Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Bundeskanzlei BK
Digitale Transformation und IKT-Lenkung DTI

14.09.2026

Em002-1 Praxisleitfaden für Open-Source-Software in der Bundesverwaltung

Version 2.0-0be65ca

Inhaltsverzeichnis

1. 1. Verwendung dieses Leitfadens 3
1.1. Absicht / Anspruchsgruppen 3
2. Begriffe 5
3. Potenzial und Herausforderungen 7
3.1. Potenzial von Open-Source-Software 8
3.2. Herausforderungen im Umgang mit OSS 9
4. Konstellationen für die Nutzung von und die Arbeit mit OSS 11
4.1. Reifegrade für OSS in der Bundesverwaltung 11
4.2. Nutzung unveränderter Open-Source-Software 13
4.3. Entwicklung mit Open-Source-Komponenten und Freigabe des Quellcodes 13
4.4. Mitarbeit in offenen Projekten (Contribution) 14
5. Eigenschaften und Auswahl von Open-Source-Lizenzen 14
6. OSS-Beschaffung 14
7. Auffindbarkeit von OSS-Lösungen 14
7.1. OSS-Katalog 14
7.2. OSS Directory 15
7.3. alternativeTo 15
7.4. Open-Source-Repositories 15
7.5. Open Hub 16
7.6. Besondere Code-Repositories 17
8. Supportmodelle für die OSS-Nutzung 17
8.1. Nutzung ohne professionellen Support 18
8.2. Nutzung mit internem Support 18
8.3. Nutzung mit Support durch einen externen Lieferanten 18
9. Anlaufstelle 19
10. Anhänge 19
10.1. Änderungen gegenüber der Vorversion 19
10.2. Referenzen 19
10.3. Abkürzungen 19

10.4. Geschäftsmodelle mit OSS 19

10.4.1. Dienstleistungen und Produkte auf Basis von OSS 20

10.4.2. Dienstleistungen für OSS 20

10.4.3. Subskriptionen 20

10.4.4. Duale Lizenzierung 20

> **ANMERKUNG** Dieses Dokument ist ein laufend weiterentwickelter Entwurf und Teil der Hilfsmittel, welche die Bundesverwaltung bei der Veröffentlichung von Open-Source-Code unterstützen. Weitere Informationen finden Sie im README.

1. 1. Verwendung dieses Leitfadens

Nach Artikel 9 des Bundesgesetzes über den Einsatz elektronischer Mittel zur Erfüllung von Behördenaufgaben (EMBAG) ist der Bund verpflichtet, den Quellcode von Software offenzulegen, die er zur Erfüllung seiner Aufgaben entwickelt oder entwickeln lässt. [1] Der Sektor DTI der Bundeskanzlei hat aus diesem gesetzlichen Auftrag in den Strategischen Leitlinien für Open-Source-Software übergeordnete Ziele abgeleitet.

Dieses Dokument dient als Praxisleitfaden zu den rechtlichen und strategischen Vorgaben. Es soll die Verwaltungseinheiten des Bundes bei der Nutzung, Beschaffung und Freigabe von Open-Source-Software unterstützen. Es bietet allen Interessierten einen umfassenden Überblick über das Thema und verweist auf weiterführende Informationen in den übrigen Hilfsmitteln.

Je nach Vorwissen und Bedarf können Sie die Leitfäden jedoch auch selektiv lesen, bevor Sie zu den anderen Hilfsmitteln übergehen. Die folgende Übersicht hilft Ihnen, den passenden Einstieg zu finden. Wenn Sie unsicher sind, empfiehlt es sich, zuerst den Leitfaden durchzulesen.

1.1. Absicht / Anspruchsgruppen

Allgemeine Einführung in Open-Source-Software

- Kapitel Begriffe
- Kapitel Potenzial und Herausforderungen
- Kapitel Auffindbarkeit von OSS-Lösungen
- Em002-5 Merkblatt OSS-Hilfsmittel
- Em002-3 Leitfaden OSS-Lizenzierung
- Em002-6 FAQ zu OSS

Allgemeine Einführung in das EMBAG für Entscheidungsträgerinnen und Entscheidungsträger

- Em002-5 Merkblatt OSS-Hilfsmittel
- Em002 Strategische Leitlinien für Open-Source-Software in der Bundesverwaltung

Reine Beschaffung von zu nutzender Software

- Kapitel Reifegrade für OSS in der Bundesverwaltung
- Kapitel Nutzung unveränderter Open-Source-Software
- Em002-7 Strategische Aspekte der Beschaffung und OSS
- Merkblatt Softwarebeschaffung und Art. 9 EMBAG [KBB-MB]
- Wegleitung OSS in der Beschaffung [BBL-WL]

1. SR 172.019: https://www.fedlex.admin.ch/eli/cc/2023/682/de

Neue IT-Entwicklung und Freigabe

- Kapitel Entwicklung mit Open-Source-Komponenten und Freigabe des Quellcodes
- Kapitel Mitarbeit in offenen Projekten (Contribution)
- Em002-2 Anleitung zur Veröffentlichung von OSS
- Em002-2.1 Checkliste OSS-Vorabklärung
- Em002-2.2 Checkliste OSS-Analyse und -Vorbereitung
- Em002-2.3 Checkliste OSS-Freigabe und -Veröffentlichung
- Em002-4 Leitfaden OSS-Community
- Em002-4.1 Checkliste OSS-Community

Beschaffung von zu erstellender Software (Dienstleistung oder Werk)

- Kapitel Reifegrade für OSS in der Bundesverwaltung
- Kapitel Entwicklung mit Open-Source-Komponenten und Freigabe des Quellcodes
- Kapitel Mitarbeit in offenen Projekten (Contribution)
- Em002-7 Strategische Aspekte der Beschaffung und OSS
- Werkzeugkasten BBL, einschliesslich der Wegleitung Open Source in der Beschaffung
- Lern- und Vorlagenplattform der öffentlichen Verwaltung des KBB (www.perimap.admin.ch) einschliesslich Softwarebeschaffung und Art. 9 EMBAG [KBB-MB]
- Wegleitung OSS in der Beschaffung [BBL-WL]
- Em002-2 Anleitung zur Veröffentlichung von OSS
- Em002-2.1 Checkliste OSS-Vorabklärung
- Em002-2.2 Checkliste OSS-Analyse und -Vorbereitung
- Em002-2.3 Checkliste OSS-Freigabe und -Veröffentlichung
- Em002-4 Leitfaden OSS-Community
- Em002-4.1 Checkliste OSS-Community

Rechtliche Abklärungen für die Freigabe

- Em002-6 Häufig gestellte Fragen zur Veröffentlichung von OSS (OSS-FAQ)
- Em002-3 Leitfaden OSS-Lizenzierung

Arbeitsgrundsätze für Projektleitende

- HERMES-Dokumente
- Em002-5 Merkblatt OSS-Hilfsmittel

Beiträge von Software an Dritte und Zusammenarbeit

- Em002-4 Leitfaden OSS-Community
- Em002-4.1 Checkliste OSS-Community
- Em002-7 Strategische Aspekte der Beschaffung und OSS
- Em002-3 Leitfaden OSS-Lizenzierung

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

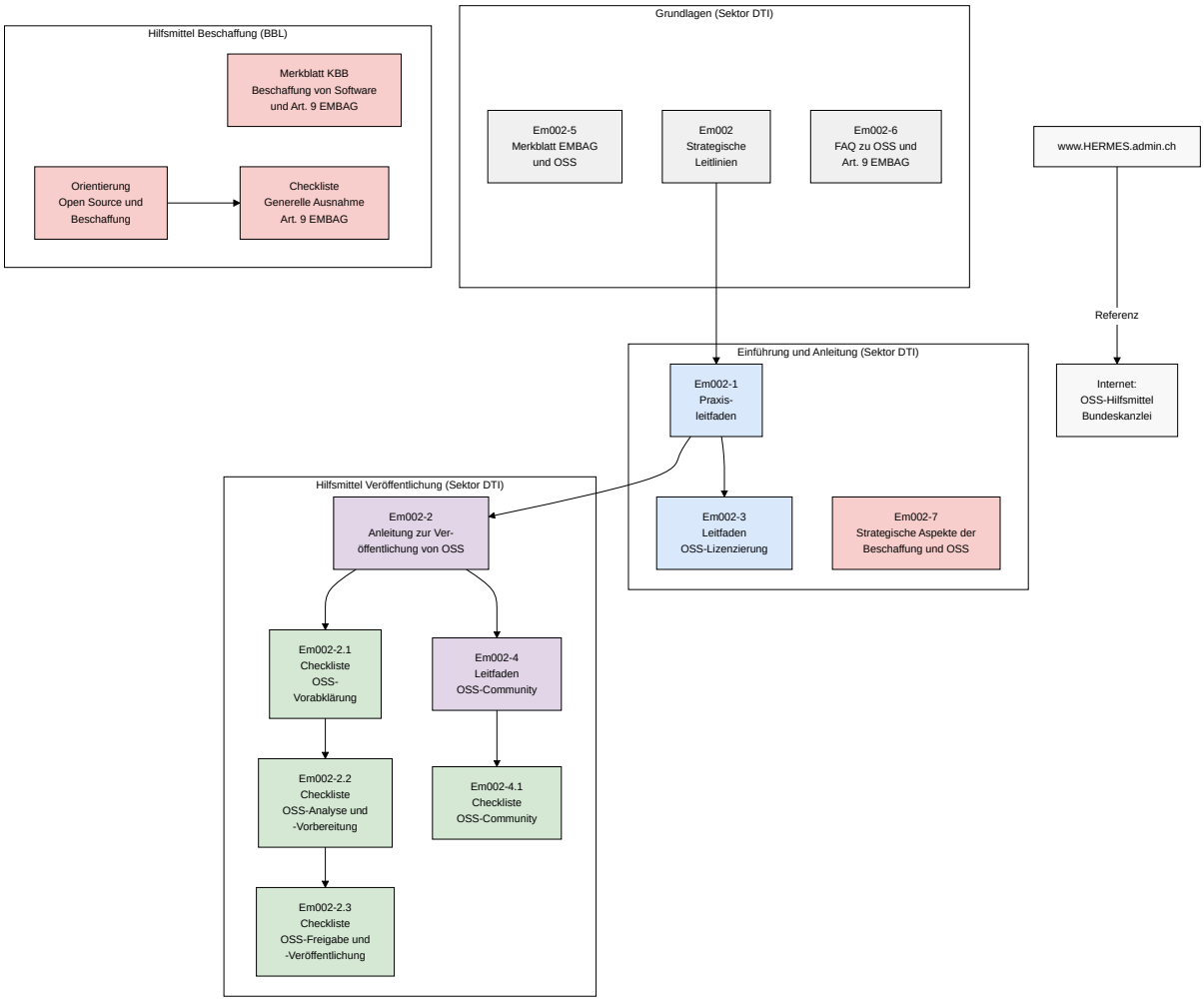


Abbildung 1: Übersicht über die OSS-Hilfsmittel im Zusammenhang mit Art. 9 EMBAG

2. Begriffe

Open-Source-Software

Software gilt als Open-Source-Software (OSS), wenn sie unter einer der rund 80 von der Open Source Initiative OSI anerkannten Lizenzen veröffentlicht wird *[OSI2024]*. Konkret legt die Definition von Open Source zehn Kriterien fest, die alle Open-Source-Lizenzen erfüllen müssen *[Pe1999]*. Diese lassen sich im Wesentlichen in vier Punkten zusammenfassen:

1. Die Software darf zu jedem Zweck genutzt werden.
2. Der Quellcode der Software ist frei zugänglich.
3. Die Software darf uneingeschränkt kopiert und verbreitet werden.

4. Die Software darf unter bestimmten Bedingungen verändert und in veränderter Form verbreitet werden.

Freie Software

Vor dem Begriff «Open Source» führte die Free Software Foundation (FSF) in den 1980er-Jahren das Konzept der «freien Software» ein. Im Kern erfüllt freie Software die Bedingungen von Open-Source-Software, zielt aber darauf ab, dass die Software stets frei zugänglich bleibt und nach Möglichkeit nicht in proprietäre Software integriert wird. [2]

Proprietäre Software

Bei proprietärer Software entwickelt ein Anbieter Software und verkauft der Nutzerin oder dem Nutzer eine Nutzungslizenz. In der Regel kennen die Nutzenden den Softwarecode nicht und dürfen die Software weder weiterverbreiten noch verändern. Sie können die Software nur gemäss den Lizenzbedingungen (z. B. End-User Licence Agreement, EULA) gegen Bezahlung von Lizenzgebühren nutzen. So kann die Software beispielsweise nur für eine bestimmte Anzahl Nutzender oder eine bestimmte Anzahl Prozessoren zugelassen sein. [3] Im Gegenzug für die meist jährlichen Wartungsgebühren verpflichtet sich der Anbieter, Fehler innerhalb einer angemessenen Frist zu beheben und die Software laufend weiterzuentwickeln.

Lizenz

Eine Lizenz ist ein Dokument, das verbindliche Vorgaben für die Nutzung und Verbreitung von Software enthält. Bei Open Source werden in der Regel Lizenzen verwendet, die der OSI entsprechen. [4]

Community

Eine Community im Kontext von Open-Source-Software ist breit definiert. Es kann sich um ein loses Ökosystem oder um eine Struktur mit Governance handeln, welche die Software entwickelt. Das Dokument Em002-4 Leitfaden OSS-Community behandelt dieses Thema. Mitglieder der Community können das Produkt gemeinsam führen, entwickeln, testen, übersetzen, Rückmeldungen geben oder die Software einfach nutzen.

Rechte Dritter

Auch bei Open-Source-Software können Schutzrechte bestehen (Urheberrecht, Markenrecht, Patentrecht) und von Dritten geltend gemacht werden, z. B. bei der Nutzung von Quellcode, der von Dritten erstellt wurde.

Ausnahmen nach EMBAG

Ausnahmen nach Art. 9 Abs. 1 EMBAG sind Rechte Dritter und sicherheitsrelevante Gründe. Beide werden in Kapitel 3 von Em002-2 Anleitung zur Veröffentlichung von Open-Source-Software behandelt.

Quellcode

In der Informatik ist der Quellcode (oder Quelltext) der für Menschen lesbare Text eines Computerprogramms, der in einer Programmiersprache verfasst ist. [5]

Veröffentlichung

Veröffentlichung bezeichnet in erster Linie die Freigabe des Quellcodes unter einer offenen Lizenz nach OSI. Aus praktischen Gründen gehören dazu in der Regel auch die Dokumentation und automatisierte Anweisungen, um die Anwendung aus dem Quellcode zu erstellen. Automatisierte Tests und Testdokumente sind üblicherweise ebenfalls enthalten. Damit sich eine Community entwickeln kann, wird eine Plattform eingesetzt, die dann mehrere Funktionen einer Entwicklungsplattform übernimmt (Build, Test, Support).

---

2. Freie Software — Wikipedia
3. Proprietäre Software — Wikipedia
4. Open Source Initiative
5. Quelltext — Wikipedia

Digitale Souveränität
: Wir verwenden die Definition aus dem Bericht zur digitalen Souveränität des EDA [St2024]: «Die digitale Souveränität eines Staates oder einer Organisation umfasst zwingend die vollständige Kontrolle über gespeicherte und verarbeitete Daten sowie die selbstständige Entscheidung darüber, wer darauf zugreifen darf. Sie umfasst weiter die Fähigkeit, technologische Komponenten und Systeme eigenständig zu entwickeln, zu verändern, zu kontrollieren und zu ergänzen.»

Repository
: Ein Repository ist ein Speicherort, den ein Versionsverwaltungswerkzeug für Dateien und Metadaten zur Codebasis nutzt. Repositories erlauben es mehreren Mitwirkenden, an denselben Dateien zu arbeiten und verschiedene Versionen abzulegen. Die meisten ermöglichen zudem Issue-Tracking, Release-Management, automatisierte Builds und Dokumentation.

Fork
: Ein Fork [6] eines Open-Source-Projekts ist der Vorgang, eine eigenständige Entwicklung des ursprünglichen Projekts aufzubauen.

Contributor Licence Agreement (CLA)
: Ein Contributor Licence Agreement [7] ist ein Dokument, das die Bedingungen beschreibt, unter denen geistiges Eigentum zu einem Projekt oder Vorhaben beigetragen werden kann.

Source Code Management (SCM)
: Source Code Management [8], auch Versionsverwaltung genannt, ist ein Werkzeug, das Änderungen und Versionsnummern insbesondere in der Softwareentwicklung wirksam verwaltet. Durchgesetzt hat sich die freie Software Git [9] zur Verwaltung verteilter SCM.

Open Source Software Development (OSSD)
: Bei der Open-Source-Softwareentwicklung (OSSD) [10] wird zusätzlich zur Veröffentlichung des Quellcodes der gesamte Entwicklungsprozess öffentlich durchgeführt. Alles von den Anforderungen (Issues) bis zum Quellcode ist transparent. OSSD wird unterstützt durch öffentliche Kommunikationswerkzeuge (Mailinglisten, Foren usw.), ein Versionsverwaltungssystem (Git), Fehler- und Funktionslisten, eine Roadmap und Entwicklerwerkzeuge.

Software Package Data Exchange (SPDX)
: Software Package Data Exchange (SPDX) [11] beschreibt ein Standardformat für Software Bill of Materials (SBOM) mit dem Ziel, den korrekten Umgang mit freier Software oder Open-Source-Software zu erleichtern.

Copyleft-Effekt
: Steht Software unter einer Lizenz mit Copyleft-Bestimmung, so muss jede Veränderung oder Erweiterung des Quellcodes wieder unter der Lizenz der veränderten Open-Source-Software freigegeben werden (siehe auch Em002-3 Leitfaden OSS-Lizenzierung).

3. Potenzial und Herausforderungen

In der heutigen IT-Welt besteht ein enormes Potenzial für Open-Source-Software. Gleichzeitig bringt der Einsatz von Open-Source-Lösungen auch Herausforderungen mit sich. Diese beiden Seiten werden im folgenden Abschnitt näher erläutert. Die Angaben beruhen unter anderem auf den Ergebnissen der Open Source Studie Schweiz 2018, 2021 und 2024 der Universität Bern, in der die Befragten Auskunft über die erkannten Vor- und Nachteile von Open-Source-Software gaben.

6. Abspaltung (Softwareentwicklung) — Wikipedia
7. Contributor License Agreement — Wikipedia
8. Versionsverwaltung — Wikipedia
9. Git — Wikipedia
10. Open-source software development — Wikipedia
11. Software Package Data Exchange — Wikipedia

3.1. Potenzial von Open-Source-Software

Die folgenden Punkte beschreiben das Potenzial, das sich bei der Nutzung und Freigabe von Open-Source-Software realisieren lässt.

1. Digitale Souveränität
Dabei geht es um die Fähigkeit, digitale Dienste zu nutzen und zu kontrollieren. Die digitale Souveränität erstreckt sich über den gesamten Lebenszyklus eines digitalen Systems. Open-Source-Software fördert die digitale Souveränität, da sie Austauschbarkeit, Gestaltungsfreiheit und Einfluss garantiert. [12]

2. Keine Lizenzgebühren
Für die Nutzung von Open-Source-Software fallen keine Lizenzkosten an. Beim Bezug komplexer Open-Source-Standardsoftwarepakete kann es jedoch sinnvoll sein, kostenpflichtige Support-Subskriptionen zu erwerben.

3. Kosteneinsparungen durch Zusammenarbeit mit anderen Nutzenden
Da Software unter einer Open-Source-Lizenz uneingeschränkt genutzt und weiterentwickelt werden darf, können Kosten von Weiterentwicklungen geteilt oder bestehende Zusatzentwicklungen anderer Verwaltungseinheiten nach dem Grundsatz «einmal entwickeln – mehrfach nutzen» übernommen werden. Gleichzeitig besteht die Möglichkeit, von den Erfahrungen und Entwicklungen anderer zu profitieren.

4. Community-Aufbau und Wissensaustausch
Open-Source-Software erleichtert den Aufbau von Communities und den Wissensaustausch, beispielsweise zwischen den verschiedenen föderalen Ebenen der Schweiz. Software kann gemeinsam weiterentwickelt, Fehler können behoben und individuelle Erfahrungen geteilt werden. Der verstärkte Wissensaustausch zwischen verschiedenen Verwaltungseinheiten führt zu einem besseren Verständnis dafür, woran andere arbeiten, sodass Doppelspurigkeiten vermieden werden und sich die besten Lösungen verbreiten können.

5. Geringere Herstellerabhängigkeit
Vendor-Lock-in (Abhängigkeit von Softwarelieferanten) gilt in der Informatik als sehr hoch. Der Einsatz von Software unter einer Open-Source-Lizenz verringert den Vendor-Lock-in, da Betrieb, Wartung, Support, Weiterentwicklung und weitere Dienstleistungen für Open-Source-Software offen ausgeschrieben werden können. Statt aufgrund eines Vendor-Lock-in ein ganzes System vollständig neu zu beschaffen, können Erweiterungen oder Lebenszyklusleistungen für eine OSS-Lösung auch separat ausgeschrieben werden. Dadurch lassen sich erhebliche Migrationskosten einsparen.

6. Offene Standards und Interoperabilität
Bei Open-Source-Software ist die Kompatibilität mit anderen Softwarelösungen und IT-Systemen (Interoperabilität) in der Regel höher als bei proprietärer Software. Open-Source-Lösungen verwenden zudem fast ausschliesslich offene Datenformate, weshalb sie sich einfach mit anderen Systemen austauschen lassen.

7. Transparenz über den Aufbau der Software
Da die Software auch in Form des Quellcodes vorliegt, kann ihre Qualität beispielsweise durch externe Reviews geprüft werden. Zudem lässt sich auf Basis des Quellcodes Dokumentation erstellen (z. B. im Hinblick auf neue Ausschreibungen für Weiterentwicklungsleistungen oder die Ablösung von Open-Source-Software am Ende ihrer Nutzungsdauer).

---

12. Wir verwenden die Definition aus *[St2024]*: «Die digitale Souveränität eines Staates oder einer Organisation umfasst zwingend die vollständige Kontrolle über gespeicherte und verarbeitete Daten sowie die selbstständige Entscheidung darüber, wer darauf zugreifen darf. Sie umfasst weiter die Fähigkeit, technologische Komponenten und Systeme eigenständig zu entwickeln, zu verändern, zu kontrollieren und zu ergänzen.»

8. Sicherheit und Vertrauen durch Transparenz
Aufgrund der Offenheit des Quellcodes kann die Sicherheit von Open-Source-Software höher sein als jene proprietärer Software. Zudem ist es deutlich schwieriger, Hintertüren und andere Lücken in Open-Source-Software einzubauen, was zu mehr Vertrauen in die Software führt.

9. Höhere Qualität und Modularität des Codes
Die Qualität von Open-Source-Software kann höher sein als jene proprietärer Software, weil die Motivation, guten Code zu schreiben, grösser sein dürfte, wenn Entwicklerinnen und Entwickler wissen, dass ihr Quellcode veröffentlicht wird. Zudem sind Open-Source-Lösungen tendenziell stark modular aufgebaut, sodass sich einzelne Module leicht ersetzen und die übrigen Module weiterverwenden lassen. [13]

10. Einfache Anpassung
Der Zugriff auf den Quellcode erlaubt es den Nutzenden, Weiterentwicklungen selbst oder durch externe Lieferanten vorzunehmen. So kann die Software rasch an die eigenen Bedürfnisse angepasst werden.

11. Rasche Innovation und Integration
Open-Source-Software wird durch die Community rasch und kontinuierlich weiterentwickelt. So werden neue Technologien und Datenstandards häufig als Open-Source-Programmbibliotheken veröffentlicht. Dies ermöglicht die rasche Umsetzung innovativer Softwarelösungen.

12. Arbeitgeberattraktivität
Der Einsatz moderner Open-Source-Technologien und die informelle Zusammenarbeit mit internationalen Communities fördern die Motivation der Mitarbeitenden und erhöhen damit die Arbeitgeberattraktivität, was wiederum die Rekrutierung junger, qualifizierter Fachpersonen erleichtert.

3.2. Herausforderungen im Umgang mit OSS

Im Folgenden werden die typischen Herausforderungen beschrieben, die in der Praxis mit Open-Source-Software auftreten, und einige Lösungsansätze skizziert. Auf allgemeine Herausforderungen, die sämtliche Software betreffen, gehen wir an dieser Stelle nicht ein (z. B. die Notwendigkeit der Prüfung auf Cybersicherheit und die Notwendigkeit, für kritische Software angemessenen Support sicherzustellen).

1. Hohe Wechselkosten durch bestehende Abhängigkeiten
Die Ablösung proprietärer Software durch Open-Source-Lösungen kann sehr kostspielig sein, wenn die Anwendung durch Schnittstellenintegration oder andere Abhängigkeiten eng im bestehenden IT-System eingebettet ist. Diese Wechselkosten führen dazu, dass sich die Einführung von Open-Source-Software oft erst am Ende des Lebenszyklus der proprietären Software lohnt.

2. Fehlende Funktionen oder keine passende Open-Source-Lösung
Bei der Beschaffung von Standardsoftware kann es vorkommen, dass es zur (bisherigen) proprietären Lösung keine Open-Source-Alternative gibt oder nur eine funktional unzureichende. Als Lösungsansatz besteht die Möglichkeit, die fehlenden Funktionen gemeinsam mit anderen

13. Ein weiterer Sicherheitsaspekt ist, dass sich die verwendeten Bibliotheken (einschliesslich Abhängigkeiten) leichter dokumentieren lassen als bei proprietärer Software. Das wiederum bedeutet, dass Schwachstellen beim Auftreten rasch erkannt und behoben werden können (siehe auch SBOM).

Nutzenden der Open-Source-Software zu entwickeln und dem Open-Source-Projekt hinzuzufügen. [14]

3. Höhere Integrationskosten

Kurzfristige Kosteneinsparungen durch wegfallende Lizenzen beim Einsatz von Open-Source-Software werden oft durch höhere Integrationskosten aufgewogen. Daher ist es wichtig zu erkennen, dass der professionelle Einsatz von Open-Source-Software nicht «gratis» ist, sondern tatsächlich interne oder externe Kosten verursacht.

4. Unklare Verantwortlichkeiten und Support

Open-Source-Software wird häufig für den fehlenden Support und die fehlende Wartung durch externe Unternehmen kritisiert. Inzwischen gibt es jedoch eine Vielzahl von Anbietern, die kommerzielle Leistungen (Long-Term-Support, laufende Wartung, Weiterentwicklung, Haftung und Gewährleistung usw.) für Open-Source-Lösungen anbieten. Die verschiedenen Geschäftsmodelle mit Open-Source-Software werden in Anhang D und die verschiedenen Supportmodelle in Kapitel 8 erläutert. Eine aktuelle Übersicht über Open-Source-Dienstleister und ihre Leistungen findet sich im Open Source Directory (https://www.ossdirectory.com) oder in der OSS-Studie 2024 [15] *[OSS2024]*.

5. Kleiner Markt mit wenigen Anbietern

Da bestimmte Open-Source-Anwendungen, etwa Desktop-Anwendungen, in der Regel wenig oder keinen Support und keine Wartung benötigen, kann der Markt für Open-Source-Anbieter von Support und Wartung klein sein. In diesem Fall kann eine Ausschreibung für Open-Source-Softwaredienstleistungen dazu beitragen, einen Markt zu schaffen, der den Wettbewerb zwischen den Anbietern erhöht. Das Bestehen einer breiten Entwicklungs-Community für die betreffende Software und die Qualität der Entwicklungsdokumentation sind bei der Ausschreibung solcher Leistungen von praktischer Bedeutung.

6. Geringe Sichtbarkeit

Communities von Open-Source-Projekten konzentrieren sich vor allem auf die Produktentwicklung und selten auf Marketing. Hersteller proprietärer Software investieren hingegen stark in die Vermarktung und den Verkauf ihrer Produkte. Diese fehlende Werbung für Open-Source-Software erweckt oft den falschen Eindruck, dass es zu proprietären Produkten keine Alternative gebe. Es gibt jedoch Plattformen wie alternativeTo [16], die einen Überblick über Lösungen für bestimmte Aufgaben bieten (mehr zu OSS-Alternativen im Kapitel «Auffindbarkeit von OSS-Lösungen»).

7. Fehlende Akzeptanz bei Endnutzenden

Aufgrund unterschiedlicher Benutzeroberflächen, fehlender Funktionen, geringer Benutzerfreundlichkeit und wenig Werbung für Open-Source-Software fehlt bei bestimmten Open-Source-Produkten die Akzeptanz der Endnutzenden. Geeignete Kommunikations-, Dokumentations- und Schulungsangebote können dieser Herausforderung entgegenwirken. Auch der Umstand, dass Open-Source-Software nach den Bedürfnissen der Nutzenden weiterentwickelt werden kann, kann zu höherer Akzeptanz führen.

8. Wenig oder kein internes Fachwissen

Open-Source-Lösungen entwickeln sich rasch weiter und erfordern gleichzeitig ein vertieftes technisches Verständnis. So kann es vorkommen, dass interne Mitarbeitende wenig oder kein Fachwissen zu bestimmter Open-Source-Software haben. Internes OSS-Know-how lässt sich durch Weiterbildung und Möglichkeiten zum Selbststudium über Online-Quellen sowie durch den Aufbau interner Wikis usw. aufbauen.

---

14. Für solche neuen Funktionen sind die Projekt- oder Applikationsverantwortlichen zuständig. Wie genau dies geschieht, bestimmt das Projekt. Em002-2 und Em002-4 helfen bei detaillierten Entscheiden. Je nach Umfang der Funktion kann es sinnvoll sein, die Anleitung nicht vollständig zu durchlaufen. Beschaffungsrechtliche Aspekte für Ressourcen sind zu beachten.
15. OSS-Studie
16. https://alternativeto.net/

9. Schwierige Zukunftseinschätzung
Für Aussenstehende einer Open-Source-Community ist oft schwer erkennbar, wie sich das Projekt künftig entwickeln wird. Daher ist es wichtig, die künftige Entwicklung einer Open-Source-Lösung realistisch einschätzen zu können. Das Kapitel «Open Hub» stellt hierzu die Plattform Open Hub vor, die eine Einschätzung der Aktivität und Heterogenität der Entwickler-Community erlaubt. Dadurch lässt sich die künftige Entwicklung eines Open-Source-Projekts besser beurteilen.

10. Rechtliche Unsicherheiten
Die Vielzahl unterschiedlicher Open-Source-Lizenzen und die bislang geringe Zahl von Gerichtsurteilen zu Auslegungsfragen können bei Open-Source-Software mitunter zu rechtlichen Unsicherheiten führen. Dieser Praxisleitfaden soll einen Überblick über die wichtigsten Open-Source-Lizenzen sowie deren Eigenschaften und Kompatibilitäten geben. Quellen für vertiefte Antworten auf rechtliche Fragen finden sich in den Dokumenten Em002-3 Leitfaden OSS-Lizenzierung und Em002-6 FAQ zu OSS und Art. 9 EMBAG.

11. Zu umfangreiches Angebot an Open-Source-Software
Die Zahl der Open-Source-Softwareprodukte hat in den letzten Jahren stark zugenommen. Potenzielle Nutzende von Open-Source-Software beklagen daher die Vielzahl bestehender Open-Source-Lösungen auf dem Markt. Aus diesem Grund stellt das Kapitel «Auffindbarkeit von …» zwei Plattformen vor, alternativeTo und Open Hub, die einen Vergleich von Open-Source-Lösungen ermöglichen.

12. Koordinationsaufwand bei gemeinsamer Entwicklung
Wenn sich die Bundesverwaltung an gemeinsamen Projekten beteiligt und dazu beiträgt, entsteht innerhalb dieser Zusammenarbeit ein Koordinationsaufwand, aber auch dann, wenn eine Community ins Boot geholt werden muss. Idealerweise lohnt sich dieser Mehraufwand dennoch dank einer verbesserten Lösung, grösserer Akzeptanz oder verstärkter Beteiligung Dritter.

4. Konstellationen für die Nutzung von und die Arbeit mit OSS

4.1. Reifegrade für OSS in der Bundesverwaltung

Bei der Nutzung und Entwicklung von OSS in der Bundesverwaltung bestehen verschiedene Konstellationen hinsichtlich der Pflichten im Zusammenhang mit Open-Source-Lizenzen: Ein typisches Reifegradmodell [17] für die Einführung von Open Source in Organisationen kann folgendermassen aussehen:

1. Denial — keine oder unbewusste Nutzung von Open-Source-Software
2. Consumption — passive Nutzung von Open-Source-Software
3. Participation — Engagement in Open-Source-Communities
4. Contribution — pragmatische Beiträge zu Open-Source-Projekten
5. Leadership — strategisches Engagement mit Open Source zur Steigerung des Geschäftswerts [18]

17. Reifegradmodell in Anlehnung an https://todogroup.org/resources/guides/a-guide-to-outbound-open-source-software/#maturity-levels
18. Creation wird nicht erwähnt, da Art. 9 EMBAG dies direkt verlangt.

Konstellation	Auswirkung
Die blosse **Nutzung** bestehender Open-Source-Software (ohne Veränderung des Codes).	Es besteht **keine Pflicht**, den Quellcode zu teilen.
Die **vollständige Neuentwicklung** von Software, die unter einer Open-Source-Lizenz veröffentlicht werden soll.	**In diesem Fall besteht völlige Freiheit bezüglich der Lizenz, unter der die Software veröffentlicht wird.** Für den Bund ist das Dokument Em002-3 Leitfaden OSS-Lizenzierung heranzuziehen. Die Dokumente Em002-2 Anleitung zur Veröffentlichung von OSS und Em002-4 Leitfaden OSS-Community sind zu beachten.
Weiterentwicklung (intern oder extern) mit bestehenden Open-Source-Komponenten (mit oder ohne Copyleft-Effekt [19]), solange die Software **ausschliesslich innerhalb derselben Organisation** genutzt und nicht weiterverbreitet wird.	Die OSS-Komponenten können beliebig mit Drittkomponenten kombiniert werden. Aus der Lizenz ergibt sich grundsätzlich keine Veröffentlichungspflicht, solange die Software ausschliesslich innerhalb derselben Organisation genutzt und nicht verbreitet wird (Ausnahme: AGPL). Da rechtlich nicht abschliessend geklärt ist, wie weit der Begriff «Verbreitung innerhalb derselben Organisation» reicht, gilt dieses Szenario für OSS-Komponenten mit Copyleft-Effekt nur in Ausnahmefällen. Ziehen Sie in diesen Fällen den Rechtsdienst Ihrer Organisation bei. Siehe Em002-3 Leitfaden OSS-Lizenzierung und Em002-6 FAQ zu OSS und Art. 9 EMBAG. Die Dokumente Em002-2 Anleitung zur Veröffentlichung von OSS und Em002-4 Leitfaden OSS-Community sind zu beachten. **Da nach Art. 9 EMBAG eine Veröffentlichungspflicht besteht, müssen die Änderungen erneut freigegeben werden. Am einfachsten geschieht dies als Änderungen am bestehenden Open-Source-Projekt.**
Weiterentwicklung (intern oder extern) **ausschliesslich mit** bestehenden Open-Source-**Komponenten ohne Copyleft-Effekt**, wenn die Software extern verbreitet werden soll (z. B. an Kantone).	Die OSS-Komponenten können beliebig mit Drittkomponenten kombiniert werden, und aus den Lizenzen ergibt sich keine Veröffentlichungspflicht (siehe auch Em002-3 Leitfaden OSS-Lizenzierung). **Im Rahmen von Art. 9 EMBAG ist die Freigabe jedoch obligatorisch. Idealerweise erfolgt die Freigabe über das bestehende Projekt (kein Fork).** Die Dokumente Em002-2 Anleitung zur Veröffentlichung von OSS und Em002-4 Leitfaden OSS-Community sind zu beachten.
Weiterentwicklung (intern oder extern) mit bestehenden Open-Source-Komponenten **mit Copyleft-Effekt**, wenn die Software extern verbreitet werden soll (z. B. an Kantone).	**Aufgrund der Copyleft-Lizenz besteht eine Veröffentlichungspflicht** (siehe auch Em002-3 Leitfaden OSS-Lizenzierung zum Copyleft-Effekt). Die Dokumente Em002-2 Anleitung zur Veröffentlichung von OSS und Em002-4 Leitfaden OSS-Community sind zu beachten.

19. Erläuterung des Begriffs «Copyleft» in Em002-3 Leitfaden OSS-Lizenzierung

Konstellation	Auswirkung
Beitrag zu einem bestehenden Open-Source-Projekt.	**Nach EMBAG besteht grundsätzlich eine Veröffentlichungspflicht. Es gelten die Governance und die Lizenz des Projekts.** Die Dokumente Em002-2 Anleitung zur Veröffentlichung von OSS und Em002-4 Leitfaden OSS-Community sind zu beachten.
Gemeinsame Entwicklung eines neuen oder bestehenden Projekts mit einer Community und Kostenteilung.	**Nach EMBAG besteht grundsätzlich eine Veröffentlichungspflicht.** Die Dokumente Em002-2 Anleitung zur Veröffentlichung von OSS, Em002-3 Leitfaden OSS-Lizenzierung und Em002-4 Leitfaden OSS-Community dienen **dem Aufbau des Projekts und seiner Governance.**
Sämtliche Varianten der Entwicklung oder Weiterentwicklung erfolgen **über einen Lieferanten oder beauftragte Dritte.**	**Nach EMBAG besteht grundsätzlich eine Veröffentlichungspflicht.** Grundlage bilden Em002-2 Anleitung zur Veröffentlichung von OSS, Em002-3 Leitfaden OSS-Lizenzierung und Em002-4 Leitfaden OSS-Community. Wenn möglich sollten die Voraussetzungen für die Veröffentlichung bereits im Beschaffungsprozess geschaffen und nötigenfalls die Anforderungen an die Softwarefreigabe dem Lieferanten bzw. den Dritten mitgeteilt werden.

4.2. Nutzung unveränderter Open-Source-Software

Bei der Nutzung von Open-Source-Software ist grundsätzlich nicht relevant, unter welcher Open-Source-Lizenz sie veröffentlicht wird, da alle Open-Source-Lizenzen (siehe Kapitel 2 «Begriffe») die uneingeschränkte Nutzung von Open-Source-Software erlauben, unabhängig davon, auf wie vielen Arbeitsplätzen, mit wie vielen gleichzeitig angemeldeten Nutzenden, auf wie vielen Servern und Prozessoren usw. die Software eingesetzt wird.

Bei der Nutzung unveränderter Open-Source-Software entstehen **keine Veröffentlichungspflichten**, auch nicht bei Programmen, die unter einer Lizenz mit striktem Copyleft-Effekt stehen (Ausnahme: AGPL-Lizenz, Affero General Public Licence).

Solange Open-Source-Software durch interne Leistungserbringer des Bundes eingeführt und betrieben wird, kann dies **ohne öffentliches Beschaffungsverfahren** erfolgen, denn «die OSS-Lizenz allein kostet die Beschaffungsstelle in der Regel nichts und ist daher für sich genommen nicht beschaffungsrelevant».

Wird ein externer Leistungserbringer für Wartung, Support und weitere Dienstleistungen für die Open-Source-Software beauftragt, so sind die Vorgaben des öffentlichen Beschaffungsrechts zu beachten. Verschiedene Supportvarianten werden in Kapitel 9 beschrieben. Wichtig ist zudem, bei der Beschaffung von Open-Source-Software geeignete Eignungs- und Zuschlagskriterien zu verwenden. Die Grundlagen dazu werden in Kapitel 8 erläutert.

4.3. Entwicklung mit Open-Source-Komponenten und Freigabe des Quellcodes

Mit der Veröffentlichungspflicht nach Art. 9 EMBAG ist hier das in Em002-2 Anleitung zur Veröffentlichung von OSS beschriebene Vorgehen zu befolgen. Die Dokumente Em002-3 Leitfaden OSS-Lizenzierung und Em002-4 Leitfaden OSS-Community sind nötigenfalls ebenfalls beizuziehen. 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 (Contribution)

Mögliche Überlegungen zum Beitrag der Bundesverwaltung an offene, bereits bestehende Projekte sind im BITKOM-Leitfaden *[BITKOM2024]* Kapitel 4.2 aufgeführt. [20] Je nach Bedeutung des Projekts für die Bundesverwaltung ist zu prüfen, wie viel Verantwortung das jeweilige Amt übernehmen will und kann.

Die Mitarbeit in offenen Projekten und die direkte Entwicklung in offenen Projekten können ebenfalls anhand der Dokumente für die Freigabe erfolgen. Im Mittelpunkt stehen Em002-4 Leitfaden OSS-Community und die zugehörige Checkliste *Em002-4.1*.

5. Eigenschaften und Auswahl von Open-Source-Lizenzen

Die Eigenschaften und die Auswahl von Open-Source-Lizenzen werden im Dokument Em002-3 Leitfaden OSS-Lizenzierung beschrieben.

6. OSS-Beschaffung

Strategische Fragen zu Beschaffung und OSS werden in Em002-7 Strategische Aspekte der Beschaffung und Open-Source-Software behandelt.

Das Kompetenzzentrum Beschaffungswesen Bund (KBB) hat ein neues Merkblatt *Softwarebeschaffung und Art. 9 EMBAG [KBB-MB]* sowie zwei Dokumente «Musterkriterien – Beschaffung EMBAG und Open Source» und «Mustervertragstexte – Softwareentwicklung» veröffentlicht.

Weitere Unterstützung steht im BBL-Intranet (nur im Bundesnetz zugänglich) mit der *Wegleitung Open Source in der Beschaffung [BBL-WL]* und der *Checkliste Integral-Ausnahme Art. 9 EMBAG [BBL-CL]* zur Verfügung.

Die Enterprise-Tauglichkeit von Open Source findet sich im Dokument *Selection Criteria for Enterprise-Ready Open Source Software [Gu2024]* [21] oder im Bitkom-Leitfaden. [22]

7. Auffindbarkeit von OSS-Lösungen

Verschiedene Plattformen erlauben es interessierten Organisationen, Open-Source-Software vor der Nutzung oder Beschaffung zu finden und zu analysieren. Die daraus gewonnenen Daten bilden eine Grundlage für den Entscheid über den Einsatz oder die Beschaffung von Open-Source-Software.

Eine Zusammenstellung von Mindestanforderungen und gegebenenfalls eine Marktanalyse sollten in jedem Fall dienen.

7.1. OSS-Katalog

Die Bundesbehörden sind verpflichtet, den Quellcode von Software offenzulegen, die sie zur Erfüllung ihrer Aufgaben entwickeln oder entwickeln lassen. Der OSS-Katalog [23] listet sämtliche Software auf, die seit dem 1. Januar 2024 von Bundesbehörden entwickelt und als Open Source freigegeben wurde. Er bietet einen Überblick über verfügbare Software und erleichtert das Auffinden und die Wiederverwendung von Open-Source-Lösungen.

---

20. Siehe https://www.bitkom.org/Bitkom/Publikationen/Bitkom-Leitfaden-zu-Open-Source-Software-20.html, Kapitel 4.2.
21. Siehe *[Gu2024]*
22. Siehe Praxisempfehlungen für Open-Source-Software | Leitfaden 2025 | Bitkom e. V., Kapitel 3.3
23. https://www.opensource.admin.ch/

Wird ein neues Repository oder eine neue Organisation erstellt, so ist dies für die Aufnahme in den OSS-Katalog per E-Mail an opensource@bk.admin.ch zu melden.

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

7.2. OSS Directory

Das von CH Open betriebene OSS Directory [24] erleichtert das Auffinden qualifizierter Anbieter, die professionellen Support für Open-Source-Software anbieten.

Open-Source-Anbieter zeigen anhand von Erfolgsgeschichten, welchen Nutzenden (Kunden) sie bei der Einführung bestimmter Open-Source-Produkte geholfen haben. Das Directory bietet zudem aktuelle Open-Source-News aus bestehenden Plattformen, kuratierte Top-News, Veranstaltungen, Communities, Stellenangebote und Artikel zum Wissensaustausch. Das OSS Directory ist auf Deutsch, Englisch, Französisch und Italienisch verfügbar. Sämtliche Einträge können von registrierten Nutzenden direkt hinzugefügt und aktualisiert werden.

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

7.3. alternativeTo

Ein praktisches Werkzeug zum Auffinden von Alternativen zu Software ist «alternativeTo – Crowdsourced software recommendations». [25] Wie der Name andeutet, sind die Alternativen und ihre Bewertungen durch Crowdsourcing entstanden, also durch die Eingaben vieler einzelner Nutzender. alternativeTo unterscheidet vier Arten von Softwarelösungen, von denen nur die erste als Open-Source-Software im Sinne von Art. 9 EMBAG gilt:

Free open source
: Software, die unter einer Open-Source-Lizenz veröffentlicht wird.

Free
: Kostenlose Software (Freeware), deren Quellcode jedoch nicht frei verfügbar ist und nicht verändert werden darf.

Freemium
: Solche Software bietet eine kostenlose und eine Premium-Version, wobei alle wichtigen Funktionen bereits in der kostenlosen Version verfügbar sind. Nicht zu dieser Kategorie gehört Software, die während einer bestimmten Zeit kostenlos getestet werden kann (z. B. eine 30-Tage-Testversion); diese fällt in die Kategorie «Commercial».

Commercial
: Dabei handelt es sich um proprietäre Software, für deren Nutzung Lizenzgebühren erhoben werden.

7.4. Open-Source-Repositories

Die derzeit weltweit beliebteste Open-Source-Entwicklungsplattform mit über 30 Millionen Nutzenden und über 100 Millionen Quellcode-Repositories ist GitHub. [26] Heute haben praktisch alle IT-Unternehmen, insbesondere Hersteller proprietärer Software, sowie viele andere Arten von Unternehmen und Organisationen eine GitHub-Präsenz. Immer mehr Behörden – rund ein Dutzend aus der Schweiz [27] – veröffentlichen ihre eigene Open-Source-Software zudem auf «GitHub and Government» unter https://government.github.com. Dieser Vorgang lässt sich auch mit anderen Repositories durchführen.

---

24. https://www.ossdirectory.com/de/home
25. https://alternativeto.net/
26. https://octoverse.github.com
27. https://government.github.com/community/#switzerland

Beispiele für andere Repositories sind:

- GitLab
- Bitbucket
- SourceForge
- LaunchPad

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

Pulse
: Überblick über die jüngsten Aktivitäten eines Open-Source-Projekts auf GitHub: Zusammenfassung der wichtigsten Informationen zu Entwicklungsintensität, Heterogenität der Community, offenen Meldungen, Verbesserungen (Pull Requests) usw.

Contributors
: Darstellung, welche Entwicklerinnen und Entwickler wann aktiv waren. Dies ist ein wichtiger Hinweis darauf, ob alles von einer Person abhängt oder ob eine grössere Community dahintersteht.

Commits
: Darstellung, welche Beiträge in welchem Zeitraum zu diesem Open-Source-Projekt 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 (z. B. relevant für das Erkennen von Sicherheitslücken und Updates).

Network
: Darstellung, wann welche Entwicklerin bzw. welcher Entwickler zu welchem Entwicklungszweig beigetragen hat.

Forks
: Liste aller Kopien des Open-Source-Projekts auf GitHub. Indikator für die Beliebtheit und Verbreitung der Open-Source-Software.

7.5. Open Hub

Sollen Informationen zu einer Open-Source-Lösung gesammelt werden, die nicht zwingend auf GitHub entwickelt wird, bietet sich Open Hub von Black Duck an. [28] Für rund eine halbe Million Open-Source-Projekte werden zahlreiche wichtige Informationen übersichtlich zusammengefasst:

Project Summary
: Eine kurze Beschreibung des Open-Source-Projekts.

In a Nutshell
: Die wichtigsten Fakten zu einem Open-Source-Projekt, etwa die Zahl der Commits, der Mitwirkenden und der Codezeilen sowie die meistverwendete Programmiersprache, der Zeitpunkt des ersten Commits und der letzten Änderung. Zudem wird eine Einschätzung der Codebasis und der Grösse des Entwicklungsteams aufgeführt (z. B. «Mozilla Firefox has a well-established, mature codebase maintained by a very large development team with stable Y-O-Y commits»).

28. https://www.openhub.net/

Quick Reference
Enthält Informationen zur Organisation, Links zum Projekt und zum Code sowie Hinweise auf ähnliche Projekte.

Licences
Gibt an, unter welcher Lizenz bzw. welchen Lizenzen das Open-Source-Projekt lizenziert ist und welche Folgen damit verbunden sind.

Project Security
Liefert Angaben zur Sicherheit und zu Schwachstellen des Open-Source-Projekts.

Code
Eine Grafik, welche die Zahl der Codezeilen im Zeitverlauf zeigt, aufgeschlüsselt nach Programmiersprache.

Activity
Eine Grafik, welche die Zahl der Commits pro Monat zeigt. Zusätzlich wird eine Zusammenfassung der letzten 30 Tage und 12 Monate bereitgestellt.

Community
Eine Grafik, welche die Zahl der aktiven Mitwirkenden pro Monat zeigt. Zudem wird eine Bewertung des Projekts auf einer Fünf-Sterne-Skala angezeigt.

Open Hub erlaubt es auch, verschiedene Open-Source-Projekte zu vergleichen. [29] Dies verschafft rasch einen Überblick darüber, welches Projekt die aktivste Community, die längste Entwicklungszeit oder die am besten geeignete Lizenz hat.

7.6. Besondere Code-Repositories

Grössere Organisationen verfügen über eigene Code-Repositories. Besonders zu erwähnen ist beispielsweise das deutsche Repository für Behörden opencode.de. [30] Von besonderem Interesse sind hier Repositories anderer öffentlicher Verwaltungen.

8. Supportmodelle für die OSS-Nutzung

Bereits auf dem Markt verfügbare Open-Source-Software kann grundsätzlich auf drei Arten genutzt werden:

1. Nutzung ohne professionellen Support
2. Nutzung mit internem Support
3. Nutzung mit Support durch einen externen Lieferanten.

Diese drei Nutzungsarten und ihre Vor- und Nachteile werden nachfolgend kurz erläutert. Welches dieser Szenarien im Einzelfall am sinnvollsten ist, muss fallweise entschieden werden. Welches Supportmodell geeignet ist, hängt von der strategischen Relevanz der Open-Source-Software, der technischen Integration und den verfügbaren Personalressourcen ab.

Bei kritischer Software muss der Support in jedem Fall professionell erbracht werden, sei es intern oder extern. Auch der geplante Lebenszyklus sollte bei der Supportplanung eine Rolle spielen. Es kann auch sein, dass Support von mehreren Anbietern bezogen wird.

29. https://www.openhub.net/p/_compare
30. https://opencode.de/de/home

8.1. Nutzung ohne professionellen Support

Open-Source-Software steht im Internet zum kostenlosen Herunterladen, Installieren und Nutzen zur Verfügung.

Vorteile

- Geringe Kosten
- Rasche Umsetzung

Nachteile

- Kein garantierter Support
- Keine Haftungsansprüche

Risiko und Absicherung
Hohes Risiko: Es bestehen keine Supportverträge oder Garantien und in der Organisation ist wenig oder kein Entwicklerwissen vorhanden.

Typischer Anwendungsbereich
Open-Source-Standardsoftware, die unabhängig von anderen Systemen aktualisiert werden kann.

8.2. Nutzung mit internem Support

Ein Unternehmen oder eine Organisation der öffentlichen Hand baut für die langfristige Nutzung Fachwissen und Ressourcen zu bestimmten Open-Source-Lösungen auf. Dieser Ansatz ist besonders in geschäftskritischen Bereichen verbreitet.

Vorteile

- Hohe Flexibilität dank internem Know-how
- Kein Lieferanten-Lock-in

Nachteile

- Hoher Investitions- und Zeitaufwand für den Aufbau von Fachwissen
- Höhere interne Fixkosten für Personal

Risiko und Absicherung
Mittleres Risiko: Der Support hängt vom Know-how und der Verfügbarkeit der internen IT ab.

Typischer Anwendungsbereich
Eigenentwicklungen, strategisch eingesetzte Open-Source-Software, für die vertieftes Know-how vorhanden ist.

8.3. Nutzung mit Support durch einen externen Lieferanten

Ein externer Open-Source-Anbieter wird beigezogen, um die Einführung und Wartung der Open-Source-Software professionell zu begleiten. Dieser Ansatz wird vor allem in geschäftskritischen Bereichen eingesetzt, in denen vertieftes Know-how der Software sofort verfügbar sein muss. Dies kann auch Teile der Freigabe oder die gesamte Freigabe umfassen.

Vorteile

- Direkter Zugang zum Know-how der Open-Source-Entwicklerinnen und -Entwickler

- Fehlerbehebungen und Erweiterungen auf Vertragsbasis
- Auswahl unter verschiedenen Open-Source-Lieferanten
- Verbindlichkeit, Absicherung gegen Compliance-Risiken

Nachteile

- Externe Kosten durch Open-Source-Lieferanten
- Know-how-Abhängigkeit vom Open-Source-Lieferanten

Risiko und Absicherung
Geringes Risiko: Die Gewährleistung erfolgt gemäss Pflichtenheft oder Service Level Agreement

Typischer Anwendungsbereich
Geschäftskritische Open-Source-Software, für die wenig oder kein internes Entwicklungs-Know-how vorhanden ist

9. Anlaufstelle

In der Bundesverwaltung gibt es keine zentrale Anlaufstelle, die als Open Source Programme Office tätig wäre. Grundsätzlich sind die Bundesämter für die Umsetzung selbst verantwortlich.

Allgemeine Anfragen zu den OSS-Hilfsmitteln des Dokumentensatzes Em002 können an den Sektor DTI der Bundeskanzlei gerichtet werden: opensource@bk.admin.ch.

10. Anhänge

10.1. Änderungen gegenüber der Vorversion

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

10.2. Referenzen

Die Referenzen zum Dokumentensatz Em002 finden sich in den Em002 Strategischen Leitlinien.

10.3. Abkürzungen

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

10.4. Geschäftsmodelle mit OSS

Open-Source-Software ist an sich kein Geschäftsmodell, weil es – anders als bei proprietärer Software – nicht möglich ist, Software unter einer Open-Source-Lizenz zu verkaufen. Dennoch bestehen für Unternehmen verschiedene Möglichkeiten, Geschäftsmodelle auf Basis von Open-Source-Software zu betreiben. Soll etwa professioneller externer Support für eine Open-Source-Lösung bezogen werden, so ist die Beschaffung entsprechender kommerzieller Leistungen erforderlich. Die vier häufigsten Geschäftsmodelle für Open-Source-Software werden nachfolgend erläutert. Weitere Aspekte dieser und anderer Geschäftsmodelle werden im BITKOM-Leitfaden *[BITKOM2023]* erläutert.

10.4.1. Dienstleistungen und Produkte auf Basis von OSS

Unternehmen können kommerzielle Dienstleistungen wie Webhosting oder Cloud Computing auf Basis von Open-Source-Software anbieten, die mit proprietärer Software wesentlich teurer wären. Deshalb bauen die meisten Start-ups, Onlineportale und E-Commerce-Anbieter ihre Plattformen heute auf Open-Source-Software auf. Andere Technologieunternehmen wie Telekommunikationsunternehmen, Streaminganbieter oder gar Hersteller proprietärer Software integrieren Open-Source-Software in ihre Software- und Hardwareprodukte sowie in Onlinedienste. Dies erlaubt es Unternehmen, weiterhin innovative Lösungen anzubieten, die mit proprietärer Software nur schwer zu erreichen wären.

10.4.2. Dienstleistungen für OSS

Open-Source-Anbieter erbringen Dienstleistungen für ausgewählte Open-Source-Software. Sie verfügen über erfahrene Open-Source-Entwicklerinnen und -Entwickler und können daher Support, Wartung, Betrieb, Entwicklung, Beratung, Schulung und weitere Leistungen für Open-Source-Software anbieten. Diese können beim Open-Source-Lieferanten im Auftrag oder im Rahmen eines Werkvertrags bezogen werden. Solche Dienstleistungen für Open-Source-Software können öffentlich ausgeschrieben werden, da kein Vendor-Lock-in besteht. Wichtig bei solchen Ausschreibungen ist die Berücksichtigung geeigneter Kriterien, damit tatsächlich kompetente Leistungserbringer ausgewählt werden (siehe Kapitel 0 und 8, *[BITKOM2023]* Kapitel 3.3 und *[Gu2024]*).

10.4.3. Subskriptionen

Werden Dienstleistungen für Open-Source-Software in standardisierter, wiederkehrender Form als eine Art Service Level Agreement (SLA) erbracht, so spricht man von Subskriptionen. Im Rahmen solcher Subskriptionen garantieren Unternehmen beispielsweise laufende Sicherheitsupdates, Support per E-Mail oder Telefon, Kompatibilität mit anderen Software- und Hardwareprodukten durch Zertifizierungen, langfristige Wartungsleistungen und Absicherungen gegen rechtliche Ansprüche (Urheberrecht, Patente). Im Gegenzug bezahlen Kundinnen und Kunden Subskriptionen pro Arbeitsplatz oder CPU, ähnlich wie Lizenz- oder Nutzungsgebühren für proprietäre Software. Anders als bei proprietärer Software sind Subskriptionen für Open-Source-Software jedoch keine Voraussetzung für die Nutzung der Software, sondern lediglich eine Möglichkeit, den Mehrwert der vom Open-Source-Lieferanten erbrachten Leistungen abzugelten.

10.4.4. Duale Lizenzierung

Besitzt ein Unternehmen das geistige Eigentum an einer Softwarelösung und stehen alle integrierten Open-Source-Komponenten unter einer permissiven Lizenz, so kann eine duale oder mehrfache Lizenzierung angewendet werden. Dies erlaubt es dem entwickelnden Unternehmen, die Software unter einer Copyleft-Open-Source-Lizenz zu veröffentlichen und sie zugleich unter einer proprietären Lizenz zu verkaufen. Diese kommerzielle Version wird oft Enterprise-Version genannt und enthält typischerweise bestimmte Zusatzfunktionen, etwa exklusive Schnittstellenintegrationen oder die Erlaubnis für Käuferinnen und Käufer, die Software in eigene proprietäre Produkte zu integrieren. Je nach Ausmass der Einschränkungen der Open-Source-Version ist beim Einsatz dual lizenzierter Software Vorsicht geboten, da der Bezug der Enterprise-Version unumgänglich sein kann, womit der Vendor-Lock-in gleich hoch ist wie bei typischer proprietärer Software.