



Kompetenzzentrum
Öffentliche IT

Forschung für den digitalen Staat

Basanta E. P. Thapa, Christian Weidner & Dorian Grosch

Ein Open-Source-Ökosystem für die öffentliche Verwaltung

Gefördert durch:



Bundesministerium
des Innern
und für Heimat



Fraunhofer
FOKUS

Impressum

Autor:innen:

Basanta Thapa, Christian Weidner & Dorian Grosch

Gestaltung:

Reiko Kammer

Illustration:

Christine Rösch

Herausgeber:

Kompetenzzentrum Öffentliche IT
Fraunhofer-Institut für Offene Kommunikationssysteme FOKUS
Kaiserin-Augusta-Allee 31, 10589 Berlin
Telefon: +49-30-3463-7173
Telefax: +49-30-3463-99-7173
info@oeffentliche-it.de
www.oeffentliche-it.de
www.fokus.fraunhofer.de

ISBN: 978-3-948582-16-6

1. Auflage August 2022

Dieses Werk steht unter einer Creative Commons Namensnennung 3.0 Deutschland (CC BY 3.0) Lizenz. Es ist erlaubt, das Werk bzw. den Inhalt zu vervielfältigen, zu verbreiten und öffentlich zugänglich zu machen, Abwandlungen und Bearbeitungen des Werkes bzw. Inhaltes anzufertigen sowie das Werk kommerziell zu nutzen. Bedingung für die Nutzung ist die Angabe der Namen der Autor:innen sowie des Herausgebers.

Logos und vergleichbare Zeichen dürfen nur im Kontext des Werkes genutzt und nicht abgewandelt werden.

Von uns verwendete Zitate unterliegen den für die Quelle geltenden urheberrechtlichen Regelungen.

Das letzte Abrufdatum der Onlinequellen in den Fußnoten ist der 25.07.2022.

Danksagung

Unser Dank gilt allen Teilnehmenden der Onlinebefragung im Januar 2021 sowie insbesondere den Expert:innen aus Verwaltung, Wirtschaft, Wissenschaft und Zivilgesellschaft, die uns Einblicke in ihre Auseinandersetzung, Arbeit und Erfahrungen zum Thema FOSS und öffentliche Verwaltung im Rahmen unserer Interviews gewährt haben: Christian Knebel, Christina Lang, Claus Mullie, Florian Effenberger, Jan-Ole Beyer, Laura Scheske, Lilith Wittmann, Stefan Kaufmann, Thomas Bönig und Tom Gordon.

Icons für Infografik: <https://fontawesome.com/>

Das Wichtigste in Kürze

Der Einsatz von Freier und Open-Source-Software (FOSS) in der öffentlichen Verwaltung steht aktuell auf der politischen Agenda. Das Besondere an FOSS: Jede:r kann die Software und ihren Quellcode frei verwenden, vervielfältigen und verändern. Damit dies in geordneten Bahnen verläuft, haben sich in der Open-Source-Szene eigene Organisationsstrukturen herausgebildet. Dazu gehören Maintainer:innen, die über die Aufnahme von Code-Ergänzungen in die nächste Programmversion entscheiden, sowie Governance-Strukturen, die Grundregeln und Entwicklungs-Roadmaps festlegen.

Wie kann ein nachhaltiges Ökosystem zwischen Open-Source-Codebases, staatlicher Finanzierung und Einflussnahme sowie Dienstleistern aus Wirtschaft und Zivilgesellschaft aussehen? Um diese Frage zu beantworten, haben wir bestehende Literatur ausgewertet und Expert:innen befragt.

Ergebnis ist eine Skizze, welche ein solches ideelles Ökosystem beschreibt. Wir haben dieses »FOSSGov-Ökosystem« als Netzwerk aus Rollen (Kreise) und deren Beziehungen zueinander (Pfeile) dargestellt (siehe Abbildung 3 auf den Seiten 16/17). Die Organisationen – bzw. die Akteure – die die Rolle ausführen, werden bei vielen funktionsbezogenen Rollen bewusst offengelassen. Für einige der Funktionen lassen sich bereits Institutionen benennen – zum Beispiel das geplante »Zentrum für Digitale Souveränität« oder die kürzlich erschienene Plattform »Open CoDE« – andere Rollen gilt es noch auszufüllen. Zu jedem Szenario der Skizze wird der entsprechende Ausschnitt des Netzwerks mit beteiligten Rollen und Beziehungen abgebildet, Kurzbeschreibungen der Rollen sind im Anhang aufzufinden.

Interaktiv und mit Detailbeschreibungen zu allen Rollen und den meisten Beziehungen ist das Ökosystem in einer externen Webanwendung zu erkunden: <https://kumu.io/oefit/fossgov>

Eine kurze Bedienungsanleitung, die in die wichtigsten Funktionen und die Datenschutzinformationen der Webanwendung einführt, ist im Anhang dieser Publikation zu finden.

Inhalt

Das Wichtigste in Kürze	3
1. Einleitung	6
1.1 Zur Entstehung der FOSSGov-Skizze	6
1.2 Anforderungen	6
1.3 Prinzipien	6
2. Grundlagen für Open-Source-Ökosysteme	9
2.1 Was ist Freie und Open Source Software (FOSS)?	9
2.2 Grundlegende Rollen in Open-Source-Ökosystemen	9
2.3 Selbstverstärkende FOSS-Innovationsspirale	10
2.4 Anreizstrukturen in FOSS-Ökosystemen	11
3. Anforderungen an das FOSSGov-Ökosystem	13
3.1 FOSS finanzieren	13
3.2 Interessen der öffentlichen Verwaltung vertreten	13
3.3 Ausreichende Staatsferne der Codebases	14
3.4 Gemeinnützigkeit erhalten	14
3.5 Neue Codebases für staatliche Bedarfe schaffen	14
3.6 Wissen zu FOSS in der Verwaltung managen	14
3.7 Entwicklungsbedarfe bündeln	14
3.8 Regionale IT-Wirtschaft fördern	15
3.9 Software zertifizieren	15
3.10 Offen für verschiedene Lösungsvarianten sein	15
4. Das FOSSGov-Ökosystem in Aktion: Anwendungsszenarien	19
5. Gestaltungsprinzipien für ein FOSS-Ökosystem für die öffentliche Verwaltung	29
5.1 Von der Nehmer- zur Gärtnermentalität wandeln	29
5.2 Koordination der Verwaltungsseite	29
5.3 Finanzierung und Kontrolle entkoppeln	29
5.4 Mittler zwischen Verwaltung und FOSS-Community schaffen	30
5.5 Vorteile von Dezentralität nutzen	30
5.6 Forks vermeiden	30
6. Fazit	32
7. Anhang	34
7.1 Verzeichnis der Rollen	34
7.2 Quellen & Weiterführendes	38
7.3 Anleitung zur Nutzung der webbasierten Skizze	40



1. Einleitung

Um Open-Source-Software für die öffentliche Verwaltung strategisch und systematisch brauchbar zu machen, muss die öffentliche Hand mehr als Gärtnerin denn als Käuferin agieren. Denn auch wenn die Logik von Open Source stellenweise als Markt mit Gewinninteressen funktioniert, muss diese in einem eigenem Ökosystem mit vielfältigen Rollen und Interessen verstanden werden, die ineinandergreifen müssen, damit Softwareprojekte gedeihen. Es gilt zu säen, gezielt zu düngen, Wachstum in die richtige Richtung zu lenken, Wildwuchs zu begrenzen und auch Zurückhaltung zu üben, um schließlich ernten zu können. Mit der wichtigen Besonderheit, dass digitale Früchte beliebig oft gegessen werden können.

Praktische Fragen, welche Rollen, Institutionen und Prozesse nachhaltige und professionelle Strukturen Freier und Open-Source-Software für die deutsche öffentliche Verwaltung schaffen können, stellten sich im Bund spätestens mit dem Koalitionsvertrag 2021. Neben dem politischen Bekenntnis der Regierungskoalition zu Open-Source-Software in der öffentlichen Verwaltung lassen die bereits zuvor vom IT-Planungsrat vorgelegten Pläne für ein Zentrum für Digitale Souveränität, das vornehmlich Open Source fördern soll, zudem deutlichen Umsetzungswillen erkennen.

Der Aufstieg des Themas auf die politische Tagesordnung ist eng mit der Debatte um Digitale Souveränität verbunden. Ganz konkrete, oft genannte Argumente für Open Source in der Verwaltung umfassen:

- geringere Abhängigkeit von einzelnen Technologieherstellern
- mehr Interoperabilität zwischen IT-Lösungen durch offene Standards¹

- regionale Wirtschaftsförderung durch die einfachere Einbindung lokaler und mittelständischer Dienstleister
- leichtere Nachnutzbarkeit von Software im föderalen System
- höhere Sicherheit

In den Diskussionen um FOSS für die Verwaltung scheinen jedoch häufig Grundfragen dazu durch, wie FOSS organisatorisch funktioniert: Wäre der Staat damit auf das wechselhafte Engagement ehrenamtlicher Entwickler:innen angewiesen? Ist der Staat den internen Dynamiken von FOSS-Communities ausgeliefert? Gibt es beauftragbare Dienstleister, die Behörden bei Installation und Wartung von FOSS unterstützen? Wer bezahlt das eigentlich alles?

An diese ersten Fragestellungen schließen sich Problematiken zur Umsetzung an: Wie entsteht ein robuster Markt kommerzieller FOSS-Dienstleister? Wie interagieren Staat und FOSS-Community? Wie vermeiden wir Wildwuchs von hunderten kommunalen Varianten des gleichen FOSS-Projektes? Wer kümmert sich um die für den Einsatz in der Verwaltung notwendigen Zertifizierungen?

Das hier entworfene FOSSGov-Ökosystem bietet mögliche Antworten auf derartige Fragen. Dabei führt die Skizze vorwiegend bestehende Lösungsideen zusammen und schafft so mehr Verständnis für die grundlegende Funktionsweise der Open-Source-Wirtschaft. Wir haben dabei jedoch ausdrücklich keine Rücksicht auf bestehende oder geplante Strukturen wie das Zentrum für Digitale Souveränität oder das Code Repository für die Verwaltung genommen. Die Skizze soll keine fertige Lösung liefern, sondern Anlass und Grundlage für vertiefende Überlegungen bieten. Erfolgreich ist sie, wenn sie zu Folgefragen, begründetem Widerspruch und besseren Alternativvorschlägen anregt.

¹ FOSS steht im engen Zusammenhang mit offenen Standards, ist jedoch nicht zwangsläufig damit gleichzusetzen. Oft etablieren sich De-facto-Standards aber durch die große Verbreitung und Zugänglichkeit einer FOSS.

1.1 Zur Entstehung der FOSSGov-Skizze

Die Skizze ist aus einem normativen, gestaltenden Forschungsansatz entwickelt, beantwortet also die Frage: »Wie sollen wir handeln?«. Auf Basis aktueller Studien und ergänzender eigener Erhebungen zu den organisatorischen Fragen im Zusammenhang mit FOSS und insbesondere FOSS für die Verwaltung haben wir Anforderungen abgeleitet, die das Ökosystem erfüllen muss, positive und negative Erfahrungen gesammelt, die es zu vermeiden oder nachzubilden gilt, wirkende Anreizstrukturen nachgezeichnet sowie vorgeschlagene und existierende Organisationsstrukturen identifiziert. Einleitend wurde eine offene Onlineumfrage in der Open-Source-Community und unter Verwaltungsfachleuten bezüglich Hürden, Kompetenzen und Lösungsansätzen des Einsatzes und der Entwicklung von FOSS in der öffentlichen Verwaltung durchgeführt. Im Zeitraum zwischen Januar und Mai 2021 haben wir diese ersten Erkenntnisse in neun Interviews mit Expert:innen aus Verwaltung, Forschung, Wirtschaft und Ehrenamt konsolidiert und anschließend mit Gestaltungsideen angereichert.

1.2 Anforderungen

Durch diese Methodik haben wir zehn **Anforderungen** identifiziert: Das zu entwerfende Ökosystem muss plausibel zeigen, wie

- Entwicklung und Pflege der FOSS-Projekte finanziert werden;
- der Staat seine Interessen in die FOSS-Projekte einbringen kann;
- die FOSS-Projekte selbst ausreichend staatsfern bleiben;
- ehrenamtliches Engagement in den FOSS-Projekten gewahrt bleibt;
- neue FOSS-Projekte für spezifische staatliche Bedarfe entstehen;
- Wissensmanagement zu FOSS in der Verwaltung stattfindet;
- staatliche Entwicklungsbedarfe zu FOSS-Projekten gebündelt werden;
- der Einsatz von FOSS in der Verwaltung die regionale IT-Wirtschaft fördert;
- FOSS zertifiziert wird, beispielsweise um IT-Sicherheitsanforderungen zu erfüllen;
- Raum für die vielen Varianten von FOSS-Projekten, Dienstleistenden und staatlichen Akteuren entsteht.

Die Anreizstrukturen für die beteiligten Akteur:innen spielen eine zentrale Rolle. Diese Strukturen fördern eine selbstverstärkende Innovationsspirale, bei der Entwickler:innen und Nutzer:innen aus Eigeninteresse zur Weiterentwicklung des FOSS-Projektes beitragen, wodurch dieses nützlicher wird und noch mehr beitragswillige Entwickler:innen und Nutzer:innen findet.

1.3 Prinzipien

Losgelöst von der konkreten Skizze haben wir aus dem Entwicklungsprozess sechs **Prinzipien für den Gestaltungsprozess** für ein FOSS-Ökosystem für die öffentliche Verwaltung abgeleitet:

- Der Staat muss in der Interaktion mit FOSS von einer Käufer- zu einer Gärtnermentalität finden – eine Einstellung, bei der langfristiges Handeln und Investieren ein Ökosystem gedeihen lassen.
- Die Verwaltung muss sich gegenüber FOSS-Projekten intern koordinieren, um deren Organisationsstrukturen nicht zu überfordern.
- Staatliche Finanzierung von FOSS-Projekten darf nicht zu einem staatlichen Kontrollanspruch führen.
- Es braucht Mittler zwischen Verwaltung und FOSS-Community, die beide Sprachen sprechen.
- Die Vorteile der dezentralen Organisation von FOSS-Projekten gilt es zu wahren und zu nutzen.
- Forks, also vom ursprünglichen FOSS-Projekt losgelöste Weiterentwicklungen, sind zu vermeiden.

Aus diesen Anforderungen und Gestaltungsprinzipien heraus haben wir das FOSSGov-Ökosystem entworfen und mithilfe kritischer Fragen und konkreter Anwendungsfälle verfeinert. Für den ersten Entwurf des Ökosystems haben wir die zur Erfüllung der Aufgaben und Anforderungen notwendigen Akteursrollen ausdifferenziert und beschrieben. Anschließend wurden die Beziehungen bzw. Aktivitäten der Akteure zueinander in zunehmender Komplexität abgebildet und in jedem Schritt kritisch hinterfragt. Zum Abschluss haben wir die Umsetzung typischer Anwendungsszenarien im Ökosystem durchgespielt und die Skizze optimiert.



2. Grundlagen für Open-Source-Ökosysteme

In diesem Abschnitt skizzieren wir die Grundlagen von Open-Source-Ökosystemen und beantworten zentrale und definitive Fragen: Was meinen wir mit FOSS? Aus welchen Rollen besteht ein FOSS-Ökosystem mindestens? Welche Anreizstrukturen herrschen rund um ein FOSS-Projekt? Und wie wird ein FOSS-Projekt erfolgreich?

2.1 Was ist Freie und Open Source Software (FOSS)?

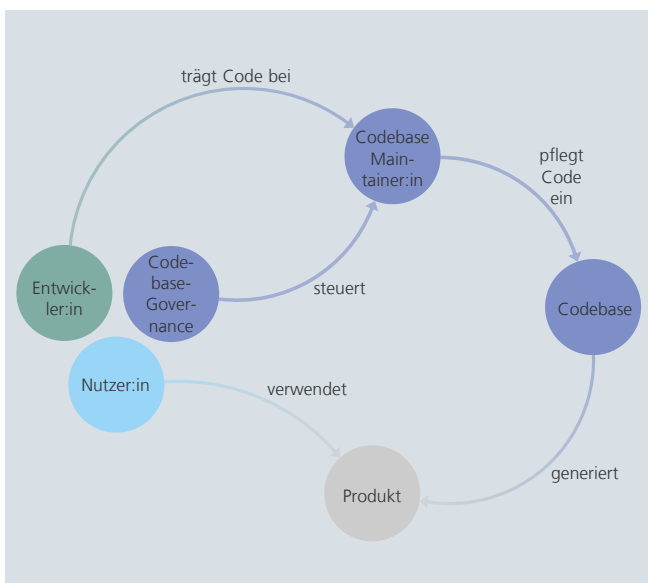
Software ist frei und »Open Source«, wenn ihr Programmcode offen einsehbar ist und vier Freiheiten² garantiert sind:

- Verwenden: Freie Software darf für jeden Zweck genutzt werden und ist frei von Einschränkungen wie dem Ablauf einer Lizenz oder willkürlichen geografischen Beschränkungen.
- Verstehen: Freie-Software-Code darf ohne Vertraulichkeitsvereinbarungen oder ähnliche Einschränkungen von allen untersucht werden.
- Verbreiten: Freie Software darf praktisch kostenfrei kopiert und weitergegeben werden.
- Verbessern: Freie Software darf beliebig modifiziert und angepasst werden. Verbesserungen dürfen weitergegeben werden.

Nutzungslizenzen, unter denen der Quellcode veröffentlicht wird, sichern diese Freiheiten.

² Siehe: Free Software Foundation Europe. <https://fsfe.org>.

Abbildung 1: Einfaches FOSS-Ökosystem, bestehend aus grundlegenden Rollen, dem Produkt und der zugehörigen Codebase.



2.2 Grundlegende Rollen in Open-Source-Ökosystemen

Ein einfaches FOSS-Ökosystem besteht aus vier Rollen: *Nutzer:innen*, *Entwickler:innen*, *Maintainer:innen* und einer *Codebase-Governance*. Diese scharen sich jeweils um die *Codebase*, also den zentralen Programmcode eines FOSS-Projektes. Diese Konstellation stellt den »Urzustand« eines FOSS-Ökosystems dar. (In kleinen Ein-Personen-Projekten könnten die Rollen der *Governance*, *Maintainerschaft*, *Entwickler:in* und *Nutzer:in* von derselben Person getragen werden.)

Nutzer:innen können FOSS kostenfrei nutzen. Fehlermeldungen und Wünsche für neue Funktionalitäten, die sie in ein zentrales Ticket-System der *Codebase* eingeben, liefern wertvolles Feedback für die *Entwickler:innen*.

Entwickler:innen sind Ehrenamtliche oder bezahlte Dienstleister, die neuen Programmcode zu einem FOSS-Projekt beisteuern, um beispielsweise Fehler zu beheben oder neue Funktionalitäten hinzuzufügen. Über ein zentrales System zur verteilten Versionsverwaltung (z. B. »Git«) melden sie ihre Absicht, bestimmte

Abschnitte des Programmcodes zu bearbeiten. Den veränderten Programmcode reichen sie über das Git bei den *Maintainer:innen* ein, damit diese den neuen Code in die nächste allgemeingültige Version der Software einbauen.

Maintainer:innen können Ehrenamtliche, Unternehmen oder gemeinnützige Organisationen sein. Sie wachen über den Programmcode eines FOSS-Projektes und entscheiden, welche von *Entwickler:innen* gelieferten Codebeiträge integriert werden. Dabei folgen sie den Vorgaben der *Codebase-Governance*.

Die *Codebase-Governance* eines FOSS-Projektes hat das Ziel des Gesamtprojektes im Blick und trifft dementsprechend technische und strategische Entscheidungen, legt dafür etwa die Mission sowie die Regeln und die einzusetzenden technischen Standards fest. Sie kann aus einer Einzelperson, Gruppen besonders aktiver Ehrenamtlicher, einem Unternehmen, welches das FOSS-Projekt vorwiegend vorantreibt, Stakeholder-Konsortien oder gewählten Gremien bestehen.

Codebase bezeichnet alle einem FOSS-Projekt zugehörigen digitalen Objekte: Quelltext- und Konfigurationsdateien, Fehlermeldungen sowie Richtlinien, Anforderungen und Dokumentationen, die in der Regel in separaten *Repositorien* in einem Versionsverwaltungssystem gepflegt werden.³ Die Versionsverwaltung kann auf branchentypischen Plattformen oder auf eigenen Servern betrieben werden. *Codebases* bestehen aus mindestens einem *Repositorium*. *Repositorien* unterschiedlicher Projekte können sich in ihrem Umfang deutlich unterscheiden: von kleineren, verwaltungsnahen Individualentwicklungen bis hin zu hochkomplexen, etablierten Softwareprojekten wie etwa Linux oder LibreOffice. Die *Codebase* ist somit der grundlegende Fokus- und Sammelpunkt einer FOSS-Community.

Komplexe Softwareprojekte verwenden oft zahlreiche bereits vorhandene Komponenten (z. B. Programmbibliotheken). Das Ökosystem würde in diesem Fall in Abhängigkeit zu diesen externen, wiederverwendeten FOSS-Komponenten stehen, wobei diese selbst aber nicht zur eigenen *Codebase* gehören.

2.3 Selbstverstärkende FOSS-Innovationsspirale

Erfolgreiche Open-Source-Projekte weisen eine selbstverstärkende Innovationsspirale auf: Je mehr *Nutzer:innen* eine Open-Source-Software nutzen bzw. je nützlicher die Software für ressourcenstarke Lead User⁴ ist, desto eher werden einige der *Nutzer:innen* zur Weiterentwicklung der Software beitragen. Die so verbesserte Software ist wiederum attraktiver und nützlicher für mehr *Nutzer:innen*, wodurch mehr Akteure zu Verbesserung der Software beitragen und so weiter.

Abbildung 2: Innovationsspirale eines FOSS-Projektes



Nach dieser Logik ist stets eine möglichst breite Basis an beitragswilligen *Nutzer:innen* und insbesondere eine Vielfalt an Lead Usern anzustreben. So bleibt die Innovationsspirale auch dann stabil, wenn die einzelnen *Nutzer:innen* nur hin und wieder einen Beitrag zum FOSS-Projekt leisten. Analog wächst dabei meist der Kreis der *Entwickler:innen*, die mit der *Codebase* vertraut sind, und somit das potenzielle Angebot an Dienstleistungen rund um das FOSS-Projekt.

³ Ein Repositorium ist ein digitales Verzeichnis, in der Regel eingebettet in einem System zur Versionsverwaltung (z. B. Git), das neben den jeweiligen Objekten auch Metadaten wie die FOSS-Lizenz, Änderungsprotokolle oder Abhängigkeitsinformationen speichert.

⁴ Lead User sind Nutzer:innen mit starkem, meist kommerziellem Bedarf an einer Software. Sie identifizieren optimalerweise bereits frühzeitig Bedürfnisse für ihren jeweiligen Markt.

2.4 Anreizstrukturen in FOSS-Ökosystemen

Damit eine Innovationsspirale dauerhaft funktioniert, sind die Anreizstrukturen für Akteure entscheidend.

Im einfachsten Fall verbessern individuelle *Entwickler:innen* die *FOSS-Codebase*, weil sie diese selbst nutzen möchten. Viele weitere *Nutzer:innen* nutzen die Software, ohne etwas zu ihrer Entwicklung beizutragen, weil ihnen dazu das Wissen oder die Ressourcen fehlen. Für Lead User mag es aber individuell vorteilhaft sein, selbst oder mittels beauftragter *Entwickler:innen* Fehler zu beheben und Funktionalitäten beizutragen. Diese Lead User sind bspw. Unternehmen, die durch die bessere Software absehbar mehr Geld verdienen, als die Entwicklung der Verbesserungen kostet. Auch der Staat kann als Lead User Interesse an der Pflege eines FOSS-Projektes haben, wenn er dieses langfristig für seine Zwecke nutzen möchte. Unverzichtbar ist das Engagement von Lead Usern bei Software, die so komplex oder anwendungsspezifisch ist, dass sich ein individuelles Engagement für einzelne *Entwickler:innen* nicht mehr lohnt.

Bei erfolgreichen FOSS-Projekten tragen daher *Nutzer:innen*, die besonders abhängig von der Verfügbarkeit und Weiterentwicklung einer Software sind, aus Eigeninteresse auch zur Grundfinanzierung des Projektes bei. Kosten fallen etwa für Infrastruktur, *Maintainer:innen*, Softwaretests und die *Codebase-Governance* an. Hier entsteht eine sogenannte »Tragik der digitalen Allmende«: Für jeden einzelnen *Nutzenden* der Software wäre es der gewinnreichste Weg, selbst nichts zu der Grundfinanzierung beizutragen, während andere diese kritischen Funktionen bereitstellen. Agieren jedoch alle *Nutzenden* nach diesem Kalkül, brechen *Maintainerschaft* und *Governance* – und somit das gesamte FOSS-Projekt – zum Nachteil aller zusammen.

Akteure mit spezieller Anreizlage sind Unternehmen, die Dienstleistungen rund um die FOSS-Lösung herum anbieten, etwa Beratung, Entwicklung, Betrieb, Installation, Support, Wartung und vieles mehr. Für sie ist es schon aus Gewinninteresse essenziell, die FOSS-Lösung zu verbessern und ihr Ökosystem zu unterstützen. Deshalb schaffen sie oft weitere Anreizstrukturen für die Weiterentwicklung einer *Codebase* durch individuelle *Entwickler:innen*, z.B. organisierte Hackathons und Bug-Bounty-Programme.

In vielen FOSS-Projekten werden Entwicklung und Gemeinaufgaben aus Altruismus und Spaß ehrenamtlich erbracht. Da dies aber stets fragile Anreizstrukturen sind, sollte der Staat darauf für kritische Aufgaben nicht bauen.



3. Anforderungen an das FOSSGov-Ökosystem

Ein einfaches FOSS-Ökosystem kann als erfolgreich gelten, solange die *Codebase* weiterentwickelt wird und sie *Nutzer:innen* findet. Ein Open-Source-Ökosystem für die öffentliche Verwaltung hingegen muss darüber hinaus noch weitere Aufgaben und Anforderungen erfüllen. Diese haben wir aus Literatur und Expert:inneninterviews gezogen, um sie bei der Gestaltung des FOSSGov-Ökosystems zugrunde legen zu können.

Zu jeder Anforderung haben wir in einem zweiten Absatz auch beschrieben, wie sie in der FOSSGov-Skizze adressiert ist. Dieser kann zunächst übersprungen werden.

3.1 FOSS finanzieren

Die öffentliche Hand muss einen Beitrag zu von ihr genutzten FOSS-Codebases leisten, wenn diese langfristig zur Verfügung stehen und besser werden sollen. Dies können bezahlte Programmieraufträge für inhaltliche Beiträge zur *Codebase* oder auch Veranstaltungen für die Community sein. Wichtiger für das Ökosystem ist aber die Grundfinanzierung von *Governance* und *Maintainerschaft* der *Codebases*, um die »Tragik der digitalen Allmende« abzuwenden.

Im skizzierten FOSSGov-Ökosystem ist der *FOSSGov-Fonds* das zentrale Vehikel für staatliche Beiträge zur Grundfinanzierung, zu dem alle Körperschaften beitragen können. Die verfügbare Summe wird nach Nutzungsbreite und strategischer Bedeutung der *Codebases* für die Verwaltung aufgeschlüsselt ausgeschüttet. Indirekte Finanzflüsse zugunsten der *Codebases* sind beispielsweise bezahlte Programmieraufträge, um inhaltliche

Beiträge zur *Codebase* zu entwickeln, Eingaben zu *Bug-Bounty-Programmen* und die zentralen Aktivitäten eines *Kompetenzzentrums GovOSPO*⁵ für die Pflege der Communities rund um durch die Verwaltung genutzte *Codebases*.

3.2 Interessen der öffentlichen Verwaltung vertreten

Die öffentliche Verwaltung muss über etablierte Strukturen und Prozesse ihre Interessen in der *Codebase-Governance* vertreten können. Dabei geht es beispielsweise um besondere Anforderungen der Verwaltung an die *Codebase* (siehe Abschnitt 3.9 »Software zertifizieren«), aber auch darum, die langfristige Entwicklung der *Codebase* weg von den Bedarfen der Verwaltung zu vermeiden.

In der *Codebase-Governance* vertritt jeweils der *Codebase-spezifische »Tisch«* im *Kompetenzzentrum GovOSPO* die gebündelten Interessen der deutschen Verwaltung. Wichtige Vorarbeit hierzu leistet das durch *GovOSPO* für jede genutzte *Codebase* beauftragte *Anforderungsmanagement*, das Erfahrungen, Wünsche und Probleme zur *Codebase* aus der gesamten deutschen Verwaltung sammelt.

⁵ Unternehmen richten Open Source Program Offices (OSPO) ein, um ihre Beziehungen zu den Open-Source-Ökosystemen zu verwalten, von denen sie abhängig sind. Durch das Verständnis des Open-Source-Ökosystems des Unternehmens ist ein OSPO in der Lage, die Investitionsrendite des Unternehmens zu maximieren und die Risiken der Nutzung, Mitwirkung und Freigabe von Open-Source-Software zu verringern.

3.3 Ausreichende Staatsferne der Codebases

Öffentlich genutzte *Codebases* sollten nicht unter direktem Einfluss des Staates stehen, um private Akteure und öffentliche Stellen anderer Staaten nicht von der Mitarbeit abzuschrecken. Akteure und Prozesse des öffentlichen Sektors an der Schnittstelle zu *FOSS-Codebases* sollten daher eine gewisse Staatsferne an den Tag legen, also beispielsweise nicht in direkter Weiskette zu einem Ministerium stehen.

In der Skizze ist dies teilweise durch das *Kompetenzzentrum GovOSPO* erfüllt. Dieses ist zwar klar eine staatliche Einrichtung, sollte aber durch Rechtsform und Mandat eher seiner langfristigen Mission denn tagespolitisch motivierten Weisungen aus den Ministerien verpflichtet sein. Hierzu trägt auch der *Aufsichtsrat* des *Kompetenzzentrums GovOSPO* mit Vertreter:innen der *Codebase-Governance* von für die Verwaltung bedeutsamen *Codebases* bei. Bei der *Codebase-Maintainerschaft* überträgt die eher staatsnahe *öffentliche Code-Treuhand* gezielt die Betreuung an externe *Stewardship-Dienstleister*, wenn die *Codebases* absehbar von substantiellem Interesse für Akteure jenseits der deutschen Verwaltung sind. So soll auch hier nach und nach ausreichende Staatsferne hergestellt werden.

3.4 Gemeinnützigkeit erhalten

Die staatliche Mitwirkung an *FOSS-Codebases* ist so zu gestalten, dass sie ehrenamtliche Mitarbeit erhält und nicht durch privatwirtschaftliche Dienstleister oder staatliche Stellen verdrängt. Das Ehrenamt ist ein wichtiger Quell von Innovation und Fachkräften und stellt in der *Codebase-Governance* eine unabhängige Stimme für die Gemeinnützigkeit des FOSS-Projektes dar.

Im FOSSGov-Ökosystem ist dem *Kompetenzzentrum GovOSPO* deshalb ein Auftrag zur *Community-Pflege* zugebracht, um das Ehrenamt aktiv zu fördern. Die Interessenvertretung der deutschen Verwaltung durch das *GovOSPO* in der *Codebase-Governance* soll zudem zugunsten des Ehrenamtes agieren. Auch die Nutzung von *Bug-Bounty-Programmen* kann das Ehrenamt stärken, da es gelegentliche finanzielle Anreize bietet, um sich mit einer *Codebase* vertraut zu machen.

3.5 Neue Codebases für staatliche Bedarfe schaffen

Das Ökosystem muss zeigen, wie aus ungedeckten öffentlichen IT-Bedarfen funktionierende *FOSS-Codebases* werden.

Die Verwaltung kann Entwicklungsaufträge an kommerzielle *Dienstleister* vergeben, deren Ergebnisse anschließend als *FOSS-Codebase* veröffentlicht werden müssen. Ob hieraus eine funktionierende Community entsteht, welche die *Codebase* weiterentwickelt, ist jedoch offen. Wir haben deshalb in die Skizze den *FOSSGov-Inkubator* aufgenommen, der systematisch aus Bedarfen der öffentlichen Hand FOSS-Start-ups entwickelt. Mit einem Start-up besteht zumindest ein initialer Treiber und professioneller Dienstleister für die *Codebase*.

3.6 Wissen zu FOSS in der Verwaltung managen

Um Fehlentscheidungen, Ineffizienzen und Parallelentwicklungen bei der Nutzung und Entwicklung von *FOSS-Codebases* durch öffentliche Stellen zu vermeiden, braucht es zentral zugängliche Informationen zu von der Verwaltung genutzten *Codebases* sowie zu diese *Codebases* betreffenden Erfahrungen und Plänen.

In der Skizze bündelt das *FOSSGov-Verzeichnis* Erfahrungen und Wissen zu den genutzten *Codebases*. Derartige Verzeichnisse wollen aber gepflegt werden. Dabei kommt den Mitarbeiter:innen des beauftragten *Anforderungsmanagements* auch als professionellen Wissenssammler:innen eine Schlüsselrolle zu: Sie gehen aktiv auf *öffentliche IT-Dienstleister* und nutzende *Behörden* in ganz Deutschland zu. Der je *Codebase* zuständige »Tisch« im *Kompetenzzentrum GovOSPO* kontrolliert die Qualität des *Verzeichnisses* in Bezug auf die jeweilige *Codebase*. Das *Anforderungsmanagement* und das *FOSSGov-Verzeichnis* legen zudem Bedarfe und Entwicklungsvorhaben offen, sodass sich Verwaltungen zusammenschließen und Parallelentwicklungen vermeiden können.

3.7 Entwicklungsbedarfe bündeln

Bedarfe öffentlicher Stellen gegenüber *FOSS-Codebases* sind zu bündeln. Dies vermeidet einerseits, dass die *Codebase-Governance* beispielsweise mit den Featurewünschen von tausenden Kommunen geflutet werden. Andererseits soll dies auch das parallele Beauftragen ähnlicher Funktionalitäten vermeiden.

Im FOSSGov-Ökosystem soll vor allem das *Anforderungsmanagement je Codebase* im Zusammenspiel mit dem *FOSSGov-Verzeichnis* für Ordnung sorgen. Schlüssel dazu sind nicht Weisungsrechte, sondern Informationen. Langfristige Bedarfe sammelt das *Anforderungsmanagement* und gibt diese an die *Codebase-Governance* weiter, damit die Bedarfe Teil der Entwicklungs-Roadmap der *Codebase* werden. Bei kurzfristigeren oder spezifischen Bedarfen, welche die *öffentlichen IT-Dienstleister* über Entwicklungsaufträge decken, wirken das *Anforderungsmanagement* und das *FOSSGov-Verzeichnis* koordinierend: Statt separate Aufträge für ähnliche Lösungen zu vergeben, melden Auftraggeberkonsortien Entwicklungsvorhaben, denen sich *öffentliche IT-Dienstleister* dann anschließen können. Hier können Auftraggeber ihre zusätzlichen Anforderungen einspeisen, wenn sie sich finanziell beteiligen. Weiterhin ist eine Unterscheidung zwischen generellen und verwaltungsspezifischen Bedarfen seitens des *Anforderungsmanagements* wünschenswert, um die Entwicklungsvorhaben besser zu priorisieren.

3.8 Regionale IT-Wirtschaft fördern

Eines der oft genannten Versprechen von Open-Source-Software ist mehr Wertschöpfung bei regionalen Entwicklungs- und Serviceunternehmen. Das Ökosystem sollte so gestaltet sein, dass eine Förderung der regionalen Wirtschaft berücksichtigt wird.

In der Skizze werden daher nicht nur Entwicklung und Service der Software an Dienstleister vergeben, sondern auch Rollen wie das *Anforderungsmanagement je Codebase* oder *Stewardship-Dienstleistungen* für international attraktive *Codebases* regelmäßig an regionale Dienstleister ausgeschrieben. Die meisten der Rollen können dabei sowohl durch gewinnorientierte Unternehmen als auch gemeinnützige Organisationen erfüllt werden. Zudem unterstützt der *FOSSGov-Inkubator* Innovationen und Unternehmensgründungen in der nationalen FOSS-Branche.

3.9 Software zertifizieren

Es muss ersichtlich sein, wie im Ökosystem komplette *Codebases*, einzelne Releases sowie Instanzen für die besonderen Anforderungen der öffentlichen Verwaltung zertifiziert werden können. Dabei geht es etwa um IT-Sicherheit, Datenschutz, Barrierefreiheit und Ähnliches.

In der Skizze fällt vorwiegend den *öffentlichen IT-Dienstleistern* die Rolle zu, Zertifizierungen für Versionen der *Codebase* anzustreben. Ist eine Softwareversion erfolgreich zertifiziert, wird

dies via *Anforderungsmanagement* in der *Codebase* hinterlegt. Informationsträgerin könnte z.B. eine sogenannte *publiccode.yml*-Datei sein, in der direkt in der *Codebase* alle für die deutsche Verwaltung wichtigen Begleitinformationen hinterlegt sind. Die italienische Verwaltung praktiziert dieses Verfahren bereits und setzt dabei auf die Auszeichnungssprache YAML.

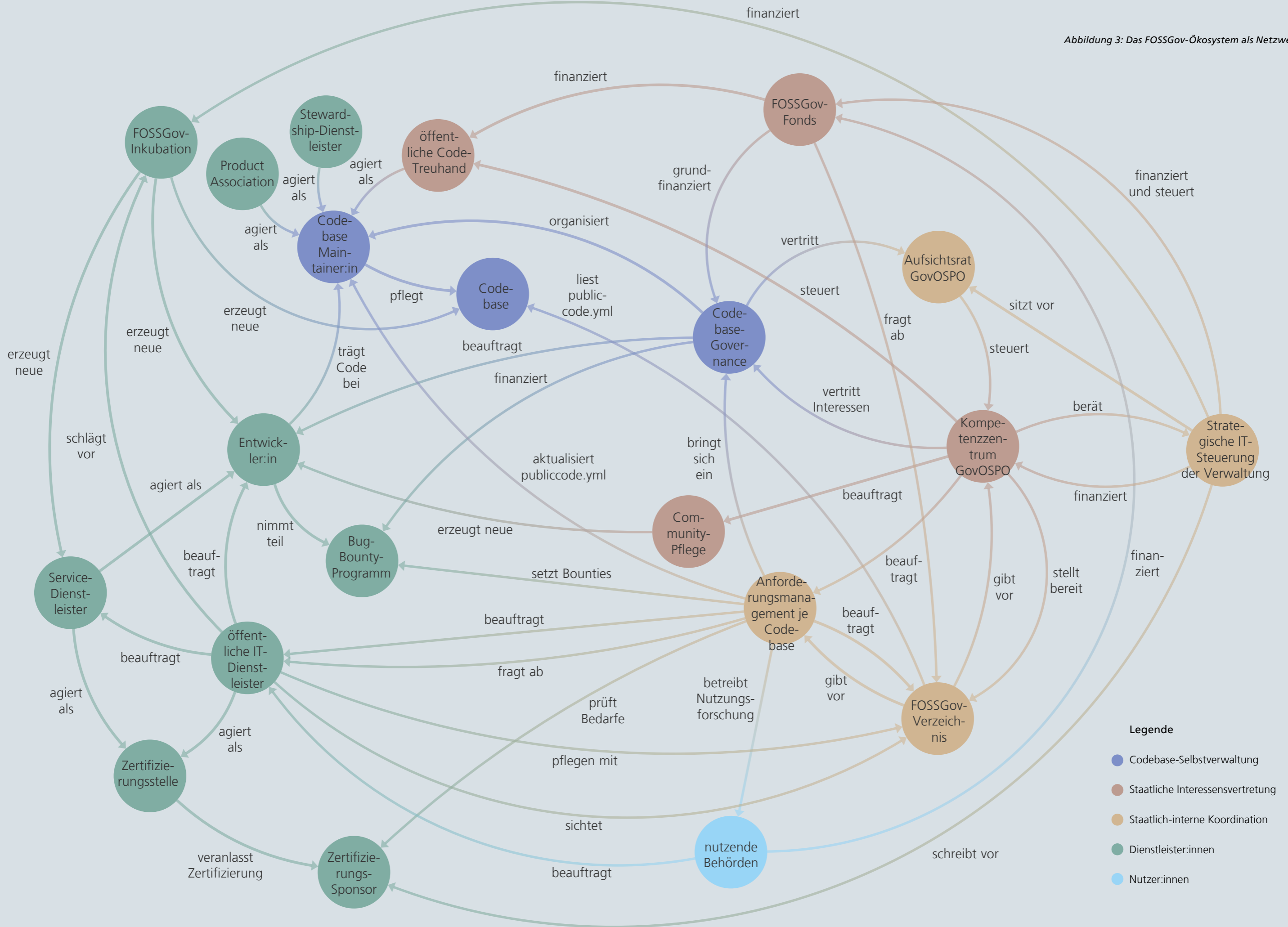
Große Releasewechsel der *Codebase* müssen eventuell neu zertifiziert werden. Wir gehen davon aus, dass *öffentliche IT-Dienstleister* die Neu-Zertifizierung aus eigenem Antrieb veranlassen, um bei ihren Kunden die neue Programmversion einsetzen zu können. Das *Anforderungsmanagement* prüft zudem proaktiv, ob zusätzliche Zertifizierungen eine *Codebase* für weitere Kreise von Nutzenden in der öffentlichen Verwaltung attraktiv machen könnten. Ist dies der Fall, beauftragt das *Anforderungsmanagement* einen *öffentlichen IT-Dienstleister*, die Zertifizierung zu veranlassen.

3.10 Offen für verschiedene Lösungsvarianten sein

Das Ökosystem soll grundsätzlich zeigen, wie mit verschiedenen Varianten umgegangen werden kann. Damit sind beispielsweise verschiedene Konstellationen von *Behörden* und *Dienstleistern*, ehrenamtlichen und gewinnorientierten *Entwickler:innen*, Standard- und Spezialsoftware, und vieles mehr gemeint.

In der Skizze führen daher meist mehrere Wege zum Ziel. So können sowohl das *Anforderungsmanagement je Codebase* als auch *öffentliche IT-Dienstleister* direkt Informationen ins *FOSSGov-Verzeichnis* beitragen. Um Probleme mit der *Codebase* zu beheben, können *Entwickler:innen* beauftragt, Kopfgelder im *Bug-Bounty-Programm* ausgesetzt oder die Probleme via *Codebase-Maintainer:innen* in die allgemeine Entwicklungs-Roadmap aufgenommen werden. Dies erhöht zwar die Komplexität der Skizze, spiegelt so aber realistischer die Handlungsmöglichkeiten in der deutschen Verwaltungs-IT.

Abbildung 3: Das FOSSGov-Ökosystem als Netzwerk



- Legende**
- Codebase-Selbstverwaltung
 - Staatliche Interessensvertretung
 - Staatlich-interne Koordination
 - Dienstleister:innen
 - Nutzer:innen



4. Das FOSSGov-Ökosystem in Aktion: Anwendungsszenarien

Das skizzierte Ökosystem erschließt sich am einfachsten in Aktion. Deshalb spielen wir hier für einige Aufgabenstellungen durch, wie die Rollenträger des FOSSGov-Ökosystems mit diesen umgehen. Entlang von Anwendungsszenarien haben wir auch die Praktikabilität unserer Skizze geprüft.

Für die meisten Aufgabenstellungen, welche sich in den Anwendungsszenarien widerspiegeln, gibt es mehrere mögliche Lösungspfade durch das Ökosystem. Im Sinne der Übersichtlichkeit zeichnen wir jeweils nur einen dieser Pfade nach. Alternative Pfade lassen sich aber leicht selbst durchschreiten.

4.1 Nutzer:innen haben IT-Bedarf, für den es bereits eine durch andere Verwaltungen genutzte FOSS-Codebase gibt

Eine Kommune oder eine *nutzende Behörde* möchte eine Tätigkeit digitalisieren, die sie bisher analog durchgeführt hat. Andere Ämter in Deutschland nutzen bereits eine geeignete FOSS-Lösung.

- Eine potenziell *nutzende Behörde* meldet einen neuen IT-Bedarf an ihren *öffentlichen IT-Dienstleister*.
- Der *öffentliche IT-Dienstleister* sichtet das *FOSSGov-Verzeichnis* auf bereits durch die Verwaltung genutzte FOSS-Lösungen und wird fündig.
- Der *öffentliche IT-Dienstleister* nimmt die Installation und Einführung der Software bei der zukünftig *nutzenden Behörde* vor oder beauftragt dafür einen mit der *Codebase* vertrauten *Service-Dienstleister*.

Abbildung 4: Beteiligte Akteure des Anwendungsszenarios 4.1 im FOSSGov-Ökosystem

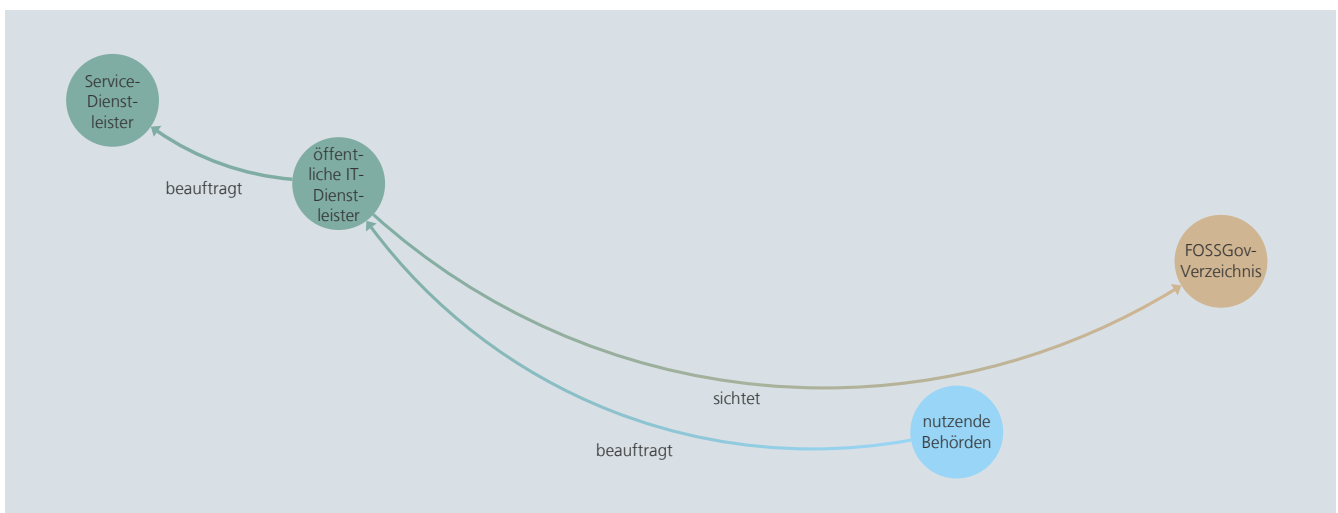
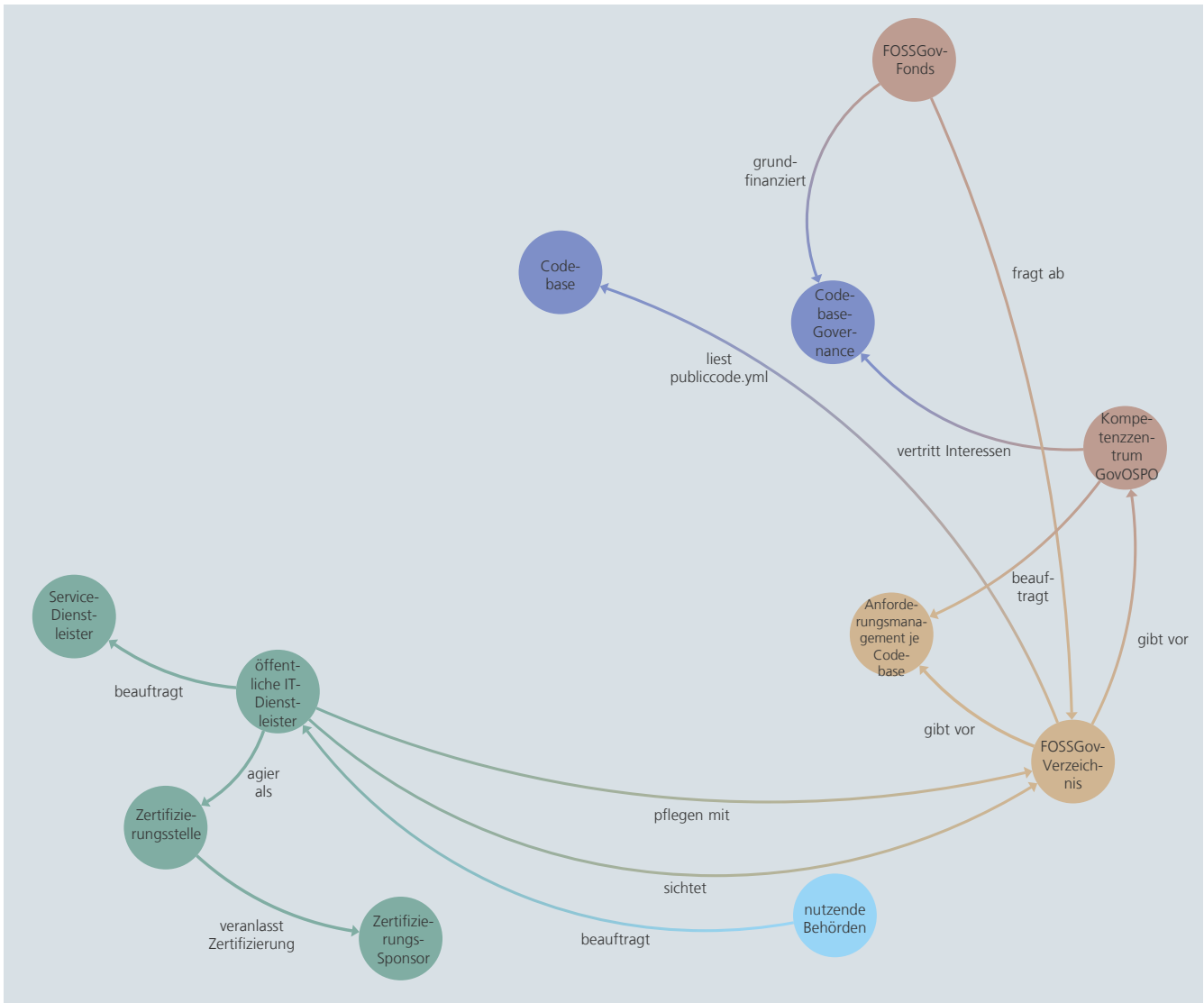


Abbildung 5: Beteiligte Akteure des Anwendungsszenarios 4.2 im FOSSGov-Ökosystem



4.2 Nutzer:innen haben IT-Bedarf, für den es bereits eine FOSS-Codebase gibt

Eine Kommune oder eine *nutzende Behörde* möchte eine Tätigkeit digitalisieren, die sie bisher analog durchgeführt hat. Es gibt eine passende *FOSS-Codebase*, welche im *FOSSGov-Verzeichnis* gelistet, jedoch in der deutschen Verwaltung bisher noch nicht im Einsatz ist.

- Eine potenziell *nutzende Behörde* meldet einen neuen IT-Bedarf an ihren *öffentlichen IT-Dienstleister*.
- Der *öffentliche IT-Dienstleister* sichtet das *FOSSGov-Verzeichnis* auf bereits durch die Verwaltung genutzte FOSS-Lösungen. Er wird nicht fündig.
- Der *öffentliche IT-Dienstleister* findet eine zum Bedarf passende *Codebase*.
- Der *öffentliche IT-Dienstleister* agiert als *Zertifizierungssponsor* und veranlasst nötige Zertifizierungen, beispielsweise zur IT-Sicherheit.
- Der *öffentliche IT-Dienstleister* beauftragt einen mit der *Codebase* vertrauten *Service-Dienstleister* mit der Installation und Einführung der Software bei der zukünftig *nutzenden Behörde*.
- Parallel zur Zertifizierung meldet der *öffentliche IT-Dienstleister* die *Codebase*, ihren Verwendungszweck und weitere Details zum *FOSSGov-Verzeichnis*.
- Das *FOSSGov-Verzeichnis* legt in der *Codebase* eine *publiccode*-Datei mit den relevanten Informationen für die deutsche Verwaltung an, u. a. zu den getätigten Zertifizierungen.
- Aufgrund des neuen Eintrags im *FOSSGov-Verzeichnis* schafft das *Kompetenzzentrum GovOSPO* einen für die *Codebase* zuständigen »Tisch«. Dieser existiert zunächst aufgrund der geringen Bedeutung der *Codebase* nur als zusätzlicher Stellenanteil eine:r Mitarbeiter:in.

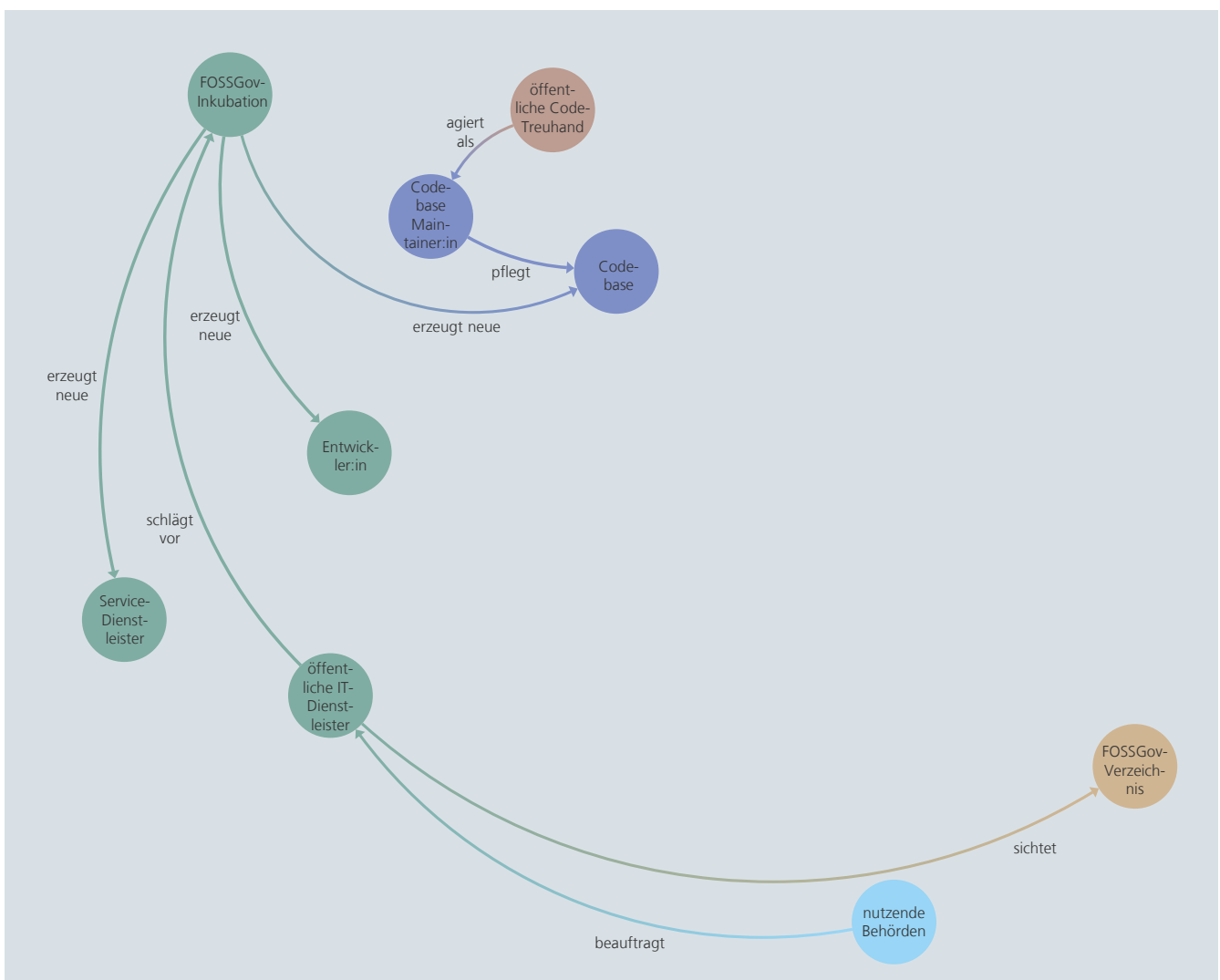
- GovOSPO beauftragt einen Dienstleister, das *Anforderungsmanagement* für die neue *Codebase* zu übernehmen. Dieses existiert zunächst aufgrund der geringen Bedeutung nur als weiteres Element eines Sammelauftrags.
- Der *FOSSGov-Fonds* fragt neue Einträge im *FOSSGov-Verzeichnis* ab. Die neue *Codebase* wird in die Grundfinanzierung aufgenommen. Damit beteiligt sich der öffentliche Geldgeber zu einem kleinen Teil an der Finanzierung der *Maintainerschaft*, der *Codebase-Governance* und des *Codebase-Betriebes*.
- Der zuständige »Tisch« im *Kompetenzzentrum GovOSPO* erkundet die *Codebase-Governance* der neuen *Codebase* und beginnt hier etwaige Interessen der deutschen Verwaltung einzubringen.

4.3 Nutzer:innen haben IT-Bedarf, für den es noch keine FOSS-Codebase gibt

Eine Kommune oder eine *nutzende Behörde* möchte eine Tätigkeit digitalisieren, die sie bisher analog durchgeführt hat. Es gibt dafür keine passende *FOSS-Codebase*. Proprietäre Angebote entsprechen nicht den Kriterien der Verwaltung. Das Anliegen ist nicht dringend.

- Eine *Behörde* meldet einen neuen, aber nicht dringenden IT-Bedarf an ihren *öffentlichen IT-Dienstleister*.
- Der *öffentliche IT-Dienstleister* sichtet das *FOSSGov-Verzeichnis* bereits durch die Verwaltung genutzter FOSS-Lösungen. Er wird nicht fündig.
- Der *öffentliche IT-Dienstleister* findet keine zum Bedarf passende *Codebase*. Proprietäre Angebote erfüllen nicht die Kriterien.

Abbildung 6: Beteiligte Akteure des Anwendungsszenarios 4.3 im FOSSGov-Ökosystem

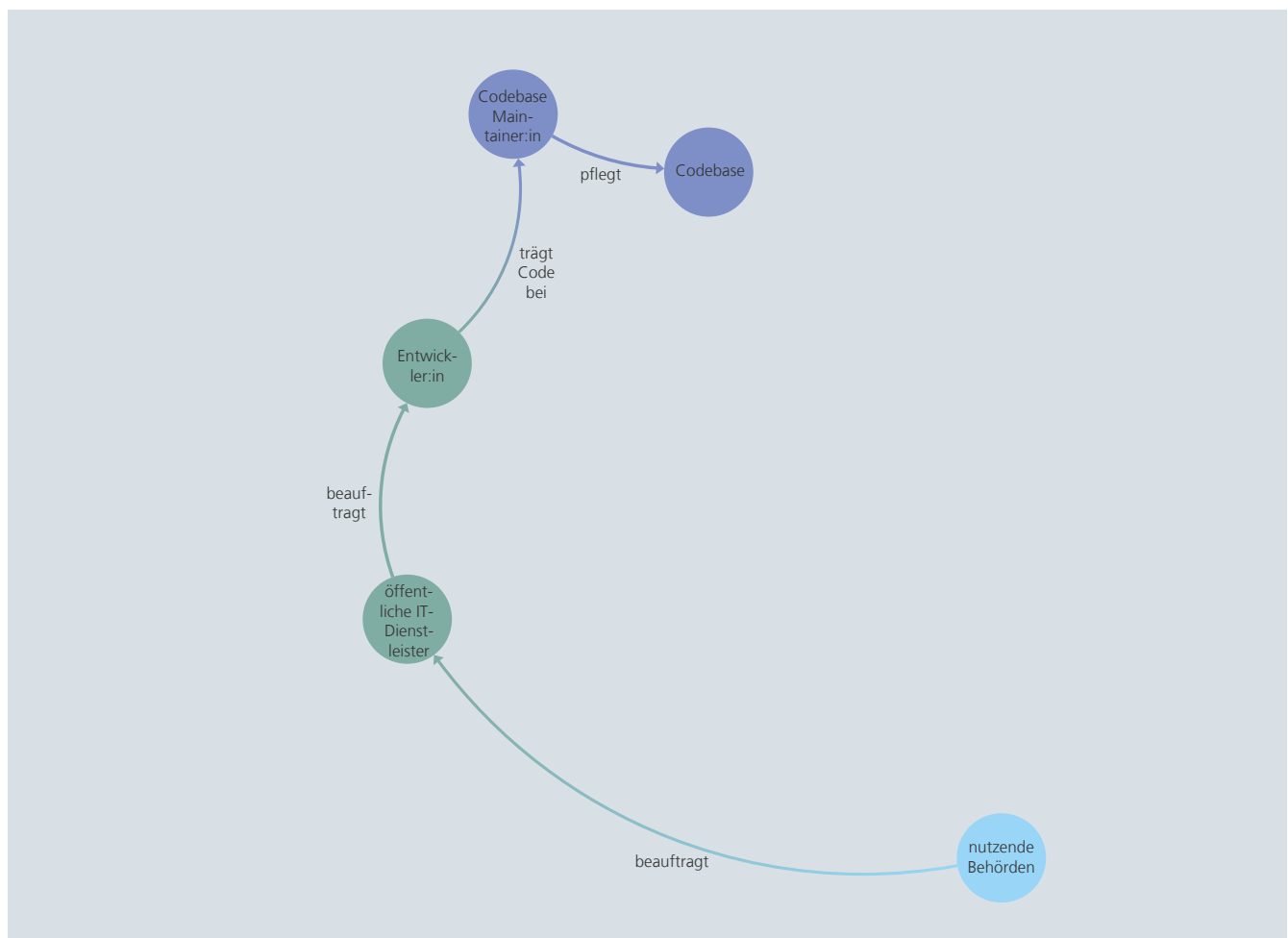


- Der *öffentliche IT-Dienstleister* reicht den Bedarf als »Challenge« beim *FOSSGov-Inkubator* ein.
- Ein Gründer:innenteam entwickelt im Rahmen des *FOSSGov-Inkubators* eine neue *Codebase*, die den Bedarf der Behörde deckt.
- Als *Codebase-Maintainerin* fungiert zunächst die *öffentliche Code-Treuhand*.
- Das Gründer:innenteam gründet sich als Unternehmen aus und agiert sowohl als *Entwickler:in* als auch als *Service-Dienstleister* für die *Codebase*.
- Der *öffentliche IT-Dienstleister* beabsichtigt, die neue *Codebase* zu betreiben und beginnt mit etwaigen Zertifizierungen sowie der Aufnahme der *Codebase* ins *FOSSGov-Verzeichnis* (siehe Abschnitt 4.2, ab Unterpunkt 4).
- Wegen Veränderungen an den Arbeitsabläufen brauchen einige Fachämter neue Funktionalitäten an einer Software, die sie schon lange einsetzen. Sie teilen ihren Bedarf ihrem *IT-Dienstleister* mit.
- Der *öffentliche IT-Dienstleister* stellt fest, dass er den Bedarf nicht aus existierenden Modulen der *Codebase* decken kann. Er vergibt einen Entwicklungsauftrag an mit der *Codebase* erfahrene *Entwickler:innen*.
- Die beauftragten *Entwickler:innen* arbeiten in Absprache mit den *Maintainer:innen* der *Codebase* und tragen den Programmcode der neuen Funktionalitäten zur *Codebase* bei.
- Die *Codebase-Maintainer:innen* akzeptieren den Code-Beitrag der beauftragten *Entwickler:innen* und integrieren diesen in die nächste Version der Software.
- Die neue Funktion steht als Teil der *Codebase* zur Verfügung.

4.4 Nutzer:innen brauchen eine neue Funktion für eine bereits genutzte FOSS-Codebase

Ein Fachamt nutzt seit vielen Jahren eine FOSS-Lösung. Wegen Änderungen am Arbeitsprozess braucht das Fachamt nun aber neue Funktionen in der Software.

Abbildung 7: Beteiligte Akteure des Anwendungsszenarios 4.4 im FOSSGov-Ökosystem



4.5 Öffentlicher IT-Dienstleister bemerkt einen Bug in der Software

Einem *öffentlichen IT-Dienstleister* fällt ein Fehler in der Software auf, der nicht dringend und verwaltungsspezifisch ist. Er meldet dies an das *Anforderungsmanagement je Codebase*. Dieses gibt den Auftrag an das *Bug-Bounty-Programm* weiter.

- Ein *öffentlicher IT-Dienstleister* bemerkt einen Fehler im Programmcode. Der Fehler ist nicht kritisch und verwaltungsspezifisch.
- Der *öffentliche IT-Dienstleister* meldet den Fehler an das *Anforderungsmanagement der Codebase*.
- Das *Anforderungsmanagement* setzt im *Bug-Bounty-Programm* ein Kopfgeld auf das Beheben des Fehlers aus, da in der Roadmap verwaltungsspezifische Fehler voraussichtlich niedrig priorisiert würden.
- *Entwickler:innen* nehmen am *Bug-Bounty-Programm* teil. Sie beheben den Fehler und erhalten das Kopfgeld.
- Die *Entwickler:innen* reichen den Code bei den *Codebase-Maintainer:innen* ein.
- Die *Codebase-Maintainer:innen* pflegen den Code in die nächste Version der Software ein.
- Der Fehler ist behoben und steht als neue Softwareversion zur Verfügung. Ab hier gilt das Vorgehen bei veränderten *Codebases* (Abschnitt 4.6).

Abbildung 8: Beteiligte Akteure des Anwendungsszenarios 4.5 im FOSSGov-Ökosystem

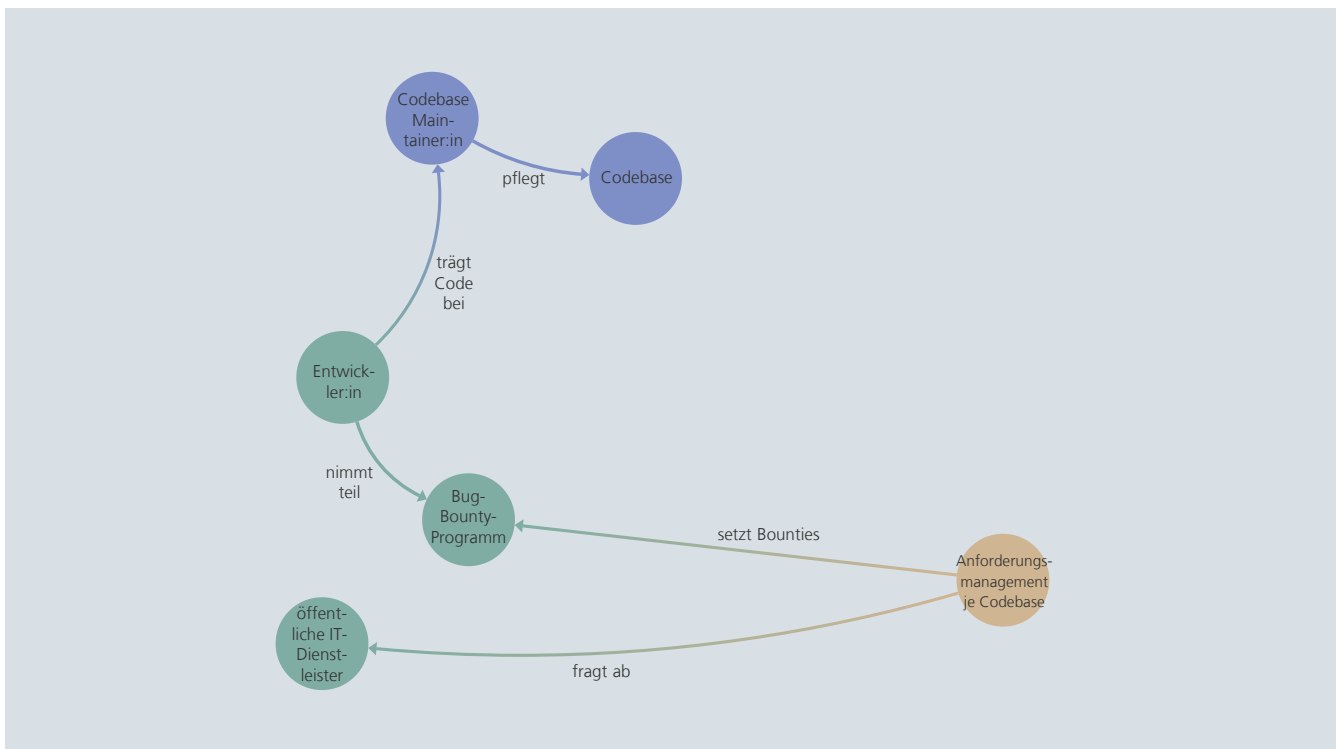
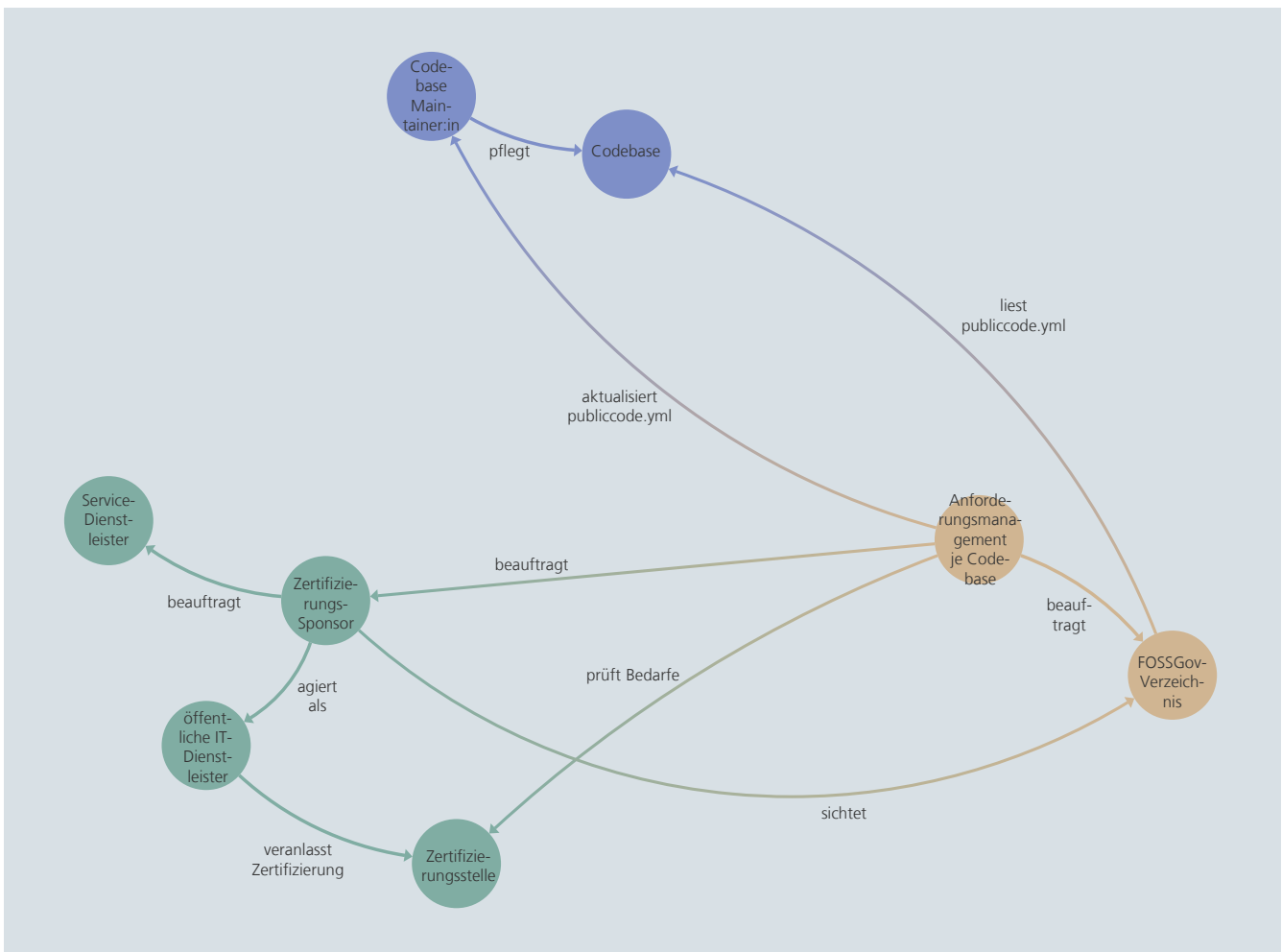


Abbildung 9: Beteiligte Akteure des Anwendungsszenarios 4.6 im FOSSGov-Ökosystem



4.6 FOSS-Codebase veröffentlicht neue Software-Version

Eine durch die öffentliche Verwaltung bereits genutzte FOSS-Codebase veröffentlicht ihre nächste, in einigen Komponenten von Grund auf neu programmierte Version. Um sie in der öffentlichen Verwaltung einzusetzen, braucht die Codebase zunächst ein Zertifikat des Bundesamtes für Sicherheit in der Informationstechnik (BSI).

- Eine durch die öffentliche Verwaltung bereits genutzte FOSS-Codebase veröffentlicht ihre nächste, in einigen Komponenten von Grund auf neu programmierte Version.
- Das FOSSGov-Verzeichnis liest automatisch das Codebase-Repository aus erfährt so von der neuen Version.

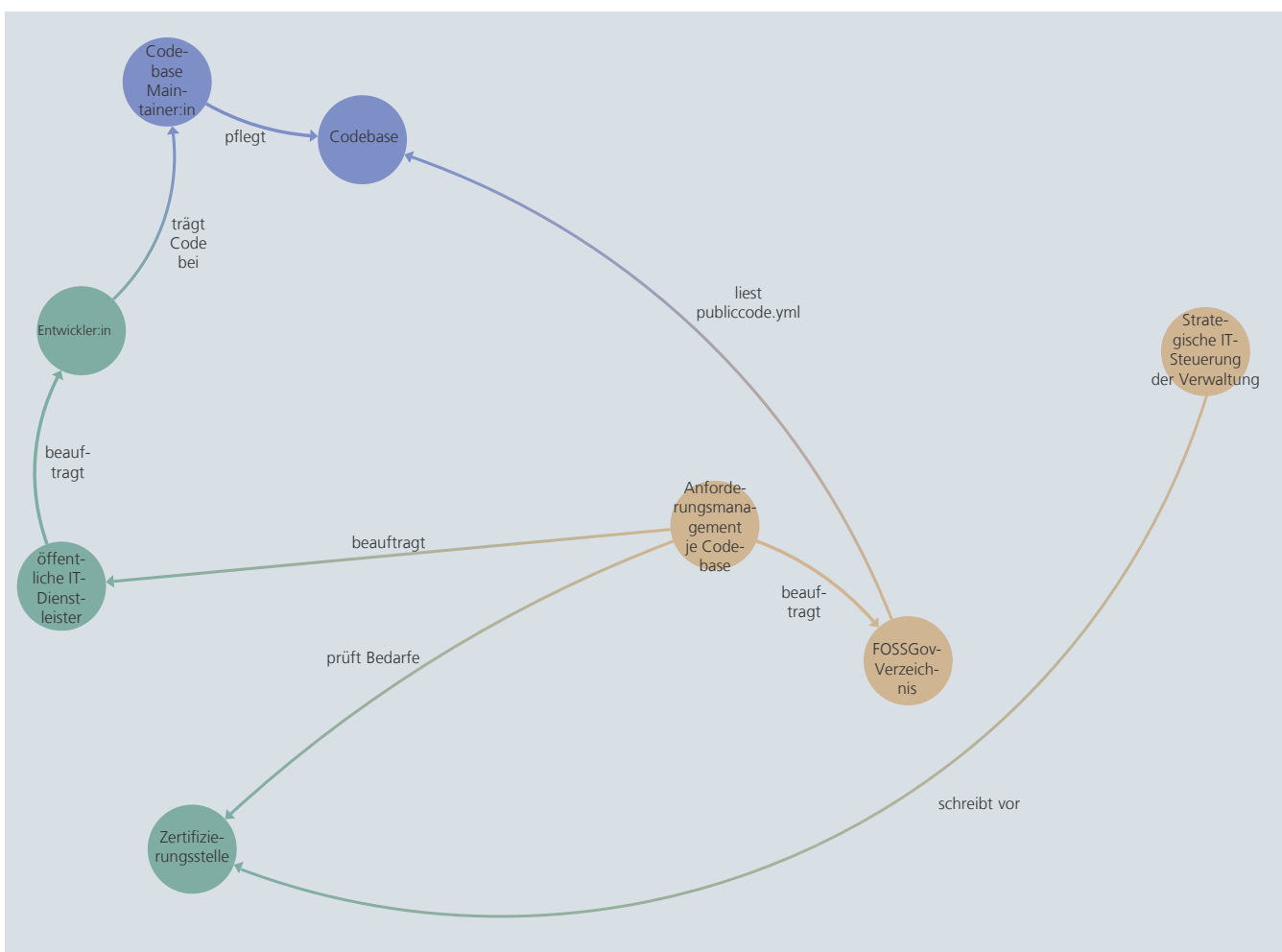
- Das Anforderungsmanagement bewertet, ob die Änderungen an der Codebase eine Neu-Zertifizierung notwendig machen. Ergebnis: Ein neues BSI-Zertifikat ist nötig.
- Das Anforderungsmanagement beauftragt einen mit der Codebase erfahrenen öffentlichen IT-Dienstleister, die Zertifizierung zu veranlassen.
- Das Anforderungsmanagement trägt die neuen Zertifikate in die publiccode-Datei der Codebase ein und übermittelt diese an die Codebase-Maintainer:innen. Somit ist die neue Software-Version für die deutsche öffentliche Verwaltung nutzbar.
- Öffentliche IT-Dienstleister erfahren über das FOSSGov-Verzeichnis von der Einsatzfähigkeit der neuen Version.
- Öffentliche IT-Dienstleister veranlassen das Ausrollen der neuen Software-Version bei den nutzenden Behörden oder beauftragen Service-Dienstleister.

4.7 Strategische IT-Steuerung der Verwaltung verschärft IT-Sicherheitsrichtlinien

Die *IT-Steuerung der deutschen Verwaltung* erlässt neue, strengere IT-Sicherheitsrichtlinien. Eine bereits genutzte *FOSS-Codebase* entspricht den neuen Anforderungen nicht.

- Die *IT-Steuerung der deutschen Verwaltung* erlässt neue, strengere IT-Sicherheitsrichtlinien.
- Das *Anforderungsmanagement* der *Codebase* prüft, ob durch die strengeren Richtlinien Änderungen an der *Codebase* nötig werden, damit diese weiterhin die notwendigen Zertifikate erhält.
- Das *Anforderungsmanagement* gibt an das *FOSSGov-Verzeichnis* weiter, dass die *Codebase* aktuell nicht die neuen Sicherheitsanforderungen erfüllt.
- Das *Anforderungsmanagement* beauftragt einen *öffentlichen IT-Dienstleister*, die notwendigen Änderungen an der *Codebase* vorzunehmen.

Abbildung 10: Beteiligte Akteure des Anwendungsszenarios 4.7 im FOSSGov-Ökosystem



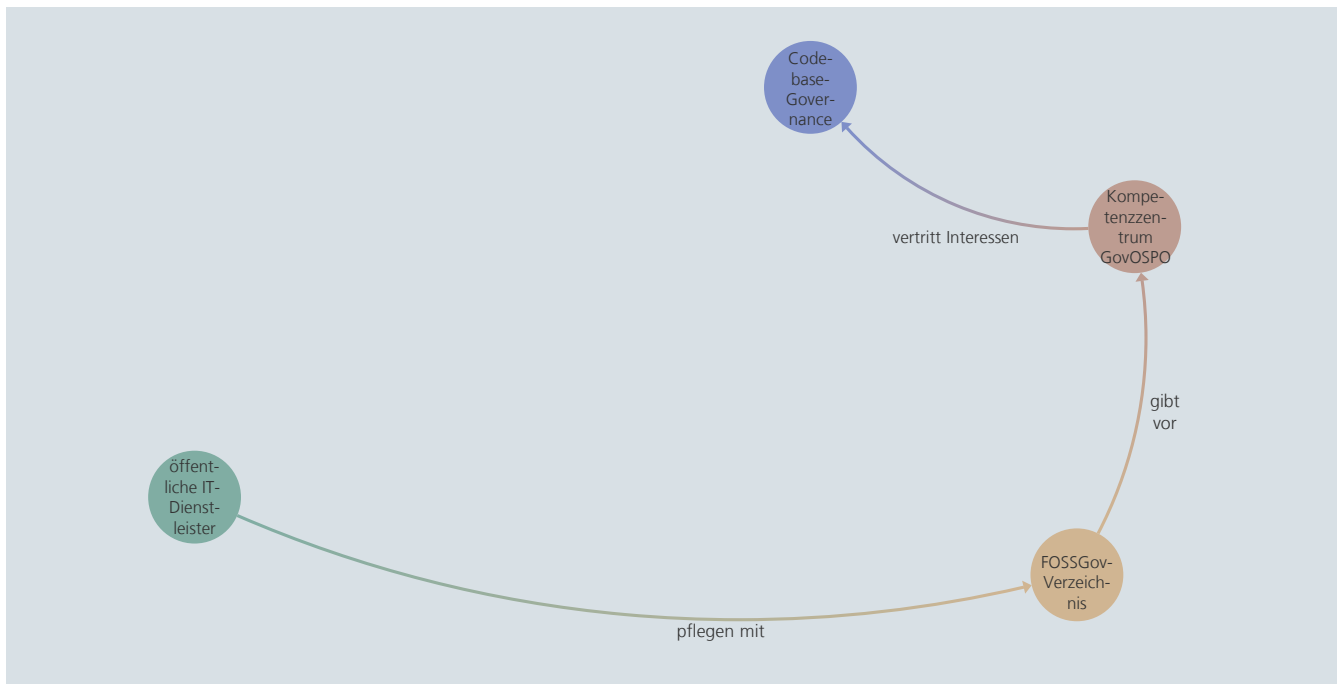


Abbildung 11: Beteiligte Akteure des Anwendungsszenarios 4.8 im FOSSGov-Ökosystem

4.8 Deutsche Verwaltung nimmt Einfluss auf Codebase-Architektur

Damit eine FOSS-Software für deutsche Verwaltung in der Fläche nutzbar wird, müssen Unterschiede im lokalen Recht und der Rechtsumsetzung leicht in den Programmen abgebildet werden können. Solche Unterschiede können sich in Parametern wie etwa unterschiedlichen Formularfeldern, Bezeichnungen, Verteilungsschlüsseln oder Steuersätzen äußern. Hierzu muss die Architektur der *Codebase* weiterentwickelt werden.

- Ein *öffentlicher IT-Dienstleister* identifiziert eine *FOSS-Codebase*, die er gerne einsetzen möchte, die aber nicht ausreichend parametrisierbar ist.
- Der *öffentliche IT-Dienstleister* meldet sein Interesse an der *Codebase* und den Anpassungsbedarf an das *FOSSGov-Verzeichnis*. Eventuell schließen sich daraufhin weitere IT-Dienstleister dem Interesse an.
- Das *Kompetenzzentrum GovOSPO* nimmt den Eintrag im *FOSSGov-Verzeichnis* zur Kenntnis und stuft das Anliegen als strategisch wichtig ein.
- Das *Kompetenzzentrum GovOSPO* tritt in Kontakt mit der *Codebase-Governance*, kommuniziert den Anpassungsbedarf und stellt eine anschließende Lead-User-Rolle der deutschen Verwaltung in der *Codebase* in Aussicht.
- Bei positiver Übereinkunft mit der *Codebase-Governance* erfolgen die Architektur-Anpassungen beispielsweise wie in Abschnitt 4.4 (»Nutzer:innen brauchen eine neue Funktion für eine bereits genutzte FOSS-Codebase«) beschrieben.

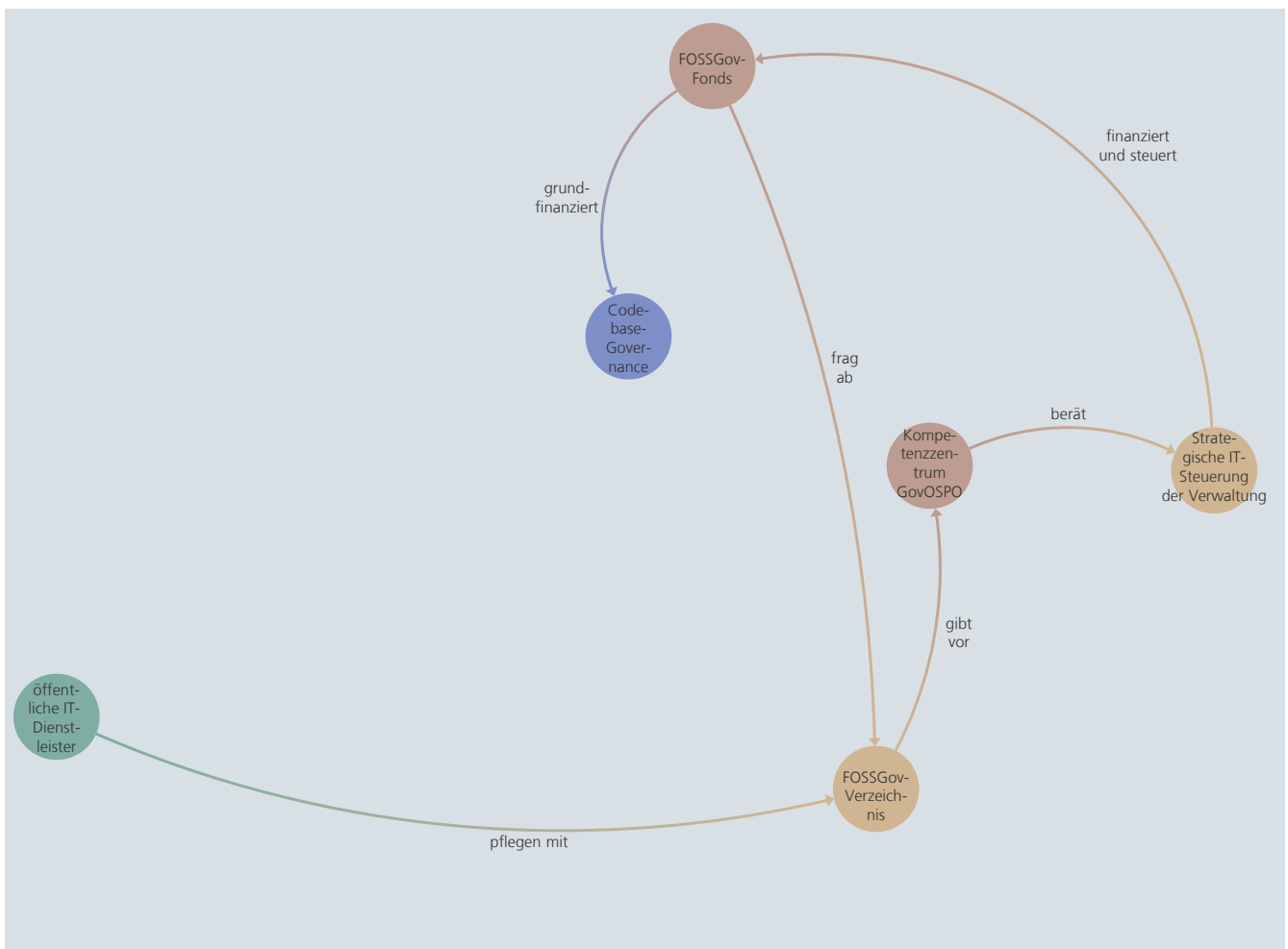
4.9 Grundfinanzierung staatlich genutzter Codebases

Deutsche *Behörden* setzen eine FOSS-Anwendung vermehrt ein. Die Verfügbarkeit der Software ist durch die quantitativ höhere Abhängigkeit kritischer, der *FOSSGov-Fonds* passt entsprechend die Finanzierung der *Codebase* an.

- *Öffentliche IT-Dienstleister* setzen eine FOSS-Lösung vermehrt ein.
- Das *FOSSGov-Verzeichnis* vermerkt die zunehmende Nutzung.

- Der *FOSSGov-Fonds* fragt u. a. turnusmäßig die Nutzungszahlen über das *FOSSGov-Verzeichnis* ab, um das Budget der FOSS-Projekte jeweils strategisch zu bewerten und ggf. anzupassen. Der *FOSSGov-Fonds* wird hierbei vom *GovOSPO* beraten.
- Der *FOSSGov-Fonds* steigert die anteilige Finanzierung der Codebase-Gemeinkosten.

Abbildung 12: Beteiligte Akteure des Anwendungsszenarios 4.9 im FOSSGov-Ökosystem





5. Gestaltungsprinzipien für ein FOSS-Ökosystem für die öffentliche Verwaltung

Im Entwurfsprozess eines FOSS-Ökosystems für die öffentliche Verwaltung haben wir durch eine qualitative Erhebung (siehe Abschnitt 1.1) sechs Prinzipien herausgearbeitet, die unabhängig von der Detailausgestaltung gelten und beachtet werden sollten.

5.1 Von der Nehmer- zur Gärtnermentalität wandeln

Damit FOSS für die öffentliche Verwaltung nachhaltig funktioniert, sollte sich die öffentliche Hand nicht nur an *Codebases* bedienen, sondern auch möglichst viel zurückgeben.

FOSS-Codebases als ein »Buffet kostenloser Lösungen« zu sehen, ist nicht falsch. Ein Gutteil des gesellschaftlichen Mehrwertes von *FOSS-Codebases* entsteht, weil die Software ohne Gegenleistung genutzt wird. Bei digitalen Gütern droht dabei auch keine Tragik der Allmende durch Übernutzung, weil der gleiche Programmcode beliebig oft wiederverwendet werden kann.

Allerdings setzt der Kauf von Lizenzen proprietärer Software ein Marktsignal und für die Anbieter den Anreiz, diese weiterzuentwickeln, um in Zukunft noch mehr Lizenzen verkaufen zu können. Die Installation einer FOSS-Anwendung allein schafft nicht in allen FOSS-Projekten einen solchen Anreiz, ergänzend braucht es aktive Beiträge.

Wie in einem öffentlichen Blumengarten, in dem Besucher:innen gerne die Blüten betrachten, aber niemand die Pflanzen wässert, können *FOSS-Codebases* trotz massenhafter Nutzung stagnieren. Darum braucht es in einem FOSS-Ökosystem eine Gärtnermentalität: Wann immer ich die Möglichkeit habe, sollte ich etwas zur *Codebase* zurückgeben – gerne auch gezielt, um von mir gewünschte Entwicklungen zu fördern. Programmcode beitragen, Grundfinanzierung beisteuern, Hackathons veranstalten – der Möglichkeiten gibt es viele und alle leisten ihren Teil.

5.2 Koordination der Verwaltungsseite

Damit die Vielstimmigkeit der deutschen Behördenlandschaft nicht zu Chaos bei der FOSS-Community führt, braucht es eine Koordination aufseiten der Verwaltung. So sollte eine zentrale Stelle die Bedarfe und strategischen Interessen der *Behörden* gegenüber der *Codebase* vorweg bündeln, ausgleichen und mit einer Stimme kommunizieren. Ohne diese Bündelung droht eine Überlastung der *Codebase-Governance*, die sich plötzlich tausenden deutschen Fachämtern gegenübersehen. Auch für den Erfahrungsaustausch innerhalb der Verwaltung über den Einsatz verschiedener FOSS-Produkte ist eine Koordination der Kommunikation hilfreich. Zudem sollten FOSS-Entwicklungsvorhaben einzelner *Behörden* oder *öffentlicher IT-Dienstleister* für alle sichtbar und teilnahmeoffen gestaltet werden, um Parallelentwicklungen zu vermeiden und Ressourcen für die nachhaltige Weiterentwicklung und Pflege zu bündeln.

5.3 Finanzierung und Kontrolle entkoppeln

Wer zahlt, bestimmt – dieses gewohnte Prinzip steht einem florierenden FOSS-Ökosystem entgegen. Stattdessen sollte die öffentliche Hand aus ihrer finanziellen Beteiligung an einer *Codebase* keinen umfassenden Kontrollanspruch ableiten, um bewusst Raum für die Beteiligung anderer zu lassen. Für kleine Unternehmen und die ehrenamtliche Entwicklungs-Community ist die Aussicht auf Mitsprache zentral für ihre Entscheidung, zu einer *Codebase* beizutragen oder nicht. Auch für ausländische Behörden ist die Mitarbeit an einer *Codebase*, die ein anderer Staat deutlich dominiert, aus Perspektive ihrer digitalen Souveränität unattraktiv. Dabei ist gerade eine Beteiligung mehrerer Staaten an einer *Codebase* wünschenswert, damit viele Stimmen für Verwaltungszwecke sprechen, viele Ressourcen für die Entwicklung bereitstehen und die *Codebase* robust gegen den Rückzug einzelner Lead User ist.

Wo es zielführend scheint, sollte die deutsche Verwaltung durchaus Einfluss auf die Entwicklungsrichtungen der *Codebase* nehmen. Einerseits, indem sie sich in die *Codebase-Governance* einbringt, wozu sie durch einen Beitrag zu den Gemeinkosten des FOSS-Projektes selbstverständlich legitimiert ist. Andererseits, indem sie Programmieraufträge für neue Funktionalitäten der *Codebase* oder für Bugfixes vergibt. Ein solcher Einfluss sollte aber stets ergänzend zu anderen Stimmen gesehen werden und nicht vollständige Kontrolle zum Ziel haben, um die *Codebase* offen für weiteres Engagement zu halten und so ihre selbstverstärkende Innovationsspirale nicht zu gefährden.

5.4 Mittler zwischen Verwaltung und FOSS-Community schaffen

Zwischen Verwaltung und FOSS-Community braucht es Mittler, die die jeweiligen Arbeitskulturen verstehen und die jeweiligen Sprachen sprechen. Reibungen entstehen hier oft eher aus organisatorischen und inhaltlichen Unterschieden – Hierarchie versus Netzwerk, Recht versus Technik, Pflicht versus Freiwilligkeit – als aus handfesten Zielkonflikten.

5.5 Vorteile von Dezentralität nutzen

Viele der Vorteile von FOSS mit Blick auf digitale Souveränität, Lock-in-Situationen und Resilienz ergeben sich aus einem im Idealfall dezentralen Ökosystem: Die Software selbst ist frei verfügbar. Für Support und Entwicklung stehen viele verschiedene Dienstleister zur Auswahl. Kein einzelner Akteur kann überraschend ein Produkt vom Markt nehmen, wie es im kommerziellen Sektor möglich wäre. Die Verwaltung kann Einfluss auf die Entwicklungs-Roadmap von Codebases nehmen, damit kritische Funktionen nicht überraschend mit dem nächsten Update verschwinden.

Daher gilt es, diese Dezentralität zu wahren oder herzustellen. Gibt es nur ein Unternehmen, das Support- oder Entwicklungskompetenz bereitstellt? Liegt die *Codebase Maintenance* in den Händen eines einzelnen gewinnorientierten Akteurs? Trägt nur ein Akteur zur Entwicklung der *Codebase* bei? Solche monopolartigen Situationen gilt es zu vermeiden und ihnen aktiv gegenzusteuern.

Daher sind auch hybride Open-Source-Geschäftsmodelle eher zu meiden, die auf einer Zweiteilung in eine quelloffene Basisversion und zahlungspflichtigen, geschlossenen Premium-Funktionalitäten beruhen. Der Einsatz der für den professionellen

Betrieb meist unverzichtbaren Premium-Module führt in eine Herstellerabhängigkeit, die Souveränitätszielen entgegensteht. Zudem ist dabei der Wirkungsgrad in Richtung der Innovationsspirale eingeschränkt.

5.6 Forks vermeiden

Die öffentliche Verwaltung sollte eigene Forks von *FOSS-Codebases* nur in Extremfällen vornehmen, um den nachhaltigen Fortbestand einer *Codebase* nicht zu gefährden.

Forks, also Abgabelungen, bezeichnen die Entscheidung von *Entwickler:innen*, eine *Codebase* fortan in zwei separate Richtungen weiterzuentwickeln. Das bekannteste Beispiel ist wohl OpenOffice und LibreOffice, die aus der gleichen *Codebase* entstammen. Forks können viele und auch berechtigte Gründe haben: Unterschiedliche Anwendungsfelder, verschiedene Software-Architekturprinzipien, persönlicher Streit unter *Entwickler:innen*, unterschiedliche Governance-Ansätze.

Als Faustregel gilt es, Forks aber zu vermeiden. Meist teilen sich dabei auch die Stakeholder, etwa die ehrenamtliche Community, die auf die *Codebase* spezialisierten Unternehmen, der Kreis finanziell beitragender *Nutzer:innen*. Dies schwächt die Innovationsspirale einer *Codebase* und kann diese sogar zusammenbrechen lassen. Forks sollten nur aus dringenden technischen Gründen oder bei inkompatiblen strategischen Entscheidungen in der *Codebase-Governance* in Betracht gezogen werden. Die Frage nach einer lebensfähigen Innovationsspirale der neuen Fork ist entscheidend.

Statt mit »Deutschland-Forks« eine vermeintliche Kontrolle und Rechtssicherheit zu erlangen, sollten die besonderen Anforderungen der öffentlichen Hand besser in den Hauptast der *Codebase* hineingetragen oder mit modularen Erweiterungen gelöst werden.



6. Fazit

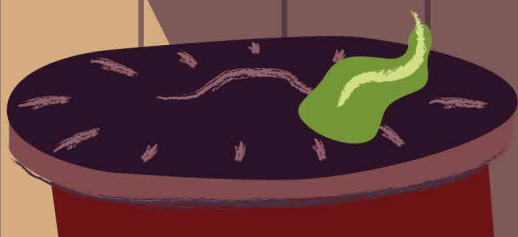
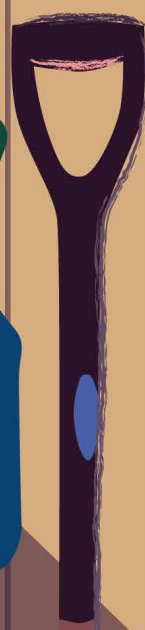
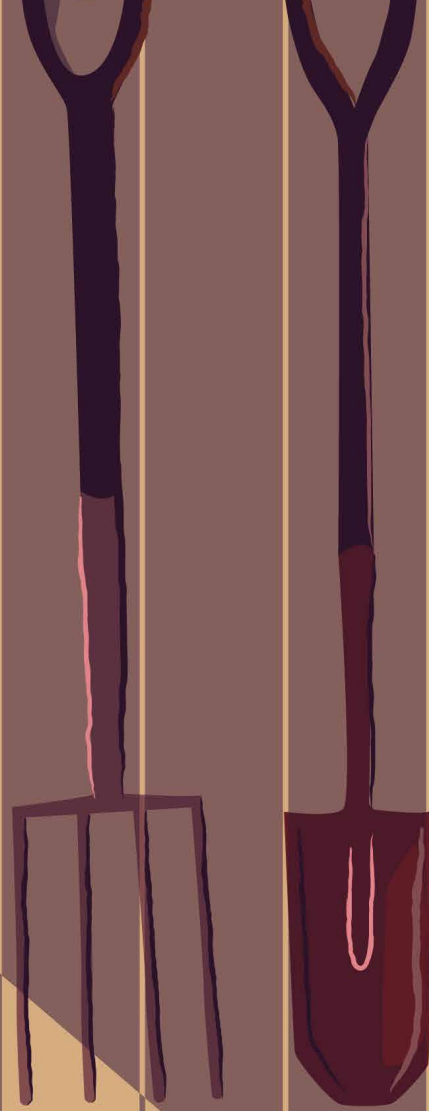
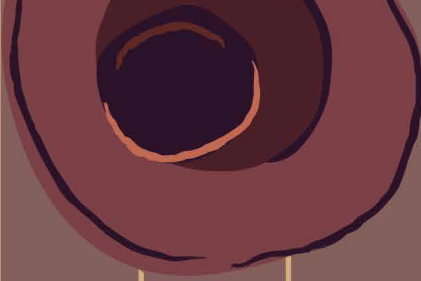
Das FOSSGov-Ökosystem skizziert eine mögliche Antwort auf zentrale Fragen von FOSS für die öffentliche Verwaltung: Wie werden Interessen und Bedarfe der öffentlichen Verwaltung gegenüber *FOSS-Codebases* gebündelt und kommuniziert? Wie nimmt die öffentliche Hand Einfluss auf die strategische Entwicklung der für sie wichtigen *Codebases*? Wie fördert der Staat das Entstehen neuer *Codebases* für seine spezifischen Aufgaben? Wie erhalten FOSS-Lösungen die für den Einsatz in der Verwaltung nötigen Zertifikate? Wie leistet die Verwaltung ihren Beitrag für die Weiterentwicklung der *Codebases*? Wer beauftragt und liefert Dienstleistungen rund um FOSS-Produkte für *Behörden*?

Die Skizze zeichnet dabei das Bild des Staates als zurückhaltenden Gemeinschaftsgärtners, der seinen Beitrag zu großen, wohlgedeihenden Projekten leistet, aber auch gezielt neue *Codebases* heranzüchtet. Dabei ist das Ziel nicht Kontrolle, sondern möglichst viele Mitstreitende für die Pflege und Weiterentwicklung zu gewinnen. Der Staat setzt dabei auf eine Vielfalt an Dienstleistern und Mitwirkenden aus Wirtschaft und Zivilgesellschaft, um einen lebendigen Markt zu schaffen.

Auch wenn die Skizze an einigen Stellen stark vereinfacht: Das FOSSGov-Ökosystem zeigt, dass der Staat mit überschaubaren Neuerungen innerhalb der bestehenden Strukturen eine funktionierende Schnittstelle zur FOSS-Community schaffen kann. Zentral ist das *Kompetenzzentrum GovOSPO* als Anker und Treiber der weiteren staatlichen Strukturen wie des *FOSSGov-Verzeichnisses* als Koordinationsplattform und des *FOSSGov-Fonds* als Finanzierungsmechanismus.

Wiederkehrendes Organisationsprinzip des FOSSGov-Ökosystems sind plattformartige Angebote, etwa im *FOSSGov-Verzeichnis*, dem *FOSSGov-Inkubator* oder der *öffentlichen Code-Treuhand*: Stets geht es darum, für freiwillige Beteiligung offenzustehen und diese in geordnete Bahnen zu lenken. Dieses Prinzip liegt sowohl der Open-Source-Szene als auch Teilen des deutschen Föderalismus zugrunde - durchaus passend für die Schnittstelle von Staat und FOSS.

Die FOSSGov-Skizze soll als ein konkreter Vorschlag die Grundlage für weiterführende Diskussionen bilden. Wir freuen uns auf Kritik und Alternativvorschläge.



7.1 Verzeichnis der Rollen

Anforderungsmanagement je Codebase

Für jede im öffentlichen Sektor genutzte – im *FOSSGov-Verzeichnis* hinterlegte – Codebase schreibt das GovOSPO die Funktion des *Anforderungsmanagements* für eine festgelegte Zeit aus. Dies ist ein natürliches Betätigungsfeld für *öffentliche IT-Dienstleister* und spezialisierte *Stewardship-Dienstleister*, kann aber auch von *Beratungs- oder IT-Serviceunternehmen* besetzt werden.

Aufgabe ist proaktive Nutzungsforschung in der Verwaltung und bei den *IT-Dienstleistern*, um Entwicklungsbedarfe zu aggregieren und zu priorisieren. Hieraus beauftragt das *Anforderungsmanagement* dringende Features oder Bugfixes aus Mitteln des *FOSSGov-Fonds*, weniger dringende Bedarfe trägt es an die *Codebase-Maintainerschaft* heran. So werden ineffiziente Parallelentwicklungen aus der öffentlichen Verwaltung heraus vermieden.

Das Wissen zum Einsatz der *Codebase* im deutschen öffentlichen Sektor bringt das *Anforderungsmanagement* zudem in die *Codebase-Governance* ein, um die Weiterentwicklung der FOSS-Lösung zielgerichtet zu unterstützen.

Das *Anforderungsmanagement* betreibt zudem das Wissensmanagement rund um den Einsatz der *Codebase* in der Verwaltung und trägt somit auch zum *FOSSGov-Verzeichnis* bei. Es dokumentiert und sammelt Implementierungserfahrungen für Nachahmende.

Auch die Prüfung von etwaigen Zertifizierungsanforderungen obliegt dem *Anforderungsmanagement*.

Aufsichtsrat GovOSPO

Der *Aufsichtsrat*, bestehend aus Vertreter:innen relevanter FOSS-Projekte und der *strategischen IT-Steuerung der Verwaltung*, setzt Mandat und strategische Leitlinien des *Kompetenzentrums GovOSPO*.

Bug-Bounty-Programm

Bug Bounties sind Prämien für das Ermitteln, Melden und Reparieren von Fehlern und Sicherheitslücken im Programmcode. Reichen *Entwickler:innen* hier beispielsweise Code ein, der eine gemeldete Sicherheitslücke schließt, erhalten sie einen zuvor festgelegten Betrag. Dies schafft Anreiz zur Pflege der *Codebase* und aktiviert insbesondere die ehrenamtliche Community der Codebase für die Zwecke der Verwaltung.

Die *Codebase-Governance* trägt zum allgemeinen Budget der *Bounty-Programme* bei. Spezifische und dringende Bounties könnten öffentliche IT-Dienstleister (mit-)finanzieren.

Codebase

Codebase bezeichnet alle einem FOSS-Projekt zugehörigen Objekte: Quelltext- und Konfigurationsdateien sowie Richtlinien und Dokumentationen, die in der Regel in separaten Repositorien in einem Versionsverwaltungssystem gepflegt werden. Ein Repository ist ein digitales Verzeichnis, in der Regel eingebettet in einem System zur Versionsverwaltung (z. B. Git), das neben den jeweiligen Objekten auch Metadaten wie die FOSS-Lizenz, Änderungsprotokolle oder Abhängigkeitsinformationen speichert. Folglich bestehen *Codebases* aus mindestens einem Repository, diese können sich in ihrem Umfang jeweils deutlich unterscheiden. Die *Codebase* ist somit die konstituierende Einheit einer FOSS-Community.

Codebase-Governance

Jede *Codebase* hat Governance-Strukturen, die wichtige technische und strategische Entscheidungen fällen. Varianten reichen von Gründer:innen-geführten Modellen über Gremien aus besonders aktiven Beitragenden und gewählten Vertretenden bis hin zu Konsortien von Stakeholdern und Sponsor:innen. Oft teilt sich dies zudem in ein technisches und ein strategisches Entscheidungsorgan auf.

Bei für sie wichtigen *Codebases* sollte die öffentliche Verwaltung sich durch das *GovOSPO* in die Governance einbringen. Ziele sollten sein:

- Klare Mission, Prinzipien und Unabhängigkeit der *Codebase*, um der Verwaltung langfristig Sicherheit zu bieten;
- Beachtung von Standards und Anforderungen, die Voraussetzung für die Nutzung in der Verwaltung sind;
- Klare Entwicklungs-Roadmap, an der sich die öffentliche Verwaltung bei Entwicklungs-Aufträgen orientieren kann.

Insgesamt geht es hier um die Wahrung von Interessen, nicht um die Kontrolle von *Codebases*.

Codebase-Maintainer:in

Maintainer:innen entscheiden, welche Code-Beiträge in die *Codebase* aufgenommen sowie welche Fehlermeldungen und Wünsche bevorzugt bearbeitet werden. Sie koordinieren also die technische Zusammenarbeit am Code.

Diese Rolle können Einzelpersonen, Dienstleister oder spezielle Organisationen übernehmen.

Community-Pflege

Die ehrenamtliche Entwicklungs-Community einer *Codebase* sichert die Unabhängigkeit und die Innovationsfähigkeit der *Codebase*, bildet ein Talentreservoir für *Bug Bounties* rund um eine *Codebase* und trägt wesentlich zur nötigen Masse wachsender Augen bei, die Bugs und Sicherheitslücken entdecken.

Neben der Grundkostenfinanzierung von *Codebase-Maintainerschaft*, *-Governance* und *-Betrieb* und der Beauftragung zur Pflege und Weiterentwicklung der *Codebase* ist daher die Pflege der Community der dritte Beitrag zur nachhaltigen Entwicklung einer *Codebase*.

Hierzu können beispielsweise Meetups, Hackathons, Programmierstipendien oder Konferenzen organisiert bzw. unterstützt werden. Best Practices der Open-Source-Szene sind z. B. das Programmierstipendium »Google Summer of Code« oder die deutsche Konferenz »FOSS-Backstage«.

Effizient ist es, diese Unterstützung über zivilgesellschaftliche Organisationen oder kommerzielle Dienstleister zu lenken, welche ohnehin nah an der Community sind.

Entwickler:in

Entwickler:innen besitzen Expertise für einzelne *Codebases* und programmieren Lösungen wie neue Funktionalitäten, bessere Algorithmen oder intuitivere Bedienoberflächen.

Dabei kann es sich um Unternehmen, Ehrenamtliche oder Mitarbeitende der öffentlichen Hand handeln, die beauftragt oder aus Eigeninitiative tätig werden.

Aus Souveränitätsperspektive ist die Auswahl zwischen vielen verschiedenen Entwicklungsdienstleistern attraktiver als ein Monopolist. Wirtschaftspolitisch interessant ist die erhöhte Chance auf regionale Wertschöpfung und Arbeitsplätze.

Bei Beauftragungen sind Open-Source-Klauseln wichtig, sodass der entwickelte Code unter freien Lizenzen steht.

FOSSGov-Fonds

Der *FOSSGov-Fonds* ist das zentrale Finanzvehikel für die öffentliche Finanzierung von FOSS. Die Gebietskörperschaften statten den Fonds mit einem Globalbudget aus.

Das *Kompetenzzentrum GovOSPO* ermittelt über das *FOSSGov-Verzeichnis* die Nutzungsintensität und die strategische Bedeutung der genutzten *Codebases*. Hiernach gewichtet trägt der *Fonds* zur Grundfinanzierung der *Codebases* bei und stellt Mittel für verwaltungsspezifische *Bug Bounties* der *Codebases* zur Verfügung.

FOSSGov-Inkubator

Der *FOSSGov-Inkubator* versucht, IT-Lösungsbedarfe der öffentlichen Verwaltung in nachhaltige FOSS-Geschäftsmodelle übersetzen zu lassen.

Mit einem offenen Call for Challenges unter den *öffentlichen IT-Dienstleistern* sucht der *Inkubator* Lösungsbedarfe. Eine Jury aus *öffentlichen IT-Dienstleistern*, dem *GovOSPO*, der *staatlichen IT-Steuerung* sowie Vertreter:innen der FOSS-Community wählt aus den Vorschlägen besonders drängende Bedarfe sowie aussichtsreiche Geschäftsmodelle aus. Für die ausgewählten Bedarfe führt der *Inkubator* einen Open-Innovation-Wettbewerb durch, dessen Gewinner:innen eine Förderung erhalten, um ein erstes Produkt und Geschäftsmodell zu entwickeln. Teil der Förderung ist eine Abnahmegarantie durch den bedarfsmeldenden IT-Dienstleister für einen festgelegten Zeitraum.

Erfolgreiche Gründungen ergänzen die *FOSS-Codebases* und den Dienstleister-Markt. Die *öffentliche Code-Treuhand* berät bei der Organisation der *Maintainerschaft* für neue *Codebases* oder übernimmt die *Maintainerschaft* direkt.

Der *Inkubator* sichert sich als Gegenleistung für die Förderung Unternehmensanteile. Diese verkauft er anschließend, um sich teilweise gegenzufinanzieren. Die öffentliche Hand grundfinanziert den *Inkubator*.

Bestehende (verwaltungsnahe) Varianten sind der Prototype Fund und die Münchner Open Source Factory. Darüber hinaus bieten u. a. die Apache Foundation und die Mozilla Foundation eigene erfolgreiche Inkubationsprogramme für diverse FOSS-Projekte an.

FOSSGov-Verzeichnis

Das *FOSSGov-Verzeichnis* listet öffentlich sämtliche durch die deutsche Verwaltung genutzten Codebases mit zusätzlichen Erklärungstexten und Informationen sowie alle nutzenden Verwaltungen und betreuenden *öffentlichen IT-Dienstleister*.

GovOSPO verantwortet das Verzeichnis, die Einträge der einzelnen Codebases pflegt aber vornehmlich das jeweilige *Anforderungsmanagement* mithilfe der *öffentlichen IT-Dienstleister*.

Kritische Informationen, etwa zur lizenzrechtlichen Bewertung oder der BSI-Zertifizierung, liegen in einer Datei in der *Codebase*, unabhängig vom *FOSSGov-Verzeichnis*.

Das Verzeichnis listet zudem Weiterentwicklungsvorhaben und Ideenskizzen zu neuen Codebases, um Parallelentwicklungen zu vermeiden und Mitstreiter:innen zu finden.

Kompetenzzentrum GovOSPO

GovOSPO ist Kompetenzzentrum und Knotenpunkt der öffentlichen Verwaltung zur FOSS-Szene. Funktionen sind unter anderem die Übersetzung zwischen FOSS-Community und Verwaltung, die Koordination innerhalb der Verwaltung und die langfristige strategische Betreuung der *Codebases*. Ebenjene Aufgaben erfüllen Open Source Program Offices (OSPO) in der Wirtschaft.

Als veraltungsebenenübergreifende Einrichtung sollte das GovOSPO im Rahmen des IT-Planungsrates als Anstalt oder Ähnliches organisiert sein.

Das Team versteht sowohl die FOSS-Szene als auch die deutsche Verwaltung, um als Schnittstelle zu wirken. Jede in der Verwaltung genutzte *Codebase* wird durch einen »Tisch« bei GovOSPO betreut, wobei eine Person auch für mehrere kleine und weniger kritische *Codebases* zuständig sein kann. Diese »Tische« steuern auch das *Anforderungsmanagement je Codebase*, kontrollieren die Einträge im *FOSSGov-Verzeichnis*, bringen sich in der *Codebase-Governance* ein und veranlassen Maßnahmen der *Community-Pflege*.

GovOSPO steuert zudem die *öffentliche Code-Treuhand*.

Nutzende Behörden

Behörden sind *Nutzer:innen* von FOSS-Lösungen und sollen keine eigene Expertise benötigen. Diese kann prinzipiell ausschließlich bei den *IT-Dienstleistern* liegen, die von den Nutzenden für Einrichtung und Betrieb ihrer IT-Infrastruktur bezahlt werden.

FOSS sollte für viele Behörden Kostenvorteile bieten, so diese keine speziell angepasste Software benötigen.

Die Nutzenden sind zudem Gegenstand der Nutzungsforschung durch das *Anforderungsmanagement je Codebase*.

Öffentliche Code-Treuhand

Die *öffentliche Code-Treuhand* ist eine eigene Rechtsperson unter der Aufsicht des *Kompetenzzentrums GovOSPO*. Die Treuhand ist der öffentliche *Maintainerschafts-Dienstleister* und maintaint⁶ öffentlich initiierte *Codebases*, die (noch) zu klein für eine eigene *Product Association* oder einen externen *Stewardship-Dienstleister* sind.

Öffentliche IT-Dienstleister

Öffentliche IT-Dienstleister bündeln IT-Kompetenzen und befinden sich in öffentlichem Besitz. Sie betreuen die IT-Landschaften der Behörden. Im Kontext dieses Dokumentes können damit sowohl eigenständige Entitäten, die z. B. mehrere Kommunen betreuen, als auch die interne IT-Abteilung einer großen Behörde gemeint sein.

Im Ökosystem übernehmen sie vereinfacht eine Generalunternehmerrolle gegenüber der Verwaltung: Behörden erhalten schlüsselfertige IT-Lösungen. Alles Weitere liegt in Verantwortung der *IT-Dienstleister*.

Product Association

Große *Codebases* werden oft von eigenen gemeinnützigen Organisationen, etwa Stiftungen, maintaint. Diese betreuen in der Regel nur eine *Codebase*, bzw. eine *Codebase-Familie*.

Eine solche *Product Association* ist z. B. The Document Foundation, die LibreOffice betreut und in der auch deutsche Verwaltungen engagiert sind.

⁶ In diesem Dokument zu verstehen als »betreibt und organisiert die inhaltliche Pflege«.

Service-Dienstleister

Ein lebhafter Markt von *Service-Dienstleistern* integriert, wartet und bietet Support für eingesetzte FOSS-Lösungen an. Zum Teil bieten diese *Dienstleister* auch Entwicklungsleistungen an. Auch die Beantragung von Zertifizierungsprozessen für FOSS-Releases zählt zum Angebot.

Hauptauftraggeber im öffentlichen Sektor sind die *öffentlichen IT-Dienstleister*. So kaufen sie Codebase-basierte Expertise ein, statt diese selbst vorzuhalten. Die zahlreichen und regional aufgestellten *öffentlichen IT-Dienstleister* treffen viele Vergabeentscheidungen, was einen kleinteiligen und lokal spezifischen Markt befördert.

Stewardship-Dienstleister

Externe *Stewardship-Dienstleister* erhalten insbesondere dann das Mandat, eine *Codebase* zu betreuen, wenn diese staatsferner maintaint werden soll. Dies ist beispielsweise bei Software der Fall, die zwar im deutschen *FOSSGov-Inkubator* entstanden ist, aber auch für die Verwaltungen anderer Staaten von Nutzen ist. Da ausländische Verwaltungen eine *Maintainerschaft* durch die deutsche *Code-Treuhand* abschrecken könnte, wechselt das Mandat zu einem der Dienstleister, die meist viele *Codebases* betreuen. Ein Beispiel hierfür ist die niederländische Foundation for Public Code.

Strategische IT-Steuerung der Verwaltung

Die *strategische IT-Steuerung* der Verwaltung fasst im Sinne der Übersichtlichkeit pauschal Verwaltungsstrukturen zusammen, die über Leitlinien, Standards und Finanzen der Verwaltungsdigitalisierung entscheiden. Dies umfasst beispielsweise den IT-Planungsrat, das ITZBund, den IT-Rat der Bundesregierung sowie einschlägige Abteilungen der dort vertretenen Ministerien u. v. m.

Für ein florierendes FOSS-Ökosystem für die öffentliche Verwaltung ist hier eine substantielle Entscheidung für FOSS notwendig, die sich auch politisch, rechtlich, technisch und finanziell ausdrückt.

Zertifizierungs-Sponsor

Die Beantragung einer Zertifizierung wird in der Regel vom herstellenden Unternehmen selbst unternommen. Steht, wie bei Community-Projekten teilweise üblich, keine einheitliche juristische Person zur Verfügung, kann die Beantragung auch von einer dritten Entität übernommen werden. Hierfür können sowohl private *Service-Dienstleister* als auch *öffentliche IT-Dienstleister* infrage kommen.

Zertifizierungsstelle

Für den Einsatz in der öffentlichen Verwaltung muss Software zum Teil speziell sicherheitszertifiziert werden. BSI-Zertifizierungen brauchen einen Antragsteller, den *Zertifizierungs-Sponsor*, was bei *FOSS-Codebases* der erste *öffentliche IT-Dienstleister* übernehmen kann, der die *Codebase* nutzen möchte (bzw. ein von ihm hierfür beauftragter *Dienstleister*).

Datenschutzfolgenabschätzungen und Barrierefreiheitszertifizierungen erfolgen installations- bzw. anwendungsfallbezogen, hier stellt FOSS keinen Sonderfall dar.

7.2 Quellen & Weiterführendes

- Bönig, T. (2021): »CDO Thomas Bönig: Open Source als Strategie mit hohem Potenzial«. <https://muenchen.digital/blog/thomas-boenig-zu-open-source-in-muenchen/>
- de Laat, P. B. (2007): »Governance of open source software: state of the art«. *Journal of Management & Governance* 11, S. 165 – 177 <https://doi.org/10.1007/s10997-007-9022-9>
- Der Beauftragte der Bundesregierung für Informationstechnik (2019): »Organisationskonzept Zentrum für Digitale Souveränität«. https://www.cio.bund.de/SharedDocs/Publikationen/DE/Strategische-Themen/organisationskonzept_zendis.pdf?__blob=publicationFile
- Developers Italia: »Software – The open source catalogue at the Public Administrations disposal«. <https://developers.italia.it/en/software>
- Gustavsson, T. (2014): »Common Criteria and Open Source: Experiences from the certification of an open source product«. *Datenschutz und Datensicherheit – DuD* 38, S. 226 – 231. <https://doi.org/10.1007/s11623-014-0096-7>
- Franck, E.; Jungwirth, C. (2003a): »Die Governance von Open-Source-Projekten«. In: Fandel, G.; Jost, P.J. (Hrsg.): »Ökonomische Analyse von Governance-Strukturen«. *Zeitschrift für Betriebswirtschaft (ZfB) Ergänzungshefte*, Vol 5. Gabler Verlag. https://doi.org/10.1007/978-3-663-12270-8_1
- Franck, E.; Jungwirth, C. (2003b): »Reconciling Rent-Seekers and Donators – The Governance Structure of Open Source«. *Journal of Management & Governance* 7, S. 401 – 421. <https://doi.org/10.1023/A:1026261005092>
- Free Software Foundation Europe & Kompetenzzentrum Öffentliche IT (2020): »Public Money Public Code – Modernisierung der öffentlichen Infrastruktur mit Freier Software«. <https://www.oeffentliche-it.de/publikationen?doc=119321>
- Foundation for Public Code: »Glossary«. <https://standard.publiccode.net/glossary>
- GitHub: »Open Source Guides«. <https://opensource.guide/de/>
- Jullien, N.; Stol, K.-J.; & Herbsleb, J. D. (2019): »A Preliminary Theory for Open-Source Ecosystem Microeconomics«. In: Fitzgerald, B.; Mockus, A.; Zhou, M. (Hrsg.): »Towards Engineering Free/Libre Open Source Software (FLOSS) Ecosystems for Impact and Sustainability – Communications of NII Shonan Meetings«, S. 49 – 68. Springer. https://doi.org/10.1007/978-981-13-7099-1_3
- KGSt (2020): »Ein Ort für den öffentlichen Code – Version 1«. <https://www.kgst.de/documents/20181/26701/Ein-Ort-f%C3%BCr-%C3%B6ffentlichen-Code.pdf/9083470a-ba68-4af4-7258-40a22aebd87c>
- KGSt (2021): »Open Source in Kommunen: Ein Baustein für mehr Digitale Souveränität – Teil 1: Grundverständnis, Potenziale und Herausforderungen«. https://www.kgst.de/documents/20181/34177/KGSt-Bericht-5-2021_Open-Source.pdf/994d10d0-ec25-f8ed-91af-1a12518c27d3
- Kilamo, T.; Hammouda, I.; Mikkonen, T.; Aaltonen, T. (2012): »From proprietary to open source – Growing an open source ecosystem«. *Journal of Systems and Software*, 85(7), S. 1467 – 1478. <https://doi.org/10.1016/j.jss.2011.06.071>

- Konferenz der IT-Beauftragten der Ressorts (2021): »**Digitale Souveränität der IT der öffentlichen Verwaltung – Beschluss 2021/05**«. https://www.cio.bund.de/SharedDocs/Publikationen/DE/Bundesbeauftragter-fuer-Informationstechnik/Konferenz_der_IT-Beauftragten_der_Ressorts_Beschluesse/2021_05_Beschluss_Konferenz_IT-Beauftragte.pdf?__blob=publicationFile
- Lerner, J.; Tirole, J. (2002): »**Some Simple Economics of Open Source**«. *The Journal of Industrial Economics*, 50, S. 197 – 234. <https://doi.org/10.1111/1467-6451.00174>
- Lerner, M.; Price, A.; Schank, H.; Gregori, B. (2020): »**Building and Reusing Open Source Tools for Government**«. New America. <https://www.newamerica.org/digital-impact-governance-initiative/reports/building-and-reusing-open-source-tools-government/section-two-building-open-source-software/>
- Link, G. J. P.; Rao, M.; Marti, D.; Leak, A.; Bodo, R. (2019): »**Marktplatz zur Koordinierung und Finanzierung von Open Source Software**«. *HMD Praxis der Wirtschaftsinformatik*, 56(2), S. 419–437. <https://doi.org/10.1365/s40702-018-00474-6>
- Lombardo, L. (2021): »**Public-Sector Code-Sharing Communities. Project on Digital Era Government**«. <https://medium.com/digitalhkspublic-sector-code-sharing-communities-9f76447667b6>
- Mergel, I. (2015): »**Open collaboration in the public sector: The case of social coding on GitHub**«. *Government Information Quarterly* 32.4, S. 464 – 472. <https://doi.org/10.1016/j.giq.2015.09.004>
- Mouakhar, K.; Tellier, A. (2017): »**How do Open Source software companies respond to institutional pressures? A business model perspective**«. *Journal of Enterprise Information Management*, 30.4, S. 534 – 554. <https://doi.org/10.1108/JEIM-05-2015-0041>
- Mustonen, M. (2003): »**Copyleft—the economics of Linux and other open source software**«. *Information Economics and Policy*, 15.1, S. 99 – 121. [https://doi.org/10.1016/S0167-6245\(02\)00090-2](https://doi.org/10.1016/S0167-6245(02)00090-2)
- Neary, D.; Berkus, J.; Novakovic, K.; Behrenshausen, K. (2020): »**Understanding open source governance models**«. Red Hat Blog. <https://www.redhat.com/en/blog/understanding-open-source-governance-models>
- O’Mahony, S. (2007): »**The governance of open source initiatives: what does it mean to be community managed?**«. *Journal of Management & Governance* 11, S. 139 – 150. <https://doi.org/10.1007/s10997-007-9024-7>
- Open CoDE: »**Über Open CoDE**«. <https://opencode.de/ueber-open-code>
- Open Source Business Alliance (OSBA) (2018): »**Handreichung zur Nutzung der EVB-IT beim Einsatz von Open Source Software**«. <https://osb-alliance.de/publikationen/veroeffentlichungen/handreichungen-zur-nutzung-der-evb-it-beim-einsatz-von-open-source-software>
- Open Source Business Alliance (OSBA) (2020): »**Nachhaltiger Open Source Einsatz für die digital souveräne Verwaltung**«. <https://osb-alliance.de/publikationen/veroeffentlichungen/nachhaltiger-open-source-einsatz-fuer-die-digital-souveraene-verwaltung-paper>
- Schmitz, P.-E. (2001): »**Study into the use of Open Source Software in the Public Sector – Part 3**«. <http://ploug22.free.fr/doc/oss-market-structure-urope.pdf>
- Schräpe, J.-F. (2016): »**Open-Source-Projekte: Vom Nischenphänomen zum integralen Bestandteil der Softwareindustrie**«. *WSI-Mitteilungen*, 69(8), S. 603 – 612. <https://doi.org/10.5771/0342-300X-2016-8-603>
- Schwan, C. (2021): »**Health of the KDE community**«. <https://carlschwan.eu/2021/04/29/health-of-the-kde-community/>
- Shaikh, M. (2016): »**Negotiating open source software adoption in the UK public sector**«. *Government Information Quarterly*, 33(1), S. 115 – 132. <https://doi.org/10.1016/j.giq.2015.11.001>
- Subramaniam, C.; Sen, R.; Nelson, M. L. (2009): »**Determinants of open source software project success: A longitudinal study**«. *Decision Support Systems*, 46(2), S. 576 – 585. <https://doi.org/10.1016/j.dss.2008.10.005>
- Technologiestiftung Berlin: »**Berlin Open Source**«. <https://berlinopensource.de/>
- Thapa, B. E. P. (2019): »**Strategische Beschaffung in der IT-Konsolidierung**«. Kompetenzzentrum Öffentliche IT. <https://www.oeffentliche-it.de/publikationen?doc=102761>
- Vitako (2021): »**Ausschreibungen von Open Source Software: Tipps und Hinweise zur Vergabe von Open Source Software für die öffentliche Verwaltung**«. https://vitako.berlin/wp-content/uploads/2022/05/VITAKO-Handreichung_Ausschreibung-von-Open-Source-Software.pdf

7.3 Anleitung zur Nutzung der webbasierten Skizze

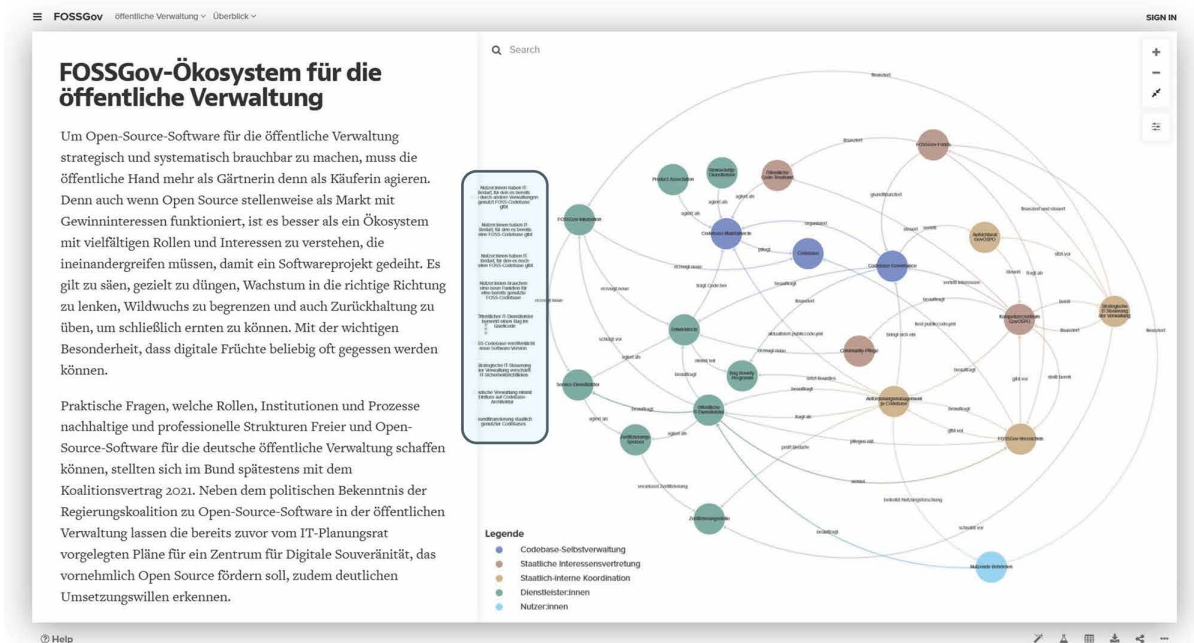
Das FOSSGov-Ökosystem wurde mit dem Concept-Mapping-Tool kumu.io erstellt. Abrufbar unter: <https://kumu.io/oefit/fossgov>

und die Möglichkeit, das Ökosystem auf Relationen ersten Grades einzugrenzen. Die Focus-Funktion wird im Folgenden am Beispiel eines Anwendungsszenarios (Loop) genauer erklärt.

Jedes Element (runde Objekte), jede Relation (Pfeile zwischen den Elementen) und jede Loop (Anwendungsszenarien auf der linken Seite der Karte) können per Mausklick ausgewählt und im rechten Navigationsmenü fokussiert werden. Mit der Auswahl eines Objektes erscheint links ein erklärender Beschreibungstext

1. Anwendungsszenario wählen

Die einzelnen Anwendungsszenarien finden Sie links auf der Karte. Sie können das Anwendungsszenario wählen, indem Sie auf den jeweiligen Text klicken.



2. Anwendungsszenario fokussieren

Nachdem Sie das Szenario ausgewählt haben, ändert sich der Beschreibungstext links neben der Karte, die Kartenansicht

bleibt vorerst gleich. Um lediglich die beteiligten Rollen und Beziehungen übersichtlich anzeigen zu lassen, können Sie rechts in der Navigationsleiste den Button »Focus« wählen.

FOSSGov öffentliche Verwaltung Überblick
SIGN IN

Nutzer:innen haben IT-Bedarf, für den es bereits eine durch andere Verwaltungen genutzt FOSS-Codebase gibt

Eine Kommune oder eine Fachbehörde möchte eine Tätigkeit digitalisieren, die sie bisher analog durchgeführt hat. Andere Ämter in Deutschland nutzen bereits eine geeignete FOSS-Lösung.

1. Eine potenziell nutzende Behörde meldet einen neuen IT-Bedarf an ihren öffentlichen IT-Dienstleister.
2. Der öffentliche IT-Dienstleister sichtet das FOSSGov-Verzeichnis auf bereits durch die Verwaltung genutzte FOSS-Lösungen und wird fündig.
3. Der öffentliche IT-Dienstleister nimmt die Installation und Einführung der Software bei der zukünftig nutzenden Behörde vor oder beauftragt dafür einen mit der Codebase vertrauten Service-Dienstleister.

#öffentliche-verwaltung/nutzer:innen-haben-it-bedarf-twar-den-es-bereits-eine-durch-andere-verwaltungen-genutzt-foss-codebase-gibt | permalink

Q Search

Legende

- Codebase-Selbstverwaltung
- Staatliche Interessensvertretung
- Staatlich-interne Koordination
- Dienstleister:innen
- Nutzer:innen

© Help
Navigation icons

3. Zur Übersicht zurückkehren

Durch erneutes Klicken auf den Button (»Clear Focus«) in der rechten Navigationsleiste, können Sie anschließend zur Übersicht (Unterpunkt 1) zurückkehren.

Hinweis

Kumu ist ein US-amerikanischer Dienst. Die Datenschutzbestimmungen können hier eingesehen werden: <https://kumu.io/privacy>

The screenshot displays the FOSSGov website interface. On the left, there is a text article titled "Nutzer:innen haben IT-Bedarf, für den es bereits eine durch andere Verwaltungen genutzt FOSS-Codebase gibt". The article includes a short introduction and a three-step process for digitalization. On the right, a Kumu diagram visualizes the relationships between different entities: "Staatliche Dienstleister", "Staatliche Interessensvertretung", "Staatlich-interne Koordination", "Dienstleister:innen", and "Nutzer:innen". A blue arrow points to the "Clear Focus" button in the Kumu interface's right-hand navigation panel.

Kontakt

Dorian Grosch
Kompetenzzentrum Öffentliche IT (ÖFIT)
Tel.: +49 30 3463-7173
Fax: +49 30 3463-99-7173
info@oeffentliche-it.de

Fraunhofer-Institut für
Offene Kommunikationssysteme FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin

www.fokus.fraunhofer.de
www.oeffentliche-it.de
Twitter: @OeffentlicheIT

ISBN: 978-3-948582-16-6

