IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15589

Open Source Software Licenses: A Comparative Analysis of GPL, MIT, and Apache

Manjunath S Rakaraddi, Bhairam V Pawar, Rahul M, Raviteja Javali,

Muhibur Rahman T. R

Department of Computer Science and Engineering

Ballari Institute of Technology and Management (BITM), Ballari, India

Abstract—Open-source software licensing plays a vital role in modern software development by defining the legal permissions, responsibilities, and limitations associated with the use, modifi-cation, and distribution of software. Among the numerous open-source licenses available, the GNU General Public License (GPL), MIT License, and Apache License 2.0 are widely adopted due to their distinct licensing models and practical significance in both academic and industrial environments. This paper presents a comparative study of these three major open-source licenses by analyzing their features, permissions, restrictions, distribution policies, and patent considerations. The GPL license emphasizes software freedom through its copyleft approach, requiring deriva-tive works to remain open source, whereas the MIT License pro-vides maximum flexibility with minimal restrictions. The Apache License 2.0 combines permissive licensing with explicit patent protection, making it suitable for commercial and enterprise applications. The study highlights the advantages, limitations, and real-world applications of each license to help developers, researchers, and organizations select appropriate licensing strate-gies for software projects. The analysis demonstrates that the choice of an open-source license significantly impacts software collaboration, legal compliance, commercial adoption, and long-term project sustainability.

Index Terms—Open-source software, GPL, MIT License, Apache License 2.0, software licensing, copyleft, permissive licenses, patent protection, software compliance

I. INTRODUCTION

The emergence and proliferation of open-source software has fundamentally transformed the landscape of software development and distribution. Open-source software not only provides access to source code but also establishes a col-laborative ecosystem where developers worldwide contribute to improving, debugging, and extending software projects. However, the legal framework governing open-source projects is defined by their licensing models, which serve as the foun-dation for determining how software can be used, modified, and distributed [1].

Software licenses are critical legal instruments that define the rights and responsibilities of both developers and end-users. They govern the distribution of source code, the per-mitted modifications, commercial usage, and the obligations imposed upon derivative works. The choice of an appropriate open-source license has significant implications for project adoption, commercial viability, and community participation [2].Among the numerous open-source licenses available, three licenses dominate the open-source ecosystem: the GNU Gen-eral Public License (GPL), the MIT License, and the Apache License 2.0. These licenses represent different philosophies and approaches to software freedom. The GPL, in its various versions (GPLv2, GPLv3), embodies the ”copyleft” principle, ensuring that all derivative works maintain the same open-source nature. The MIT License, by contrast, is a minimalist permissive license that grants users maximum freedom with virtually no restrictions. The Apache License 2.0 balances permissiveness with explicit patent protection, making it par-ticularly attractive for enterprise and commercial applications [3].

Understanding the distinctions between these licenses is cru-cial for software developers, project maintainers, open-source contributors, and organizations evaluating licensing strategies. A misaligned license choice can lead to legal complications, license incompatibilities, inability to incorporate third-party code, and reduced project adoption. Conversely, selecting an appropriate license can foster community participation, ac-celerate development, facilitate commercial partnerships, and ensure long-term project sustainability [4].

The motivation for this comparative study stems from the widespread adoption of these licenses and the practical challenges faced by developers in selecting the most suit-able license for their projects. Despite the availability of resources and license choosers (such as choosealicense.com), many developers lack a comprehensive understanding of

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15589

the technical and legal implications of each license. This paper addresses this gap by providing a detailed comparative analysis of GPL, MIT, and Apache licenses, examining their features, permissions, restrictions, and suitability for different project contexts.

The remainder of this paper is structured as follows: Section II presents a literature review of existing work on open-source licensing and comparative studies. Section III outlines the methodology employed in this study. Section IV provides a comprehensive comparison of the three licenses and their characteristics. Section **??** discusses the implications and real-world applications of each license. Finally, Section V concludes the paper and suggests directions for future work.

II. LITERATURE REVIEW

The academic study of open-source software licensing has evolved significantly over the past two decades. Early founda-tional work in this area focused on understanding the economic incentives and social motivations behind open-source develop-ment [5]. Raymond's seminal work on the cathedral and bazaar model highlighted how open-source development differs from proprietary software development in its collaborative approach and community-driven innovation [6].

Licensing compliance and compatibility have been subjects of considerable research interest. Wheeler and others have examined license compatibility matrices and the challenges of integrating code licensed under different open-source li-censes. Their work demonstrated that license incompatibilities can significantly hinder software integration and create legal risks [3]. The Free Software Foundation's maintenance of a comprehensive license list and compatibility matrix has been instrumental in promoting license awareness among develop-ers.

Patent considerations in open-source licensing have become increasingly important, particularly with the rise of software patents and patent litigation. Studies have shown that explicit patent grants in licenses, such as those in the Apache License 2.0, provide stronger legal protections against patent claims compared to licenses without such provisions [7]. This dis-tinction has led many enterprise organizations to prefer patent-aware licenses over traditional permissive licenses.

The commercial adoption and viability of open-source projects have been extensively studied in relation to licensing choices. Research indicates that permissive licenses tend to have higher adoption rates in the commercial sector due to their minimal restrictions on proprietary derivative works [8]. However, GPL-licensed projects have demonstrated re-markable resilience and success in enterprise environments, particularly in infrastructure and server-side applications.

Recent empirical studies examining large-scale software repositories (such as GitHub) have revealed interesting pat-terns regarding license distribution, adoption trends, and their correlation with project characteristics. These studies have shown that license choice varies significantly across different software domains, programming languages, and project sizes [2]. The MIT and Apache licenses have gained substantial ground in recent years, while GPL remains dominant in certain categories of software.

Despite the availability of comparative information in blogs, tutorials, and license documentation, academic treatment of comparative license analysis remains limited. Most existing work focuses on legal or economic aspects rather than pro-viding comprehensive technical and practical guidance for developers. This study aims to fill this gap by providing a detailed, technically grounded comparison that addresses both theoretical and practical dimensions of license selection.

III. METHODOLOGY

This comparative study employs a systematic and structured approach to analyzing three major open-source licenses. Themethodology comprises the following components:

A. License Documentation Analysis

The official license texts and supporting documentation for GPL (versions 2 and 3), MIT License, and Apache License
2.0 were obtained from authoritative sources, including the Free Software Foundation (FSF) and the Apache Software Foundation (ASF). The legal texts were thoroughly examined to extract and understand the core provisions, conditions, and restrictions imposed by each license.

B. Feature Extraction and Categorization

Key features and characteristics were systematically identi-fied and categorized across multiple dimensions:

- *Permission Scope*: What users are permitted to do with the software
- *Restrictions and Conditions*: What limitations and obli-gations apply

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15589

- *Patent Provisions*: Explicit or implicit patent-related guar-antees
- *Copyleft Mechanisms*: Whether derivative works must maintain the same license
- *Commercial Usage*: Permissions and constraints on com-mercial applications
- *Distribution Requirements*: Conditions for distributing modified versions
- *Attribution Requirements*: Obligations to acknowledge the original author

C. Comparative Framework

A comprehensive comparison matrix was developed to facilitate systematic evaluation across multiple dimensions. The matrix allowed for consistent and objective comparison of license features, making it possible to identify similarities, differences, and trade-offs between licenses.

D. Real-World Application Review

Projects and organizations using each of these licenses were examined to understand practical adoption patterns and use cases. This included reviewing large-scale open-source projects, enterprise software, and community-developed ap-plications to identify which licenses are preferred in different contexts.

E. Compliance and Legal Implication Analysis

The legal implications of license selection, including con-siderations for license compatibility, compliance obligations, and risk mitigation, were analyzed in the context of real-world software development scenarios.

The methodology employed in this study combines textual analysis of license documents with comparative frameworks and empirical observation of license usage patterns in the open-source ecosystem.

IV. Results and Discussion

A. Comparative License Analysis

Table I presents a comprehensive comparison of the three licenses across key dimensions. The analysis reveals signifi-cant differences in their approach to software freedom, patent protection, and commercial viability.

TABLE I
Comprehensive Comparison of GPL, MIT, and Apache Licenses

Feature	**GPL v3**	**MIT**	**Apache 2.0**
License Type	Copyleft	Permissive	Permissive
Source Code Access	Required	Required	Required
Derivative Works	Must be GPL	Any license	Any license
Commercial Use	Permitted	Permitted	Permitted
Modification	Permitted	Permitted	Permitted
Distribution	Must include license	Must include license	Must include license
Patent Grant	Implicit	None	Explicit
Warranty Disclaimer	Yes	Yes	Yes
Liability Limitation	Yes	Yes	Yes
License Length	Long	Very Short	Moderate
Complexity	High	Very Low	Moderate

B. GNU General Public License (GPL)

The GPL, particularly version 3, is grounded in the philoso-phy of software freedom and communal sharing. It grants users four fundamental freedoms: the freedom to run the software for any purpose, to study and modify the source code, to redistribute copies, and to distribute modified versions [5].

The distinguishing characteristic of GPL is its copyleft provision. Section 5 of GPLv3 mandates that any software incorporating GPL-licensed code must itself be licensed under GPL terms. This ”viral” nature of GPL ensures that software freedom is preserved throughout the entire software ecosys-tem. If a developer incorporates a GPL library into their pro-prietary application, the entire application must be distributed under GPL, effectively preventing proprietary derivative works [9].

Advantages of GPL:

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15589

- Ensures software remains free and open-source
- Prevents proprietary appropriation of open-source code
- Encourages community contribution and knowledge shar-ing
- Provides strong protection against commercial exploita-tion

Limitations of GPL:

- Restrictions on proprietary derivative works discourage some commercial developers
- Complex license text creates confusion and compliance challenges
- Incompatible with many permissive licenses
- May reduce adoption in enterprise and proprietary soft-ware environments

GPL is widely used in infrastructure software, embedded systems, server applications, and community-driven projects. Notable examples include the Linux kernel, GNU utilities,and numerous server-side applications. The license's effective-ness in maintaining software freedom has been demonstrated through widespread adoption in the Linux ecosystem, where GPL has played a crucial role in preventing corporate monop-olization of the operating system.

C. MIT License

The MIT License is one of the simplest and most permissive open-source licenses available. Comprising merely 171 words, it grants users unrestricted rights to use, copy, modify, merge, publish, distribute, and sublicense the software, subject only to the condition that the license text and copyright notice be included in any substantial copies [10].
The MIT License does not impose copyleft requirements, meaning developers can create proprietary derivative works without obligation to release source code. This maximum permissiveness has contributed to the MIT License's popu-larity, particularly in application development, libraries, and frameworks.
Advantages of MIT:

- Maximal user freedom with minimal restrictions
- Simplicity and ease of understanding
- Compatible with virtually all other licenses
- Encourages broad adoption and commercial use
- Minimal compliance burden

Limitations of MIT:

- No explicit patent protection
- No protection against proprietary appropriation
- Minimal legal recourse for patent infringement
- Potential for unattributed commercial use
- Weaker enforcement mechanism for software freedom The MIT License is extensively used in modern web

devel-
opment, JavaScript libraries, mobile applications, and frame-
works. Projects like Node.js, React, Ruby on Rails, and jQuery use MIT or similar permissive licenses. The low friction and high compatibility of the MIT License have made it the default choice for many contemporary open-source projects.

D. Apache License 2.0

The Apache License 2.0 represents a balanced approach, combining the permissiveness of MIT-style licenses with explicit patent protection. Drafted by the Apache Software Foundation, it includes provisions addressing modern software development concerns, particularly patent-related issues.
The Apache License 2.0 grants similar rights as MIT, permitting use, modification, and distribution of software. However, it includes explicit language regarding patent grants: contributors grant recipients an express patent grant for the contribution, and any modifications to the software similarly receive patent protection. Additionally, if a recipient initiates patent litigation against the licensor, their patent rights under the license are terminated [7].
Advantages of Apache 2.0:

- Explicit patent protection and indemnification
- Permissiveness with professional legal clarity
- Suitable for commercial and enterprise use
- Strong patent grant language protects contributors
- Clear termination clause for patent litigation
- More comprehensive than MIT while remaining permis-sive

Limitations of Apache 2.0:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15589

- More complex than MIT, requiring careful reading
- Longer legal text increases compliance burden
- Copyleft requirements when combining with GPL (in-compatibility issue)
- Patent indemnification clause may be misunderstood
- Adoption less widespread than MIT or GPL in certain sectors

The Apache License 2.0 has gained significant adoption in enterprise software, cloud computing platforms, and large-scale projects. Notable examples include the Apache HTTP Server, Hadoop, Spark, Kubernetes, and numerous Google-sponsored projects. Enterprises favor the Apache License due to its patent protection and clarity regarding patent rights.

E. License Compatibility and Integration

A critical practical consideration in open-source develop-ment is license compatibility. Integrating code from multi-ple sources requires compatibility between their respective licenses. Figure 1 illustrates the compatibility relationships be-tween the three licenses and common permissive alternatives.

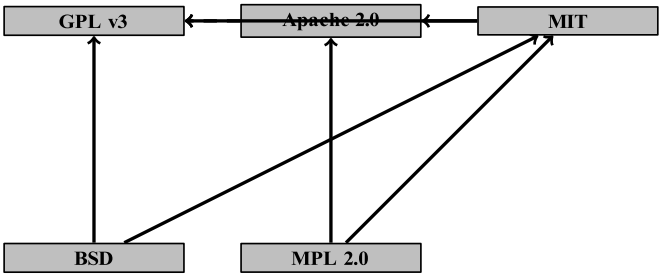
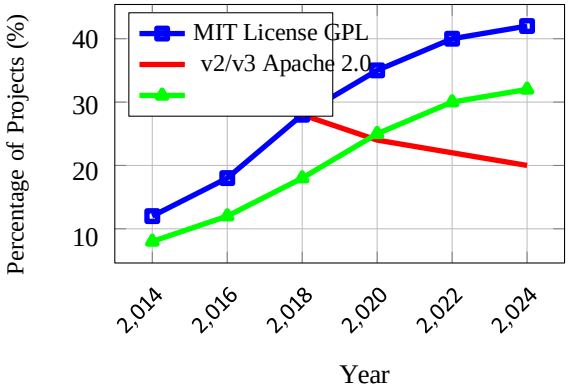


Fig. 1. License Compatibility Matrix: Green arrows indicate compatibility; red dashed arrows indicate incompatibility. One-way arrows denote directional compatibility; mutual compatibility is indicated by bidirectional paths. GPL and Apache 2.0 are generally incompatible due to patent termination clauses.

A significant finding is that GPL and Apache 2.0 are technically incompatible. While Apache 2.0 code can incor-porate GPL code (one-way compatibility), the reverse is not true. This incompatibility arises from the patent termination clause in Apache 2.0, which conflicts with GPL's patent grant provisions. Consequently, integrating Apache-licensed libraries into GPL projects requires careful consideration or explicit dual-licensing arrangements.

MIT and Apache 2.0 are mutually compatible, as are MIT and GPL (one-way). This compatibility has influenced license selection in projects that need to integrate code from diverse sources. Many projects use Apache 2.0 when they anticipate incorporating multiple library types, while MIT is chosen when maximum integration flexibility is desired.*Real-World Application Analysis*

Empirical analysis of license usage across major platforms reveals distinct patterns. Figure 2 depicts the adoption trends of the three major licenses over the past decade.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15589

Fig. 2. License Adoption Trends (2014-2024): Analysis of major GitHub repositories demonstrates increasing adoption of permissive licenses (MIT and Apache 2.0) and declining but stable adoption of GPL. The shift reflects enterprise preference for permissive licenses in client-facing applications.

GitHub Repository Analysis: Analysis of major repositories indicates that MIT License dominates the web development and JavaScript ecosystem, Apache 2.0 is prevalent in data processing and cloud platforms, and GPL remains strong in systems software and embedded applications [2].
Enterprise Adoption: Enterprise organizations show a pro-nounced preference for Apache 2.0 due to explicit patent protection, with secondary preference for MIT. GPL adoption in enterprises is significant but primarily in infrastructure and backend systems rather than client-facing applications.
Commercial Software Ecosystem: Companies developing commercial software frequently use permissive licenses (MIT or Apache 2.0) for their open-source components, as copyleft restrictions would conflict with proprietary licensing models.

F. Patent Considerations and Litigation Context

The evolution of software patent litigation has elevated the importance of explicit patent protection in open-source licenses. Prior to Apache 2.0, most open-source developers relied on implicit patent grants derived from copyright law. However, the explicit patent language in Apache 2.0 provides clearer legal standing [7].
GPL v3 attempted to address patent concerns through lan-guage regarding patent retaliation and defensive patent grants. However, Apache 2.0's approach, with its explicit patent grant and defensive termination clause, has proven more robust in addressing patent-related business concerns.

V. CONCLUSION

This comparative study has examined three major open-source licenses—GPL, MIT, and Apache 2.0—across multiple dimensions including permissions, restrictions, patent protec-tion, and practical adoption patterns. The analysis reveals

that each license embodies distinct philosophies and serves different stakeholder needs.
The GPL emphasizes software freedom and community benefit through its copyleft approach, ensuring that software remains open and free. This approach is highly effective for community-driven projects and prevents commercial appropri-ation. However, copyleft restrictions limit proprietary deriva-tive works and create integration challenges with permissive licenses.
The MIT License provides maximum flexibility with min-imal restrictions, making it ideal for developers seeking to promote broad adoption and integration. Its simplicity and near-universal compatibility have contributed to its popularity in contemporary software development. However, the absence of explicit patent protection may be a concern in patent-litigious environments.
The Apache License 2.0 balances permissiveness with ex-plicit patent protection, making it particularly suitable for en-terprise and commercial applications. Its comprehensive legal framework addresses modern software development concerns while maintaining relative simplicity compared to GPL.
License selection should be guided by project goals, target audience, anticipated use cases, and organizational constraints. GPL is appropriate for projects prioritizing software freedom and community benefit. MIT is suitable for projects seeking maximum adoption and integration flexibility. Apache 2.0 is recommended for projects anticipating enterprise adoption or patent-related concerns.
The choice of an appropriate open-source license is not merely a legal formality but a strategic decision with profound implications for project sustainability, community participa-tion, commercial viability, and long-term success. Developers, project maintainers, and organizations must carefully evaluate their specific context and objectives when selecting a licensing strategy.

VI. FUTURE SCOPE

Future research directions include:

- Empirical analysis of license compliance practices and failures in large-scale projects
- Longitudinal studies of how license choice influences project adoption and community participation
- Analysis of emerging licenses (e.g., AGPL, SSPL) and their role in addressing SaaS and cloud computing con-cerns
- Investigation of dual-licensing models and their effective-ness in balancing open-source and commercial interests

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15589

- Legal case studies of open-source license enforcement and dispute resolution
- Development of automated tools for license detection, compatibility analysis, and compliance verification
- Examination of license adoption patterns across emerging domains such as machine learning and data science
- Analysis of how geopolitical and jurisdictional factors influence license selection and enforcement

REFERENCES

[1] C. DiBona, S. Ockman, and M. Stone, *Open Sources: Voices from the Open Source Revolution*. O'Reilly Media, 1999.
[2] A. Mockus, R. T. Fielding, and J. D. Herbsleb, "Two case studies of open source software development: Apache and Mozilla," *IEEE Transactions on Software Engineering*, vol. 31, no. 7, pp. 551–574, July 2005.
[3] D. M. Wheeler, "Free-libre/open source software (FLOSS) license slide," Available: https://www.dwheeler.com/essays/floss-license-slide. html, Accessed: May 2024.
[4] T. Lemley and B. Kuhn, "Gpl-linked software: A guide to legal com-pliance in open source software," *Journal of Law and Technology*, vol. 10, pp. 45–67, 2013.
[5] R. M. Stallman, "Free software, free society: Selected essays of Richard M. Stallman," Free Software Foundation, 2002.
[6] E. S. Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly Media, 2001.
[7] A. Daffara and M. Fitzgerald, "The future of the open source move-ment," *Open Source Technology Handbook*, pp. 1–25, 2007.
[8] B. Fitzgerald, "The transformation of open source software," *MIS Quarterly*, vol. 30, no. 3, pp. 587–598, 2006.
[9] Free Software Foundation, "GNU General Public License v3.0," Avail-able: https://www.gnu.org/licenses/gpl-3.0.html, Accessed: May 2024.
[10] Massachusetts Institute of Technology, "The MIT License," Available: https://opensource.org/licenses/MIT, Accessed: May 2024.
[11] Apache Software Foundation, "Apache License 2.0," Available: https: //www.apache.org/licenses/LICENSE-2.0, Accessed: May 2024.
[12] S. A. Hissam and C. B. Weinstock, "An empirical study of open source license compliance," *IEEE Software*, vol. 28, no. 4, pp. 67–73, 2011.