

# 知財部門のためのOSS対応の注意点と対応策

ソフトウェア委員会  
第 2 小委員会\*

**抄 録** ソフトウェア開発の大規模化や複雑化に伴いオープンソースソフトウェア（以下、OSS）の利用が広がる中で、企業においてOSSのライセンス遵守を含むオープンソースコンプライアンスを推進する必要性が高まっている。特に近年は、商品・サービス提供に自社以外の様々な関与者が存在するため、サプライチェーンにおけるオープンソースコンプライアンスを考慮する必要がある。本稿では、OSSに関する基礎知識やサプライチェーンのOSSを含む近年の話題を俯瞰した上で、オープンソースコンプライアンス推進に向けた現場における注意点と対応策を整理した。加えてオープンソースコンプライアンス推進に知財部門が関与する意義を考察した。

## 目 次

1. はじめに
2. ソフトウェアおよびOSSの歴史
  2. 1 はじめに
  2. 2 1950年代
  2. 3 1960～80年代初頭
  2. 4 1980～1990年代
  2. 5 2000年代
  2. 6 2010年以降
  2. 7 小 括
3. OSSライセンスの基礎知識
  3. 1 はじめに
  3. 2 パーミッシブ・ライセンスの特徴
  3. 3 コピーレフト・ライセンスの特徴
  3. 4 新しいタイプのライセンスについて
  3. 5 小 括
4. OSSのサプライチェーン
  4. 1 はじめに
  4. 2 OpenChainプロジェクト
  4. 3 ソフトウェアのライセンス等の情報交換手段について
  4. 4 ライセンス等の表示手段について
  4. 5 小 括
5. OSS利活用時の注意事項
  5. 1 はじめに
  5. 2 フェーズの定義
  5. 3 企画フェーズ
  5. 4 開発フェーズ
  5. 5 出荷前フェーズ
  5. 6 出荷後フェーズ
  5. 7 その他
  5. 8 小 括
6. 知財部門とOSSの関わり
  6. 1 知財部門がOSS管理に関わる意義
  6. 2 競争領域におけるOSSと知財部門の役割
  6. 3 共創領域におけるOSSと知財部門の役割
  6. 4 知財部門が進化する一歩として
7. おわりに

## 1. はじめに

ソフトウェア開発の大規模化や複雑化に伴いOSSは積極的に利活用されてきている。OSSの利用には定められたライセンス条件に従うなど遵守すべきルールがあり、ルールを遵守しないと企業イメージの低下や訴訟のリスクがある。このため、企業においてOSS利用時のルールを遵守するというオープンソースコンプライアンス

\* 2019年度 The Second Subcommittee, Software Committee

ス推進の必要性が高まっている。

オープンソースコンプライアンスの推進には、OSSのライセンス条件を正しく理解して利用すること、利用OSSの最新の動向を把握することが必要である。OSSの利活用の拡大に伴い、OSSに関与する人々は、企画・開発部門だけでなく、営業や調達部門、経営層など多様となりつつある。加えて、近年では、商品・サービス提供のプロセスに社外の人々も多く関与する。そのため、様々な職種の人々が正しい知識をもつこと、自社が関与するサプライチェーンにも働きかけること、がオープンソースコンプライアンス推進に必要である。企業においてOSSの相談を受けることが多い知財部門は、様々な職種の現場を踏まえたアドバイス、上記に対する新たな役割を求められると考えられる。

そこで本稿では、オープンソースコンプライアンスの推進を始めようとする知財部員向けに、OSSの歴史や基礎知識、およびサプライチェーンにおけるOSSといった近年のOSSに関する話題を踏まえて、商品提供のフェーズごとに注意点および対応策を整理した。最後に企業の知財部門がオープンソースコンプライアンス推進に関わる意義を考察する。

本稿は、2019年度ソフトウェア委員会第2小委員会第2ワーキンググループのメンバーである、山中優輝（富士ゼロックス）、伊藤百合香（パナソニック）、奥智照（KDDI）、佐久間拓也（いすゞ自動車）、清水誠也（東芝テック）、丹下聖子（富士通）、忍頂寺毅（DeNA）、堀川大介（日

本IBM）が執筆したものである<sup>1)</sup>。なお、本稿は公知情報を基に構成したメンバーの私見であり、メンバーの勤務先等のいずれの組織の意見を代表するものではない。また、本文献におけるOSSライセンスの省略表記はSPDX short identifierの記載に基づく。

## 2. ソフトウェアおよびOSSの歴史

### 2.1 はじめに

1章で述べた通り、OSS利用のリスクを回避するために、OSS利用のルールに沿った正しい利活用が必要である。特にOSSライセンスは、OSSが生まれた背景や開発者の根底にある思想が反映されているため、それらを理解することがOSSライセンスの正しい理解、そしてOSSの正しい利活用につながる。また、OSS利用のリスクも年代により様々である。これらの内容を理解するために、OSSが辿ってきた歴史を把握することが一つの方法であると考え、本章では図1のようにソフトウェアの誕生からいくつかの年代ごとに分けて動向を記して、その考え方の変遷を整理した。

### 2.2 1950年代

1950年代から60年代にかけて、コンピュータの主な利用者は大学・研究機関であり、ソフトウェアは独自に開発され、パブリックドメインとして扱われるのが通常であった。ソフトウェアはハードウェアの付随物としてソースコード

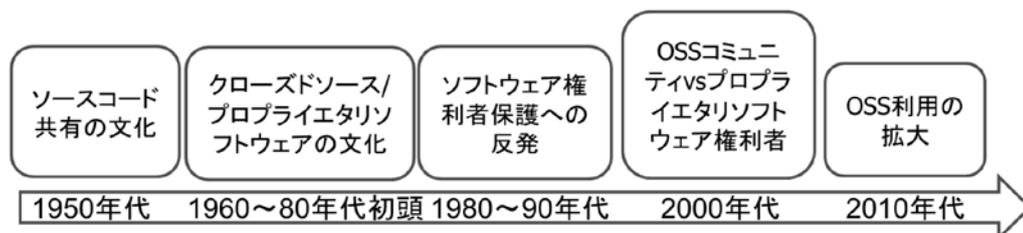


図1 ソフトウェアの歴史

を同梱する形で販売されるのが一般的であった。大学・研究機関も、「公開性と協力の原則」のもと、ソースコードを配布していた。

## 2. 3 1960～80年代初頭

1960年代から80年代初頭にかけて、コンピュータの利用機会が増大してくるとともに、利用者自らのニーズに合わせたソフトウェアの開発／選択が強まった。ハードウェアとソフトウェアは分離して販売<sup>2)</sup>され、ソースコード自体が各社の無形資産として価値を有する様になると、「非公開、改変不可」の流れが加速する。結果、「クローズドソース／プロプライエタリソフトの文化」の傾向が表出する。

この時期に米国では、プロパテント政策が強まり、ソフトウェア関連発明について特許化を目指すようになる。特許適格性に関して米国最高裁が、「アルゴリズムを包含するクレームは、アルゴリズムを包含するという理由だけで特許性を否定されず、そのクレームに関するシステムあるいは方法全体として特許性があれば特許され得る。」とするソフトウェア関連発明の適格性を認めたことにより、特許による保護が始まる (Diamond v. Diehr (1981年)<sup>3)</sup>)。著作権に関して、保護の適格性が争われ (Apple v. Franklin (1983年)<sup>4)</sup>)、プログラムが著作権で保護されるようになる。

また日本でも、ソフトウェア関連発明が特許保護対象になる審査運用指針<sup>5)</sup>が発表される。

## 2. 4 1980～1990年代

1980年代から90年代になると、ソフトウェアの権利者を手厚く保護し、他者による改良や頒布を制限することは、ソフトウェアの発展を妨げるという思想が台頭してくる。即ちソフトウェアの公開と、自由な改良・頒布を認めることこそが、産業の発展に資するという考えである。

この考えに基づき、リチャード・ストールマ

ン氏が、ソフトウェアの利用者にその複製・改変・頒布等の自由を提供する「フリーソフトウェア」の概念を提唱<sup>6)</sup>する。また、フリーソフトウェアのみで構成されたOSを実装する「GNUプロジェクト」が開始され、その成果物がLinuxカーネルと結合したことにより、ほぼすべてのソフトウェアがフリーソフトウェアで出来たOSが完成する。フリーソフトウェアの推進・GNUプロジェクトの支援を目的とした非営利団体「フリーソフトウェア財団」の設立や、GNU General Public License (GNU GPL) がリリースされたのもこの時期である。

## 2. 5 2000年代

2000年代に入ると、「ソフトウェアの自由な改良・頒布を認めることでソフトウェアの発展を促進しようとするOSSコミュニティ」と、「ソフトウェアの権利者を手厚く保護することでソフトウェアの販売等による収益性を高めようとするプロプライエタリソフトウェア権利者」という二つの勢力の対立構造が鮮明化する。(注：OSSを利用する個人や企業は、必ずしもこのいずれか一方に属するとは限らず、むしろ状況次第でどちらの立場にもなりうるということである。)

まず、OSSコミュニティの代表的な活動としては、米国のフリーソフトウェア財団や欧州の [gpl-violations.org](http://gpl-violations.org)をはじめとするOSS保護団体が、OSSを搭載した商品を販売する企業の監視を強化し、OSSライセンス違反に対して積極的に警告活動を行ったことが挙げられる。その中には法廷闘争にまで発展したものも相当数あり、そこではSoftware Freedom Law Center等の法律専門組織が、OSS陣営の代理人として活躍した (Free Software Foundation, Inc. v. Cisco Systems, Inc. (2008)<sup>7)</sup>)。

これに対し、プロプライエタリソフトウェア権利者は、OSSが自らの知的財産権を侵害して

いるとして、積極的な権利行使を展開した。当初その標的は主にOSSコミュニティであったが、OSSコミュニティに損害賠償金を支払うだけの資力がないことから、次第にその矛先はOSSを利用する企業向けられることとなる(Microsoftによる特許権侵害主張(2007~))<sup>8)</sup>。

こうしたプロプライエタリソフトウェア権利者からの権利行使に対抗すべく、OSSコミュニティ側ではその結束の強化が図られた。例えば、2005年には、企業コンソーシアムOpen Invention Network (OIN) が設立された。OINではLinuxやLinux関連システム・アプリケーションに関連する特許を積極的に買収するとともに、LinuxやLinux関連システム・アプリケーションに対して自身の特許を主張しないことに同意した企業や個人に対し当該特許をロイヤルティ・フリーでライセンスする、という活動が展開された。

## 2. 6 2010年以降

2010年以降になると、こうした対立構造が一転し、OSSがIT業界においてベストプラクティスとしての地位を確立していく。その背景には、世界中のエンジニアにより日々開発・改良を重ねられるOSSが、コスト面のみならず、品質、信頼性、セキュリティの面でも優れているという認識が広く浸透していったことがある<sup>9)</sup>。2018年にThe Linux FoundationがGlobal Fortune 2000企業を対象に実施した「2018年オープンソース プログラム管理調査」によれば、72%の企業がオープンソースを非商用または社内で頻繁に使用、55%の企業がオープンソースを商用製品に使用しているといい、企業におけるOSSの利用が今や一般的なものとなっていることが見て取れる。

こうした潮流の変化を象徴する出来事として、2018年にMicrosoftがOINに加入したことが挙げられる<sup>10)</sup>。これにより、かつてLinux陣営に対

して積極的な特許権行使を行っていたMicrosoftが一転、Linux陣営に対して特許6万件を無償開放することとなった。

こうしてOSSコミュニティ・プロプライエタリソフトウェア権利者両陣営の対立はある程度の落ち着きを見せたものの、かといってOSSを利用する企業や個人にとってのリスクが完全に消え去ったわけではない。当然のことながら、OSSライセンス条件に違反すれば、OSSコミュニティ等から、通常の契約違反の場合と同様に是正措置や損害賠償を求められうる。自らはOSSを使った事業を行わず、従ってOSSコミュニティへ歩み寄るインセンティブを持たないPatent Assertion Entities (以下、PAEs) は、今もなおOSS利用者に対する権利行使を継続している。(Sound View Innovations v. OSS利用者(2016~))<sup>11)</sup>。他にも、PAEsの一つとして知られるRothschild Patent Imaging (RPI) が、非営利のOSS開発団体であるGNOME Foundationを特許侵害で提訴した。本件は、PAEsがOSSの利用者ではなくOSSの非営利団体の開発元を提訴したことが特徴といえる。本件では、OINが他の企業や組織と連携してGNOME Foundationを支援することを表明した<sup>12)</sup>。

## 2. 7 小 括

歴史を振り返れば、ソフトウェアの発展には自由な公開・頒布が必要であり、それらを実現する手段の一つがOSSのライセンスであることがわかる。

また、IT業界においてOSSの利用は避けて通れないものとなりつつある。そのため、OSS利用のメリットやデメリット、PAEsからの訴訟を含むリスクを十分に把握した上で、OSSとどのように向き合うのかといった戦略を立てることが非常に重要になってくる。

次章からは、オープンソースコンプライアンス実践のために必要なOSSライセンスの基礎知

識、およびOSSのサプライチェーンに関する内容を見ていく。

### 3. OSSライセンスの基礎知識

#### 3.1 はじめに

OSSのライセンスは、パーミッシブ・ライセンス、準コピーレフト・ライセンス、コピーレフト・ライセンスに大別される。これらに共通する特徴は、OSSの複製や使用だけでなく、改変や配布・再配布も自由に行うことを許諾する点である。一方、これらのライセンスは、OSSの配布・再配布に際して要求される義務という点においては相互に異なる特徴を有する。

パーミッシブ・ライセンスは、OSSの配布・再配布に際してソースコードの公開義務を課さない。対照的に、コピーレフト・ライセンスでは、OSSの配布・再配布に際して、OSSのオリジナルのソースコードはもちろんのこと、これを改変した派生ソフトウェア（二次的著作物）、さらには自社で開発したソフトウェアについてまで、ソースコードの公開義務が課される場合がある。準コピーレフト・ライセンスは、それ

ら2つのライセンスの中間に位置し、OSSを改変して他に提供する場合にのみ、その改変部分を含めたソースコードの公開義務が課される。

表1にパーミッシブ・ライセンス、準コピーレフト・ライセンス、コピーレフト・ライセンスの要約を示した。紙面の関係上、ここでは代表的なパーミッシブ・ライセンス（MIT, BSD (BSD-2-ClauseやBSD-3-Clauseなど), Apache License, Version 2.0（以下、Apache-2.0）とコピーレフト・ライセンス（GNU General Public License: GPL-2.0, GPL-3.0）に絞って、ライセンスの特徴を概説する。加えて、近年の新しいタイプのライセンスについても言及する。

#### 3.2 パーミッシブ・ライセンスの特徴

##### (1) ライセンスの下でできること

MIT, BSD (BSD-2-Clause, BSD-3-Clause), Apache-2.0などはパーミッシブ・ライセンスの代表例である。これらの条件でライセンスされるOSSは、そのまま複製して使用することができる他、これを改変し、さらに再配布することが可能である。例えば、ソースコードを改変して二次的著作物を作成して使用したり、改変

表1 ライセンスの特徴

ライセンスタイプ	ライセンス名	ライセンスの特徴 (v=該当)				
		使用や利用 (複製, 改変, 再配布など) ができる	保証や補償が ない	再配布時の 著作権および 利用許諾条件 の表示義務	再配布時の ソースコード の開示義務	特許許諾義務
パーミッシブ	MIT	v	v	v	-	-
	Apache-2.0	v	v	v	-	v
	BSD-2-Clause	v	v	v	-	-
	BSD-3-Clause	v	v	v	-	-
準コピーレフト	MPL-2.0	v	v	v	v 1	v
コピーレフト	GPL-2.0	v	v	v	v 2	-
	GPL-3.0	v	v	v	v 2	v

v 1. OSSを改変して他に提供する場合にのみ、その改変部分を含めたソースコードの公開義務が課される

v 2. OSSの再配布に際して、OSSのオリジナルのソースコードと、これを改変して作成した派生ソフトウェア、さらには自社で開発したソフトウェアについてまでソースコードの公開義務が課される場合がある

の有無にかかわらずソフトウェアを再配布することができる。

また、パーミッシブ・ライセンスの場合、OSSの再配布時にオリジナルのOSSライセンスとは異なるライセンスを適用して配布することも許容される。このため、企業独自のライセンス条件をOSSに適用してこれを配布することも可能である。例えば、OSSを同梱した商用ソフトウェアを販売する場合、この商用ソフトウェアのライセンス条件と同一条件でOSSを提供することもできる。

## (2) ユーザに課される義務

上述の通り、OSSを広範に利用することができる一方、パーミッシブ・ライセンスの下でOSSを配布・再配布する場合には、著作権表示(copyright notice)と利用許諾条件の表示(license notice)の2点セットの表記を含める必要がある。パーミッシブ・ライセンスの場合、たとえ改変を加えたソフトウェアを提供する場合であっても、OSSの再配布時にこの2点セットを含めておけば、すでに述べたように企業独自のライセンス条件の下でOSSを配布することが可能である。これが、後述するコピーレフト・ライセンスと比べた場合に、パーミッシブ・ライセンスが企業にとって寛容なライセンスと言われる所以である。

## 3. 3 コピーレフト・ライセンスの特徴

### (1) ライセンスの下でできること

コピーレフト・ライセンスの代表例としてGPL (GPL-2.0とGPL-3.0があるが、以下これらをまとめてGPLとして参照する)を取り上げる。GPLの下で提供されるOSSは、自由にこれを複製して使用し、改変や再配布を含めた広範な利用(改変や配布)が可能である点で、パーミッシブ・ライセンスで提供されるソフトウェアと同じである。しかし、とりわけその配布・

再配布に際しては、パーミッシブ・ライセンスに見られない特徴がある点に注意を要する。

### (2) ユーザに課される義務

コピーレフト・ライセンスのGPLは、OSSの再配布時にGPLと異なるライセンス条件での配布を認めていない。このため、ユーザは、OSSに独自に改変等を加えた場合であっても、その派生ソフトウェアをGPLと同じ条件の下でしか配布することができない。結果として、ユーザは、その派生ソフトウェアについても、OSSのオリジナルのソフトウェアと同様、そのソースコードを利用可能にしなければならない。これがユーザに課せられる再配布時のソースコード公開義務である。

注意点として、GPLでいう「配布」には、自社内や一定のグループ会社内での配布は含まれないと考えられている。このため、自社内等の従業員間でOSSを共有してもソースコードの公開義務等は課せられない。あくまで社外の第三者への配布をトリガとして課せられる義務であるが、かかるソースコードの公開先は、配布する相手方のみで足り(GPL-2.0の3条, GPL-3.0の6条)、不特定多数の第三者向けにソースコードを一般公開することが必須ではない。

また、ソースコードを公開する範囲については、GPLの「伝播性」についても考慮しなければならない。GPLでライセンスされるコード(GPLコード)を改変して得られた派生ソフトウェア(二次的著作物)のみならず、そもそもソースコードの公開を意図せず自社で開発したコード(プロプライエタリ・コード)であっても、これに一部でもGPLコードが含まれている場合やGPLコードへのリンクが含まれている場合<sup>13)</sup>、かかるプロプライエタリ・コードについても、GPLの適用が必要となる。GPLの持つこのような「伝播性」から、GPLと同一条件の下での配布が必要となるため、上述のような派生

ソフトウェアやプロプライエタリ・コードを第三者に配布する場合には、ソースコードを公開する義務が生じるわけである。GPLコードとプロプライエタリ・コードを組みわせて商用ソフトウェアを開発する場合は、商用ソフトウェアのソースコード開示を避けるべく、GPLの「伝播性」の有無、特にリンクの有無などを確認する必要がある。

最後に、ソースコードの公開方法についてであるが、具体的には、(1) OSSのソースコードを商用ソフトウェアに同梱して一緒に配布するか、(2) 商用ソフトウェアのユーザからの要求に応じてOSSのソースコードを提供するか、(3) 商用ソフトウェアのユーザがソースコードをダウンロードできるようにWebサイトにソースコードをアップロードするか、の何れかを実施する必要がある。

### 3. 4 新しいタイプのライセンスについて

近年、クラウドサービスや倫理的観点などに基づく新しいタイプのライセンスが話題となっている。

クラウドサービスに関するライセンスとして Server Side Public License (SSPL)<sup>14)</sup> や Business Source License (BSL)<sup>15)</sup> などが新しい。SSPLはGPL-3.0に修正を加えたものであって、例えば、SSPLが適用されるソフトウェアの機能をSaaSで第三者に提供する場合、そのソフトウェアを含むサービス用のソースコードを第三者に開示しなければならない。一方で、BSLは、無償で利用できる範囲を定めてそれを超える場合は有償契約を求め、所定期間経過後にライセンス条件を変更する。また、開発者自身の倫理に基づき、利用目的を制限するライセンスは Ethical License (倫理的ライセンス) と呼ばれる。たとえば、ソフトウェアの反人道的な利用を許さないとするもの<sup>16)</sup> や、労働条件に基づいて利用許諾するもの<sup>17)</sup>、などがある。

## 3. 5 小 括

本章では、OSSライセンスの基礎知識と新しいタイプのライセンスについて述べた。近年、複数の企業が異なるOSSを利用して一つのソフトウェアを開発することが多くなっており、企業間のサプライチェーンにおいてもライセンス遵守が重要となってきている。次章では、OSSのサプライチェーンについて概観する。

## 4. OSSのサプライチェーン

### 4. 1 はじめに

ソフトウェア開発の大規模化や複雑化が進むなか、製品にOSSが取り込まれる事例が増えてきている。そのため、最終製品に至るまでには、OSSコミュニティ、ソフトウェア開発事業者、SDKなどの提供事業者、最終製品ベンダーなどがサプライチェーンを構成することになる。このとき、OSSが他のOSSに依存している場合もある。ライセンス遵守などのオープンソースコンプライアンスは、企業などの組織内ではもちろんのこと、サプライチェーンを構成する組織間でも十分に実施される必要がある。

本章では、OSSのサプライチェーンにおけるオープンソースコンプライアンスに関連する話題を取り上げる。最初にサプライチェーンにおけるオープンソースコンプライアンスを推進する取組である OpenChain プロジェクトについて、次いでソフトウェアのライセンス等の情報交換手段について、最後にソースコードにおけるライセンス等の表示手段について、見ていく。

### 4. 2 OpenChain プロジェクト

The Linux Foundation (LF) のOpenChainプロジェクトは、信頼できる一貫したコンプライアンス情報とともにOSSが流通するサプライチェーンの実現を目的として、組織内におけるコ

ンプライアンスのためのプログラムの構築及びその実施をサポートする取組である。OpenChainではSpecification（仕様）、Curriculum（一連の教育内容、カリキュラム）及びConformance（適合）を大きな柱としている。

OpenChain仕様は組織内で必要となるコンプライアンスプログラムを定義し、その主要な3つの要素にProcess（手順）、Policy（方針）、Training（教育）を挙げる。仕様は、個別要件、証跡となる資料、論拠（それらを必要とする理由）で構成されている。これは、様々な産業や規模の企業で適用できるようにするため、仕様の策定にあたり詳細な手順や基準までは規定しないという方針に基づくものである。次いで、OpenChainカリキュラムとしては仕様に定める教育で活用可能な素材を作成し提供している。そして、OpenChain適合は組織間の相互信頼を確かなものにするために重要となるが、その認証の手続きも事業や企業の態様に合わせて選択できるものとし、Self-Certification（自己認証）、Independent Compliance Assessment（独立社外評価）、Third-Party Certification（第三者認証）などの手続きを利用できる。

OpenChainプロジェクトはその仕様を業界標準として普及させるため、ISOのJoint Technical Committee 1（JTC-1）にて Publicly Available Standard（PAS）規格としての標準化を働きかけている。OpenChain 2.1仕様に基づく内容を提案しているため、OpenChain 2.0適合は、ISO適合となる予定である。

#### 4. 3 ソフトウェアのライセンス等の情報交換手段について

自社管理やサプライチェーンのいずれでも、製品やサービスを構成するソフトウェアコンポーネント（部品）の情報管理が重要である。こうした情報は、Software Bill of Materials（Software BOM, SBOM, ソフトウェア部品（構成）表）と

して管理される。インフラや医療機器でのOSSの利用を背景として、その脆弱性管理のためにも、サプライチェーンにおけるソフトウェアコンポーネントの透明性が重要視されている。関連する取組に、経済産業省商務情報政策局サイバーセキュリティ課によるサイバー・フィジカル・セキュリティ対策フレームワーク<sup>18)</sup>や、米商務省電気通信情報局（National Telecommunications and Information Administration: NTIA）による“Software Component Transparency”<sup>19)</sup>などがある。

LFのSPDX（Software Package Data Exchange）<sup>20)</sup> ワークグループは、ワークグループ名称がそのままフォーマット名称を示しており、ソフトウェアパッケージに関連するソフトウェア名やバージョン、ライセンス、著作権表示などの情報を共有するための標準的なフォーマットを策定する取組である。最新仕様v2.2の特徴の一つに、Appendix VIII “SPDX Lite”の追加が挙げられる。SPDX LiteはOpenChainプロジェクト Japan Work Group がSPDXに基づいて「必要最小限」に簡素化した記述フォーマットとして提案するものである。このミニマム性を指向するコンセプトはSPDXワークグループで評価されており、SPDXの次期仕様v3.0の検討に影響を与えている。

なお、SBOMの表現方法は、SPDX以外にもSWID（ISO/IEC 19770-2:2015）<sup>21)</sup>が知られている。

#### 4. 4 ライセンス等の表示手段について

個別のファイルに著作権表示や適用するライセンスの宣言の明示がない場合、そのファイルだけが流通すると適切に著作権とライセンスに関する情報を得られないことが問題となる。

そこで、Free Software Foundation Europe（FSFE）のREUSEプロジェクト<sup>22)</sup>は、個別ファイルに対してSPDXの書式に基づく著作権及び

ライセンス情報の明記を提案している。識別子と要素が明確なため、人が見ても、ツールなどで自動処理しても、判別しやすい表示方法として注目されつつある。表示例は次のようになる：

```
/*  
 * SPDX-FileCopyrightText: 2020 Jane Doe  
<jane@example.com>  
 *  
 *SPDX-License-Identifier: MIT  
 */
```

OSSについて2人以上の開発者がいる場合、個々人が著作権者となりえる。個々のファイルでは数名の開発者の程度としても、プロジェクト全体では数十名に上ることもありえる。そのような場合、どのような著作権表示が適切になるかが問題となる。この点について2020年1月10日にLFから一つの見解<sup>23)</sup>が示された。ここでは、個別の著作権者を表示するのではなく、LF関連プロジェクトの多くのコミュニティが採用する表示の推奨例としてOSSプロジェクト名の利用を挙げている：

(以下の「XYZ」はOSSプロジェクト名)

- ・ Copyright The XYZ Authors.
- ・ Copyright The XYZ Contributors.
- ・ Copyright Contributors to the XYZ project.

## 4.5 小 括

本章では、OSSライセンスを遵守してのサプライチェーンを組織間で一貫して信頼あるものにするための取組について紹介した。重要なことは、サプライチェーンを構成する個々の組織がオープンソースコンプライアンスを適切に行い、サプライチェーンを構成する組織間で情報共有することである。次章では、ソフトウェア開発運用業務に即してOSSの選定を含む利活用やライセンス遵守のための注意点を見ていく。

## 5. OSS利活用時の注意事項

### 5.1 はじめに

OSSの活用頻度が高まる中、知財部員としてOSSに関して、特に、ライセンスに関連したアドバイスを求められる機会が増えていると推測される。その際に、OSSが利用される現場ではどういった観点での考慮や検討が必要であるかを、知財部員も知っておくことで、現場からの相談に実務的なアドバイスができると考えられる。

そこで本章では、“現場（営業、開発（SE含む）、保守、調達）”の担当者目線で、何を考えなければならぬか、どのような対応が必要か、を製品・サービスのフェーズごとに紹介する。

なお、前提として、どのようなソフトウェア開発が行われるかは、ビジネス形態によって異なる。以下では、

- ・ ソフトウェア製品（アプリ含む）の開発
- ・ ソフトウェアの受託開発やシステム構築
- ・ クラウドサービス等のサービスで利用するソフトウェア開発

の場合を取り上げる。

### 5.2 フェーズの定義

様々な部署や担当者がかかわるソフトウェア製品・サービスでは、ライフサイクルにおけるフェーズごとに、注意事項を整理しておくことが、全社でOSSを正しく利活用するために有用である。ここでは、ソフトウェア製品・サービスのライフサイクルを、(1) 企画、(2) 開発、(3) 出荷前、(4) 出荷後というフェーズに定義する。

- ・ 企画（PLAN）：製品やサービスの機能を決定し、必要な構成部品、条件、関係組織等を決定する
- ・ 開発（DEVELOP）：基本設計、設計、プログラミング、試験（テスト）
- ・ 出荷前（DELIVER）：出荷判定や製品登録な

どの最終審査

- ・出荷後(OPERATE)：維持管理,顧客サポート  
なお,近年主流となりつつある,開発と運用が一体となったDevOps型の開発を行う製品・サービスは,「出荷後」フェーズでの知見を,「企画」フェーズに投影する形で,サイクルを回している。

### 5.3 企画フェーズ

利用するOSSを,企画フェーズで決定可能な場合もあれば,続く開発フェーズで決定する場合もある。ここでは,OSSを部品としての利用を検討する際に考慮すべき観点を記載する。

- ・候補となるOSSを調査し,必要な要件(例:機能・性能)を満たすかを評価する。
- ・OSS入手時のバージョン,日時,入手時のバージョンに応じたライセンス条件を保存する(変更される可能性があるため)。
- ・他社への開発発注をする場合,発注先でのOSS利用について契約に規定する(OSS利用時のライセンス遵守,責任分担など)。
- ・他社製品を採用する場合,OSS利用有無の確認,利用している場合にはライセンス条件を遵守していることの確認,責任分担など契約へのOSS関連の条項の盛り込みの検討を行う。
- ・システムなどの受託開発の場合,OSS利用について,ライセンス文書等を顧客に提示したうえで,顧客承認を取得する。さらに責任分担についても契約を締結する。
- ・ライセンス条件の中に,プログラム外で対応する必要がある義務(例:GPLライセンスによるソースコードの公開義務)が含まれる場合,義務を果たすための対応を実施する。

#### (1) サポート体制の確認

企業が顧客に提供するソフトウェアには,顧客との契約条件に対応できるサポート体制が必要となる。社内での対応や社外からのサポート

の可能性を確認する。

#### 1) 社内での対応可能性

過去に社内で使用実績がある場合,OSSの使用に関する情報(実際に使用してわかったメリット・デメリット)を収集する。また,使用実績がない場合には,OSSが採用するプログラム言語への自社の習熟具合,出版物の有無,インターネット情報の量等を踏まえ,そのOSSへの自社の対応可能性(対応のための学習コスト)を検討する。

#### 2) コミュニティによるサポート

OSSの開発コミュニティは,商用ソフトウェアと同様,OSS毎にバグ修正に対応するサポート期間を定めていることがある。OSSの選定においては,まずこのコミュニティによるサポート期間を確認する。ここで,OSSによっては,特定のOSSのバージョンをベースとし,他のOSSのバージョンよりも長期間のサポートを行うLTS(Long Term Support)版が用意されている場合がある。更に,近年ではLinux Foundation下において,LTS版より更に長いコミュニティサポートを提供するSLTS(Super Long Term Support)版に関する取り組みも始まっている(Civil Infrastructure Platform(CIP))。特に,OSSの採用を検討しているソフトウェアが長い期間のライフサイクル(顧客へのサポート)を想定している場合には,LTS版やSLTS版がないかを確認すると良いだろう。

#### 3) 開発主体や出資状況

OSSの開発は有志コミュニティにより行われるものと認識されてはいるが,開発の方向づけや最終決定を行うOSS開発の中心部においては,主要開発者の賃金等の資金を必要としていることがある。OSSの開発活動が他者から一定以上の支援を受けていることが確認できた場合,そのOSSの継続性を推定できる。この支援の一例としては,Linux FoundationのCore Infrastructure Initiative(CII)がある。また,2019年にお

いては、Linux FoundationでCommunityBridge (funding.communitybridge.org)、GitHubでGitHub Sponsors (github.com/sponsors/)が立ち上がっており、OSSの利用者等がOSS開発者に金銭的な支援を行う仕組みが構築されつつある。

#### 4) 社外の有償サービスでの対応可能性

OSSに関連するサービスを行う社外の事業者の存在及びそのサービス内容を確認する。

### (2) リスクの評価

OSSは無保証で提供されており、OSSの利用は、様々なリスクを内在する。従って、リスクを確認・評価した上でのOSSの利用が望まれる。ここでは、OSS利用のリスクの一例とその確認方法を述べる。

#### 1) ライセンス条件を遵守できるか

3章で述べたものに代表されるOSS利用者に課される各種の義務を果たすことができるかを確認する。また、Apache-2.0やGPL-3.0等のOSSには、OSSのユーザが保有する特許権の権利行使の制限に繋がる条項が含まれている。OSSの利用により制限される権利行使の範囲及び保有する特許の活用目的を踏まえて、利用しても問題ないOSSであるかを確認する。

OSSの中には、ライセンス条件が提示されていないOSSも存在する。そのようなOSSは、著作権法のデフォルトルールに基づき、OSS作者(著作権者)以外は自由に使えないと解釈することが安全である。そのためライセンス条件の提示されていないOSSは、利用を避けるか、OSSの作者にコンタクトを取り、許諾を受けた上で利用すると良い。

#### 2) 訴訟や係争の有無

前述のSound View Innovationsの事例のように、OSSを利用した企業が特許権の侵害により特許権者から訴訟を提起される場合がある。利用したいOSSに関する訴訟や係争があれば、利用を避けるのが安全である。

### 3) 脆弱性(セキュリティ)の情報

OSSに脆弱性が報告されていないかをコミュニティ内での報告や専門のWebサイト等で確認する。脆弱性の情報が公開されているWebサイトの例として、JPCERT/CCと情報処理推進機構(IPA)により運営されるJapan Vulnerability Notes (<https://jvn.jp/>)がある。

### (3) OSSの信頼性

OSSは、その開発状況等を確認することで、OSSの信頼性を推定できる。コミュニティを通じた共創活動により、多様な視点から、より良いソフトウェア部品へと改善がなされるためである。ここでは、OSSの信頼性を推定する方法の一例について述べる。

#### 1) 初期バージョンからの経過日数

開発が開始されてから間もないOSSは、開発の方向性の大幅な変更の可能性や、他のOSSとの競争(競争に負けた場合に開発が停滞することがある)等の要因により、利用企業が事業で必要とする期間の使用に耐えられるか推定しにくい状況にある。従って、開発が開始されてからある程度の期間が経過したOSSを選択するのが安全である。

#### 2) Community Health

OSSの継続性、バグや脆弱性への将来における対応力の予測のため、そのOSSの開発コミュニティの活発度合い(Community Health)に関わる情報を確認する。ここでは、Community Healthを確認する要素や方法として議論されているものの一部を紹介する。

##### ・定期的、計画的なバージョンアップ

定期的なバージョンアップや、計画的なバージョンアップがなされているかを確認する。この様なOSSは、将来の継続性について一定の期待ができるOSSであると考えられる。OSSが、最近バージョンアップがなされたものであるとなお良い。長期間にわたってバージョンアップ

がなされていないOSSは、脆弱性への対応が放置されている可能性がある等、そのOSSの利用が事業のリスクにつながる懸念がある。

・コードの変更量

特定の期間における、コードの変更（リポジトリがGit管理である場合にはCommit）が定期的及び一定量以上行われているかを確認する。

・コードの変更提案及びその対応

特定の期間における、コミュニティからのコードの変更提案数、それらの変更提案がされてから審査が完了するまでの期間等を確認する。

・バグ（Issue）の数及びその対応

特定の期間内における、コミュニティからのバグ（Issue）の報告量及びそのバグ（Issue）の解決期間を確認する。

・その他

上述の要素の他、OSSコミュニティの主要開発メンバーのバックグラウンドの多様性、ライセンスが明示されたソースコードの割合や数、ソースコードのOSI (Open Source Initiative) の認定を受けたライセンスの割合等がCommunity Healthを推定する項目とされている。

### 3) Community Healthの確認手段

Community Healthの確認に関しては、ソフトウェアやWebサービスなどが整備されつつある。例えば、GitHubにおいてはサービスの内部にCommunity Healthに関連する指標の一部を表示する機能を備えている (Insights)。また、OpenHub (www.openhub.net) のようなOSSの情報を提供するWebサービスにより確認することもできる。Community Healthに関連するデータがインターネット上で取得できないOSSについては、リポジトリのデータに基づき、GrimoireLab (chaoss.github.io/grimoirelab/) やGitStats (gitstats.sourceforge.net/) 等のソフトウェアを使用して視覚化・確認することができる。Community Healthに関する指標の視覚化の一例として、GrimoireLabを用いて集計した、

機械学習フレームワークPyTorchのCommit (プログラムコードの変更) 数の推移を図2に示す。AI技術の盛り上がりに伴い開発が活発になっていることが見て取れる。

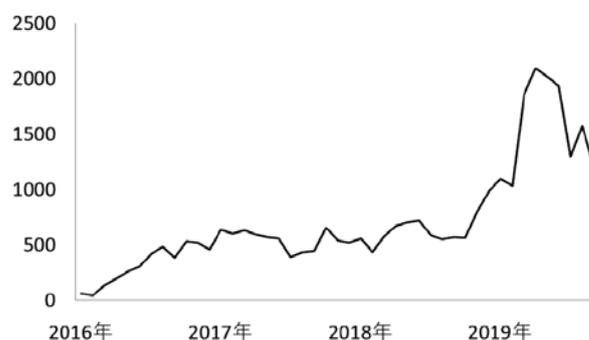


図2 PyTorchのCommit数推移

## 5. 4 開発フェーズ

企画フェーズにおいてライセンス等の必要な情報が保存されているかなどの再確認と合わせて、以下の点に注意する。なお、開発フェーズにおいてOSSの利用を決定した場合には、前節の企画フェーズ同様の対応を行う。

- ・プログラムに認識していないOSSが混入していないか。混入していた場合にはOSSの調査を実施し、採用可否を決定する。
- ・OSSとOSS以外を分離して管理する (例：ディレクトリを分けるなど)。
- ・伝播性のあるライセンス (例：GPL) の有無を検討し、必要に応じて伝播性の回避策を講じる。
- ・ライセンス条件の中にプログラム上で対応する必要がある義務 (例：起動画面で対象OSSを含む旨の表示義務) が含まれるかを確認し、必要に応じて開発に盛り込む。
- ・OSSの追加または変更時には、再度、企画フェーズの検討を実施する。
- ・発注に対する納品物における、OSS利用有無の確認、利用OSSのライセンス条件確認を実

施する。

## 5.5 出荷前フェーズ

出荷判定を行う検査部門は、企画フェーズ、開発フェーズでの確認結果のエビデンスを精査したうえで、最終的な出荷判定を行う。そこで、事前に、社内で、検査部門が精査する書類一式を定義し、開発・企画部門がその書類を提出するルール作りをしておくことが望ましい（例：利用OSSリスト、ライセンス名及びライセンス文書等）。

## 5.6 出荷後フェーズ

製品・サービスのサポート範囲において、顧客からの対応要請に対応する。サポート範囲においてOSSの修正対応などをする場合には、例えば以下の観点での確認を行う。

- ・OSSのバグ等により、バージョンアップ版を採用する場合の互換性を確認する。
- ・自社でバグを修正する場合には、ライセンス条件の改版条件を確認し遵守する。
- ・自社での修正版をコミュニティに投稿する場合には、コミュニティ参加の社内ルールを遵守する。

## 5.7 その他

海外との取引が予測される場合、企画や開発部門は、輸出管理の手続きを確認する。フェーズとしては「企画」のタイミングから、確認作業を進めることが好ましい。また、社内に輸出管理の専門部署があれば連携して進めることとなる。

以下、輸出管理について、OSSの側面から、簡単に説明する。

我が国を始め各国では、安全保障の観点から、貨物や技術の輸出を管理する。我が国では外国為替及び外国貿易法のもと、「リスト規制」と「キャッチオール規制」に従うことになる。

ソフトウェアも技術の一つとして注意が必要になる。一方で、不特定多数の者に公開されている書籍や論文などの「公知の技術」であれば、経済産業大臣の許可を受ける必要がないとされる。プログラムを公知技術として取引する例としては、「ソースコードが公開されているプログラムを提供する取引」が例示されている。独自に改変することなく利用するOSSは、およそこれに該当することになるだろう。

米国の場合でも、上記の原則は同様である。ただし、米国では、「再輸出規制（米国産の技術の輸出管理について米国の規制の域外適用を求めるもの）」がある。米国産のOSSを改変して利用する場合には、十分な注意が必要である。

## 5.8 小 括

本章ではOSSを利用する現場目線で、商品・サービスのフェーズ毎に注意点と対応策を紹介した。紹介した観点を起点に、適用される企業の実態に合わせてOSSの注意点と対応策を整理していくとよいであろう。

次章では、知財部門がOSS管理に関わる意義について考察する。

## 6. 知財部門とOSSの関わり

### 6.1 知財部門がOSS管理に関わる意義

AI（Artificial Intelligence）やDX（Digital transformation）の進展に伴い、企業活動はOSSの利用と切っても切り離せない時代に突入している。知財部門は今後、

- ・競争領域：知財獲得、知財権の行使
- ・共創領域：OSS利用、知財開放

の線引きを主導する等、知財とOSSを理解する立場から事業戦略に深く関与することが求められる。本章では各領域における知財部門の果たすべき役割、および今後の知財部門のあるべき姿を考察する。なお、知財獲得とは、伝統的な

特許権等の産業財産権の取得に留まらず、自社の開発ノウハウや、コミュニティとの関係構築等の無形資産も含まれる。

## 6. 2 競争領域におけるOSSと知財部門の役割

OSSの利用は、ライセンスが付随しているものの、無保証の技術利用である。OSSの利用が他社特許権の侵害になることもあるが、OSSコミュニティやOSSに関わる技術情報から自社技術の競争性や特許取得の必要性の判断機会や、OSSを部品として利用した自社製品・サービスに関する新機能自体に関する特許権の獲得機会も得られる。また、3章、4章で説明した通り、ライセンスの解釈により自社の保有知財が棄損する可能性や、サプライチェーン全体を俯瞰するとライセンス矛盾が生じている可能性もある。

知財部門はこれまで、技術と事業の特性を見極めながら知財の権利化戦略を立案し、取得した権利の契約交渉に携わってきた。OSS利用は、5章で説明した通り、ライセンスを正しく理解し、コミュニティを通じた技術動向を注視し、係争の有無を確認するという点で、知財部門がこれまで担ってきた役割の延長にあると言える。今後競争領域において、知財部門はOSSがもたらす自社の事業戦略への影響を考慮して、知財獲得、知財権の行使を検討していくべきである。

## 6. 3 共創領域におけるOSSと知財部門の役割

2章で説明した通り、OSSを巡っては、過去には産業財産権による保護を支持する立場と、自由な利用と改変を支持する立場との対立が顕在であった。しかし、近年ではその対立構造が一転し、協調してOSSを活用する傾向へと変わってきている。また、OSSに限らず、企業同士が協調・協力することが増えてきた中で、事業戦

略において設定した共創領域での自社の優位性を確立するために、知財部門は従来にない新たな役割を担う必要が出てくる。

例えば、国連サミットで採択されたSDGs<sup>24)</sup>では、目標9として「産業と技術革新の基盤をつくろう」とある。知財部門は伝統的に権利取得とライセンス交渉の担当を担い、適切な権利獲得と行使を行うことで技術革新の基盤作りに貢献してきた。今後は今までの役割に加え、権利化をしない又は知財を開放するといった手段を用いてOSSコミュニティを活性化させる役割を担うことで、技術革新へ貢献する活動も求められるであろう。

## 6. 4 知財部門が進化する一歩として

2章でみたように、特許制度とオープンソースは一見すると対立する概念のように思えるが、「公開の代償として独占権を付与することで、技術の累積的進歩を図る特許制度」と、「公開を前提として、みんなで改良技術を考えるオープンソース」とでは、「技術の公開により技術の累積的進歩を目指す点」で共通の思想を有する。競争優位の源泉が、有形資産から無形資産にシフトしている今日においては、全社視点に立ち、自社の事業に必要な知財（無形資産）形成をデザインすることが知財部門の新たな役割として求められる。このような新たな役割を知財部門が担うには、OSSに関わる部門内外の人々へ、現場での円滑なコンプライアンスの実施に向けた啓発活動も欠かせない。また、知財形成のデザインを円滑に行うために、開発部門との連携をこれまで以上に強化することも大切となる。知財部門がOSS対応を率先して主体的に取り組むことは、競争領域において特許権の活用を中心に考えてきた「特許部門」としての知財部門から脱皮する大きな一歩になると言えよう。

## 7. おわりに

冒頭述べたように、OSSの利活用の拡大に伴い企業におけるオープンソースコンプライアンス推進の必要性が高まっている。企業内における推進体制を構築するだけでなく、サプライチェーンといった企業間での実施も必要であり、OSSへの対応に習熟していない組織や企業においては、「何に注意すべきか」を知ることから始めなければならない。そのような組織においてOSSの対応を推進する方々に、本稿が注意点や対策の方向性を把握する手助けとなれば幸いである。オープンソースコンプライアンスに関する事項は企業間での実施が重要なため、オープンに議論・共有されている。OSSに関して、社内で解決できない悩みがあれば、例えば本稿内に記載した社外の組織等にその解決策を求めてもよいだろう。

### 注 記

- 1) メンバーの所属は、本研究の検討または執筆当時の所属である。
- 2) 野田哲夫 情報通信技術の発達と情報産業の成立 第2回 コンピュータ産業の成立, IBMの市場支配  
<http://www.eco.shimane-u.ac.jp/nodat/infoind/infoind0602.pdf>
- 3) 一般社団法人 日本国際知的財産保護協会 各国における近年の判例等を踏まえたコンピュータソフトウェア関連発明等の特許保護の現状に関する調査研究報告書  
[https://www.aippi.or.jp/pdf/hokoku/h29/h29\\_02.pdf](https://www.aippi.or.jp/pdf/hokoku/h29/h29_02.pdf)
- 4) コンピュータプログラムは「言語著作物 (literary works)」として、著作権法上保護される。  
田内幸治 米国におけるコンピュータプログラムの法的保護  
<http://www.tokugikon.jp/gikonshi/268/268kiko2.pdf>
- 5) 日本特許庁, マイクロコンピュータ応用技術に関する発明についての運用指針, 1982年

- 6) 岡村久道 オープンソース/フリーソフトウェア・ライセンスの法的分析  
<https://www.law.co.jp/okamura/copyleft/OssOkm.pdf>
- 7) 2008年, Cisco Systems, Inc.が自社のソフトウェア製品にGPLに基づきライセンスされるOSSを組み込んで販売しているにもかかわらずソースコード開示を怠ったとして, フリーソフトウェア財団が, Software Freedom Law Centerを代理人として同社を提訴。  
Free Software Foundation, Free Software Foundation Files Suit Against Cisco For GPL Violations  
<https://www.fsf.org/news/2008-12-cisco-suit>  
2009年, Cisco SystemsによるGPL遵守・和解金支払いを条件に和解。  
Free Software Foundation, FSF Settles Suit Against Cisco  
<https://www.fsf.org/news/2009-05-cisco-settlement.html>
- 8) 2007年, Microsoftが, Linuxを含むOSSが自社特許235件を侵害しているとして, 当該OSSを提供しているベンダーらに損害賠償請求を行っていく考えを発表。  
CNN Money, Microsoft takes on the free world  
[https://money.cnn.com/magazines/fortune/fortune\\_archive/2007/05/28/100033867/index.htm?postversion=2007051409](https://money.cnn.com/magazines/fortune/fortune_archive/2007/05/28/100033867/index.htm?postversion=2007051409)  
2009年, Microsoftが特許侵害に基づきTomTomを提訴。対象特許8件のうち3件がLinuxカーネルの実装に関連。  
Ars Technica, Microsoft suit over FAT patents could open OSS Pandora's Box  
<https://arstechnica.com/information-technology/2009/02/microsoft-sues-tomtom-over-fat-patents-in-linux-based-device/>
- 9) 企業がOSSを採用する理由としては, 上位から, 品質76%, 信頼性71%, セキュリティ70%, コスト50%。  
赤井誠, OSSのベストプラクティスを体感—Enterprise User's Meeting 2012レポート  
<https://gihyo.jp/news/report/2012/11/0602>
- 10) Microsoft Azure, Microsoft joins Open Invention Network to help protect Linux and open source

- <https://azure.microsoft.com/en-us/blog/microsoft-joins-open-invention-network-to-help-protect-linux-and-open-source/>
- 11) OSSが自社保有特許を侵害しているとして、PAEsであるSound View Innovationsが<sup>3</sup>、当該OSS利用者であるFacebook, Hulu, Twitter, Fidelity, AMC Networks, CBS Interactive, The CW Network, Walmart, Delta Airlines等を提訴。一部は特許無効判決により終結するも、多くはなお係属中。Bart Eppenauer, Sound View Escalates Patent Assault on Open Source Software with New Wave of Lawsuits  
<https://cloudipq.com/2019/04/30/sound-view-escalates-patent-assault-on-open-source-software-with-new-wave-of-lawsuits/>
  - 12) Steven J. Vaughan-Nichols, Open Invention Network comes to GNOME's aid in patent troll fight  
<https://www.zdnet.com/article/open-invention-network-comes-to-gnomes-aid-in-patent-troll-fight/>  
GNOME, Patent case against GNOME resolved  
<https://www.gnome.org/news/2020/05/patent-case-against-gnome-resolved/>
  - 13) 静的・動的リンクを含む。プロプライエタリのバイナリとの関係で見ると、静的リンクとはソフトウェアの実行ファイル (executable) として、GPLのバイナリとプロプライエタリのバイナリが同一ファイルになること、動的リンクとはプロプライエタリの実行ファイルとは別ファイル (GPLライブラリなど) ではあるものの実行中に当該別ファイルを呼び出すようにリンク付けすること  
一般財団法人ソフトウェア情報センター, 「IoT時代におけるOSSの利用と法的諸問題Q&A 集」  
[https://www.softic.or.jp/ossqa/all\\_180328](https://www.softic.or.jp/ossqa/all_180328)
  - 14) MongoDB, Server Side Public License (SSPL),  
<https://www.mongodb.com/licensing/server-side-public-license>
  - 15) MariaDB, Business Source License 1.1,  
<https://mariadb.com/bsl11/>
  - 16) Hippocratic License,  
<https://firstdonoharm.dev/>
  - 17) 996.ICU,  
<https://github.com/996icu/996.ICU>
  - 18) 経済産業省, 産業サイバーセキュリティ研究会,  
[https://www.meti.go.jp/shingikai/mono\\_info\\_service/sangyo\\_cyber/](https://www.meti.go.jp/shingikai/mono_info_service/sangyo_cyber/)
  - 19) National Telecommunications and Information Administration, "NTIA Software Component Transparency",  
<https://www.ntia.doc.gov/SoftwareTransparency>
  - 20) SPDX, The Software Package Data Exchange<sup>®</sup> (SPDX<sup>®</sup>)  
<https://spdx.org/>
  - 21) ISO, ISO/IEC 19770-2:2015 Information technology — IT asset management — Part 2: Software identification tag,  
<https://www.iso.org/standard/65666.html>
  - 22) REUSE software, "REUSE project",  
<https://reuse.software/>
  - 23) Steve Winslow, "Copyright Notices in Open Source Software Projects",  
<https://www.linuxfoundation.org/blog/2020/01/copyright-notices-in-open-source-software-projects/> (日本語版「オープンソース ソフトウェア プロジェクトにおける著作権表示」,  
<https://www.linuxfoundation.jp/blog/2020/01/copyright-notices-in-open-source-software-projects/>)
  - 24) UNITED NATIONS, About the Sustainable Development Goals  
<https://www.un.org/sustainabledevelopment/sustainable-development-goals/>
- (URLの参照日は全て2020年5月28日)
- (原稿受領日 2020年6月30日)