

INTERNATIONAL JOURNAL FOR LEGAL RESEARCH AND ANALYSIS



Open Access, Refereed Journal Multi-Disciplinary
Peer Reviewed

www.ijlra.com

DISCLAIMER

No part of this publication may be reproduced or copied in any form by any means without prior written permission of Managing Editor of IJLRA. The views expressed in this publication are purely personal opinions of the authors and do not reflect the views of the Editorial Team of IJLRA.

Though every effort has been made to ensure that the information in Volume II Issue 7 is accurate and appropriately cited/referenced, neither the Editorial Board nor IJLRA shall be held liable or responsible in any manner whatsoever for any consequences for any action taken by anyone on the basis of information in the Journal.

Copyright © International Journal for Legal Research & Analysis

EDITORIALTEAM

EDITORS

Dr. Samrat Datta

Dr. Samrat Datta Seedling School of Law and Governance, Jaipur National University, Jaipur. Dr. Samrat Datta is currently associated with Seedling School of Law and Governance, Jaipur National University, Jaipur. Dr. Datta has completed his graduation i.e., B.A.LL.B. from Law College Dehradun, Hemvati Nandan Bahuguna Garhwal University, Srinagar, Uttarakhand. He is an alumnus of KIIT University, Bhubaneswar where he pursued his post-graduation (LL.M.) in Criminal Law and subsequently completed his Ph.D. in Police Law and Information Technology from the Pacific Academy of Higher Education and Research University, Udaipur in 2020. His area of interest and research is Criminal and Police Law. Dr. Datta has a teaching experience of 7 years in various law schools across North India and has held administrative positions like Academic Coordinator, Centre Superintendent for Examinations, Deputy Controller of Examinations, Member of the Proctorial Board



Dr. Namita Jain

Head & Associate Professor

School of Law, JECRC University, Jaipur Ph.D. (Commercial Law) LL.M., UGC -NET Post Graduation Diploma in Taxation law and Practice, Bachelor of Commerce.

Teaching Experience: 12 years, AWARDS AND RECOGNITION of Dr. Namita Jain are - ICF Global Excellence Award 2020 in the category of educationalist by I Can Foundation, India. India Women Empowerment Award in the category of "Emerging Excellence in Academics by Prime Time & Utkrisht Bharat Foundation, New Delhi.(2020). Conferred in FL Book of Top 21 Record Holders in the category of education by Fashion Lifestyle Magazine, New Delhi. (2020). Certificate of Appreciation for organizing and managing the Professional Development Training Program on IPR in Collaboration with Trade Innovations Services, Jaipur on March 14th, 2019



Mrs.S.Kalpana

Assistant professor of Law

Mrs.S.Kalpana, presently Assistant professor of Law, VelTech Rangarajan Dr.Sagunthala R & D Institute of Science and Technology, Avadi. Formerly Assistant professor of Law, Vels University in the year 2019 to 2020, Worked as Guest Faculty, Chennai Dr.Ambedkar Law College, Pudupakkam. Published one book. Published 8Articles in various reputed Law Journals. Conducted 1Moot court competition and participated in nearly 80 National and International seminars and webinars conducted on various subjects of Law. Did ML in Criminal Law and Criminal Justice Administration. 10 paper presentations in various National and International seminars. Attended more than 10 FDP programs. Ph.D. in Law pursuing.



Avinash Kumar



Avinash Kumar has completed his Ph.D. in International Investment Law from the Dept. of Law & Governance, Central University of South Bihar. His research work is on "International Investment Agreement and State's right to regulate Foreign Investment." He qualified UGC-NET and has been selected for the prestigious ICSSR Doctoral Fellowship. He is an alumnus of the Faculty of Law, University of Delhi. Formerly he has been elected as Students Union President of Law Centre-1, University of Delhi. Moreover, he completed his LL.M. from the University of Delhi (2014-16), dissertation on "Cross-border Merger & Acquisition"; LL.B. from the University of Delhi (2011-14), and B.A. (Hons.) from Maharaja Agrasen College, University of Delhi. He has also obtained P.G. Diploma in IPR from the Indian Society of International Law, New Delhi. He has qualified UGC – NET examination and has been awarded ICSSR – Doctoral Fellowship. He has published six-plus articles and presented 9 plus papers in national and international seminars/conferences. He participated in several workshops on research methodology and teaching and learning.

ABOUT US

INTERNATIONAL JOURNAL FOR LEGAL RESEARCH & ANALYSIS
ISSN

2582-6433 is an Online Journal is Monthly, Peer Review, Academic Journal, Published online, that seeks to provide an interactive platform for the publication of Short Articles, Long Articles, Book Review, Case Comments, Research Papers, Essay in the field of Law & Multidisciplinary issue. Our aim is to upgrade the level of interaction and discourse about contemporary issues of law. We are eager to become a highly cited academic publication, through quality contributions from students, academics, professionals from the industry, the bar and the bench. INTERNATIONAL JOURNAL FOR LEGAL RESEARCH & ANALYSIS ISSN 2582-6433 welcomes contributions from all legal branches, as long as the work is original, unpublished and is in consonance with the submission guidelines.

SOFT BOUNDARIES; MAPPING THE UNCERTAIN TERRAIN OF SOFTWARE PATENTABILITY IN A POST- ALICE WORLD.

AUTHORED BY - AISWARYA THULASI

INTRODUCTION.

Patents can be granted for inventions across various fields as long as they meet the necessary criteria for patentability. This includes computer software and related inventions. However, the specific requirements for patentability vary according to the laws and regulations of each country. Intellectual property rights refer to the legal protections granted to original inventions and ideas conceived by individuals. These rights are typically awarded to creators, allowing them a limited monopoly for a certain duration. Intellectual property rights are generally classified into two primary categories: the first pertains to aesthetic creations, while the second focuses on industrial innovations. The fundamental aim of granting these exclusive rights is to safeguard the effort, time, and resources invested by individuals in developing new creations or ideas, whether for industrial applications or other purposes.¹ The growing movements of globalization and intensified market competition have made the patentability of technological innovations increasingly complex, particularly impacting industrialized countries. Following the Uruguay Round of GATT², the WTO was established, mandating all member nations to revise and align their laws with WTO regulations. India, as a member of the WTO and a participant in the TRIPs agreement, is subject to these requirements. Article 27(1) of the TRIPs agreement outlines the criteria for patentability³, indicating that all inventions meeting these criteria are eligible for patents. This includes software programs and computer-related inventions, which can be patented as long as they fulfil the specified conditions. In India, the Patents Act of 1970 specifically states that 'computer programs' in their own right are not eligible for patent protection.⁴ This stance has been upheld in India's latest Patent (Amendment) Bill of 2005. Consequently, computer programs themselves continue to receive protection under copyright law.

¹ YASHWARDHAN BANDI, Patenting of Software: A Move in the Right Direction, AIR Pvt. Ltd. Nagpur, Published in CLC 2006.

² The Final Act Embodying the Results of the Uruguay Round of Multilateral Trade Negotiations, Marrakesh, 1994.

³ TRIPs Agreement art. 27(1), Marrakesh Agreement Establishing the World Trade Organization, Annex 1C, Apr. 15, 1994, 1869 U.N.T.S. 299

⁴ Patents Act, No. 39 of 1970, § 3(k) (India).

I. SOFTWARE: WHAT IT TRULY MEANS?

To engage in a meaningful discussion about the patentability of software, it is crucial to first grasp the fundamental concepts related to computers. This topic can often lead to confusion, particularly in distinguishing between hardware and software. Hardware refers to the tangible components of a computer system, while software consists of precisely crafted instructions that enable the hardware to perform specific tasks.⁵ A straightforward interpretation of this definition can lead to potential doubts and uncertainties. There is confusion regarding what precisely constitutes an instruction for a machine. If software is merely a set of instructions expressed as thoughts or written down, it may not be capable of operating a machine as intended. Conversely, if software exists in a tangible form, such as SD cards, chips, or magnetic tape, and is physically connected to the computer system, does it not become part of the hardware?⁶ Typically, software is distributed on physical media, such as a disk, which is inserted into a computer. The machine then reads this disk to execute the tasks designated by the software. On a technical level, the disk may contain a grid of magnetic heads, where each cell of the software is composed of specific information. The orientation of these cells, arranged in north-south directions, forms the basis for machine code and information that can be converted into binary code.⁷

Without resorting to technical jargon, one can consider software as a distinct pattern in its own right. However, this definition does not clarify the required form of that pattern, whether it be written, recorded, or represented in other physical modes, or if it simply exists as an idea or thought. Furthermore, this definition fails to distinguish between what constitutes software and what is classified as data. The concept of software cannot be confined to a single definition or meaning. It goes beyond mere assumptions and is continually undergoing both qualitative and quantitative transformations.

1. SOFTWARE IS A MACHINE – READABLE PATTERN

One fundamental requirement for software is that it should be readable or interpretable by the designated machine.⁸ To achieve this readability, two essential criteria must be met.

⁵ Peter Suber, what is Software? Vol. II, No. 2, Journal of Speculative Philosophy, 89, 89, 1988.

⁶ Peter Suber, what is Software? Vol. II, No. 2, Journal of Speculative Philosophy, 89, 90, 1988.

⁷ Ibid

⁸ Melanie Cook, Stephen Quinn, David Waltermire, Dragos Prisaca, Security Content Automation Protocol (SCAP) Version 1.2 Validation Program Test Requirements, Rev. 4, NIST- U.S. 1, 11, 2016.

First, the representation must utilize an appropriate material form. This means that the encoding should align with the machine's capabilities for interpretation. For instance, if the machine is designed to detect magnetic fields, the information should be encoded in terms of magnetic poles rather than using switches, pits, chips, or punched cards.⁹

The second requirement is that the created pattern must be in a format that machines can understand. While English can be digitized and is relatively straightforward, machines struggle to execute software instructions written in digitalized English. High-level programming languages such as Python, Java, or C++ need to be not only digitized but also translated to bridge this gap. Ultimately, these high-level languages are crafted for human comprehension, and to be processed by machines, they must be converted into machine language.¹⁰

2. SOFTWARE IS A MACHINE- EXECUTABLE PATTERN.

Software can be understood as a pattern that machines can read and execute. This leads to the question: If a code can be executed by a machine, what happens when it doesn't run correctly? Issues such as bugs can prevent the code from executing as intended. The rapid advancement of technology has resulted in increasingly complex machines, which may lead to built-in code that fails to function consistently.

It is possible for a code to be well-written yet poorly executed due to various factors. In this context, the term "executable" in the definition of software can be further clarified: patterns that fail to function as software today may become operational in the future if feasible adjustments are made—through debugging, physical modifications, language conversions, or even adapting them for different machines. These changes could enable previously idle code or patterns to serve their intended purpose as software in the future.¹¹ However, it can be understood that codes, whether currently functional or potentially executable with relevant modifications, can be considered software.

3. PORTABILITY OF SOFTWARE.

Hardware operates as distinct physical entities, whereas software exhibits remarkable flexibility. Software can be replicated, transferred, and executed across diverse systems with

⁹ Peter Suber, what is Software? Vol. II, No. 2, Journal of Speculative Philosophy, 89,98, 1988.

¹⁰ Ibid.

¹¹ Peter Suber, what is Software? Vol. II, No. 2, Journal of Speculative Philosophy, 89,95, 1988.

minimal or no alterations. What stands out is not merely the ability to access numerous software applications on a single hardware device, but rather the capacity for the same software to function seamlessly across multiple hardware platforms, such as smartphones, laptops, and desktop computers. In essence, the most striking characteristic of software is its portability. While hardware can be programmed, software truly exemplifies the concept of adaptability.¹² The information stored in the magnetic fields of a disk can be transferred to solid areas and blank spaces on punched cards, and the process can also go in reverse. More commonly, this information is moved to the random-access memory (RAM) of a computer, making it accessible for the processor. This transfer occurs each time a program is executed on a modern computer.¹³

Portability can be included in our understanding of software, but it's important to recognize that it is secondary, not foundational. It results from defining software primarily as patterns and secondarily as embodied patterns. Even though it is secondary, portability still plays an important role in our definition.

II. DECODING THE DILEMMA: THE PATENT PUZZLE OF SOFTWARE INNOVATION.

Software has consistently posed challenges within the realm of intellectual property. Its ambiguous and often incomplete relationship with copyright, trade secrets, and patents complicates the assignment of appropriate protections.¹⁴ Courts around the globe have grappled with significant difficulties in applying traditional patent eligibility tests to computer software, largely due to the absence of a universally defined standard.¹⁵ In 2014, the U.S. Supreme Court's decision in the case of *Alice Corp. Pty. Ltd. v. CLS Bank International*¹⁶ raised important questions about the criteria for patenting software. Historically, software was not considered patentable because it was viewed as merely a mathematical algorithm. However, this case required a new analysis of traditional patenting criteria. While the Alice decision did not outright ban software patentability, it introduced a test for patent eligibility that made it

¹² Peter Suber, what is Software? Vol. II, No. 2, Journal of Speculative Philosophy, 89,103, 1988.

¹³ Ibid.

¹⁴ Gregory J. Maier, Software Protection- Integrating Patent, Copyright and Trade Secret Law, 69 J. PAT& TRADEMARK OFF. SOC'Y 151, 151 (1987).

¹⁵ Andrew Beckerman- Rodau, What Should Be Patentable? – A Proposal for Determining the Existence of Statutory subject Matter Under 35 U.S.C. Section 101, 13 WAKE FOREST J BUS. & INTELL. PROP. L. 145, 147-148 (2013).

¹⁶ *Alice Corp. Pty. Ltd. v. CLS Bank Int'l*, 134 S. Ct. 2347 (2014).

more challenging for software to be patented. This ruling raised significant questions about the eligibility of previously granted patents, potentially invalidating several legal precedents.¹⁷ The Federal Circuit seems to be expanding upon Alice's two-part analysis by introducing what can be considered a "technological art"¹⁸ standard. This approach evaluates whether the claims in question are aimed at providing a technological solution to a specific technological issue.¹⁹

1. TRACING THE HISTORY OF SOFTWARE PATENTS.

The criteria for software patenting have changed over the years. There has been a shift from software generally being ineligible for patents to a considerable number of software programs now qualifying for patent protection.²⁰ In the case of *Gottschalk v. Benson*²¹, the court observed, that computer algorithms are ineligible for patent protection. The disputed claims pertained to an algorithm designed to transform binary coded decimals into actual binary numbers through a mathematical formula.²² The Court determined that this algorithm represented nothing more than an abstract concept and that granting a patent for it would effectively restrict the fundamental mathematical formula it was based on.²³ The court established the "machine-or-transformation"²⁴ test, stating that the key to determining the patentability of a process claim, when specific machines are not involved, lies in the transformation and conversion of an article into a different state or thing.²⁵

In *Parker v. Flook*²⁶, it was asserted that adding a novel algorithm to an existing process does not qualify for patent protection, thereby reinforcing the ruling established in the *Gottschalk* case. The case in question revolved around a method for adjusting an alarm threshold in a catalytic conversion process.²⁷ The Court concluded that the only distinction between the

¹⁷ *OIP Techs., Inc. v. Amazon.com*, 788 F.3d 1359, 1364 (Fed. Cir. 2015); *Internet Patents Corp. v. Active Network Inc.*, 790 F.3d 1343, 1348–49 (Fed. Cir. 2015); *Content Extraction & Transmission LLC v. Wells Fargo Bank, Nat'l Ass'n*, 776 F.3d 1343.

¹⁸ Austin Steelman, Note, Curiouser and Curiouser! Why the Federal Circuit Can't Make Sense of Alice, 98 J. PAT. & TRADEMARK OFF. SOC'Y 374, 384 (2016).

¹⁹ Joseph Allen Craig, DECONSTRUCTING WONDERLAND: MAKING SENSE OF SOFTWARE PATENTS IN A POST-ALICE WORLD, *Berkeley Technology Law Journal*, 2017, Vol. 32, Annual Review (2017), pp. 359–378.

²⁰ Brianna Dolmage, The Evolution of Patentable Subject Matter in the United States, 27 *WHITTER L. REV.* 1023, 1034 (2006).

²¹ *Gottschalk v. Benson*, 409 U.S. at 68, 71–72.

²² *Id.*

²³ *Id.*

²⁴ Joyce Li, Note, Pre-emption and Diagnostics: The Federal Circuit's Misguided Return to the Machine-or-Transformation Test, 32 *BERKELEY TECH. L.J.* 24–26 (2017).

²⁵ *Supra* note 21 at 5.

²⁶ *Parker v. Flook*, 437 U.S. 584, 594–95 (1978).

²⁷ *Id.*

claims presented and the existing prior art was the particular algorithm employed to modify the alarm threshold.²⁸ As a result, the Court found that the claims essentially represented a novel way of calculating the alarm threshold, thus rendering them ineligible for patent protection.²⁹ While the Court noted that its ruling should not be interpreted as a blanket rejection of software patents, the combination of the decisions in *Gottschalk* and *Flook* seemingly cast doubt on the eligibility of all software patents.³⁰ *Gottschalk* determined that algorithms could not be patented, while *Flook* indicated that even innovative algorithms introduced to existing processes were also considered ineligible for patenting.³¹

In the landmark case of *Diamond v. Diehr*³², the court made a significant departure from its previous legal interpretations by ruling for the first time that software is eligible for patent protection. The disputed claims pertained to a method for curing synthetic rubber, ensuring that the moulded products achieved proper curing.³³ This method utilized software that incorporated an algorithm to dynamically adjust the curing time based on the temperature of the rubber, which was monitored at regular intervals. Notably, the claims encompassed additional steps for heating and removing the rubber, alongside the algorithm itself. The Court determined that the claims collectively focused on the comprehensive process of rubber curing, rather than solely on the algorithm associated with the software. While this case did not explicitly overturn the findings from *Flook*, differentiating between the two cases can be quite challenging.³⁴

The ability to obtain software patents became significantly more accessible following the Federal Circuit ruling in the case of *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*³⁵ In this ruling, the court determined that an abstract concept that results in a practical and tangible outcome qualifies for patent protection. The case involved claims related to an investment system for mutual funds, designed to operate via a computer, which allowed different mutual funds to consolidate their assets into a single investment entity.³⁶ The Federal

²⁸ *Id.*

²⁹ *Id.*

³⁰ Ognen Zivojnovic, Note, Patentable Subject Matter After Alice—Distinguishing Narrow Software Patents from Overly Broad Business Method Patents, 30 BERKELEY TECH. L.J. 807, 813 (2015).

³¹ *Gottschalk v. Benson*, 409 U.S. 63, 68, 71–72 (1972); *Flook*, 437 U.S. at 594–95.

³² *Diamond v. Diehr*, 450 U.S. 175, 192–93 (1981).

³³ *Id.*

³⁴ Bernard Chao, Finding the Point of Novelty in Software Patents, 28 BERKELEY TECH. L.J. 1217, 1235 (2013).

³⁵ *State St. Bank & Tr. Co. v. Signature Fin. Grp.*, 149 F.3d 1368, 1373.

³⁶ *Id.*

Circuit concluded that these claims met the criteria for patent eligibility because they involved the transformation of data through a "machine" by performing a series of mathematical operations. This interpretation enabled individuals to bypass the subject matter requirements by simply incorporating conventional business methods alongside technical terminology in their claims.³⁷

The case of *Bilski v. Kappos*³⁸ addressed the requirement for subject matter in patent law, where the Supreme Court determined that a patent for a hedging method was not eligible because it essentially monopolized an abstract idea. The Court dismissed the notion that the "machine-or-transformation" test should be seen as the sole standard for assessing patent eligibility, stating that it is merely one of several factors to consider. Additionally, the Court reiterated that achieving a "useful, concrete, and tangible result" does not guarantee patent protection.³⁹

In the case of *Alice Corp. Pty. Ltd. v. CLS Bank Int'l*⁴⁰, the Court established a two-step evaluation inspired by the 2012 decision in *Mayo Collaborative Services v. Prometheus Labs., Inc.*⁴¹, applying it specifically to software patents. The claims at issue in Alice involved a type of escrow arrangement, where two parties could make a contract to exchange funds at a later date. The Alice test consists of two key components. First, the court must determine whether the claims pertain to an abstract idea. If they do, the second step requires the court to assess if there exists an "inventive concept" substantial enough to render the claims eligible for patent protection. The Court concluded that the claims were indeed centred around the abstract notion of escrow transactions between two entities. Furthermore, it ruled that there was no inventive concept present, as the claims merely represented a standard computer execution of an abstract idea. Consequently, the claims were deemed ineligible for patent protection.⁴²

After the Alice case, numerous software patents were revoked because one could no longer make a basic idea patentable merely by claiming it runs "on a computer."⁴³

³⁷ Robert A. Hulse, Patentability of Computer Software After *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*: Evisceration of the Subject Matter Requirement, 33 U.C. DAVIS L. REV. 491, 519 (2000).

³⁸ *Bilski v. Kappos*, 561 U.S. 593, 611–12 (2010).

³⁹ *Id.*

⁴⁰ *Alice Corp. Pty. Ltd. v. CLS Bank Int'l*, 134 S. Ct. 2347, 2355–60 (2014).

⁴¹ See *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, 132 S. Ct. 1289 (2012).

⁴² *Supra* note 40 at 7.

⁴³ *Internet Patents Corp. v. Active Network Inc.*, 790 F.3d 1343, 1348–49 (Fed. Cir. 2015); *OIP Techs., Inc. v. Amazon.com*, 788 F.3d 1359, 1363–64.

III. PATENT ELIGIBILITY IN THE POST - ALICE ERA⁴⁴

The introduction of the two-tier test for software patentability established in the Alice case made obtaining patent protection for software more challenging. However, several cases have successfully secured patents by applying the criteria outlined in this landmark decision. This indicates that the Alice case did not complicate the process of obtaining software patents; rather, it aimed to create a standard that ensures only deserving inventions receive patent protection.

Examining five significant cases where software patents were granted by the U.S. Supreme Court clarifies how the Alice decision has influenced the patent landscape for software.⁴⁵

1. DDR HOLDINGS, LLC V. HOTELS.COM, L.P.⁴⁶ - SOLUTIONS LEVERAGING TECHNOLOGY THAT COULD BE PATENTABLE.

The case of *DDR Holdings, LLC v. Hotels.com, L.P.* explored the notion that solutions which are fundamentally based in technology and address issues in the context of computer networks could be eligible for patent protection. The claims presented in this case focused on a method for generating a webpage on a host website that maintained the same aesthetic as the host site while featuring content from a third-party retailer. This innovative approach enabled users to purchase products directly from a third-party merchant without leaving the host site.⁴⁷ By retaining users on the host website, this method enhanced e-commerce opportunities, thus increasing the likelihood of purchases. Both the USPTO and the district court affirmed the validity of the associated patents.

On appeal, the Federal Circuit upheld the patent eligibility of the claims, although it did not clearly state if they met the first step of the Alice test regarding abstract ideas. Instead, the court determined that the claims included an inventive concept, regardless of their abstract nature. It highlighted that these claims were inherently tied to computer technology and effectively addressed a problem unique to the internet: preventing users from being redirected to external websites. The proposed solution modified conventional internet practices, standing apart as it

⁴⁴ Joseph Allen Craig, DECONSTRUCTING WONDERLAND: MAKING SENSE OF SOFTWARE PATENTS IN A POST-ALICE WORLD, Berkeley Technology Law Journal, 2017, Vol. 32, Annual Review (2017), pp. 359-378.

⁴⁵ Id.

⁴⁶ *DDR Holdings, LLC v. Hotels.com, L.P.*, 773 F.3d 1245, 1257 (Fed. Cir. 2014).

⁴⁷ Id.

did not involve any standard mathematical algorithms or fundamental business methods. Additionally, the court noted that the solution offered did not monopolize all possible alternatives and therefore was not excessively preclusive. This emphasis on the detail and specificity of the solution contributed to the conclusion that it represented more than just the appropriation of an abstract idea.

2. **ENFISH, LLC V. MICROSOFT CORP.⁴⁸ – RECOGNIZING THAT SOFTWARE ISN'T ALWAYS ABSTRACT.**

The case of *Enfish, LLC v. Microsoft Corp.* established that not all software patents are automatically considered abstract concepts. Enfish's patent sought to enhance the traditional relational database model, which typically organizes data across multiple tables—such as having separate tables for individuals and companies. In contrast, Enfish's innovation involved a self-referential database, consolidating all information within a single table, leading to improved data storage, easier setup, and quicker search capabilities. Initially, the district court ruled the claims invalid under § 101, arguing they related to the abstract idea of organizing and retrieving data in tables.⁴⁹

However, the Federal Circuit found that the lower court had oversimplified the claims and overlooked their advantages. It highlighted that the patents were not abstract under the first step of the Alice test, emphasizing that some software claims reflect a genuine technological enhancement. The court clarified that the level of abstraction must be appropriately set, asserting that the capacity to operate on general-purpose computers and the absence of specific hardware do not invalidate the claims.⁵⁰

3. **BASCOM GLOBAL INTERNET SERVICES V. AT&T MOBILITY LLC⁵¹ – DISCOVERING INNOVATIVE CONCEPTS IN ABSTRACT SOFTWARE.**

In the case of *BASCOM Global Internet Services v. AT&T Mobility LLC*, the Federal Circuit addressed the patent eligibility of claims related to internet content filtering. The patents focused on a method that combined various filtering techniques previously used: individual software on user computers, local server filtering, and ISP-level filtering, each having its own advantages and disadvantages. While traditional methods could be too generic and difficult to

⁴⁸ *Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327, 1335 (Fed. Cir. 2016). 81. Id. at 1330.

⁴⁹ Id.

⁵⁰ Id.

⁵¹ *BASCOM Glob. Internet Servs. v. AT&T Mobility LLC*, 827 F.3d 1341, 1352 (Fed. Cir. 2016).

maintain, BASCOM's approach allowed for customized filtering at the ISP level, using subscriber information to tailor options for individual users. Initially, the district court deemed the patent invalid, categorizing it as an abstract idea without an inventive concept, focusing solely on individual claim components.⁵²

However, the Federal Circuit disagreed, acknowledging that the claims were indeed abstract but also represented a significant technological improvement. The court emphasized that the inventive aspect lay in the combination of elements—specifically, the deployment of a customizable filtering tool at a remote location, which was not a conventional method. This ruling highlights BASCOM's contribution to overcoming the limitations of existing content filtering solutions.⁵³

4. *MCRO, INC. V. BANDAI NAMCO GAMES AMERICA INC.*⁵⁴ – HIGHLIGHTING THE IMPORTANCE OF PRE-EMPTION.

The case of *McRO, Inc. v. Bandai Namco Games America Inc.* is significant due to its emphasis on the concept of pre-emption in patent law. The patents involved pertained to methods for automating the synchronization of lip movements and facial expressions for animated characters. Specifically, the patents revolved around the automation of setting keyframes—critical frames that contribute to the visual flow of animation. This innovation allowed animators to achieve superior results in a more efficient manner than was previously feasible. During the trial, the district court deemed the claims ineligible, arguing that they were overly broad and pre-emptive under § 101. The court believed that the claims were attempting to patent the general idea of employing rules to automate keyframe setting, rather than a particular method for doing so.⁵⁵

However, the Federal Circuit found that the claims were not directed to an abstract idea, aligning with the precedent set in *Elfish*. The appeals court criticized the district court for simplifying the claims and for failing to recognize the distinctive limitations detailed in them. By analysing the claim limitations alongside the patent specifications, the Federal Circuit concluded that the rules for automating the animation process were specific enough to avoid broad pre-emption. This decision highlighted the risk associated with claiming a broad category

⁵² Id.

⁵³ Id.

⁵⁴ *McRO, Inc. v. Bandai Namco Games Am. Inc.*, 837 F.3d 1299, 1314–16 (Fed. Cir. 2016).

⁵⁵ Id.

of inventions, yet acknowledged the existence of alternative methods that rendered the claims less pre-emptive. The Federal Circuit clarified that the claims did not inhibit “all techniques for automating 3-D animation that utilize rules.” Furthermore, the importance of this case lies in the fact that the claims were directed towards a specific process rather than a general advancement in computing or internet technologies, emphasizing an improvement in technology through a defined method rather than a vague concept.⁵⁶

5. *AMDOCS (ISR.) LTD. V. OPENET TELECOM, INC*⁵⁷.- ESTABLISHING THE TECHNOLOGICAL ARTS EVALUATION CRITERIA.

In the case of *Amdocs (Isr.) Ltd. v. Openet Telecom, Inc.*, the Federal Circuit determined that several network accounting patents met the patentability criteria under the technological arts test. The four patents involved pertained to a system⁵⁸ aimed at addressing billing and accounting challenges encountered by network service providers.⁵⁹ Specifically, these patents focused on enhancing the collection of network accounting data through a distributed architecture. This approach reduced the strain on both network and system resources by positioning the data close to its sources, thereby alleviating network congestion while ensuring centralized access to the information. Initially, the district court ruled that all four patents were ineligible under § 101, asserting that the claims were centred on the abstract idea of correlating two network accounting records to improve the first record.

Furthermore, during the second step of the Alice test, the court found no adequate inventive concept to establish eligibility. However, the Federal Circuit overturned the district court's decision, declaring the claims to be patentable. The court drew comparisons to other legal precedents, noting similarities to cases like *BASCOM* and *DDR Holdings*. It identified that *Amdocs's* claims addressed challenges stemming from the substantial data flow associated with large databases—similar to how claims in *DDR Holdings* tackled issues with conventional internet hyperlink protocols that hindered website retention of visitors. Additionally, the court observed that the claims, taken as an ordered combination, represented an invention that transcended mere routine or conventional use of technology. The Federal Circuit further established that, akin to the findings in *Elfish*, the claims presented an unconventional technological solution—namely, enhancing data in a distributed manner—to a significant

⁵⁶ Id.

⁵⁷ *Amdocs (Isr.) Ltd. v. Openet Telecom, Inc.*, 841 F.3d 1288, 1302–03 (Fed. Cir. 2016).

⁵⁸ U.S. Patent Nos. 7,631,065, 7,412,510, 6,947,984, and 6,836,797; *Amdocs*, 841 F.3d at 1290.

⁵⁹ *Amdocs*, 841 F.3d at 1291.

technological problem, specifically the overwhelming data flows that had previously necessitated extensive databases. Although the claims incorporated standard computing components, they required these components to function in a nontraditional way that resulted in an improvement in computer performance. Consequently, Amdocs reaffirmed the necessity for claims to offer a technological resolution to a technical issue.⁶⁰

CONCLUSION

In summary, the current landscape of software patent eligibility remains fraught with uncertainty, primarily due to the complexities introduced by the Supreme Court's landmark decision in *Alice Corp. v. CLS Bank International*. This case established a framework that has left many innovators questioning what constitutes a patentable invention in the realm of software. The Federal Circuit's recent shift towards implementing a “technological arts” test aims to navigate this ambiguity by providing clearer guidance on which software innovations can be protected under patent law.

The “technological arts” test serves a dual purpose: it effectively filters out generic business methods that simply use software as a vehicle and concentrates instead on inventions that represent meaningful advancements in technology. This distinction is crucial because it allows for the protection of genuine innovations—those that enhance the functionality of software, computers, or the internet—while preventing an influx of patents on abstract concepts that do not contribute to technological progress. Promoting originality in software development is essential for fostering a competitive and innovative environment, as software plays an integral role in nearly every aspect of modern life, from healthcare to finance and beyond. A strong framework for patent protection not only incentivizes inventors to invest in research and development but also reassures stakeholders that their innovations will be safeguarded against infringement.

Given the significance of these issues, it is vital for the Supreme Court to affirm the Federal Circuit's application of the “technological arts” test. By doing so, the Court would establish a clearer legal standard that aligns with the original intent of the *Alice* decision. This endorsement would help solidify a coherent approach to software patents, ultimately encouraging technological advancements while ensuring that the patent system effectively

⁶⁰ Supra note 57 at 10.

supports genuine innovation rather than hindering it with vague or overly broad claims. With a more defined path forward, inventors can navigate the patent landscape with greater confidence, leading to a richer tapestry of software innovations that can drive progress across multiple sectors.

