Em002-1 Praxis-Leitfaden Open Source Software in der Bundesverwaltung

Empfehlung zur Bundesinformatik[1]

Dieses Dokument ist eine Beilage zum Hauptdokument Em002.

Klassifizierung:[2]	Nicht klassifiziert
Verbindlichkeit:[3]	Empfehlung
Planungsfeld:[4]	IKT der Bundesverwaltung
Diese Version:	2.0
Ersetzt Version:	1.1 vom 25.02.2025
Status:	Genehmigt
Freigabedatum (diese Version):	9.12.2025
Freigegeben durch, Rechtsgrundlage:	Freigegeben durch den Delegierten für digitale Transformation und IKT-Lenkung (D-DTI), gestützt auf Artikel 17 Absatz 1 der Verordnung vom 25. November 2020 über die Koordination der digitalen Transformation und die IKT-Lenkung in der Bundesverwaltung (VDTI), SR 172.010.58
Sprachen:	Deutsch (Original), Französisch, Italienisch, Englisch (Übersetzung)

[1] «Empfehlung zur Bundesinformatik» gemäss [P035], *Abschnitt 4.6*

[2] Zu der Klassifizierung INTERN und VERTRAULICH vgl. *Verordnung vom 8. November 2023 über die Informationssicherheit in der Bundesverwaltung und der Armee (ISV, SR 128.1)*

[3] Vgl. Fussnote 1

[4] Planungsfelder gemäss *IKT-Strategie des Bundes 2020-2023 vom 3. April 2020 (SB000)*

Inhaltsverzeichnis

1. Verwendung dieses Leitfadens 3

2. Definitionen 5

3. Potenzial und Herausforderungen 8

3.1 Potenzial von Open Source Software 8
3.2 Herausforderungen im Umgang mit OSS 10

4. Konstellationen Nutzung/Bearbeitung von OSS 13

4.1 Reifegrade für OSS in der Verwaltung 13
4.2 Einsatz unveränderter Open Source Software 15
4.3 Entwickeln mit Open Source Komponenten und Freigabe von Quellcode 15
4.4 Mitarbeit in offenen Projekten (Kontribution) 15

5. Eigenschaften und Auswahl von Open Source Lizenzen 15

6. Beschaffung von OSS 16

7. Auffindbarkeit von OSS-Lösungen 17

7.1 OSS-Katalog 17
7.2 OSS Directory 17
7.3 alternativeTo 17
7.4 Repositories von Open Source 18
7.5 Open Hub 19
7.6 Spezielle Code-Repositoires 19

8. Support-Modelle beim Einsatz von OSS 20

8.1 Einsatz ohne professionellen Support 20
8.2 Einsatz mit internem Support 20
8.3 Einsatz mit Support durch externen Anbieter 21

9. Ansprechstelle 21

Anhänge 22

A. Änderungen gegenüber Vorversion 22
B. Referenzen 22
C. Abkürzungen 22
D. Geschäftsmodelle mit OSS 23

D.1 Services und Produkte basierend auf OSS 23

D.2 Services für OSS 23

D.3 Subscriptions 23

D.4 Dual Licensing 24

1. Verwendung dieses Leitfadens

Der Bund ist gemäss Artikel 9 des «Bundesgesetz über den Einsatz elektronischer Mittel zur Erfüllung von Behördenaufgaben» (EMBAG) verpflichtet, den Quellcode von Software offenzulegen, den er zur Erfüllung der Aufgaben entwickelt oder entwickeln lässt.[5] Aus diesem gesetzlichen Auftrag hat der Bereich DTI der Bundeskanzlei übergeordnete Ziele im Strategischen Leitfaden Open Source Software abgeleitet.

Der vorliegende Praxis-Leitfaden konkretisiert die rechtlichen und strategischen Vorgaben. Er dient den Verwaltungseinheiten des Bundes bei der Benutzung, Beschaffung und der Freigabe von Open Source Software. Er vermittelt allen Interessierten einen kompletten Überblick über das Thema und verweist auf weiterführende Informationen in den anderen Hilfsmitteln.

Je nach Vorwissen und Bedürfnis der Lesenden macht es jedoch Sinn, den Leitfaden nur selektiv zu lesen und danach in die weiteren Hilfsmittel einzusteigen. Die folgende Übersicht hilft Ihnen, den richtigen Einstieg zu finden. Falls Sie unsicher sind, lesen Sie zuerst den Leitfaden durch.

Intention / Stakeholder	**Relevante Unterlagen**
Generelle Einführung zu Open Source Software	• Abschnitte 2, 3 und 7 • *«Em002-5 Merkblatt OSS-Hilfsmittel»* • *«Em002-3 Leitfaden OSS-Lizenzen»* • *«Em002-6 FAQ OSS»*
Generelle Einführung EMBAG für Entscheider	• *«Em002-5 Merkblatt OSS-Hilfsmittel»* • *«Em002 Strategischer Leitfaden Open Source Software in der Bundesverwaltung»*
Beschaffung nur zu nutzender Software	• Kapitel 4.1, 4.2 • *«Em002-7 Strategische Aspekte zu Beschaffung und OSS»* • *«Merkblatt Beschaffung von Software und EMBAG Art. 9» [KBB-MB]* • *«Wegleitung OSS in der Beschaffung» [BBL-WL]*
Neue Informatikentwicklung und Freigabe	• Kapitel 4.3, 4.4 • *«Em002-2 Anleitung zur Veröffentlichung von OSS» inkl. Checklisten* • *«Em002-4 Leitfaden OSS Community» inkl. Checkliste*
Beschaffung von erstellter Software (Dienstleistung oder Werk)	• Kapitel 4.1, 4.3, 4.4 • *«Em002-7 Strategische Aspekte zu Beschaffung und OSS»* • *««Werkzeugkasten BBL» u.a. mit Wegleitung Open Source in der Beschaffung* • *Lern- und Vorlagenplattform öffentliche Verwaltung der KBB (www.perimap.admin.ch) u.a. «Merkblatt Beschaffung von Software und Art.9 EMBAG» [KBB-MB]* • *«Wegleitung OSS in der Beschaffung» [BBL-WL]* • *«Em002-2 Anleitung zur Veröffentlichung von Open Source Software» inkl. Checklisten* • *«Em002-4 Leitfaden Community» inkl. Checkliste*

[5] SR 172.019

Rechtliche Beleuchtung Freigabe	• *«Em002-6 Häufig gestellte Fragen zur Veröffentlichung von OSS (OSS-FAQ)»* • *«Em002-3 Leitfaden OSS-Lizenzen»*
Arbeitsgrundlagen für Projektleitende	• *«HERMES Unterlagen» (www.hermes.admin.ch)* • *«Em002-5 Merkblatt OSS-Hilfsmittel anwenden»*
Kontributionen von Software an Dritte und Kollaborationen	• *«Em002-4 Leitfaden OSS Community» inkl. Checkliste* • *«Em002-7 Strategische Aspekte zu Beschaffung und OSS»* • *«Em002-3 Leitfaden OSS-Lizenzen»*

Die folgende Grafik gibt einen Überblick über die für OSS relevanten Dokumente in der Bundesverwaltung.

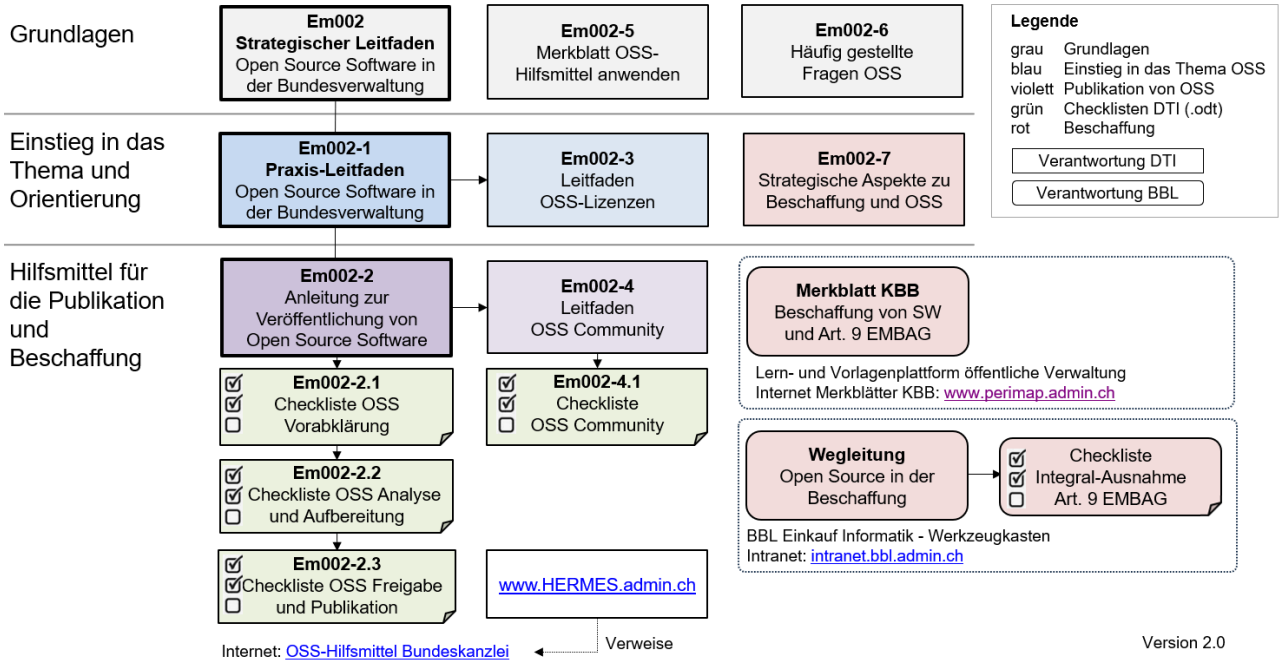

Abbildung 1:Überblick OSS Hilfsmittel in Bezug auf Art. 9 EMBAG

2. Definitionen

Open Source Software	Eine Software wird als Open Source Software (OSS) bezeichnet, wenn sie unter einer der rund 80 Lizenzen veröffentlicht ist, welche durch die Open-Source-Initiative OSI anerkannt sind [OSI2019]. Im Detail gibt die «Open-Source-Definition» zehn Kriterien vor, welche alle Open-Source-Lizenzen zu erfüllen haben *[PE1999]*. Im Wesentlichen lassen sich diese in vier Punkte zusammenfassen: 1. Die Software darf beliebig eingesetzt werden. 2. Der Quellcode der Software ist frei zugänglich. 3. Die Software darf beliebig kopiert und verbreitet werden. 4. Die Software darf verändert und in veränderter Form unter bestimmten Bedingungen weitergegeben werden.
Freie Software	Noch vor dem Begriff «Open Source» wurde von der Free Software Foundation (FSF) in den 80er-Jahren der Begriff der «Freien Software» eingeführt. Inhaltlich entspricht Freie Software den Bedingungen von Open Source Software, zielt jedoch darauf ab, dass die Software stets frei zugänglich ist und möglichst nicht in proprietäre Software integriert wird.[6]
Proprietäre Software	Bei einer proprietären Software entwickelt ein Anbieter eine Software und verkauft eine Nutzungslizenz an den Anwender. Der Anwender kennt in der Regel den Software-Code nicht und darf die Software nicht weitergeben oder verändern. Er kann die Software nur gemäss den Lizenzvorgaben (z.B. End-User License Agreement, EULA) gegen Bezahlung von Lizenzgebühren nutzen: z.B. darf die Software nur von einer bestimmten Anzahl Nutzenden verwendet werden oder nur für eine bestimmte Anzahl von Prozessoren.[7] Für die meist jährlich anfallenden Wartungsgebühren verpflichtet sich der Anbieter innert nützlicher Frist Fehler zu beheben und die Software kontinuierlich weiterzuentwickeln.
Lizenz	Eine Lizenz ist ein Dokument, das verbindliche Richtlinien für die Nutzung und Verbreitung von Software enthält. Im Falle von Open Source werden meist Lizenzen verwendet, die OSI-konform sind.[8]
Community	Eine Community im Zusammenhang mit einer Open Source Software ist weit definiert. Es kann sich um ein lockeres Ökosystem handeln oder um eine Struktur mit Governance, die die Software weiterentwickelt. Das Dokument *«Em002-4 Leitfaden OSS Community»* beschäftigt sich mit diesem Thema. Mitglieder der Community können gemeinsam das Produktmanagement übernehmen, entwickeln, testen, Übersetzungen machen, Feedback geben oder einfach die Software brauchen.
Rechte Dritter	Auch bei Open Source Software können Schutzrechte bestehen (Urheberrecht, Markenrecht, Patentrecht) und von Drittparteien geltend gemacht werden. – beispielsweise bei der Verwendung von Source Code, der von Dritten erstellt wurde.
Ausnahmen gemäss EMBAG	Als Ausnahmen gemäss Art. 9 Absatz 1 EMBAG gelten Rechte Dritter und sicherheitsrelevante Gründe. Beide werden im Abschnitt 3 der *«Em002-2 Anleitung zur Veröffentlichung von Open Source Software»* behandelt.

[6] Siehe https://de.wikipedia.org/wiki/Freie_Software

[7] Siehe https://de.wikipedia.org/wiki/Propriet%C3%A4re_Software

[8] Siehe https://opensource.org/

Source Code / Quellcode	Source Code, auch Quellcode genannt, ist in der Informatik der für Menschen lesbare, in einer Programmiersprache geschriebene Text eines Computerprogrammes.[9]
Publikation	Mit Publikation ist im Kern die Veröffentlichung des Quelltexts unter einer offenen Lizenz gemäss OSI gemeint. Aus Praktikabilitätsgründen sind meist auch Dokumentationen und automatische Anleitungen zum Bau der Applikation aus dem Quellcode mitgemeint. Auch automatische Tests und Testunterlagen werden mitgemeint. Damit eine Community entstehen kann, wird eine Plattform verwendet, die dann etliche Funktionen einer Entwicklungsplattform mitübernimmt (Build, Test, Support).
Digitale Souveränität	Wir verwenden die Definition aus dem Bericht Digitale Souveränität des EDA *[St2024]*: „«Digitale Souveränität eines Staates oder einer Organisation umfasst zwingend die vollständige Kontrolle über gespeicherte und verarbeitete Daten sowie die unabhängige Entscheidung darüber, wer darauf zugreifen darf. Sie umfasst weiterhin die Fähigkeit, technologische Komponenten und Systeme eigenständig zu entwickeln, zu verändern, zu kontrollieren und durch andere Komponenten zu ergänzen.“
Repository	Ein Repository ist ein Aufbewahrungsort, der von einem Versionenkontrollwerkzeug für Dateien und Metadaten über die Codebasis verwendet wird. Repositories erlauben es mehreren Kontributoren, an denselben Dateien zu arbeiten und verschiedene Versionen zu speichern. Die meisten erlauben auch Issue-Tracking, Releasemanagement, automatische Builds und Dokumentation.
Fork	Ein Fork[10] eines Open-Source-Projekts ist der Vorgang, in dem eine unabhängige Weiterentwicklung zum ursprünglichen Projekt aufgesetzt wird.
Contributor Licence Agreement (CLA)	Ein Contributor License Agreement[11] ist ein Dokument, in dem die Bedingungen beschrieben sind, unter denen geistiges Eigentum zu einem Projekt oder Vorhaben beigetragen werden kann.
Source Code Management (SCM)	Source Code Management[12], auch Version Control oder Quellcodemanagement, ist ein Werkzeug, welches insbesondere bei der Softwareentwicklung Änderungen und Versionsnummern effektiv verwaltet. Die freie Software «Git»[13] zur Verwaltung von verteilten SCM hat sich etabliert.
Open Source Software-Development (OSSD)	Bei Open Source Software-Development (OSSD)[14] wird ergänzend zur Veröffentlichung des Quellcodes der ganze Entwicklungsprozess öffentlich durchgeführt. Dabei ist von den Anforderungen (Issues) bis hin zum Quellcode alles transparent. OSSD wird durch öffentliche Kommunikationstools (Mailinglist, Foren, etc.), ein Version Control System (git), Bug und Featureliste, Roadmap und Entwicklertools unterstützt.

[9] Siehe: https://de.wikipedia.org/wiki/Quelltext

[10] Siehe: https://en.wikipedia.org/wiki/Fork_(software_development)

[11] Siehe: https://de.wikipedia.org/wiki/Contributor_License_Agreement

[12] Siehe: https://en.wikipedia.org/wiki/Version_control

[13] Siehe: https://de.wikipedia.org/wiki/Git

[14] Siehe: https://en.wikipedia.org/wiki/Open-source_software_development

Software Package Data Exchange (SPDX)	Software Package Data Exchange (SPDX)[15] beschreibt ein Standardformat für Software Bill of Materials (SBOM); Dies mit dem Ziel, den korrekten Umgang mit Freier Software bzw. Open-Source-Software zu erleichtern.
Copyleft-Effekt	Steht eine Software unter einer Lizenz mit einer Copyleft-Bestimmung, muss jegliche Veränderung oder Erweiterung des Quellcodes wieder unter der Lizenz der bearbeiteten Open Source Software freigegeben werden (siehe auch *«Em002-3 Leitfaden OSS-Lizenzen»*).

[15] Siehe: https://en.wikipedia.org/wiki/Software_Package_Data_Exchange

3. Potenzial und Herausforderungen

Das Potenzial von Open Source Software in der heutigen Informatik ist hoch. Gleichzeitig bringt die Nutzung von Open Source-Lösungen aber auch Herausforderungen mit sich. Diese zwei Seiten werden im nachfolgenden Abschnitt näher erläutert. Die Angaben basieren u.a. auf den Resultaten der Open-Source-Studie Schweiz 2018, 2021 und 2024 der Universität Bern, bei der die Befragten Auskunft über die erkannten Vor- und Nachteile von Open Source Software erteilten.

3.1 Potenzial von Open Source Software

Die folgenden Punkte beschreiben das Potenzial, das bei der Verwendung und der Freigabe von Open Source Software realisiert werden kann.

1. Digitale Souveränität	Es geht dabei um die Fähigkeit, digitale Leistungen nutzen und kontrollieren zu können. Die Selbstbestimmung geht über den gesamten Lebenszyklus eines digitalen Systems. Open Source Software fördert die digitale Souveränität, da die Wechselfähigkeit, Gestaltungsfähigkeit und Einflussnahme gewährleistet ist[16].
2. Keine Lizenzgebühren	Für die Nutzung von Open Source Software fallen keine Lizenzkosten an. Beim Bezug von komplexen Standard Open-Source-Software-Paketen kann jedoch der Bezug von so genannten «Subscriptions» (Support-Abo) Sinn machen, für die eine Gebühr bezahlt wird.
3. Kostenersparnisse durch Kooperationen mit anderen Nutzern	Da Software unter einer Open-Source-Lizenz unbeschränkt genutzt und weiterentwickelt werden kann, können nach dem Prinzip «Einmal entwickeln – mehrfach anwenden» die Kosten für Weiterentwicklungen geteilt bzw. bestehende Zusatzentwicklungen von anderen Verwaltungsstellen übernommen werden. Gleichzeitig besteht die Möglichkeit von Erfahrungen und Entwicklungen anderer zu profitieren.
4. Community Building & Wissensaustausch	Open Source Software erleichtert das Community Building und den Wissensaustausch beispielsweise zwischen den verschiedenen föderalen Ebenen der Schweiz. Es kann gemeinsam an einer Software weiterentwickelt, Fehler behoben und individuelle Erfahrung geteilt werden. Der gesteigerte Wissensaustausch zwischen verschiedenen Verwaltungsstellen führt dazu, dass ein besseres Verständnis davon entsteht, woran andere arbeiten, so dass Doppelspurigkeit vermieden werden und sich die besten Lösungen verbreiten.

[16] Wir verwenden die Definition von *[St2024]*: „«Digitale Souveränität eines Staates oder einer Organisation umfasst zwingend die vollständige Kontrolle über gespeicherte und verarbeitete Daten sowie die unabhängige Entscheidung darüber, wer darauf zugreifen darf. Sie umfasst weiterhin die Fähigkeit, technologische Komponenten und Systeme eigenständig zu entwickeln, zu verändern, zu kontrollieren und durch andere Komponenten zu ergänzen."

5. Niedrigere Hersteller-abhängigkeit	Die Abhängigkeit von Software-Herstellern (Vendor Lock-in) wird in der Informatik als sehr hoch eingeschätzt. Die Verwendung von Software unter einer Open-Source-Lizenz reduziert die Abhängigkeit von den Herstellern, da Betrieb, Wartung, Support, Weiterentwicklung und andere Dienstleistungen für Open Source Software offen ausgeschrieben werden können. Anstelle eines gesamten Systems aufgrund von einem Vendor Lock-in komplett neu zu beschaffen, können auch Erweiterungen oder Lifecycle an einer OSS-Lösung separat ausgeschrieben werden. Dies kann grössere Migrationsaufwände einsparen.
6. Offene Standards und hohe Interoperabilität	Bei Open Source Software ist die Kompatibilität mit anderen Softwarelösungen und Informatiksystemen (Interoperabilität) in der Regel höher als bei proprietärer Software. Auch verwenden Open-Source-Lösungen praktisch ausschliesslich offene Datenformate, weshalb diese einfach mit anderen Systemen ausgetauscht werden können.
7. Transparenz über den Aufbau der Software	Da die Software auch in Form des Source Codes vorliegt, kann ihre Qualität z.B. durch externe Reviews geprüft werden. Zudem lassen sich anhand des Quellcodes Dokumentationen (z.B. im Hinblick auf Neuausschreibungen für Weiterentwicklungsleistungen oder die Ablösung einer Open Source Software am Ende der Lebensdauer) erstellen.
8. Sicherheit und Vertrauen durch Transparenz	Durch die Offenheit des Quellcodes kann die Sicherheit von Open Source Software höher sein als bei proprietärer Software. Ausserdem ist es bei Open Source Software sehr viel schwieriger, Hintertüren und andere Lücken einzubauen, was zu mehr Vertrauen in die Software führt.
9. Höhere Qualität und Modularität des Quellcodes	Die Qualität von Open Source Software kann höher sein als bei proprietärer Software, da die Motivation guten Programmier-Code zu schreiben höher sein kann, wenn die Entwickler wissen, dass ihr Quellcode veröffentlicht werden wird. Zudem sind Open-Source-Lösungen typischerweise hoch modular aufgebaut, so dass einzelne Module einfach ersetzt und zugleich die übrigen weiterverwendet werden können.[17]
10. Einfache Anpassungen an eigene Bedürfnisse	Der Zugriff auf den Quellcode ermöglicht es Nutzern, selbst oder durch externe Anbieter Weiterentwicklungen vorzunehmen. So kann die Software rasch den eigenen Bedürfnissen angepasst werden.
11. Rasche Innovation und Integration	Es findet eine schnelle, kontinuierliche Weiterentwicklung von Open Source Software durch die Community statt. Neue Technologien und Datenstandards werden bspw. oft als Open-Source-Programmier-Bibliotheken veröffentlicht. Dies ermöglicht die rasche Realisierung von innovativen Software-Lösungen.

[17] Ein zusätzlicher Aspekt der Sicherheit ist die - im Vergleich zu proprietärer Software - einfachere Möglichkeit die verwendeten Libraries (inkl. Dependencies) zu dokumentieren. Das führt im Umkehrschluss dazu, dass bei aufgetretenen Schwachstellen schnell identifiziert und behoben werden können (siehe auch SBOM).

12. Arbeitgeber-attraktivität	Der Einsatz von modernen Open-Source-Technologien und die informelle Zusammenarbeit mit internationalen Communities fördert die Motivation der Mitarbeitenden und erhöht dadurch die Arbeitgeberattraktivität, was wiederum die Rekrutierung von jungen, qualifizierten Fachkräften erleichtert.

3.2 Herausforderungen im Umgang mit OSS

Im Folgenden werden die typischen, in der Praxis angetroffenen Herausforderungen mit Open Source Software beschrieben und mögliche Lösungsansätze aufgezeigt. Auf allgemeine Herausforderungen, die jegliche Software betreffen gehen wir an dieser Stelle nicht ein (z.B. Notwendigkeit der Prüfung der Cybersicherheit und die Notwendigkeit, für kritische Software den entsprechenden Support zu sichern).

1. Hohe Wechselkosten aufgrund bestehender Abhängigkeiten	Eine Ablösung von proprietärer Software durch Open-Source-Lösungen kann sehr aufwändig sein, wenn die Applikation über Schnittstellen-Integration oder anderen Abhängigkeiten eng in das bestehende Informatik-System eingebettet ist. Diese Wechselkosten führen dazu, dass sich Einführungen von Open Source Software oftmals nur beim Ablauf des Lebenszyklus der proprietären Software lohnen.
2. Fehlende Features oder keine passende Open Source-Lösung	Soll eine Standardsoftware beschafft werden, kann es vorkommen, dass zur (bisherigen) proprietären Lösung keine oder nur eine funktional unzureichende Open-Source-Software-Alternative zur Verfügung steht. Als Lösungsansatz besteht die Möglichkeit, zusammen mit anderen Nutzern der Open Source Software die fehlenden Features gemeinsam zu entwickeln und diese dem Open-Source-Projekt hinzuzufügen.[18]
3. Höhere Aufwände bei der Integration	Kurzfristige Kostenersparnisse, die durch Lizenzeinsparungen beim Einsatz von Open Source Software anfallen, werden oftmals durch höhere Aufwände kompensiert, die bei der Integration von Open Source Software entstehen. Somit ist es wichtig zu erkennen, dass der professionelle Einsatz von Open Source Software nicht «gratis» ist, sondern internen oder externen Aufwand generiert.

[18] Für solche neuen Features sind die Projekte oder die Anwendungsverantwortlichen zuständig. Das Projekt regelt, wie dies genau gemacht wird. Em002-2 und Em002-4 helfen bei Detailentscheiden. Je nach Grösse des Features macht es keinen Sinn, die Anleitungen vollständig zu durchlaufen. Beschaffungsrechtliche Aspekte bei den Ressourcen sind einzuhalten.

4. Verantwortlichkeiten und Support unklar	Kritisiert wird an Open Source Software oftmals das Fehlen von Support und Wartung durch externe Unternehmen. Zwischenzeitlich gibt es allerdings eine Vielzahl an Anbieter, die kommerzielle Dienstleistungen (langfristigen Support, fortlaufende Wartung, Weiterentwicklung, Haftung und Gewährleistung etc.) für Open-Source-Lösungen anbieten. Die unterschiedlichen Geschäftsmodelle mit Open Source Software sind in Anhang D erläutert, die verschiedenen Support-Modelle in Abschnitt 8 dargestellt. Eine aktuelle Übersicht über Open Source-Dienstleister und deren Leistungen sind im Open Source Directory (*https://www.ossdirectory.com*) oder in der OSS-Studie 2024[19] *[OSS2024]* zu finden.
5. Teilweise kleiner Markt mit wenig Anbietern	Da gewisse Open-Source-Applikationen wie bspw. Desktop-Anwendungen grundsätzlich wenig oder gar keinen Support- und Wartungsaufwand verursachen, besteht in gewissen Fällen kaum ein Markt von Open-Source-Anbietern für Support- und Wartungsdienstleistungen. In diesem Fall kann eine Ausschreibung von Dienstleistungen für Open Source Software dazu beitragen, dass sich ein Markt bildet, der den Wettbewerb unter den Anbietern stärkt. Im Hinblick auf die Ausschreibung solcher Leistungen sind das Bestehen einer breit abgestützten Entwicklungs-Community für die betreffende Software und die Qualität der Entwicklungsdokumentation von praktischer Bedeutung.
6. Geringe Visibilität	Communities von Open-Source--Projekten fokussieren sich vorwiegend auf die Produktentwicklung und kaum auf deren Vermarktung. Demgegenüber investieren Hersteller von proprietärer Software intensiv in Marketing und den Verkauf ihrer Produkte. Durch diese fehlende Werbung für Open Source Software entsteht oftmals das falsche Bild, dass keine Alternative zu den proprietären Produkten existiert. Es gibt jedoch Plattformen wie alternativeTo[20], welche einen Überblick über Lösungen für bestimmte Aufgabestellungen bieten (mehr zu OSS-Alternativen in Kapitel 0).
7. Mangelnde Akzeptanz der Endbenutzenden	Aufgrund andersartiger Benutzeroberflächen, fehlender Funktionen, niedriger Benutzerfreundlichkeit und wenig Werbung für Open Source Software besteht bei gewissen Open-Source-Produkten eine mangelhafte Akzeptanz der Endbenutzenden. Entsprechende Kommunikation, Dokumentation und Schulungsangebote können dieser Herausforderung entgegenwirken. Dass sich Open Source Software entsprechend den Bedürfnissen der Benutzenden weiterentwickeln lässt, kann ebenfalls zu einer höheren Akzeptanz führen.

[19] Siehe https://www.oss-studie.ch/

[20] https://alternativeto.net/

8. Wenig oder kein internes Know-how	Open-Source-Lösungen entwickeln sich rasch weiter und benötigen gleichzeitig ein vertieftes technisches Verständnis. So kann es vorkommen, dass bei internen Mitarbeitenden für gewisse Open Source Software kein oder unzureichendes Know-how besteht. Durch Weiterbildungen und Möglichkeiten für Selbststudium über Online-Quellen und den Aufbau eigener Wikis etc. kann internes Open Source-Know-how aufgebaut werden.
9. Schwierige Zukunftseinschätzung	Oftmals ist es für Aussenstehende einer Open Source Community schwierig zu erkennen, wie sich das Projekt in Zukunft weiterentwickeln wird. Deshalb ist es wichtig, eine realistische Einschätzung über die künftige Weiterentwicklung einer Open-Source-Lösung vornehmen zu können. In Abschnitt 7.4 wird diesbezüglich die Plattform Open Hub vorgestellt, die eine Einschätzung zur Aktivität und Heterogenität der Entwickler-Community erlaubt. Dies ermöglicht es, die zukünftige Entwicklung eines Open Source-Projektes besser einzuschätzen.
10. Rechtliche Unsicherheiten	Die Vielzahl unterschiedlicher Open-Source-Lizenzen und die bisher geringe Anzahl von Gerichtsentscheiden zu Auslegungsfragen führen mitunter zu rechtlichen Unsicherheiten mit Open Source Software. Einen Überblick über die wichtigsten Open-Source-Lizenzen und deren Eigenschaften und Kompatibilitäten soll dieser Praxisleitfaden vermitteln. Quellen für vertiefende Antworten zu juristischen Fragenstellungen sind in den Dokumenten *«Em002-3 Leitfaden OSS-Lizenzen»* und in den *«Em002-6 FAQ OSS und Art. 9 EMBAG»* zu finden.
11. Zu umfangreiches Angebot an Open Source Software	In den letzten Jahren hat die Anzahl von Open Source Software massiv zugenommen. Potenzielle Nutzer von Open Source Software bemängeln deshalb die Vielzahl vorhandener Open-Source-Lösungen, die es auf dem Markt gibt. Aus diesem Grund werden in Kapitel « Auffindbarkeit von …» mit alternativeTo und Open Hub zwei Plattformen vorgestellt, die einen Vergleich von Open-Source-Lösungen ermöglichen.
12. Abstimmungsaufwand bei kollaborativer Entwicklung.	Wenn die Bundesverwaltung sich an Kollaborationen beteiligt und in diesen entwickelt, so ergibt sich daraus ein Abstimmungsaufwand in Kollaborationen, aber auch wenn eine Community mitgenommen werden muss. Dieser Zusatzaufwand sollte sich idealerweise durch eine verbesserte Lösung, bessere Akzeptanz oder Beteiligung der Dritten trotzdem rechnen.

4. Konstellationen Nutzung/Bearbeitung von OSS

4.1 Reifegrade für OSS in der Verwaltung

Bei der Nutzung und Entwicklung von OSS durch die Bundesverwaltung lassen sich im Hinblick auf die Pflichten in Zusammenhang mit Open Source-Lizenzen verschiedene Konstellationen unterscheiden. Ein typisches Reifegradmodell[21] für die Einführung von Open Source in Organisationen kann wie folgt aussehen:

1. Denial - Keine oder unbewusste Nutzung von Open Source Software
2. Consumption- Passive Nutzung von Open Source Software (Konsumation)
3. Participation- Engagement in Open-Source-Communities (Kontribution)
4. Contribution- Pragmatische Beiträge zu Open-Source-Projekten (Kontribution)
5. Leadership- Strategische Beteiligung an Open Source zur Steigerung des Geschäftswerts (Kollaboration)[22]

Konstellation	***Auswirkung***
Die **blosse Verwendung** von bestehender Open Source Software (ohne Modifikationen des Programmcodes).	Es besteht **keine Verpflichtung** zur Weitergabe des Sourcecodes.
Die **vollständige Neuentwicklung** von Software, welche unter einer Open-Source-Lizenz publiziert werden soll.	**In diesem Fall besteht vollständige Freiheit, unter welcher Lizenz die Software publiziert wird.** Für den Bund sollte der *«Em002-3 Leitfaden OSS-Lizenzen»* verwendet werden. Die *«Em002-2 Anleitung zur Veröffentlichung von OSS»* und der *«Em002-4 Leitfaden Community»* sind zu beachten.
Das **Weiterentwickeln** (intern oder extern) mit bestehenden Open-Source-Komponenten (mit oder ohne Copyleft-Effekt[23]), solange die Software **ausschliesslich innerhalb der gleichen Organisation genutzt** und nicht weitergegeben wird.	Die OSS-Komponenten können beliebig mit Drittkomponenten kombiniert werden. Es besteht grundsätzlich von der Lizenz her keine Publikationspflicht, solange die Software ausschliesslich innerhalb der gleichen Organisation genutzt und nicht weitergegeben wird (Ausnahme AGPL). Da rechtlich nicht abschliessend geklärt ist, wie weit der Begriff einer ‚Weitergabe innerhalb der gleichen Organisation' reicht, kommt dieses Szenario im Fall von OSS-Komponenten mit Copyleft-Effekten nur in Ausnahmefällen zum Tragen. Ziehen Sie in diesen Fällen den Rechtsdienstes ihrer Organisation bei. Siehe *«Em002-3 Leitfaden OSS-Lizenzen»* und *«Em002-6 FAQ OSS und EMBAG Art.9»*. Die *«Em002-2 Anleitung zur Veröffentlichung von OSS»* und

[21] Reifegradmodell in Anlehnung an https://todogroup.org/resources/guides/a-guide-to-outbound-open-source-software/#maturity-levels

[22] Kreation ist nicht erwähnt, da Art. 9 EMBAG das direkt verlangt.

[23] Erläuterung zum Begriff Copyleft in *«Em002-3 Leitfaden OSS-Lizenzen»*

	der *«Em002-4 Leitfaden Community»* sind zu beachten. **Da für die die Publikationspflicht gemäss EMBAG Art. 9 besteht, muss sie die Anpassungen wieder freigeben. Am einfachsten als Changes zum bestehenden Open-Source-Projekt.**
Das **Weiterentwickeln** (intern oder extern) **ausschliesslich mit** bestehenden Open Source **Komponenten ohne Copyleft-Effekt**, wenn die Software auch extern (z.B. an Kantone) weitergegeben werden soll.	Die OSS-Komponenten können beliebig mit Drittkomponenten kombiniert werden und es besteht keine Publikationspflicht aus den Lizenzen (siehe auch *«Em002-3 Leitfaden OSS-Lizenzen»*). **Im Rahmen von Art. 9 EMBAG ist die Freigabe hingegen zwingend. Idealerweise erfolgt die Freigabe über das bestehende Projekt (kein Fork).** Die *«Em002-2 Anleitung zur Veröffentlichung von OSS»* und der *«Em002-4 Leitfaden Community»* sind zu beachten.
Das **Weiterentwickeln** (intern oder extern) mit bestehenden Open Source Komponenten **mit Copyleft-Effekt**, wenn die Software auch extern (z.B. an Kantone) weitergegeben werden soll.	**Es besteht eine Pflicht zur Publikation** schon der Copyleft-Lizenz (siehe zum Copyleft-Effekt auch *«Em002-3 Leitfaden OSS-Lizenzen»*). Die *«Em002-2 Anleitung zur Veröffentlichung von OSS»* und der *«Em002-4 Leitfaden Community»* sind zu beachten.
Das **Beitragen** an ein bestehendes Open Source Projekt.	**Es besteht grundsätzlich eine Publikationspflicht gemäss EMBAG.** **Es werden Governance und Lizenz des Projekts verwendet.** Die *«Em002-2 Anleitung zur Veröffentlichung von OSS»* und der *«Em002-4 Leitfaden Community»* sind zu beachten.
Das **gemeinsame Entwickeln** eines neuen oder bestehenden Projekts mit einer Community und Kostenbeteiligung.	**Es besteht grundsätzlich eine Publikationspflicht gemäss EMBAG.** Anhand der *«Em002-2 Anleitung zur Veröffentlichung von OSS»*, des *«Em002-3 Leitfaden OSS-Lizenzen»* und des *«Em002-4 Leitfaden Community»* **wird das Projekt und seine Governance aufgebaut.**
Alle Varianten der Entwicklung/Weiterentwicklung erfolgen **über einen Lieferanten oder beauftragten Dritten**.	**Es besteht grundsätzlich eine Publikationspflicht gemäss EMBAG**. Anhand der *«Em002-2 Anleitung zur Veröffentlichung von OSS»,* des *«Em002-3 Leitfaden OSS-Lizenzen»* und des *«Em002-4 Leitfaden Community».* Wenn möglich, sind bereits im Beschaffungsprozess die Voraussetzungen für eine Publikation zu schaffen und allenfalls auch die Anforderungen an die Freigabe der Software dem Lieferanten/Dritten mitzuteilen.

4.2 Einsatz unveränderter Open Source Software

Beim Einsatz von Open Source Software ist grundsätzlich nicht von Relevanz, unter welcher Open-Source-Lizenz diese veröffentlicht ist, denn sämtliche Open-Source-Lizenzen (siehe Abschnitt 2 «Definitionen») erlauben die uneingeschränkte Nutzung von Open Source Software, egal auf wie vielen Arbeitsplätzen, gleichzeitig eingeloggten Benutzern, Anzahl Servern und Prozessoren etc. die Software eingesetzt wird.

Beim Einsatz von unveränderter Open Source Software entstehen selbst bei Programmen, welche unter einer Lizenz mit strengem Copyleft-Effekt stehen, **keine Publikationsverpflichtungen** (Ausnahme AGPL-Lizenz, Affero General Public License).

Solange Open Source Software durch die bundesinternen Leistungserbringer eingeführt und betrieben wird, kann dies **ohne öffentliches Beschaffungsverfahren** geschehen, denn «Die OSS-Lizenz allein kostet die beschaffende Stelle in der Regel nichts und ist für sich allein daher auch nicht beschaffungsrelevant.»

Soll ein externer Dienstleister für Wartung, Support und andere Dienstleistungen für die Open Source Software beauftragt werden, sind die Vorgaben des öffentlichen Beschaffungsrechts zu beachten. Unterschiedliche Support-Varianten sind in Abschnitt 9 beschrieben. Wichtig ist ausserdem der Einsatz passender Eignungs- und Zuschlagskriterien bei der Beschaffung von Open Source Software. Die Grundlagen dafür sind in Abschnitt 8 erläutert.

4.3 Entwickeln mit Open Source Komponenten und Freigabe von Quellcode

Mit der Publikationspflicht gemäss Art. 9 EMBAG sollte hier gemäss der *«Em002-2 Anleitung zur Veröffentlichung von Open Source Software»* vorgegangen werden.
Der *«Em002-3 Leitfaden OSS-Lizenzen»* und der *«Em002-4 Leitfaden OSS-Community»* müssen bei Bedarf auch konsultiert werden. Das Ziel ist es, das Projekt anhand der drei Checklisten *[Em002-2.1], [Em002-2.2], [Em002-2.3]* kontrolliert freizugeben.

4.4 Mitarbeit in offenen Projekten (Kontribution)

Mögliche Überlegungen zur Kontribution der Bundesverwaltung zu offenen, bereits bestehenden Projekten sind im BITKOM-Leitfaden *[BITCOM2024]* Kapitel 4.2 aufgeführt[24]. Je nach der Bedeutung des Projekts für die Bundesverwaltung, ist zu prüfen, wie viel Verantwortung das entsprechende Amt übernehmen will und kann.

Die Mitarbeit in offenen Projekten und das direkte Entwickeln in offenen Projekten kann auch anhand der Dokumente für die Freigabe erfolgen. Der Fokus liegt dabei auf dem *«Em002-4 Leitfaden OSS-Community»* und seiner Checkliste *[Em002-4.1]*.

5. Eigenschaften und Auswahl von Open Source Lizenzen

Die Eigenschaften und Auswahl von Open Source Lizenzen sind im Dokument *«Em002-3 Leitfaden OSS-Lizenzen»* beschrieben.

[24] Siehe https://www.bitkom.org/Bitkom/Publikationen/Bitkom-Leitfaden-zu-Open-Source-Software-20.html in Kapitel 4.2.

6. Beschaffung von OSS

Strategische Themen in Zusammenhang von Beschaffung und OSS werden in *«Em002-7 Strategische Aspekte zu Beschaffung und Open Source Software»* behandelt.

Das Kompetenzzentrum Beschaffungswesen Bund KBB hat ein neues Merkblatt «*Beschaffung von Software und Art. 9 EMBAG*» *[KBB-MB]* sowie zwei Dokumente «Muster Kriterien – Beschaffung EMBAG und Open Source» und «Mustertexte Vertrag – Softwareentwicklung» publiziert.

Weitere Hilfestellung bietet das Intranet des BBL (nur im Bundesnetz zugänglich) mit der «*Wegleitung Open Source in der Beschaffung*» *[BBL-WL]* und der «*Checkliste Integral-Ausnahme Art. 9 EMBAG*» *[BBL-CL]*.

Für die Enterprise Readiness von Open Source kann das Dokument *«Selection criteria for enterprise-ready open source software» [Gu2024]*[25] oder auch der Bitkom-Leitfaden[26] konsultiert werden.

[25] Siehe *[Gu2024]*

[26] Siehe Praxisempfehlungen für Open-Source-Software | Leitfaden 2025 | Bitkom e. V., Kapitel 3.3

7. Auffindbarkeit von OSS-Lösungen

Verschiedene Plattformen ermöglichen interessierten Organisationen, Open Source Software vor dem Einsatz/der Beschaffung zu finden und zu analysieren. Die daraus resultierenden Daten stellen eine Grundlage für Entscheidungen zum Einsatz oder zur Beschaffung von Open Source Software dar.

Als Basis sollte auf jeden Fall eine minimale Zusammenstellung der Anforderungen und allenfalls eine Marktanalyse dienen.

7.1 OSS-Katalog

Die Bundesbehörden sind verpflichtet, den Quellcode der Software zu veröffentlichen, die sie zur Erfüllung ihrer Aufgaben entwickeln oder entwickeln lassen. Der OSS-Katalog[27] listet die seit dem 1. Januar 2024 von den Bundesbehörden entwickelte und als Open Source veröffentlichte Software auf. Dieser Katalog bietet eine Übersicht und erleichtert die Auffindbarkeit und Wiederverwendung von Open Source Software.

Falls ein neues Repository oder eine Organisation erstellt wird, soll dies via E-Mail opensource@bk.admin.ch für die Aufnahme in den OSS-Katalog gemeldet werden.

OSS-Katalog: https://www.opensource.admin.ch

7.2 OSS Directory

Das OSS Directory von CH Open[28] ermöglicht das einfache Auffinden von kompetenten Anbietern, die professionellen Support für Open Source Software erbringen.

Die Open Source Anbieter zeigen mittels Success Stories auf, bei welchen Nutzern (Kunden) sie bestimmte Open Source Produkte eingeführt haben. Ausserdem werden aktuelle Open Source News von bestehenden Plattformen, selber erstellte Top News, Events, Communities, Job-Angebote und Knowhow-Beiträge angezeigt. Das OSS Directory ist sowohl auf Deutsch als auch auf Englisch, Französisch und Italienisch verfügbar. Alle Einträge können durch registrierte Benutzende selber erfasst und aktualisiert werden.

OSS Directory: https://www.ossdirectory.com

7.3 alternativeTo

Ein praktisches Tool, um Alternativen zu einer Software ausfindig zu machen, ist "alternativeTo – Crowdsourced software recommendations".[29] Wie es der Name schon sagt, sind die Alternativen und die Bewertungen dieser Alternativen durch crowdsourcing ("crowd" steht für "Menschenmenge" und "sourcing" für "Beschaffung") durch den Input vieler einzelner Anwender entstanden. alternativeTo unterscheidet vier Arten von Software-Lösungen, wobei nur die erste als Open Source Software gemäss Art. 9 EMBAG gilt:

Free Open Source	Software, die unter einer Open Source Lizenz veröffentlicht ist.
Free	Kostenlose Software (Freeware), deren Quellcode jedoch nicht frei zugänglich ist und die nicht verändert werden darf.

[27] https://www.opensource.admin.ch/

[28] https://www.ossdirectory.com/de/home

[29] https://alternativeto.net/

Freemium	Von solcher Software wird eine kostenlose und eine Premium Version angeboten, wobei bereits in der kostenlosen Version alle wichtigen Funktionalitäten ausgeübt werden können. Nicht zu dieser Kategorie zählt eine Software, die für eine gewisse Zeit gratis getestet werden kann (Bsp. 30-tägige Probeversion), diese fällt in die Kategorie "Commercial".
Commercial	Dabei handelt es sich um proprietäre Software, bei deren Nutzung Lizenzkosten anfallen.

7.4 Repositories von Open Source

Die weltweit populärste Open-Source-Entwicklungsplattform mit über 30 Millionen Benutzenden und über 100 Millionen Quellcode-Verzeichnissen[30] ist zurzeit GitHub. Heute verfügen praktisch alle Informatik-Unternehmen, insbesondere auch Hersteller proprietärer Software, sowie jegliche andere Art von Firmen und Organisationen über einen GitHub-Auftritt. Dort veröffentlichen die Unternehmen Open-Source-Applikationen, Tools, Programmier-Bibliotheken etc. Auch immer mehr Behörden – rund ein Dutzend aus der Schweiz[31] – veröffentlichen auf «GitHub and Government» unter https://government.github.com ihre eigene Open Source Software. Dieser Prozess kann auch mit anderen Repositories durchgeführt werden.

Andere Repositories sind z.B.:

- Gitlab
- BitBucket
- Sourceforge
- launchpad

Auf GitHub «Insights» lassen sich zahlreiche relevante Statistiken eines Open Source Projekts auf GitHub ablesen:

Pulse	Übersicht der jüngsten Aktivitäten eines Open Source Projekts auf GitHub: Zusammenfassung der wichtigsten Angaben über Entwicklungs-Intensität, Heterogenität der Community, offenen Meldungen, Verbesserungen (Pull Requests) etc.
Contributors	Darstellung, welche Entwickler wann wie aktiv gewesen sind: Dies ist ein wichtiger Indikator, ob alles von einer Person abhängt oder ob eine grössere Community dahintersteht.
Commits	Anzeige, welche Beiträge zu diesem Open Source Projekt in welchem Zeitraum geleistet wurden.
Code Frequency	Visualisierung, wie viel Quellcode wann hinzugefügt oder entfernt wurde.
Dependency graph	Abhängigkeiten des Open Source Projekts von anderer Open Source Software (bspw. relevant bei der Identifizierung von Sicherheitslücken und Updates).
Network	Darstellung, wann welcher Entwickler einen Beitrag zu welchem Entwicklungs-Ast geleistet hat.

[30] https://octoverse.github.com

[31] https://government.github.com/community/#switzerland

Forks	Liste aller Kopien des Open Source Projekts auf GitHub: Indikator der Popularität und Verbreitung der Open Source Software.

7.5 Open Hub

Sollen Informationen zu einer Open-Source-Lösung gesammelt werden, die nicht unbedingt auf GitHub entwickelt wird, bietet sich Open Hub von der Unternehmung Black Duck[32] an. Von rund einer halben Million Open-Source-Projekten ist eine Vielzahl an wichtigen Informationen übersichtlich zusammengefasst:

Project Summary	Eine kurze Beschreibung des Open-Source-Projektes.
In a Nutshell	Die wichtigsten Fakten zu einem Open-Source-Projekt, wie die Anzahl Commits, Contributors und Codezeilen sowie die meistverwendete Programmiersprache, der Zeitpunkt des ersten Commits und der letzten Änderung. Zudem wird eine Einschätzung zur Codebasis und der Grösse des Entwicklerteams aufgeführt (Bspw. "Mozilla Firefox has a well-established, mature codebase maintained by a very large development team with stable Y-O-Y commits.")
Quick Reference	Beinhaltet Informationen zur Organisation, Links zum Projekt und zum Code sowie Verweise auf ähnliche Projekte.
Licenses	Zeigt an, unter welcher Lizenz bzw. welchen Lizenzen das Open-Source-Projekt steht und welche Konsequenzen damit verbunden sind.
Project Security	Gibt Auskunft zur Sicherheit und Vulnerabilität des Open-Source-Projektes.
Code	Die Anzahl Codezeilen im Zeitverlauf werden in einer Grafik visualisiert – aufgeteilt nach den Programmiersprachen.
Activity	Die Anzahl Commits pro Monat wird in einer Grafik dargestellt. Zudem wird eine Zusammenfassung der letzten 30 Tage und 12 Monate aufgeführt.
Community	Die Anzahl aktiver Contributors pro Monat wird in einer Grafik dargestellt. Zudem wird ein Rating des Projektes auf einer Fünf-Sterne-Skala angezeigt.

Darüber hinaus bietet Open Hub die Möglichkeit, verschiedene Open-Source-Projekte miteinander zu vergleichen.[33] Dadurch wird schnell ein Überblick geschaffen, welches Projekt über die aktivste Community, die längste Weiterentwicklungszeit oder die passende Lizenz verfügt.

7.6 Spezielle Code-Repositoires

Grössere Organisationen haben eigene Code-Repositories. Speziell hervorzuheben ist z.B. das deutsche Repository für Behörden opencode.de.[34] Gerade Repositories von anderen öffentlichen Verwaltungen sind hier von Interesse.

[32] https://www.openhub.net/
[33] https://www.openhub.net/p/_compare
[34] https://opencode.de/de

8. Support-Modelle beim Einsatz von OSS

Der Einsatz von bereits auf dem Markt vorhandener Open Source Software kann grundsätzlich auf drei Arten erfolgen:

1. Einsatz ohne professionellen Support,
2. Einsatz mit internem Support oder
3. Einsatz mit Support durch externen Anbieter.

Diese drei Einsatzarten sowie deren Vor- und Nachteile werden im Folgenden kurz erläutert. Welches dieser Szenarien im spezifischen Fall am sinnvollsten ist, muss situativ entschieden werden. Welches Support-Modell geeignet ist, hängt von der strategischen Relevanz der Open Source Software, von der technischen Einbindung und den vorhandenen Personal-Ressourcen ab.

Bei kritischer Software ist der Support auf jeden Fall professionell zu gewährleisten, ob dies intern oder extern geschieht. Der geplante Lebenszyklus sollte auch eine Rolle bei der Planung des Supports spielen. Es kann auch sein, dass der Support von mehreren Providern bezogen wird.

8.1 Einsatz ohne professionellen Support

Open Source Software wird kostenlos aus dem Internet heruntergeladen, installiert und so wie sie ist eingesetzt.

Vorteile	- Niedrige Kosten - Rasche Umsetzung
Nachteile	- Kein garantierter Support - Keine Haftungsansprüche
Risiko und Absicherung	Hohes Risiko: Es bestehen keinerlei Supportverträge oder Garantien und es existiert wenig bis kein Entwickler-Know-how bei der Organisation.
Typisches Einsatzgebiet	Open Source Standardsoftware, welche unabhängig von anderen Systemen aktualisiert werden kann.

8.2 Einsatz mit internem Support

Ein Unternehmen, beziehungsweise eine öffentliche Institution baut Know-how und Ressourcen zu bestimmten Open-Source-Lösungen auf, um diese langfristig einzusetzen. Dieses Vorgehen wird insbesondere in geschäftskritischen Bereichen angewendet.

Vorteile	- Hohe Flexibilität dank internem Know-how - Keine Anbieterabhängigkeiten
Nachteile	- Hohe Investitionen und grosser Zeitaufwand durch Know-how-Aufbau - Höhere interne Fixkosten für Mitarbeitende
Risiko und Absicherung	Mittleres Risiko: Der Support hängt von Know-how und Verfügbarkeit der internen IT ab.
Typisches Einsatzgebiet	Inhouse-Entwicklungen, strategisch eingesetzte Open Source Software, zu der vertieftes Know-how vorhanden ist.

8.3 Einsatz mit Support durch externen Anbieter

Ein externer Open-Source-Anbieter wird beigezogen, der die Einführung und Wartung der Open Source Software professionell begleitet. Dieses Vorgehen wird insbesondere auch in geschäftskritischen Bereichen angewendet, in denen unmittelbar vertieftes Know-how der Software vorhanden sein muss. Dies kann auch Teile oder die gesamte Freigabe umfassen.

Vorteile	- Direkter Zugang zum Know-how der Open Source Entwickler - Korrekturen und Weiterentwicklung auf Auftragsbasis - Auswahl verschiedener Open Source Anbieter - Verbindlichkeit, Absicherung von Compliance-Risiken
Nachteile	- Externe Kosten durch Open Source Anbieter - Know-how-Abhängigkeit zum Open Source Anbieter
Risiko und Absicherung	Niedriges Risiko: Gewährleistung geschieht gemäss Auftragsbeschreibung oder Service Level Agreement
Typisches Einsatzgebiet	Geschäftskritische Open Source Software, zu der kein oder wenig internes Entwickler-Know-How vorhanden ist

9. Ansprechstelle

Es gibt in der Bundesverwaltung keine dedizierte zentrale und verantwortliche Ansprechstelle im Sinne eines Open Source Program Offices (OSPO). Grundsätzlich sind die Ämter für die Umsetzung selbst zuständig.

Generelle Anfragen zu den OSS-Hilfsmitteln des Dokumentensets Em002 können an den Bereich DTI der Bundeskanzlei gerichtet werden: opensource@bk.admin.ch.

Anhänge

A. Änderungen gegenüber Vorversion

- Kapitel 4.1: Reifegrade ergänzt
- Kapitel 6: Beschaffung von OSS mit den neuen Dokumenten des BBL ergänzt und angepasst
- Kapitel 7.1 neu OSS-Katalog des Bundes (opensource.admin.ch)
- Weitere kleinere redaktionelle Anpassungen und Verbesserungen

B. Referenzen

Die Referenzen des Dokumentensets Em002 befinden sich im «Strategischen Leitfaden» Em002».

C. Abkürzungen

Ein Abkürzungsverzeichnis befindet sich im Hauptdokument Em002.
Ein Glossar befindet sich im Dokument «Em002-6 Häufig gestellte Fragen zur Veröffentlichung von OSS (FAQ-OSS»).

D. Geschäftsmodelle mit OSS

Open Source Software per se ist kein Geschäftsmodell, da im Gegensatz zu proprietärer Software der Verkauf von Software unter einer Open Source Lizenz keine Option darstellt. Dennoch bestehen für Unternehmen unterschiedliche Möglichkeiten, Geschäftsmodelle basierend auf Open Source Software zu betreiben. Denn wenn beispielsweise für eine Open Source Lösung professioneller externer Support bezogen werden soll, dann ist die Beschaffung von entsprechenden kommerziellen Dienstleistungen notwendig. Im Folgenden sind die vier häufigsten Geschäftsmodelle mit Open Source Software erläutert. Zusätzliche Aspekte dieser und weiterer Geschäftsmodelle sind im BITKOM Leitfaden *[BIT-KOM2023]* ausgeführt.

D.1 Services und Produkte basierend auf OSS

Unternehmen können basierend auf Open Source Software kommerzielle Services wie Web Hosting oder Cloud Computing anbieten, welche mit proprietärer Software wesentlich teurer ausfallen würden. Entsprechend bauen heute die meisten Startups, Online-Portale und E-Commerce Anbieter ihre Plattformen auf Open Source Software auf. Auch andere Technologiefirmen wie Telekomunternehmen, Streaming-Anbieter oder auch Hersteller proprietärer Software integrieren Open Source Software in ihre Software- und Hardware Produkte sowie Online-Services. So können die Unternehmen fortlaufend innovative Lösungen anbieten, die mit proprietärer Software kaum möglich wären.

D.2 Services für OSS

Explizit Dienstleistungen für ausgewählte Open Source Software erbringen sogenannte Open Source Anbieter. Diese verfügen über erfahrene Open Source Entwickler und können dadurch Support, Wartung, Betrieb, Weiterentwicklung, Beratung, Schulung und weitere Dienstleistungen für Open Source Software anbieten. Diese können beim Open Source Anbieter im Auftragsverhältnis oder Werkvertragsverhältnis bezogen werden. Solche Services für Open Source Software können öffentlich ausgeschrieben werden da keine Abhängigkeit zum jeweiligen Dienstleister (Vendor Lock-In) besteht. Wichtig bei solchen Ausschreibungen ist die Berücksichtigung von entsprechenden Kriterien, damit die tatsächlich kompetenten Dienstleister ausgewählt werden (siehe Kapitel 0 und 8, *[BITKOM2023]* Abschnitt 3.3 und *[Gu2024]*).

D.3 Subscriptions

Werden Services für Open Source Software in einer standardisierten, wiederkehrenden Form als eine Art «Service Level Agreements» (SLA) erbracht, werden diese als «Subscriptions» bezeichnet. Im Rahmen von solchen Subscriptions garantieren die Unternehmen beispielsweise fortlaufende Sicherheits-Updates, Support per E-Mail oder Telefon, Kompatibilität mit anderen Software- und Hardware-Produkten über Zertifizierungen, langfristige Wartungsleistungen und Absicherungen gegen Rechtsansprüche (Copyright, Patente). Im Gegenzug bezahlen Kunden Subscriptions pro Arbeitsplatz oder CPU, ähnlich wie Lizenzgebühren oder Nutzungsgebühren von proprietärer Software. Anders als bei proprietärer Software stellen Subscriptions für Open Source Software jedoch keine Voraussetzung für die Nutzung von Software dar, sondern begleichen lediglich den Mehrwert der jeweiligen Leistungen des Open Source Anbieters.

D.4 Dual Licensing

Wenn ein Unternehmen über das geistige Eigentum einer Software-Lösung verfügt und alle integrierten Open Source Komponenten unter einer permissiven Lizenz stehen, dann kann eine duale Lizenzierung oder auch Mehrfachlizenzierung angewendet werden. Diese erlaubt es der Entwicklerfirma, die Software einerseits unter einer Copyleft Open Source Lizenz zu veröffentlichen und andererseits unter einer proprietären Lizenz zu verkaufen. Diese kommerzielle Variante wird oftmals auch als «Enterprise» Version bezeichnet und beinhaltet typischerweise gewisse zusätzliche Funktionalitäten wie exklusive Schnittstellenintegrationen oder die Erlaubnis an den Käufer, die Software in eigene proprietäre Produkte zu integrieren. Je nach Ausprägung der Einschränkungen der Open Source Variante ist beim Einsatz von dual lizenzierter Software Vorsicht geboten, da der Bezug der Enterprise-Version unumgänglich sein kann und damit die Herstellerabhängigkeit wie bei üblicher proprietärer Software hoch ist.