Innovationsverbund
Öffentliche Gesundheit

Stellungnahme zur Anhörung Open Source

Bianca Kastl, Innovationsverbund Öffentliche Gesundheit e. V. (InÖG)

01.12.2024

2024.12.01
17:35:31 +01'00'

Vorbemerkung

In diese Stellungnahme fließen die Erfahrungen aus Open-Source-Projekten aus Sicht der Zivilgesellschaft seit 2020 ein. Diese Stellungnahme ist bewusst aus Sicht der Vorsitzenden des gemeinnützigen Innovationsverbund Öffentliche Gesundheit e.V. geschrieben, enthält an bestimmten Stellen aber unweigerlich auch Verweise auf im beruflichen Kontext gemachte Erfahrungen aus der Verwaltung heraus. Dies ist in keinster Weise repräsentativ für meinen Arbeitgeber.

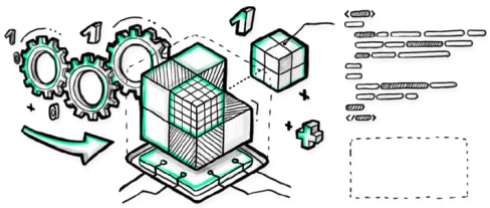
Vor dem Eingehen auf die Fragen aus dem Fragenkatalog sei eine persönliche Vorbemerkung angebracht:

Gegen Ende der Legislatur sei auf einen Satz des inzwischen hinfälligen Koalitionsvertrags hingewiesen, der sich in ebendiesem im Abschnitt Digitaler Staat und Digitale Verwaltung findet:

> *Entwicklungsaufträge werden in der Regel als Open Source beauftragt, die entsprechende Software wird grundsätzlich öffentlich gemacht.*

Gemessen an der Anzahl der letztendlich als Open Source beauftragten und öffentlich gemachten Software im Bereich der digitalen Verwaltung muss leider attestiert werden, dass Open Source hier leider keine durchgängige Regel geworden ist; grundsätzlich öffentlich gemacht ist Software der digitalen Verwaltung leider auch nicht. Open Source in der Digitalisierung von Verwaltung und Gesundheitswesen ist bedauerlicherweise eher noch sehr selten. Je nach Bundesministerium variiert der als Open Source entwickelten Software-Vorhaben vom Bereich von quasi nicht vorhanden bis zu einem Anteil von etwa 80 Prozent, der noch am ehesten als regelhaft bezeichnet werden kann.

Da der vorliegende Fragenkatalog eher nur analytisch auf mögliche Fallstricke und Chancen eingeht, soll die Vorbemerkung vor allem eins: Ermutigen, dass Open Source auch in der Verwaltung oder im Gesundheitswesen möglich ist. Aus der Erfahrung von zwei Projekten, die als Open Source für oder aus der Verwaltung entwickelt wurden, will ich Ihnen allen zurufen: **Ja, Open Source in der Verwaltung und im Gesundheitswesen ist möglich. Eigentlich sogar sehr einfach.**



Manchmal ist es für Open-Source-Projekte sogar möglich, aus einem Impuls der Zivilgesellschaft heraus, ganz viele Kommunen in unterschiedlichen Bundesländern während einer Pandemie mit ein und derselben Software zu verbinden. In ganz unterschiedlichen Betriebsszenarien in mehreren Bundesländern – über Landes- und Kommunengrenzen hinweg. Klingt eigentlich unmöglich, ist dem InÖG aber mit IRIS Connect tatsächlich gelungen. Am Ende stellte sich in diesem sehr heterogenen Feld die Frage: Ohne Open Source? Wie hätte das denn gehen sollen? Ein Vorgehen nach Open Source war hier letztendlich sogar Gelingensbedingung. Weil offenes Entwickeln und das offene Bereitstellen von Software es überhaupt erst möglich gemacht haben, Software so schnell an ganz unterschiedliche Kommunen zu verteilen.

Aber auch aus der Verwaltung selbst heraus ist es möglich, Open-Source-Software als Verwaltung selbst zu entwickeln. Aktuell kann ich beruflich – getrennt von meinem zivilgesellschaftlichen Engagement – etwa das Projekt GA-Lotse betreuen – aus der Verwaltung heraus. Open Source in der Verwaltungsdigitalisierung ist zwar immer noch ein immer wieder neues Abenteuer. Ein mögliches Abenteuer aber.

Bianca Kastl (bianca.kastl@inoeg.de)

Open Source in der Verwaltungsdigitalisierung ist möglich, wenn es mindestens einen Menschen gibt, dem der Aspekt Open Source wichtig ist, der sich dafür einsetzt. In meinen Gesprächen und dem Erfahrungsaustausch in den letzten Jahren mit anderen Open-Source-Projekten gab es dabei ganz unabhängig von der Größe oder dem Budget eines Softwarevorhabens oftmals immer die eine Person in Behörden, im Gesundheitswesen oder der Zivilgesellschaft, die sich für Open Source in diesem Projekt verantwortlich eingesetzt hat. Meist ist es oft nur diese eine Person, die die Begeisterung für Open Source in Softwareprojekte oder Verwaltung hineinträgt.

Dazu möchte ich Sie ermutigen: Sie können die eine Person sein, die Open Source in Verwaltung oder Gesundheitswesen möglich macht! Egal ob in leitender Position, als Mitarbeitende einer Behörde oder Teil der Zivilgesellschaft. Auch Sie können Open Source möglich machen.

Ein paar Hinweise zu Vorteilen und möglichen Fallstricken finden Sie in den Antworten auf den Fragenkatalog.

Die Fragen sind soweit möglich in Themenkomplexen zusammengefasst, enthalten teilweise Querweise auf andere Fragen, aber jeweils ihre ursprüngliche Nummerierung.

Bianca Kastl (bianca.kastl@inoeg.de)

Themenblock Vorteile und Nachteile von Open Source, im Kontext der Verwaltungsdigitalisierung

Frage 1: Vor- und Nachteile von Open Source

Welche Vor- und Nachteile hat Open Source-Technologie allgemein und besonders im Hinblick auf technische, sicherheitsrelevante, konzeptionelle, soziale, finanz-, außenpolitische und gesellschaftliche Aspekte? Welche der genannten Vor- und Nachteile kommen besonders zum Tragen, wenn Open Source-Technologien im staatlichen Kontext eingesetzt werden?

Frage 6: Vorteile und Herausforderungen von Open Source in der Verwaltungsdigitalisierung

Welche Vorteile oder Herausforderungen für die Verwaltungsdigitalisierung ergeben sich durch die Nutzung von Open Source-Technologien?

tl;dr:

Vorteile: Wegfall von Lizenzkosten 👩‍🏫, Überprüfbarkeit der Software anhand des Quelltexts 👩‍🏫, einfachere Nachnutzung und Anpassbarkeit 👩‍🏫, Transparenz gegenüber Bürgerinnen 👩‍🏫, bessere Absicherbarkeit, Möglichkeit zur Integration von Beiträgen von Bürgerinnen oder Entwicklungscommunity 👩‍🏫, Nutzung von international etablierten Open-Source-Lizenzen, Vermeidung von Lock-in-Effekten 👩‍🏫

Nachteile: höhere Anforderungen an IT-Kompetenzen 👩‍🏫, Community-Pflege notwendig, tiefergehendes Lizenzverständnis notwendig, Haftungsfrage

Die für den staatlichen Kontext besonders relevanten Aspekte haben eine Markierung (👩‍🏫).

Ausführliche Antwort:

Frage 1 und Frage 6 werden wegen ihres Sinnzusammenhangs gemeinsam beantwortet.

Open Source hat eine Reihe von Vorteilen:

- Open-Source-Software kann **unter Einhaltung der jeweiligen Lizenzbedingungen kostenfrei genutzt und verteilt** werden. Damit ist Open-Source-Software im Allgemeinen kostengünstiger in der Gesamtbetrachtung durch den Wegfall von Lizenzgebühren. Software an sich kostet in der Nutzung aber unabhängig vom Lizenzmodell an irgendeiner Stelle immer Geld, sei es in der Entwicklung, im Betrieb oder der Weiterentwicklung oder dem Aufrechterhalten eines lauffähigen Zustands. 👩‍🏫
- **Open-Source-Software kann durch Dritte** auf Ebene des Quelltexts und idealerweise auch der Instruktionen zum Bauen (Build) von lauffähiger Software **eingesehen und geprüft werden**. Das macht Open-Source-Software auch in Bereichen mit hohen Sicherheitsanforderungen wie etwa Teilen der Verwaltung oder dem Gesundheitswesen zu einer guten Wahl. Software kann hier vor der Nutzung hinsichtlich ihrer Qualität und Eignung sehr genau geprüft werden kann im Gegensatz zu proprietären Softwarelösungen. 👩‍🏫

Bianca Kastl (bianca.kastl@inoeg.de)

- Durch das Vorliegen des Quelltexts kann Open-Source-Software **einfach angepasst und nachgenutzt** werden. Durch die Verwendung von Methodiken zum Abzweigen von unterschiedlichen Software-Versionen (Branches und Forks) können aus einer Open-Source-Software auch sehr unterschiedliche Produkte entstehen, die sich nur einen Kern an Software teilen, sich aber in Zukunft technisch unabhängig voneinander verhalten können. Durch die Anpassbarkeit ist eine passgenaue Nachnutzung in unterschiedlichsten Einsatzszenarien möglich. 💁‍♀️
- Freie und offene Software liefert Bürger*innen die Möglichkeit, **staatliches Handeln im Digitalen transparent nachvollziehen** zu können. Dies ist zum Beispiel im Kontext Software für Wahlen wichtig, aber auch im Bereich von Algorithmen, etwa im Sozialwesen. Durch die direkte Nachvollziehbarkeit der Funktionsweise einer Open-Source-Software kann die konkrete Funktionsweise durch neutrale Dritte verifiziert werden und eventuelles Misstrauen kann reduziert werden. 💁‍♀️
- **Open-Source-Software ist besser absicherbar**. Im Wesentlichen ist Software dann sicher, wenn eine Software sicher ist. Das klingt jetzt vielleicht wie eine Binsenweisheit, ist aber gar nicht so einfach, durch einige wenige Personen selbst durch eine gewisse Ignoranz eigener Fehler und Unzulänglichkeiten zu erreichen. Eine gute Softwaresicherheit ist oftmals nur dann wirklich erreichbar, wenn eine Software durch viele Entwickler*innen überprüft werden kann. Mögliche Fehler können auch direkt angemerkt und mittels vorgeschlagener Änderung am Quelltext in einem Versionsverwaltungssystem (sog. Commits und Merge Requests) eingebracht werden können, die wiederum selbst durch andere überprüft werden können.
- Bei Open-Source-Software können Bürger*innen oder eine Entwicklungscommunity selbst zu einer Softwarefunktion oder Inhalte beitragen, etwa Übersetzungen von Software. So kann zum Beispiel eine technisch versierte Zivilgesellschaft bei der Bereitstellung von Software in anderen Sprachen als der Amtssprache eines Landes helfen und so die Software für viele Menschen zugänglicher machen. 💁‍♀️
- Es gibt international anerkannte Open-Source-Lizenzen, die die wichtigsten rechtlichen und regulatorischen Fragen klärt, etwa die Bedingungen, unter denen eine Software weitergegeben werden kann. Etabliert ist hier als internationale Referenz die Liste der durch die Open Source Initiative anerkannten Lizenzen.
- Die Nutzung von Open Source führt zur **Vermeidung von Lock-in-Effekten**. Durch das Offenliegen des Quelltexts können beispielsweise mit dieser Software erstellte Dateien einfacher in andere Formate konvertiert werden, falls die Software gewechselt werden soll. Dadurch entsteht keine Abhängigkeit von einem speziellen Anbieter und nicht die Gefahr des Ausgeliefertseins überzogenen Lizenzgebührforderungen und Preiserhöhungen. Ebenso besteht keine Abhängigkeit von geopolitischen Rahmenbedingungen wie Zöllen oder internationalen Datenschutzabkommen zur Verarbeitung von Daten in bestimmten Ländern. 💁‍♀️

Bianca Kastl (bianca.kastl@inoeg.de)

Open Source hat aber auch ein paar mögliche Nachteile, die je nach Einsatzzweck unterschiedlich schwer wiegen können:

- Open-Source-Software kann unter bestimmten Bedingungen ein höheres Maß an IT-Kompetenzen im Einsatz notwendig machen, speziell, wenn es um die Verwendung von Serversoftware o. ä. geht. Hier sollte der Umgang mit Versionsverwaltungssystemen oder die Möglichkeit, Software direkt aus dem Quelltext zu kompilieren, nicht abschrecken. Im Allgemeinen ist Open-Source-Software inzwischen gut nutzbar auch für normal begabte IT-Nutzende, jedoch kann das Nutzen von Software oder die Konfiguration von Software auf Quellcode-Ebene teilweise anspruchsvoller sein als bei proprietärer Software. 🤦
- Open-Source-Software wird oftmals mit kostenfreier Software verwechselt, die einfach kostenlos genommen werden kann, ohne sich weiter kümmern zu müssen. Aber auch Open-Source-Software benötigt entsprechende Pflege der eigenen Software-Community bzw. Entwickler*innen benötigen Unterstützung aus der Community, um weiterhin nachhaltig und sicher Open-Source-Software bereitstellen zu können. Hier ist speziell im staatlichen Interesse ein Aufrechterhalten eines gesunden Open-Source-Ökosystems wichtig, um bei bestimmten Softwarekomponenten keine Risiken hinsichtlich schlecht gepflegter Software zu erzeugen.
- Bei der Nutzung von unterschiedlichen Open-Source-Softwarekomponenten unter einer neuen Software kann ein tiefergehendes Verständnis von Lizenzen notwendig sein, um mögliche Inkompatibilitäten von Softwarekomponenten wegen der jeweiligen Lizenzen zu vermeiden.
- Open-Source-Software wird zumeist mit einer **„As is“-Haftungsklausel** bereitgestellt, also im Prinzip wird die Software so, wie sie gesehen wurde, eingesetzt und die jeweilige Entität, die die Software einsetzt, haftet entsprechend für den Betrieb. Das kann für bestimmte staatliche Stellen problematisch sein, weil hier ein Risikoübergang eher zum Hersteller einer Software sinniger sein kann. Die Probleme mit der Haftung lassen sich aber durch Softwarepflege- oder Betriebsverträge mit Open-Source-Softwareherstellern beheben, sind aber nicht immer standardmäßig möglich.

Bianca Kastl (bianca.kastl@inoeg.de)

Themenblock Open Source und Bedingungen in der Verwaltungsdigitalisierung

Frage 2: Open Source und Verwaltungsdigitalisierung

Welche Voraussetzungen und Infrastrukturen braucht der erfolgreiche Einsatz von Open Source-Technologien im staatlichen Kontext?

tl;dr: Benötigt werden vor allem Kompetenzen, sowohl in der Durchsetzung als auch im IT-Wissen, und gleich funktionierende, universell verfügbare Infrastrukturkomponenten.

Ausführliche Antwort:

Abgesehen von den generellen Rahmenbedingungen, die jede Art von Technologie unabhängig vom Lizenzmodell braucht, etwa ausreichende Finanzierung, klare Zuständigkeiten und politische Priorisierung, **benötigt Open Source für den erfolgreichen Einsatz im staatlichen Kontext vor allem Kompetenzen**.

Diese Kompetenzen umfassen einerseits Kenntnisse im Bereich der Informationstechnik, die im Kontext von Open Source besonders sind, also etwa Besonderheiten von Open-Source-Lizenzen, aber auch genaueres Verständnis von Versionsverwaltungssystemen wie git. Für eine genaue Beurteilung einer Software anhand des offenen Quelltexts kann darüber hinaus auch tiefgreifendes IT-Wissen notwendig sein. Zum erfolgreichen Einsatz von Open Source im staatlichen Kontext sollten Verwaltungsmitarbeitende also prinzipiell technologisch verstehen, was dieser Quelltext denn da eigentlich zumindest abstrakt tut. So tiefgreifende Kenntnisse sind wegen des meist eher verwaltungsrechtlichen Schwerpunkts speziell von Führungspositionen in Verwaltungen selten.

Dennoch kann – auch aus eigener beruflicher Erfahrung – der Einsatz von Open Source nur dann gelingen, wenn innerhalb der Verwaltung auch intern Kompetenzen aufgebaut werden, die tieferes technologisches Verständnis haben. Die von Lilith Wittmann beschriebene Beratertreppe, bei der ein Dienstleister einen anderen Dienstleister im Auftrag der Verwaltung steuert, ist unbedingt zu vermeiden.

Etwas einfacher erscheint die **Durchsetzung von Open Source mittels politischer Zielsetzung,** was auch als Bereich der Kompetenz gesehen werden muss. Hier können konsequente Open-Source-Strategien, wie etwa in Schleswig-Holstein, die von Seiten der Staatsregierung begleitet werden, Vorbild sein. Open Source ist in dem Sinne – wie Digitalisierung selbst – immer auch eine Führungsaufgabe und sollte von der Leitungsebene zumindest ideell unterstützt werden.

Auf technischer Seite ist ein **Vorhandensein von gleich funktionierenden, universell verfügbaren Infrastrukturkomponenten** im Sinne von Infrastructure as a Service (IaaS), etwa Cloud-Technologien wie Servercluster und deren Orchestrierung z. B. über Kubernetes notwendig, um entsprechende Open-Source-Software sinnvoll verteilen zu können. Dabei kann die Nutzung von standardisierten, aber verteilt von unterschiedlichen Betreibern bereitstellbaren Cloud-Stacks wie aktuell in Thüringen zielführend sein. Oftmals folgt sonst nach der Beschaffung oder Entwicklung einer Open-Source-Software noch das Problem des Betriebs, der ohne sofort nutzbare Basiskomponenten nochmals Vergabeverfahren notwendig macht.

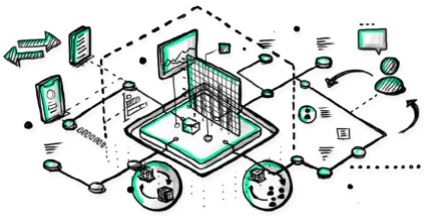
Bianca Kastl (bianca.kastl@inoeg.de)

Frage 9: Skalierung von Open Source im staatlichen Einsatz

Welche Herausforderungen beim Thema Skalierung und Rollout von Open Source Software Projekten im staatlichen Einsatz sind Ihnen begegnet und welche strukturellen Maßnahmen schlagen Sie vor, um diesen zu begegnen?

tl;dr: Herausforderungen: Heterogene IT-Landschaft und ganz unterschiedliche Betriebsszenarien. Unterschiedliches Kompetenzniveau. Empfehlung: Dezentral nutzbare, aber gemeinsam und standardisiert betreibbare Softwarelösungen.

Ausführliche Antwort:



Meine Erfahrung bei der Skalierung und dem Rollout von Open-Source-Projekten gründet sich auf die gemachten Erfahrungen in mehreren Open-Source-Projekten, welche sich in der Größenordnung der Nutzung in mehreren Bundesländern bewegen (etwa IRIS connect als Betriebsverantwortliche sowie GA-Lotse als technische Gesamtverantwortung). Da beide Projekte vollständig Open Source sind und auf entsprechenden dafür geeigneten Betriebsplattformen aufsetzen, ergeben sich bei der Skalierung und beim Rollout folgende Herausforderungen:

- **Die IT-Landschaft in Kommunen und Ländern ist hochgradig heterogen**. Das führt oftmals zur Mischung von Betriebsmodellen. Bei IRIS connect führte dies etwa zu einem Mischszenario von vollständig selbstständig durch uns managbaren Software as a Service Betrieb zentral für das jeweilige Bundesland mittels Cloudtechnologien bei einem Kommunaldienstleister für 2 Bundesländer, Betrieb auf On-Premises Infrastruktur eines Kommunaldienstleisters zentral für 1 Bundesland sowie hochgradig verteilter Betrieb (mehrere Kommunaldienstleister On-Premises, einzelne Kommunen On-Premises) auf Ebene einzelner Kommunen oder kreisfreier Städte in einem weiteren Bundesland.
- Durch die heterogene IT-Landschaft gibt es auch **unterschiedliche Kompetenzniveaus** im Umgang etwa mit Open-Source-Betriebssystemen wie Linux und Cloudtechnologien wie Docker oder Kubernetes. Hier reichte die Bandbreite von persönlichem, assistierten Installieren im Beisein von Dienstleistern bis zu vollständigem, eigenständigem Management durch den jeweiligen Dienstleister.
- Letztlich hat IRIS connect auch gezeigt, dass ein Betrieb gemeinsamer Softwarekomponenten in so einem verteilten und heterogenen Umfeld auf unterschiedlichsten Betriebsplattformen in verteilter Verantwortlichkeit möglich ist.
- Hinsichtlich der eigentlichen Anwendungsbetreuung ergeben sich immer sehr menschlich individuelle Anforderungen an Schulung und IT-Support, die aber ohnehin bei jedem Projekt anfallen. Hier hat Open Source keine besondere Auswirkung.

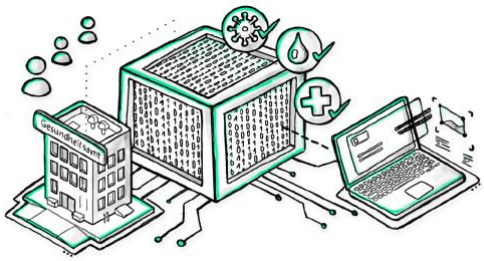
Prinzipiell verwendet GA-Lotse ein ähnliches Modell an Betriebsoptionen, hier zeigt sich aber, dass angesichts der Anforderungen an den Betrieb eher ein Software-as-a-Service-Modell seitens der Verwaltung präferiert wird. Es ist davon auszugehen, dass Software in der Verwaltung zukünftig immer mehr als Service konsumiert werden wird – ohne diese selbst betreiben zu müssen.

Bianca Kastl (bianca.kastl@inoeg.de)

Um die Software in Zukunft schneller und einfacher auszurollen, sind folgende Handlungsempfehlungen hilfreich:

- Für die Nutzung von Open-Source-Software, zumindest auf Webservices basierenden Diensten, sind **dezentral nutzbare, aber gemeinsam und standardisiert betreibbare Softwarelösungen** zu empfehlen. Ähnlich wie bei IRIS connect oder auch bei GA-Lotse sind hier verteilte, aber gemeinsame entwickelte, aber verteilt nutzbare Open-Source-Softwarekomponenten gerade wegen des Föderalismus letztlich einzig zielführend.
- Auf Basis gemeinsamer, aber verteilt betriebener Open-Source-Softwarekomponenten lassen sich dann im Verwaltungskontext auch Government-as-a-Platform-Ansätze einfacher darstellen. Diese Ansätze wären notwendig, um Dienste entsprechend resilient zu verteilen und möglicherweise dynamisch neu zuordnen zu können, etwa in Krisensituationen.
- Diese gemeinsam genutzten Komponenten brauchen eine finanzielle und operative Verstetigung, das könnte z.B. über das ZenDiS realisiert werden.
- Auf Betriebsplattformebene sind **standardisierte Infrastructure- oder Platform-as-a-Service-Dienste** für einen verteilten, aber gleichartigen Betrieb notwendig. Dabei sollte die eigentliche Betriebsumgebung selbst nicht wieder in einen weiteren Vendor-Lock-in führen (etwa durch Festlegung auf einen bestimmten großen Cloud-Anbieter). Ebenso sollten die Umgebungen nicht nur formal standardisiert sein, etwa in Form von einer gemeinsamen Strategie wie in der Deutschen Verwaltungscloudstrategie, sondern sich auch technisch gleich verhalten. Bereits existierende Beispiele für von speziellen Herstellern unabhängige Betriebsumgebungen sind etwa der Sovereign Cloud Stack mit einer starken Automatisierung der Compliance des Technologiestacks.
- Die Betriebsplattformeben braucht noch zwangsläufig eine zentrale Governance, aber eine Abstimmung der technischen Standards und klare Zuständigkeit und möglicherweise gemeinsames Monitoring.

Aus der Erfahrung im Betrieb von verteilten Open-Source-Softwareprojekten ist anzumerken, dass es eher unerheblich ist, ob verteilte Komponenten in Cloud-Umgebungen oder On-Premises betrieben werden, solange sich diese gleich verhalten und entsprechend gut gepflegt werden. Hybride Mischszenarien sind ebenfalls machbar.



Bianca Kastl (bianca.kastl@inoeg.de)

Frage 14: Open Source als Katalysator für innovative Verwaltung

Inwiefern könnte Open Source-Software als Katalysator für innovative Ansätze in der Verwaltung fungieren? Welche neuen Dienstleistungen oder Modelle könnten durch Open Source realisiert werden, um die Bürger besser zu bedienen?

tl;dr: Innovative Ansätze können durch Open Source besser multipliziert werden. Beispiele dazu sind Projekte von Code for Germany, wie etwa Ableger von Gieß den Kiez. Durch offene Schnittstellen und offenen Quelltext kann die Verwaltung als eine Art Plattform gedacht werden, auf der neue Anwendungen und Dienstleistungen erbracht werden können.

Ausführliche Antwort:

Open Source ermöglicht eine bessere Verteilung und Multiplizierung von Leistungen aus einzelnen Kommunen / Länder / Ressorts und vereinfacht die Anpassung dieser Leistungen an andere Kommunen / Länder / Ressorts. Ein praktisches Beispiel hierzu ist die die Anwendung Gieß den Kiez. Diese wurde 2020 vom CityLAB Berlin als Plattform zur Verbesserung der Bewässerung von städtischer Baumvegetation geschaffen. Diese wurde dann durch die Verfügbarkeit des Quelltexts als Open Source und die Datenverfügbarkeit als Open Data schnell zum Beispiel auf Leipzig und andere Städte durch die dortigen Code for Germany Ableger übertragen.

Generell kann durch die Bereitstellung von offenen Schnittstellen und offenen Daten die Verwaltung als Plattform dienen, die neue Dienstleistungen ermöglicht. Das kann etwa die Ermöglichung neuer Apps und Services beinhalten, die von der Verwaltung selbst nicht erbracht oder gesehen werden. Es kann aber auch der Verwaltung selbst dabei helfen, Lösungen aus anderen Kommunen / Ländern / Ressorts zu übertragen und zu adaptieren.

Frage 17: Barrieren gegen höheren Einsatz von Open Source bei staatlichen Stellen

Welche Barrieren sehen Sie gegen einen höheren Einsatz Open Source bei staatlichen Stellen, und wie bewerten Sie insbesondere folgende Barrieren:
• „harte“ Lock-In-Effekte zum Beispiel durch technische Abhängigkeiten, wenn Hardware nur mit bestimmter Software läuft, oder Software nur mit bestimmter proprietärer Software interoperabel ist,
• weiche Abhängigkeitsfaktoren wie Gewöhnungseffekte,
• mangelnde IT-Kompetenz im Einkauf, was zur Verlängerung von Rahmenverträgen oder mehr Einkauf von Vertrautem führt, weil man Alternativen nicht kennt oder ihre Risiken überschätzt,
• mangelnde IT-Kompetenz im Betrieb, weil es weniger Erfahrung mit Open-Source-Dienstleistenden gibt,
• Folgen von Lobbyismus großer Hersteller proprietärer Software,
• fehlende Transparenz zum Einsatz von Open Source und proprietärer Software,
• mangelnde strategische Weitsicht beziehungsweise Überschätzung von kurzfristigem Nutzen bei Unterschätzung langfristiger Risiken?

tl;dr: Die oftmals fehlende IT-Kompetenz ist das Hauptproblem. Angespannte finanzielle Situation begünstigt Open Source eher. Schwere der Probleme der Schwere nach absteigend: IT-Kompetenz im

Bianca Kastl (bianca.kastl@inoeg.de)

Betrieb, Lobbyismus, mangelnde Weitsicht, IT-Kompetenz im Einkauf, Lock-in-Effekte, Gewöhnungseffekte, Transparenz.

Ausführliche Antwort:

In Frage 2↘, 6↘, 7↘, 8↘ und 10↘ wurden bereits einige Lösungsansätze und Probleme erörtert, weswegen hier nur noch kurz auf die Bewertung der Barrieren eingegangen werden soll.

Eigentlich begünstigt eine angespannte finanzielle Situation der Haushalte Open-Source-Software, da hier, wie in Frage 1↘ erwähnt, zum einen Lizenzkosten gespart werden können, durch den freien Austausch von Software aber auch Synergien genutzt werden können ohne weitere Auflagen.

In der Priorisierung ergibt sich die Höhe der Barrieren in etwa wie folgt:

- **IT-Kompetenz im Betrieb**: Ist am höchsten zu bewerten, weil diese kontinuierlich vorhanden sein muss und nicht im Falle von Vergabeverfahren temporär ergänzt werden kann. Durch Vorhandensein dieser Kompetenzen wäre auch ein Stückweit eine eigenständige autarke Nutzung bereits bestehender Open-Source-Software einfacher möglich, was bei einem Ökosystem an Open-Source-Software bereits viele Anwendungsfälle abdecken kann.
- **Lobbyismus**, weil dieser stark das Bild in der Verwaltung oftmals gegen Open Source prägt und eine Veränderung hin zu Open Source in einer ohnehin schon veränderungsschweren Umgebung weiter verlangsamt. Oftmals in Verbindung mit Fehlinformationen zur Sicherheit von Open Source, vgl. Frage 8↘.
- **Mangelnde Weitsicht**, weil Open Source die Vorteile meist nur auf längere Sicht voll ausspielen kann, wie etwa die Unabhängigkeit von bestimmten Anbietern. Bei der initialen Beschaffung von IT-Produkten fehlt oftmals diese Weitsicht, weswegen Open Source hier oftmals nicht voll gewürdigt wird.
- **IT-Kompetenz im Einkauf**, weil die Lizenzierung und die Vertragsrahmenbedingungen für Open-Source-Software oftmals nicht in der Form bekannt ist, um hier eine qualifizierte Entscheidung und gute Vergabebedingungen für Open Source zu schaffen.
- **Lock-in-Effekte**, größtenteils sind diese intern als zu überwindend anerkannt, allerdings führen die vorgenannten Barrieren zu Problemen beim Versuch der Abschaffung dieser Lock-in-Effekte.
- **Gewöhungseffekte**, weil diese im Kontext der Digitalisierung der Verwaltung ohnehin ein Problem darstellen und es eher unerheblich ist, welches Lizenzmodell die Software hat, auf die sich Mitarbeitende weigern umzusteigen.
- **Transparenz** als letztgenanntes, weil eine generelle Transparenzkultur in der Verwaltung ohnehin oftmals fehlt und der Wunsch nach Transparenz der Software oftmals kein entscheidender Faktor für Open Source ist.

Themenblock Open Source und Gemeinwohl

Frage 3: Beispiele erfolgreicher Open-Source-Projekte für das Gemeinwohl

Können Sie Beispiele für Open Source-Projekte nennen, die in den vergangenen Jahren besonders zum Gemeinwohl beigetragen haben und welche Erfolgsfaktoren und Best Practices lassen sich aus diesen Projekten ableiten? Im Gegenzug: Woran scheitern Open Source-Projekte und Projekte, die auf Open Source-Technologien aufbauen häufig? Welche Fallstricke sehen Sie?

tl;dr: Corona-Warn-App als direkt greifbares Beispiel für ein vorbildliches Open-Source-Projekt mit Wirkung in der breiten Bevölkerung.

curl oder WireGuard als Basiskomponenten, vorangetrieben durch wenige Personen, aber mit Nutzen für die IT weltweit.

Das Zentrum für Digitale Souveränität der Öffentlichen Verwaltung (ZenDiS) und die Sovereign Tech Agency als Entitäten für die Entwicklung und Pflege des Open-Source-Ökosystems.

Erfolgsfaktoren dieser Projekte und Strukturen sind u. a. :

- Open Development von Beginn an
- Einfache Lösungen (Do one thing well Ansatz)
- Modularität der Lösungen
- Transparenz
- Bereitschaft, Erfahrungen zu teilen
- Gezielter Fokus von Maßnahmen

Blocker und Gründe für Scheitern sind oftmals in Projekten:

- Open Source wird nur als nachgelagertes Entwicklungsprinzip gedacht
- Neuerstellung von Komponenten, die in Synergie mehr Sinn ergeben würden, wegen „not invented here“
- Fehlendes Verständnis, dass Software oder unterstützende Strukturen mehr als eine einmalige Beschaffung oder Entwicklung sind und einen Lebenszyklus haben

Ausführliche Antwort:

Die Auswahl der genannten Open-Source-Projekte und Strukturen ist verkürzt und soll nur exemplarische Erfolgsfaktoren für das Gemeinwohl zeigen. Es gibt natürlich noch viel mehr tolle Projekte im Open-Source-Ökosystem.

Herausragendes Beispiel für ein erfolgreiches Open-Source-Projekt ist die Corona-Warn-App. Zum einen wegen ihres Pandemie eindämmenden Effekts, zum anderen wegen der konsequent offenen Entwicklung von Beginn an (Open Development). Dieses offene Vorgehen war ein Grund, warum der CWA so viel Vertrauen auch aus sicherheitskritischen Kreisen entgegengebracht wurde.

Bianca Kastl (bianca.kastl@inoeg.de)

Gute Beispiele für die Anwendung des Unix-Prinzips „do one thing and do it well“ und damit gleichzeitig Beispiel für gute, modulare Lösungen sind Komponenten wie curl oder WireGuard, die von einer kleinen Community entwickelt werden (oft liegt die Hauptarbeit bei einer Person).

curl ist die Standardkomponenten zum Übertragen von Daten von allerlei Protokollen und ist oftmals quasi Geburtshelfer für die Installation anderer Software. Curl wird seit 1998 von Daniel Stenberg maßgeblich vorangetrieben.

WireGuard ist eine schnelle und einfache VPN-Komponente und hat die Einrichtung von VPN-Verbindungen erheblich vereinfacht (weil es nur eine mögliche Übertragungs-Konfiguration gibt). WireGuard ist inzwischen Bestandteil von Heimroutern (FRITZ!Box) zum Aufbau von VPN-Verbindungen von unterwegs, aber auch die Standardkomponente zur Verbindung von Gesundheitseinrichtungen mit Gatewaylösungen der Telematikinfrastruktur. WireGuard wird hauptsächlich von Jason A. Donenfeld entwickelt.

An Strukturen im Open-Source-Ökosystem sei sowohl das Zentrum für Digitale Souveränität der Öffentlichen Verwaltung (ZenDiS) und die Sovereign Tech Agency (bisher Sovereign Tech Fund) als Entitäten für die Entwicklung und Pflege des Open-Source-Ökosystems als Beispiele für einen besonderen Beitrag zum Gemeinwohl genannt.

Beide helfen beim Austausch unter der Open-Source-Community und der Verstetigung und Verbesserung wichtiger Projekte. Diese Transparenz und die Bereitschaft, Erfahrungen teilen zu können, ist im Sinne von Open Source extrem wichtig. Sinnvoll ist auch der Fokus der Sovereign Tech Agency, ganz gezielt bestimmte Softwarekomponenten zu fördern, die sonst der Öffentlichkeit eher nur schwerlich bekannt sein dürften, aber das Rückgrat unser aller digitalen Infrastrukturen und damit der digitalen Daseinsvorsorge bilden.

Oftmals scheitern Open-Source-Projekte daran, wenn Open Source nur als nachgelagertes Prinzip gedacht wird. Wird ein Projekt aber nicht von Beginn an konsequent „in the open“ entwickelt, erreicht ein Softwareprojekt keine gute Nachnutzbarkeit, etwa wegen fehlender kontinuierlicher Dokumentation oder schlecht pflegbarem und nachvollziehbarem Quelltext.

Nicht selten leiden Open-Source-Projekte – auch im staatlichen Kontext – auch an einem „not invented here“-Syndrom, weswegen es viele Open-Source-Projekte nicht schaffen, ein nachhaltiges Ökosystem aufzubauen und dann verwaisen. Aktuell zeichnet sich dieses Problem etwa im Kontext der Entwicklungen des Nationalen Once Only Technical Systems (NOOTS) im Kontext der Registermodernisierung ab, was eigentlich erhebliche Synergien mit den Konzepten für Zero Trust in der Telematikinfrastruktur 2.0 hätte, wenn wir davon ausgehen, dass beide System offengelegt werden letztendlich. Es gibt keinen Grund, warum im Gesundheitswesen und der Verwaltung für sicheren Datenaustausch zwei unterschiedliche Systeme existieren müssten. Am Ende existieren dann aber schlimmstenfalls zwei schlechte gepflegte Insellösungen, die allein nicht sinnvoll weiternutzbar sind. Eine mögliche Synergiebildung findet so nicht statt. In Ländern, die das mit Digitalisierung schon länger machen – Estland etwa – geht das auch mit einem gemeinsamen Open-Source-Technologiestack (X-Road).

Dazu kommt die beschränkte Sichtweise, dass Software etwas sei, was einmal gekauft werde und dann gut sei – zumindest oftmals auch im staatlichen Kontext. Software hat nach der initialen

Entwicklung oder Beschaffung einen Lebenszyklus mit Wartungs- und Sicherheitsupdates, der einen gewissen Prozentsatz des initialen Budgets kontinuierlich benötigen wird. Oftmals sind viele Projekte sich dieser kontinuierlichen Aufgabe nicht bewusst. Gleiches gilt für den Aufbau von unterstützenden Strukturen.

Frage 18: Positive soziale Effekte durch Open Source

Inwiefern kann eine Stärkung der Verbreitung von Open Source Anwendungen auch positive soziale Effekte haben und Grundrechte fördern, und welche Rolle spielen dabei und generell eine hohe Interoperabilität und Maßnahmen zur Erleichterung der Nachnutzung bereits existierender Open Source Software?

tl;dr: Stärkung von Open Source kann aktive Beteiligung der Zivilgesellschaft fördern sowie die Nachhaltigkeit von Lösungen. Das Vertrauen in den Staat kann durch Transparenz gesteigert werden. Interoperabilität und Nachnutzungsmöglichkeiten können dabei sowohl als digitale Teilhabemöglichkeiten als auch Wirtschaftsförderung gesehen werden.

Ausführliche Antwort:

Die Stärkung von Open Source kann eine aktive Beteiligung der Zivilgesellschaft fördern, wie bereits in Frage 14⇲ mit dem Beispiel Gieß den Kiez erwähnt. Da Open Source durch seine Offenheit, Nachnutzbarkeit und Anpassbarkeit immer wieder an neue Gegebenheiten anpassbar sind, sind Open-Source-Lösungen nachhaltiger. Das kann ein stärkeres Engagement für diese Lösungen bewirken.

Durch Open Source kann darüber hinaus **das Vertrauen in den Staat mittels Transparenz** gesteigert werden. Transparenz hinsichtlich Software für Wahlen etwa führt zu weniger Missverständnissen, die das Vertrauen in die Demokratie erodieren könnten, wie etwa der Chaos Computer Club feststellt.

Die Entwicklung von Open Source ermöglicht durch einen offenen Entwicklungsprozess im Open Development auch darüber hinaus eine Demokratisierung der Entwicklung selbst. Neue Funktionen oder Anpassungen können im Entwicklungsprozess bereits mittels Nutzer*innen-Feedback passgenauer priorisiert werden. Dieser Prozess ist aktuell allerdings mit einer technischen Eingangshürde verbunden und muss für das Erreichen der breiten Bevölkerung sicherlich noch in seiner Zugänglichkeit verbessert werden, wie etwa auch der Soziotechnologist tante in einem Podcast zu Open Source anmerkt.

Interoperabilität und Nachnutzungsmöglichkeiten bestehender Open-Source-Lösungen können einerseits als eine Art Wirtschaftsförderung gesehen werden, weil hier Unternehmen Zugang zu Basistechnologien erhalten. Es kann aber auch als ein einfacherer Zugang zu digitalen Leistungen dienen, etwa durch die Möglichkeit der Nutzung anderer Frontends wie etwa Apps oder Automationen, aber auch durch das Hinzufügen weiterer Sprachen. **Das stärkt am Ende die digitalen Teilhabemöglichkeiten.**

Bianca Kastl (bianca.kastl@inoeg.de)

Themenblock Souveränität und Open-Washing

Frage 4: Relevanz von Open-Washing

Für wie relevant halten Sie das Problem des „Open-Washings“, in Anlehnung an „Greenwashing“, also vermeintliche Open Source Entwicklung, die dann schlussendlich doch wieder in proprietärem Code endet? Welche anderen Probleme sehen Sie bei der Entwicklung von Open Source Technologien?

tl;dr: Problem wird relevanter, weil Open Source gleichzeitig im Kontext Digitaler Souveränität gut klingt. Problematisch ist Open-Washing für das Image von Open Source an sich. Problematisch sind auch übermäßiger Gebrauch von Open Core als Lizenzmodell sowie Lizenzänderungen von etablierten Komponenten.

Ausführliche Antwort:

Da Open Source in besonderer Weise auch inhärent der Aspekt der digitalen Souveränität innewohnt, nimmt auch die Zahl der als Open Source angemalten Projekte unweigerlich zu. Das **schadet aber am Ende dem Konzept Open Source** und daher sollte dem entgegengewirkt werden, z. B. durch **klare Benennung von Open-Washing, sobald es auftritt**.

Um aktuelle und vergangene Beispiele klar zu benennen:

- dPhoenixsuite von Dataport – bestand aus Open-Source-Komponenten, war als Gesamtsoftware aber nie wirklich ebenso offen verfügbar zur freien Nachnutzung. Lediglich die Open-Source-Bestandteile wurden angegeben. Inzwischen steht dem Projekt aber durch openDesk ein echter Open-Source-Nachfolger verfügbar.
- Entwurf eines „Closed Group Open Source“ Entwicklungsmodells, etwa im Kontext des Pakt ÖGD. Open Source ist entweder allen dauerhaft frei verfügbar oder es ist kein Open Source und damit dreistes Open-Washing.

Für weitere Beispiele sei auf einen Vortrag von Johannes Näder von der FSFE zu diesem Thema verwiesen.

Problematisch ist im Kontext Open-Washing auch die immer weitere Verbreitung von Open Core Lizenzmodellen, bei denen nur eine Kernfunktionalität einer Software frei verfügbar ist, nicht aber der ganze Funktionsumfang.

Ein weiteres problematisches Beispiel sind **mögliche Lizenzwechsel** von ehemals unter einer OSI geprüften Lizenz zu einer eigenen Lizenz, wie etwa im Kontext von Redis. Hier kann es dann nötig sein, aus der Open-Source-Community einen freien Fork des letzten freien Quelltextstands zu liefern, um anderen Open-Source-Projekten keine Abhängigkeiten aufzubürden. Bei Redis ist das durch Valkey unter Obhut der Linux Foundation geschehen.

Frage 5: Open Source und digitale Souveränität

In welchem Zusammenhang stehen Open Source-Technologien und Fragen der digitalen Souveränität und wäre eine Bevorzugung von Open Source-Technologien in diesem Zusammenhang erstrebenswert – wo liegen konkret die Chancen und Risiken?

tl;dr: Die Bevorzugung von Open Source zum Erreichen von Souveränität ist sinnvoll, aber: Da digitale Souveränität weniger eine Frage der Nationalflagge an Software ist, sondern die Möglichkeit, Software und deren technische Grundlagenkomponenten vollständig durchblicken und deren explizite und implizite Abhängigkeiten soweit wie möglich steuern zu können, definiert sich Souveränität eher an Offenheit, Austauschbarkeit und Reproduzierbarkeit. Ein Neuaufbau von Komponenten ist in diesem Zuge teilweise notwendig und mühsam, führt aber zu stärkerer Unabhängigkeit, besserer Möglichkeit der Synergiebildung, Schaffung eines Ökosystems und Minimierung von unausweichlichen Abhängigkeiten.

Ausführliche Antwort:

Der Begriff der digitalen Souveränität ist als solcher nicht unumstritten, soll in diesem Kontext aber mit „volle Kontrolle über Software in all ihren Abhängigkeiten haben“ umschrieben werden und weniger, ob es sich um eine deutsche, europäische oder wie auch immer geartete Software handelt.

Nach dieser Definition von Souveränität ist aber zum Beispiel die Delos Cloud zum möglichen „souveränen“ Einsatz von Microsoft-Cloud-Produkten z. B. in der Verwaltung eben nicht souverän. Das ist darin begründet, da hier die Abhängigkeit zu Microsoft sogar noch weiter manifestiert wird, weil hier proprietäre Microsoft-Software vorab aufwändig geprüft wird und in einem speziellen Rechenzentrum unter anderer Hoheit als der von Microsoft nachgebaut wird. Damit handelt es sich hier um das Gegenteil von Souveränität. Im Prinzip wird hier einem Produkt der Anschein von Souveränität gegeben, mit Aufpreis zu einer ohnehin vorhandenen Abhängigkeit. Das ist nicht souverän, das ist Etikettenschwindel.

Open Source kann einen Beitrag zur digitalen Souveränität leisten, da hier richtig und konsequent alle Abhängigkeiten weitgehend auflösbar sind. Diese Abhängigkeiten können etwa die Abhängigkeit von einer Betriebsplattform, Festlegung auf bestimmte Dateiformate oder die Einschränkung auf bestimmte Abrechnungsmodelle sein. Open-Source-Software kann, richtig angewendet, hier alle Abhängigkeiten auflösen. Dafür notwendig ist die Offenheit, Austauschbarkeit und Reproduzierbarkeit von Software, die bei gut gemachter Open-Source-Software implizit vorliegen sollte.

Die Erlangung dieser Art von Souveränität kann aber den Neuaufbau von Software(-Komponenten) notwendig machen, teils Reverse Engineering erforderlich machen, etwa um proprietäre und nicht dokumentierte Dateiformate auszutauschen. Der Prozess der Erlangung von Souveränität kann also aufwändig sein und ist als Selbstzweck nicht immer zielführend. Die Vorteile sind ganz klar die stärkere Unabhängigkeit, bessere Möglichkeit der Synergiebildung mit anderer Software, die Schaffung eines Ökosystems und sowie **Minimierung von unausweichlichen Abhängigkeiten, was eine finanzielle oder geopolitische Abhängigkeit und damit zusammenhängende Lock-in-Effekte** reduziert.

Themenblock Vergabe und Förderung von Open Source in der Beschaffung

Frage 7: Vergabekriterien und Mindestanteil Open Source

Welche Vergabekriterien sollten im Vergaberecht mit Blick auf die Beschaffung digitaler Produkte und Dienstleistungen reformiert werden und welche Gründe sprechen dafür oder dagegen, hier einen Mindestanteil von Open Source-Technologien einzuführen?

tl;dr: Open Source by default, zumindest für Individualentwicklungen. Durchsetzung über weitere Rahmenbedingungen, wie etwa Förderbedingungen. Gründe dafür: Es gibt bereits einen versteckten Mindestanteil Open Source in proprietären Produkten, sowie weitere Gründe aus Frage 1↘.

Ausführliche Antwort:

Vergabeverfahren können eigentlich eine Forderung nach „Public Money, public code“ als feste Bedingung enthalten, zumindest bei neuen Individualentwicklungen.

Dafür gibt es folgende mögliche Rationale:

Prinzipiell gibt es im Sinne eines Vergabeverfahrens Open Source eigentlich zumindest bei neuen Individualentwicklungen keinen formalen Grund gegen Open Source. Denn eigentlich ist der Sinn eines Vergabeverfahrens laut Bundeskartellamt:

> *Das Ziel des Vergaberechts ist die wirtschaftliche Verwendung von Haushaltsmitteln, aber auch der Schutz eines fairen Wettbewerbs zwischen den Unternehmen sowie die Gewährung eines freien Marktzugangs im europäischen Binnenmarkt.*

Wird eine neue Individualsoftware vergeben, stellt ein Einfordern von Open Source eigentlich einen zusätzlichen Schutz des fairen Wettbewerbs dar. Durch das Offenlegen des Quelltexts ist der Auftraggeber nach einem Projekt(abschnitt) nicht mehr an einen bestimmten Dienstleister gebunden und kann Folgeaufträge wieder im fairen Wettbewerb zu gleichen Ausgangsbedingungen für alle Beteiligten vergeben. Bei proprietärer Software ist das keinesfalls so.

Demnach könnte aber eine Nichtoffenlegung des Quelltexts wie bei Open Source als Wettbewerbsverzerrung verstanden werden für Folgeaufträge. Ein davon abweichendes Vorgehen müsste im Sinne einer Beweislastumkehr begründet werden.

In Verweis auf Frage 8↘ ist eine Begründung der Nicht-Offenlegung des Quelltextes wegen Sicherheit keine hinreichende Begründung gegen Open Source.

Oftmals bieten auch weitere Richtlinien und Bedingungen für Softwareprojekte weitere Begründungen für Open Source. Das können etwa Förderrichtlinien sein, die die Zuwendung von Förderung an die freie Nachnutzung der Ergebnisse binden, wie etwa im Kontext des Pakt ÖGD formuliert. Es können aber auch eigentlich verbindliche IT-Richtlinien wie die föderalen Architekturrichtlinien sein. §4 des Onlinezugangsgesetz mit seiner Formulierung, nach der „Open-Source-Software vorrangig vor solcher Software eingesetzt werden“ soll, ist ein ähnlicher Hebel.

Es gibt keine Gründe gegen einen Mindestanteil Open Source, denn dieser Mindestanteil Open Source liegt auch in proprietären Produkten bereits vor. Die Hinweise zu Fremdbibliotheken zu

Microsoft Word für Mac enthalten bereits Referenzen unterschiedlicher Bibliotheken unter diversen Open-Source-Lizenzen, darunter etwa Softwarekomponenten wie SQLite. Ähnliches gilt auch für viele andere proprietäre Softwareprodukte. Die als Log4Shell bekannte Sicherheitslücke betraf etwa einen beachtlichen Teil der Cisco-Produktpalette.

Es ist also nicht die Frage, ob es einen Mindestanteil an Open Source in Software geben soll, sondern eher die Frage, auf welchen Prozentwert dieser möglicherweise verbindlich gesetzt werden sollen unter welchen Umständen. Damit sprechen auch keine Gründe dagegen, da dies nur eine Formalisierung der Realität wäre.

Frage 10: Möglichkeiten zur Förderung von Open Source in Vergabeverfahren

Welche vergaberechtlichen und verwaltungsrechtlichen Möglichkeiten werden derzeit nicht ausreichend genutzt, um den Einsatz von Open Source Software im staatlichen Bereich zu fördern und proprietäre Software perspektivisch durch quelloffene Alternativen zu ersetzen? Welche zusätzlichen gesetzlichen Vorgaben wären wünschenswert, um diesen Übergang zu unterstützen?

tl;dr: Aus der Erfahrung mit Vergaben sind Teilnahmewettbewerbe mit Verhandlungsverfahren besser geeignet für die Findung geeigneter Partner für Open-Source-Software. Oftmals fehlt eine Open Source spezifische Definition der eigenen Vergabekriterien (z. B. Bewertung eines Entwicklungs-Community-Konzepts, Testkonzepten und des Entwicklungsvorgehens). Als Auftraggeber muss Quelltext in der Form eingefordert werden, dass dieser mindestens am Ende einer Projektphase von anderen Dienstleistern weiter pflegbar ist. Ebenso sollte der Umgang von Upstream-Beiträgen zu anderen Projekten geklärt werden.

Für die gesetzlichen Empfehlungen sei auf Frage 7⇲ verwiesen.

Ausführliche Antwort:

Oft wird Software in der Beschaffung noch mittels einfacher Bewertungsverfahren in offenen Verfahren beschafft. Dabei werden Angebote dann z. B. nach nur zwei Kriterien wie Preis und Leistung bewertet, meist in einfachen Verhältnissen zueinander.

Aus der eigenen Erfahrung ist die Entwicklung und der Betrieb von Open Source und dafür notwendige Auswahl geeigneter Dienstleister aber an weitere Kriterien gebunden, die eine Zusammenarbeit erst zielbringend möglich machen.

So können **Bewertungskriterien auch für Open Source wichtige Aspekte** wie ein geplantes Testkonzept, Dokumentationsstandards oder die Bewertung eines Entwicklungs-Communitykonzepts umfassen. Hierfür braucht es zwar formal sachlich vergleichbare Faktoren, die über einen Preisvergleich oder eine Summe von Leistungspunkten hinausgehen. Aber letztendlich schaffen diese zusätzlichen Kriterien eine bessere Auswahlmöglichkeit eines Dienstleisters, der auch als Open Source gut nachnutzbare Software möglich macht.

Prozessual sind hier oftmals mehrstufige Vergabeverfahren, etwa Teilnahmewettbewerbe mit Verhandlungsverfahren zur Auswahl des passenden Dienstleisters eher geeignet. Einerseits kann hier eine Vorauswahl geeigneter Anbieter getroffen werden anhand einfacher Kriterien wie passender Referenzen in den letzten Jahren und dahinterliegender Projektgrößen. Andererseits kann im

Teilnahmewettbewerb gemeinsam das gewünschte gemeinsame Ziel besser gemeinsam vor einem finalen Angebot im Austausch definiert werden.

Nach dem Zuschlag muss ein Auftraggeber als Verantwortliche für ein Open-Source-Projekt aber auch kontinuierlich bestimmte Artefakte einfordern. Das betrifft im Wesentlichen den Anspruch, dass der Quelltext eines Softwareprojekts bereits früh im Projekt veröffentlicht werden kann, so früh wie möglich. Dazu gehört auch eine entsprechend geeignete, kontinuierliche Dokumentation. Das Ziel muss speziell bei Open-Source-Projekten sein, dass das Projekt ab einem gewissen Zeitpunkt durch gute Dokumentation, nachvollziehbaren und klaren Code sowie alle notwendigen Build-Prozesse jederzeit an einen beliebigen anderen geeigneten Dienstleister übergeben werden könnte. Dieser Anspruch steht im krassen Gegensatz zu proprietären Produkten.

Ebenso muss die **Verwendung von offenen Schnittstellen und offenen Standards** eingefordert werden. Werden in einem Softwareprojekte neue Standards oder Schnittstellen geschaffen, müssen diese ebenfalls offengelegt und dokumentiert werden.

Abschließend wird in Vergaben von Open-Source-Projekten oftmals nicht hinreichend geklärt, wie mit **Upstream-Beiträgen zu anderen Projekten,** also zum Beispiel Fehlerbehebungen oder neue Funktionen, die im Rahmen des Ausgangsprojekts an einem eingebundenen Projekt durchgeführt werden. Hier sollten seitens des Auftraggebers zumindest die Möglichkeit gestattet werden, diese Beiträge Upstream zurückzugeben, bzw. sogar eine Verpflichtung definiert werden.

Bianca Kastl (bianca.kastl@inoeg.de)

Themenbereich Open Source, Gemeinnützigkeit und Insitutionalisierung

Frage 11: Open Source und Gemeinnützigkeit

Welche Auswirkungen und Folgen sehen Sie voraus für den Fall, dass die Entwicklung und der Betrieb quelloffener Software als gemeinnütziger Zweck in der Abgabenordnung aufgenommen wird? Halten Sie dies für wünschenswert?

tl;dr: Eine Aufnahme von Entwicklung und Betrieb quelloffener Software als gemeinnütziger Zweck ist wünschenswert. Dies kann den Wirtschaftsstandort stärken, aber auch in Hinblick auf den Cyber Resilience Act die Pflege und Haftung für Software-Komponenten vereinfachen, die Hersteller im Sinne des CRA als Open-Source-Komponenten nutzen. Eine Abgrenzung der Gemeinnützigkeit von kommerzieller Aktivität kann im Sinne des CRA gezogen werden.

Ausführliche Antwort:

Ende April 2024 hat die gemeinnützige Organisation hinter dem dezentralen Microblogging-Dienst Mastodons die Gemeinnützigkeit verloren. Eigentlich ist Mastodon alles, was wir uns im Sinne des Gemeinwohls von einem Sozialen Netzwerk wünschen würden: offener Quelltext, kostenloser und dezentraler Betrieb in Eigenregie möglich, keine Algorithmen wie bei anderen Sozialen Netzwerken. Die Entwicklung und der Betrieb quelloffener Software sind aber bisher kein gemeinnütziger Zweck im Sinne der Abgabenordnung. Mastodon wanderte deshalb organisatorisch in die USA ab zur Gründung einer 501(c)(3) non-profit entity.

Dieser Fall ist bedauerlich und zeigt die Notwendigkeit, Open Source in Entwicklung und Betrieb als möglichen gemeinnützigen Zweck dringend zu überdenken. **Eine Aufnahme von Entwicklung und Betrieb quelloffener Software als gemeinnütziger Zweck in der Abgabenordnung ist daher mehr als wünschenswert.**

Dies würde einerseits den Wirtschaftsstandort in Deutschland stärken, weil es vielen Einzelorganisationen, die sich seit langem in ihrer Freizeit oder im Nebenerwerb bei der Entwicklung und dem Betrieb von Open Source ohne Gewinnerzielungsabsicht engagieren, bessere Handlungsspielräume bietet und am Ende auch eine Art von Anerkennung ihrer gemeinwohlorientierten Arbeit von Seiten des Staates darstellt. Das Open-Source-Ökosystem ist als Ganzes auf eine enge und funktionierende Kooperation von gemeinnützigen und kommerziellen Akteuren angewiesen.

In Hinblick auf den Cyber Resilience Act (CRA) kann die Gemeinnützigkeit von Open Source zusätzlich die Kooperation von kommerziellen Akteuren und dem Open-Source-Ökosystem vereinfachen und damit Bürokratie abbauen. Eine Gemeinnützigkeit von Open Source bietet kommerziellen Akteuren eine niederschwellige Möglichkeit, bei gemeinnützigen Akteuren Fehlerbehebungen oder neue Funktionen finanziell zu belohnen.

Im Sinne der mit dem CRA stärker notwendigen Produkthaftung wären solche Anforderungen an gemeinnützige Akteure ohne Gemeinnützigkeit von Open Source mit vielen möglichen Individualverträgen zur Bezahlung oder Einfordern kostenloser Arbeit bei gemeinnützigen Akteuren

Bianca Kastl (bianca.kastl@inoeg.de)

verbunden, um die Anforderungen die Produkthaftung seitens kommerzieller Anbieter nach CRA zu erfüllen. Diese Optionen sind beide in der Breite nicht tragfähig.

Eine Gemeinnützigkeit von Open Source könnte hier im Open-Source-Ökosystem bereits etablierte Konstrukte wie Feature Bounties oder Bug Bounties organisatorisch nutzen, um eine gemeinsame Verbindlichkeit im Sinne des CRA zur Fehlerbehebung zu schaffen, gleichzeitig durch die Gemeinnützigkeit aber auch die Finanzierung mittels Spenden vereinfachen und steuerlich korrekt regeln.

Eine **Abgrenzung von gemeinnützigen und kommerziellen Open-Source-Anbietern kann dabei nach CRA** durchgeführt werden (Manufacturers versus Open Source Software Stewards). Durch die hohen Auflagen hinter der Gemeinnützigkeit nach Abgabenordnung ist auch von keinem Missbrauch der Gemeinnützigkeit auszugehen.

Der InÖG hat zu weiteren Aspekten des CRA im Hinblick auf das Open-Source-Ökosystem ein Thesenpapier veröffentlicht und bringt sich im BSI Dialog für Cybersicherheit im Workstream „Cybersicherheit in der Gesellschaft – die Rolle des CRA für Open Source“ mit ein.

Frage 12: Strukturen zur Förderung von Open Source

Welche institutionellen Strukturen, wie z.B. Stiftungen oder NGOs wären im Bereich der Open Source Förderung wünschenswert und welche Aufgaben oder Ziele sollten diese hypothetischen Strukturen erreichen?

tl;dr: Ergänzend zu einer eher näher an ausgewählten Projekten liegenden Struktur wie der Sovereign Tech Agency, ist eine Stiftung für Open-Source-Software anzuraten, welche im Kontext von Bund, Ländern und Kommunen verwendet wird. Diese kann die Rolle eines Open Source Software Stewards für die vom Bund entwickelte Open-Source-Software einnehmen sowie Unterstützung von freier und offener Software (FOSS) anbieten, deren kontinuierliche Pflege im Sinne des Bundes ist. Üblicherweise bieten keine geeigneten institutionalisierten Strukturen, die ein Open-Source-Ökosystem wegen der jeweiligen Trägheit von Beschaffungsprozessen sinnvoll unterstützen könnten.

Ausführliche Antwort:

Wünschenswert wäre nach dem Vorbild der etablierten Stiftungen wie der Linux Foundation **eine bundesweite Stiftung für Open Source, die langfristig die Pflege von Softwarekomponenten der öffentlichen Hand unterstützen kann**. Das können eigene Softwareprojekte der öffentlichen Verwaltung, aber auch externe Softwareabhängigkeiten sein. Diese Stiftung könnte im Sinne des Cyber Resilience Act als **Open-Source-Software Steward für den Code der öffentlichen Hand** dienen. Dies beinhaltet etwa Verwaltungen des Bundes, der Länder und der Kommunen, kann auch aber andere Bereiche umfassen wie schulische Bildung oder den Bildungsbereich allgemein, das Gesundheitswesen, aber auch Angebote im Bereich Kultur, wie etwa die KulturPass App. Damit wäre nicht nur finanzielle, sondern auch rechtliche Sicherheit für Open-Source-Projekte gegeben.

Open-Source-Projekte in der öffentlichen Hand oder für die öffentliche Hand haben oftmals Schwierigkeiten mit der Verstetigung, da Beschaffungsprozesse für Fehlerkorrekturen oder Funktionserweiterungen von Open-Source-Software oftmals langwierig sind und die jeweilige Haushaltslage unbestimmt ist. Rahmenverträge für eine langfristige Pflege mit hinreichender Planungssicherheit sind oftmals nur für große kommerzielle Anbieter wegen bestimmter unternehmerischen Leistungsanforderungen machbar.

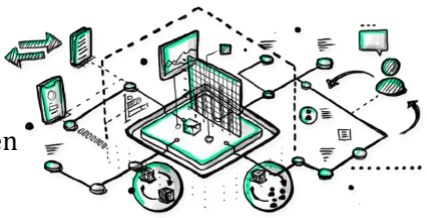
Eine Stiftung kann hier ein Open-Source-Ökosystem mit Softwareprojekten ganz unterschiedlicher Größe längerfristig stabil unterstützen und kann damit eine Anlaufstelle auch für kleine Open-Source-Projekte sein.

Das würde den Ansatz der Sovereign Tech Agency, die sich gezielt auf wichtige Projekte im Open-Source-Ökosystem fokussiert, sinnvoll ergänzen, da hier der Fokus nicht bestimmte wichtige Projekte für das ganze Open-Source-Ökosystem, sondern gezielt der Einsatz in der öffentlichen Verwaltung der Schwerpunkt wäre. Sovereign Tech Agency und die Stiftung würden sich so gut ergänzen.

Eine derartige Stiftung kann auch Länderkooperationen über Zuständigkeitsgrenzen hinweg vereinfachen, ohne gleich neue Organisationen oder Staatsverträge schaffen zu müssen. Ein Beispiel für eine mögliche Zusammenarbeit im Projektkontext, welche eine solche Stiftung mittels Open Source vereinfachen könnte, wäre etwa die Vereinbarung zwischen Nordrhein-Westfalen und Sachsen aus dem Jahr 2020 zur elektronischen Ermöglichung von Bürgerbeteiligung.

In der Position als Open Source Steward kann eine solche Stiftung kommerziellen Unternehmen bei der Erfüllung der Pflichten nach CRA unterstützen, in dem sie bei der Bereitstellung von vertrauenswürdigem Code durch einzelne Open-Source-Maintainer hilft und einen niedrigschwelligen Finanzierungsrahmen bietet. Dies ist aber idealerweise im Verbund mit einer Gemeinnützigkeit von Open Source, wie in Frage 11↘ erläutert, zu sehen.

Der Fokus einer solchen Stiftung kann auch die Unterstützung von Open Source in im Bereich der akademischen Lehre und wissenschaftlicher Einrichtungen umfassen, etwa der GWDG als gemeinsame Einrichtung der Universität Göttingen und der Max-Planck-Gesellschaft oder dem Deutschen Forschungsnetz (DFN).



Bianca Kastl (bianca.kastl@inoeg.de)

Themenbereich Open Source und IT-Sicherheit

Frage 8: Cybersicherheit und Open Source

Wie bewerten Sie die Fragen der Cybersicherheit im Kontext von Open-Source-Technologien, insbesondere mit Blick auf den Einsatz in öffentlichen Verwaltungen?

tl;dr: Open Source ist besser absicherbar als proprietäre Software. Schwierigkeit beim Einsatz in der öffentlichen Verwaltung: Oft fehlende schnelle Aktualisierungsfähigkeit, oft fehlende Grundkompetenzen im Umgang mit Open-Source-Systemen, oft Fehlen von vollständigen Softwarestücklisten mit Unterabhängigkeiten. Oft kein DevSecOps.

Ausführliche Antwort:

Wie bereits in Frage 1⇲ erwähnt, ist Open-Source-Software besser absicherbar als proprietäre Software. **Sicher ist Software dann, wenn Software sicher ist. Open Source bietet als Entwicklungsvorgehen mehr Möglichkeiten, Software auch wirklich sicher durch Audits und Hinweise Dritter zu bekommen als etwa proprietäre Software.**

Befürchtungen, dass das Offenlegen des Quelltexts von Software zu mehr Unsicherheit führt, sind nur eine andere Umschreibung für „meine Software ist unsicher“. Ein Verbergen des Quelltexts führt in den meisten Einsatzzwecken nicht zu mehr Sicherheit und widerspricht etablierten Prinzipien aus der Kryptografie etwa den Grundsätzen von Auguste Kerckhoff:

> *Es darf nicht der Geheimhaltung bedürfen und soll ohne Schaden in Feindeshand fallen können.*

Für den überwiegenden Großteil der Anwendungen in der öffentlichen Verwaltung ist dieses Prinzip uneingeschränkt anwendbar, etwa auch im Gesundheitswesen mit Daten mit hohem oder sehr hohem Schutzbedarf.

Hinsichtlich des Betriebs von Software muss die Situation der IT-Sicherheit differenzierter betrachtet werden, weil hier ganz losgelöst vom Lizenzmodell in der öffentlichen Verwaltung noch andere Schwierigkeiten vorliegen. Die erwähnten Probleme gelten auch für proprietäre Software und sind nicht Open Source spezifisch:

- Basierend auf möglichen Gefährdungslagen, etwa nach Baustein Allgemeiner IT-Betrieb des BSI IT-Grundschutzkompendiums, liegen in öffentlichen Verwaltungen oftmals unzureichende Personalkapazitäten für sicheren IT-Betrieb vor.
- Oftmals ist die Kompetenz in der öffentlichen Verwaltung nicht auf den Betrieb von Open-Source-Software ausgerichtet, oft ist die IT-Infrastruktur Windows-fokussiert, tiefgreifende Linux-Kenntnisse oder Kenntnisse anderer Betriebssysteme sind nicht immer vorhanden.
- Updateprozesse, um wichtige Sicherheitsupdates an alle Verwaltungssysteme zu verteilen, sind oftmals nicht schnell genug, um z. B. auf Zero-Day-Schwachstellen adäquat reagieren zu können. Generell ist hier eine Updatefähigkeit innerhalb von weniger als 24 Stunden eine für alle Teile der Verwaltung anzustrebende Zielmarke.

Bianca Kastl (bianca.kastl@inoeg.de)

- Oftmals fehlen aktuelle Softwarestücklisten mit Unterabhängigkeiten, etwa in Form von Software Bill of Materials. Softwarestücklisten haben bei Open-Source-Software eine höhere Relevanz als bei proprietärer Software wegen den oftmals umfangreicheren Unterabhängigkeiten, sind aber auch im Kontext proprietärer Software relevant.
- DevSecOps-Ansätze zum sicheren Betrieb von Software, aber auch Best Practices zur sicheren Softwareentwicklung sind oftmals nicht etabliert bzw. werden oftmals nicht von Dienstleistern eingefordert.

All diese Punkte wären – unabhängig von der Frage Open Source – notwendig, um ein Mindestmaß an IT-Sicherheit zu erreichen. Zuträglich wäre hier auch eine Berücksichtigung eine Umsetzung der NIS-2-Richtlinie für die Kernverwaltung der Kommunen und damit das Zurücknehmen bisher möglicher Ausnahmen.

Frage 13: Advisory Board auf Bundesebene

Sollte auf Bundesebene ein Open-Source-Advisory-Board initiiert werden, von dem aus auch OS-Entwicklungen monitored werden, um Probleme wie in der Vergangenheit (Log4j-Attacke) zu minimieren?

tl;dr: Im Hinblick auf IT-Sicherheit: Ein Advisory Board sendet hier wahrscheinlich wie das bisherige staatliche Vorgehen in den Bereich IT-Sicherheit und Open Source ambivalente Signale. Unterstützung bestehender, vertrauenswürdiger Entitäten sinnvoll. Das aktuelle staatliche Handeln im Umgang mit Sicherheitslücken ist der Sicherheit eher abträglich (Hackerparagraph, Abhängigkeit des BSI / Schwachstellenmanagement).

Im Hinblick auf politische und organisatorische Rahmenbedingungen: Hier kann ein Advisory-Board helfen, abzuzeichnende Probleme, die Sicherheitsprobleme begünstigen, zu minimieren.

Mögliche Werkzeuge: Open-Source-Politikfolgenabschätzung, Förderung Präventionsmaßnahmen, auch im staatlichen Kontext, Beachtung bereits erwähnter Maßnahmenempfehlungen.

Ausführliche Antwort:

Ein Open-Source-Advisory-Board auf Bundesebene muss differenziert betrachtet werden. Log4Shell hat ein paar Probleme gezeigt, die für die Bewertung eines Advisory-Boards mit einbezogen werden sollten.

Im Hinblick auf den Bereich der operativen IT-Sicherheit, dem sich Kümmern um eigentliche Lücken also, konnten wir im Zusammenhang mit dem Workstream BuntesBugBounty (B3) des BSI Dialog für Cybersicherheit mit Christian Grobmeier vom Log4j-Team sprechen und etwas mehr Einblicke in die Zeit bekommen, als die Lücke Log4Shell (CVE-2021-44228) bekannt wurde.

Ein wesentlicher Satz aus diesem Vortrag sei hier erwähnt, weil er auch die mögliche Sinnhaftigkeit eines staatlichen Advisory-Boards aufzeigt:

When you report security issues, your government may harm you.

Letztlich wurde im Kontext Log4Shell auch der Melder Chen Zhaojun von Alibaba Cloud bestraft, weil er die Lücke direkt den Open-Source-Maintainern gemeldet hatte.

Bianca Kastl (bianca.kastl@inoeg.de)

Überträgt man diese Situation auf Deutschland, so ergibt sich ein ähnliches Problem in Hinblick auf ein mögliches beratendes Gremium: Durch bisher ungelöste politische Richtungsentscheidungen im Bereich der IT-Sicherheit wie die noch nicht erfolgte Abschaffung des sogenannten Hackerparagraphen oder die immer wieder im Raum stehende Option, Sicherheitslücken zurückhalten zu wollen im Sinne eines „Schwachstellenmanagements“ für mögliche „Hackbacks“, sendet so ein Advisory Board sehr ambivalente Signale.

Das aktuelle staatliche Handeln im Umgang mit Sicherheitslücken ist der Sicherheit eher abträglich, weil ein staatlich unterstütztes Advisory-Board hier zumindest von Seiten der Open-Source-Community zumindest mehr als kritisch beäugt werden würde. Am Ende muss es letztlich bei all diesen Maßnahmen darum gehen, Sicherheitslücken so schnell wie möglich ausnahmslos zu schließen. Eine Meldung, die zur Schließung von Sicherheitslücken führt, darf dabei nicht wie etwa im Fall Modern Solution mit einer Verurteilung enden.

Weitaus sinnvoller erscheint hier eine Stärkung bereits bestehender, in der Open-Source-Community akzeptierter Entitäten wie der Sovereign Tech Agency, um solche Angebote in Richtung Community anzubieten.

Ein Advisory-Board könnte aber eher auf einer Metaebene in politischer und organisatorischer Hinsicht unterstützen, bessere Prävention auf Seiten z. B. der Verwaltung durch die Maßnahmen unter Frage 8↘ forcieren.

Darüber hinaus könnte **ein Advisory-Board die politischen Rahmenbedingungen hinsichtlich möglichst effektiver Schwachstellenbehebung** über die Phasen Meldung, Behebung, Erkennung der eigenen Betroffenheit und Deployment von Updates in betroffene Systeme begleiten. Dies wäre also eher eine Beratung auf politischer und organisatorischer Ebene im Sinne einer Open-Source-Politikfolgenabschätzung. Damit sollten sich auch potenziell für die Open-Source-Community schädliche Regulierungen, wie ursprünglich im Cyber Resilience Act angedacht, zukünftig verhindern. Ein Advisory-Board sollte daher aber im engen Austausch z.B. mit einer möglichen Stiftung, wie in Frage 12↘ angedeutet, agieren.

Frage 15: Open Source und Protestware

Bei der Entwicklung von Open Source Software (OSS) kann durchaus auch unbemerkt Schad-Software eingebaut werden, zB ist dann von sogenannter Protestware die Rede. Wie sicher ist OSS im Vergleich zu proprietärer Software, gibt es dazu empirische Befunde, wer haftet für etwaige Folgeschäden und mit welcher Zunahme von Protestware rechnen Sie, angesichts des allgegenwärtigen Aktivismus der sogenannten Zivilgesellschaft?

tl;dr: Open Source wirkt im Kontext Protestware eher eindämmend im Vergleich zu proprietärer Software – wegen Offenlegung des Quelltexts sowie des Entstehungsprozesses. Empirisches Beispiel wäre die Backdoor in XZ Utils (CVE-2024-3094). Keine besondere Zunahme von Protestware zu erwarten. Haften würden eigentlich Akteure, die mutwillig zerstörerische Änderungen eingebracht haben, meist ist diesen Akteuren Haftung aber ohnehin egal.

Bianca Kastl (bianca.kastl@inoeg.de)

Ausführliche Antwort:

Als Teil der Zivilgesellschaft, die für die Freiheitliche demokratische Grundordnung eintritt, sind die Aktivitäten hinsichtlich Protestware im Zusammenhang mit Open Source folgendermaßen zu bewerten:

Aufgrund **des offenen Quelltexts und offenen Entstehungsprozesses durch eine offene Historie von Änderungen hemmt Open Source Protestware** in höherem Maße, als eine Öffnung des Quelltexts Protestware begünstigen würde. Prinzipiell sind aus Sichtweise eines Angreifers nach Bedrohungsmodellierung speziell der Aspekt der Nichtabstreitbarkeit einer Änderung durch eine öffentliche Versionsverwaltungssoftware eher sehr hinderlich. Bösartige Akteure agieren eher so lange wie möglich unentdeckt im Hintergrund. So schnell wie eine Änderung scheinbar eingebaut zu sein scheint, würde sie durch öffentliche Einsicht in den Code auch wieder auffallen.

Üblicherweise agieren bösartige Akteure im staatlichen Umfeld (APTs) im Verborgenen, was auch das Vorgehen im Kontext der Backdoor in XZ Utils gut aufzeigt. Der Backdoor ging eine jahrelange Vorbereitung voraus, wobei die eigentliche Backdoor dann durch Mitwirken der Open-Source-Community innerhalb von wenigen Stunden gefunden und innerhalb von Tagen weitgehend vollständig aufgelöst wurde.

Somit gilt, wie schon erwähnt in Frage 1⇲, dass Open-Source-Software besser absicherbar ist.

Bei mutwilliger bösartiger Anpassung einer Software würde die in Frage 1⇲ beschriebene „As is“-Haftungsklausel nicht greifen.

Allerdings ist angesichts der Akteure, die solche bösartigen Veränderungen möglicherweise durchführen, etwa staatlich geduldete Akteure, nicht unbedingt davon auszugehen, dass diese Haftung rechtlich auch vollstreckt werden kann. Die Frage der Haftung ist hier eine eher nachgelagerte Frage im Vergleich zu dem möglichen Schaden im Bereich der Informationssicherheit.

Ich gehe von **keiner besonderen Zunahme von Protestware** im Vergleich zur allgemeinen Cyber-Bedrohungslage aus.

Bianca Kastl (bianca.kastl@inoeg.de)

Themenfeld Open Source und sogenannte Künstliche Intelligenz

Frage 16: Open Source im Bereich generative KI

Die Bildgenerierungssoftware Stable Diffusion ist eine quelloffene Lösung, die ähnlich gute und verblüffende Ergebnisse liefert wie ihre proprietären Pendants; gleiches gilt für den Textgenerator Mistral. Wäre es aus Ihrer Sicht möglich, im Bereich generativer KI mit quelloffenen Lösungen die sich abzeichnenden Oligopole der großen Technologiekonzerne zu brechen?

tl;dr: Nein, die Oligopole basieren auf multiplen Faktoren wie Konzentration von Rechenleistung, teils skrupelloser Anhäufung von Trainingsdaten und immensen Kapitaleinsatz.

Ausführliche Antwort:

Die **Oligopole der großen Technologiekonzerne basieren nicht nur auf dem Faktor Software**, sondern auf multiplen Faktoren abseits der Offenheit der Software:

- Neben der Software werden Unmengen an Trainingsdaten benötigt, welche teils skrupellos ohne Rücksicht auf Copyrights abgezogen werden.
- Es werden Unmengen an Rechenleistung benötigt, deren Energiebedarf für das Umfeld der jeweiligen Rechenzentren eine enorme Belastung darstellen kann.
- Die Kosten für das Training großer Modelle steigen kontinuierlich an und benötigen immer mehr Kapitaleinsatz.
- Für das Training von großen Modellen ist zudem billige menschliche Arbeitskraft notwendig, meist aus Staaten mit Arbeitsbedingungen, die an Kolonialismus erinnern.

Damit ist – ungeachtet der problematischen Definition von „quelloffener KI“ – die Marktmacht großer Technologiekonzerne mit der Offenlegung des Quelltexts nicht zu brechen und ethisch und moralisch schon gar nicht in irgendeiner Art erstrebenswert.

Bianca Kastl (bianca.kastl@inoeg.de)

Bezug der Sachverständigen zum Themengebiet

Bianca Kastl ist 1. Vorsitzende des Innovationsverbund Öffentliche Gesundheit und engagiert sich für eine bessere Digitalisierung von Verwaltung und Gesundheitswesen in Deutschland.

Ihr fachlicher Schwerpunkt sind skalierende, sichere digitale Infrastrukturen, Systemarchitekturen, Cloud native Anwendungen, IT-Security mit dem Fokus auf Zero Trust Prinzipien, Privacy sowie Barrierefreiheit und User Experience, seit Jahren in Open Source

Für den InÖG hat sie in der Pandemie IRIS connect, eine Open Source Kommunikationsinfrastruktur im öffentlichen Gesundheitsdienst, als Betriebsverantwortliche von 2021 an begleitet.

Beruflich ist sie als Tech Lead und Chief Product Owner im Öffentlichen Dienst tätig und entwickelt vernetzte Anwendungsplattformen unter dem Markenmamen GA-Lotse für den Öffentlichen Gesundheitsdienst nach Stand der Technik in Hessen aus der Verwaltung heraus für die Verwaltung.

Über den Innovationsverbund Öffentliche Gesundheit e. V. (InÖG)

Der Innovationsverbund Öffentliche Gesundheit (InÖG) entstand 2020 aus einem Zusammenschluss von Projekten, die sich im Rahmen des #WirVsVirus Hackathons unter der Schirmherrschaft des Bundeskanzleramts verknüpft haben. In der Tradition etablierter zivilgesellschaftlicher freier Träger und in Anlehnung an das THW wird der Öffentliche Gesundheitsdienst als Schnittstelle von Verwaltung und Gesundheitswesen gezielt und nachhaltig mit Open Source Technologie unterstützt. In Zusammenarbeit mit der Björn Steiger Stiftung entstand als erstes Digitalprojekt IRIS connect. IRIS connect ist eine Open Source Kommunikationsinfrastruktur im öffentlichen Gesundheitsdienst und wurde in der Pandemie über vier Bundesländer (HE, NRW, TH, SA) hinweg eingesetzt.

Darüber hinaus hat der InÖG eine Privatsphäre-freundliche, deutschlandweit skalierbare Impfplattform in der Pandemie entwickelt.

In Kooperation mit dem Dialog für Cybersicherheit des BSI hat der InÖG im Jahr 2022 /2023 den Workstream BuntesBugBounty (B3) begleitet, der Bug-Bounty-Programms für die IT-Systeme der öffentlichen Hand sowie Freie und Open-Source-Software (FOSS) im Dialog mit Vertretern aus Open-Source-Community und staatlichen Stellen sowie die Auswirkung von Responsible Disclosure Meldeprozessen von Sicherheitslücken diskutiert hat.

Darüber hinaus wurde im Rahmen des Prototype Fund im Projekt SiC – Abgesicherte Verwaltungsvorgänge durch signierte Container die Absicherung von Softwarelieferketten im Open-Source-Umfeld erprobt.

Der InÖG arbeitet an der Schnittstelle zwischen Akademia, Politik, Verwaltung und Open Source Community. Das interdisziplinäre Team besteht aus Software Entwickler*innen, Forscher*innen, Unternehmer*innen, Hacker*innen, Berater*innen, Software Architekt*innen und Mitarbeiter*innen des öffentlichen Dienstes und Gesundheitswesens.

Der InÖG ist gemeinnützig, agiert überparteilich, gemeinwohlorientiert sowie unabhängig von Unternehmen und Verbänden.