users seeking to be excepted from liability.

The USA presents a much better model to be followed in the event that programs such as APIs are found to be copyrightable. It is malleable – able to keep up with the times and protect users in certain instances. The only weakness of the fair use doctrine is that legal certainty is diluted given that judges will determine its applicability with regard to the facts of each respective matter. In comparison to the issues inherent in the fair dealing doctrine, however, the problem of uncertainty is not so large. It is a natural part of the legal order that courts are the final arbiters of justice, and so in some circumstances it is inescapable that certain questions must be left to the judge. Also, this allows for the protection of the interests of API copyright owners as well, thereby creating a fair and balanced system of law.

Reverse engineering, which is permitted in both jurisdictions, affords the user an opportunity to develop their own API in given circumstances as described above. Although this does not solve the issue that a user is precluded from making use of an existing API as it is, which is a better alternative, the permitting of reverse engineering is progressive. It also affords a fairer balance between the rights of the copyright owner

and the interests of the user. The copyright owner still has protection over his API, while the user may create an API to solve issues of, for example, interoperability.

Fairness, balance and proportionality of treatment are key values in the overarching constitutional context. In this regard, the exceptions in the SA Copyright Act are inadequate as they offer no equal treatment. The Copyright Bill, on the other hand, combines elements of both the USA and the UK to achieve a considerably treatment of the parties involved. This includes the ability of a user to reverse engineer computer programs. This is a limited solution on its own, but it is still much better than the existing South African provisions. Furthermore, the Copyright Bill permits fair dealing strictly where interoperability is the aim of the use. This obviates the issues that users face when faced with existing fair dealing provisions, and also protects copyright owners because its ambit is narrow so as to prevent abuse of their rights. The minor issues related to the definitions used in its exceptions provision need resolution as suggested; otherwise, the provision directly gives users of APIs a genuine opportunity to innovate without backlash from the law.

On this basis, I submit that it is advisable that the Copyright Bill exception in relation to computer programs apply in South Africa. It grants the courts the opportunity to apply the values of the Constitution in each matter as they use it to aid their considerations of the issues at hand. The question of fair use relating to APIs in particular will not prejudice copyright owners or users. The Copyright Bill secures the interests of both parties and is commendable and recommended, subject to revisions of some of its wording as relayed above.

## CONCLUSION

The technological frontier is developing at a rapid pace. The space occupied by software development is expanding at exponentially increasing rates, and APIs are fuelling the acceleration. Without APIs, coding would be a tedious, slow activity which would require strenuous additional programming for each new service added to a piece of software. Now, a programmer can call on a whole host of pre-programmed services using APIs, which enables the rapid growth in software development.

This speedy progression of technology has brought the age-old friction between the law and innovation to the fore: is the law sufficiently developed to cater to new issues that have been and are to be created by development and, if not, how should the law be formulated or reformulated in order to have the capability of resolving these problems? Other, familiar, issues have also been implicated in the debate about copyright ownership and APIs, including the threat of monopolies, the threat to innovation, the effects on the economy of intellectual property protection, and the ever-present need to balance the interests of those persons who make the work, with the interests of the community. The dichotomy between ideas and expressions in copyright has played centre stage on a set comprised of the creative and functional components of the non-literal elements of computer programs.

I have argued that our law will be constitutionally sound should it be developed to preclude APIs from eligibility for copyright. Failing this, I have argued that the SA Copyright Act needs to be developed so that a flexible fair use provision is incorporated into its mechanisms so that a fair balance between the rights of the copyright owner and those persons who seek to use APIs in their software development may be achieved. This development is aligned with the proposed computer program infringement exceptions in the Copyright Bill. These conclusions have been reached with due consideration given to the utilitarian justification for copyright protection, as well as of the dicta espoused in foreign law jurisdictions – namely, the USA and the UK. Analysis of the prevailing law in South Africa has also served to yield direction as to the positions which should be taken locally.

In considering the question of the copyrightability of APIs, the seminal case of *Oracle* in the USA has given two powerful yet contradicting views between the Federal Court and the court *a quo*. The decision of the court *a quo* has rested on the functional aspects of the API. It has found that the non-literal components of an API are so void of creativity, due to its utilitarian purpose, that its non-literal components cannot be held to constitute an expression. The Federal Court, however, has found its reasoning profoundly incorrect and argued rather that the non-literal aspects of the API are actually capable of being products of creative input. In the UK, the courts have followed the direction set by the EUCJ which has pronounced a blanket ban on the copyrightability of the functionality of computer programs. The reasons for this have included that the regional enactments do not allow for it; and that such rights would

create monopolies for copyright owners that would operate to the detriment of innovation.

In South Africa, it has become clear that no position has been taken on the copyrightability of APIs. As of yet, the question of the copyrightability of the non-literal components of a computer program, in general, has not been answered. I have suggested in this paper that the South African approach should mirror that which is favoured in the UK and by the court *quo* in the *Oracle* matter. This is so because APIs are purely functional creations developed simply to enable different programs to work in tandem. Its utilitarian nature is befitting, at best, of patent protection. Ownership rights over APIs cannot be found in the realm of copyright, where creativity, not mere functionality, is meant to find encouragement and reward. Finally, the detrimental effects on competition and access to the market that will be caused by the copyrightability of APIs is yet another nail in the coffin of pro-copyrightability arguments. It is important to secure the economic interests of developers in order to incentivise them to create these works; however, the cost of granting a broad exclusionary right in copyright is too great, and inappropriate. Persons who seek to create a monopoly over pieces of technology must be directed to the law of patents, where requirements for patentability are more stringent.

In the alternative, if APIs are found to be eligible for copyright protection, an approach similar to the fair use approach used in the USA is most preferred. The fair use doctrine has been shown to be flexible and thus able to accommodate the influx of issues created by changing technology. In this regard, an opportunity seems to be available to those who seek to benefit from the use of APIs. This much is clear from the *Oracle* judgment, where the court has been shown to be in favour of granting a fair use right over APIs on the basis of, amongst other things, their ability to facilitate interoperability between programs.

The fair dealing provisions in the UK have expectedly been shown to be highly restrictive. Legal certainty created by a limited set of permissible uses is outweighed by the fact that in most cases the use of computer programs, no less APIs, will not constitute fair dealing. Nevertheless, both the USA and the UK stances are favourable to the current South African approach because the reverse engineering of computer

software is permitted in those jurisdictions. This allows a programmer to develop her own API, if she has the relevant skill and access to resources.

I have argued that the fair dealing provisions in South Africa are in need of improvement. Although the exceptions which relate to quotations might possibly provide some semblance of an opportunity for users of APIs (although unlikely), the uncertainty related to how a court may interpret the provision, as well as the limited range of uses that the provision (and others like it) will create, has lead me to conclude that the fair dealing provisions as they are, are inadequate. The Copyright Bill displays a significant step towards progressive law that recognises the multitude of uses related to computer programs, and the need for such uses to be available to all persons in order to spur innovation. It has been shown to have some shortcomings, but marks an improvement on the current law. I have argued that the law should be developed in keeping with the provisions of the Copyright Bill which relate to computer programs, subject to the making of key changes which will enable the provision to be fully exploited by users within the bounds of reason.

The overarching implications of my arguments are for a greater balance of interests between vested parties to be achieved. Developers of APIs and users thereof alike can engage in mutually beneficial innovation as the availability of freely usable APIs spurs on the next wave of software development which can be used by all parties to improve their programs. I have stated that one of the ironic criticisms of copyright is that it is a hindrance to innovation. This is one of those cases where it is best to refrain from constricting the law of copyright so that its ultimate purpose – to breed creativity and innovation – may be fulfilled.

# **Bibliography**

## **Primary Sources**

### **Statutes**

Constitution of the Republic of South Africa, 1996.

Copyright Act 98 of 1978.

Designs Act 195 of 1993.

Patents Act 57 of 1978.

Performers' Protection Amendment Act 8 of 2002.

Trademarks Act 194 of 1993.

### **Government Gazette**

Copyright Amendment Bill 2015 in GN 646 GG 39028 of 27 July 2015.

### **International Agreements**

Agreement on the Trade-Related Aspects of Intellectual Property (TRIPS), 15 April 1994, Marrakesh Agreement Establishing the World Trade Organization, Annex 1C, 1869 U.N.T.S. 299, 33 I.L.M. 1197.

Berne Convention for the Protection of Literary and Artistic Works, 9 September 1886, as last revised at Paris on 24 July 1971, 1161 U.N.T.S. 30.

WIPO Copyright Treaty (WCT), 20 December 1996, S. Treaty Doc. No. 105–107 (1997), 36 I.L.M. 65 (1997).

## **Foreign Law**

### *Australia*

Copyright Amendment Act 158 of 2006.

### *European Union*

Directive 2009/24/EC of the European Parliament and of the Council on the legal protection of computer programs of April 23, 2009 (*Computer Program Directive 2009*).

Council Directive 91/250/EEC of May 14, 1991 *on the legal protection of computer programs (Computer Program Directive 1991)*.

EC, Directive 2001/29/EC of the European Parliament and of the Council of 22 May 2001 *on the harmonization of certain aspects of copyright and related rights in the information society*, [2001] O.J.L. 167.

### *USA*

Copyright Act of 1976, 17 U.S.C.

Digital Millennium Copyright Act Pub. L. 105-304-Oct 28, 1998.

United States of America: Constitution, 17 September 1787.

UK

Copyright, Designs and Patents Act 1988.

## Cases

*Cellular Insurance Managers (Pty) Ltd v Foschini Retail Group (Pty) Limited* [2011] ZASCA 85 (27 May 2011).

*Galago Publishers (Pty) Ltd. and Another v Erasmus* 1989 (1) All SA 431 (A).

*Gallo Africa Ltd v Sting Music (Pty) Ltd* 1103 JOC (A).

*Gramophone Record Co (Pty) Ltd v Ban-Nab Radio and TV* 1988 (2) SA 281 (D).

*Haupt t/a Softcopy v Brewers Marketing Intelligence (Pty) Ltd. and Others* 2006 (4) SA 458 (SCA).

*Holomisa v Argus Newspapers Ltd* 1996 (2) SA 588 (W).

*Payen Components SA Ltd v Bovic CC* 1995 4 SA 441 (A).

*Pastel Software (Pty) Ltd V Pink Software (Pty) Ltd and Another* 399 JOC (T).

*Poswa v MEC for Economic Affairs, Environment and Tourism, Eastern Cape* 2001 (3) SA 582 (SCA).

*S v Makwanyane and Another* 1995 (3) SA 391 (CC).

*Steenkamp and Others v Edcon Ltd* 2016 (3) SA 251 (CC).

*University of Cape Town v Cape Bar Council* 1986 (2) ALL SA 619 (A).

*Waylite Diary CC v First National Bank Ltd* 1995 1 SA 645 (A).

## Foreign Cases

### European Union

*Infopaq International A/S v Danske Dagblades Forening* (2009) ECR I-6569.

### United Kingdom

*Ashdown v. Telegraph Group Ltd.*, (2001) EWCA Civ 1142.

*Hubbard v Vosper* (1971) 1 All ER 1023 CA.

*SAS Institute Inc v World Programming Ltd* (2013) EWHC 69 (Ch).

### United States of America

*Atari Games Cor. v. Nintendo of America Inc.*, 975 F.2d 832 (Fed. Cir. 1992).

*Baker v Seldon* 101 U.S. 99 (1879).

*Campbell v. Acuff-Rose Music, Inc* 510 U.S. 569 (1994).

*Computer Associates International, Inc. v. Altai*, 982 F.2d 693 (2d Cir. 1992).

*Francis Day and Hunter Ltd v Twentieth Century Fox Corporation Ltd and others* [1940] AC 112.

*Folsom v Marsh* 9 F. Cas. 342 (C.C.D. Mass. 1841) (No. 4901).

*Gates Rubber Co. v. Bando Chemical Industries, Ltd.*, 9F.3d 823 (10<sup>th</sup> Cir. 1993).

*Harper & Row Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 549 (1985).

*Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989).

*Kelly v. Arriba Soft Corp.*, 336 F.3d 811, 820-21 (9th Cir. 2003).

*Kenrick & Co v Lawrence & Co* (1890) 25 QBD 99.

*Lotus Development Corp. v. Borland International. Inc.*, 49 F.3d 807 (1<sup>st</sup> Cir. 1995).

*Oracle Am., Inc. v. Google, Inc.*, 872 F. Supp. 2d 974 (N.D. Cal. 2012).

*Oracle Am., Inc. v. Google Inc.*, No. 13-1021 (Fed. Cir. 2014).

*Perfect 10, Inc. v. Amazon.com, Inc.*, 508 F.3d 1146 (9<sup>th</sup> Cir. 2007).

*Sega Enterprises. Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992).

*Sony Computer Entertainment, Inc., v Connectix Corporation*, 203 F.3d 596 (9<sup>th</sup> Cir. 2000).

*Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222 (3d Cir. 1986).

## **Secondary Sources**

### **Books**

David I Bainbridge *Introduction to Computer Law* 2 ed (1993) FT Prentis Hall.

Owen Dean and Alison Dyer *Dean & Dyer: Introduction to Intellectual Property Law* (2014) Oxford University Press, Oxford.

Bently, Lionel & Brad Sherman *Intellectual Property Law 3rd Edition* (2009) Oxford University Press, Oxford.

Burrell, Timothy D. *Burrell's South African Patent and Design Law* (1999) Butterworths Publishers.

Correa, Carlos M. *Intellectual Property Rights, the WTO and Developing Countries: The TRIPS Agreement and Policy Options* (2000, ).

AJC Copeling 'Copyright and The Act of 1978' ed (1978) Butterworths, Durban.

Hennie Klopper, Tania Pistorius, Brian Rutherford, Lee-Ann Tong, Pieter van der Spuy,

Andries van der Merwe *Law of Intellectual Property in South Africa* (2011) Lexis Nexis.

Vakul Sharma, *Information Technology-Law and Practice* 3ed (2011) Universal Law Publishing Co., New Delhi, India.

William Cornish and David Llewelyn *Intellectual Property: Patents, Copyrights, Trade Marks and Allied Rights* 6ed (2007) Sweet & Maxwell, Limited.

## **Commentaries**

The Anton Mostert Chair of Intellectual Property *Commentary on the Copyright Amendment Bill 2017* (2017).

Universtiy of Cape Town Intellectual Property Unit *Copyright Amendment Bill 2017: Comments* (2017).

## Internet Sources

Association of Independent Video and Filmmakers (ANF) *et al.*, 'Documentary Filmmakers' Statement of Best Practices in Fair Use' available at <http://centerforsocialmedia.org/files/pdf/fair-use-final.pdf> accessed on 12 February 2018.

Cambridge Dictionary 'Quote' available at <https://dictionary.cambridge.org/dictionary/english/quote> accessed on 12 February 2018.

English Oxford Living Dictionaries 'Quote' available at <https://en.oxforddictionaries.com/definition/quote> accessed on 12 February 2018.

Kurt Rowley 'Understanding Software Interoperability in a Technology Supported System of Education' available at <https://www.educause.edu/ir/library/pdf/cem9535.pdf> accessed on 12 February 2018.

National Alliance for Health Information Technology 'What Is Interoperability?' available at [www.nahit.org](http://www.nahit.org) accessed on 12 February 2018.

Nicolas Suzor 'Fair Use v Fair Dealing: How Australian Copyright Law Differs' available at <https://www.lifehacker.com.au/2017/07/fair-use-vs-fair-dealing-how-australian-copyright-law-differs/> accessed on 12 February 2018.

John-Paul Kamath 'APIs lead the Way to Interoperability' available at <http://www.computerweekly.com/feature/APIs-lead-the-way-to-better-interoperability> accessed on 12 February 2018.

Shireen Smith 'Are APIs Protected by Copyright? – Google v Oracle' available at <https://blogs.mulesoft.com/dev/quiz/api-expert-knowledge-test/> accessed on 12 February 2018.

Simon Crowfoot 'Five Computer Programs We Use Everyday' available at <http://www.iceni.com/blog/five-computer-programs-we-use-every-day/> accessed on 12 February 2018.

Timothy B. Lee 'The Court that Created the Patent Troll Mess is Screwing Up Copyright Too' available at <https://www.vox.com/2014/5/9/5699960/this-court-decision-is-a-disaster-for-the-software-industry> accessed on 12 February 2018.

Uri Sarid 'A Non-Apocalypse: APIs, Copyright and Fair Use' available at <https://www.wired.com/insights/2014/05/non-apocalypse-apis-copyright-fair-use/> accessed on 12 February 2018.

## Journal Articles

AJ Van der Walt & RM Shay 'Constitutional Analysis of Intellectual Property' (2014) 17 *Potchefstroom Electronic Law Journal* 54.

Allen Rosen 'Reconsidering the Idea/Expression Dichotomy' (1992) 26 *Universtiy of British Columbia Law Review* 263.

Arjun Krishnan 'Testing for Copyright Protection and Infringement in Non-Literal Elements of Computer Programs' (2005) 10 *Journal of Intellectual Property Rights* 9.

Caroline B Ncube 'Equitable Intellectual Property Protection of Computer Programs In South Africa: Some Proposals For Reform' (2012) 3 *Stellenbosch Law Review* 438.

Catherine Colston 'Fair Dealing: What is Fair?' (1995) 10 *Denning Law Journal* 91.

Charles R. McManis 'Intellectual Property Protection and Reverse Engineering of Computer Programs in the United States and the European Community' (1993) 8 *Berkeley Technology Law Journal* 26.

D.R Jones 'Commerciality and Fair Use' (2015) 15 *Wake Forest Journal of Business and Intellectual Property Law* 620.

David I. Bainbridge 'The Look and Feel of Computer Programs after Richardson v. Flanders' (1993) 2 *Laws, Computers and Artificial Intelligence* 269.

David R. Owen 'Interfaces and Interoperability in Lotus v. Borland: A Market-Oriented Approach to the Fair Use Doctrine' (1996) 64 *Fordham Law Review* 2381.

Deba Alam 'Oracle v Google' (2015) 26 *DePaul Journal Art, Technology and IP Law* 39

Edward Samuels 'The Idea-Expression Dichotomy in Copyright Law' (1989) 56 *Tennessee Law Review* 321.

Gluseppina D'Agostino 'Healing Fair Dealing – A Comparative Copyright Analysis of Canada's Fair Dealing to U.K. Fair Dealing and U.S. Fair Use' (2008) 53 *McGill Law Journal* 309.

Jean-Luc Piotraut 'An Authors' Rights-Based Copyright Law: The Fairness and Morality of French and American Law Compared' (2006) 24 *Cardozo Arts and Entertainment* 549.

Jacqueline Litpon 'The Law of Unintended Consequences: The Digital Millenium Copyright Act and Interoperability' (2005) 62 *Washington and Lee Law Review* 487.

John Abbot 'Reverse Engineering of Software: Copyright and Interoperability' (2003) 14 *J.L. & Inf. Sci* 7.

John Cady 'Copyrighting Computer Programs: Distinguishing Expression from Ideas' (2003) 22 *Temple Environmental Law and Technology Journal* 15.

John T. Soma, Gus Weinfeld & Letty Freinsein 'Software Interoperability and Reverse Engineering' (1994) 20 *Rutgers Technology and Law Journal* 189.

Katie Sykes 'Towards a Public Justification of Copyright' (2003) 61 *University of Toronto Faculty Law Review* 1.

Matt Flinders 'Protecting Computer Software – Analysis and Proposed Alternative' (2007) 7 *Journal of High Technology Law* 71.

Mark A Lemley & Ziv Shafir 'Who chooses Open-Source Software' (2011) 78 *University of Chicago Law Review* 139.

Lee-Ann Tong 'Copyright Protection for Computer Programs in South Africa: Aspects of Sui Generis Categorization' (2009) 12 *The Journal of World Intellectual Property* 266.

Lerinda Saint Waltrip 'Copyright Law – the Idea/Expression Dichotomy: Where Has It Gone' (1986) 11 *Southern Illinois University Law Journal* 411.

Lloyd L. Weinreb 'Fair Use' (1999) 67 *Fordham Law Review* 1291.

Mark A. Lemley 'Ex Ante v Ex Post Justifications for Intellectual Property' (2004) 71 *University of Chicago Law Review* 129.

Michael F. Morgan 'Trash Talking: The Protection of Intellectual Property Rights in Computer Software' (1994) 26 *Ottawa Law Review* 425.

Peter S. Menell "API Copyrightability Bleak House: Unravelling and Repairing the Oracle v Google Jurisdictional Mess' (2016) 31 *Berkely Technology Law Journal* 1515.

Richard Raysman & Peter Brown 'Copyright Infringement of Computer Software and the 'Altai' Test' *New York Law Journal* 9 May 2006.

Roux de Villiers 'Computer Programs and Copyright: The South African Perspective' (2006) 123 *SALJ* 315.

Sean O'Donnell 'Fair Use in the UK, Fair Use in the USA ... No Future, No Future' (2007) 7 *Hibernian Law Journal* 57.

Shubha Ghosh 'Duty, Consequences, & Intellectual Property' (2013) 10 *University of St. Thomas Law Journal* 801.

Vidya Atal & Kameshwari Shankar 'Developers' Incentives and Open Source Software Licensing: GPL vs BSD' (2015) 15 *B.E Journal of Economic Analysis and Policy* 1381.

## **Theses**

Sadulla Karjiker *Open-Source Software and the Rationale for Copyright Protection of Computer Programs* (published PhD thesis, Stellenbosch University, 2013) 34.

## **Reports**

National Commission on New Technological Uses of Copyrighted Works, Final Report (1979).