# Em002-4 Leitfaden OSS Community

**Empfehlung zur Bundesinformatik**[1]

Dieses Dokument ist eine eigenständige Beilage zum Hauptdokument Em002.

| | |
|---|---|
| Klassifizierung:[2] | Nicht klassifiziert |
| Verbindlichkeit:[3] | Empfehlung |
| Planungsfeld:[4] | IKT der Bundesverwaltung |
| Diese Version: | 2.0 |
| Ersetzt Version: | V1.0 vom 24.02.2025 |
| Status: | Genehmigt |
| Freigabedatum (diese Version): | 9.12.2025 |
| Freigegeben durch, Rechtsgrundlage: | Der Delegierte für digitale Transformation und IKT-Lenkung (D-DTI), gestützt auf Artikel 40 der Verordnung vom 1. Mai 2025 über die digitalen Dienste und die digitale Transformation in der Bundesverwaltung (Digitalisierungsverordnung, DigiV), SR 172.019.1 |
| Sprachen: | Deutsch (Original), Französisch, Italienisch, Englisch (Übersetzung) |
| Lizenz | CC0 1.0 Universal<br>Dieses Dokument ist unter der CC0-Lizenz veröffentlicht. Es darf frei verwendet, verändert und weitergegeben werden, auch für kommerzielle Zwecke und in jedem Format |

[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 Das Wichtigste in Kürze

Im Folgenden sind die wichtigsten Punkte des Dokuments aufgeführt.

- Der Aufbau oder die Pflege einer OSS Community ist gemäss Art. 9 EMBAG **nicht gefordert.**
- Eine Community ist dann relevant, wenn eine User Group sinnvoll ist und **Synergien** mit anderen Anwendern der Software geschaffen werden sollen.
- Eine Community beschränkt sich nicht nur auf die Software selbst, sondern kann auch die Prozesse im Umfeld und die Standardisierung von Daten umfassen.
- Ein **initialer Effort** muss für den Aufbau einer Community immer geleistet werden. Dieser dauert meist an. In vielen Fällen macht es Sinn, wenn das federführende Amt auch die Koordinationskosten übernimmt.
- Communities sollen **einen Nutzen** erfüllen. Es braucht Interessenten und aktive Beteiligte.
- Eine Community kann auch von selbst entstehen und erst später formalisiert werden.
- Das Ziel der Community ist es, den langfristigen **Nutzen für alle** zu steigern.
  Motto: «Ich bekomme mindestens so viel wie ich investiere».
- Die Community sollte **so einfach wie möglich strukturiert** sein.
- In einem ersten Schritt soll die Community kurz anhand der Checkliste *[Em002-4.1*] geprüft werden. Erst wenn sie sinnvoll ist, sollte sie weiter aufgebaut werden.
- Die Formalisierung geschieht über ein **Community-Konzept**. Dabei sollte möglichst viel aus bestehenden Quellen übernommen werden
  Motto: «Das Rad nicht neu erfinden».
- Wenn eine Zusammenarbeit (Kollaboration) angestrebt wird, so muss dies ganz zu Beginn des Projekts berücksichtigt werden, da potenzielle Partner bereits bei der Anforderungsaufnahme abgeholt werden sollten. Es gilt auch dies rechtlich und beschaffungsrechtlich von Beginn weg gut zu planen.
- Der Beitritt zu einer bestehenden Kollaboration sind die rechtlichen und beschaffungsrechtlichen Möglichkeiten zu prüfen.
- Bei Kontributionen muss keine Community aufgebaut werden. Falls es vom Projekt gefordert ist, muss aber das entsprechende **Contributor License Agreement (CLA)** unterschrieben werden oder die Entwickler müssen durch das Projekt zur Abgabe der Software unter einem Developer Certificate of Origin (DCO) berechtigt werden.

# 2 Einleitung

Dieser Leitfaden ist für IKT-Fachpersonen bestimmt, die für eine Applikation verantwortlich sind und diese als Open Source anbieten wollen oder an einer solchen mitarbeiten.

Dies muss entweder formell oder informell organisiert werden.

Dieses Dokument liefert wichtige Informationen für die Organisation und den Aufbau einer Community[5] und auch zu offener Entwicklung direkt als Open-Source-Projekt.

Als zweites Element wird zur Förderung der Standardisierung ein Verzeichnis von bereits bestehenden Community-Konzepten als Teil dieses Dokuments geführt. Die Idee ist, dass mit minimalem Aufwand basierend auf erfolgreichen Lösungen neue Communities aufgebaut werden können.

Dieses Dokument beinhaltet keine direkten Empfehlungen für die Organisation von Communities für Softwarebibliotheken (Teile einer Software, die Teilfunktionen erfüllen, z.B. die Generierung von PDFs). Für diese wird im Normalfall kein eigenes Konzept entwickelt, sondern sie verwenden die in der im Template der Ablage des Repositories definierten, simplen Vorlage.

*«Em002-01 Praxis-Leitfaden Open Source Software in der Bundesverwaltung»* enthält in Kapitel 4 grundsätzliche Zusammenarbeitsformen, in Kapitel 8 die verschiedenen Supportmodelle und im Anhang Angaben zum Marktverhalten von Open Source.

[5] Die Community eines Open-Source-Projektes stellt typischerweise eine Pyramide dar.
Fundament dieser Pyramide sind die Nutzerinnen und Nutzer der Software, im Speziellen die engagierten, die sich aktiv an der Community beteiligen, beispielsweise in Form von Bug Reports und Feature-Wünschen oder durch Beiträge in Mailinglisten.
In der Pyramide direkt darüber sind Kontributorinnen und Kontributoren angesiedelt. Dies sind Teile der Community, die eigene Codebeiträge als Kontributionen vorschlagen. Sie haben in der Regel keine Schreibrechte auf das Repository. Ihre Beiträge werden von Maintainern des Projekts begutachtet und nach Erreichen der projekttypischen Qualität in das Repository übernommen.
An der Spitze der Pyramide stehen die Maintainer oder Committer eines Projekts. In dieser Rolle hat eine Entwicklerin eine höhere Verantwortung. Dies äußert sich zum Beispiel darin, dass sie Kontributionen annehmen kann. Dieses Recht äußert sich häufig durch Schreibrechte auf das Repository. Auf dieser Ebene findet die Kontrolle über die Software, deren Qualität und Funktionsvielfalt statt. In komplexeren Projekten kann diese Stufe noch weiter aufgefächert werden. Bekannt ist zum Beispiel der Linuxkernel, in dem es Subsystemmaintainer auf mehreren Stufen gibt und letztlich mit Linus Torvalds nur ein einzelner Entwickler die Patches in das Projektrepository übernimmt. Ein weiteres Beispiel sind Projekte der Eclipse Foundation, in diesen gibt es typischerweise noch einen Projekt-Lead, die zusätzliche Rechte im Kontext des Eclipse Foundation Entwicklungsprozesses hat, so kann sie zum Beispiel den Releaseprozess auslösen oder die formelle Wahl von Committern oder Projekt-Leads initiieren *[BITKOM2023]*.

# 3 Art, Ziel und Zweck einer Community

Eine Community kann die folgenden Zwecke verfolgen:

- Erhebung zusätzlicher Anforderungen
- Verbreitung von Wissen über die Anwendung
- Besseres Feedback von Nutzern
- Übersetzungen, Dokumentationen und Training fördern
- Standardisierung verbreiten
- Generell verbesserte Interaktion mit Nutzern
- Erweiterung der Zusammenarbeit um andere Parteien
- Förderung von auf der Software aufbauender weiterer Software

Welche Art der Community für eine Applikation am besten geeignet ist, hängt sehr stark vom fachlichen Umfeld, den möglichen Nutzerinnen und Nutzern, sowie der Strategie der veröffentlichenden Institution ab. Es gibt sehr viele verschiedene Varianten Communities aufzubauen. Im Idealfall kann bei einem geeigneten Projekt auch einer bereits bestehenden Community beigetreten werden. Alternativ kann eine Community mit einem oder mehreren gleichwertigen Partnern aufgebaut werden.

Wichtig ist, dass **eine Governance** festgelegt wird und die relevanten Punkte in einem **Community-Konzept** festgehalten sind (siehe auch *[BITKOM2023]* Abschnitt 4.1 und *[IZqCab2023]*.

Es gilt zu beachten, dass Communities für Software auch mit solchen für Daten, Standardisierung und die Geschäftsthematik kombiniert werden können. Die Community kann dann Prozesse, Standardisierung, Software und Daten umfassen. Es können u.U. auch bestehende Gefässe wie eOperations, DVS und eCH verwendet werden.

Wichtiger Aspekt einer funktionierenden Community ist, dass die Teilnehmenden mehr zurückerhalten als sie investieren.

Es gibt folgende Konstellationen:

- **Community um eine OSS des Bundes**: Governance und Community-Konzept muss erstellt werden. In diesem Zusammenhang ist auch zu regeln, wie Dritte Kontributionen machen können.
- **Neue Kollaboration für die Lösung eines Problems**: Neben Governance und Community-Konzept muss die ganze rechtliche Struktur der Zusammenarbeit aufgebaut werden.
- **Beitritt zu einer Kollaboration**: Hier muss die Bundesverwaltung prüfen, ob und wie sie das machen kann.
- **Kontribution an ein Drittprojekt**: Der zu prüfende Punkt ist die Policy des Projekts für Kontribution (Contributor License Agreement, Developer Certificate of Origin u.ä.).

Die beschaffungsrechtlichen Fragen werden in *«Em002-7 Strategische Aspekte zu Beschaffung und Open Source Software»* behandelt.

## 3.1 Spezialfall: Neue Kollaboration zur Erstellung, Wartung und Support von Open Source Software

Die kollektiven Kosten bei Open Source Software sinken, wenn gemeinschaftlich entwickelt und dann auch weiterentwickelt wird. Das kann auch ohne direkte Kooperationen geschehen oder nur mit gemeinsamer Planung.

Wird eine formellere Kollaboration angestrebt, so sollte mittels Marktabklärung sichergestellt werden, dass die Kollaboration die sinnvolle Variante zur Zusammenarbeit ist. Die Wirtschaftlichkeit bei Kollaborationen ist höher, als wenn jeder selbst entwickelt (siehe aus *«Em002-4.1 Checkliste OSS Community»*.

Es gilt zu beachten, dass Kollaborationen sinnvollerweise sehr früh in HERMES[6], nämlich bereits bei der Situationsanalyse und bei den Anforderungen, aufgegleist werden sollten.

Die Zusammenarbeit ist basierend auf Art. 4 EMBAG in der Schweiz grundsätzlich erlaubt.

Eine Kollaboration haben neben allem anderen eine rechtliche und eine beschaffungsrechtliche Komponente. Im Moment sind es immer Einzelfallbetrachtungen. Mit der Zeit werden sich standardisierte Muster etablieren.

### 3.1.1 Rechtliche Komponente

Vereinsmitgliedschaften sind machbar. Es handelt sich dabei meist um Einzelfallbetrachtung: Wer steht dahinter, welche Verpflichtungen werden eingegangen. Die Delegation von Mitgliedschaften an eOperations oder ähnliche Organisationen wären möglich.

Solange für die Zusammenarbeit kein Geld fliesst und keine formellen Verpflichtungen eingegangen werden müssen, ist ein Memorandum of Understanding auf Ebene Amt oder Bundeskanzler möglich.

Es gilt zu vermeiden, dass ein Staatsvertrag erstellt werden muss.
Zusätzlich gibt es zwei Spezialfälle[7], bei denen eine Kollaboration bereits abgedeckt wäre:

- Aufträge aufgrund eines internationalen Abkommens betreffend die gemeinsame Umsetzung eines Projekts durch Unterzeichnerstaaten.
- Aufträge, die gemäss den besonderen Verfahren oder Bedingungen einer internationalen Organisation vergeben werden.

### 3.1.2 Beschaffungsrechtliche Komponente

In vielen Fällen wird die Kollaboration zwischen Organisationen der öffentlichen Hand stattfinden. Dort ist eine Beschaffung unproblematisch, wenn es sich um «in-state» handelt (siehe *«Em002-7 Strategische Aspekte zu Beschaffung und Open Source Software»*).

Verträge mit ausländischen Firmen ohne Sitz in der Schweiz sind möglich, solange das Beschaffungsrecht eingehalten wird.

Die Beschaffung der eigenen Ressourcen erfolgt im Rahmen einer regulären Beschaffung oder ist bereits erfolgt.

[6] https://www.hermes.admin.ch/de/pjm-2022/verstehen/aufgaben/ausschreibung-erarbeiten.html
[7] https://www.eda.admin.ch/deza/de/home/partnerschaften/auftraege-beitraege/auftraege/anforderungen/rechtlich.html

## 3.2 Spezialfall: Beitritt zu einer bestehenden Kollaboration

Hier muss einerseits eine rechtliche Grundlage gegeben sein und andererseits muss die beschaffungsrechtliche Seite geregelt sein.

Am einfachsten ist es, sich über die «jeder trägt seine Kosten zu beteiligen». Dies kann auch Overhead umfassen, wenn entweder eigene oder korrekt ausgeschriebene Ressourcen verwendet werden.

Ein direkter Beitritt zu ausländischen Kollaborationen ist schwieriger. Eine einfache Assoziation über ein Memorandum Of Understanding ist am einfachsten zu erreichen.

## 3.3 Spezialfall: Kontribution von Software bei OSS-Projekten von Dritten.

Wenn die Bundesverwaltung Anpassungen am Code einer bestehenden OSS vornimmt, dann macht es Sinn, diese direkt «up-stream» im Originalprojekt zu ändern. Dies führt dann dazu, dass die Wartung nicht mehr durch die Bundesverwaltung gemacht werden muss. Aufgrund von Art. 9 EMBAG ist dies sogar erwünscht. Beschaffungsrechtlich ist dies unproblematisch.

Wichtig ist, dass der Inhaber der Rechte die Kontribution vornehmen muss. Dies führt dazu, dass die Bundesverwaltung dies für ihre Mitarbeitenden und Beauftragten übernimmt. Namentlich muss dazu das allfällige Contributor License Agreement des Projekts unterzeichnet werden oder ein Developer Certificate of Origin muss im Pull Request mitgeliefert werden.

Ein **Developer Certificate of Origin[8] (DCO)** für die Bundesverwaltung könnte lauten:

> Developer Certificate of Origin
> Version 1.1
>
> Copyright (C) 2004, 2006 The Linux Foundation and its contributors.
>
> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
>
> Developer's Certificate of Origin 1.1
>
> By making a contribution to this project, we certify that:
>
> (a) The rights to this contribution are owned by the Swiss Confederation and I Am authorized to submit it under the open source license indicated in the file; or
>
> (b) The contribution is based upon previous work that, to the best of my knowledge, is covered under an appropriate open source license and I have the right under that license to submit that work with modifications, whether created in whole or in part by me, under the same open source license (unless I am permitted to submit under a different license), as indicated in the file; or
>
> (c) The contribution was provided directly to me by some other person who certified (a), (b) or (c) and I have not modified it.
>
> (d) I understand and agree that this project and the contribution are public and that a record of the contribution (including all personal information I submit with it, including my sign-off) is maintained indefinitely and may be redistributed consistent with this project or the open source license(s) involved.

Der Aufbau eines Contributor License Agreements ist in *«Em002-3 OSS-Lizenzen»* Abschnitt 8.9 beschrieben.

Die effektive Übergabe des Codes geschieht nach den Richtlinien des jeweiligen Projekts.

[8] https://developercertificate.org/

# 4 Community-Konzept

Am Schluss ist der Kern einer Community dessen Struktur und Governance. Diese werden in einem Community-Konzept festgehalten.

Das vom Projekt- bzw. vom Applikationsverantwortlichen zu erstellende (oder bei Dritten zu bestellende) Community-Konzept soll der folgenden Kapitelstruktur folgen. Je nach Art der spezifischen Community sind nicht alle Unterkapitel erforderlich. Im Idealfall wird in vielen Kapiteln auf bereits standardisierte Dokumente verwiesen, bzw. Standardprozesse der entsprechenden Repositories verwendet. Das Community-Konzept kann auch für bereits bestehende und von anderen geführten Projekten angewandt werden, insbesondere wenn die effektive Governance noch nicht schriftlich festgehalten ist.

Durch eine Formalisierung der Community werden alle Teilnehmenden abgeholt und das Onboarding von neuen Teilnehmenden vereinfacht.

Strukturentwurf für ein Community-Konzept:

1. Übersichtstabelle
   (Name, Repository, Kontaktadresse, Lizenz, Tiefe des Supports, bestehend/neu)
2. Zielsetzung
   a. Der Applikation
   b. Der Community
3. Organisation
   a. Eigentümerin / Eigentümer des Codes
   b. Organisationsform / Gremien
   c. Stimmrechte
   d. Projektmanagement
   e. Beschaffungsfragen (falls sie sich stellen)
   f. Andere Governance-Aspekte
4. Roadmap und Change-Prozess
5. Entwicklungsprozess
   a. Open Development
   b. Committer-Rechte
   c. Review-Prozess
   d. Aufgabenverteilung und Namensnennungen
   e. Backups
6. Finanzierung der Community
7. Vermarktung
8. Umgang mit externen Beiträgen
   a. Umgang mit gemeldeten Fehlern
   b. Umgang mit Anfragen
   c. Umgang mit Pull Requests
   d. Umgang mit Forks
   e. CLA, DCO und Abtretung Rechte/Geistiges Eigentum
9. Betrieb der Applikation

Grundsätzlich ist jede Bundesbehörde selbst für die Erstellung der Community-Konzepte zuständig. DTI kann bestehende Muster/Beispiele publizieren und nimmt die entsprechenden Beispiele zur Publikation entgegen.
Es macht auch Sinn, wenn sich Community-Verantwortliche untereinander austauschen.

Eine mögliche Quelle für Teile sind auch die Dokumente der Eclipse Foundation[9], wobei hier ein entsprechendes Tailoring, im Sinne von «nur das machen, was nötig ist», durchgeführt werden sollte.

[9] Siehe https://www.eclipse.org/projects/handbook/#starting

# 5 Grundsatzentscheide beim Aufbau einer Community

Die Grundsätze der Community, oder Fehlen derselbigen, wird anhand von *«Em002-4.1 Checkliste OSS Community»* ermittelt. Die relevanten Grundsatzfragen ergeben sich aus dem morphologischen Kasten in Abbildung 1.

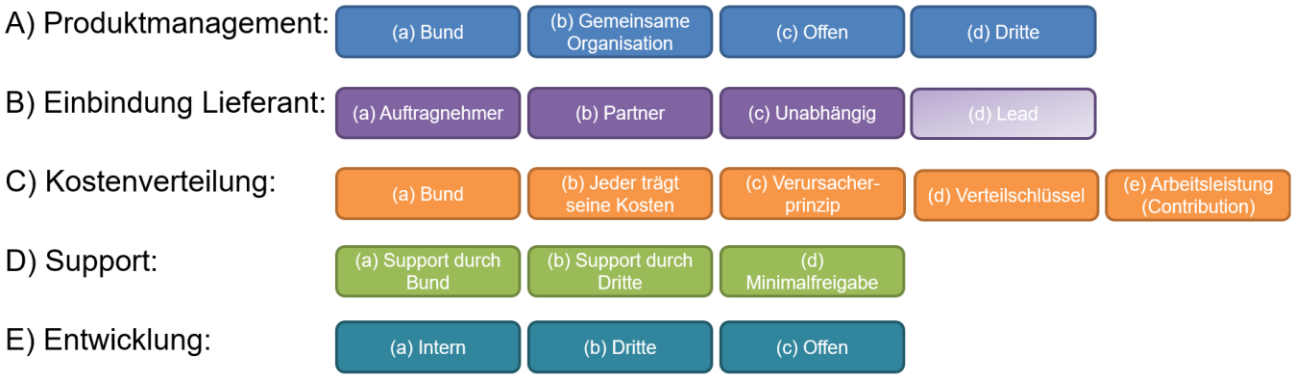


Abbildung 1 Morphologischer Kasten OSS Community

Es empfiehlt sich, für die Ausgestaltung der Community **vor allem auf die nähere Zukunft zu fokussieren**; wenn also beispielsweise noch keine konkreten Interessentinnen oder Interessenten für die Applikation vorhanden sind, dann macht es meist wenig Sinn, eine Community mit einer einfachen Gesellschaft, einem eigenen Verein und komplexen Prozessen zur Erarbeitung der Roadmap zu definieren[10][11].

## 5.1 Produktmanagement

Je nach Art der Applikation, ihrer strategischen Bedeutung und ihrem Status im Lebenszyklus gibt es unterschiedliche Varianten, das Produktmanagement abzubilden. Bei Organisationen innerhalb von Bundesbehörden betrifft dies immer die Anwendungsverantwortlichen. Insbesondere relevant für das Community-Konzept sind die Definition der fachlichen und technischen Roadmap sowie die Release-Prozesse.

Möglichkeiten:

a) **Organisationen innerhalb des Bundes**: Der Bund will die Kontrolle behalten, er definiert allein die Roadmap.
b) **Gemeinsame Organisation**: Gemeinsam mit anderen Nutzenden oder Lieferantin.
c) **Offene Organisation**: Lose Verbindungen, keine aktive Roadmap.
d) **Dritte**: Entweder besteht das Produkt bereits oder die Arbeit soll ausgelagert werden.

### 5.1.1 Organisationen innerhalb des Bundes

Wenn die Applikation eine hohe strategische Bedeutung hat oder der Bund darauf angewiesen ist, seine Änderungen schnell implementieren zu können, dann kann es Sinn machen, die Kontrolle zu behalten.

[10] Mit einem Verein (oder einer Stiftung) wird eine eigene juristische Person geschaffen, die sich um die Software und das Produkt kümmert. Dies rentiert erst ab einer gewissen Bedeutung, Komplexität und der Existenz mehrerer (gleichberechtigter) Partner. Eine Roadmap dient dazu, gegenüber einer breiteren Öffentlichkeit und potenziellen weiteren Benutzern der Software zu zeigen, was geplant ist.

[11] Es ist u.U. gerade bei Projekten, die von Anfang an mehrere staatliche Akteure umfassen auch möglich, auf bestehende Foundations (also Stiftungen) zuzugehen und zu schauen, ob unter ihrem Dach die Projekte und Governance geführt werden können, analog zu https://finos.org bzw. https://osr.finos.org oder https://www.eclipse.org/collaborations/ bzw. https://iot.eclipse.org bzw. https://outreach.eclipse.foundation/open-regulatory-compliance.

Die Bundesbehörde entscheidet allein, über den Anforderungsverantwortlichen, was wann in die Software aufgenommen wird.

Für mögliche andere Nutzende der Applikation bedeutet dies aber, dass sie fast gezwungen sind, eine eigene Version (fork) der Applikation zu verwenden.

Ansonsten haben sie kaum die Möglichkeit, die für sie wichtigen Anpassungen und Erweiterungen umzusetzen. Dies spricht nicht gegen eine Veröffentlichung als Open Source, die Werkzeuge zur Code-Verwaltung[12] bieten weitgehende Unterstützung für solche Szenarien; Anpassungen und Fehlerkorrekturen können selektiv in beide Richtungen übernommen werden.

Eine Teilung der Kosten ist bei diesem Modell typischerweise nur schwer möglich.

### 5.1.2 Gemeinsam (Bund im Lead oder gleichwertige Partner)

Die Entscheide betreffend Roadmap und Prioritäten sollen mit den anderen Mitgliedern der Community abgestimmt und gemeinsam getroffen werden. Der Bund bestimmt als Mitglied der Community mit, ist aber nicht federführende Organisation.

Damit eine gemeinsame Beschlussfassung funktionieren kann, müssen vorgängig Strukturen und Regeln bestimmt werden. Dabei wird auch festgelegt, wie die Kosten für die Umsetzung der gemeinsam beschlossenen Anpassungen untereinander aufgeteilt werden.

Mit diesem Modell entsteht eine einzige Version der Applikation, die dann von allen Mitgliedern genutzt wird. Dadurch sind die Gesamtkosten verglichen mit den anderen Varianten deutlich tiefer. Die Flexibilität der einzelnen Nutzenden ist aber ebenfalls geringer, sie müssen ihre Anpassungswünsche vorgängig mit allen andern abstimmen; die Entscheidungsfindung ist langsamer und aufwändiger.

Dieses Modell eignet sich am besten für Applikationen, die gemeinsam mit klar definierten anderen Organisationen (beispielsweise Kantone) weiterentwickelt werden sollen.

### 5.1.3 Offene Organisation

In diesem Modell werden die Prozesse und Strukturen nur lose definiert. Der Bund behält zwar formell die Kontrolle, ist aber offen für Beiträge und Anpassungen.

Es wird keine oder nur eine sehr grobe Roadmap definiert, der Konsens wird informell und in Diskussionen direkt auf der Publikationsplattform hergestellt. Die Community selbst ist eher dynamisch, Mitglieder können eine Zeit lang sehr aktiv sein und sich dann wieder zurückziehen.

Dieses Modell eignet sich auch für eher kleinere Applikationen, die bereits produktiv genutzt werden und «fertig» sind.
Auch für technisch orientierte Applikationen respektive Komponenten, die potenziell eine breitere, in der vornherein nicht genau definierbaren Zielgruppe ansprechen, ist dies meist die beste Form.

### 5.1.4 Dritte

In diesem Modell werden die Prozesse und Strukturen von einer dritten Partei definiert und die Bundesbehörde ist als Mitglied in der Community dabei und übernimmt keine führende Funktion. Dies kann geschehen sein, weil z.B. das Projekt von einem anderen Staat oder Kanton definiert wurde. Oder allenfalls ist der Bund selbst an der Fortführung nicht interessiert und gibt diese Aufgabe ab.

[12] Beispielsweise „GitHub“

## 5.2 Einbindung der Lieferanten

Bei der Einbindung der Lieferanten gibt es grundsätzlich folgende Möglichkeiten:

a) **Lieferant als Auftragnehmer**: Er wird grundsätzlich als mittelfristig ersetzbar betrachtet. Der Lieferant soll die Applikation basierend auf Aufträgen des Produktmanagements (oder der Community) kosteneffizient und in hoher Qualität weiterentwickeln, hat aber höchstens beratend Einfluss auf die Roadmap und die Priorisierung.
b) **Langfristige Partner mit Mitbestimmung**: Sie sollen bei der Entwicklung aktiv mitbestimmen können. Dadurch wird die Rolle gestärkt und sie sind potenziell bereit, auch selbst in die Applikation zu investieren. Typischerweise vertreiben in diesem Modell die Lieferanten dann die Software und suchen aktiv nach neuer Kundschaft.
c) **Unabhängig:** Die Lieferanten definieren selbst, ob und wie sie im Projekt mitarbeiten wollen. Dies kann z.B. auftreten, wenn mehrere Lieferanten basierend auf der Software selbst Geschäfte generieren wollen. Liberale Lizenzen können dies u.U. fördern. Auch wenn die Strategie zwischen Bundesbehörde und Lieferanten divergiert, kann es dazu kommen.
   In diesem Szenario ist es wahrscheinlich, dass der Lieferant einen Fork erstellen wird.
d) **Lead**: Wenn der Lieferant das Produktmanagement übernimmt, dann ist er im Lead. Die Entwicklung wird dann durch ihn bestimmt. Andererseits kann das auch dazu führen, dass die Verwaltung sehr wenig Verantwortung übernehmen muss. Bei einer Minimalfreigabe ist es dem Lieferanten möglich, diese Rolle von sich aus zu übernehmen.

Nimmt der Lieferant eine starke Rolle ein, bringt dies der Verwaltung Vorteile: Durch die aktive Vermarktung der Applikation besteht die Chance, dass die Community schneller und stärker wächst und somit die Kosten pro Mitglied schneller sinken.

Wenn ein Modell mit einer starken Mitbestimmung der Lieferanten gewählt wird, muss darauf geachtet werden, dass dadurch keine Abweichung zu beschaffungsrechtlich vorgeschriebenen Regelungen entsteht. Es empfiehlt sich meist, mindestens zwei Lieferanten mit einzubinden, damit keine zu starke Abhängigkeit zu einem einzelnen Lieferanten entsteht. Auch sollte die Möglichkeit für den Beitritt weiterer Lieferanten zur Community bestehen.

## 5.3 Verteilung der Kosten

Zur Aufteilung der Kosten für Wartung, Weiterentwicklung und neue Funktionen gibt es grundsätzlich folgende Varianten:

a) **Organisationen innerhalb des Bundes***:* Gerade beim Aufbau von Ökosystemen oder wenn die Bundesbehörde die Kontrolle ganz behalten wird, muss die Bundesbehörde auch die Kosten übernehmen.
b) **Jeder seine**: Jeder bezahlt für die von ihm gewünschten Changes selbst.
   Allenfalls wird auf einer Fall-zu-Fall-Basis entschieden, gewisse Erweiterungen gemeinsam zu bezahlen.
   Für wiederkehrende Kosten gibt es zusätzlich einen Verteilschlüssel oder wird nach Bedarf abgerechnet.
c) **Verursacherprinzip**: Diejenigen, die Dienstleistungen und Features benötigen, zahlen dafür. Allenfalls kommen standardisierte Stundenansätze zur Anwendung.
d) **Verteilschlüssel**: Die Kosten werden nach einem definierten Schlüssel innerhalb der Community aufgeteilt. Nur für spezifische Erweiterungen wird davon abgewichen. Diese sollen direkt von der betreffenden Nutzenden bezahlt werden.

Der Kostenteiler kann zum Beispiel die Grösse des betreffenden Kantons oder die Anzahl der Nutzerinnen und Nutzer sein.

e) **Contribution**: Wenn die Bundesbehörde nur personelle Leistung an einem bestehenden Projekt einbringt.

Grundsätzlich macht die Variante b (Kostenteiler) meist nur Sinn, wenn auch ein gemeinsames Produktmanagement wie bei Variante b (Gemeinsam) besteht. Nur wer mitbestimmen kann, wird auch bereit sein, die Folgekosten der Entscheide zu tragen.

## 5.4 Support

Innerhalb der Kostenverteilung und beim Aufwand ist es wichtig zu klären, wer welchen Support leistet. Gemäss Art. 9 EMBAG ist eine reine Minimalfreigabe möglich. Diese Variante liefert u.U. nicht den gewünschten Effekt und baut an sich keine Community auf, in der die Bundesbehörde Einfluss hat.

a) **Minimalfreigabe***:* Kein Support (d.h. die Freigabe erfolgt generell ohne organisierten Support und Beteiligung der Bundesbehörde).
b) **Support durch Bund***:* Die Bundesbehörde (bzw. deren Leistungserbringer) stellt definierten Support zur Verfügung.
c) **Support durch Dritte***:* Die Bundesbehörde, bzw. die Community definiert eine Supportorganisation.

Die zu erzielende Wirkung z.B. auf das Ökosystem Schweiz oder eine Anschubfinanzierung einer Community können dazu führen, dass der Bund Support leistet. Das Level des Supports muss natürlich definiert werden. Grundsätzlich kann auch Entgelt verlangt werden. Aufgrund der Mechanismen des Bundes sind dann eher Leistungserbringer oder Lieferanten in der Lage, kommerziellen Support zu bieten.

Ämter dürfen gemäss EMBAG bezahlten Support anbieten (allenfalls auch über Leistungserbringer und Dritte). Dazu müssen die die entsprechenden Gebühren auf ihrer Webseite publizieren, da die notwendige gesetzliche Grundlage vorhanden ist. Die Anbindung ans Zahlungssystem kann mit dem EFD besprochen werden.

Beispiele für Support und Gebühren finden sich in Anhang G.

## 5.5 Entwicklung

Die Grundmethodik der Entwicklung kann auch auf die Art der Community einen Einfluss haben.

a) **Intern***:* Die Verwendung interner Repositories führt dazu, dass Dritte nicht direkt mitarbeiten können. Je strikter die Abtrennung erfolgt, desto mehr Integrationsaufwand muss auch der Bund leisten, wenn er Erweiterungen Dritter bei sich einbauen will. Auch die Freigabe ist aufwändiger.
b) **Dritte**: Entweder arbeitet ein Lieferant bei sich oder das bereits durch Dritte gestartete/geführte Produkt wird nach definierten Regeln entwickelt.
c) **Offen**: Die Softwareentwicklung findet direkt in einem öffentlichen Repository statt. Die Freigabe nach EMBAG ist damit weitgehend gewährleistet. Die Aufwände sind bei der Entwicklung zu leisten. In diesem Szenario kann auch früher (und einfacher) mit Dritten zusammengearbeitet werden.

# 6 Detaillierung Inhalte für das Community-Konzept

Dieses Kapitel gibt Hilfestellungen zum Schreiben des Community-Konzepts basierend auf den getroffenen Grundschatzentscheiden (gemäss Kapitel 5), wo je nach getroffenen Grundsatzentscheiden unterschiedliche Inhalte sinnvoll sind.

| Produkt-management | Lieferant als | Verteilung Kosten | Vorschlag |
|---|---|---|---|
| Offen | Partner | Bund | Inhalte oder Anregungen, wenn die Grundsatzentscheide lauten:<br>Produktmanagement: c) Offene Organisation<br>Einbindung Lieferant: b) Lieferant als Partner<br>Verteilung der Kosten: b) Jeder seine<br>*Der obenstehende Text kann in die Vorlage kopiert und angepasst werden.* |
| Gemeinsam | - | Jeder seine | Inhalte oder Anregungen, wenn die Grundsatzentscheide lauten:<br>Produktmanagement: b) Gemeinsame Erarbeitung<br>Einbindung Lieferant: a) oder b) (nicht relevant)<br>Verteilung der Kosten: c) Verursacherprinzip oder d) Kostenteiler |
| **Gemein-sam** | - | Verur-sacher | |

Das Community-Konzept soll dabei ein dynamisches Dokument sein, das in Absprache mit der Community angepasst werden kann. Das Konzept soll die aktuelle Situation abbilden und zukünftige Entwicklungsoptionen vorschlagen. Wenn dann später mehr Mitglieder der Community beitreten, dann können die Organisation und Prozesse weiter ausgearbeitet und komplexere Mechanismen eingeführt werden. Die nachfolgenden Unterkapitel richten sich direkt nach der vorgeschlagenen Struktur des Community-Konzepts (siehe Kapitel 4).

Wenn das Produktmanagement bei Dritten liegt, so werden sie den die Konzepte für die Community definieren.

## 6.1 Zentrale Angaben

Die folgenden zentralen Angaben sollten für jede Community erstellt und publiziert werden, so dass sich potenzielle Interessenten für die Software und die Community melden können:

- Name der Software
- Repository
- Owner
- Verantwortliche Organisationseinheit/Amt
- Kontaktadresse
- Verwendete Lizenz
- Support-Level
- Bestehendes Projekt

Diese Angaben lassen sich alle mit publiccode.yml darstellen[13].

## 6.2 Zielsetzung

Die Zielsetzung soll einerseits enthalten, was der eigentliche Zweck bzw. die «Vision» der Applikation ist. Diese Vision soll auch in die Datei «**README.md**» im Repository einfliessen (vgl. *«Em002-2.2 Checkliste Analyse und Aufbereitung»*).

Andererseits soll beschrieben werden, was aus Sicht der Community angestrebt wird:

- Wer soll beitreten?
- Wer sind mögliche Interessentinnen und Interessenten?
- Soll aktiv nach neuen Mitgliedern gesucht werden?
- Welche Hauptziele werden mit der Veröffentlichung als Open Source angestrebt?

## 6.3 Organisation und Governance

Grundsätzlich wird angestrebt, dass bei Neuentwicklungen der Bund der Besitzer der Stammrechte am Quellcode ist.

Die Art des Projektmanagements hängt davon ab, welche Organisation federführend ist (SAFe, HERMES) im Falle des Bundes. Bei der Organisationsform sollte immer auf minimalen Overhead geachtet werden (bestehend vor neu, einfache Gesellschaft vor Verein).

[13] https://yml.publiccode.tools/

| Produkt-management | Lieferant als | Entwicklung | Vorschlag |
|---|---|---|---|
| **Bund** | - | Intern | Da der Bund selbst direkt die Kontrolle behält, ist kein Aufbau zusätzlicher Organisationen notwendig.<br>Die publizierende Stelle (z. B. ein Amt) behält das Eigentum und übernimmt alle Koordinationsaufgaben. Das Kapitel «Organisation» kann in diesem Fall sehr kurzgehalten werden. |
| **Gemeinsam** | - | Intern | Es empfiehlt sich für diesen Fall meistens die Organisation als **einfache Gesellschaft**.<br>Damit die Roadmap und die Anforderungen gemeinsam erarbeitet und abgestimmt werden können, empfiehlt sich die Schaffung zweier Gruppen, in denen pro Community-Mitglied je eine Teilnehmerin oder ein Teilnehmer dabei ist:<br>• Management Board: Definiert die Strategie und die Priorisierung.<br>• Fachgruppe: Erarbeitet auf Expertenebene die Anforderungen und konkreten Inhalte. Je nach Umfang kann auch pro Thema eine eigene Fachgruppe eingesetzt werden.<br>Eigentümerin oder Eigentümer des Codes ist das Amt, die den Code publiziert hat.<br>Wenn die Strukturen komplexer werden oder mehr als fünf Anwenderinnen oder Anwender bzw. weitere Mitglieder beteiligt sind, kann geprüft werden, ob sich die Community besser als Verein organisieren sollte. Vergleichen Sie hierzu die Ausführungen in der Variante direkt unterhalb. |
| **Gemeinsam** | - | Dritte | Es empfiehlt sich eine Organisation als **Verein**. Entweder wird spezifisch für die Applikation ein eigener Verein gegründet oder aber die Applikation wird an einen bestehenden Verein übergeben. Ein Verein deshalb, weil sonst die Aufteilung der Kosten aufwändig wird und wahrscheinlich im Rahmen der Koordinationsaufgaben genug Arbeit anfällt, um eine nebenamtliche Geschäftsführerin oder einen Geschäftsführer für den Verein zu beschäftigen.<br>Dem Verein werden dann die Rechte am Code übertragen, er übernimmt das Community-Management und die Bestellung bei den Lieferanten. |

| | | | |
|---|---|---|---|
| **Gemeinsam** | Auftragnehmer | Intern | Es empfiehlt sich eine Organisation als **Verein**. Entweder wird spezifisch für die Applikation ein eigener Verein gegründet oder aber die Applikation wird an einen bestehenden Verein übergeben. Ein Verein deshalb, weil sonst die Aufteilung der Kosten aufwändig wird und wahrscheinlich im Rahmen der Koordinationsaufgaben genug Arbeit anfällt, um eine nebenamtliche Geschäftsführerin oder einen Geschäftsführer für den Verein zu beschäftigen.<br>Dem Verein werden dann die Rechte am Code übertragen, er übernimmt das Community-Management und die Bestellung bei den Lieferanten.<br>Hier sind die Lieferanten keine Mitglieder des Vereins, sondern nur Auftragnehmer. Die Geschäftsführerin oder der Geschäftsführer soll nicht von einem Lieferanten gestellt werden. |
| **Gemeinsam** | Auftragnehmer | Dritte | Es empfiehlt sich eine Organisation als **Verein.** Entweder wird spezifisch für die Applikation ein eigener Verein gegründet oder aber die Applikation wird an einen bestehenden Verein übergeben. Ein Verein deshalb, weil sonst die Aufteilung der Kosten aufwändig wird und wahrscheinlich im Rahmen der Koordinationsaufgaben genug Arbeit anfällt, um eine nebenamtliche Geschäftsführerin oder einen Geschäftsführer für den Verein zu beschäftigen.<br>Dem Verein werden dann die Rechte am Code übertragen, er übernimmt das Community-Management und die Bestellung bei den Lieferanten.<br>Die Lieferanten sind Mitglieder des Vereins, der Geschäftsführer kann auch von einem Lieferanten gestellt werden. |
| **Offen** | - | Intern | Es braucht **keine grosse Beschreibung einer Organisation**, da diese ja bewusst offen gestaltet werden soll. Es muss aber geregelt werden, wer Anfragen von aussen entgegennimmt, und diese bearbeitet (Zuständigkeit).<br>Ebenfalls soll bestimmt werden, ob und inwiefern externe Personen mit eingebunden werden können:<br>• Keine direkte Einbindung: Externe können nur Vorschläge und Beiträge (als Pull Request) einbringen.<br>• Einbindung als Beitragende: Externe, die häufig und gute Beiträge leisten, werden direkt mit eingebunden.<br>• Übergabe an Externe möglich: Falls nach der Veröffentlichung Externe mehr zur Weiterentwicklung beitragen als das jeweilige Amt, dann kann die Applikation auch an diese übertragen werden.<br>Die publizierende Stelle behält das Eigentum.<br>Diese Variante entspricht der Organisation «klassischer» Open-Source-Projekte, siehe zum Beispiel https://opensource.guide/leadership-and-governance/ für weiterführende Informationen. |

| | | | |
|---|---|---|---|
| **Offen** | Auftragnehmer | Intern | Es braucht **keine grosse Beschreibung einer Organisation**, da diese ja bewusst offen gestaltet werden soll. Es muss aber geregelt werden, wer Anfragen von aussen entgegennimmt, und diese bearbeitet (Zuständigkeit).<br>Ebenfalls soll bestimmt werden, ob und inwiefern externe Personen mit eingebunden werden können:<br>• Keine direkte Einbindung: Externe können nur Vorschläge und Beiträge (als Pull Request) einbringen.<br>• Einbindung als Beitragende: Externe, die häufig und gute Beiträge leisten, werden direkt mit eingebunden.<br>• Übergabe an Externe möglich: Falls nach der Veröffentlichung Externe mehr zur Weiterentwicklung beitragen als das jeweilige Amt, dann kann die Applikation auch an diese übertragen werden.<br>Die publizierende Stelle behält das Eigentum.<br>Diese Variante entspricht der Organisation «klassischer» Open-Source-Projekte, siehe zum Beispiel https://opensource.guide/leadership-and-governance/ für weiterführende Informationen.<br>Der Lieferant sollte nur rein technische Koordinationsaufgaben (Code-Review, Beurteilung) übernehmen. |
| **Dritte** | Partner | Intern | Es braucht **keine grosse Beschreibung einer Organisation**, da diese ja bewusst offen gestaltet werden soll. Es muss aber geregelt werden, wer Anfragen von aussen entgegennimmt, und diese bearbeitet (Zuständigkeit).<br>Ebenfalls soll bestimmt werden, ob und inwiefern externe Personen mit eingebunden werden können:<br>• Keine direkte Einbindung: Externe können nur Vorschläge und Beiträge (als Pull Request) einbringen.<br>• Einbindung als Beitragende: Externe, die häufig und gute Beiträge leisten, werden direkt mit eingebunden.<br>• Übergabe an Externe möglich: Falls nach der Veröffentlichung Externe mehr zur Weiterentwicklung beitragen als das jeweilige Amt, dann kann die Applikation auch an diese übertragen werden.<br>Die publizierende Stelle behält das Eigentum.<br>Diese Variante entspricht der Organisation «klassischer» Open-Source-Projekte, siehe zum Beispiel https://opensource.guide/leadership-and-governance/ für weiterführende Informationen.<br>Der Lieferant kann die Koordinationsaufgaben übernehmen. |

Bezüglich Governance kann auch *[BITKOM2023]* Kapitel 4.1, *[IzCab2023]* oder https://github.com/todogroup/ospo-career-path/blob/main/OSPO-101/module7/README.md#governance-models konsultiert werden.

Beispielhafte Statuten finden sich in Anhang E.

Achtung: Da die Freigabe gemäss Art. 9 EMBAG garantiert sein muss, muss die Bundesverwaltung sicherstellen, dass die Publikation nicht plötzlich zurückgenommen werden kann (siehe die «Wegleitung Open-Source in der Beschaffung – Software-Einkauf unter Berücksichtigung von EMBAG Art. 9» *[BBL-WL]*, Absatz V.2).

## 6.4 Roadmap und Change Prozess

| Produkt-management | Lieferant als | Entwicklung | Vorschlag |
|---|---|---|---|
| **Bund** | Auftrag-nehmer | - | Die Bundesbehörde definiert die Roadmap und entscheidet darüber, welche Changes umgesetzt werden. Es kommen dabei die Standardprozesse des Amtes XY zum Einsatz. |
| **Bund** | Partner | - | Die Bundesbehörde definiert unter Einbezug der Firma XY die Roadmap. Sie entscheidet, welche Changes umgesetzt werden. Es kommen dabei die Standardprozesse des Amtes XY zum Einsatz. |
| **Gemein-sam** | - | Intern | Der Prozess zur Erarbeitung der Roadmap und zur Genehmigung von Changes muss von den Mitgliedern der Gesellschaft respektive des Vereins bestimmt werden. Je nach Mitglieder-Struktur (Anzahl, Grössenverhältnis, etc.) sind andere Prozesse sinnvoll.<br>Die Prozesse müssen aber Massnahmen zur Konfliktlösung und Mehrheitsfindung beinhalten. |
| **Gemeinsam** | - | Dritte | Der Prozess zur Erarbeitung der Roadmap und zur Genehmigung von Changes muss von den Mitgliedern der Gesellschaft respektive des Vereins bestimmt werden. Je nach Mitglieder-Struktur (Anzahl, Grössenverhältnis, etc.) sind andere Prozesse sinnvoll.<br>Die Prozesse müssen aber Massnahmen zur Konfliktlösung und Mehrheitsfindung beinhalten<br>Mit dem Zusatz, dass auch ein Verteilschlüssel für die Kosten festgelegt werden muss. Es soll auch berücksichtigt werden, wie mit der Kostenbeteiligung an von den betreffenden Mitgliedern nicht gewünschten Anpassungen umgegangen werden soll. Gerade bei Vereinen mit ungleich grossen Mitgliedern kann es zu Situationen kommen, in denen ein grosses Mitglied (z. B. die Bundesbehörde) einen wesentlichen Teil der Kosten für eine Erweiterung trägt, ohne dass es einen Nutzen aus dieser zieht. |

| | | | |
|---|---|---|---|
| **Offen** | - | - | Beispiel, passt nicht für alle Situationen<br>Auf die Erstellung einer Roadmap wird verzichtet, die Applikation erfüllt derzeit die Bedürfnisse.<br>Über Changes wird in einer offenen Diskussion entschieden und ein Konsens gesucht. Der Stichentscheid liegt beim Eigentümer des Codes (Schweizerische Eidgenossenschaft). |

## 6.5 Entwicklungsprozess

Wenn Open Development zum Tragen kommt, so ist dies hier zu definieren. Aufgabenverteilung und Namensnennung der zentralen Stellen im Community-Konzept sind wichtig. Auch wie die Sicherung des Inhalts des Repositories sichergestellt wird.

| Produkt-maagement | Lieferant als … | Entwicklung | Vorschlag |
|---|---|---|---|
| **Bund** | - | - | Die Committer-Rechte (Schreibzugriff auf den Code) liegen bei den vom Bund gewählten Personen/Stellen/Lieferanten. Nach Vertragsablauf werden die betreffenden Rechte wieder entzogen.<br>Die Leistungserbringer und Lieferanten sind dafür zuständig, für externe Beiträge, die vom Bund fachlich akzeptiert wurden, den technischen Code-Review-Prozess durchzuführen. Sie haben die Verantwortung für die technische Qualität. Die Lieferanten werden für diese Arbeit vom Bund entschädigt. |
| **Gemeinsam** | Auftragnehmer | Intern | Grundsätzlich erhalten alle von einem Mitglied der Community beauftragten Stellen Schreibzugriff auf den Code (Committer-Rechte). Nach Vertragsablauf werden die betreffenden Rechte wieder entzogen.<br>Wo geänderter Code Kernbereiche der Applikation betrifft, müssen alle Änderungen von einer von der Community dafür speziell beauftragten Stelle reviewt werden. Diese Stelle hat die Verantwortung für die technische Qualität der Applikation. Sie ist auch dafür zuständig, Änderungen zu reviewen, die von der Community fachlich akzeptiert wurden. |

| | | | |
|---|---|---|---|
| **Gemeinsam** | Partner | Dritte | Die Committer-Rechte (Schreibzugriff auf den Code) liegen bei den Mitgliedern der Community sind. Diese Personen organisieren sich selbst und tragen gemeinsam die Verantwortung für die Qualität des Codes.<br>Wenn ein Community-Mitglied einen externen Lieferanten mit einer Anpassung oder Erweiterung beauftragt, dann erfolgt ein Review der Anpassung durch ein Mitglied der Community. Er wird dafür von der Auftraggeberin oder vom Auftraggeber entschädigt.<br>Änderungen am Kern der Applikation müssen von einem zweiten Mitglied der Community ist, reviewt werden.<br>Die Lieferanten, die Community-Mitglieder sind, sind ebenfalls für den Review von externen Beiträgen und neuem technischem Code zuständig, die vom Verein fachlich akzeptiert wurden. |
| **Offen** | - | - | Die Committer-Rechte liegen initial bei den Entwicklerinnen und Entwicklern bzw. ihren Arbeitgebern. Es werden Committer-Rechte an alle Entwicklerinnen und Entwickler vergeben, die während mehreren Monaten aktiv zum Projekt beitragen und eine entsprechende Sozialkompetenz aufweisen.<br>Die einzelnen Entwicklerinnen und Entwickler behalten grundsätzlich ihre Committer-Rechte, auch wenn sie ihren Arbeitgeber wechseln oder der Bund einen anderen Lieferanten wählt. Committer-Rechte verfallen nur, wenn sie freiwillig abgegeben werden oder die Mehrheit der anderen Committer den Entzug beschliesst.<br>Der Bund hat das Recht, einzelne Entwickler der von ihm beauftragten Firmen als Committer zu bestimmen. Er hat nicht das Recht, jemandem Committer-Rechte grundlos zu entziehen.<br>Code-Reviews erfolgen durch mindestens eine Person, die als Committer tätig ist. Committer organisieren sich selbst. Der Bund bekennt sich dazu, die Review-Arbeit zu entschädigen, soweit sie für ihn finanzierbar ist. |

## 6.6 Kostenteilung und Lieferanten

| Produkt-management | Lieferant als … | Entwicklung | Vorschlag |
|---|---|---|---|
| **Bund** | Auftrag-nehmer | Intern | Die Lieferanten werden periodisch mittels WTO-Ausschreibung oder freihändigem Verfahren (je nach Umfang der Wartungsleistungen) gewählt. |
| **Bund** | Partner | - | Wichtig: Vorher prüfen, ob dies im konkreten Fall beschaffungsrechtlich zulässig ist.<br>Zu diesem Punkt werden wir noch Ergänzungen in der Dokumentation vornehmen.<br>Der Bund wählt den Lieferanten XXX als strategischen Partner und strebt eine langfristige Zusammenarbeit an. |
| **Gemeinsam** | Auftragnehmer | Intern | Jedes Mitglied der Community beauftragt in eigener Regie einen oder mehrere Software-Lieferanten zur Umsetzung der von ihm gewünschten Anpassungen und Erweiterungen.<br>Für Anpassungen, die von mehreren Mitgliedern gewünscht werden, wird auf einer Fall-zu-Fall-Basis eine Kostenteilung vereinbart. Das Mitglied mit dem grössten Anteil der Kosten übernimmt die Auswahl und Beauftragung des Lieferanten.<br>(Regelungen zur Wartung der Software) |
| **Gemeinsam** | Auftragnehmer | Dritte | Der Verein wählt die Software-Lieferanten periodisch mittels WTO-Ausschreibung oder freihändigem Verfahren (je nach Umfang der Wartungsleistungen) zentral.<br>Die Kosten werden nach einem Schlüssel auf die Mitglieder umgelegt: (zu definieren: nach Bevölkerung, Benutzer etc.) |

| | | | |
|---|---|---|---|
| **Gemeinsam** | Partner | Intern | Jedes Mitglied der Community beauftragt in eigener Regie einen oder mehrere Software-Lieferanten zur Umsetzung der von ihm gewünschten Anpassungen und Erweiterungen.<br>Für Anpassungen, die von mehreren Mitgliedern gewünscht werden, wird auf einer Fall-zu-Fall-Basis eine Kostenteilung vereinbart. Das Mitglied mit dem grössten Anteil der Kosten übernimmt die Auswahl und Beauftragung des Lieferanten.<br>*(Regelungen zur Wartung der Software)*<br>Allenfalls ergänzt um einen Beitrag der teilnehmenden Lieferanten. Beispiel:<br>*Jede Software-Entwicklungsfirma, die Mitglied der Community ist, verpflichtet sich, pro Jahr mindestens XY Arbeitstage auf eigene Kosten in die Weiterentwicklung, technologische Aktualisierung oder Vermarktung der Applikation zu investieren.* |
| **Gemeinsam** | Partner | Dritte | Der Verein wählt die Software-Lieferanten periodisch mittels WTO-Ausschreibung oder freihändigem Verfahren (je nach Umfang der Wartungsleistungen) zentral.<br>Die Kosten werden nach einem Schlüssel auf die Mitglieder umgelegt: (zu definieren: nach Bevölkerung, Benutzer etc.)<br>Allenfalls ergänzt um den Zusatz:<br>*Jede Software-Entwicklungsfirma, die Mitglied der Community ist, verpflichtet sich, pro Jahr mindestens XY Arbeitstage auf eigene Kosten in die Weiterentwicklung, technologische Aktualisierung oder Vermarktung der Applikation zu investieren.* |
| **Dritte** | Auftragnehmer | - | Die Lieferanten werden periodisch mittels WTO-Ausschreibung oder freihändigem Verfahren (je nach Umfang der Wartungsleistungen) gewählt.<br>Mit dem Zusatz:<br>*Die von den potenziellen Lieferanten beigetragenen Verbesserungen und die Aktivitäten in der Community werden dabei als Faktor in der Bewertung berücksichtigt.*<br>Von Externen gewünschte Changes werden nicht vom Bund finanziert, sondern müssen von diesen selbst beauftragt und bezahlt werden. |

| | | | |
|---|---|---|---|
| **Dritte** | Partner | - | Jedes Mitglied der Community beauftragt in eigener Regie einen oder mehrere Software-Lieferanten zur Umsetzung der von ihm gewünschten Anpassungen und Erweiterungen.<br>Für Anpassungen, die von mehreren Mitgliedern gewünscht werden, wird auf einer Fall-zu-Fall-Basis eine Kostenteilung vereinbart. Das Mitglied mit dem grössten Anteil der Kosten übernimmt die Auswahl und Beauftragung des Lieferanten.<br>(Regelungen zur Wartung der Software)<br>Mit dem Zusatz:<br>*Vom Lieferanten wird eine aktive Mitarbeit in der Community erwartet, auch wo diese Aufwände nicht direkt entschädigt werden.*<br>*Von Externen gewünschte Changes werden nicht vom Bund finanziert, sondern müssen von diesen selbst beauftragt und bezahlt werden. Der Lieferant erklärt sich bereit, diese auf Wunsch des Externen zu fairen Preisen umzusetzen.* |

## 6.7 Vermarktung

| Produkt-management | Lieferant als | Entwicklung | Vorschlag |
|---|---|---|---|
| **Bund** | Auftrag-nehmer | - | Der Bund unternimmt keine Aktivitäten zur Vermarktung der Applikation, veröffentlicht diese aber auf den Plattformen für Open-Source-Software. In den Fachgremien wird auf die Publikation der Applikation hingewiesen.<br>Falls sich interessierte Personen melden, wird geprüft, ob eine gemeinsame Community gegründet werden soll, oder ob mit ihrer Version der Software weitergearbeitet wird. |
| **Gemeinsam** | Partner | - | ergänzend zu oben.<br>Die Lieferanten dürfen die Applikation vermarkten und weitere Interessentinnen und Interessenten suchen. Der Bund darf als Referenz genannt werden. Wir nehmen bewusst in Kauf, dass dabei abweichende Versionen der Software entstehen. |

| | | | |
|---|---|---|---|
| **Gemeinsam** | Auftragnehmer | Intern | Die Applikation wird von den Mitgliedern oder dem Verein vermarktet. |
| **Gemeinsam** | Partner | Intern | Die Applikation wird von den Lieferanten vermarktet. |
| **Gemeinsam** | - | Dritte | Die Applikation wird vom Verein, der einfachen Gesellschaft und deren Mitgliedern vermarktet. |
| **Offen** | - | - | Die Applikation wird von den Lieferanten vermarket oder es wird auf eine explizite Vermarktung verzichtet.<br>Variante ohne explizite Vermarktung:<br>Der Bund unternimmt keine Aktivitäten zur Vermarktung der Applikation, veröffentlichen diese aber auf den Plattformen für Open-Source-Software. In den Fachgremien weisen wir darauf hin, dass wir die Applikation publiziert haben und ihre Mitarbeit erwünscht ist.<br>Interessenten führen wir die Software vor und versuchen sie in unsere Prozesse zur Gestaltung der Roadmap miteinzubinden. |

## 6.8 Umgang mit externen Beiträgen

| Produkt-management | Lieferant als … | Entwicklung | Vorschlag |
|---|---|---|---|
| **Bund** | - | - | Externe Beiträge und Anfragen müssen auch bearbeitet werden, wenn der Bund der alleinige Entscheider über die Software bleibt (ausser bei Minimalfreigabe, aber dann ist ein Fork unausweichlich).<br>Wir schlagen folgenden Inhalt vor:<br>***Umgang mit gemeldeten Fehlern***<br>*Wir versuchen, von externen Personen gemeldete Fehler zu reproduzieren und fragen dabei falls nötig nach Präzisierungen. Falls die Reproduktion gelingt, dann beauftragt der Bund die Behebung der Fehler.*<br>*Kann der Fehler nicht nachvollzogen werden oder ist der Aufwand zur Behebung unverhältnismässig hoch, dann entschuldigen wir uns beim Meldenden und fragen, ob ein Pull Request von ihm zur Behebung des Fehlers möglich wäre.*<br>***Umgang mit Anfragen***<br>*Wir reagieren auf jede eingehende Anfrage. Falls sich eine Anfrage nicht mit vernünftigem Aufwand klären lässt und die Aufwendung weiterer Zeit nicht effizient erscheint, dann entschuldigen wir uns mit dem Hinweis auf beschränkte Mittel und bieten der anfragenden Person an, mit einem Lieferanten Kontakt aufnehmen zu können für eine weitere Beratung.*<br>***Umgang mit Pull Requests***<br>*Wir prüfen jeden Pull Request wohlwollend. Um unnötigen Aufwand seitens des Beitragenden zu vermeiden, weisen wir gleich zu Beginn darauf hin, wenn etwas unserer Roadmap widerspricht oder es fraglich ist, dass wir den Beitrag akzeptieren werden. Der Bund entscheidet allein und nach fachlichen Kriterien, welche Beiträge akzeptiert werden. Immer akzeptiert werden Fehlerkorrekturen, die den Review-Prozess bestanden haben.*<br>***Umgang mit Forks***<br>*Wir unterstützen Forks von unserer Applikation. Wer die Applikation einsetzt, soll selbst einen Fork davon erstellen. Wir schauen mindestens einmal pro Jahr die entstandenen Forks an und übernehmen gute und für uns passende Erweiterungen.* |

| | | | |
|---|---|---|---|
| **Gemeinsam** | - | - | Extern bezieht sich hier auf Beiträge von ausserhalb der definierten Community (Mitglieder im Verein). Abweichungen zu oben:<br>***Umgang mit Forks***<br>*Wir streben an, dass keine (langlebigen) Forks der Applikation entstehen. Stattdessen sollen Interessierte der Community direkt beitreten. Innerhalb der Community versuchen wir, dass alle Mitglieder die Hauptversion verwenden.* |
| **Offen** | - | - | *Für uns ist der Aufbau einer funktionierenden Community und einer offenen Kultur wichtig.*<br>***Umgang mit gemeldeten Fehlern***<br>*Wir versuchen gemeldete Fehler zu reproduzieren und zu korrigieren. Hierbei fordern wir die aktive Mitarbeit der meldenden Personen ein. Falls es für sie möglich ist, sollen sie direkt einen Pull Request mit einer Fehlerkorrektur erstellen.*<br>***Umgang mit Anfragen***<br>*Wir behandeln jede Anfrage innert maximal einer Woche. Anfragen von aktiv Beitragenden behandeln wir prioritär und vertieft. Wir versuchen, andere Beitragende in die Behandlung der Anfragen miteinzubeziehen.*<br>***Umgang mit Pull Requests***<br>*Wir versuchen möglichst viele und gute Pull Requests zu bekommen. In die Reviews und technische sowie fachliche Beurteilung der Pull Requests binden wir alle aktiv Beitragenden ein. Bei kontroversen Beiträgen versuchen wir mit informellen Abstimmungen (Votings) einen Entscheid zu finden.*<br>*Umgang mit Forks*<br>*Wir versuchen, dass Forks möglichst nicht nötig sind, da wir ein aktives und offenes Community-Management betreiben. Falls Forks entstehen, schauen wir diese aktiv an und versuchen die da entstandenen Erweiterungen und Verbesserungen zurückzugeben. Wir fragen in diesem Fall aktiv nach Pull Requests.* |

Generell ist zu beachten, dass Kontributionen vom Unterstützen anderer User bis zur Verantwortung ganzer Teile eines Projekts gehen können (siehe *[BITKOM2023]* Abschnitt 4.2).

## 6.9 Betrieb der Applikation

| Produkt-management | Lieferant als … | Entwicklung | Vorschlag |
|---|---|---|---|
| **Gemeinsam** | - | - | Der Betrieb kann zentral optional zusätzlich durch den zentralen Verein angeboten werden, im Sinne von Software-as-a-Service. Es soll geprüft werden, ob dies gewünscht wird. |
| - | Partner | - | Festhalten, ob der Lieferant die Software auch gleich betreiben soll. |
| - | Lead | - | Festhalten, dass der Lieferant für den Betrieb verantwortlich ist. |

Supportlevel und Supportkontakte sollen in diesem Kapitel auch festgehalten werden.

# 7 Wichtige Hinweise für Community-Verantwortliche

## 7.1 Es braucht Zeit

Der Aufbau von Communities ist etwas, was langfristig angedacht sein sollte. Durchhaltevermögen und Konstanz sind zwei wichtige Eigenschaften, die es für eine erfolgreiche Community braucht.

## 7.2 Weitere Anregungen für Community-Aufbau

Weitere Anregungen für ein gutes Community-Management finden sich in https://opensource.guide/code-of-conduct/ und https://opensource.guide/best-practices/

## 7.3 Kommunikation

Mit der Veröffentlichung von Software und Dokumentationen wird ein weiterer Kommunikationskanal der Bundesverwaltung eröffnet. Wird dies nicht sorgsam und professionell gemacht, kann das öffentliche Ansehen des Bundes darunter leiden.

OSS Communities befinden sich in der Bundesverwaltung bezüglich Kommunikation in einem Spannungsfeld. Einerseits gelten die allgemeinen **Richtlinien zur Kommunikation**, andererseits soll es bei der Community auch um einfache **formelle und informelle Zusammenarbeit** über die Organisationsgrenzen hinweg möglichst ohne formelle Qualitätskontrollen und Freigaben gehen.

Die Verwendung öffentlicher Repositories und unter Umständen öffentlichem Issue-Tracking und Wikis führt dazu, dass die Äusserungen einzelner Projektmitarbeitenden ungefiltert für die Öffentlichkeit sichtbar sind. Dies bedingt, dass in einem solchen Setting die Projektmitarbeitenden zu einem professionellen Auftreten anzuhalten und die Qualitätsstandards für Publikationen einzuhalten sind.

In OSS-Projekten wird dies normalerweise über den **Code of Conduct** geregelt.
Der Verhaltenskodex der Bundesverwaltung[14] selbst ist sehr generell gehalten, doch im Kern ist das Prinzip auch hier anwendbar:
»Staff carry out their duties with a sense of responsibility, integrity and loyalty. In their private lives they also take care not to tarnish the reputation, the credibility and the image of the Confederation»

«Die Angestellten verhalten sich in ihrer beruflichen Tätigkeit verantwortungsbewusst, integer und loyal. Sie achten auch im Privatleben darauf, den guten Ruf, das Ansehen und die Glaubwürdigkeit des Bundes nicht zu beeinträchtigen»

## 7.4 Offene Entwicklung

Wenn die Entwicklung von Software von Beginn weg in einem offenen Repository geplant ist, so erfolgt die Freigabe automatisch. Dazu müssen die entsprechenden Checklisten gemäss der Anleitung *[Em002-2]* bereits zu Beginn ausgefüllt sein und die Entwicklungsprozesse des Leistungserbringers/Lieferanten müssen dies erlauben. Der Vorteil ist, dass die Community gleich von Beginn weg aufgebaut werden kann. Es können auch mehrere Organisationen an der Entwicklung mitarbeiten.

Eine Zwischenform sind geschlossene Repositories, die aber für andere Partner bei einer Kollaboration genutzt werden.

[14] https://www.epa.admin.ch/dam/epa/fr/dokumente/aktuell/medienservice/120_verhaltenskodex_e.pdf.download.pdf/120_verhaltenskodex_e.pdf

## 7.5 Benutzermanagement

Ob die Entwickler und andere Projektmitarbeitende mit eigenem Namen oder anonym an dem Projekt mitarbeiten, legt das Projekt (bzw. die Supportorganisation fest). Falls die Leute nicht anonym arbeiten und bereits über Loggings auf den Plattformen verfügen, so ist es am einfachsten diese (ob geschäftlich oder privat) zu verwenden.

Die Plattform sollte die Aufnahme Dritter grundsätzlich erlauben, da ansonsten keine Entwickler von ausserhalb des Bundes mitarbeiten können.

Beim Austritt aus dem Projekt müssen die entsprechenden Rechte auf dem Projekt gelöscht werden. Das Produktmanagement ist dafür verantwortlich.

## 7.6 Zusammenlegen von Communities

Wenn möglich, soll die Anzahl Communities klein gehalten werden. D.h.

- falls bereits eine Community existiert, wo dann für ähnliche Probleme mit denselben Leuten eine ähnliche Struktur angedacht ist, so sollte nur eine Community verwendet werden.
- Mehrere Projekte, die zusammengehören und dieselbe Zielgruppe ansprechen, können in einer Community zusammengefasst werden.
- Ohne Not sollten nicht grundsätzlich andere Community-Strukturen erstellt werden, als diejenigen, die es schon gibt.

## 7.7 Umgang mit Kontributionen Dritter

Die Kontribution von Software an andere Projekte wird in Abschnitt 3.3 behandelt.
In der umgekehrten Richtung gelten diese sinngemäss.

Wichtig sind folgende Punkte:

- Ein geeignetes Contributor License Agreement muss verwendet werden, damit der Source Code anschliessend dem Bund gehört (siehe auch *«Em002-3 OSS-Lizenzen»* Abschnitt Contributor License Agreement und. Dieses muss vom Eigentümer der Kontribution unterschrieben sein.
- Die Bestätigungen müssen abgelegt werden.
- Die Kontribution muss reviewt werden, bevor sie in die Codebasis integriert wird.
- Eingaben sollten zeitgerecht beantwortet werden (Issues, Anfragen, Pull Requests). Sie sollten möglichst wohlwollend beantwortet werden.
- Alle Kanäle und die Governance sollten im Community-Konzept, auf der Webseite und im Repository aufgeführt sein.

## 7.8 Sicherheitsrelevante Meldungen

Neben normalen Fehlermeldungen muss – wenn dies wegen Art des Projekts notwendig ist – die Möglichkeit gegeben werden, sicherheitsrelevante Fehler vertraulich zu melden, wie dies zum Beispiel bei einem Bug Bounty-Programm vorhanden ist.

Als Beispiel können die Unterlagen von trustbroker.swiss dienen:

- https://github.com/trustbroker-swiss/trustbroker.swiss/blob/main/Security-and-Vulnerability-Disclosure.md

## 7.9 Vermeidung von Forks

Forks sind zwar in der Welt von Open Source etwas Natürliches. Nichtsdestotrotz ist die Reintegration von Quellcode, Features etc. aus Forks nicht trivial. Allenfalls macht es Sinn, mit den Personen, Institutionen, die einen Fork in Erwägung ziehen, das Gespräch zu suchen und zu versuchen, diese beim Hauptprojekt zu behalten. Ein Interessenausgleich ist in diesen Fällen oft notwendig.

## 7.10 Kostenverteilung

Es macht Sinn, generelle Verteilschlüssel zwischen den wichtigsten Partnern zu vereinbaren. Diese können alle paar Jahre oder, wenn sich die Situation grundlegend ändert, angepasst werden. Damit die Arbeit an der Software effizient erfolgt, sollten die grossen Partner tendenziell etwas mehr Anteile als sie eigentlich müssten übernehmen (allenfalls auch die Gesamtkoordination). Die Organisation soll ja arbeiten und sich nicht über die Finanzen streiten. Umfasst die Zusammenarbeit mehrere Projekte (als Beispiel seien die Branchenlösungen im Bereich Normalspurbahnen genannt) so sollte für alle Projekte ein standardisierter Verteilschlüssel verwendet werden.
Aus Sicht der Bundesbehörden ist es eine Verbesserung, wenn die Kosten nicht allein getragen werden müssen.

Mögliche Kriterien für Verteilschlüssel:

- Abschätzung der Verteilung des Nutzens (z.B. zwischen Bundesbehörde und einem Lieferanten, der auf andere Kunden hofft)
- Anzahl Einwohner / Nutzer (z.B. zwischen Ämtern, Gemeinden, Städten)
- Finanzstärke (z.B. Kantone)
- Bruttosozialprodukt (z.B. zwischen Kantonen)
- Wer mehr Kontrolle haben will

Um Diskussionen zu vermeiden, sollte der Verteilschlüssel nicht einfach am jährlichen Budget angehängt sein. Eine Klausel nach der Art Verursacherprinzip kann u.U. schnellere Anpassungen für einen Teil der Partner erlauben, wenn sie bereit sind, extra zu bezahlen.

Der Verteilschlüssel sollte auch für Overhead, Wartung/Support und ein Minimum an Weiterentwicklung gelten. Diese Teile sollten möglichst langfristig gelöst sein.

Es gilt zu beachten, dass, wenn Geld von Dritten kommt, diese auch mehr Einfluss wollen. Damit muss das Community-Management umgehen können.

## 7.11 Ergänzende Dienstleistungen gemäss Art. 9 Abs. 5 und 6 EMBAG

Die Bundesbehörden können (selbst, über die Leistungserbringer oder Lieferanten) ergänzende Dienstleistungen, insbesondere zur Integration, Wartung, Gewährleistung der Informationssicherheit und zum Support erbringen.

Die Einschränkung ist, dass sie den Behördenaufgaben diesen sollen und mit verhältnismässigem Aufwand erbracht werden können.

Daraus folgt, dass die Bundesbehörde nicht die Entwicklungsfirma für Dritte sein kann. Sie flickt Fehler, sie kann sicherheitsrelevante Aspekte behandeln, sie kann etwas Support und Integrationsunterstützung leisten (alles Tätigkeiten, die helfen, eine Community zu bauen). Neuentwicklungen und zusätzliche Features für Dritte sollten sich aber primär auf den normalen Arbeitszweck der Behörde fokussieren. Es kann natürlich sein, dass die Bereitstellung der Software Teil der Aufgabe ist, dann können Entwicklungen jederzeit

gemacht werden. Es ist auch dargelegt, dass die Kernaufgabe der Bundesbehörde nicht die Softwareentwicklung ist.

Das bedeutet, wenn substanziell Features ausschliesslich für Dritte entwickelt werden, so sind diese partnerschaftlich oder über den Lieferanten zu integrieren.

Damit dies keine beschaffungsrechtlichen Probleme verursacht, sollte bei Beschaffungen von Software und Dienstleistungen darauf geachtet werden, dass Features durch die Lieferanten auch an Dritte (u.U. nur staatliche Dritte) erbracht werden können.

Die Festlegung von Entgelten wurde schon im Abschnitt «Support» gestreift. Das EMBAG ist bereits die rechtliche Grundlage für solche Gebühren. Diese müssen auf der Webseite der jeweiligen Bundesbehörde kommuniziert werden. Beispiele sind in Anhang G aufgeführt.

Art. 9 Abs.6 sagt, dass im Normalfall die Privatwirtschaft nicht konkurrenziert werden darf. Das heisst, die Gebühren sollten nicht zu tief angesetzt werden und zumindest kostendeckend sein. Im Zweifelsfall sollten die Stundensätze daher eher zu hoch angesetzt werden.

Generell wird empfohlen mit möglichst 1-3 verschiedenen Standardstundensätzen zu arbeiten.

Wenn ein Departement Ausnahmen zu Abs. 6 definieren will, so darf dadurch die Privatwirtschaft nicht konkurrenziert werden. Unter Umständen ist es dabei am einfachsten, die Gebühren festzulegen und sie bei Widerspruch von privatwirtschaftlichen Firmen, die tatsächlich in der Lage sind, die Dienstleistung anzubieten, die Gebühren zu erhöhen.

## 7.12 Digitale Souveränität

Solange der Bund über kein eigenes Repository verfügt, empfiehlt sich, um die digitale Souveränität sicherzustellen, regelmässig Kopien von den öffentlichen Repositories zu ziehen.

Dabei ist zu beachten, dass nicht nur der Quellcode, sondern auch weitere Informationen wie Issue Tracking, Wiki, Konfigurationen, etc. kopiert werden.

Weiter ist das Usermanagement zu lösen, um die Kontrolle über den veröffentlichten Quellcode zu gewährleisten. Das übergeordnete Ziel ist es, eine Entwicklung unabhängig von der jeweiligen Plattform sicherzustellen und der Community einen Wechsel zu ermöglichen[15].

[15] Siehe auch https://www.bfh.ch/de/aktuell/news/2024/neue-studie-digitale-souveraenitaet/

# Anhang

## A. Änderungen gegenüber Vorversion

- Kapitel 1 «Das Wichtigste in Kürze» neu aufgenommen
- Im Kapitel 3 Zweck Community hinzugefügt
- Kapitel 3.1 bis 3.3 zu Collaboration neu hinzugefügt
- Neues Kapitel 7.7 Umgang mit Contributionen Dritter
- Vermerkt, dass die Lieferanten nicht ganz alleine publizieren sollten (gemäss *[BBL-WL]*, V.2)
- Publicode.yml erwähnt
- Abgleich mit neuem Em002-7
- Weitere redaktionelle Anpassungen und Verbesserungen

## B. Referenzen

Siehe *«Em002 Strategischer Leitfaden Open Source Software in der Bundesverwaltung»*.

## C. Abkürzungen

Siehe *«Em002 Strategischer Leitfaden Open Source Software in der Bundesverwaltung»* und *«Em002-6 FAQ OSS»*.

## D. Beispiele von bestehenden Community-Konzepten

Im Sinne einer Sammlung von Best-Practices sind nachfolgend bereits erarbeitete Community-Konzepte sowie vergleichbare Dokumente aufgeführt. Es handelt sich dabei nicht notwendigerweise nur um Konzepte von Bundesbehörden, sondern auch von anderen Organisationen.

- GERES-Community (http://geres-community.ch)
  Einordnung:
  - Produktmanagement: a) Gemeinsame Erarbeitung
  - Lieferant: eher a) Auftragnehmer
  - Verteilung der Kosten: b) Kostenteiler

  Anmerkung: Das Gemeinderegister GERES ist nicht Open Source, jedoch können viele Aspekte aus der Organisation auch für Open-Source-Applikationen relevant sein. Dokumente: Statuten (öffentlich) und Geschäftsordnung (auf Anfrage).

- iGov Portal

Dieser Abschnitt beinhaltet im Moment Konzepte des Kantons Bern und wird ergänzt, sobald Beispiele des Bundes zur Verfügung stehen.

# E. Beispiele von Kollaborationen

Im Folgenden werden Beispiele aufgeführt, die zeigen, wie Kollaborationen aufgebaut wurden.

## E.1: Kollaboration: ZenDiS[16] in Deutschland

ZenDiS ist eine Organisation mit dem Ziel für die öffentliche Verwaltung (namentlich in Deutschland) mittels Förderung von Open Source Software die Abhängigkeit zu reduzieren.

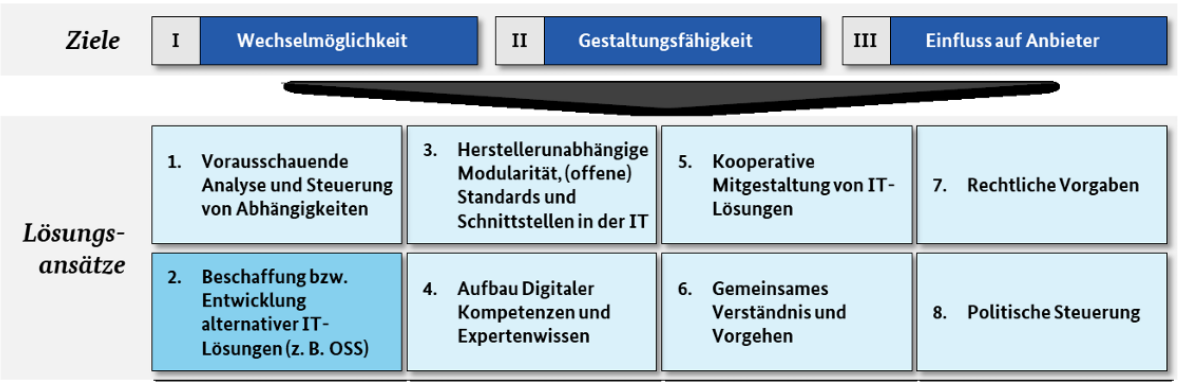


*Abbildung 2: Ziele und Lösungsansätze zur Stärkung der Digitalen Souveränität (ZenDiS)*

ZenDiS arbeitet dazu mit strategischen Partnern zusammen.

Die Zusammenarbeit zwischen der Bundesverwaltung und ZenDiS dürfte daher nicht beliebig eng sein und die Abstimmung nicht verbindlich. Dies kann sich allenfalls ändern. Konkret müsste die Bundesverwaltung für die Schweiz eine eigene Organisation aufbauen und sich informell mit ZenDiS abstimmen. Allenfalls kann dies über eOperations passieren, allerdings wird eOperations nur aktiv, wenn die entsprechende Anforderung kommt und ein Budget bereitgestellt wird. Eine Argumentation zu dieser Parallelorganisation liesse sich über eGovernment bzw. EMBAG finden.
Achtung: ZenDiS unterliegt dem Beschaffungsrecht EU/Deutschland. Es gibt dort erhebliche Unterschiede zur Schweiz. ZenDiS kann daher nicht 1:1 in der Schweiz nachgebildet werden und es kann nicht ganz so einfach mit ZenDiS kooperiert werden.

## E.2: Kollaboration: Open Trip Planner (OTP)

Open Trip Planner[17] ist eine Open Source Software für Reiseplanung. Sie ist als Projekt der Software Freedom Conservacy in New York organisiert. Die einzelnen Entwickler sind von Transportunternehmen und Lieferanten bezahlt. Speziell ist, dass namentlich mit ENTUR[18] ein staatlicher Akteur eine der Hauptrollen übernommen hat. Er entwickelt primär für sich, hat aber auch einen Kollaborationsfokus namentlich für die nordischen Länder. ENTUR ist als Firma organisiert, die dem norwegischen Ministry of Transport and Communication gehört. ENTUR übernimmt auch einen wesentlichen Teil der Koordinationskosten als grösster Akteur. Jeder kann Features eingeben, diese werden in Koordinationsmeetings besprochen und wenn jemand sie bezahlt, so werden sie entwickelt. Die Finanzierung erfolgt dann primär pro Feature durch die interessierte(n) Stellen. Hier zahlt also im Wesentlichen jede Partei ihre Kosten. Auch Supportleistungen müssten in der Schweiz neu ausgeschrieben werden.

[16] Startseite ZenDiS: Gemeinsam. Digital. Souverän.

[17] OpenTripPlanner

[18] https://entur.no/

## E.3 Neue Kollaboration durch Bund: Swiss Trust Broker19

Der Trust Broker ist eine Software des Bundes für EIAM. Er ist als Open Source verfügbar. Die Entwicklung erfolgt durch den Bund. Mehrere Kantone haben beschlossen, ihn entweder direkt als SaaS oder mit eigenen Instanzen einzusetzen. Da er als AGPL zur Verfügung steht, kommen allfällige Erweiterungen wieder dem Bund zugute. Aufgrund der sicherheitskritischen Natur erfolgt die Entwicklung für die Bundesverwaltung ausschliesslich beim Bund. Andere Organisationen können Anforderungen einbringen, diese werden umgesetzt, wenn das Projektteam diese als sinnvoll erachtet.

## E.4 Neue Kollaboration durch Kantone: inosca

Mit inosca[20] haben einige Kantone eine Entwicklungsgemeinschaft geschaffen, die sich mit elektronischen Bewilligungen beschäftigt. Begonnen hat dies mit eBau[21].
eBau beinhaltet mit Caluma[22] auch ein Framework für kollaboratives Editieren. Das ist ein Beispiel für einen Teil aus einer Lösung, die in ein eigenes, breit anwendbares Framework überführt wurde. Dies kann z.B. auch bei anderen Projekten zum Einsatz kommen, selbst wenn dann ein Teil den Ausnahmebedingungen unterliegt. So können immer noch maximal viele andere Stellen vom Code profitieren.

Die Community trifft sich mindestens einmal im Monat (optional zweimal) um sich zu fachlichen Themen auszutauschen und die weitere gemeinsame Roadmap zu planen. Alle Teilnehmenden können Themen einbringen. Wird ein Thema als interessant für die Community eingestuft, melden sich alle Kantone, welche an der Konzeption interessiert sind und organisieren sich danach in einer Arbeitsgruppe. Resultate werden stetig in die Community zurückgeführt.
Die inosca hat als grundlegendes Prinzip: Wer konzipiert finanziert. Sprich alle Kantone, welche in der Arbeitsgruppe teilnehmen teilen sich danach die Kosten des neuen Features. Das Feature wird offeriert und die Kosten werden nach einem vordefinierten Kostenschlüssel der Community aufgeteilt. Ist ein Feature abgeschlossen und eingeführt, steht es per sofort allen anderen Kantonen auch frei zur Verfügung.

Zusätzlich hat die Community entschieden einen kleinen Pauschalbetrag pro Jahr fix einzuplanen, um administrative Aufwände für die Organisation der Community zu decken. Dieses Budget wird jedes Jahr an die Parteien vergeben, welche eine organisierende Rolle einnehmen.

## E.5: SNOWPACK von WSL – Direkte Kontributionen von Entwicklern

SNOWPACK23 ist ein Open Source Projekt, das vom WSL für die Modellierung der Bildung der Schneedecke entwickelt wurde. Es ist ein hochspezialisiertes Modell. Daher wurde die Community in die bestehende Fach-Community integriert. Einige Issues sind als mögliche Contributionen schon vorgemerkt. D.h. hier wird damit gearbeitet, dass jeder seine Kosten selbst trägt und Features beiträgt. In diesem Zusammenhang sind auch Coding Style und Releaseprozess beschrieben.

[19] GitHub - trustbroker-swiss/trustbroker.swiss: Documentation of the trustbroker.swiss service

[20] https://inosca.ch/

[21] https://github.com/inosca/ebau

[22] https://github.com/projectcaluma

[23] https://snow-models.gitlab-pages.wsl.ch/snowpack-web/

# F. Inspirationen für Vereinsstatuten

Die Statuten sollten so einfach wie möglich gehalten werden.
Die folgenden Statuten können als Inspiration für eigene dienen:

- eCH: https://www.ech.ch/sites/default/files/page/STAT_d_DEF_2014-04-10_ech-Statuten.pdf
- Stop Piracy: https://www.stop-piracy.ch/wp-content/uploads/2022/01/Statuten_d_10_09_2021.pdf

Weitere Beispiele von Statuten werden ergänzt, sobald solche zur Verfügung stehen.

# G. Beispiele für Support und Gebühren

- Gebühren statistische Dienstleistungen: https://www.fedlex.admin.ch/eli/cc/2003/326/de
- Gebühren EDÖB: https://www.edoeb.admin.ch/edoeb/de/home/datenschutz/grundlagen/gebuehren.html
- Freie Dienstleistungen IGE für Kurse im Bereich Geistiges Eigentum (nicht Software, aber als Beispiel): https://www.ige.ch/de/uebersicht-dienstleistungen/weiterbildung-und-kurse/preise

Beispiele werden ergänzt, sobald welche zur Verfügung stehen.