
BearingPoint®
MENOLD BEZLER
Open Source
Compliance im Transaktions-geschäft

BearingPoint®

MENOLD BEZLER

Inhalf

1 Einleitung

In diesem Whitepaper erfahren Sie, wie Sie die rechtlichen und technischen Risiken durch Open Source Software im Transaktionsgeschäft (M&A) erkennen und reduzieren können. Die Erstellung einer Software Bill of Materials (SBOM), die alle verwendeten Open Source Komponenten auflistet, und die Ermittlung eines technischen Risikoprofils zur Lizenzkonformität und Sicherheit werden erläutert. Außerdem erhalten Sie Einblicke, in die juristische Due Diligence-Prüfung und wie Sie die Lizenzbedingungen und Compliance-Anforderungen der Open Source Software erfüllen können.

Die Kombination aus rechtlicher und technischer Risikoanalyse geben dem Käufer Transparenz über die Open Source Risiken und Sicherheit, sodass der geplante Business Case auch umgesetzt werden kann.

2 Open Source und seine rechtliche Relevanz

Open-Source-Software (OSS) ist weit verbreitetet und ein integraler Bestandteil vieler Softwarelösungen. Es handelt sich hierbei um Software, deren Quellcode frei zugänglich ist und von jedermann bearbeitet werden darf. Das bedeutet aber keinesfalls, dass es sich auch um „rechtsfreie“ Software handelt. Vielmehr ist auch OSS urheberrechtlich geschützt. Die Erlaubnis, die OSS zu nutzen, erfolgt wie auch bei „herkömmlicher“ Software durch die Einräumung von Nutzungsrechten nach den Grundsätzen des Urheberrechtsgesetzes und auf Grundlage eines OSS-Lizenzvertrages. Hinter den OSS-Lizenzen verbergen sich insoweit stets auch vertragliche Verpflichtungen. Bei der Nichteinhaltung dieser Pflichten liegt daher eine Vertragsverletzung und häufig zugleich eine Urheberrechtsverletzung vor. So sehen eine Vielzahl von OSS-Lizenzen eine Einräumung der Nutzungsrechte nur unter der Bedingung vor, dass die Lizenzverpflichtungen eingehalten werden. Ein Verstoß gegen die Bestimmungen der Lizenz führt dann automatisch zu einem „Rückfall“ der Rechte an den Urheber. Weitere Nutzungshandlungen stellen dann eine Verletzung des Urheberrechts dar, was insbesondere Unterlassungs- und Schadensersatzansprüche zur Folge haben kann.

Anhand der Pflichten, die an den Lizenznehmer gestellt werden, wird üblicherweise zwischen Lizenzen mit und ohne eines sog. „Copyleft“-Effekts unterschieden:

- Copyleft-Lizenzen sehen regelmäßig eine Offenlegung des Quellcodes sowie eine Verpflichtung vor, auch etwaige Weiterentwicklungen der OSS unter dieselben Lizenzbedingungen stellen zu müssen. Man spricht daher auch von einem „viralen Effekt“. Hierdurch soll gewährleistet werden, dass die ursprüngliche „Freiheit“ der OSS beibehalten und an Weiterentwicklungen weitergegeben wird. Die auf Grundlage der OSS weiterentwickelte Software kann daher im Regelfall nicht wie herkömmliche Software ohne Quellcode-Offenlegung und gegen Lizenzgebühren vertrieben werden, da andernfalls gegen die OSS-Lizenzbedingungen verstoßen wird. Während die Wissenschaft den hierdurch entstehenden Wissensaustausch begrüßt, liegt es auf der Hand, dass dieser Effekt bei kommerziellen Softwareentwicklern unerwünscht ist, da eine Monetarisierung einschränken und Wettbewerbsnachteile bedeuten kann.

- OSS-Lizenzen ohne Copyleft-Klausel gestatten es dem Nutzer dagegen, Weiterentwicklungen auf Basis der OSS auch unter andere, eigene Lizenzbedingungen zu stellen. Dennoch sehen auch sie bestimmte Pflichten vor, wie etwa Weitergabe von Urhebervermerken oder Haftungsklauseln.

3 Sonderfall: KI-generierter Code

Mit der zunehmenden Integration von Künstlicher Intelligenz (KI) in die Softwareentwicklung entstehen neue Herausforderungen im Bereich der Open Source Compliance. KI-Systeme, die Code generieren sollen, werden auf großen Datensätzen trainiert, in denen regelmäßig auch eine Vielzahl von OSS-Code enthalten ist. Folglich ist es wahrscheinlich, dass die KI bei der Generierung von neuem Code Teile des ursprünglichen OSS-Codes reproduziert. Wie dargelegt verlangen Copyleft-Lizenzen aber, dass auf ihnen basierende neue Software unter dieselbe OSS-Lizenz gestellt und insbesondere der Quellcode offengelegt werden muss. Die Nutzung und Kommerzialisierung des KI-generierten Codes kann daher das Risiko von Verstößen gegen entsprechende OSS-Lizenzen und somit zugleich das Begehen von Lizenz- und Urheberrechtsverletzungen bergen. Dass der Programmierer hiervon regelmäßig keine Kenntnis hat, ist unbeachtlich.

4 Open Source Compliance im Transaktions-geschäft

Im Rahmen von Unternehmenstransaktionen gilt es, die Zielgesellschaft (sog. Target) vor einer Kaufentscheidung gründlich zu überprüfen (sog. Due Diligence-Prüfung). Besonders, wenn es sich bei dem Targetunternehmen um einen Softwarehersteller handelt, umfasst dies auch eine detaillierte Analyse der bestehenden Urheberrechte und etwaiger OSS-Lizenzen. Der Käufer hat insoweit ein wesentliches Interesse daran, zu erfahren, inwieweit OSS bei der Eigensoftware zum Einsatz zum kommt und ob das Target sich stets Lizenzkonform verhalten hat. Auf der Kehrseite lässt sich die Verhandlungsposition des Verkäufers stärken, wenn er nachweisen kann, dass auch bei OSS-Einsatz keine Risiken für den Käufer bestehen.

Ein zentrales Risiko aus Käufersicht besteht dabei darin, dass ein gutgläubiger Erwerb von (Nutzungs-) Rechten nicht möglich ist. Hat das Targetunternehmen gegen OSS-Lizenzbedingungen verstoßen und stehen ihm deshalb schon keine Nutzungsrechte an der OSS zu, wird diese Rechtsverletzung also auch an den Käufer „weitergegeben". Für den Käufer ist es daher wesentlich, zu prüfen, ob das Target die OSS-Lizenzbedingungen eingehalten hat, um selbst keine ungewollte Urheberrechtsverletzung zu begehen.

Die juristische Due Diligence-Prüfung ist im Hinblick auf OSS dabei im Wesentlichen auf die Überprüfung der verwendeten Lizenzen und maßgeblichen vertraglichen Regelungen beschränkt. Ob die Pflichten technisch korrekt umgesetzt werden, entzieht sich dagegen der juristischen Überprüfbarkeit. Für eine vollumfängliche Überprüfung der OSS-Compliance und Risikominimierung ist im Regelfall auch daher eine technische Überprüfung des Quellcodes erforderlich.

5 Open Source Software in Produkten und Systemen

Softwareentwicklung ist heute ohne Verwendung von Open Source Software nicht mehr vorstellbar. Diese kann auf verschiedenen Wegen in eine Codebasis gelangen. Der offensichtlichste ist die bewusste Verwendung von Open Source Komponenten durch das Entwicklungsteam. Für bestimmte Funktionalitäten werden Open Source Komponenten ausgewählt, die dann während des Build-Prozesses automatisiert eingebunden werden. Viele Open Source Komponenten benötigen selbst weitere Komponenten, um zu funktionieren, die dann ebenfalls automatisiert im Build-Prozess eingebunden werden. Dies geschieht indirekt, ohne Kenntnis des Entwicklungsteams. Je nach verwendeter Technologie können so hunderte oder sogar tausende Komponenten eingebunden werden.

Open Source Software kann aber auch durch direkte Integration in den Quellcode in ein Produkt/System gelangen. Solche Open Source Codefragmente, oft auch Codesnippets genannt, stammen entweder direkt aus Open Source Komponenten, die nur teilweise genutzt werden sollen, oder aus Portalen, wie z.B. StackOverflow, die Code zur Lösung spezifischer Problemstellungen bereitstellen, oder werden von Generativer KI generiert. Auch für diese Codesnippets gelten die jeweiligen Lizenzbestimmungen, die der ursprüngliche Autor festgelegt hat. Die besondere Problematik ist hier, dass diese Lizenzbestimmungen vielfach nicht offensichtlich sind, oder vom Entwicklungsteam bei der Integration des Codesnippets nicht übernommen werden.

Werden Teile eines Produktes oder Systems von externen Partnern entwickelt, kann auf diesem Weg zusätzliche Open Source Software unbemerkt vom Entwicklungsteam in ein Produkt oder System gelangen. Auftraggeber fordern deshalb von ihren Lieferenten oft detaillierte Informationen über die eingesetzte Open Source Software.

6 Software Bill Of Materials (SBOM)

Bevor eine Überprüfung der Open Source Compliance durchgeführt werden kann, muss zunächst eine Bestandsaufnahme der verwendeten Open Source Software erfolgen. Dazu wird typischerweise für jedes Produkt oder System eine Software Bill Of Materials (SBOM) erstellt, eine vollständige Inventarliste, mit allen relevanten Informationen über die eingesetzte Open Source Software. Dazu gehört neben dem Namen der Komponente auch deren Version und alle für die Komponente relevanten Lizenzen, sowie Urheberinformationen, wie z.B. Copyright-Vermerke. Oft enthält die SBOM zusätzlich Informationen über öffentlich bekannte Sicherheitslücken in der jeweiligen Komponente.

SBOMs manuell zu erstellen und zu pflegen, ist schon allein wegen der der großen Anzahl der direkt oder indirekt verwendeten Open Source Komponenten nicht praktikabel. Daher werden oft Werkzeuge eingesetzt, die im Build-Prozess integriert SBOMs automatisiert erstellen. Diese stoßen jedoch schnell an ihre Grenzen, da sie Codesnippets oder modifizierte Open Source Komponenten nicht erfassen können. Automatisiert erstellte SBOMs sind daher regelmäßig nicht vollständig und sind für die Beurteilung der Compliance-Situation nur bedingt geeignet.

7 Forensische Codeanalyse

Voraussetzung für eine juristische Due Diligence-Prüfung im Transaktionsgeschäft ist das Vorliegen einer kompletten SBOM, die sämtliche verwendete Open Source Software erfasst, unabhängig davon, wie sie in das Produkt/System gelangt ist. Werden SBOMs vom Targetunternehmen zur Verfügung gestellt, ist oft nicht transparent, wie diese erstellt wurden. Eine Beurteilung der Vollständigkeit und Korrektheit ist oft nicht zweifelsfrei möglich, weshalb eine juristische Due Diligence-Prüfung auf dieser Basis nicht verlässlich durchgeführt werden kann.

Aus diesem Grund wird im Rahmen einer Technischen Due Diligence zunächst eine vollständige SBOM neu erstellt. Dazu muss das Targetunternehmen den kompletten Quellcode der Produkte/Systeme zur Verfügung stellen. Dazu zählen neben dem selbstentwickelten Code auch sämtliche Bibliotheken/Komponenten, die während des Build-Prozesses direkt oder indirekt eingebunden werden. Typischerweise wird das Targetunternehmen die Codebasis dem Käufer jedoch nicht vor Abschluss der Transaktion aushändigen. Deshalb wird die Codeanalyse meist von einem neutralen Dritten, z.B. einem Dienstleister, durchgeführt.

Bei der forensischen Codeanalyse wird die Codebasis zunächst mit einem speziellen Scanning-Werkzeug analysiert. Dieses ist in der Lage den Quellcode vom Targetunternehmen mit einer umfassenden Datenbank aus öffentlich verfügbarem Open Source Quellcode zu vergleichen und Übereinstimmungen zu ermitteln. Dies ist in etwa vergleichbar mit der Plagiatserkennung bei wissenschaftlichen Texten.

Die vom Scanning-Werkzeug ermittelten Übereinstimmungen werden dann von erfahrenen Codeanalyse-Spezialisten ausgewertet. Für jede Übereinstimmung wird ermittelt, aus welcher Open Source Komponente das fragliche Codestück ursprünglich stammt und deren Lizenz(en) bestimmt. Das Resultat ist eine vollständige SBOM, die auch Codesnippets erfasst.

8 Technische Risikoanalyse

Bevor die juristische Due Diligence-Prüfung beginnen kann, ist noch eine technische Risikoanalyse erforderlich. Copyleft-Lizenzen wirken sich z.B. unterschiedlich aus, je nachdem wie die Komponente konkret integriert und genutzt wird. Daher können je nach Nutzung unterschiedliche rechtliche Risiken entstehen. Bei der technischen Risikoanalyse werden die in der SBOM verzeichneten Lizenzen zunächst bzgl. ihres Copyleft-Effekts kategorisiert. Hier unterscheidet man typischerweise zwei Kategorien. Bei starken Copyleft-Lizenzen greift der Copyleft-Effekt nur dann nicht, wenn die Open Source Komponente als eigenständiges Programm genutzt wird. Schwache Copyleft Lizenzen erlauben zusätzlich z.B. die Nutzung als Bibliothek, die dynamisch oder statisch eingebunden wird.

Nach der Klassifizierung wird für jede Open Source Komponente mit Copyleft Lizenz anhand der Produkt-/Systemarchitektur die Art der Integration ermittelt und daraus abgeleitet, wie weit der Copyleft Effekt aus technischer Sicht reicht. Hierzu muss auch festgestellt werden, ob die Komponente unverändert genutzt wird, oder vom Targetunternehmen modifiziert wurde.

Für jede Open Source Komponente in der SBOM wird so ein technisches Risikoprofil erstellt, welches als Grundlage für die folgende juristische Due Diligence-Prüfung dient.

Kontakt

BearingPoint

www.bearingpoint.services/foss

Claus-Peter Wiedemann
Director
E-Mail: claus-peter.wiedemann@bearingpoint.com

Menold Bezler

www.menoldbezler.de

Jessica Hawighorst
Rechtsanwältin
E-Mail: jessica.hawighorst@menoldbezler.de

At Menold Bezler around 350 employees, including more than 140 lawyers, tax advisors, auditors, and business advisors work together under one roof to provide a broad range of innovative and individual consulting services. This interdisciplinary advisory approach guarantees optimal results for medium-sized companies, entrepreneurs, and the public sector. We find pragmatic solutions for their complex challenges and provide concrete recommendations for action.

For more information, visit our website at www.menoldbezler.com.

MENOLD
BEZLER

BearingPoint®

About BearingPoint

BearingPoint is an independent management and technology consultancy with European roots and a global reach. The company operates in three business units: Consulting, Products, and Capital. Consulting covers the advisory business with a clear focus on selected business areas. Products provides IP-driven digital assets and managed services for business-critical processes. Capital delivers M&A and transaction services.

BearingPoint's clients include many of the world's leading companies and government organizations. The firm has a global consulting network with more than 10,000 people and supports clients in over 70 countries, engaging with them to achieve measurable and sustainable success.

For more information, visit our website www.bearingpoint.com