THE LINUX FOUNDATION | Research

オープンソース ライセンス コンプライアンス

今後の課題

Ibrahim Haddad, Ph.D., Vice President,
Strategic Programs (AI), The Linux Foundation

序文 Jimmy Ahlberg, Director,
Open Source Policy, Ericsson

2024 年 1 月

In partnership with

LF AI & DATA

オープンソース ライセンス コンプライアンス：今後の課題

オープンソースソフトウェア（OSS）のライセンスコンプライアンスは、OSSを製品やサービスに組み込む際に、**著作権表示を遵守し、ライセンス義務を果たすことを要求します。**

OSS ライセンスのコンプライアンスを確保することは、複雑で入り組んだ作業になる可能性があります。
多様なライセンス、さまざまな条件、ソフトウェア開発のペースの速さなどを考慮する必要があるからです。

コンプライアンスのプロセスには、製品やサービスに組み込まれているすべてのOSSを特定し、適用されるすべてのライセンス義務を果たすための計画を策定することが含まれます。

OSSライセンスのコンプライアンスを効果的に管理するには、包括的な機能を備えた**高度なソフトウェア構成分析（SCA）ツール**が必要です。

組織は、コンプライアンス上の問題や問い合わせにどのように対処しているかを**ユーザーに可視化する**ことで、オープン、アカウンタビリティ、コラボレーションの文化を促進することができます。

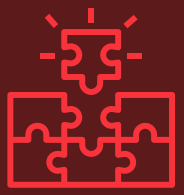
コンプライアンスを開発プロセスに統合することで、組織はコンプライアンス違反のリスクを低減すると同時に、健全な社内オープンソース ガバナンス文化を促進することができます。

組織は、**適切なツールを活用**し、社内のサポートを受けることで、OSSライセンスのコンプライアンスを効果的かつ大規模に管理し、コンプライアンス リスクを軽減することができます。

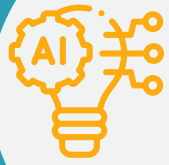
SCAツールは、正確で一貫性があり、複雑性に対応し、すべてのOSSを特定し、**OSSのライセンス状況を常に更新していなければなりません。**

OSSおよびライセンス コンプライアンスに関連するすべての活動の透明で包括的な監査証跡を維持しなければならないため、**監査可能性は組織にとって重要な課題**です。

SCAツールは、ソフトウェア開発ライフサイクルと**統合**し、オープンソース コンポーネントやライセンス要件についてコードを自動的にスキャンできるものでなければなりません。



人工知能ーによって生成されるコードは、開発者に提供されるポリシー オプションやガイダンスを用いて、組織が最初に対処できる**新たな課題を提示します。**



OSSの普及が進むにつれ、法的リスクや風評リスクを回避するために、**強固で自動化されたコンプライアンスプロセスを確立することが重要になっています。**

目次

序文

“私たちは歴史の真ん中の子供です。地球を探検するには遅く生まれすぎ、宇宙を探検するには早く生まれすぎました。”

— 匿名

今、物理学オープンソース ライセンス コンプライアンスで新たに発見されるものは何もない。残されているのは、より精密な測定だけである。

このセリフは、初めて聞いたときから心に残っています。この言葉は、オープンソースやオープンソース ライセンスのコンプライアンスにも関連しています。私は、すでにいくつかの偉大な決定的瞬間が起こった後にこの業界に入ってきた世代の人間です。私は、掲示板やアーカイブされたメーリングリストでの白熱した議論を後から読んだり、当時その場にいた人たちから「戦記」を聞いたりしたことしかありません。「探求」はすでに終わっていたのです。私がこの広大で、混乱に満ちた、複雑で、素晴らしいコミュニティの一員であった 10 年間、私は対面でもオンラインでも、多くの白熱した情熱的な議論に参加してきました。私は、彼らが行ってきた仕事だけでなく、彼らの知識と専門知識を分かち合おうとする姿勢に対しても、その探究者たちに計り知れない感謝の念を抱いています。

Ibrahim と初めて会ったのは、彼の電子書籍『Open Source Compliance in the Enterprise』が出版されたときでした。私にとって、それを読むことは真の啓示でした。オープンソース ライセンスのコンプライアンスとは何か、そしてどのように組織的に取り組むべきかについて、私や他の多くの人たちの考えを形成する助けとなりました。これは、この分野における Ibrahim の深い専門知識と、その知識を他の人々と共有することで、私たち全員が一丸となって向上していけるという彼の意欲と能力の証です。Ibrahim は間違いなく、オープンソース コンプライアンスの分野を定義するようになった人物の一人です。探求者であるだけでなく、” 生まれるのが遅すぎた “私たちの多くのためにこの分野を導いてきました。

この序文の引用の後半に触れる前に、ケルビン卿の言葉を引用したいです：

1897 年の物理学においてその声明が間違っていたのと同様に、今日のオープンソース ライセンス コンプライアンスにおいても同じくらい間違っています。我々は巨人の肩に乗っており、それによってより遠くを見ることができるようになっています。Ibrahim がこの思慮深い論文で指摘しているように、我々の業界と技術は進化しました。現在、私たちは以前よりも遠くを見ることができ、新しい課題に取り組む必要があります。特に、AI によって生成されたコードは、多くの新しい課題を我々に投げかけています。

この論文を読むときには、新しい課題に対する警告として受け取らないでください。むしろ、未来へのガイドとして、投げかけられた挑戦として、そして私たち全員、初期の探検者たちや後から来た者たちが取り組むべき課題として読んでください。私たちは一緒にこの新しい領域を探求することができます。おそらく、私たちはオープンソース ライセンス コンプライアンスにおける歴史の真ん中の子供たちかもしれません。もしそうであるならば、Ibrahim が指摘する課題は非常に興味深い立場にあることを示唆しています。この論文を読むことで、あなたもこれら新しい課題の興奮を共有し、それらに取り組む挑戦を受け入れ、今後数年間に直面するオープンソース ライセンス コンプライアンスにおけるこれらおよび他の予期しない課題に対処しようとすることを願っています。

JIMMY AHLBERG
DIRECTOR, OPEN SOURCE POLICY, ERICSSON

概要

オープンソースソフトウェア（OSS）は、現代のソフトウェア開発における重要な要素として台頭し、多くの組織がそのコスト削減、柔軟性、イノベーションのメリットを活用しています。しかし、OSS の採用には、ライセンス要件を遵守するという責任が伴います。オープンソース ライセンスの不遵守は、組織にとって重大な法的、財政的、および風評的結果をもたらす可能性があり、オープンソース エコシステム全体に影響を及ぼします。

オープンソース ライセンス コンプライアンスは、ライセンスの多様性、さまざまな条件、ソフトウェア開発の急速な進展のために、困難で複雑になることがあります。オープンソース ライセンスの状況は絶えず進化しており、新しいライセンスが頻繁にリリースされます。新しいライセンスもあれば、既存のライセンスの改訂版もあります。

オープンソース ライセンスのコンプライアンスを確保するには、ライセンス要件を深く理解し、コンプライアンス プログラムを明確に定義し、プロセスを管理するための効果的なツールと戦略が必要です。

このような状況において、組織は、適用されるすべてのオープンソース ライセンスのコンプライアンスを維持しながら OSS を使用するために、ベストプラクティスを採用し、効果的なコンプライアンス戦略を実施しなければなりません。このプロセスには、オープンソース ライセンスの意味を理解し、ソフトウェア開発プロジェクト内のオープンソース コンポーネントを特定し、すべてのライセンス義務を果たすことが含まれます。

本稿では、オープンソース ライセンス コンプライアンスにおける最重要課題を掘り下げます。これらの課題の複雑さを理解し、効果的なコンプライアンス戦略を実施することで、組織は関連するリスクを軽減し、OSS プロジェクトへの参加からイノベーションと成長を促進する利益を得ることができます。今後のレポートでは、これらの課題に対処するための推奨プラクティスを提供することを目指します。

導入

伝統的に、組織は独自のソフトウェアや、交渉によってライセンス契約を結んだ第三者の商用ソフトウェアを使用して、自社のソフトウェア プラットフォームやスタックを構築していました。このアプローチにより、ソフトウェア スタック内の各コンポーネントの提供者を簡単に特定することができ、組織はソフトウェア提供者やベンダーとのライセンスおよび契約交渉を通じて潜在的なリスクを軽減しています。時が経つにつれて、組織は様々な利点から OSS を自社のプラットフォームやソフトウェア スタックに組み込むようになりました。オープンソース コンポーネントは魅力的な機能を提供し、分散型開発による迅速な市場投入を可能にし、ソースコードのカスタマイズを許容しました。この新しい実践により、新しいマルチソース開発モデルが登場しました。

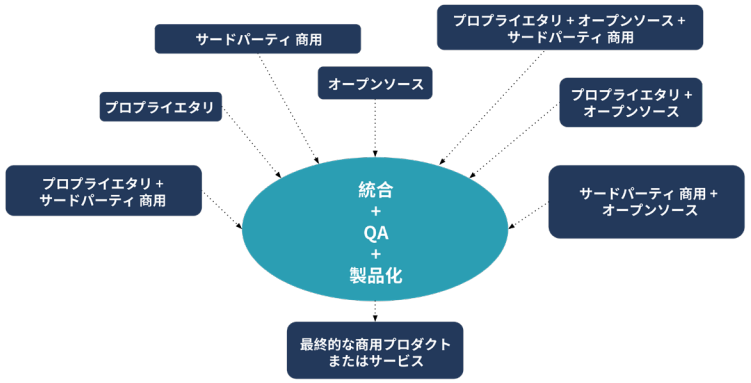


図 1: マルチソース開発モデル

図 2: オープンソース開発モデルが提供する利点

このモデルのもとでは、製品は以下のどのような組み合わせも可能です:

- オープンソース コードを含む可能性のある、自社開発の独自コード
- オープンソース コンポーネントと統合されているが、上流のオープンソース プロジェクトには貢献されていない独自コード
- 商用ライセンスで受領したサードパーティの商用コードで、オープンソース コードを含む場合もある。
- オープンソース コミュニティによって開発され、オープンソース ライセンスの下で会社が受け取ったオープンソース コード
- オープンソースでもプロプライエタリでもないが、未知の、あるいはよく理解されていないカスタム ライセンスの下で公開されているコード

この新しい開発モデルには多くの利点がある一方で、組織にとっては新たな課題もあります。関連するオープンソース ライセンスを確実に遵守しなければなりませんが、これはスタックに組み込む OSS によって大きく異なります。オープンソース ライセンスの義務を遵守しないと、場合によっては法的リスクや財務的リスクにつながる可能性があります。

図 1 は、マルチソース開発モデルと、入力ソースコードのさまざまな組み合わせを示しています。このモデルでは、ソフトウェア コンポーネントは、さまざまなソースに由来するソースコードで構成され、異なるライセンスの下でライセンスされることができます。たとえば、ソフトウェア コンポーネント A には、プロプライエタリなソースコードとサードパーティのソースコードを含めることができます。対照的に、ソフトウェア コンポーネント B は、プロプライエタリなソース コードと、1 つまたは複数のオープンソース プロジェクトのソースコードで構成することができます。

プロプライエタリなソフトウェアスタックに組み込まれる OSS の量が増えるにつれて、オープンソース エコシステムに参入し、OSS を自社の製品やサービスに統合し始めた多くの組織にとって、ビジネス環境はより複雑で不慣れなものになりました。この複雑さは、従来の独自ソフトウェア開発では直面しなかった課題や潜在的なリスクを引き起こしました。組織は、さまざまなオープンソース コンポーネントのライセンス要件を把握し、これらのライセンスに準拠することで、法的および財務的リスクを軽減することができます。

図 3 は、さまざまなプラットフォームやソフトウェア スタックのレベルで、組織がどのように OSS を採用しているかを示しています。プロプライエタリな開発モデルとマルチソース開発モデルの主な違いの 1 つは、OSS のライセンスが交渉制ではないことです。プロプライエタリ ソフトウェアとは異なり、ソフトウェア プロバイダ（通常はオープンソースの開発者やプロジェクト）との契約はありません。代わりに、オープンソース プロジェクトを開始する個人がライセンスを選択し、プロジェクトが一定の規模に達すると、ライセンスを変更することは事実上不可能になります。

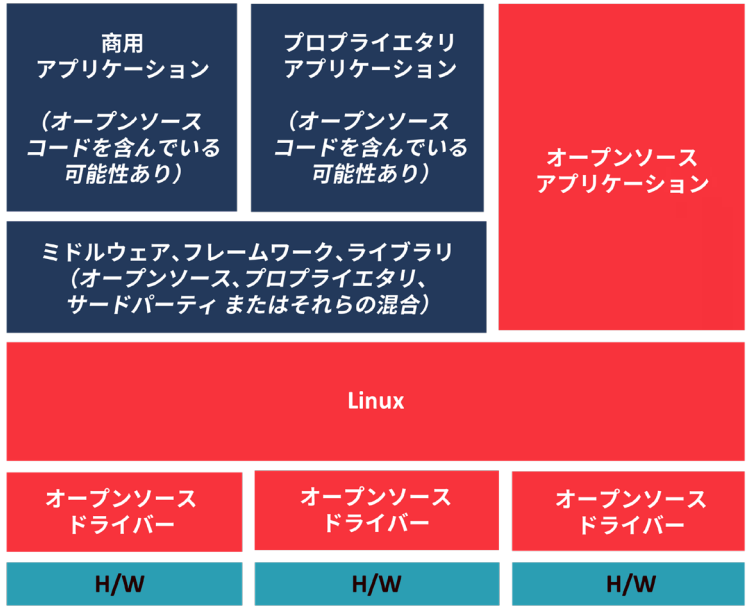


図 3: 最新のソフトウェア・プラットフォームの簡略化されたアーキテクチャ図 - オープンソースはあらゆるビルディングブロックを増殖させた

マルチソース開発モデルを使用する場合、組織は、著作権者である何千、何万ものライセンサーやコントリビューターから提供される可能性のある、何十種類ものライセンスやライセンスの組み合わせの意味を理解しなければなりません。その結果、組織は、企業間のライセンスや契約の交渉ではなく、強固なコンプライアンス プログラムと慎重なエンジニアリングの実践を通じて、コンプライアンス リスクを管理しなければなりません。

この新しい開発モデルは、オープンソースライセンス、コンプライアンス義務、開発ライフサイクルを通じて使用される OSS を追跡・管理するための効果的なプロセスを深く理解するよう組織に迫りました。

オープンソース コンプライアンスへの参加

オープンソースのコンプライアンスは、OSS を採用し、製品やサービスに組み込む上で極めて重要な要素です。オープンソースコミュニティとのコラボレーションは大きなメリットをもたらしますが、同時に不可欠な責任も伴います。

オープンソースライセンスを遵守するには、著作権表示を遵守し、OSS の使用から生じるライセンス義務を満たす必要があります。効果的なオープンソース コンプライアンス プログラムは、意図しない開示やその他の悪影響から企業の知的財産やサードパーティ サプライヤを保護しながら、オープンソース ライセンスの条項の遵守を保証する必要があります。

包括的なオープンソース コンプライアンス プログラムを導入することで、組織は OSS の使用に関連するリスクを軽減し、オープンソース コミュニティとのコラボレーションのメリットを十分に享受することができます。

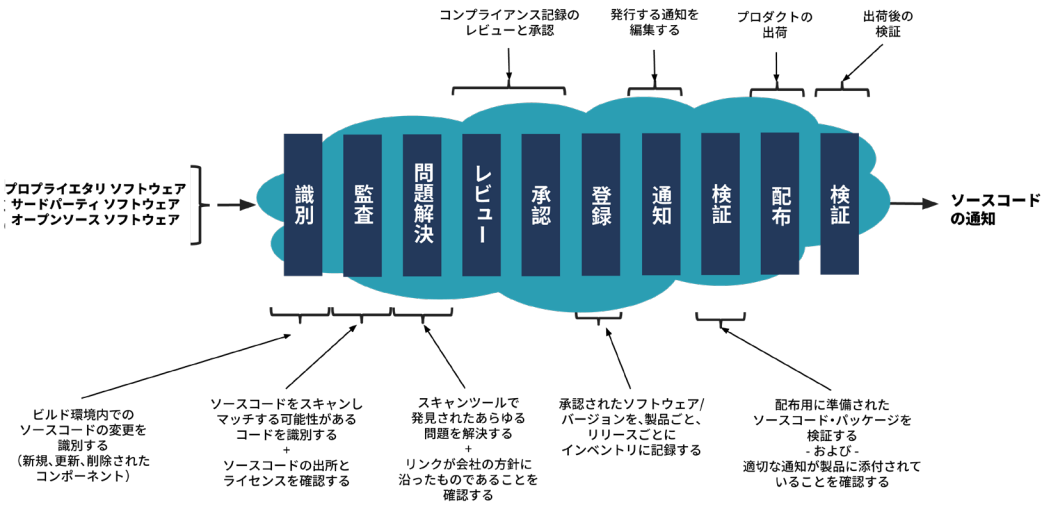


図 4: オープンソース コンプライアンス プロセスの概要

オープンソース コンプライアンス プロセスの実装は、コンプライアンスを適合させる必要がある基本的な製品開発プロセス、コードベースの規模と性質、製品の数、外部から供給されるコードの量、企業の規模と組織構造など、いくつかの要因に基づいて企業によって異なる可能性があります。しかし、核となるコンプライアンス要素は通常変わりません：

1. コードベース内のオープンソースを特定する
2. その使用を審査し承認する
3. オープンソースライセンスの義務を満たす

コンプライアンス デューデリジェンス プロセスでは、外部配布を意図した製品で使用されるすべての OSS と、それに付随するライセンス義務を満たすための計画を特定します。

図 4 は、エンドツーエンドのコンプライアンス プロセスの例をハイレベルで概観したもので、OSS を含むコンポーネントが、外部配布を意図した製品での使用を承認されるまでに通過するさまざまなコンプライアンス ステップまたはフェーズを示しています。コンプライアンス プロセスを組織化する他の方法でも、コンプライアンスを確保するという同じ目標を達成できる可能性は十分にあります。

効果的なコンプライアンス プロセスには、組織ごとに異なるいくつかのステップが含まれ、組織内のオープンソースへの取り組みをどのように構成し、管理するかによって異なります：

- 組織に入るすべてのソフトウェアの識別
- すべてのソースコードの監査
- 監査によって発見された問題の解決
- 適切なレビューの完了
- OSS の使用承認を受ける

- 使用するすべての OSS をソフトウェア インベントリに登録
- 製品に使用されている OSS を反映させるために、エンドユーザ向け文書を更新
- 配布前のすべてのステップの検証の実施
- オープンソース パッケージのコードを配布
- 最終検証の実施

ソースコンプライアンスプロセスの詳細については、Linux Foundation が発行した電子書籍『**Open Source Compliance in the Enterprise (2nd Edition)**』をお勧めします。この電子書籍では、組織が製品やサービスにオープンソースコードを採用して使用し、オープンソース コミュニティに合法的かつ責任ある方法で参加するためのベストプラクティスが概説されています。

図 5: 次の 10 年に向けたコンプライアンスの中核的課題

新たな時代、新たな課題

近年、オープンソース コンプライアンスの状況は大きく変化しており、ソフトウェア開発において OSS の利用がますます普及するにつれ、より複雑な課題が浮上しています。過去 20 年間、私たちは、オープンソース コンポーネントの特定と帰属、ライセンス条項の遵守の確保、OSS の消費と貢献を管理するプロセスの確立といった課題に取り組んできました。しかし、OSS の普及、クラウド サービスの利用増加、新しいライセンス制度、OSS を使用して学習させた AI 生成コードなどにより、今日の課題はより複雑になっています。

OSS の利用が拡大し続ける中、企業はソフトウェア スタック全体のコンプライアンスを、これまでに経験したことのない大規模な規模で管理するという課題に直面しています。そのため、企業がソフトウェアスタック全体のコンプライアンスを効果的に管理できるよう、新たな課題に対処するための高度なコンプライアンス戦略とツールが求められています。

以下のセクションでは、8 つの重要な課題を探ります：

- アクセシビリティ
- 透明性
- 高度な機能
- スケーラビリティ
- スピード
- 正確性
- 監査可能性
- AI 生成コード

アクセシビリティ

アクセシビリティの確保は、オープンソース ライセンス コンプライアンスにおける重要な課題です。数多くのオープンソース ライセンスが使用されているため、組織はこれらのライセンスの条項を特定し、遵守するために必要なツールとリソースを持たなければなりません。しかし、すべての開発者がオープンソース ライセンシングに関して同じ専門知識を持っているわけではないことを考慮することは極めて重要です。したがって、オープンソースのコンプライアンス活動をサポートするために使用されるツールは、経験レベルに関係なく、すべての開発者が簡単にアクセスできるものでなければなりません。コンプライアンスの確保は、OSS の取り込みとオープンソース プロジェクトへの貢献が行われる開発者から始まります。

導入されるコンプライアンス ツールは、ユーザーフレンドリーなインターフェース、明確なドキュメント、およびアクセス可能なトレーニング リソースを備え、ユーザーがツールをナビゲートするのを支援する必要があります。開発者がコンプライアンス プロセスに直接関与することで、ライセンス要件を理解し、ライセンスの条件に準拠することができます。さらに、コンプライアンス ツールはソフトウェア開発プロセスに簡単に統合されるべきであり、開発チームが使用する他のツールともシームレスに統合されるべきです。このような統合により、コンプライアンスの取り組みが開発プロセスの一部となり、エンジニアリングおよび開発の負担と見なされる別個の作業ではなくなります。

コンプライアンスにアクセスしやすくし、開発プロセスに統合することで、組織はコンプライアンス違反のリスクを減らすと同時に、優れたオープンソース ガバナンスの文化を促進することができます。

透明性

透明性は、OSS を採用する組織とオープンソース プロジェクト コミュニティとの間の信頼と信用を育むのに役立つ、オープンソース ライセンス コンプライアンスの重要な側面です。組織は、製品やサービスに組み込まれているすべての OSS を特定し、適用されるすべてのオープンソース ライセンスに準拠する仕組みの確立を支援する、信頼性と透明性の高いコンプライアンス ツールにアクセスできなければなりません。透明性を達成するために、組織は、個人が潜在的なコンプライアンス上の懸念や発見された問題を報告できるようにする問題追跡ツールと、これらの問題や問い合わせを処理するための透明なプロセスを持つべきです。ユーザーとの定期的なコミュニケーションは、懸念事項の調査の進捗状況を知らせる上で極めて重要です。

ソフトウェア部品表（SBOM）の提供は、製品やサービスのユーザーに透明性を提供するための一般的な方法となりつつあります。これらの SBOM は、コンポーネントの使用状況、オープンソース ライセンス、および使用中のオープンソース コンポーネントの潜在的な脆弱性の状況について可視性を提供します。

オープンソース ライセンス コンプライアンスにおける信頼と信用を構築するためには、透明性が不可欠です。組織は、コンプライアンスに関する問題や問い合わせにどのように対処しているかをユーザに可視化することで、オープン性、説明責任、およびコラボレーションの文化を促進し、責任あるオープンソース開発の実践へのコミットメントを示すことができます。

高度な機能セット

オープンソース ライセンス コンプライアンスは、特に今日の複雑化するソフトウェア開発の状況において、困難な場合があります。OSS の普及が進むにつれ、法的リスクや風評リスクを回避するためには、強固なコンプライアンス プロセスを導入することが不可欠です。高度な機能を備えたコンプライアンス ツールは、複雑な開発を補完し、包括的なコンプライアンスを確保する必要があります。コンプライアンス ツールには、自動ライセンス識別機能などの高度な機能を持たせ、開発者がオープンソース コンポーネントとその関連ライセンスを識別できるようにする必要があります。さらに、ライセンス条項と義務を遵守するようユーザーをガイドするライセンス義務管理機能を備えている必要があります。また、組織の内部ポリシーに基づいて潜在的なコンプライアンス リスクを特定し、その軽減を推奨するリスク分析機能も備えている必要があります。さらに、コンプライアンス ツールは、ソースコード リポジトリ、継続的インテグレーションおよびデリバリー システム、プロジェクト管理ツールなど、開発チームが使用する他のツールとの統合をサポートする必要があります。このような統合により、コンプライアンス ツールを既存の開発ワークフローにシームレスに接続し、すべての開発チーム メンバーが使用できるようになります。

重要なのは、ツールが組織の製品やサービスを構築し実行するために使用されるすべてのオープンソース アーティファクトを発見し管理することです。現在の多くのソフトウェア コンポーネント分析(SCA)ツールは、パッケージ マネージャーによって管理されているコンポーネントのみを対象としています。この種のスキャンでは、ソースコード ファイルやパッケージ管理されていないバイナリー ライブラリ、ソースコードの切り貼りなど、重要なアーティファクトは調査されません。さらに、これらのツールの多くは、コンポーネントとその推移依存関係のトップレベルのライセンスのみを調査します。完全なオープンソース コンプライアンスには、イントラ パッケージのスキャンが必要であり、または事前にスキャンされた SBOM のようなサードパーティ コンポーネントのオープンソース開示を受け渡せることが必要です。

さらに、ツールはコンテナ、クラウドネイティブ アプリケーション、サーバーレス アーキテクチャなど、さまざまなソフトウェア開発環境をサポートする必要があります。コンテナを例にとってみましょう。コンテナは、開発者がソフトウェアを軽量でポータブルな形式でパッケージ化し、デプロイと管理を容易にします。しかし、コンテナは、コンテナ イメージのライセンス管理や、異なるコンテナ レジストリ間でのコンプライアンスの確保など、オープンソースのコンプライアンスに関する新たな課題ももたらします。高度なコンテナ コンプライアンス機能を備えたコンプライアンス ツールは、コンテナ イメージがオープンソース ライセンス要件に準拠していることを確認し、コンプライアンス違反のリスクを最小限に抑えるのに役立ちます。

オープンソースのライセンス コンプライアンス ツールの評価については、Linux Foundation が発行した「**An Open Guide To Evaluating Software Composition Analysis Tools**」をお勧めします。

スケーラビリティ

大規模なオープンソース ライセンス コンプライアンスを確保する場合、スケーラビリティは非常に重要な要素です。組織が OSS に依存し続けるにつれて、コードの量とユーザーの数はすぐに圧倒的なものになる可能性があります。そのため、製品やサービスに OSS が広く採用され、その規模や複雑さに対応できるコンプライアンス インフラストラクチャーとそれをサポートするツールが不可欠です。

しかし、スケーラビリティとは、ツールの技術的な能力だけではありません。コンプライアンス プロセスを大規模に管理するための専任のリソースや担当者など、組織的なサポートも必要です。これには、コンプライアンス専門家、トレーニング プログラム、およびコンプライアンスがソフトウェア開発プロセスに不可欠であることを保証するガバナンス ポリシーが含まれます。このような内部サービスを提供するために、オープンソース プログラム オフィス（Open Source Program Office; OSPO）を設立する組織も一般的になってきています。

大規模なオープンソース ライセンスのコンプライアンスを確保するには、スケーラブルなアーキテクチャ、最適化されたパフォーマンス、および他のツールとの統合をサポートするコンプライアンス ツールが必要です。また、専用のリソースや人員などの組織的なサポートも必要です。組織によっては、スタッフの増員やベースラインのコンプライアンス監査のために、外部サービスの利用を決定する場合もあります。

スピード

すべての外部からのオープンソース コードのコンプライアンスを確保する速度は、組織が内部の開発ペースと製品やサービスに組み込まれる OSS の量に遅れることなく追いつかなければならない重要な課題です。どんな組織でも、オープンソースライセンス コンプライアンス インフラは、開発プロセスを遅くすることなく迅速に結果を提供するべきです。組織は、軽量なコンプライアンス ポリシーとプロセスを実装し、コードを高速でスキャンしオープンソースのコンポーネントとライセンス要件を迅速かつ正確に特定するツールを自動化することで、この困難な目標に対処できます。さらに、コンプライアンス ツールは、他のツールがライセンス情報をクエリするための API を提供し、開発者がコードをスキャンして迅速に結果を受け取るためのコマンドライン インターフェースも提供すべきです。組織は、速度とより完全な分析とコンプライアンスのためのトレードオフとリスクを認識するべきです。

もう一つの重要な速度の側面は、コンプライアンス プロセスを完全に自動化する能力です。コンプライアンス ツールは、ソフトウェア開発ライフサイクル（SDLC）に統合し、自動的にオープンソース コードとライセンス要件をスキャンできる必要があります。このような統合により、専任のコンプライアンス チームの作業量が減少し、コンプライアンス プロセスの一貫性と正確性が確保され、開発プロセスが加速されます。

正確性

オープンソース ライセンス コンプライアンスにおいて正確さは重要です。これにより、組織がオープンソース コンポーネントのライセンス要件に準拠することが保証されます。コンプライアンスツールは、オープンソース コンポーネントを正確に特定し、ライセンスの義務を検出し、リスク分析を提供できる必要があります。また、同じコンポーネントに複数のライセンスが適用される場合や、コンポーネント間にライセンスの依存関係がある場合など、複雑なシナリオを処理できる能力も求められます。

さらに、コンプライアンス ツールは、大規模またはレガシー コードベースを取り扱う際でも信頼性のある結果を提供する必要があります。さまざまなプログラミング言語やフレームワークを扱うことができ、コードベースやライセンス要件の変更に柔軟に対応できる能力を持つべきです。

オープンソース コンプライアンスは正確なオープンソース ライセンスの検出、著作権管理、およびライセンス義務管理に大きく依存していますので、強力なライセンス検出と管理機能、およびデータを持つツールを選択することが重要です。多くのツールが、コード内に実際に存在するライセンス テキストではなく「最良の推測」として表示し、正確なライセンスの開示を作成する能力を欠いていることが一般的です。修正されたオープンソース ライセンスを発見する能力を持つことは重要なコンプライアンス機能です。なぜなら、これらの修正によってライセンスに求められる義務が大幅に変わることがあるからです。

さらに、オープンソース ライセンス コンプライアンスの品質は、オープンソースコミュニティ内でも大きく異なります。コンプライアンス ツールは、発見されたライセンスとオープンソース プロジェクトが開示するトップレベルのライセンス、またはリポジトリ マネージャのメタデータで見られるライセンスとの不一致を警告できる必要があります。

コンプライアンス ツールは定期的に更新されるべきです。これにより、オープンソース ライセンスの変更や新しいオープンソース コンポーネントに追随し、組織は正確なコンプライアンス ツールを用いることでコンプライアンス リスクを最小限に抑え、オープンソース ライセンスの義務を果たすことができます。

監査可能性

オープンソース ライセンス コンプライアンスにおいて監査可能性は重要です。なぜなら、組織はオープンソース ライセンス コンプライアンスに関連するすべての活動の明確な監査トレイルを提供する必要があるからです。コンプライアンス インフラは、監査可能性の機能を提供すべきであり、つまり組織が変更を追跡し、それを行った人物とその時期を特定できる必要があります。また、導入されたツールは、コンプライアンス活動の履歴を提供し、標準プロセスからの例外や逸脱を含む情報を提供できるべきです。この側面は法的コンプライアンス、リスク管理、および顧客やオープンソース コミュニティとの信頼構築にとって重要です。さらに、使用するツールは、オープンソース ライセンス要件へのコンプライアンスを示すために使用できるレポートを生成できる必要があります。

人工知能ー生成コード[1]

AI の最近の進展により、AI によって生成されるコードがソフトウェアの作成プロセスを自動化することで、開発者の新しい波を推進しています。人工知能ツールは、開発者が提供した高レベルの指示に基づいて、関数、クラス、アルゴリズム、さらにはプログラム全体の生成をサポートする能力を持っています。このようなツールは、迅速なプロトタイピングや実験に非常に有益であり、アイデアやコンセプトをテストするための機能的なコード スニペットを素早く生成することができます。AI によって生成されるコードは、有望な面を示しており、経験の浅い開発者のソフトウェア開発におけるスキルギャップを埋めるのに役立っていますが、それ自体の課題も抱えています。AI によって生成されたコードのライセンス コンプライアンスを確保することは、以下の点で複雑な課題となります：AI システムが学習したコードのライセンス、AI システムがその出力で全体または一部を再現しているコードのライセンス、その出力がオープンソース コードの文言そのままを含んでいるかどうか、そしてそのコードが派生成果物と見なされる可能性があるかどうかなどです。

本稿では、AI によって生成されたコードを使用し、それをオープンソース コンポーネントや組織内で開発されるソフトウェアに組み込む際に対処すべき具体的な課題を提示したいと考えています：

1. **著作権に関する課題:** 生成 AI ツールは、訓練に使用された資料（この場合、ソースコード）の一部を再現しますが、その中には著作権の対象となる可能性があるものも含まれています。
2. **ライセンス互換性に関する課題：**生成 AI モデルが公開されている OSS で訓練された場合、AI の出力で再現された既存の OSS コードに適用される OSS ライセンスが、適用されるプロジェクト ライセンスと互換性がない場合があります。
3. **ライセンスコンプライアンスに関する課題:**多くの AI ツールは現在、プロンプトに対する出力として提供するソースコードの出所やライセンスについての情報を提供していません。したがって、これらの AI システムやモデルが訓練される際に許可された（または他の方法で互換性のある）OSS コードのみを使用したとしても、ライセンス コンプライアンスの課題が依然として存在します。つまり、AI の出力に再現されたソースコードのライセンス条件に適合するために、著作権者やライセンス条件を知らないまま、ダウンストリームの利用者がライセンスに関する通知や帰属要件をどのように満たすことができるでしょうか。
4. **ソフトウェア資産管理（SBOM）に関する課題：**AI システムが提供するソースコードの出所やライセンスに関する情報が欠落している場合、ダウンストリームの利用者は正確な SBOM（ソフトウェア部品表）を作成することができません。この状況は、セキュリティの脆弱性を追跡できないという具体的な懸念を引き起こします。

しかしながら、これらの課題のすべてがオープンソースにとって新しいものではありません。生成 AI がなくても、貢献者が許可されていない資料をコピーしたり、互換性のないライセンスであるソースコードをオープンソース プロジェクトにコントリビュートするリスクが存在します。ただし、生成 AI は個々の開発者が責任を持ってサードパーティのコードにコントリビュートする孤立した事例とは異なり、より広範かつ体系的な方法でこのリスクを提示する可能性があります。

1 Original work by Joanna Lee, Vice President of Strategic Programs (Legal), Linux Foundation. Please refer to the Acknowledgment section.

これらの懸念や課題にどのように対処すればよいのでしょうか?

本稿の目標はオープンソース ライセンス コンプライアンスの課題を提示することに焦点を当てていますが、生成 AI は非常に注目されるトピックであり、上記で言及された他の課題よりも議論をさらに拡大します。

AI 生成コードの使用によるライセンス コンプライアンス リスクを緩和するために、ポリシーやガイドラインが使用されることがあります。組織が AI 生成ソースコードを考慮する際に採用できるいくつかのポリシーの選択肢があります。

シナリオ 1：保守的

このシナリオでは、組織は保守的なアプローチを取り、開発者に対して AI ツールを使用してコード生成を行わないよう助言するでしょう。

シナリオ 2：選択的

a. **使用によって:**このシナリオでは、組織は寛容なアプローチを取り、特定のコンテキスト（例：デバッグ）に基づいて一部の使用を条件付きで許可するかもしれませんが、新機能を実装するためのコード生成など、他の使用やコンテキストは受け入れないでしょう。

b. **ツールによって：**このシナリオでは、組織はより寛容なアプローチを取り、AI システムやツールがオープンソース プロジェクトからのソースコードを出力する際に通知と帰属情報を提供する場合に限り、開発者がそれらを使用し、採用することを条件付きで許可するでしょう。このような場合、組織はコンプライアンスとライセンス互換性を確保しながら、出力に含まれるオープンソース コードの出典情報を提供しない AI ツールを禁止することができます。

シナリオ 3：開発者を鼓舞する

このシナリオでは、組織は開発者に AI ツールを使用してコード生成を許可する一方で、そのようなツールの使用に関する推奨事項やガイドラインを提供するかもしれません。

この問題に対処する一つの方法は、AI の訓練データやアルゴリズムに使用されるオープンソース コードのライセンスを注意深くレビューし、その結果生成されるコードがそれらの条件に適切にライセンスされていることを確認することです。また、AI の訓練や生成に使用されるすべてのオープンソース コードの出所について詳細な記録を保持することも重要です。これにより、生成されたコードの出典を追跡し、適用されるオープンソース ライセンスに準拠していることを確認できます。

まとめ

オープンソース ライセンス コンプライアンスは挑戦を伴いますが、組織は適切なツールと戦略を用いてこれらの課題を克服することができます。オープンソース ライセンス コンプライアンスを支援するツールは、すべての開発者が簡単にアクセスでき、オープンな問題について透明性を提供し、高度な機能セットを持ち、拡張可能で、迅速な結果を提供し、正確で監査可能である必要があります。適切なツールの使用に加えて、組織はオープンソース ライセンス コンプライアンスのための明確で軽量かつ簡潔なポリシーとプロセスを持ち、開発者向けのトレーニングとサポートを提供し、定期的にコンプライアンスの実践を見直し更新するべきです。これにより、組織はオープンソース ライセンス要件に準拠し、オープンソース コミュニティとの信頼関係を築くことができます。

将来数年間のライセンス コンプライアンスの領域では、本論文で議論された課題だけでなく、途中で明らかになる他の課題に対処していくでしょう。Linux Foundation は、オープンソース エコシステムにおけるさまざまな課題に取り組むために組織を連携させるいくつかの取り組みを主催しています。私たちのイニシアチブを通じて、オープンソース ライセンス コンプライアンスの状況を進展させるために、皆さんが私たちと協力して取り組むことをお勧めします。

フィードバック

著者は予め何か間違いがあった場合にお詫び申し上げ、修正や改善の提案を歓迎します。

謝辞

著者はこの論文の作成において、貴重な貢献をしてくれた Linux Foundation の研究スタッフとリーダーシップに感謝の意を表します。

特に、**Jimmy Ahlberg** 氏（Ericsson のオープンソースポリシー部門のディレクター）には、序文の寄稿と Open Source Summit EU での議論に対して特別な感謝を申し上げます。

また、**Jeffrey Luszcz** 氏には、論文のレビューや提案、編集作業における重要な貢献に対して心から感謝します。彼の提案と編集により、議論の内容が大幅に改善されました。

さらに、Linux Foundation の戦略プログラム（法務）バイス プレジデントである **Joanna Lee** 氏には、オープンソースコンプライアンスに関連する生成 AI の研究に対する大きな感謝を申し上げます。彼女の研究成果は、「人工知能ー生成コード」と題したセクションで本稿にまとめられています。

改めて、皆様に心から感謝いたします。

本訳文について

この日本語文書は、**Open Source License Compliance** の参考訳として、The Linux Foundation Japan が便宜上提供するものです。英語版と翻訳版の間で齟齬または矛盾がある場合（翻訳版の提供の遅滞による場合を含むがこれに限らない）、英語版が優先されます。

この日本語文書を引用する際には、下記の一文を記載してください。

引用：Open Source License Compliance 参考訳（The Linux Foundation Japan 提供）

翻訳協力：小笠原徳彦

Linux Foundation の参考資料

活動

1. **OpenChain:** OpenChain プロジェクトは、Linux Foundation がホストするコミュニティの取り組みであり、オープンソース ライセンス コンプライアンスのための標準とベストプラクティスを開発しています。
2. **SPDX**: Software Package Data Exchange（SPDX）は、ソフトウェア パッケージの内容とそれに適用されるライセンスを識別するための標準です。SPDX はオープンソース ライセンス コンプライアンスの文書化を標準化し、ビジネスがコンプライアンス義務をより効率的に管理するのに役立ちます。
3. **Open Compliance Program**: Linux Foundation の Open Compliance Programは、オープンソース コンプライアンスの実践を改善したい企業向けにリソースとツールを提供しています。このプログラムには、コンプライアンス義務の管理に役立つベスト プラクティスやトレーニング資料が含まれています。
4. **TODO Group**: TODO Group は、OSS 開発のためのベストプラクティス、ツール、プログラムを確立するために協力する組織のコミュニティです。このグループは、オープンソース プログラム管理、ガバナンス、コンプライアンスなどの問題に焦点を当て、ガイドやトレーニングなどのリソースを提供して、組織がこれらのトピックをうまくナビゲートできるよう支援しています。

トレーニング

- **Open Source Licensing Basics for Developers**
- **Implementing Open Source License Compliance Management**
- **Introduction to Open Source License Compliance**
- **Generating a Software’s Bill of Materials**

イベント

- **Open Compliance Summit**
- **Linux Foundation Legal Summit**

E-Books and Papers

1. **Open Source Guides for the Enterprise**
2. **An Open Guide To Evaluating Software Composition Analysis Tools**
3. **Recommended Open Source Compliance Practices for the Enterprise**
4. **Assessment of Open Source Practices in M&A Transactions**
5. **Open Source Audits in Merger and Acquisition Transactions**
6. **Publishing Source Code for FOSS Compliance: Lightweight Process and Checklists**
7. **A Glimpse into Recommended Practices in FOSS Compliance Management Process**
8. **Free and Open Source Software Compliance: The Basic You Must Know**
9. **A Five-Step Compliance Process for FOSS Identification and Review**
10. **FOSS Compliance: Who Does What—Roles and Responsibilities**
11. **Practical Advice to Scale Open Source Legal Support**
12. **Achieving FOSS Compliance in the Enterprise**
13. **Establishing Free and Open Source Software Programs: Challenges and Solutions**

著者について

Ibrahim Haddad（Ph.D.）は、Linux Foundation の AI およびデータ戦略プログラムのバイスプレジデントを務めています。彼はオープンソース AI プラットフォームの進展と、オープンイノベーターの世代を支援することに焦点を当てています。彼は LF AI & Data Foundation と PyTorch Foundation の両方をリードしています。Linux Foundation に加わる前は、Samsung Electrronics で R&D の副社長およびグローバルオープンソース部門の責任者を務めていました。Haddad 氏は以前、Ericsson Research、Open Source Development Labs、Motorola、Palm、Hewlett-Packard で技術および R&D 管理の役職を歴任しています。

LinkedIn | **Website**

THE LINUX FOUNDATION | Research

2021 年に設立された **Linux Foundation Research** は、拡大するオープン ソース コラボレーションを調査し、新たな技術トレンド、ベストプラクティス、オープンソース プロジェクトのグローバルな影響に関する洞察を提供してい ます。プロジェクトのデータベースやネットワークを活用し、定量的・定性 的手法のベストプラクティスに取り組むことで、Linux Foundation Research は、世界中の組織にとって有益なオープンソースの知見を提供す るライブラリを構築しています。

この著作物を参照する場合は、以下のように引用してください：Ibrahim Haddad, Ph.D.,“ Open Source License Compliance: Challenges Ahead,” foreword by Jimmy Ahlberg, The Linux Foundation, January 2024.