Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Bundeskanzlei BK
Digitale Transformation und IKT-Lenkung DTI

14.09.2026

Em002-4 Leitfaden OSS-Community

Version 2.0-f4b576a

Inhaltsverzeichnis

1. Das Wichtigste auf einen Blick 3

2. Einleitung 3

3. Art, Ziel und Zweck einer Community 4

3.1. Sonderfall: Neue Zusammenarbeit für Entwicklung, Wartung und Support von Open-Source-Software 5

3.1.1. Rechtliche Dimension 6

3.1.2. Beschaffungsrechtliche Dimension 6

3.2. Sonderfall: Beitritt zu einer bestehenden Zusammenarbeit 6

3.3. Sonderfall: Codebeiträge an Drittprojekte 6

4. Community-Konzept 7

5. Grundsatzentscheide beim Aufbau einer Community 9

5.1. Produktmanagement 11

5.1.1. Organisationen innerhalb des Bundes: 11

5.1.2. Gemeinsam (Bund hat die Führung oder gleichberechtigte Partner) 12

5.1.3. Offene Organisation 12

5.1.4. Dritte 12

5.2. Einbindung der Lieferanten 12

5.3. Kostenverteilung 13

5.4. Support 13

5.5. Entwicklung 14

6. Detaillierte Inhalte für das Community-Konzept 14

6.1. Eckdaten 15

6.2. Ziele 15

6.3. Organisation und Governance 15

6.4. Roadmap und Änderungsprozess 18

6.5. Entwicklungsprozess 18

6.6. Kostenteilung und Lieferanten 19

6.7. Marketing 21

6.8. Umgang mit externen Beiträgen 21

6.9. Betrieb der Anwendung 22

7. Wichtige Informationen für Community-Managerinnen und -Manager 23

7.1. Es braucht Zeit 23

7.2. Weitere Anregungen zum Community-Aufbau 23

7.3. Kommunikation 23

7.4. Offene Entwicklung 23

7.5. Benutzerverwaltung 24

7.6. Zusammenführung von Communities 24

7.7. Umgang mit Beiträgen Dritter 24

7.8. Sicherheitsrelevante Meldungen 24

7.9. Forks vermeiden 25

7.10. Kostenaufteilung 25

7.11. Ergänzende Leistungen nach Art. 9 Abs. 5 und 6 EMBAG 25

7.12. Digitale Souveränität 26

8. Anhang 26

8.1. Änderungen gegenüber der Vorversion 26

8.2. Referenzen 27

8.3. Abkürzungen 27

8.4. Beispiele bestehender Community-Konzepte 27

8.5. Beispiele von Zusammenarbeiten 27

8.6. Zusammenarbeit: ZenDiS in Deutschland 27

8.7. Zusammenarbeit: Open Trip Planner (OTP) 28

8.8. Neue vom Bund angestossene Zusammenarbeit: Swiss Trust Broker 29

8.9. Neue von den Kantonen angestossene Zusammenarbeit: inosca 29

8.10. SNOWPACK der WSL – direkte Beiträge von Entwickelnden 29

8.11. Anregungen für Vereinsstatuten 29

8.12. Beispiele für Support und Gebühren 30

> **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. Das Wichtigste auf einen Blick

Nachfolgend sind die wichtigsten Punkte des Dokuments aufgeführt.

- Der Aufbau oder Betrieb einer OSS-Community ist nach Art. 9 EMBAG **nicht erforderlich**.
- Eine Community ist relevant, wenn eine Anwendergruppe sinnvoll ist und **Synergien** mit anderen Nutzenden der Software geschaffen werden sollen.
- Eine Community beschränkt sich nicht auf die Software selbst, sondern kann auch die Prozesse im Umfeld und die Standardisierung von Daten umfassen.
- Für den Aufbau einer Community ist stets ein **Initialaufwand** zu leisten. Dieser setzt sich in der Regel fort. Vielfach ist es sinnvoll, dass das federführende Amt auch die Koordinationskosten trägt.
- Communities sollen einen **Zweck** erfüllen. Es braucht Interessierte und Beteiligte.
- Eine Community kann entstehen und anschliessend formalisiert werden.
- Ziel der Community ist es, den **Nutzen für alle** zu erhöhen. Grundsatz: «Man erhält mindestens so viel zurück, wie man einbringt».
- Die Community sollte **so einfach wie möglich strukturiert** sein.
- In einem ersten Schritt sollte die Community anhand der Checkliste Em002-4.1 kurz geprüft werden. Nur wenn sie sinnvoll ist, sollte sie weiterentwickelt werden.
- Die Formalisierung erfolgt über ein **Community-Konzept**. Möglichst viel sollte aus bestehenden Quellen übernommen werden. Grundsatz: «Das Rad nicht neu erfinden».
- Wird eine Zusammenarbeit angestrebt, so ist dies bereits zu Beginn des Projekts zu berücksichtigen, da potenzielle Partner schon bei der Anforderungsanalyse abgeholt werden sollten. Auch rechtlich und beschaffungsrechtlich muss dies von Anfang an gut geplant sein.
- Die Möglichkeiten für den Beitritt zu einer bestehenden Zusammenarbeit sind rechtlich und beschaffungsrechtlich zu prüfen.
- Für Beiträge muss keine Community aufgebaut werden. Verlangt es das Projekt jedoch, so ist das entsprechende **Contributor Licence Agreement (CLA)** zu unterzeichnen oder müssen die Entwickelnden vom Projekt befugt sein, die Software unter einem Developer Certificate of Origin (DCO) einzureichen.

2. Einleitung

Dieser Leitfaden richtet sich an IKT-Fachpersonen, die für eine Anwendung verantwortlich sind und diese als Open Source anbieten oder zu einer solchen Anwendung beitragen wollen.

Dies ist entweder formell oder informell zu organisieren.

Dieses Dokument liefert wichtige Informationen für die Organisation und den Aufbau einer Community[1].] sowie für die offene Entwicklung direkt als Open-Source-Projekt.

Als zweites Element wird zur Förderung der Standardisierung im Rahmen dieses Dokuments ein Verzeichnis bestehender Community-Konzepte geführt. Die Idee ist, dass neue Communities mit minimalem Aufwand auf der Grundlage erfolgreicher Lösungen aufgebaut werden können.

Dieses Dokument enthält keine direkten Empfehlungen für die Organisation von Communities für Softwarebibliotheken (Teile von Software, die Teilfunktionen erfüllen, z. B. PDF-Erzeugung). Normalerweise wird für diese kein eigenes Konzept erarbeitet; stattdessen verwenden sie die einfache Vorlage, die in der Ablagevorlage des Repositorys festgelegt ist.

Das Dokument Em002-01 Praxisleitfaden für Open-Source-Software in der Bundesverwaltung enthält in Kapitel 4 grundlegende Formen der Zusammenarbeit, in Kapitel 8 die verschiedenen Supportmodelle und im Anhang Informationen zum Marktverhalten von Open Source.

3. Art, Ziel und Zweck einer Community

Eine Community kann die folgenden Zwecke verfolgen:

- Erhebung zusätzlicher Anforderungen
- Verbreitung von Wissen über die Anwendung
- Bessere Rückmeldungen von Nutzenden
- Förderung von Übersetzungen, Dokumentation und Schulungen
- Verbreitung der Standardisierung
- Allgemein verbesserte Interaktion mit den Nutzenden
- Ausweitung der Zusammenarbeit auf weitere Beteiligte
- Förderung weiterer Software auf Basis der Software

1. Die Community eines Open-Source-Projekts stellt typischerweise eine Pyramide dar. + Das Fundament dieser Pyramide bilden die Nutzenden der Software, insbesondere die engagierten, die sich aktiv in der Community beteiligen, etwa in Form von Fehlermeldungen und Feature-Wünschen oder durch Beiträge zu Mailinglisten. + Direkt darüber stehen in der Pyramide die Contributors. Dies sind Teile der Community, die eigene Codebeiträge vorschlagen. Sie haben typischerweise keinen Schreibzugriff auf das Repository. Ihre Beiträge werden von Maintainern des Projekts geprüft und nach Erreichen der projekttypischen Qualität ins Repository übernommen. + An der Spitze der Pyramide stehen die Maintainer bzw. Committer eines Projekts. In dieser Rolle trägt eine Entwicklerin oder ein Entwickler eine höhere Verantwortung. Dies zeigt sich beispielsweise in der Möglichkeit, Beiträge anzunehmen. Dieses Recht manifestiert sich häufig über den Schreibzugriff auf das Repository. Auf dieser Ebene findet die Kontrolle über die Software, ihre Qualität und ihren Funktionsumfang statt. Bei komplexeren Projekten kann diese Ebene weiter unterteilt sein. So hat beispielsweise der Linux-Kernel bekanntlich Subsystem-Maintainer auf mehreren Ebenen, und schliesslich übernimmt nur eine einzige Entwicklerpersönlichkeit, Linus Torvalds, die Patches ins Projekt-Repository. Ein weiteres Beispiel sind Projekte der Eclipse Foundation, die typischerweise über einen Project Lead mit zusätzlichen Rechten im Rahmen des Entwicklungsprozesses der Eclipse Foundation verfügen, etwa das Auslösen des Release-Prozesses oder die formelle Wahl von Committern oder Project Leads *[BITKOM2023*

Welche Art von Community für eine Anwendung am besten geeignet ist, hängt stark vom fachlichen Umfeld, den potenziellen Nutzenden und der Strategie der veröffentlichenden Institution ab. Es gibt viele unterschiedliche Wege, Communities aufzubauen. Idealerweise kann sich ein geeignetes Projekt einer bestehenden Community anschliessen. Alternativ kann eine Community mit einem oder mehreren gleichberechtigten Partnern aufgebaut werden.

Wichtig ist, dass eine **Governance-Struktur** festgelegt und die massgebenden Punkte in einem **Community-Konzept** dokumentiert werden (siehe auch *[BITKOM2023]* Kapitel 4.1 und *[IZqCab2023]*).

Zu beachten ist, dass Communities für Software auch mit solchen für Daten, Standardisierung und fachliche Themen kombiniert werden können. Die Community kann dann Prozesse, Standardisierung, Software und Daten umfassen. Bestehende Gefässe wie eOperations, DVS und eCH können ebenfalls genutzt werden, wo dies sinnvoll ist.

Ein wichtiger Aspekt einer funktionierenden Community ist, dass die Beteiligten mehr erhalten, als sie investieren.

Es bestehen die folgenden Konstellationen:

- **Community rund um eine OSS des Bundes**: Governance und Community-Konzept sind zu erstellen. In diesem Zusammenhang ist auch zu regeln, wie Dritte Beiträge leisten können.
- **Neue Zusammenarbeit zur Lösung eines Problems**: Neben Governance und Community-Konzept ist die gesamte rechtliche Struktur der Zusammenarbeit aufzubauen.
- **Beitritt zu einer Zusammenarbeit**: Die Bundesverwaltung muss prüfen, ob und wie sie dies tun kann.
- **Beitrag zu einem Drittprojekt**: Zu prüfen ist die Politik des Projekts für Beiträge (Contributor Licence Agreement, Developer Certificate of Origin usw.).

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

3.1. Sonderfall: Neue Zusammenarbeit für Entwicklung, Wartung und Support von Open-Source-Software

Die Gesamtkosten von Open-Source-Software sinken, wenn sie gemeinsam entwickelt und weiter gepflegt wird. Dies kann ohne förmliche Kooperationsvereinbarungen erfolgen oder mit nicht mehr als einer gemeinsamen Planung.

Wird eine förmlichere Zusammenarbeit angestrebt, so sollte eine Marktabklärung durchgeführt werden, um sicherzustellen, dass die Zusammenarbeit die geeignetste Form der Kooperation ist. Zusammenarbeiten sind kostengünstiger als eine Einzelentwicklung durch jede Partei (siehe Em002-4.1).

Zu beachten ist, dass es bei Zusammenarbeiten sinnvoll ist, die Zusammenarbeit sehr früh in der Projektmethodik HERMES[2] anzustossen – konkret während der Phasen Situationsanalyse und Anforderungen.

Zusammenarbeit ist in der Schweiz nach Artikel 4 EMBAG grundsätzlich zulässig.

Neben allen weiteren Überlegungen haben Zusammenarbeiten sowohl eine rechtliche als auch eine beschaffungsrechtliche Dimension. Derzeit wird jeder Fall einzeln beurteilt. Mit der Zeit werden sich

2. https://www.hermes.admin.ch/de/projektmanagement/methodenuebersicht.html

standardisierte Vorlagen etablieren.

3.1.1. Rechtliche Dimension

Vereinsmitgliedschaften sind machbar, werden aber in der Regel im Einzelfall beurteilt: Wer steht hinter dem Verein, und welche Verpflichtungen werden eingegangen? Eine Delegation von Mitgliedschaften an eOperations oder ähnliche Organisationen wäre möglich.

Solange kein Geld fliesst und keine förmlichen Verpflichtungen eingegangen werden müssen, ist ein Memorandum of Understanding auf Ebene des Bundesamts oder der Bundeskanzlei möglich.

Eine Situation, die den Abschluss eines völkerrechtlichen Vertrags erfordert, sollte vermieden werden.

Es gibt zudem zwei Sonderfälle,[3] in denen eine Zusammenarbeit bereits abgedeckt wäre:

- Aufträge, die im Rahmen einer internationalen Vereinbarung über die gemeinsame Durchführung eines Projekts durch Signatarstaaten vergeben werden.
- Aufträge, die nach den besonderen Verfahren oder Bedingungen einer internationalen Organisation vergeben werden.

3.1.2. Beschaffungsrechtliche Dimension

In vielen Fällen findet die Zusammenarbeit zwischen Organisationen des öffentlichen Sektors statt. In solchen Fällen ist die Beschaffung unproblematisch, sofern die Konstellation «staatsintern» ist (siehe Em002-7 Strategische Aspekte der Beschaffung und Open-Source-Software).

Verträge mit ausländischen Unternehmen ohne Sitz in der Schweiz sind möglich, sofern die Anforderungen des Beschaffungsrechts erfüllt sind.

Die Beschaffung eigener Ressourcen einer Organisation erfolgt im Rahmen eines üblichen Beschaffungsprozesses oder ist bereits abgeschlossen.

3.2. Sonderfall: Beitritt zu einer bestehenden Zusammenarbeit

Dies erfordert sowohl eine rechtliche Grundlage als auch die Einhaltung des Beschaffungsrechts.

Am einfachsten ist die Beteiligung nach dem Grundsatz «jede Partei trägt ihre eigenen Kosten». Damit lassen sich auch Gemeinkosten abdecken, sofern entweder interne Ressourcen oder korrekt ausgeschriebene externe Ressourcen eingesetzt werden.

Eine direkte Beteiligung an ausländischen Zusammenarbeiten ist schwieriger. Ein einfacher Zusammenschluss über ein Memorandum of Understanding ist am einfachsten zu erreichen.

3.3. Sonderfall: Codebeiträge an Drittprojekte

Nimmt die Bundesverwaltung Änderungen am Code eines bestehenden OSS-Projekts vor, so ist es sinnvoll, diese Änderungen direkt upstream in das ursprüngliche Projekt einzubringen. Damit muss die laufende Pflege nicht mehr von der Bundesverwaltung geleistet werden.

Nach Artikel 9 EMBAG sind solche Beiträge sogar erwünscht. Beschaffungsrechtlich ist dies unproblematisch.

Wichtig ist, dass die Rechtsinhaberin den Beitrag leistet. Das heisst, dass die Bundesverwaltung dies stellvertretend für ihre Mitarbeitenden und Auftragnehmer verantwortet. Konkret ist ein allfälliges

3. https://www.eda.admin.ch/deza/de/home/partnerschaften/auftraege-beitraege/auftraege/anforderungen/rechtlich.html

Contributor Licence Agreement des Projekts zu unterzeichnen oder ist dem Pull Request ein Developer Certificate of Origin beizulegen.

Ein **Developer Certificate of Origin**[4] **(DCO)** für die Bundesverwaltung könnte lauten:

> Copyright © 2004, 2006 The Linux Foundation and its contributors.
>
> Everyone is permitted to copy and distribute verbatim copies of this licence 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 authorised to submit it under the open source licence 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 licence and I have the right under that licence 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 licence), 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 licence(s) involved.
>
> *— Developer Certificate of Origin*
> *Version 1.1*

Der Aufbau eines Contributor Licence Agreement wird in Em002-3 Leitfaden OSS-Lizenzierung, Kapitel 8.9, beschrieben.

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

4. Community-Konzept

Der Kern einer Community sind letztlich ihre Struktur und ihre Governance. Diese werden in einem Community-Konzept festgehalten.

Das von der Projekt- oder Applikationsverantwortung zu erstellende (oder bei Dritten in Auftrag gegebene) Community-Konzept sollte dieser Kapitelstruktur folgen. Je nach Art der konkreten Community sind nicht alle Unterkapitel nötig. Idealerweise verweisen viele Kapitel auf bereits standardisierte Dokumente oder nutzen Standardprozesse der entsprechenden Repositories. Das Community-Konzept lässt sich auch auf bestehende, von anderen geführte Projekte anwenden, insbesondere wenn die tatsächliche Governance noch nicht schriftlich dokumentiert ist.

4. https://developercertificate.org/

Mit der Formalisierung der Community werden alle Beteiligten zusammengeführt und das Onboarding neuer Beteiligter wird vereinfacht.

Strukturentwurf eines Community-Konzepts:

1. Übersichtstabelle (Name, Repository, Kontaktadresse, Lizenz, Tiefe des Supports, bestehend/neu)
2. Ziele
    a. der Anwendung
    b. der Community
3. Organisation
    a. Eigentümerin des Codes
    b. Organisationsform / Gremien
    c. Stimmrechte
    d. Projektleitung
    e. Beschaffungsfragen (sofern zutreffend)
    f. Weitere Governance-Aspekte
4. Roadmap und Änderungsprozess
5. Entwicklungsprozess
    a. Offene Entwicklung
    b. Committer-Rechte
    c. Review-Prozess
    d. Aufgabenverteilung und Namensnennungen
    e. Backups
6. Finanzierung der Community
7. Marketing
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 Zuweisung von Rechten/geistigem Eigentum
9. Betrieb der Anwendung

Grundsätzlich ist jede Bundesbehörde für die Erstellung der Community-Konzepte verantwortlich. DTI kann bestehende Muster/Beispiele veröffentlichen und nimmt entsprechende Beispiele zur

Veröffentlichung entgegen. Sinnvoll ist zudem, dass sich die Community-Managerinnen und -Manager untereinander austauschen.

Eine mögliche Quelle für Teile ist auch die Dokumentation der Eclipse Foundation,[5] wobei hier im Sinne von «nur das Nötige tun» ein angemessenes Tailoring vorzunehmen ist.

5. Grundsatzentscheide beim Aufbau einer Community

Die Grundsätze der Community – oder deren Verzicht – werden anhand von Em002-4.1 festgelegt. Die massgebenden Grundsatzfragen finden sich im nachstehenden morphologischen Kasten; jede Dimension wird in den folgenden Kapiteln erläutert.

5. Siehe https://www.eclipse.org/projects/handbook/#starting

Dimension	(a)	(b)	(c)	(d)	(e)
A — Produktmanagement	Bund	Gemeinsame Organisation	Offene Organisation	Dritte	
B — Einbindung der Lieferanten	Auftragnehmer	Partner	Unabhängig	Führung	
C — Kostenverteilung	Bund	Jede und jeder für sich	Verursacherprinzip	Verteilschlüssel	Beitrag
D — Support	Minimale Freigabe	Support durch den Bund	Support durch Dritte		
E — Entwicklung	Intern	Dritte	Offen		

Morphologischer Kasten OSS-Community.

Es empfiehlt sich, bei der Gestaltung der Community **primär auf die nahe Zukunft zu fokussieren**; gibt es beispielsweise noch keine konkreten Interessierten für die Anwendung, so ist es meist wenig sinnvoll, eine Community mit einer einfachen Gesellschaft, einem eigenen Verein und komplexen Prozessen für die Entwicklung der Roadmap zu definieren.[6][7]

5.1. Produktmanagement

Je nach Art der Anwendung, ihrer strategischen Bedeutung und ihrem Stand im Lebenszyklus bestehen verschiedene Varianten zur Abbildung des Produktmanagements. Bei Organisationen innerhalb von Bundesbehörden betrifft dies stets die Applikationsverantwortlichen. Für das Community-Konzept besonders relevant sind die Definition der fachlichen und technologischen Roadmap sowie die Release-Prozesse.

Möglichkeiten:

a. **Organisationen innerhalb des Bundes**: Der Bund will die Kontrolle behalten; er definiert die Roadmap allein.

b. **Gemeinsame Organisation**: Zusammen mit anderen Nutzenden oder Lieferanten.

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:

Ist die Anwendung von hoher strategischer Bedeutung oder ist der Bund darauf angewiesen, seine Änderungen rasch umsetzen zu können, so kann es sinnvoll sein, die Kontrolle zu behalten.

Die Bundesbehörde entscheidet über die Anforderungsverantwortung allein, was wann in die Software einfliesst.

Für potenzielle andere Nutzende der Anwendung bedeutet dies, dass sie praktisch gezwungen sind, eine eigene Version (Fork) der Anwendung zu verwenden.

Andernfalls haben sie kaum Möglichkeiten, für sie allenfalls wichtige Anpassungen und Erweiterungen umzusetzen. Dies spricht nicht gegen eine Veröffentlichung als Open Source: Die Werkzeuge zur Codeverwaltung[8] bieten umfangreiche Unterstützung für solche Szenarien, und Anpassungen sowie Fehlerkorrekturen können in beide Richtungen selektiv übernommen werden.

Eine Kostenteilung ist bei diesem Modell typischerweise schwierig.

---

6. Mit einem Verein (oder einer Stiftung) wird eine eigene juristische Person geschaffen, die sich um die Software und das Produkt kümmert. Dies lohnt sich erst ab einer gewissen Bedeutung, Komplexität und dem Bestehen mehrerer (gleichberechtigter) Partner. Eine Roadmap dient dazu, der breiteren Öffentlichkeit und potenziellen weiteren Nutzenden der Software aufzuzeigen, was geplant ist.
7. Insbesondere bei Projekten, an denen von Anfang an mehrere staatliche Akteure beteiligt sind, kann es auch möglich sein, auf bestehende Stiftungen zuzugehen und zu prüfen, ob die Projekte und die Governance unter deren Dach geführt werden können, analog zu https://finos.org oder https://osr.finos.org oder https://www.eclipse.org/collaborations/ oder https://iot.eclipse.org oder https://outreach.eclipse.foundation/open-regulatory-compliance.
8. z. B. GitHub

5.1.2. Gemeinsam (Bund hat die Führung oder gleichberechtigte Partner)

Entscheide zu Roadmap und Prioritäten sollten mit den übrigen Mitgliedern der Community abgestimmt und gemeinsam gefällt werden. Der Bund beteiligt sich als Mitglied der Community, ist aber nicht die federführende Organisation.

Damit die gemeinsame Entscheidfindung funktioniert, müssen Strukturen und Regeln vorgängig festgelegt werden. Damit wird auch definiert, wie die Kosten für die Umsetzung der gemeinsam beschlossenen Anpassungen untereinander aufgeteilt werden.

Bei diesem Modell entsteht eine einzige Version der Anwendung, die dann von allen Mitgliedern genutzt wird. Dadurch sind die Gesamtkosten im Vergleich zu den anderen Varianten deutlich tiefer. Die Flexibilität der einzelnen Nutzenden ist jedoch geringer, und sie müssen ihre Anpassungswünsche vorgängig mit allen anderen abstimmen. Die Entscheidfindung ist langsamer und aufwendiger.

Dieses Modell eignet sich am besten für Anwendungen, die gemeinsam mit klar definierten anderen Organisationen (z. B. Kantonen) weiterentwickelt werden sollen.

5.1.3. Offene Organisation

Bei diesem Modell sind die Prozesse und Strukturen nur lose definiert. Der Bund behält 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 Veröffentlichungsplattform erreicht. 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 kleinere Anwendungen, die bereits produktiv im Einsatz und «fertig» sind. Es ist zudem meist die beste Form für technisch orientierte Anwendungen oder Komponenten, die potenziell eine breitere, vorgängig nicht genau bestimmbare Zielgruppe ansprechen.

5.1.4. Dritte

Bei diesem Modell werden die Prozesse und Strukturen von Dritten definiert und die Bundesbehörde ist Mitglied der Community, hat aber nicht die Führung. Dies kann daher rühren, dass das Projekt beispielsweise von einem anderen Staat oder Kanton definiert wurde. Oder es kann sein, dass der Bund nicht daran interessiert ist, das Programm selbst weiterzuführen, und diese Aufgabe überträgt.

5.2. Einbindung der Lieferanten

Für die Einbindung von Lieferanten bestehen grundsätzlich die folgenden Möglichkeiten:

a. **Lieferant als Auftragnehmer**: Er gilt mittelfristig grundsätzlich als ersetzbar. Der Lieferant soll die Anwendung auf Basis von Aufträgen des Produktmanagements (oder der Community) kostengünstig und in hoher Qualität entwickeln, hat aber höchstens beratenden Einfluss auf Roadmap und Priorisierung.

b. **Langfristige Partner mit Mitbestimmung**: Sie sollen die Entwicklung aktiv mitbestimmen können. Dies stärkt ihre Rolle und sie sind potenziell bereit, selbst in die Anwendung zu investieren. Typischerweise vertreiben die Lieferanten in diesem Modell die Software und suchen aktiv neue Kundinnen und Kunden.

c. **Unabhängig:** Die Lieferanten legen selbst fest, ob und wie sie am Projekt arbeiten wollen. Dies kann etwa vorkommen, wenn mehrere Lieferanten auf Basis der Software eigenes Geschäft generieren wollen. Liberale Lizenzen können dies potenziell fördern. Es kann auch vorkommen, wenn die Strategie des Bundes von jener des Lieferanten abweicht. In diesem Szenario ist wahrscheinlich, dass der Lieferant einen Fork erstellt.

d. **Führung**: Übernimmt der Lieferant das Produktmanagement, so hat er die Führung. Die Entwicklung wird dann von ihm bestimmt. Andererseits kann dies auch bedeuten, dass die Bundesverwaltung nur sehr wenig Verantwortung übernehmen muss. Bei einer minimalen Freigabe ist es möglich, dass der Lieferant diese Rolle eigenständig übernimmt.

Übernimmt der Lieferant eine starke Rolle, so bringt dies Vorteile für die Bundesverwaltung: Durch die aktive Vermarktung der Anwendung besteht die Chance, dass die Community schneller und stärker wächst und so die Kosten pro Mitglied rascher sinken.

Wird ein Modell mit starker Mitbestimmung der Lieferanten gewählt, so ist darauf zu achten, dass daraus keine Abweichung von beschaffungsrechtlichen Vorgaben resultiert. In der Regel empfiehlt es sich, mindestens zwei Lieferanten einzubeziehen, um keine zu starke Abhängigkeit von einem einzigen Lieferanten zu schaffen. Zudem sollte die Möglichkeit bestehen, dass weitere Lieferanten der Community beitreten.

5.3. Kostenverteilung

Für die Aufteilung der Kosten für Wartung, Weiterentwicklung und neue Funktionen stehen die folgenden Optionen zur Verfügung:

a. **Organisationen innerhalb des Bundes**: Insbesondere beim Aufbau von Ökosystemen oder wenn die Bundesbehörde die volle Kontrolle behalten will, muss die Bundesbehörde auch die Kosten tragen.

b. **Jede und jeder für sich**: Alle bezahlen die von ihnen gewünschten Änderungen selbst. Nötigenfalls wird im Einzelfall entschieden, bestimmte Erweiterungen gemeinsam zu bezahlen. Wiederkehrende Kosten werden nach einem Verteilschlüssel oder nach Bedarf verrechnet.

c. **Verursacherprinzip**: Wer die Leistungen oder Funktionen tatsächlich benötigt, bezahlt dafür. Es können standardisierte Stundensätze angewendet werden.

d. **Verteilschlüssel**: Die Kosten werden innerhalb der Community nach einem definierten Schlüssel aufgeteilt. Davon weichen nur spezifische Erweiterungen ab. Diese sollten direkt von den betroffenen Nutzenden bezahlt werden. Der Kostenteiler kann beispielsweise die Grösse des betroffenen Kantons oder die Zahl der Nutzenden sein. Zum Verteilschlüssel siehe auch das Kapitel «Kostenaufteilung».

e. **Beitrag**: Wenn die Bundesbehörde lediglich Personal in ein bestehendes Projekt einbringt.

Grundsätzlich ist Variante b (Kostenteilung) in der Regel nur dann sinnvoll, wenn auch ein gemeinsames Produktmanagement wie in Variante b (gemeinsam) besteht. Nur wer mitreden kann, wird bereit sein, die Folgekosten der Entscheide zu tragen.

5.4. Support

Im Rahmen von Kostenverteilung und Aufwand ist zu klären, wer welchen Support leistet. Nach Art. 9 EMBAG ist eine reine minimale Freigabe möglich. Diese Variante erzielt unter Umständen nicht die gewünschte Wirkung und baut für sich genommen keine Community auf, in der die Bundesbehörde Einfluss hat.

a. **Minimale Freigabe**: Kein Support (d. h. die Freigabe erfolgt in der Regel ohne organisierten Support und ohne Beteiligung der Bundesbehörde).

b. **Support durch den Bund**: Die Bundesbehörde (oder ihr Leistungserbringer) leistet definierten Support.

c. **Support durch Dritte**: Die Bundesbehörde oder die Community definiert eine Supportorganisation.

Die zu erzielende Wirkung, etwa auf das Schweizer Ökosystem oder eine Anschubfinanzierung einer Community, kann dazu führen, dass der Bund Support leistet. Der Umfang des Supports ist selbstverständlich zu definieren. Grundsätzlich kann auch eine Gebühr erhoben werden. Aufgrund der Mechanismen des Bundes dürften Leistungserbringer oder Lieferanten eher in der Lage sein, kommerziellen Support anzubieten.

Ämter dürfen nach EMBAG kostenpflichtigen Support anbieten (allenfalls auch über Leistungserbringer und Dritte). Dazu müssen sie die entsprechenden Gebühren auf ihrer Website veröffentlichen, da die nötige gesetzliche Grundlage besteht. Die Anbindung an das Zahlsystem kann mit dem EFD besprochen werden.

Beispiele für Support und Gebühren finden sich in Anhang G.

5.5. Entwicklung

Auch die grundsätzliche Methodik der Entwicklung kann Einfluss auf die Art der Community haben.

a. **Intern**: Die Nutzung interner Repositories bedeutet, dass Dritte nicht direkt mitarbeiten können. Je strikter die Trennung, desto mehr Integrationsaufwand hat der Bund zu leisten, wenn er Erweiterungen Dritter übernehmen will. Auch der Freigabeprozess ist aufwendiger.

b. **Dritte**: Entweder arbeitet ein Lieferant eigenständig, oder das bereits von Dritten gestartete bzw. geführte Produkt wird nach definierten Regeln entwickelt.

c. **Offen**: Die Softwareentwicklung erfolgt direkt in einem öffentlichen Repository. Die Freigabe nach EMBAG ist damit weitgehend gewährleistet. Die Aufwände fallen während der Entwicklung an. In diesem Szenario kann die Zusammenarbeit mit Dritten auch früher (und einfacher) erfolgen.

6. Detaillierte Inhalte für das Community-Konzept

Dieses Kapitel gibt Hinweise zum Verfassen des Community-Konzepts auf Basis der getroffenen Grundsatzentscheide (gemäss Kapitel 5).

Produktmanagement / Lieferant als / Entwicklung — Vorschlag

offen / Partner / Bund
Inhalte oder Vorschläge, wenn die Grundsatzentscheide lauten:
Produktmanagement c) Offene Organisation
Einbindung der Lieferanten: b) Lieferant als Partner
Kostenverteilung: b) Jede und jeder für sich
Der obige Text kann in die Vorlage kopiert und angepasst werden.

Gemeinsam / - / Jede und jeder für sich
Inhalte oder Vorschläge, wenn die Grundsatzentscheide lauten:
Produktmanagement b) Gemeinsame Entwicklung
Einbindung der Lieferanten: a) oder b) (nicht relevant)
Kostenverteilung: c) Verursacherprinzip oder d) Kostenteilung

Gemeinsam / - / Verursacher

Das Community-Konzept sollte ein dynamisches Dokument sein, das in Absprache mit der Community angepasst werden kann. Das Konzept sollte die aktuelle Situation abbilden und künftige Entwicklungsoptionen vorschlagen. Treten später weitere Mitglieder der Community bei, so können Organisation und Prozesse weiter ausgearbeitet und komplexere Mechanismen eingeführt werden. Die folgenden Unterkapitel folgen direkt der vorgeschlagenen Struktur des Community-Konzepts (siehe Kapitel 4).

Liegt das Produktmanagement bei Dritten, so definieren diese die Konzepte für die Community.

6.1. Eckdaten

Die folgenden Eckdaten sollten für jede Community erarbeitet und veröffentlicht werden, damit sich potenziell Interessierte für die Software und die Community anmelden können:

- Name der Software
- Repository
- Eigentümerin
- Verantwortliche Organisationseinheit/Amt
- Kontaktadresse
- Verwendete Lizenz
- Supportstufe
- Bestehendes Projekt

Diese Angaben lassen sich alle mit publiccode.yml abbilden.[9]

6.2. Ziele

Das Ziel sollte den eigentlichen Zweck oder die «Vision» der Anwendung enthalten. Diese Vision sollte auch in der Datei **README.md** im Repository aufgeführt werden (siehe Em002-2.2 Checkliste OSS-Analyse und -Vorbereitung).

Gleichzeitig sollte auch beschrieben werden, was die Community anstrebt:

- Wer soll beitreten?
- Wer sind die potenziell Interessierten?
- Sollen neue Mitglieder aktiv gesucht werden?
- Welches sind die Hauptziele, die mit der Veröffentlichung als Open Source verfolgt werden?

6.3. Organisation und Governance

Grundsätzlich wird bei Neuentwicklungen angestrebt, dass der Bund Inhaber der Hauptrechte am Quellcode ist.

Die Art der Projektführung hängt davon ab, welche Organisation beim Bund die Führung hatte (SAFe, HERMES). Bei der Organisationsform ist stets ein minimaler Overhead anzustreben (bestehendes statt neues Gefäss; einfache Gesellschaft statt Verein).

Produktmanagement / Lieferant als / Entwicklung — Vorschlag

Bund / - / Intern
: Da der Bund die direkte Kontrolle behält, sind keine zusätzlichen Organisationsstrukturen nötig. Die veröffentlichende Stelle (z. B. ein Bundesamt) bleibt Eigentümerin und übernimmt sämtliche Koordinationsaufgaben. In diesem Fall kann das Kapitel «Organisation» sehr kurz gehalten werden.

9. https://yml.publiccode.tools/

Gemeinsam / - / Intern

Für dieses Szenario wird in der Regel die Organisation als **einfache Gesellschaft** empfohlen. Damit die gemeinsame Entwicklung und die Abstimmung von Roadmap und Anforderungen möglich sind, empfiehlt es sich, zwei Gremien mit je einer teilnehmenden Person pro Community-Mitglied zu bilden:

- Steuerungsgremium: definiert Strategie und Priorisierung.
- Fachgruppe: erarbeitet Anforderungen und konkrete Inhalte auf Fachebene. Je nach Umfang kann auch je Thema eine eigene Fachgruppe eingesetzt werden.

Eigentümerin des Codes ist das Amt, das ihn veröffentlicht hat.

Werden die Strukturen komplexer oder sind mehr als fünf Nutzende bzw. zusätzliche Mitglieder beteiligt, so kann es sich lohnen zu prüfen, ob die Community besser als Verein organisiert werden sollte. Vergleiche die Erläuterungen in der Variante direkt darunter.

Gemeinsam / - / Dritte

Empfohlen wird die Organisation als **Verein**.
Entweder wird eigens für die Anwendung ein Verein gegründet, oder die Anwendung wird an einen bestehenden Verein übertragen. Ein Verein wird empfohlen, weil sonst die Kostenverteilung komplex wird und innerhalb der Koordinationsaufgaben voraussichtlich genügend Arbeit anfällt, um eine Geschäftsführung in Teilzeit für den Verein anzustellen.

Die Rechte am Code werden dann an den Verein übertragen, der das Community-Management und die Bestellungen bei Lieferanten übernimmt.

Gemeinsam / Auftragnehmer / Intern

Empfohlen wird die Organisation als **Verein**. Entweder wird eigens für die Anwendung ein Verein gegründet, oder die Anwendung wird an einen bestehenden Verein übertragen. Ein Verein wird empfohlen, weil sonst die Kostenverteilung komplex wird und innerhalb der Koordinationsaufgaben voraussichtlich genügend Arbeit anfällt, um eine Geschäftsführung in Teilzeit für den Verein anzustellen.

Die Rechte am Code werden dann an den Verein übertragen, der das Community-Management und die Bestellungen bei Lieferanten übernimmt.
Hier sind die Lieferanten nicht Mitglieder des Vereins, sondern lediglich Auftragnehmer. Die Geschäftsführung sollte nicht von einem Lieferanten gestellt werden.

Gemeinsam / Auftragnehmer / Dritte

Empfohlen wird die Organisation als **Verein**. Entweder wird eigens für die Anwendung ein Verein gegründet, oder die Anwendung wird an einen bestehenden Verein übertragen. Ein Verein wird empfohlen, weil sonst die Kostenverteilung komplex wird und innerhalb der Koordinationsaufgaben voraussichtlich genügend Arbeit anfällt, um eine Geschäftsführung in Teilzeit für den Verein anzustellen.
Die Rechte am Code werden dann an den Verein übertragen, der das Community-Management und die Bestellungen bei Lieferanten übernimmt.
Die Lieferanten sind Mitglieder des Vereins; die Geschäftsführung kann auch von einem Lieferanten gestellt werden.

offen / - / Intern

Es besteht **kein Bedarf für eine detaillierte Beschreibung einer Organisation**, da sie bewusst offen gestaltet ist. Wichtig ist jedoch festzulegen, wer externe Anfragen entgegennimmt und bearbeitet (Verantwortung). Zudem ist zu bestimmen, ob und in welchem Umfang externe Personen einbezogen werden können:

- Keine direkte Einbindung: Externe können nur Vorschläge und Beiträge einreichen (als Pull Requests).
- Einbindung als Contributors: Externe, die häufig und gute Beiträge leisten, werden direkt eingebunden.
- Übergabe an Externe möglich: Tragen nach der Veröffentlichung Externe mehr zur Weiterentwicklung bei als die veröffentlichende Organisation, so kann die Anwendung auch an sie übertragen werden.

Die veröffentlichende Organisation bleibt Eigentümerin. Diese Variante entspricht der Organisation «klassischer» Open-Source-Projekte; siehe für weitere Informationen beispielsweise https://opensource.guide/leadership-and-governance/.

offen / Auftragnehmer / Intern

Es besteht **kein Bedarf für eine detaillierte Beschreibung einer Organisation**, da sie bewusst offen gestaltet ist. Wichtig ist jedoch festzulegen, wer externe Anfragen entgegennimmt und bearbeitet (Verantwortung). Zudem ist zu bestimmen, ob und in welchem Umfang externe Personen einbezogen werden können:

- Keine direkte Einbindung: Externe können nur Vorschläge und Beiträge einreichen (als Pull Requests).
- Einbindung als Contributors: Externe, die häufig und gute Beiträge leisten, werden direkt eingebunden.
- Übergabe an Externe möglich: Tragen nach der Veröffentlichung Externe mehr zur Weiterentwicklung bei als die veröffentlichende Organisation, so kann die Anwendung auch an sie übertragen werden.

Die veröffentlichende Organisation bleibt Eigentümerin.
Diese Variante entspricht der Organisation «klassischer» Open-Source-Projekte; siehe für weitere Informationen beispielsweise https://opensource.guide/leadership-and-governance/.
Der Lieferant sollte nur rein technische Koordinationsaufgaben übernehmen (Code-Review, Beurteilung).

Dritte / Partner / Intern

Es besteht **kein Bedarf für eine detaillierte Beschreibung einer Organisation**, da sie bewusst offen gestaltet ist. Wichtig ist jedoch festzulegen, wer externe Anfragen entgegennimmt und bearbeitet (Verantwortung). Zudem ist zu bestimmen, ob und in welchem Umfang externe Personen einbezogen werden können:

- Keine direkte Einbindung: Externe können nur Vorschläge und Beiträge einreichen (als Pull Requests).
- Einbindung als Contributors: Externe, die häufig und gute Beiträge leisten, werden direkt eingebunden.
- Übergabe an Externe möglich: Tragen nach der Veröffentlichung Externe mehr zur Weiterentwicklung bei als die veröffentlichende Organisation, so kann die Anwendung auch an sie übertragen werden.

Die veröffentlichende Organisation bleibt Eigentümerin. Diese Variante entspricht der Organisation «klassischer» Open-Source-Projekte; siehe für weitere Informationen beispielsweise https://opensource.guide/leadership-and-governance/.

Bezüglich Governance können auch *[BITKOM2023]* Kapitel 4.1, *[IZqCab2023]* oder https://github.com/todogroup/ospo-career-path/blob/main/OSPO-101/module7/README.md#governance-models beigezogen werden.

Beispiele für Vereinsstatuten finden sich in Anhang E.

Hinweis: Da die Freigabe nach Art. 9 EMBAG gewährleistet sein muss, hat die Bundesverwaltung sicherzustellen, dass die Veröffentlichung nicht plötzlich zurückgezogen werden kann (siehe «Wegleitung Open Source in der Beschaffung – Softwareeinkauf nach Art. 9 EMBAG» *[BBL-WL]*, Ziffer V.2).

6.4. Roadmap und Änderungsprozess

Produktmanagement / Lieferant als / Entwicklung — Vorschlag

Bund / Auftragnehmer / -
: Die Bundesbehörde definiert die Roadmap und entscheidet, welche Änderungen umgesetzt werden. Dafür werden die Standardprozesse des Amtes XY verwendet.

Bund / Partner / -
: Die Bundesbehörde definiert die Roadmap in Zusammenarbeit mit der Firma XY. Sie entscheidet, welche Änderungen umgesetzt werden. Dafür werden die Standardprozesse des Amtes XY verwendet.

Gemeinsam / - / Intern
: Der Prozess zur Entwicklung der Roadmap und zur Genehmigung von Änderungen ist von den Mitgliedern der Gesellschaft oder des Vereins festzulegen. Je nach Mitgliederstruktur (Anzahl, Grössenverhältnis usw.) können andere Prozesse angemessen sein.
Die Prozesse müssen jedoch Massnahmen zur Konfliktlösung und Mehrheitsfindung enthalten.

Gemeinsam / - / Dritte
: Der Prozess zur Entwicklung der Roadmap und zur Genehmigung von Änderungen ist von den Mitgliedern der Gesellschaft oder des Vereins festzulegen. Je nach Mitgliederstruktur (Anzahl, Grössenverhältnis usw.) können andere Prozesse angemessen sein.

Die Prozesse müssen jedoch Massnahmen zur Konfliktlösung und Mehrheitsfindung enthalten.

Mit dem Zusatz, dass auch ein Kostenverteilschlüssel festzulegen ist. Zu überlegen ist zudem, wie mit der Kostenteilung für Anpassungen umzugehen ist, die von den betreffenden Mitgliedern nicht gewünscht sind. Insbesondere in Vereinen mit unterschiedlich grossen Mitgliedern können Situationen entstehen, in denen ein grosses Mitglied (z. B. die Bundesbehörde) einen erheblichen Teil der Kosten für eine Erweiterung trägt, ohne daraus einen Nutzen zu ziehen.

offen / - / -
: *Beispiel, nicht für alle Situationen geeignet*
Es besteht kein Bedarf, eine Roadmap zu erstellen; die Anwendung erfüllt die Bedürfnisse derzeit.

Über Änderungen wird in einer offenen Diskussion entschieden und ein Konsens angestrebt. Der endgültige Entscheid liegt bei der Eigentümerin des Codes (Schweizerische Eidgenossenschaft).

6.5. Entwicklungsprozess

Kommt offene Entwicklung zum Tragen, so ist dies hier zu definieren. Wichtig sind die Aufgabenverteilung und die Nennung zentraler Stellen im Community-Konzept. Ebenso, wie die Sicherheit der Repository-Inhalte gewährleistet wird.

Produktmanagement / Lieferant als / Entwicklung — Vorschlag

Bund / - / -

Die Committer-Rechte (Schreibzugriff auf den Code) liegen bei den vom Bund ausgewählten Personen/Stellen/Lieferanten. Nach Ablauf des Vertrags werden die jeweiligen Rechte entzogen.

Die Leistungserbringer und Lieferanten sind für die Durchführung des technischen Code-Review-Prozesses für externe Beiträge verantwortlich, die vom Bund fachlich angenommen wurden. Sie sind für die technische Qualität verantwortlich. Die Lieferanten werden für diese Arbeit vom Bund entschädigt.

Gemeinsam / Auftragnehmer / Intern

Grundsätzlich erhalten alle von einem Community-Mitglied beauftragten Stellen Schreibzugriff auf den Code (Committer-Rechte). Nach Ablauf des Vertrags werden die jeweiligen Rechte entzogen.

Betrifft geänderter Code Kernbereiche der Anwendung, so müssen alle Änderungen von einer eigens dafür von der Community beauftragten Stelle geprüft werden. Diese Stelle ist für die technische Qualität der Anwendung verantwortlich. Sie ist zudem für die Prüfung von Änderungen zuständig, die von der Community fachlich angenommen wurden.

Gemeinsam / Partner / Dritte

Die Committer-Rechte (Schreibzugriff auf den Code) liegen bei den Mitgliedern der Community. Diese Personen organisieren sich selbst und tragen gemeinsam die Verantwortung für die Qualität des Codes.

Beauftragt ein Community-Mitglied einen externen Lieferanten für eine Anpassung oder Erweiterung, so wird die Anpassung von einem Mitglied der Community geprüft. Es wird dafür vom Auftraggeber entschädigt.

Änderungen am Kern der Anwendung müssen von einem zweiten Mitglied der Community geprüft werden.

Die Lieferanten, die Community-Mitglieder sind, sind zudem für die Prüfung externer Beiträge und neuen technischen Codes verantwortlich, der vom Verein fachlich angenommen wurde.

offen / - / -

Anfänglich liegen die Committer-Rechte bei den Entwickelnden oder ihren Arbeitgebern. Committer-Rechte werden allen Entwickelnden erteilt, die während mehrerer Monate aktiv zum Projekt beitragen und angemessene Sozialkompetenz zeigen.

Die einzelnen Entwickelnden behalten ihre Committer-Rechte grundsätzlich, auch wenn sie den Arbeitgeber wechseln oder der Bund einen anderen Lieferanten wählt. Committer-Rechte erlöschen nur, wenn sie freiwillig abgegeben werden oder die Mehrheit der übrigen Committer deren Entzug beschliesst.

Der Bund hat das Recht, einzelne Entwickelnde der von ihm beauftragten Unternehmen als Committer zu benennen. Er hat nicht das Recht, jemandem die Committer-Rechte ohne Grund zu entziehen.

Code-Reviews werden von mindestens einer Person durchgeführt, die als Committer tätig ist. Die Committer organisieren sich selbst.

Der Bund verpflichtet sich, die Review-Arbeit zu entschädigen, soweit dies für ihn finanziell machbar ist.

6.6. Kostenteilung und Lieferanten

Produktmanagement / Lieferant als / Entwicklung — Vorschlag

Bund / Auftragnehmer / Intern

Die Lieferanten werden periodisch über WTO-Ausschreibungen oder ein freihändiges Verfahren ausgewählt (je nach Umfang der Wartungsleistungen).

Bund / Partner / -

Wichtig: Vorgängig prüfen, ob dies im konkreten Fall beschaffungsrechtlich zulässig ist.

Wir werden die Dokumentation zu diesem Punkt ergänzen. Der Bund wählt den Lieferanten XXX als strategischen Partner und strebt eine langfristige Zusammenarbeit an.

Gemeinsam / Auftragnehmer / Intern

Jedes Mitglied der Community beauftragt eigenständig einen oder mehrere Softwarelieferanten mit der Umsetzung der von ihm benötigten Anpassungen und Erweiterungen. Für Anpassungen, die von mehreren Mitgliedern gewünscht werden, wird die Kostenteilung im Einzelfall vereinbart. Das Mitglied mit dem grössten Kostenanteil ist für die Auswahl und Beauftragung des Lieferanten verantwortlich. (Regelungen zur Softwarewartung)

Gemeinsam / Auftragnehmer / Dritte

Der Verein wählt die Softwarelieferanten periodisch zentral über WTO-Ausschreibungen oder freihändige Verfahren aus (je nach Umfang der Wartungsleistungen). Die Kosten werden den Mitgliedern nach einem Schlüssel zugewiesen: (festzulegen: nach Bevölkerung, Nutzenden usw.)

Gemeinsam / Partner / Intern

Jedes Mitglied der Community beauftragt eigenständig einen oder mehrere Softwarelieferanten mit der Umsetzung der von ihm benötigten Anpassungen und Erweiterungen. Für Anpassungen, die von mehreren Mitgliedern gewünscht werden, wird die Kostenteilung im Einzelfall vereinbart. Das Mitglied mit dem grössten Kostenanteil ist für die Auswahl und Beauftragung des Lieferanten verantwortlich.

(Regelungen zur Softwarewartung) Nötigenfalls ergänzt durch einen Beitrag der beteiligten Lieferanten. Beispiel: *Jedes Softwareentwicklungsunternehmen, das Mitglied der Community ist, verpflichtet sich, jährlich mindestens XY Arbeitstage auf eigene Kosten in die Weiterentwicklung, die technologische Aktualisierung oder das Marketing der Anwendung zu investieren.*

Gemeinsam / Partner / Dritte

Der Verein wählt die Softwarelieferanten periodisch zentral über WTO-Ausschreibungen oder freihändige Verfahren aus (je nach Umfang der Wartungsleistungen). Die Kosten werden den Mitgliedern nach einem Schlüssel zugewiesen: (festzulegen: nach Bevölkerung, Nutzenden usw.) Nötigenfalls ergänzt durch den Zusatz: *Jedes Softwareentwicklungsunternehmen, das Mitglied der Community ist, verpflichtet sich, jährlich mindestens XY Arbeitstage auf eigene Kosten in die Weiterentwicklung, die technologische Aktualisierung oder das Marketing der Anwendung zu investieren.*

Dritte / Auftragnehmer / -

Die Lieferanten werden periodisch über WTO-Ausschreibungen oder ein freihändiges Verfahren ausgewählt (je nach Umfang der Wartungsleistungen). Mit dem Zusatz: *Die von potenziellen Lieferanten beigetragenen Verbesserungen und Aktivitäten in der Community werden als Faktor in der Bewertung berücksichtigt.*

Von Externen gewünschte Änderungen werden nicht vom Bund finanziert, sondern müssen von diesen selbst in Auftrag gegeben und bezahlt werden.

Dritte / Partner / -

Die Lieferanten werden periodisch über WTO-Ausschreibungen oder ein freihändiges Verfahren ausgewählt (je nach Umfang der Wartungsleistungen). Mit dem Zusatz: *Die von potenziellen Lieferanten beigetragenen Verbesserungen und Aktivitäten in der Community werden als Faktor in der Bewertung berücksichtigt. Von Externen gewünschte Änderungen werden nicht vom Bund finanziert, sondern müssen von diesen selbst in Auftrag gegeben und bezahlt werden.*

6.7. Marketing

Produktmanagement / Lieferant als / Entwicklung — Vorschlag

Bund / Auftragnehmer / -
Der Bund vermarktet die Anwendung nicht aktiv, veröffentlicht sie aber auf Open-Source-Softwareplattformen. Die Veröffentlichung der Anwendung wird in Fachgremien bekannt gegeben. Melden sich Interessierte, so prüfen wir, ob eine gemeinsame Community aufgebaut oder mit deren Version der Software weitergearbeitet wird.

Gemeinsam / Partner / -
Zusätzlich:
Lieferanten dürfen die Anwendung vermarkten und weitere Interessierte suchen. Der Bund darf als Referenz genannt werden. Wir nehmen bewusst in Kauf, dass dadurch abweichende Versionen der Software entstehen können.

Gemeinsam / Auftragnehmer / Intern
Die Anwendung wird von den Mitgliedern oder vom Verein vermarktet.

Gemeinsam / Partner / Intern
Die Anwendung wird von den Lieferanten vermarktet.

Gemeinsam / - / Dritte
Die Anwendung wird vom Verein, von der einfachen Gesellschaft und deren Mitgliedern vermarktet.

offen / - / -
Die Anwendung darf von Lieferanten vermarktet werden, oder es kann auf explizites Marketing verzichtet werden.

Alternative ohne explizites Marketing: Der Bund vermarktet die Anwendung nicht aktiv, veröffentlicht sie aber auf Open-Source-Softwareplattformen. In Fachgremien weisen wir darauf hin, dass wir die Anwendung veröffentlicht haben und dass eine Zusammenarbeit willkommen ist. Wir zeigen Interessierten die Software und versuchen, sie in unsere Prozesse zur Gestaltung der Roadmap einzubinden.

6.8. Umgang mit externen Beiträgen

Produktmanagement / Lieferant als / Entwicklung — Vorschlag

Bund / Auftragnehmer / -
Externe Beiträge und Anfragen müssen auch dann bearbeitet werden, wenn der Bund alleiniger Entscheidträger bezüglich der Software bleibt (ausser bei minimaler Freigabe, wo ein Fork unvermeidlich ist). Wir schlagen folgendes Vorgehen vor:

Umgang mit gemeldeten Fehlern
Wir versuchen, von externen Personen gemeldete Fehler zu reproduzieren, und fragen nötigenfalls nach. Gelingt die Reproduktion, so gibt der Bund die Fehlerbehebung in Auftrag. Lässt sich der Fehler nicht verifizieren oder ist der Aufwand zu dessen Behebung unverhältnismässig hoch, so entschuldigen wir uns bei der meldenden Person und fragen, ob sie einen Pull Request zur Behebung einreichen könnte.

Umgang mit Anfragen
Wir beantworten jede eingehende Anfrage. Lässt sich eine Anfrage nicht mit vertretbarem Aufwand klären und erscheint ein weiterer Zeiteinsatz ineffizient, so entschuldigen wir uns unter Hinweis auf beschränkte Ressourcen und bieten an, die anfragende Person für die weitere Beratung mit einem Lieferanten in Kontakt zu bringen.

Umgang mit Pull Requests
Wir prüfen jeden Pull Request sorgfältig. Um unnötigen Aufwand auf Seiten der beitragenden

Person zu vermeiden, weisen wir sofort darauf hin, wenn etwas unserer Roadmap widerspricht oder fraglich ist, ob wir den Beitrag annehmen. Der Bund entscheidet eigenständig und nach fachlichen Kriterien, welche Beiträge er annimmt. Fehlerkorrekturen, die den Review-Prozess bestanden haben, werden immer angenommen.

Umgang mit Forks
Wir unterstützen Forks unserer Anwendung. Wer die Anwendung einführt, sollte einen eigenen Fork erstellen. Wir prüfen die entstehenden Forks mindestens einmal jährlich und übernehmen gute und passende Erweiterungen.

Gemeinsam / - / -
Extern bezeichnet Beiträge von ausserhalb der definierten Community (Vereinsmitglieder). Abweichungen von oben:
Umgang mit Forks
Wir wollen (langlebige) Forks der Anwendung verhindern. Interessierte sollen stattdessen der Community direkt beitreten. Innerhalb der Community versuchen wir sicherzustellen, dass alle Mitglieder die Hauptversion nutzen.

Offen / - / -
Der Aufbau einer funktionierenden Community und einer offenen Kultur ist uns wichtig.

Umgang mit gemeldeten Fehlern
Wir versuchen, gemeldete Fehler zu reproduzieren und zu beheben. Wir setzen die aktive Mitwirkung der meldenden Personen voraus. Wenn möglich sollten sie direkt einen Pull Request mit einer Fehlerkorrektur erstellen.

Umgang mit Anfragen
Wir behandeln jede Anfrage innerhalb von höchstens einer Woche. Anfragen aktiver Beitragender werden mit Priorität und vertieft behandelt. Wir versuchen, andere Mitglieder in die Bearbeitung von Anfragen einzubeziehen.

Umgang mit Pull Requests
Wir wollen möglichst viele qualitativ hochwertige Pull Requests erhalten. Wir beziehen alle aktiven Beitragenden in die Reviews und die technische und fachliche Beurteilung von Pull Requests ein. Bei umstrittenen Beiträgen versuchen wir, über eine informelle Abstimmung zu einem Entscheid zu gelangen.

Umgang mit Forks
Wir versuchen, Forks durch aktives und offenes Community-Management unnötig zu machen. Entstehen dennoch Forks, so prüfen wir diese aktiv und versuchen, die dort entwickelten Erweiterungen und Verbesserungen zu übernehmen. In solchen Fällen fordern wir aktiv Pull Requests an.

Allgemein ist festzuhalten, dass Beiträge von der Unterstützung anderer Nutzender bis zur Übernahme der Verantwortung für ganze Teile eines Projekts reichen können (siehe *[BITKOM2023]* Kapitel 4.2).

6.9. Betrieb der Anwendung

Produktmanagement / Lieferant als / Entwicklung — Vorschlag

Gemeinsam / - / -
Ein zentraler Betrieb kann optional zusätzlich vom zentralen Verein angeboten werden, im Sinne von Software as a Service (SaaS). Es ist zu prüfen, ob dies gewünscht ist.

- / Partner / -
Es ist festzulegen, ob der Lieferant die Software auch betreiben soll.

- / Führung / -
Es ist festzuhalten, dass der Lieferant für den Betrieb verantwortlich ist.

Supportstufen und Supportkontakte sollten in diesem Kapitel ebenfalls festgelegt werden.

7. Wichtige Informationen für Community-Managerinnen und -Manager

7.1. Es braucht Zeit

Der Aufbau von Communities ist etwas, das langfristig geplant werden sollte. Ausdauer und Beständigkeit sind zwei wichtige Eigenschaften für eine erfolgreiche Community.

7.2. Weitere Anregungen zum Community-Aufbau

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

7.3. Kommunikation

Die Veröffentlichung von Software und Dokumentation eröffnet der Bundesverwaltung einen neuen Kommunikationskanal. Geschieht dies nicht sorgfältig und professionell, so kann das öffentliche Ansehen des Bundes leiden.

OSS-Communities in der Bundesverwaltung stehen bezüglich Kommunikation in einem Spannungsfeld: Einerseits gelten die allgemeinen **Vorgaben für die Kommunikation**, andererseits soll die Community eine **einfache formelle und informelle Zusammenarbeit** über Organisationsgrenzen hinweg ermöglichen, vorzugsweise ohne formelle Qualitätskontrollen und Freigaben.

Die Nutzung öffentlicher Repositories und allenfalls öffentlicher Issue-Trackings und Wikis bedeutet, dass Äusserungen einzelner Projektmitarbeitender ungefiltert für die Öffentlichkeit sichtbar sind. Dies verlangt von den Projektmitarbeitenden ein professionelles Auftreten und die Einhaltung von Qualitätsstandards für Veröffentlichungen in solchen Umgebungen.

In OSS-Projekten wird dies normalerweise über den **Code of Conduct** geregelt. Der Verhaltenskodex der Bundesverwaltung[10] ist sehr allgemein gehalten, doch der Kerngrundsatz ist hier anwendbar: «Die Mitarbeitenden erfüllen ihre Aufgaben verantwortungsbewusst, integer und loyal. Auch im Privatleben achten sie darauf, das Ansehen, die Glaubwürdigkeit und das Image des Bundes nicht zu beeinträchtigen.»

7.4. Offene Entwicklung

Ist die Softwareentwicklung von Anfang an in einem offenen Repository geplant, so erfolgt die Freigabe automatisch. Die massgebenden Checklisten nach der Anleitung Em002-2 sind zu Beginn auszufüllen, und die Entwicklungsprozesse des Leistungserbringers bzw. Lieferanten müssen dies zulassen. Der Vorteil ist, dass die Community von Anfang an aufgebaut werden kann. Mehrere Organisationen können zudem gemeinsam an der Entwicklung mitwirken.

Geschlossene Repositories sind eine Zwischenform, werden aber für andere Partner in einer Zusammenarbeit eingesetzt.

10. https://www.epa.admin.ch/dam/epa/de/dokumente/aktuell/medienservice/120_verhaltenskodex_d.pdf.download.pd

7.5. Benutzerverwaltung

Das Projekt (oder die Trägerorganisation) legt fest, ob Entwickelnde und andere Projektmitarbeitende unter eigenem Namen oder anonym am Projekt arbeiten. Arbeiten die Personen nicht anonym und verfügen sie bereits über Logins auf den Plattformen, so ist es am einfachsten, diese zu verwenden (ob geschäftlich oder privat).

Die Plattform sollte grundsätzlich die Einbindung Dritter erlauben, da sonst keine Entwickelnden ausserhalb des Bundes mitarbeiten können.

Beim Verlassen des Projekts sind die entsprechenden Rechte am Projekt zu löschen. Dafür ist das Produktmanagement verantwortlich.

7.6. Zusammenführung von Communities

Wo möglich sollte die Zahl der Communities klein gehalten werden. Das bedeutet:

- Besteht bereits eine Community, in der für ähnliche Probleme mit denselben Personen eine ähnliche Struktur vorgesehen ist, so sollte eine einzige Community genutzt werden.
- Mehrere Projekte, die zusammenhängen und dieselbe Zielgruppe ansprechen, können zu einer Community zusammengeführt werden.
- Es sollten nicht unnötig grundlegend andere Community-Strukturen geschaffen werden, wo bereits bestehende vorhanden sind.

7.7. Umgang mit Beiträgen Dritter

Der Beitrag von Software an andere Projekte wird in Kapitel 3.3 behandelt. Die gleichen Grundsätze gelten umgekehrt.

Wichtig sind die folgenden Punkte:

- Es muss ein geeignetes Contributor Licence Agreement verwendet werden, um sicherzustellen, dass der Quellcode anschliessend dem Bund gehört (siehe auch Em002-3 Leitfaden OSS-Lizenzierung, Kapitel zu den Contributor Licence Agreements). Dieses ist von der Eigentümerin bzw. dem Eigentümer des Beitrags zu unterzeichnen.
- Bestätigungen sind abzulegen.
- Beiträge sind vor der Integration in die Codebasis zu prüfen.
- Eingaben sollten zeitnah beantwortet (Issues, Requests, Pull Requests) und möglichst konstruktiv behandelt werden.
- Sämtliche Kanäle und Governance-Regelungen sollten im Community-Konzept, auf der Website und im Repository aufgeführt werden.

7.8. Sicherheitsrelevante Meldungen

Neben üblichen Fehlermeldungen ist – wo es die Art des Projekts erfordert – vorzusehen, dass sicherheitsrelevante Schwachstellen vertraulich gemeldet werden können, wie dies beispielsweise bei einem Bug-Bounty-Programm der Fall ist.

Als Beispiel können die Dokumente von trustbroker.swiss dienen:

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

7.9. Forks vermeiden

Forks sind in der Open-Source-Welt zwar natürlich, doch ist die Wiedereingliederung von Quellcode, Funktionen usw. aus Forks alles andere als einfach. Es kann sich lohnen, auf die Personen oder Organisationen zuzugehen, die einen Fork erwägen, und zu versuchen, sie im Hauptprojekt zu halten. Ein Ausgleich zwischen konkurrierenden Interessen ist in solchen Fällen oft nötig.

7.10. Kostenaufteilung

Es ist sinnvoll, zwischen den Hauptpartnern allgemeine Verteilschlüssel zu vereinbaren. Diese können alle paar Jahre oder bei grundlegender Änderung der Situation angepasst werden. Für eine effiziente Arbeit an der Software sollten die grossen Partner tendenziell etwas grössere Anteile übernehmen, als sie eigentlich müssten (nötigenfalls auch die Gesamtkoordination). Die Organisation soll arbeiten und nicht über Finanzen streiten. Erstreckt sich die Zusammenarbeit über mehrere Projekte (beispielsweise die Branchenlösungen im Bereich Normalspurbahnen), so sollte für alle Projekte ein einheitlicher 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 Nutzenverteilung (z. B. zwischen Bundesbehörde und einem Lieferanten, der auf weitere Kundschaft hofft)
- Zahl der Einwohnenden/Nutzenden (z. B. zwischen Ämtern, Gemeinden, Städten)
- Finanzkraft (z. B. Kantone)
- Bruttoinlandprodukt (z. B. zwischen Kantonen)
- Wer mehr Kontrolle will

Um Diskussionen zu vermeiden, sollte der Verteilschlüssel nicht einfach an das Jahresbudget gekoppelt werden. Eine Klausel im Sinne des Verursacherprinzips kann für einzelne Partner raschere Anpassungen ermöglichen, wenn sie bereit sind, zusätzlich zu zahlen.

Der Verteilschlüssel sollte auch für Gemeinkosten, Wartung/Support und ein Minimum an Weiterentwicklung gelten. Diese Teile sollten wenn möglich langfristig gelöst werden.

Zu bedenken ist, dass bei Drittfinanzierung die betreffenden Parteien typischerweise mehr Mitsprache wollen. Das Community-Management muss darauf vorbereitet sein.

7.11. Ergänzende Leistungen nach Art. 9 Abs. 5 und 6 EMBAG

Die Bundesbehörden können (selbst, über Leistungserbringer oder Lieferanten) ergänzende Leistungen erbringen, insbesondere für Integration, Wartung, Sicherstellung der Informationssicherheit und Support.

Die Einschränkung ist, dass sie den Aufgaben der Behörden dienen und mit verhältnismässigem Aufwand erbracht werden können.

Daraus folgt, dass die Bundesbehörde nicht das Entwicklungsunternehmen für Dritte sein kann. Sie behebt Fehler, kann sicherheitsrelevante Aspekte behandeln, kann etwas Support und Integrationshilfe leisten (alles Tätigkeiten, die dem Aufbau einer Community dienen). Neuentwicklungen und zusätzliche Funktionen für Dritte sollten sich in erster Linie am normalen Arbeitszweck der Behörde orientieren. Selbstverständlich kann es sein, dass die Bereitstellung der Software Teil der Aufgabe ist; in diesem Fall können jederzeit Entwicklungen erfolgen. Festgehalten wird zudem, dass die Softwareentwicklung nicht die Kernaufgabe der Bundesbehörde ist.

Das bedeutet: Werden Funktionen im Wesentlichen ausschliesslich für Dritte entwickelt, so sind sie partnerschaftlich oder über den Lieferanten zu integrieren.

Um beschaffungsrechtliche Probleme zu vermeiden, ist bei der Beschaffung von Software und Leistungen darauf zu achten, dass Funktionen Dritten (allenfalls nur staatlichen Dritten) durch Lieferanten bereitgestellt werden können.

Die Festlegung von Gebühren wurde bereits im Kapitel «Support» angesprochen. Das EMBAG ist bereits die gesetzliche Grundlage für solche Gebühren. Diese sind auf der Website der jeweiligen Bundesbehörde zu kommunizieren. Beispiele sind in Anhang G aufgeführt.

Art. 9 Abs. 6 hält fest, dass der Privatwirtschaft normalerweise keine Konkurrenz gemacht werden soll. Das bedeutet, dass die Gebühren nicht zu tief angesetzt werden und mindestens kostendeckend sein sollten. Im Zweifelsfall sind Stundensätze eher zu hoch als zu tief anzusetzen.

Generell wird empfohlen, mit ein bis drei verschiedenen Standardstundensätzen zu arbeiten.

Will ein Bundesdepartement Ausnahmen zu Abs. 6 festlegen, so darf dies der Privatwirtschaft keine Konkurrenz machen. Unter Umständen ist es am einfachsten, die Gebühren festzulegen und sie zu erhöhen, falls Widerstand von Privatunternehmen entsteht, welche die Leistung tatsächlich anbieten können.

7.12. Digitale Souveränität

Solange der Bund kein eigenes Repository hat, wird zur Sicherstellung der digitalen Souveränität empfohlen, regelmässig Kopien öffentlicher Repositories zu erstellen.

Zu beachten ist, dass nicht nur der Quellcode, sondern auch weitere Informationen wie Issue-Tracking, Wiki, Konfigurationen usw. kopiert werden.

Zudem ist die Benutzerverwaltung zu lösen, um die Kontrolle über den veröffentlichten Quellcode sicherzustellen. Übergeordnetes Ziel ist es, eine von der jeweiligen Plattform unabhängige Entwicklung zu gewährleisten und der Community einen Wechsel zu ermöglichen.[11]

8. Anhang

8.1. Änderungen gegenüber der Vorversion

- Kapitel 1 «Das Wichtigste auf einen Blick» ergänzt
- Zweck der Community in Kapitel 3 ergänzt
- Kapitel 3.1 bis 3.3 zur Zusammenarbeit ergänzt
- Neues Kapitel 7.7 Umgang mit Beiträgen Dritter
- Festgehalten, dass Lieferanten nicht vollständig eigenständig veröffentlichen sollen (nach *[BBL-WL]*, V.2)
- Publicode.yml erwähnt
- Abstimmung mit dem neuen Em002-7
- Weitere redaktionelle Änderungen und Verbesserungen

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

8.2. Referenzen

Siehe *Em002 Strategische Leitlinien für Open-Source-Software in der Bundesverwaltung*.

8.3. Abkürzungen

Siehe Em002 Strategische Leitlinien für Open-Source-Software in der Bundesverwaltung und Em002-6 FAQ zu OSS.

8.4. Beispiele bestehender Community-Konzepte

Im Sinne einer Sammlung von Best Practices folgen nachstehend Community-Konzepte und vergleichbare Dokumente, die bereits erarbeitet wurden. Dies sind nicht zwingend nur Konzepte von Bundesbehörden, sondern auch von anderen Organisationen.

- GERES Community (http://geres-community.ch)

Einordnung:

- Produktmanagement a) Gemeinsame Entwicklung
- Lieferant eher a) Auftragnehmer
- Kostenverteilung: b) Kostenteilung

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

- iGov Portal

Dieses Kapitel enthält derzeit Konzepte des Kantons Bern und wird ergänzt, sobald Beispiele des Bundes verfügbar sind.

8.5. Beispiele von Zusammenarbeiten

Die folgenden Beispiele zeigen, wie Zusammenarbeiten strukturiert wurden.

8.6. Zusammenarbeit: ZenDiS in Deutschland

ZenDiS[12] ist eine Organisation, deren Ziel es ist, die Abhängigkeit in der öffentlichen Verwaltung (primär in Deutschland) durch die Förderung von Open-Source-Software zu verringern.

12. ZenDiS-Website: Co-creating Digital Sovereignty

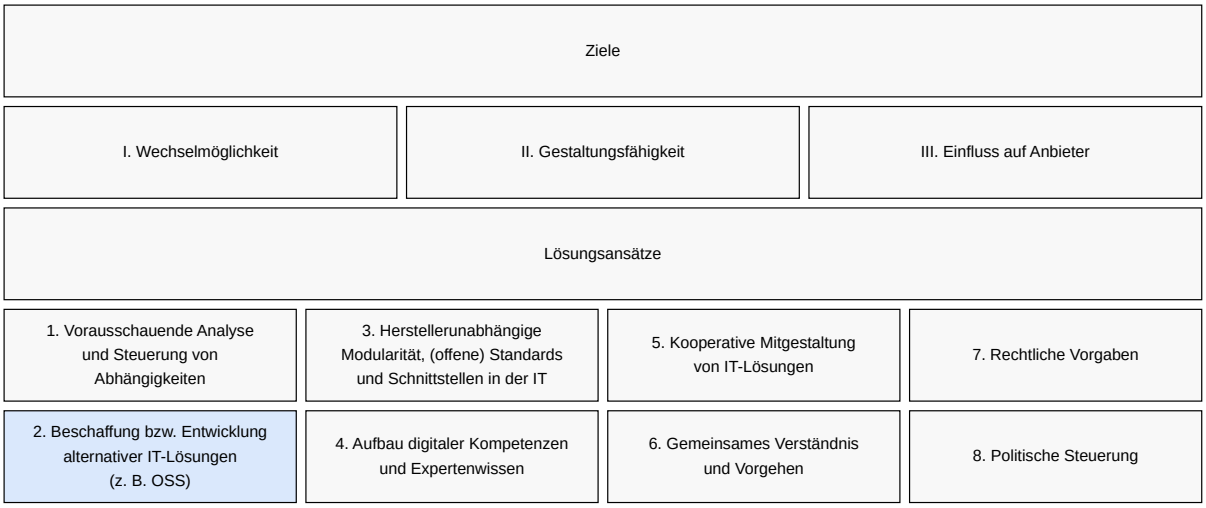


Abbildung 1: Ziele und Ansätze zur Stärkung der digitalen Souveränität (ZenDiS)

ZenDiS arbeitet dazu mit strategischen Partnern zusammen.

Eine Zusammenarbeit zwischen der Bundesverwaltung und ZenDiS dürfte daher kaum uneingeschränkt möglich sein, und eine Koordination wäre nicht verbindlich. Dies kann sich mit der Zeit ändern.

Konkret müsste die Bundesverwaltung eine eigene Organisation für die Schweiz aufbauen und sich informell mit ZenDiS abstimmen. Dies könnte allenfalls über eOperations erfolgen, wobei eOperations erst tätig wird, wenn ein konkreter Bedarf angemeldet und ein Budget bereitgestellt wird. Eine Begründung für eine solche Parallelorganisation liesse sich im Kontext von E-Government oder EMBAG finden.

Hinweis: ZenDiS untersteht dem EU- bzw. deutschen Beschaffungsrecht. Es bestehen erhebliche Unterschiede zur Schweiz. ZenDiS lässt sich daher nicht eins zu eins in der Schweiz nachbilden, und eine Zusammenarbeit mit ZenDiS ist nicht ohne Weiteres möglich.

8.7. Zusammenarbeit: Open Trip Planner (OTP)

Open Trip Planner[13] ist eine Open-Source-Anwendung für die Reiseplanung. Sie ist als Projekt der Software Freedom Conservancy in New York organisiert. Einzelne Entwickelnde sind bei Verkehrsunternehmen und Lieferanten angestellt. Bemerkenswert ist, dass ein Akteur der öffentlichen Hand – konkret ENTUR[14] – eine der führenden Rollen übernommen hat. ENTUR entwickelt primär für den eigenen Bedarf, hat aber auch einen Kooperationsfokus, insbesondere für die nordischen Länder. ENTUR ist als Unternehmen im Eigentum des norwegischen Verkehrsministeriums organisiert. Als grösster Akteur übernimmt ENTUR auch einen erheblichen Teil der Koordinationskosten. Alle können Feature-Wünsche einreichen; diese werden in Koordinationssitzungen diskutiert und entwickelt, wenn jemand sie finanziert. Die Finanzierung erfolgt primär pro Feature durch die interessierte Partei bzw. die interessierten Parteien. Im Kern trägt jede Partei ihre eigenen Kosten. Supportleistungen müssten in der Schweiz zudem separat ausgeschrieben werden.

13. OpenTripPlanner
14. https://entur.no/

8.8. Neue vom Bund angestossene Zusammenarbeit: Swiss Trust Broker

Der Trust Broker[15] ist eine Software des Bundes für eIAM, die als Open Source verfügbar ist. Die Entwicklung erfolgt durch den Bund. Mehrere Kantone haben entschieden, sie entweder direkt als SaaS oder mit eigenen Instanzen einzusetzen. Da sie unter der AGPL-Lizenz verfügbar ist, kommen vorgenommene Erweiterungen wiederum dem Bund zugute. Aufgrund ihrer sicherheitskritischen Natur erfolgt die Entwicklung für die Bundesverwaltung ausschliesslich durch den Bund. Andere Organisationen können Anforderungen einreichen, die umgesetzt werden, wenn das Projektteam sie als angemessen erachtet.

8.9. Neue von den Kantonen angestossene Zusammenarbeit: inosca

Über inosca[16] haben mehrere Kantone eine Entwicklungsgemeinschaft für elektronische Bewilligungen aufgebaut, die ihren Ursprung im Projekt eBau hat.[17] eBau umfasst auch Caluma,[18] ein Framework für die gemeinsame Bearbeitung – ein Beispiel für eine Komponente, die aus einer breiteren Lösung herausgelöst und zu einem eigenständigen, breit anwendbaren Framework entwickelt wurde. Dieses kann auch in anderen Projekten genutzt werden, selbst wenn einzelne Teile Ausnahmebedingungen unterliegen, sodass möglichst viele andere Stellen vom Code profitieren können.

Die Community trifft sich mindestens einmal monatlich (optional zweimal), um sich über technische Themen auszutauschen und die gemeinsame Roadmap zu planen. Alle Beteiligten können Themen einbringen. Wird ein Thema als für die Community relevant erachtet, so melden sich interessierte Kantone und organisieren sich in einer Arbeitsgruppe. Die Ergebnisse fliessen laufend an die Community zurück. inosca arbeitet nach einem Grundprinzip: Wer gestaltet, finanziert. Das heisst, alle an einer Arbeitsgruppe beteiligten Kantone teilen sich die Kosten der neuen Funktion. Die Funktion wird ausgeschrieben und die Kosten werden nach einer vordefinierten Kostenteilungsformel unter der Community aufgeteilt. Sobald eine Funktion fertiggestellt und eingeführt ist, steht sie sofort allen anderen Kantonen frei zur Verfügung.

Die Community hat zusätzlich entschieden, einen bescheidenen festen Jahresbetrag für die administrativen Kosten des Community-Betriebs zurückzustellen. Dieses Budget wird jedes Jahr den Parteien zugewiesen, die eine organisatorische Rolle übernehmen.

8.10. SNOWPACK der WSL – direkte Beiträge von Entwickelnden

SNOWPACK[19] ist ein Open-Source-Projekt der WSL zur Modellierung der Schneedeckenbildung. Es handelt sich um ein hochspezialisiertes Modell, und die Community wurde entsprechend in die bestehende Fachgemeinschaft integriert. Einzelne Issues sind bereits als mögliche Beiträge vorgemerkt. Das Arbeitsmodell ist hier, dass jede Partei ihre eigenen Kosten trägt und Funktionen beisteuert. Richtlinien zu Coding Style und Release-Prozess sind in diesem Zusammenhang ebenfalls dokumentiert.

8.11. Anregungen für Vereinsstatuten

Die Statuten sollten so einfach wie möglich gehalten werden.

Die folgenden Statuten können als Anregung dienen.

15. GitHub - trustbroker-swiss/trustbroker.swiss: Documentation of the trustbroker.swiss service
16. https://inosca.ch/
17. https://github.com/inosca/ebau
18. https://github.com/projectcaluma
19. https://snow-models.gitlab-pages.wsl.ch/snowpack-web/

- 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 sie verfügbar sind.

8.12. Beispiele für Support und Gebühren

- Gebühren für statistische Dienstleistungen: https://www.fedlex.admin.ch/eli/cc/2003/326/de
- Gebühren des EDÖB: https://www.edoeb.admin.ch/edoeb/de/home/datenschutz/grundlagen/dsfa.html
- Kostenlose Leistungen des IGE für Kurse im Bereich des geistigen Eigentums (nicht Software, aber als Beispiel): https://www.ige.ch/de/dienstleistungen/ip-academy/allgemeine-informationen/preise

Beispiele werden ergänzt, sobald sie verfügbar sind.