Vergabekriterien für eine nachhaltige Beschaffung von Open Source Software

Ein Positionspapier der Open Source Business Alliance
Bundesverband für digitale Souveränität e.V.

11. Februar 2025

Autoren:

Working Group Beschaffung der Open Source Business Alliance (vertreten durch die beiden Sprecher der Working Group Birgit Becker und Claus Wickinghoff)

Disclaimer

Das vorliegende Papier gibt der öffentlichen Verwaltung Hinweise an die Hand, wie Open Source Software nachhaltig beschafft werden kann, so dass die Verwaltung qualitative Angebote erkennen und auswählen kann. Es handelt sich hierbei um eine Hilfestellung und nicht um eine Rechtsberatung.

Inhaltsverzeichnis

Open Source Software in der öffentlichen Verwaltung

Die öffentliche Verwaltung in Deutschland strebt digitale Souveränität an, also die Fähigkeit, die eigenen IT-Systeme unabhängig überprüfen, gestalten und austauschen zu können. Zur Erreichung dieses Ziels wird vermehrt Open Source Software eingesetzt.

Open Source Software zeichnet sich durch den offenen und kooperativen Ansatz aus, sowie durch die Freiheiten, welche die Softwarelizenzen gewähren. Daraus ergeben sich besondere Entwicklungs- und Vertriebsmodelle für Open Source Software. Dies kann eine Herausforderung für die öffentliche Verwaltung darstellen, die sich insbesondere in ihren Beschaffungs- und Vergabeprozessen jahrzehntelang ausschließlich auf proprietäre Software eingestellt hat.

Die Geschäftsmodelle hinter Open Source funktionieren grundsätzlich anders als bei proprietärer Software: Bei proprietärer Software partizipiert der Hersteller an jeder verkauften Lizenz, auch wenn die Vergabe über einen Drittanbieter erfolgt ist. Bei Open Source Software werden üblicherweise statt der Softwarelizenz ergänzende Dienstleistungen angeboten. Dies eröffnet Drittanbietern die Möglichkeit, über Dumpingangebote den eigentlichen Software-Hersteller bei einer Vergabe auszustechen. In dieser Konstellation fließt kein Geld an den Open-Source-Hersteller und dieser kann in der Folge nicht ausreichend in die Weiterentwicklung und Pflege der Open Source Software investieren.

Ein Argument für den Einsatz von Open Source Software in der Verwaltungsdigitalisierung ist u.a. das Nachnutzungspotential, dass also eine Behörde die bereits entwickelte und bezahlte Software einer anderen Behörde weiterverwenden kann, wenn beispielsweise der entsprechende Quellcode auf einer öffentlich zugänglichen Plattform wie openCode veröffentlicht wird.[1] Dieses Versprechen der Nachnutzung erfüllt sich aber nicht, wenn keine ausreichenden Mittel in die kontinuierliche Weiterentwicklung und Pflege der Software fließen und es für die Hersteller nicht mehr wirtschaftlich ist, überhaupt Open Source Software zu entwickeln. Das ist insbesondere für die Auftraggeberseite nachteilig, weil dann weniger qualitativ hochwertige und aktuell gehaltene Open Source Software am Markt zur Verfügung steht.

Dadurch wird auch die IT-Sicherheit der entsprechenden Open-Source-Lösung in der öffentlichen Verwaltung gefährdet oder das Scheitern von IT-Projekten in Kauf genommen. Die entstehenden Probleme werden dann oft verallgemeinernd auf den Umstand geschoben, dass Open Source Software eingesetzt wurde. Das beschädigt den allgemeinen Ruf von Open Source Software.

Die Herausforderung für die öffentliche Verwaltung besteht also darin, die richtigen Anforderungen und Vergabekriterien zu entwickeln, damit sie aus den Anbietern zuverlässig diejenigen auswählen kann, die nachhaltig sichere und qualitativ hochwertige Software und Dienstleistungen anbieten.

Schon heute gibt es die Möglichkeit, in einer Ausschreibung neben dem Preis auch andere Auswahlkriterien mit in die Bewertung eines Angebotes einzubeziehen. In der Unterlage für Ausschreibung und Bewertung von IT-Leistungen (UfAB 2018)[2] werden in Abschnitt 5 (Seite 71 ff.) zusätzliche

1 Die Bundesregierung hat das Zentrum für die Digitale Souveränität der öffentlichen Verwaltung (ZenDiS) ins Leben gerufen, um den verstärkten Einsatz von Open Source Software in der Verwaltung voranzutreiben. Hierfür setzt das ZenDiS auch eigene Projekte um wie beispielsweise openCode. Dies ist eine zentrale Plattform zum Austausch von Open Source Software speziell für die öffentliche Verwaltung.

2 https://www.cio.bund.de/SharedDocs/downloads/Webs/CIO/DE/digitale-loesungen/it-beschaffung/ufab/ufab2018.html

Kriterien für die Beschaffungskonzeption aufgeführt: Aspekte der Nachhaltigkeit, Einbeziehung von Know-how der Anbietermärkte, Festlegung von Eignungskriterien und Zuschlagskriterien.

Das vorliegende Papier gibt der öffentlichen Verwaltung Hinweise an die Hand, wie Open Source Software nachhaltig beschafft werden kann, so dass die Verwaltung qualitative Angebote erkennen und auswählen kann und gleichzeitig das Open-Source-Ökosystem langfristig gestärkt und das Ziel der digitalen Souveränität erreicht wird.

Wie funktioniert Open Source Software?

Open Source Software ist grundsätzlich Software wie jede andere auch. So gibt es etliche Open-Source-Lösungen, die von professionellen Unternehmen entwickelt und mit professionellem Support sowie Gewährleistung und Haftung angeboten werden.

Der Unterschied zwischen Open Source Software und anderer Software liegt in der Lizenz. Während Nicht-Open-Source-Software (oft auch als proprietäre oder Closed-Source-Software bezeichnet) die Verwendung der Software wesentlich einschränkt – meist auf das Starten/Abspielen und das Erstellen einer Sicherungskopie – räumen die Lizenzen von Open Source Software umfangreiche Nutzungsrechte ein.

Open Source Software darf grundsätzlich beliebig eingesetzt, eingesehen, verbessert und sogar weitergegeben werden. Damit dies funktioniert, wird der Quellcode (Source Code) mitgegeben. Viele Open-Source-Lizenzen sind mit einfach erfüllbaren Lizenzbedingungen versehen, etwa dass ein Haftungsausschluss beigefügt werden muss. Manche dieser Lizenzen erfordern die Weitergabe des (modifizierten) Quelltextes unter derselben Lizenz an den Empfänger, wenn abgeleitete Werke erstellt werden – dies ist das sogenannte „Copyleft“. Die Europäische Union hat zur Unterstützung von Open Source Software in der öffentlichen Verwaltung mit der EUPL eine eigene Lizenz mit Copyleft-Character entwickelt, die in allen EU-Sprachen zur Verfügung steht.[3]

Die wesentliche Idee von Open-Source-Lizenzen ist, dass Anwender in die Lage versetzt werden, eine Software auf die eigenen Bedürfnisse anzupassen und auch bei Änderungen der Anforderungen weiter mit der Software arbeiten zu können. Aus Sicherheitsperspektive ergibt sich bei Open Source Software die Möglichkeit, die Software auf Schwachstellen und Hintertüren zu prüfen. Mit Blick auf Nachhaltigkeit legt der Programmcode natürlich auch das genutzte Datenformat offen, so dass vorhandene Daten importiert oder auch exportiert und in anderen Systemen weiter genutzt werden können.

Was sind die Vorteile von Open Source Software?

Open Source Software fördert insgesamt unternehmens- und grenzüberschreitende Zusammenarbeit sowie Innovation und Wettbewerb. In den letzten 30 Jahren sind auf diese Weise viele relevante Softwareprojekte entstanden:

- die Basistechnologien für das Internet, ohne die die Digitalisierung gänzlich anders verlaufen wäre
- der Linux-Kernel, der heute in Android-Smartphones und vielen Geräten wie Routern u.ä. steckt

3 https://eupl.eu/

- Kubernetes, das die Basis für die moderne Container-Orchestrierung darstellt
- Datenbank-Managementsysteme wie MariaDB oder PostgreSQL
- Open Source AI Frameworks
- Webbrowser wie Firefox
- Office-Programme wie LibreOffice
- Open-Source-Videokonferenzsysteme wie Jitsi oder BigBlueButton

Der Wunsch, mit öffentlichen Geldern nachhaltige Lösungen zu unterstützen oder sogar zu schaffen, lässt sich mit Open Source Software ideal umsetzen. Da der Quellcode offenliegt und frei genutzt werden darf, können Behörden die entwickelte Open Source Software untereinander frei teilen. Und es können im Laufe der Zeit weitere Funktionalitäten ergänzt werden und jede Institution kann die Software auf ihre spezifischen Bedürfnisse anpassen. Der Slogan "Public Money, Public Code" bringt es auf den Punkt: Öffentlich finanzierte Software sollte unter einer Open-Source-Lizenz veröffentlicht werden, damit die Ergebnisse der Allgemeinheit wieder zur Verfügung gestellt werden können. In diesem Sinne trägt die Entwicklung und Beschaffung von Open Source Software auch zu einer verantwortungsbewussten und nachhaltigen Verwendung von öffentlichen Geldern bei.

Wer ist die Community?

Im Zusammenhang mit Open Source Software wird oft der Begriff der „Community“ verwendet. Bei jeder Software gibt es eine Anwendergemeinschaft, welche die Software nutzt. Dies ist im Englischen die „User Community“. Bei Open Source Software gibt es durch die freie Lizenz jedoch für jeden die Möglichkeit der Mitgestaltung, so dass aus dieser Anwendergemeinschaft heraus auch Beiträge (engl. „contributions“) zur Software selbst erfolgen, die oft auch langfristig aktiv gepflegt werden.

Die Bandbreite der Mitglieder einer Entwickler-Community reicht von interessierten Anwenderinnen und Anwendern, die Software möglicherweise in ihrer Freizeit entwickeln, über Programmierinnen und Programmierer, die im Rahmen ihrer Arbeitszeit an einer Software arbeiten, bis hin zu Unternehmen, deren Business-Modell vollständig auf der Unterstützung und Entwicklung von Open Source Software beruht.

Eine Open-Source-Community besteht also aus unterschiedlichsten Personen und Organisationen, die die entsprechende Software nutzen und sich ggf. auch an der Pflege und Weiterentwicklung beteiligen.

Die meisten Software-Produkte bestehen aus verschiedenen Komponenten, hinter denen jeweils eine eigene Entwickler-Community stecken kann. Wenn professionelle Hersteller solche Komponenten für ihre Software-Produkte einsetzen, beauftragen sie häufig eigene Mitarbeiterinnen und Mitarbeiter damit, sich als Teil der jeweiligen Community an der Entwicklung zu beteiligen.

Beschaffung von Open Source Software

Das Open-Source-Ökosystem besteht also aus vielfältigen kommerziellen und anderen (z.B. ehrenamtlichen) Akteuren. Die Beschaffung von Software erfolgt typischerweise bei kommerziellen Anbietern, denn nur diese sind rechtlich-organisatorisch in der Lage, die vergaberechtlichen Anforderungen zu erfüllen.

Um wirtschaftlich und finanziell leistungsfähig zu sein (vgl. § 45 Vergabeverordnung über die Vergabe öffentlicher Aufträge), muss der Bewerber oder Bieter Geld verdienen. Bei Open Source Software verdient der Anbieter sein Geld üblicherweise nicht über den Verkauf von Lizenzgebühren. Die öffentliche Hand beschafft bei Open Source Software stattdessen im Regelfall Dienstleistungen, also faktisch Support, Betrieb, Individualisierung oder Weiterentwicklung. Die Qualität der Leistungen ist daher abhängig davon, dass der Anbieter geeignetes Personal in ausreichender Anzahl für diese Aufgaben zur Verfügung stellt.

Zugleich benötigt Open Source Software, wie jede andere Software auch, laufende Pflege und Aktualisierung. Dies gilt sowohl für Lösungen und Anpassungen, die von einem kommerziellen Unternehmen individuell für einen Auftraggeber entwickelt wurden, als auch für Standardsoftware.[4] Diese Aufwände müssen beim Angebotspreis berücksichtigt werden.

Werden Leistungen der öffentlichen Hand allein oder in erster Linie nach Preis vergeben, benachteiligt dies Bewerber/Bieter, die ein nachhaltiges Geschäftsmodell verfolgen. Diese investieren in laufende Pflege und Aktualisierung der Software und sichern – wie es eigentlich übliche und gelebte Praxis ist – die Lieferkette ab, indem sie Supportverträge mit den Herstellern einzelner Softwarekomponenten abschließen.

Unseriöse Bewerber/Bieter können in mehrerer Hinsicht den Preis für ihr konkretes Angebot drücken, um den Zuschlag zu erhalten:

- Der Anbieter plant z.B. nicht ausreichend Ressourcen ein für eigene Entwicklungen oder Pflegeleistungen, sondern verlässt sich in zu großem Maße auf das, was von anderen Herstellern oder ehrenamtlichen Entwicklern frei zur Verfügung gestellt wird.
- Der Anbieter spart Ressourcen, indem er keine Zeit darauf verwendet, die von ihm vorgenommenen Entwicklungen oder Anpassungen der Allgemeinheit wieder frei zur Verfügung zu stellen.
- Möglicherweise verzichtet der Anbieter ganz auf eigene Entwickler und wird bei Problemen mit der Software nur Hilfe über öffentlich verfügbare Supportkanäle (z.B. Onlineforen) des eigentlichen Herstellers suchen. Durch den Verzicht auf Supportverträge mit den Herstellern einzelner integrierter Komponenten kann der Preis des Angebots gesenkt werden. Dies geschieht allerdings zu Lasten der Lieferkette, die die Qualität und Sicherheit der Software absichern sollte. Erhalten Unternehmen mit derartigen Geschäftspraktiken den Zuschlag im Vergabeverfahren, fließen nicht ausreichend Investitionen zurück in die Pflege und Weiterentwicklung der entsprechenden Software. Dies führt unmittelbar zu Qualitätsproblemen bei der von der öffentlichen Hand beschafften Leistung und mittelbar dazu, dass der Code,

4 Standardsoftware beschreibt Softwareprogramme, Programm-Module, Tools etc., die für die Bedürfnisse einer Mehrzahl von Kunden am Markt und nicht speziell vom Auftragnehmer für den Auftraggeber entwickelt wurden.

welcher der Allgemeinheit zur Verfügung steht, nicht in dem Maße verbessert und aktualisiert wird, wie es dem Umfang seiner Nutzung entspricht.

Dies wirkt sich zugleich nachteilig auf das Open-Source-Ökosystem insgesamt und auf die zukünftige Möglichkeit der Beschaffung von Open Source Software durch die öffentliche Hand aus.

Beschaffung - aber richtig!

Dass die Bedenken gegenüber unseriösen Anbietern nicht aus der Luft gegriffen sind, zeigen einige Beispiele aus einer Befragung, die die Open Source Business Alliance im Frühjahr 2024 unter ihren Mitgliedern durchgeführt hat. Hier werden auch die Risiken für die öffentliche Hand bei der Beschaffung sichtbar:

Die Erfahrung eines Herstellers von Open Source Software:

> *"Ein Bundesland hat sich entschieden, für alle Schulen unsere Software auszuschreiben, das haben Unternehmen x und Unternehmen y gewonnen, beide haben vorher noch nie etwas mit unserer Software gemacht.*
>
> *Wir haben uns auch an der Ausschreibung beteiligt, das Bundesland hatte sehr hohe Anforderungen mit harten Kriterien zu Performance und anderen Sachen. Hierzu haben wir uns intern sehr viele Gedanken gemacht, ob und wie man das erfüllen kann.*
>
> *Unternehmen x und Unternehmen y haben die Ausschreibung mit Dumpingpreisen gewonnen und sich im Nachhinein bei uns gemeldet, ob wir das Consulting für Skalierung und Performanceprobleme machen können. Wir haben das abgelehnt und auf unsere normale Subskription verwiesen, für die aber im Projekt kein Budget mehr vorhanden war."*

Fazit: Der Hersteller hat ein realistisch kalkuliertes Angebot vorgelegt, das auch Skalierungsprobleme und Performance berücksichtigt. Er wurde allerdings von einem konkurrierenden Anbieter mit einem deutlich reduzierten Angebot unterboten. Für den Auftraggeber erscheint es so, als ob die Software seinen Anforderungen nicht genügt, dabei liegt das Problem beim Auftragnehmer: Er hat weder die erforderliche Expertise selbst mitgebracht, noch diese über einen Supportvertrag mit dem eigentlichen Software-Hersteller eingeholt.

Bei proprietärer Software bezieht der Anbieter immer Lizenzen vom Hersteller, so dass der Hersteller über diese Lizenzkette immer finanziell beteiligt wird. Bei Open Source Software ist dies nicht zwangsweise gegeben und im konkreten Beispiel gab es kein Vertragsverhältnis mit dem Open-Source-Hersteller. Dieser hat daher auch kaum Motivation, unentgeltliche Unterstützung bei dem Projekt zu leisten. Die Folge: Der Auftraggeber fühlt sich alleine gelassen.

Die Erfahrung eines weiteren Herstellers von Open Source Software:

> *"Vor anderthalb Jahren hat mir der CIO eines Bundeslandes zum Deal für die Videoarbeitsplätze mit unserer Software gratuliert und geschwärmt, dass er ein großer Open-Source-Verfechter sei. Ich wusste zuerst nicht, wovon er sprach und dann stellte sich heraus, das das Unternehmen z mit unserer Software eine Ausschreibung gewonnen hat.*
>
> *Letztlich haben sie einen Dritten damit beauftragt, unsere Software zu forken und mit entsprechendem Wissen um die Anforderungen in dem Bundesland Anpassungen zu Barrierefreiheit und anderem vorzunehmen. Der CIO hat gedacht, das wäre super für Open Source -*

aber das, was da für viel Geld für das Bundesland entwickelt wurde, ist nun alles Closed Source.

Das Bizarre ist, dass ein Bundesministerium jetzt erneut viel Geld in Barrierefreiheit investiert, obwohl Unternehmen z diese gleichen Funktionen ja bereits mit Steuergeldern für das Bundesland entwickelt hat. Aber diese Weiterentwicklungen sind eben nicht Open Source."

Fazit: Der Anbieter hat seine Anpassungen in diesem Beispiel in einem Fork, also einer Abspaltung des eigentlichen Open Source Codes durchgeführt. Gleichzeitig hat der Anbieter diesen Fork nicht als Open Source für die allgemeine Weiterverwendung freigegeben.[5]

Für die Verwaltung ist es gerade bei der Beauftragung von Weiterentwicklungen bzw. Anpassungen von Open Source Software wichtig, vorab festzulegen, dass die Arbeitsergebnisse wieder unter einer Open-Source-Lizenz veröffentlicht werden und möglichst in das eigentliche Open-Source-Projekt zurückgespielt werden. Grundsätzlich ist auch der Übergabeort der Software festzulegen, z.B. openCode. Sonst bewegt sich die Entwicklung in eine technologische Sackgasse, da Weiterentwicklungen des ursprünglichen Codes bei jedem Update-Zyklus immer wieder mit finanziellem Aufwand in den Fork eingearbeitet werden müssen. Hinzu kommt die Abhängigkeit von diesem einzelnen Anbieter, der als einziger über den abgespaltenen Quellcode verfügt. Das Ziel der Anbieterunabhängigkeit ist damit nicht erreicht und sollte der Anbieter z.B. seinen Geschäftsbetrieb einstellen, ist die modifizierte Software nicht weiter sicher nutzbar. Auch die Nachnutzung der angepassten Software durch andere Institutionen ist nicht ohne weiteres möglich.

Die Erfahrung eines weiteren Herstellers von Open Source Software:

“Manche Leute überschwemmen uns mit Supportanfragen und behaupten, die Software funktioniere nicht. Was sie nicht sagen, ist, dass sie selbst einen Supportvertrag mit einem Kunden erfüllen müssen. Ihnen selbst fehlt aber die Expertise hierfür und uns wollen sie nicht für die Unterstützung bezahlen. Solche Fälle sind für uns schwer zu trennen von kleinen Anwendern, die wir gerne auch kostenfrei unterstützen als Teil unserer Community-Arbeit.”

Fazit: Die meisten Unternehmen im Open-Source-Umfeld bieten durchaus in einem gewissen Rahmen kostenfreie Unterstützung für Nutzer ohne Supportvertrag. Dies geschieht jedoch freiwillig und meist auf offenen Kanälen wie in Foren und öffentlichen Ticketsystemen. Dieses Angebot richtet sich nicht an kommerzielle Nutzer der Software, sondern eher an beispielsweise Einzelpersonen oder kleine Vereine. Deshalb sind hier natürlich auch keinerlei Reaktionszeiten des Herstellers festgelegt.

Wer dieses Angebot ausnutzt und ohne Supportvertrag mit dem Hersteller eigenen, kostenpflichtigen Support für einen Kunden anbietet und diesen über öffentliche Kanäle des Herstellers abbildet, handelt unsozial und dem Auftraggeber gegenüber unverantwortlich. Unsozial, weil der Hersteller bzw. letztendlich dessen Kunden die Mitarbeiterinnen und Mitarbeiter im Support bezahlen und diesen zusätzliche und unvergütete Arbeit aufgelastet wird, wodurch für die eigentliche

5 Bei der Erstellung eines Forks, also einer Abspaltung, sind einige Dinge zu beachten. Hierbei entsteht eine völlig neue Codebasis, die eigenständig gepflegt werden muss. Ob ein Fork sich langfristig etablieren kann, hängt von der Anzahl der Nutzenden, der generellen Nachnutzung und der darum entstehenden Community ab. Beispiele für erfolgreiche Forks sind u.a. LibreOffice und Nextcloud. Ein Fork bietet die Möglichkeit, eigene Änderungen und Wünsche in der Software umzusetzen, die aus unterschiedlichen Gründen im Kernprojekt nicht realisiert werden. Dies ist nur mit Open Source möglich.

Community-Unterstützung weniger Ressourcen verfügbar sind. Unverantwortlich, weil in einem Vertrag üblicherweise Service-Level-Agreements (SLAs) vereinbart werden, die aber so nicht umsetzbar sind.

Kriterien für die nachhaltige Beschaffung

Die zuvor beschriebenen Negativbeispiele illustrieren, wie das Open-Source-Ökosystem mittel- und langfristig beschädigt werden kann, wenn die öffentliche Beschaffung die Besonderheiten von Open Source Software nicht berücksichtigt. Das wird für den öffentlichen Auftraggeber zum Problem, weil dann weniger qualitativ hochwertige und aktuell gehaltene Open Source Software am Markt zur Verfügung steht und damit auch die Möglichkeit der Nachnutzung von Open Source Software eingeschränkt wird.

Die Grundidee der Open-Source-Lizenzen ist es, dass der Anwender einer Software – in diesem Fall die öffentliche Hand – in die Lage versetzt wird, selbst Anpassungen vorzunehmen, evtl. auf Basis der Software eigene Weiterentwicklungen zu erstellen und diese ggf. weiter zu verbreiten. Im Falle der öffentlichen Verwaltung werden zumeist Dienstleister beauftragt, diese Weiterentwicklungen oder Anpassungen vorzunehmen.

Dabei sollte die öffentliche Verwaltung darauf achten, dass die Anpassungen und Arbeitsergebnisse des Anbieters wieder veröffentlicht werden und an das entsprechende Open-Source-Projekt zurückfließen – andernfalls entsteht hier (ungewollt) wieder eine Abhängigkeitssituation von einem einzelnen Anbieter.

Für eine nachhaltige Nutzung von Open Source Software ist es daher wichtig, dass diese Software nicht nur konsumiert, sondern vom Anwender auch etwas in das Ökosystem "investiert" wird. Hierfür gibt es viele verschiedene Möglichkeiten: Von eigenen regelmäßigen Codebeiträgen über die Erstellung von Dokumentationen bis zur Bezahlung von Entwicklern u.v.m. Gerade bei der Beschaffung von Open Source Standard-Software, die von einem einzelnen Unternehmen gepflegt wird, sollte dieses Unternehmen im ausgeschriebenen Projekt beteiligt werden. So stehen auch weiterhin die Mittel zur Verfügung, die Entwicklung und Pflege der Software zu finanzieren und auch regulatorische Anforderungen wie etwa aus dem Cyber Resilience Act (CRA) umzusetzen. Dies kommt letztlich auch dem Auftraggeber wieder zugute.

Für die Beschaffung von Open Source Standard-Software eignet sich die Methode der reinen Preiswertung in der Regel nicht, weil die Pflege und Weiterentwicklung der Software nicht automatisch in jedem Angebot inkludiert ist.

In der Unterlage für Ausschreibung und Bewertung von IT-Leistungen (UfAB) werden in Abschnitt F 4.2.1 Möglichkeiten aufgezeigt, nicht nur den Preis als Entscheidungskriterium für die Vergabe heranzuziehen. In Abschnitt B 5.4.1 der UfAB wird konkret die Nachhaltigkeit als Aspekt benannt. Dort heißt es:

> *“Nachhaltigkeit bei Ausführungsbedingungen nach § 128 Abs. 2 GWB können auftragsbezogene Ausführungsbedingungen insbesondere wirtschaftliche, innovationsbezogene, umweltbezogene, soziale oder beschäftigungspolitische Belange umfassen“.*

Anbieter, die die kontinuierliche Weiterentwicklung und Pflege von Open Source Software gewährleisten, fördern insbesondere wirtschaftliche und innovationsbezogene Belange, die dem öffentlichen Auftraggeber (z.B. im Rahmen der Nachnutzung) zugute kommen.

Im Folgenden werden Kriterien genannt, die in Ausschreibungen für Open Source Software benutzt werden sollten, damit eine nachhaltige Beschaffung gewährleistet wird und ein kompetenter Dienstleister ermittelt werden kann. Im Anhang sind konkrete Textbausteine zu diesen Kriterien zu finden. Selbstverständlich ergeben fachliche Anforderungen weitere Kriterien, die hier aber nicht im Fokus stehen.

1. Beziehung zum Software-Hersteller / der Community

Der öffentliche Auftraggeber profitiert auf unterschiedliche Weise davon, wenn der Dienstleister eine Beziehung zum Software-Hersteller bzw. der entsprechenden Open-Source-Community nachweisen kann.

Anbieter mit einer engen Beziehung zum Hersteller bzw. der Community spielen eine aktive Rolle in der Weiterentwicklung und Wartung der Software. Sie können spezifische Anforderungen und Verbesserungsvorschläge direkt einbringen, dies ermöglicht eine Anpassung der Software an die besonderen Bedürfnisse des Auftraggebers. Der Auftraggeber erhält somit die gewünschte Funktionalität und verbesserte Benutzerfreundlichkeit. Dies minimiert zudem das Risiko von Inkompatibilitäten und fördert grundsätzlich eine stabile IT-Infrastruktur.

Eine enge Zusammenarbeit des Auftragnehmers mit dem Hersteller bzw. der Community der genutzten Open Source Software gewährleistet direkten Zugang zu technischem Support und den neuesten Software-Updates. Dies ist entscheidend für die schnelle Behebung von Problemen und verbessert die Betriebssicherheit der Softwarelösung. Eine schnelle Reaktionsfähigkeit ist besonders in kritischen Anwendungsfällen von unschätzbarer Bedeutung.

Der Auftraggeber profitiert von frühzeitigem Wissen über potenzielle Sicherheitslücken und kann präventive Maßnahmen ergreifen, um das Risiko von Sicherheitsverletzungen zu minimieren. Der Cyber Resilience Act (CRA) widmet sich umfassend dieser Thematik und fordert die Absicherung der Lieferkette und die Aufschlüsselung der im Produkt enthaltenen Softwarekomponenten über eine Software Bill Of Materials (SBOM). Eine enge Kooperation mit dem Software-Hersteller bzw. der Community gewährleistet die für die öffentliche Hand notwendige hohe Sicherheitsqualität.

Die Beziehung zum Hersteller bzw. der Community ist auch relevant dafür, wie lange Entwicklung, Pflege und Support für die genutzte Software verfügbar sind. Je länger eine Open Source Software in der Verwaltung genutzt werden kann, desto wirtschaftlicher ist auch die Verwendung der eingesetzten Mittel. Die langfristige Verfügbarkeit einer genutzten Open Source Software ist daher auch relevant für die Nachnutzungsmöglichkeiten innerhalb der Verwaltung.

Die Verwendung dieses Kriteriums trägt also dazu bei, dass die genutzte Open Source Software langfristig, wirtschaftlich, auf die eigenen Bedürfnisse angepasst und sicher verwendet werden kann.

2. Sicherstellung der Upstream-Veröffentlichung vorgenommener Anpassungen

Ein ganz wesentliches Merkmal von Open Source Software ist die Möglichkeit zur Anpassung an die eigenen Bedürfnisse. Fehlende Funktionen lassen sich beispielsweise in Form eines "Patches" integrieren. Werden diese Änderungen jedoch nicht "upstream" in den zentralen Code zurückgegeben, ist die eigene modifizierte Version ein "Fork", d.h. eine Abspaltung. Langfristig ergibt sich daraus für den Auftraggeber das Problem der Pflege. Der zentrale Code wird vom Hersteller bzw. der Community permanent weiter entwickelt. Gerade im Hinblick auf neue Funktionen und besonders bei behobenen Sicherheitslücken muss die Abspaltung des Auftraggebers permanent angepasst werden. Da es sich hierbei oft um Millionen von Programmcodezeilen handelt, entstehen schnell nicht mehr leistbare und unwirtschaftliche Aufwände.

Es ist notwendig und sinnvoll, den Umgang mit Patches bereits im Vergabeverfahren zu klären. Bei der Beschaffung von Open Source Software sollte vom Auftragnehmer daher gefordert werden, dass die vorgenommenen Patches "upstream" in den zentralen Code zurück gegeben werden. Bei der Bewertung und der Vertragsgestaltung ist zu berücksichtigen, dass der Hersteller bzw. die Community über die Annahme solcher Patches entscheidet. Je enger die Beziehung des Dienstleisters zum Hersteller bzw. der Community ist, desto größer ist daher die Wahrscheinlichkeit, dass die Patches auch tatsächlich in den zentralen Code aufgenommen werden.

Die Rückführung von Anpassungen ermöglicht es der gesamten Community, diese Änderungen zu überprüfen, zu testen und weiter zu verbessern. Dieser offene Überprüfungsprozess trägt zur Erhöhung der Softwarequalität und zur Identifikation sowie Behebung von Sicherheitsproblemen bei. Anbieter, die aktiv zur Open-Source-Community beitragen, nutzen die kollektive Expertise und fördern eine robustere und sicherere Software.

Ein Auftragnehmer, der sicherstellt, dass seine Anpassungen und Patches upstream veröffentlicht werden, erleichtert im ausgeschriebenen Projekt zukünftige Software-Updates und die Wartung. Denn individuell entwickelte Features und Verbesserungen sind dann bereits im zentralen Code integriert. Dies reduziert den Aufwand und die Kosten für die Aktualisierung und Wartung der Software für den Auftraggeber.

Konsequente Upstream-Veröffentlichungen fördern eine nachhaltige Softwareentwicklung, da die Verbesserungen allen Nutzern der Software zugutekommen und nicht nur einem einzelnen Projekt oder Kunden. Das ermöglicht auch die Nachnutzung von einmal entwickelten Softwarelösungen bzw. Teilkomponenten für die Verwaltung durch andere Behörden (“Einer-für-alle-Prinzip”). Zudem entspricht dies auch dem Prinzip "Public Money, Public Code", demzufolge öffentlich finanzierte Software der Allgemeinheit auch wieder zur Verfügung gestellt werden sollte.

3. Sicherstellung eines hoch qualitativen Third-Level-Supports

Die Qualität des Supports hat direkten Einfluss auf die Zufriedenheit und Produktivität der Endanwender. Ein strukturiertes Support-System, das schnelle Reaktionszeiten und fachliche Kompetenz gewährleistet, sorgt dafür, dass Anwender und Administratoren effektive Lösungen für ihre Anliegen erhalten. Dies spart nicht nur wertvolle eigene Arbeitszeit des Auftraggebers, sondern trägt zur positiven Nutzererfahrung bei und fördert die Akzeptanz der Softwarelösung.

Moderne Softwarelösungen sind zunehmend komplex und erfordern ein fundiertes technisches Verständnis für eine effektive Unterstützung. Die Verbreitung von Open Source Software unterliegt keinen wesentlichen Beschränkungen und letztlich kann jedes Unternehmen Dienstleistung und Support zu einer Open Source Software anbieten. First- und Second-Level-Support sind dabei vergleichsweise unkritisch. Denn Anwenderfragen z.B. zu einem vergessenen Passwort (First-Level-Support) oder dem Neustarten eines Servers, dem Einspielen von Updates oder Konfigurationsanpassungen (Second-Level-Support) verlangen nur allgemeine IT-Kenntnisse. Bei Open Source Software erfordert der Third-Level-Support jedoch die Expertise des Auftragnehmers mit dem Quellcode des konkreten Produkts. Denn für die Behebung von Funktionsstörungen der genutzten Software (Third-Level-Support) ist eine fundierte Analyse des Sourcecodes notwendig.

Um vertraglich vereinbarte Reaktionszeiten z.B. im Rahmen von Service-Level-Agreements (SLAs) einhalten zu können, muss der Anbieter entweder selber über eigenes qualifiziertes Personal verfügen, das die notwendige Expertise besitzt, oder der Anbieter muss sich (vertraglich) die spezialisierte Unterstützung des Herstellers verschaffen.

Dieses Kriterium ist für die Aufrechterhaltung der Geschäftsprozesse des Auftraggebers von kritischer Bedeutung. So wird die Betriebskontinuität gewährleistet und das Risiko von Ausfallzeiten durch eine schnelle und fachkundige Problembehebung minimiert.

4. Absicherung der Lieferkette durch Unterstützung von Basiskomponenten

Die langfristige Verfügbarkeit einer aktuellen und gepflegten Software-Lösung ist für den Auftraggeber von entscheidender Bedeutung. Open Source Software besteht zumeist aus unterschiedlichen Basiskomponenten, die immer wieder in neue Softwarelösungen integriert und daher in unterschiedlichsten Kontexten verwendet werden können. Diese Basiskomponenten, die Teil der Lieferkette sind, werden oftmals von ehrenamtlichen Akteuren oder Organisationen innerhalb der Open-Source-Community entwickelt und gepflegt. Die Sicherheit jeder Softwarelösung hängt davon ab, dass alle in ihr verwendeten Komponenten auf einem aktuellen Stand sind. Die Verwaltung sollte daher bei der Vergabe auch ein Augenmerk auf die Lieferkette richten und Anbieter auswählen, die sich ihrerseits innerhalb der Open-Source-Community für die Unterstützung der von ihnen genutzten Basiskomponenten engagieren.

Anbieter können beispielsweise die Entwickler der Basiskomponenten dabei unterstützen, die im Cyber Resilience Act (CRA) geforderten Anforderungen und Dokumentationspflichten umzusetzen.

Eine Mitarbeit des Auftragnehmers an den Open-Source-Basiskomponenten signalisiert eine transparente Arbeitsweise und die Bereitschaft, Wissen und Erfahrungen mit anderen zu teilen. Dies stärkt das Vertrauen in den Anbieter, da es zeigt, dass er nicht nur von der Open-Source-Community profitiert, sondern auch bereit ist, eigene Beiträge zum gemeinsamen Nutzen einzubringen und damit die langfristige Verfügbarkeit der Software zu ermöglichen. Ein Anbieter, der sich als aktiver Teil der Community beweist, kann bei der Lösung von Problemen seinerseits auf eine breite Unterstützung zählen, wovon letztlich auch der Auftraggeber profitiert.

Dieses Kriterium sichert für die öffentliche Verwaltung die langfristige Verfügbarkeit von Support, Updates und Weiterentwicklungen aller in Open-Source-Produkten verwendeten Komponenten und gewährleistet damit die Sicherheit der Lieferkette.

Zertifizierungen im Open-Source-Bereich

Ähnlich wie bei proprietärer Software gibt es auch im Open-Source-Bereich unterschiedliche Zertifizierungen, mit denen Anbieter ausgezeichnet werden können. Wenn die öffentliche Hand aus den Anbietern einen geeigneten Dienstleister auswählen möchte, kann es sich lohnen, sich zu informieren, ob es im Umfeld des Ausschreibungsgegenstandes eine geeignete Zertifizierung gibt, die die Qualität des Anbieters nachweist. Falls entsprechende Zertifizierungen vorgelegt werden können, sollten diese in die Bewertung der Anbieter einfließen.

Einige Open-Source-Hersteller wie beispielsweise Nextcloud oder Univention zertifizieren beispielsweise Partnerunternehmen, mit denen sie zusammen arbeiten. Solche Zertifizierungen sind gut geeignet, um eine bestehende Beziehung des Anbieters zum Software-Hersteller nachzuweisen (siehe Kriterium 1).

Einige Open-Source-Hersteller oder -Communities wie beispielsweise LibreOffice oder RedHat zertifizieren hingegen Entwickler, die ein besonderes Fachwissen im Zusammenhang mit der entsprechenden Open Source Software besitzen. Wenn ein Dienstleister solche Entwickler bei sich angestellt hat, ist diese Zertifizierung gut geeignet, um die Expertise des Anbieters im Zusammenhang mit der Upstream-Veröffentlichung vorgenommener Anpassungen und Patches (siehe Kriterium 2) sowie der Sicherstellung eines hoch qualitativen Third-Level-Supports nachzuweisen (siehe Kriterium 3).

Der internationale “OpenChain”-Standard ISO/IEC 5230 konzentriert sich auf Software-Lieferketten, einfachere Beschaffung und Lizenz-Compliance. Der OpenChain-Standard kann von einer akkreditierten Zertifizierungsstelle verliehen werden oder über eine Selbst-Zertifizierung erlangt werden. Diese Zertifizierung ist geeignet, um allgemein nachzuweisen, dass ein Anbieter sich bereits intensiver mit den Open-Source-Lizenz-Anforderungen und den Besonderheiten des Open-Source-Ökosystems auseinandergesetzt hat. Da auch das Erzeugen einer Software Bill Of Materials (SBOM) zu dem Standard gehört, kann über diese Zertifizierung auch das Engagement eines Anbieters im Zusammenhang mit der Absicherung der Lieferkette durch Unterstützung von Basiskomponenten nachgewiesen werden (Kriterium 4).

Nicht in allen Bereichen oder Open-Source-Produktgruppen gibt es entsprechende Zertifizierungen. Dennoch kann es sich für die Verwaltung lohnen, im Vorfeld einer Ausschreibung eine kurze Recherche über mögliche passende Zertifizierungen im Umfeld des Ausschreibungsgegenstandes durchzuführen. Obwohl Anbieter über Zertifizierungen ihre entsprechende Qualität nachweisen können, kann es in Einzelfällen kontraproduktiv sein, eine Zertifizierung anzufordern, die für den Ausschreibungsgegenstand überdimensioniert oder inhaltlich unpassend ist.

Anhang: Kriterienkatalog

Auswahl und Bewertung durch die fachliche Stelle

Dieser Anhang ist als "Baukasten" gedacht und sollte jeweils auf die konkrete Ausschreibungssituation angepasst werden. Natürlich können auch Texte aus dem vorherigen Kapitel "Kriterien zur Nachhaltigkeit" zur ergänzenden Erläuterung herangezogen werden. Diese Vorschläge stellen keine Rechtsberatung dar und müssen von der ausschreibenden Stelle jeweils vergaberechtlich geprüft werden.

In der Unterlage für Ausschreibung und Bewertung von IT-Leistungen (UfAB) wird in Abschnitt 4.2.1 der vergaberechtliche Rahmen für die Ermittlung des wirtschaftlichsten Angebots abgesteckt. Da bei Open Source Software kein eigentlicher Preis für die Software und damit für deren Weiterentwicklung und Pflege zu entrichten ist, ist die Methode der einfachen Preiswertung zur Auswahl eines nachhaltigen Angebots ungeeignet.

In der Praxis findet bei vielen Vergaben ohnehin keine reine Preiswertung statt, stattdessen wird eine Kombination aus Preis- und Leistungskriterien betrachtet. Dies passiert z.B. oft über B-Kriterien im Lastenheft oder bei komplexeren Beschaffungen über den Nachweis von Betriebs-, Service- und Weiterentwicklungskonzepten.

In einer Ausschreibung wird bei Open-Source-Lösungen üblicherweise ein bestimmter Dienstleistungsumfang gefordert. Zusätzlich zu anderen Leistungskriterien können auch Nachhaltigkeitskriterien – abhängig vom gewünschten Ziel der Ausschreibung – mit A- oder B-Kriterien in die Bewertung bei der Vergabe einfließen.

1. Kriterium: Beziehung zum Software-Hersteller / Community

Vorschlag für die Formulierung als A-Kriterium

- Wird die entsprechende Software von einem kommerziellen Softwarehersteller entwickelt und gepflegt, kann der Anbieter u.a. einen Vertrag mit dem Softwarehersteller nachweisen und darlegen, welche Leistungen durch diesen Vertrag abgedeckt werden (z.B. enge Beziehung zum Softwarehersteller durch bestehende Premium-Partnerschaft o.ä.).
- Wird die entsprechende Software von einer Community entwickelt und gepflegt, kann der Anbieter u.a. nachweisen, dass er
    - bereits Beiträge zu der entsprechenden Software (erfolgreiche Merges etc.) geleistet hat
    - Core-Contributoren/Hauptentwickler der entsprechenden Software im eigenen Unternehmen beschäftigt
    - an Veröffentlichungen zu der entsprechenden Software mitgewirkt hat (zu Neuentwicklungen, Dokumentation, Schulungsmaterialien, Absicherung der Lieferkette o.ä.)

Vorschlag für die Formulierung als B-Kriterium

Der Anbieter legt seine Beziehung zum Hersteller bzw. zur Community dar. Bewertet wird der Zugang zu technischem Support, zu Software- und Security-Updates, zu Insider-Wissen über geplante Entwicklungen und Einfluss auf die weitere Softwareentwicklung.

Begründung: Bei Software ist eine enge Verzahnung mit dem Hersteller bzw. der Maintainer-Community eminent wichtig, um hohe Standards bei IT-Sicherheit und Support zu gewährleisten.

Max. 4000 Zeichen. Gewichtung: 50 (=> 0 bis 250 Wertungspunkte)

Punkteskala:

- 0 Punkte: Keine Angaben oder keine qualifizierenden Angaben des Bieters.
- 1 Punkt: Minimale Beziehung zum Hersteller oder zur Community ohne nennenswerte Einflussnahme.
- 2 Punkte: Grundlegende Beziehung, die jedoch nicht signifikant zur Verbesserung der Software beiträgt.
- 3 Punkte: Gute Beziehung zum Hersteller oder zur Community mit nachweislichem Einfluss auf die Softwareentwicklung.
- 4 Punkte: Produktive Beziehung mit Zugang zu Insider-Wissen und direktem Support durch Core-Entwickler.
- 5 Punkte: Der Anbieter hat eine vertraglich geregelte Partnerschaft mit Zugang zu Insider-Wissen und direktem Support oder der Anbieter beschäftigt einen oder mehrere Core-Entwickler des Open-Source-Projektes.

2. Kriterium: Sicherstellung der Upstream-Veröffentlichung vorgenommener Anpassungen

Vorschlag für die Formulierung als A-Kriterium

- Der Anbieter kann Commitrechte im Projekt nachweisen und/oder ist Certified Contributor
- Der Anbieter beschäftigt einen oder mehrere Core-Contributoren/Hauptentwickler des Projekts
- Der Anbieter hat eine aktive Partnerschaft mit dem Hersteller der Software und maßgeblichen Einfluss auf die Entwicklung der Software (z.B. Einreichung von Feature-Requests, Beteiligung an der Priorisierung von Bugfixes, usw...)

Vorschlag für die Formulierung als B-Kriterium

Der Anbieter legt dar, wie beauftragte Anpassungen oder sich eventuell im Projektverlauf ergebende Anpassungen in den Code der Software upstream zurückfließen können.

Begründung: Die Nachhaltigkeit der Investition und die Verfügbarkeit von Erweiterungen und Sicherheitsupdates soll für den Auftraggeber auch langfristig bei neuen Versionen gesichert sein.

Max. 4000 Zeichen. Gewichtung: 50 (=> 0 bis 250 Wertungspunkte)

Punkteskala:

- 0 Punkte: Keine Angaben oder keine qualifizierenden Angaben des Bieters

- 1 Punkt: Bieter kann generell eigenes Entwickler-Personal nachweisen
- 2 Punkte: Bieter legt konzeptionell dar, dass und wie Entwicklungen upstream einfließen
- 3 Punkte: Bieter legt dar, dass er bereits relevanten eigenen Code zum Projekt beigetragen hat
- 4 Punkte: Bieter kann Commitrechte im Projekt nachweisen und/oder ist Certified Contributor
- 5 Punkte: Bieter beschäftigt einen oder mehrere Core-Entwickler des Projekts

3. Kriterium: Sicherstellung eines hoch qualitativen Third-Level-Supports

Vorschlag für die Formulierung als A-Kriterium

- Der Anbieter kann vertraglich sicherstellen, dass Third-Level Support durch den entsprechenden Hersteller geleistet wird
- Der Anbieter kann nachweisen, dass er Entwickler mit Commitrechten oder Core-Entwickler der entsprechenden Software beschäftigt
- Der Anbieter kann nachweisen, dass er in den letzten Jahren regelmäßig Bugfixes an der entsprechenden Software durchgeführt hat (z.B. mit Link o.ä. auf die entsprechenden Fixes)

Vorschlag für die Formulierung als B-Kriterium

Der Anbieter legt dar, wie er den Third-Level-Support entsprechend der geforderten Service-Level-Agreements (SLAs) entweder durch einen Vertrag mit dem Hersteller oder durch eigenes qualifiziertes Personal sicherstellt.

Begründung: Zur Absicherung des Betriebes und der Sicherheit der Software ist ein qualifizierter Third-Level-Support unerlässlich. Bei Open Source Software kann das entweder durch einen Vertrag mit dem Hersteller abgesichert werden oder durch qualifizierte Entwickler der Software.

Max. 4000 Zeichen. Gewichtung: 50 (=> 0 bis 250 Wertungspunkte)

Punkteskala:

- 0 Punkte: Keine Angaben oder keine qualifizierenden Angaben des Bieters
- 1 Punkt: Support-Mitarbeitende sind vorhanden, es mangelt aber an Details zur Qualifikation des Personals
- 2 Punkte: Eigenes Personal mit nachgewiesener Qualifikation zur Fehleranalyse, jedoch werden die geforderten SLAs nicht ausreichend erfüllt
- 3 Punkte: Eigenes Personal mit nachgewiesener Qualifikation zur Fehleranalyse. Die geforderten SLAs werden abgedeckt
- 4 Punkte: Vertraglich geregelter Herstellersupport ist vorhanden oder der Bieter beschäftigt einen oder mehrere Core-Entwickler des Projekts, jedoch werden die geforderten SLAs nicht ausreichend erfüllt
- 5 Punkte: Vertraglich geregelter Herstellersupport ist vorhanden oder der Bieter beschäftigt einen oder mehrere Core-Entwickler des Projekts. Die geforderten SLAs werden abgedeckt

4. Kriterium: Absicherung der Lieferkette durch Unterstützung von Basiskomponenten

Vorschlag als Formulierung als A-Kriterium

Der Anbieter unterstützt im Sinne des Cyber Resilience Act (CRA) Basiskomponenten oder Projekte, die Teil seiner Softwarelösung sind:

- Der Anbieter unterstützt die Entwickler der Basiskomponenten maßgeblich bei der im Cyber Resilience Act (CRA) geforderten Absicherung der Lieferkette und der Aufschlüsselung der in den Basiskomponenten enthaltenen Software über eine Software Bill Of Materials (SBOM)
- Der Anbieter hat sich mit seinen Beiträgen zu einer Basiskomponente oder einem externen Projekt als führende Kraft etabliert, insbesondere im Hinblick auf die Verbesserung der Sicherheit der Software
- Der Anbieter beschäftigt Core-Entwickler eines externen Projekts oder einer Basiskomponente, die Teil der Lieferkette ist

Vorschlag für die Formulierung als B-Kriterium

Der Anbieter legt dar, wie er Beiträge zur Förderung, Weiterentwicklung und Sicherheit der in seiner Software integrierten Basiskomponenten leistet.

Begründung: Software wird meist unter Verwendung vorhandener Komponenten (Bibliotheken, Datenbanken, Frameworks…) entwickelt. Diese bilden einen Teil der Lieferkette und müssen qualitative wie sicherheitsrelevante Anforderungen erfüllen. Durch die Unterstützung dieser Basiskomponenten beteiligt sich der Anbieter an der Absicherung der Lieferkette.

Punkteskala:

- 0 Punkte: Es gibt keine Nachweise für inhaltliche Beiträge oder aktive Unterstützung
- 1 Punkt: Minimale Unterstützung der Lieferkette: Es gibt vereinzelte Belege für Beiträge, jedoch ohne signifikante inhaltliche Tiefe oder Regelmäßigkeit
- 2 Punkte: Ausreichende Unterstützung der Lieferkette: Der Anbieter hat einige inhaltliche Beiträge geleistet und an Konferenzen zu involvierten Software-Projekten teilgenommen, jedoch bleibt das Engagement sporadisch oder von begrenztem Einfluss
- 3 Punkte: Gute Unterstützung der Lieferkette: Der Anbieter zeigt regelmäßige und relevante Mitarbeit und leistet Beiträge bei Konferenzen der involvierten Software-Projekte
- 4 Punkte: Der Anbieter ist für seine Kompetenz und Unterstützung einzelner Projekte aus der Lieferkette in der entsprechenden Open-Source-Community anerkannt. Beispielsweise unterstützt er bei der im Cyber Resilience Act (CRA) geforderten Absicherung der Lieferkette und der Aufschlüsselung der enthaltenen Komponenten über eine Software Bill Of Materials (SBOM)
- 5 Punkte: Der Anbieter beschäftigt Core-Entwickler eines externen Projekts, das Teil der Lieferkette ist, und hat sich mit seinen Beiträgen als führende Kraft auch in Hinblick auf die Verbesserung der Sicherheit als Innovator etabliert

Über die Open Source Business Alliance – Bundesverband für digitale Souveränität e.V.

Die Open Source Business Alliance (OSBA) ist der Verband der Open Source Industrie in Deutschland. Sie vertritt über 230 Mitgliedsunternehmen, die jährlich mehr als 126 Milliarden EUR erwirtschaften. Gemeinsam mit wissenschaftlichen Einrichtungen und Anwenderorganisationen setzt sie sich dafür ein, die zentrale Bedeutung von Open Source Software und offenen Standards für einen erfolgreichen digitalen Wandel im öffentlichen Bewusstsein nachhaltig zu verankern. Zudem sollen Innovationen im Bereich Open Source vorangetrieben werden. Das Ziel der Open Source Business Alliance ist es, Open Source als Standard in der öffentlichen Beschaffung und bei der Forschungs- und Wirtschaftsförderung zu etablieren. Denn Open Source und offene Standards sind zwingende Grundlagen für digitale Souveränität, Innovationsfähigkeit und Sicherheit im digitalen Wandel und damit die Antwort auf eine der größten Herausforderungen unserer Zeit.

Lizenz für die Veröffentlichung