Em002-1 Guida pratica Software a codice sorgente aperto nell'Amministrazione federale

Raccomandazione sull'informatica nell'Amministrazione federale[1]

Il presente documento è un allegato al documento principale Em002.

Classificazione[2]:	Non classificato
Carattere vincolante[3]:	Raccomandazione
Settore di pianificazione[4]:	TIC dell'Amministrazione federale
Versione attuale:	2.0
Sostituisce versione:	1.0 del 2018
Stato:	Bozza
Data di approvazione (versione attuale):	14.02.2025
Approvato da, base legale:	Approvato dal delegato alla trasformazione digitale e alla governance delle TIC (delegato TDT) in virtù dell'articolo 17 capoverso 1 dell'ordinanza del 25 novembre 2020 sulla trasformazione digitale e l'informatica (OTDI), RS 172.010.58
Lingue:	Tedesco (originale), francese, italiano, inglese (traduzioni)

[1] «Empfehlung zur Bundesinformatik» secondo [P035], n. 4.6

[2] Riguardo alla classificazione AD USO INTERNO e CONFIDENZIALE cfr. ordinanza dell'8 novembre 2023 sulla sicurezza delle informazioni (OSIn, RS **128.1**)

[3] Si veda la nota 1.

[4] Settori di pianificazione secondo la Strategia TIC della Confederazione 2020–2023 del 3 aprile 2020 (SB000)

Indice

1. Utilizzo della presente guida 3

2. Definizioni 5

3. Potenziale e sfide 8

3.1 Potenziale dei software a codice sorgente aperto 8
3.2 Sfide legate agli OSS 10

4. Situazioni di utilizzo/elaborazione di OSS 12

4.1 Panoramica situazioni OSS 12
4.2 Impiego di software a codice sorgente aperto senza l'apporto di modifiche 13
4.3 Sviluppo con componenti open source e rilascio del codice sorgente 14
4.4 Collaborazione in progetti aperti (contributo) 14

5. Caratteristiche e selezione di licenze open source 14

6. Acquisto di OSS 15

7. Reperibilità di alternative / soluzioni OSS 16

7.1 alternativeTo 16
7.2 «Repository» di open source 16
7.3 Open Hub 17
7.4 «Repository» speciali di codici 18
7.5 Directory OSS 18

8. Modelli di supporto nell'impiego di OSS 19

8.1 Impiego senza supporto professionale 19
8.2 Impiego con supporto interno 19
8.3 Impiego con supporto a cura di fornitori esterni 20

9. Punto di contatto 20

Appendice 21

A. Modifiche rispetto alla versione precedente 21
B. Riferimenti 21
C. Abbreviazioni 21
D. Modelli commerciali con OSS 22

D.1 Servizi e prodotti basati su OSS 22

D.2 Servizi per OSS 22

D.3 «Subscription» 22

D.4 «Dual Licensing» 23

1. Utilizzo della presente guida

Secondo l'articolo 9 della legge federale concernente l'impiego di mezzi elettronici per l'adempimento dei compiti delle autorità (LMeCA), la Confederazione è tenuta a pubblicare il codice sorgente dei software che sviluppano o fanno sviluppare per l'adempimento dei propri compiti[5]. Da questo mandato legale, il settore TDT della Cancelleria federale ha ricavato una serie di obiettivi sovraordinati elencati nella «Guida strategica Software a codice sorgente aperto nell'Amministrazione federale».

La presente guida pratica concretizza le disposizioni giuridiche e strategiche e serve alle unità amministrative della Confederazione per l'utilizzo, l'acquisto e il rilascio di software a codice sorgente aperto (open source software, OSS). Fornisce a tutti gli interessati una panoramica completa sull'argomento e rimanda a ulteriori informazioni in altri ausili.

A seconda delle conoscenze pregresse e delle esigenze dei lettori, è tuttavia ragionevole leggere la guida solo in modo mirato e passare in seguito agli altri documenti. La panoramica seguente ha lo scopo di aiutare a individuare gli argomenti rilevanti. In caso di dubbi, si può iniziare leggendo integralmente la guida.

Intenzione / interesse	**Documentazione rilevante**
Introduzione generale ai software a codice sorgente aperto	• Numeri 2, 3 e 7 • *«Em002-5 Scheda informativa LMeCA e OSS»* • *«Em002-3 Guida Licenze OSS»* • *«Em002-6 FAQ OSS e articolo 9 LMeCA»*
Introduzione generale LMeCA per decisori	• *«E002-5 Scheda informativa LMeCA e OSS»* • *«Em002 Guida strategica Software a codice sorgente aperto nell'Amministrazione federale»*
Acquisto di software solo da utilizzare	• Numeri 4.1, 4.2 • *«Em002-3 Guida Licenze OSS»* • *«Merkblatt Softwarebeschaffung» (Promemoria acquisto di software) [UFCL 2015] (deprec.)* • *«Projektvorlagen Beschaffung» (Modelli di progetto Acquisti)*
Nuovo sviluppo nel campo dell'informatica e rilascio	• Numeri 4.3, 4.4 • *«Em002-2 Istruzioni sulla pubblicazione di OSS»* • *«Em002-2.1 Lista di controllo Chiarimenti preliminari»* • *«Em002-2.2 Lista di controllo Analisi e preparazione»* • *«Em002-2.3 Lista di controllo Rilascio e pubblicazione»* • *«Em002-4 Guida Community»* • *«Em002-4.1 Lista di controllo Community OSS»*
Acquisto di software creati (servizio o opera)	• Numeri 4.1, 4.3, 4.4 • *«Merkblatt Beschaffungen und EMBAG» (Promemoria acquisti e LMeCA) [UFCL]* • *«Anmeldung Ausschreibung» (Iscrizione bando)* • *«Kickoff-Folien Ausschreibung» (Diapositive di avvio bando)* • *«Merkblatt Softwarebeschaffung» (Promemoria acquisto di software) [UFCL 2015] (deprec.)* • *«Projektvorlagen Beschaffung» (Modelli di progetto Acquisti)*

[5] FF 2023 787 – Legge federale concernente l'impiego di mezzi elettronici per i compiti delle autorità Fedlex (admin.ch)

	• *«Em002-2 Istruzioni sulla pubblicazione di software a codice sorgente aperto»* • *«Em002-2.1 Lista di controllo Chiarimenti preliminari»* • *«Em002-2.2 Lista di controllo Analisi e preparazione»* • *«Em002-2.3 Lista di controllo Rilascio e pubblicazione»* • *«Em002-4 Guida Community»* • *«Em002-4.1 Lista di controllo Community OSS»*
Considerazione giuridica del rilascio	• *«Em002-6 FAQ OSS e articolo 9 LMeCA»* • *«Em002-3 Guida Licenze OSS»*
Basi di lavoro per capiprogetto	• *«Documentazione HERMES»*

Il grafico seguente mostra una panoramica dei documenti rilevanti per gli OSS nell'Amministrazione federale.

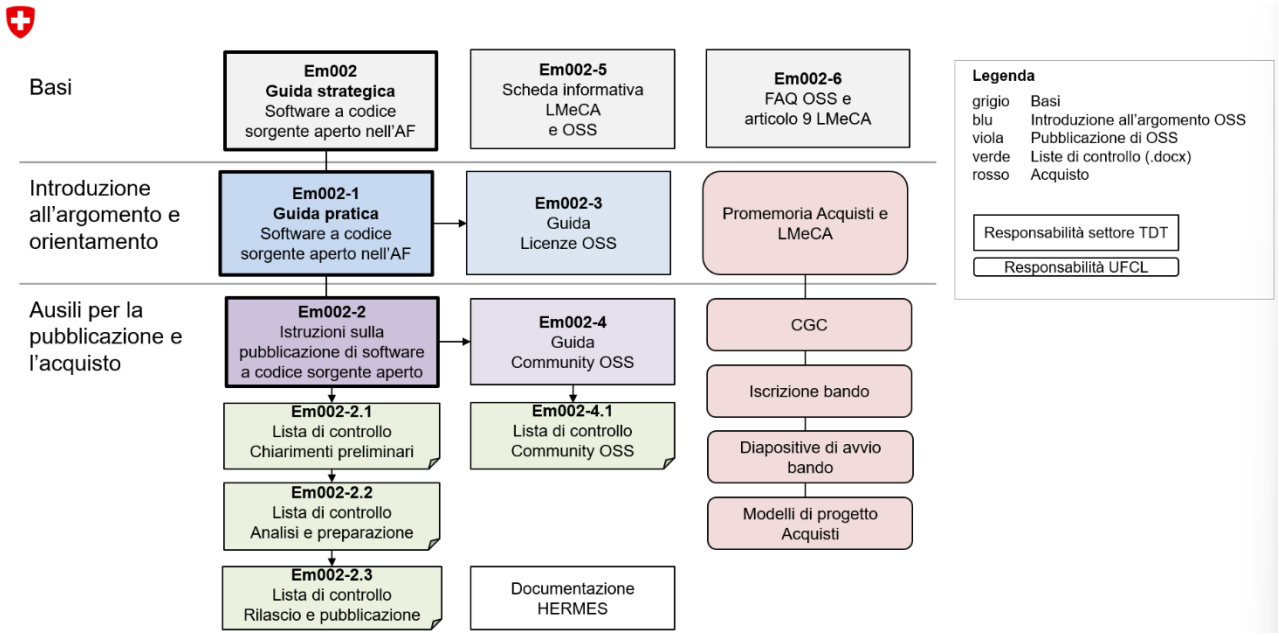

Figura 1: Panoramica degli ausili OSS in relazione con l'articolo 9 LMeCA

2. Definizioni

Software a codice sorgente aperto	Si parla di software a codice sorgente aperto (open source software, OSS) se questo viene pubblicato con una delle circa 80 licenze riconosciute dalla Open Source Initiative (OSI) [OSI2019]. La definizione di open source definisce dieci criteri che tutte le licenze open source devono soddisfare [PE1999]. In sostanza, possono essere riassunti in quattro punti: 1. Il software può essere impiegato liberamente. 2. Il codice sorgente del software è liberamente accessibile. 3. Il software può essere copiato e diffuso liberamente. 4. Il software può essere modificato e trasmesso in forma modificata a determinate condizioni.
Software libero	Ancor prima del concetto di «open source», negli anni Ottanta è stato introdotto dalla «Free Software Foundation» (FSF) quello di «software libero». In termini di contenuto, il software libero corrisponde alle condizioni del software a codice sorgente aperto, ma con l'obiettivo di mantenere il software sempre liberamente accessibile senza che sia integrato, per quanto possibile, in software proprietari[6].
Software proprietario	Nel caso di un software proprietario, un fornitore sviluppa un software e vende una licenza di utilizzo all'utente. In genere l'utente non conosce il codice sorgente e non può trasmettere o modificare il software; può utilizzare il software solo secondo le prescrizioni della licenza (p. es. «end-user license agreement», EULA) dietro pagamento dei rispettivi diritti: per esempio, il software può essere utilizzato solo da un certo numero di persone o per un certo numero di processori[7]. A fronte di costi di manutenzione con cadenza perlopiù annuale, il fornitore si impegna a eliminare i problemi in tempo utile e a sviluppare ulteriormente il software con regolarità.
Licenza	Una licenza è un documento che contiene direttive vincolanti per l'utilizzo e la diffusione di un software. Nel caso degli open source, sono utilizzate perlopiù licenze conformi alla OSI[8].
Community	Una community nel contesto di un software open source ha una definizione molto ampia. Può trattarsi di un ecosistema informale o di una struttura con una vera e propria governance che sviluppa software. Il documento «*Em002-4 Guida Community*» prende in esame questo argomento. I membri della community possono assumere insieme la gestione del prodotto, sviluppare, svolgere test, eseguire traduzioni, fornire feedback o semplicemente utilizzare il software.
Diritti di terzi	Anche in ambito di software a codice sorgente aperto possono sussistere diritti di protezione (diritto d'autore, diritto della marca, diritto di brevetto) e possono essere fatti valere da parti terze – per esempio nell'utilizzo di un codice sorgente creato da terzi.
Eccezioni secondo la LMeCA	I diritti di terzi e motivi importanti in materia di sicurezza valgono come eccezioni secondo l'articolo 9 capoverso 1 LMeCA. Entrambi i casi vengono affrontati al numero 3 del documento «*Em002-2 Istruzioni sulla pubblicazione di software a codice sorgente aperto*».

[6] Cfr. https://it.wikipedia.org/wiki/Software_libero

[7] Cfr. https://it.wikipedia.org/wiki/Software_proprietario

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

Source code / Codice sorgente	Il «source code», detto anche codice sorgente, in informatica è il testo di un programma per computer leggibile per l'essere umano e scritto in un linguaggio di programmazione[9].
Pubblicazione	Per pubblicazione si intende in sostanza la divulgazione del codice sorgente con una licenza aperta secondo l'OSI. Per ragioni di ordine pratico, spesso si intendono anche la documentazione e le istruzioni automatiche per la creazione dell'applicazione a partire dal codice sorgente, nonché test automatici e documentazione degli stessi. Perché possa nascere una community, si utilizza una piattaforma in grado di assumere tutte le funzioni di una piattaforma di sviluppo («build», test, supporto).
Sovranità digitale	Facciamo riferimento alla definizione riportata nel rapporto sulla sovranità digitale del DFAE [St2024], secondo cui la sovranità digitale di uno Stato o di un'organizzazione comprende necessariamente il controllo totale dei dati salvati ed elaborati nonché la decisione indipendente riguardo a chi vi può accedere. Comprende inoltre l'abilità di sviluppare autonomamente componenti e sistemi tecnologici, controllarli e integrarvi altri componenti.
Repository	Un «repository» è un luogo di conservazione che viene utilizzato da uno strumento di controllo delle versioni per i file e i metadati sul «code base». I «repository» permettono a più contributori di lavorare agli stessi file e salvare diverse versioni. La maggior parte dei «repository» consentono anche funzionalità di tracciamento delle problematiche («issue tracking»), gestione delle release, «build» automatici e documentazione.
Fork	Il «fork»[10] di un progetto open source è il processo attraverso cui si sviluppa un nuovo progetto indipendente a partire da quello originario.
Contributor license agreement (CLA)	Un «contributor license agreement»[11] è un documento che stabilisce le condizioni alle quali si può apportare proprietà intellettuale a un progetto.
Source code management (SCM)	Il «source code management»[12], detto anche controllo versione o gestione del codice sorgente, è uno strumento che, soprattutto nello sviluppo di software, gestisce efficacemente modifiche e numeri delle versioni. Per la gestione degli SCM distribuiti si è affermato il software libero Git[13].
Open source software development (OSSD)	Con «open source software development» (OSSD)[14] si intende lo svolgimento pubblico dell'intero processo di sviluppo, in aggiunta alla pubblicazione del codice sorgente. Tutto il processo, dai requisiti («issue») fino al codice sorgente, è trasparente. L'OSSD è supportato da strumenti di comunicazione pubblici (mailing list, forum ecc.), un sistema di controllo delle versioni (Git), bug ed elenchi delle caratteristiche, roadmap e strumenti per gli sviluppatori.

[9] Cfr. https://it.wikipedia.org/wiki/Codice_sorgente

[10] Cfr. https://it.wikipedia.org/wiki/Fork_(sviluppo_software)

[11] Cfr. https://it.wikipedia.org/wiki/Contributor_License_Agreement

[12] Cfr. https://it.wikipedia.org/wiki/Controllo_versione

[13] Cfr. https://it.wikipedia.org/wiki/Git_(software)

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

Software package data exchange (SPDX)	Il «software package data exchange» (SPDX)[15] descrive un formato standard per il «software bill of materials» (SBOM), con l'obiettivo di agevolare la corretta gestione del software libero o open source.
Effetto copyleft	Se un software è pubblicato con una licenza che prevede il «copyleft», tutte le modifiche e gli ampliamenti del codice sorgente devono essere nuovamente rilasciati con la licenza del software a codice sorgente aperto elaborato (cfr. anche «*Em002-3 Guida Licenze OSS*»).

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

3. Potenziale e sfide

Nell'informatica odierna, i software a codice sorgente aperto hanno un potenziale elevato, ma, allo stesso tempo, l'utilizzo di soluzioni open source comporta anche diverse sfide. La sezione seguente illustra più nel dettaglio questi due aspetti. I dati si basano, fra l'altro, sui risultati dello studio sui programmi open source in Svizzera («Open Source Studie Schweiz») 2018, 2021 e 2024 dell'Università di Berna, in cui le persone intervistate hanno fornito informazioni su vantaggi e svantaggi riconosciuti dei software a codice sorgente aperto.

3.1 Potenziale dei software a codice sorgente aperto

I punti seguenti descrivono il potenziale che si può raggiungere grazie all'utilizzo e al rilascio di software a codice sorgente aperto.

1. Sovranità digitale	Si tratta della capacità di utilizzare e controllare prestazioni digitali. L'autodeterminazione riguarda l'intero ciclo di vita di un sistema digitale. I software a codice sorgente aperto incentivano la sovranità digitale in quanto gli stessi software possono essere in ogni caso ulteriormente sviluppati in maniera autonoma[16].
2. Nessun pagamento di diritti di licenza	L'utilizzo di software a codice sorgente aperto non prevede il pagamento di costi di licenza. Tuttavia, per l'acquisto di complessi pacchetti standard di software open source può avere senso acquistare le cosiddette «subscription» (abbonamento al supporto) a pagamento.
3. Risparmio di costi grazie alla cooperazione con altri utenti	Poiché i software con licenza open source possono essere utilizzati e sviluppati ulteriormente senza limitazioni, in base al principio «sviluppare una volta – applicare più volte» è possibile condividere i costi dell'ulteriore sviluppo oppure sfruttare gli sviluppi messi in atti da altri servizi amministrativi. Ciò consente al tempo stesso di beneficiare delle esperienze e degli sviluppi altrui.
4. Community building e scambio di conoscenze	I software a codice sorgente aperto agevolano la costituzione di comunità («community building») e lo scambio di conoscenze, per esempio tra i diversi livelli federali della Svizzera. Ciò consente di lavorare congiuntamente all'ulteriore sviluppo di un software, eliminare gli errori e condividere le esperienze individuali. Uno scambio di conoscenze più intenso fra i vari servizi amministrativi permette di comprendere meglio il lavoro degli altri, in modo da evitare sovrapposizioni e diffondere le soluzioni migliori.
5. Minore dipendenza dai produttori	Nell'informatica, la dipendenza dai produttori di software («vendor lock-in») viene valutata come molto elevata. L'utilizzo di software con licenza open source riduce la dipendenza dai produttori poiché gestione, manutenzione, supporto, ulteriore sviluppo e altri servizi per i software a codice sorgente aperto possono essere messi a concorso pubblicamente.

[16] Facciamo riferimento alla definizione riportata in [St2024], secondo cui la sovranità digitale di uno Stato o di un'organizzazione comprende necessariamente il controllo totale dei dati salvati ed elaborati nonché la decisione indipendente riguardo a chi vi può accedere. Comprende inoltre l'abilità di sviluppare autonomamente componenti e sistemi tecnologici, controllarli e integrarvi altri componenti.

6. Standard aperti ed elevata interoperabilità	In genere la compatibilità dei software a codice sorgente aperto con altre soluzioni software e informatiche (interoperabilità) è maggiore rispetto ai software proprietari. Inoltre, le soluzioni open source utilizzano quasi esclusivamente formati di dati aperti: per questo motivo i dati possono essere scambiati facilmente con altri sistemi.
7. Trasparenza sulla struttura del software	Essendo il software disponibile anche sotto forma di codice sorgente, la sua qualità può per esempio essere verificata tramite revisioni esterne. Inoltre, sulla base del codice sorgente è possibile produrre documentazione (p. es. riguardo a nuovi bandi per prestazioni di ulteriore sviluppo o la sostituzione di un software a codice sorgente aperto alla fine del suo ciclo di vita).
8. Sicurezza e fiducia grazie alla trasparenza	L'accessibilità del codice sorgente può aumentare la sicurezza dei software a codice sorgente aperto rispetto ai software proprietari. Inoltre, nell'open source è molto più difficile creare «backdoor» e altre falle e questo genera una maggiore fiducia nel software.
9. Qualità e modularità maggiori del codice sorgente	Gli sviluppatori possono essere più motivati a scrivere codici di programmazione validi se sanno che il loro codice sorgente sarà pubblicato: questo va a vantaggio della qualità dei software a codice sorgente aperto, che può così risultare maggiore rispetto a quelli proprietari. Inoltre, le soluzioni a codice sorgente aperto sono di solito strutturate in modo altamente modulare per consentire di sostituire facilmente singoli moduli e continuare a utilizzare gli altri nello stesso tempo[17].
10. Facile adattamento alle proprie esigenze	L'accesso al codice sorgente consente agli utenti di eseguire ulteriori sviluppi in autonomia o a opera di fornitori esterni. È così possibile adattare rapidamente il software alle proprie esigenze.
11. Innovazione e integrazione rapide	La community mette in atto un ulteriore sviluppo rapido e continuo del software a codice sorgente aperto. Per esempio, nuovi standard di dati e tecnologie vengono spesso pubblicati come librerie di programmazione open source, rendendo possibile la rapida realizzazione di soluzioni software innovative.
12. Attrattiva come datore di lavoro	L'impiego di moderne tecnologie open source e la collaborazione informale con le community internazionali incentivano la motivazione dei collaboratori incrementando così l'attrattiva come datore di lavoro; questo aspetto agevola a sua volta il reclutamento di professionisti giovani e qualificati.

[17] Un ulteriore aspetto della sicurezza, rispetto ai software proprietari, è la maggiore semplicità nel poter documentare le librerie utilizzate (dipendenze incluse). D'altra parte, ciò significa poter identificare e risolvere rapidamente eventuali vulnerabilità che dovessero presentarsi (cfr. anche SBOM).

3.2 Sfide legate agli OSS

Di seguito sono descritte le sfide tipiche legate ai software a codice sorgente aperto che si incontrano nella prassi e sono illustrati i possibili approcci per farvi fronte. In questa sede non sono trattate le sfide generali che riguardano tutti i software (p. es. la necessità di verificare la cibersicurezza e di garantire il supporto per i software critici).

1. Elevati costi di sostituzione dovuti alle dipendenze presenti	La sostituzione di un software proprietario con uno open source può rivelarsi molto onerosa se l'applicazione è radicata nel sistema informatico esistente tramite integrazione di interfacce o altre dipendenze. A causa di questi costi di sostituzione, spesso l'introduzione di software a codice sorgente aperto è ragionevole solo alla fine del ciclo di vita del software proprietario.
2. Caratteristiche mancanti o nessuna soluzione open source adatta	Se si deve acquistare un software standard, può capitare che come alternativa alla soluzione proprietaria (da sostituire) sia disponibile soltanto una soluzione open source insufficiente dal punto di vista funzionale oppure che non ve ne sia alcuna. Una possibile soluzione può essere quella di sviluppare le caratteristiche mancanti insieme ad altri utenti del software a codice sorgente aperto e aggiungerle a tale progetto[18].
3. Maggiori costi di integrazione	I risparmi nel breve termine derivanti dall'assenza di costi legati alle licenze per i software a codice sorgente aperto sono spesso compensati da oneri maggiori che per l'integrazione. Perciò è importante riconoscere che l'impiego professionale di software a codice sorgente aperto non è gratuito, ma genera un onere interno o esterno.
4. Responsabilità e supporto non chiari	Ai software a codice sorgente aperto viene spesso criticata la mancanza di supporto e manutenzione a cura di aziende esterne. A oggi, vi sono tuttavia molti fornitori che offrono servizi commerciali (supporto a lungo termine, manutenzione continua, ulteriore sviluppo, garanzie ecc.) per soluzioni open source. L'appendice D illustra i diversi modelli commerciali con i software a codice sorgente aperto, mentre al numero 8 sono descritti i vari modelli di supporto. Una panoramica attuale dei fornitori di servizi open source e delle loro prestazioni è disponibile nella Open Source Directory (https://www.ossdirectory.com/it) o nello studio OSS 2024[19] [OSS2024].
5. Mercato in parte ristretto con pochi fornitori	Poiché certe applicazioni open source, come per esempio quelle desktop, in genere comportano pochi o persino nessun costo di supporto o manutenzione, in alcuni casi è quasi del tutto assente un mercato di fornitori open source che offrano prestazioni di supporto e manutenzione. In questo caso, la messa a concorso di prestazioni per software a codice sorgente aperto potrebbe contribuire alla creazione di un mercato che rafforzi la concorrenza tra gli offerenti. Per la messa a concorso di tali prestazioni, sono importanti dal punto di vista pratico l'esistenza di una community di sviluppatori largamente sostenuta per il software in questione e la qualità della documentazione di sviluppo.

[18] La responsabilità di queste nuove caratteristiche è assunta dai project manager o dai gestori dell'applicazione. Il progetto ne disciplina la procedura esatta. I documenti Em002-2 e Em002-4 aiutano in caso di decisioni legate a dettagli. In base all'entità della caratteristica in questione può non essere necessario leggere interamente le istruzioni. Per quanto riguarda le risorse, si devono osservare gli aspetti legati al diritto in materia di appalti pubblici.

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

6. Scarsa visibilità	Le community dei progetti open source si concentrano prevalentemente sullo sviluppo di prodotti e scarsamente sulla loro commercializzazione. Al contrario, i produttori di software proprietari investono intensamente nel marketing e nella vendita dei loro prodotti. Questa mancata pubblicità di software a codice sorgente aperto genera spesso la falsa credenza che non esista un'alternativa ai software proprietari. Esistono invece piattaforme come alternativeTo[20] che offrono una panoramica delle soluzioni per determinati compiti (ulteriori informazioni sulle alternative OSS sono disponibili al n. 7).
7. Scarsa accettazione da parte degli utenti finali	A causa di interfacce utenti diverse, mancanza di alcune funzioni, bassa usabilità e poca pubblicità per i software a codice sorgente aperto, gli utenti finali faticano ad accettare alcuni prodotti di questo tipo. Una comunicazione adeguata e l'offerta di documentazione e formazione possono contrastare questa sfida. Il fatto che i software a codice sorgente aperto possano essere ulteriormente sviluppati secondo le esigenze degli utenti può a sua volta portare a una maggiore accettazione.
8. Know-how interno scarso o assente	Le soluzioni open source si evolvono rapidamente e necessitano al tempo stesso di approfondite conoscenze tecniche. Perciò può capitare che i collaboratori interni possiedano uno scarso know-how su alcuni software a codice sorgente aperto o che tale know-how sia persino assente. Questa conoscenza può essere però costruita grazie a formazioni continue e possibilità di studio autonomo su fonti online e alla creazione di wiki *ad hoc*.
9. Difficile valutazione del futuro	Spesso, per chi non partecipa a una community open source è difficile riconoscere come si svilupperà ulteriormente il progetto in futuro. È quindi importante poter svolgere una valutazione realistica sullo sviluppo futuro di una soluzione open source. A tale riguardo, al numero 7.3 viene presentata la piattaforma Open Hub, che consente di valutare l'attività e l'eterogeneità della community di sviluppatori. Questo consente di valutare meglio lo sviluppo futuro di un progetto open source.
10. Incertezze dal punto di vista giuridico	La moltitudine di licenze open source e il numero finora scarso di decisioni giudiziarie riguardo a dubbi d'interpretazione causano talvolta insicurezze dal punto di vista giuridico relative ai software a codice sorgente aperto. Questa guida si pone l'obiettivo di fornire una panoramica sulle licenze open source più importanti e sulle loro caratteristiche e compatibilità. I documenti «*Em002-3 Guida Licenze OSS*» e «*Em002-6 FAQ OSS e articolo 9 LMeCA*» contengono fonti presso cui reperire risposte approfondite sulle questioni giuridiche.
11. Offerta troppo ampia di software a codice sorgente aperto	Negli ultimi anni il numero di software a codice sorgente aperto è aumentato notevolmente. I potenziali utenti di software a codice sorgente aperto criticano perciò la grande quantità di soluzioni presenti sul mercato. Per questo motivo il numero 7 «Reperibilità di alternative / soluzioni OSS Reperibilità di » presenta due piattaforme, alternativeTo e Open Hub, con cui è possibile confrontare le diverse soluzioni open source.

[20] https://alternativeto.net/

4. Situazioni di utilizzo/elaborazione di OSS

4.1 Panoramica situazioni OSS

Nell'utilizzo e nello sviluppo di OSS a opera dell'Amministrazione federale è possibile individuare diverse situazioni riguardo agli obblighi legati alle licenze open source.

Situazione	Conseguenza
Il **mero utilizzo** di software a codice sorgente aperto esistenti (senza modifiche del codice di programmazione).	Non sussiste **alcun obbligo** di distribuzione del codice sorgente.
Lo **sviluppo completamente nuovo** di un software che deve essere pubblicato con una licenza open source.	**In questo caso si ha la piena libertà riguardo alla licenza con cui viene pubblicato il software.** Per la Confederazione si dovrebbe ricorrere al documento «*Em002-3 Guida Licenze OSS*». Si devono osservare i documenti «*Em002-2 Istruzioni sulla pubblicazione di OSS*» e «*Em002-4 Guida Community*».
L' **ulteriore sviluppo** (interno o esterno) con componenti open source esistenti (con o senza effetto «copyleft»[21]) purché il software venga **utilizzato esclusivamente all'interno della stessa organizzazione** e non venga distribuito.	I componenti OSS possono essere combinati a piacimento con componenti di terzi. In generale la licenza non prevede alcun obbligo di pubblicazione purché il software venga utilizzato esclusivamente all'interno della stessa organizzazione e non venga distribuito (eccezione AGPL). Poiché in termini giuridici l'estensione del concetto di «distribuzione all'interno della stessa organizzazione» non è ben definita, questo scenario trova applicazione solo in casi eccezionali di componenti OSS con effetti «copyleft». In casi simili occorre coinvolgere il servizio giuridico della propria organizzazione. Cfr. «*Em002-3 Guida Licenze OSS*» e «*Em002-6 FAQ OSS e articolo 9 LMeCA*». Si devono osservare i documenti «*Em002-2 Istruzioni sulla pubblicazione di OSS*» e «*Em002-4 Guida Community*». **Poiché per l'autorità federale sussiste l'obbligo di pubblicazione secondo l'articolo 9 LMeCA, essa deve rilasciare di nuovo anche le modifiche. Il modo più semplice per farlo è sotto forma cambiamenti (change) apportati al progetto open source esistente.**
Lo **sviluppo ulteriore** (interno o esterno) **esclusivamente con componenti** open source esistenti **senza effetto «copyleft»**, quando il software	I componenti OSS possono essere combinati a piacimento con componenti di terzi e la licenza non prevede alcun obbligo di pubblicazione (cfr. anche «*Em002-3 Guida Licenze*

[21] Spiegazioni sul concetto di «copyleft» nel documento «Em002-3 Guida Licenze OSS».

deve essere distribuito anche all'esterno (p. es. ai Cantoni).	*OSS*»). **Nel quadro dell'articolo 9 LMeCA il rilascio è invece obbligatorio. Idealmente il rilascio avviene tramite il progetto esistente (nessun fork).** Si devono osservare i documenti «*Em002-2 Istruzioni sulla pubblicazione di OSS*» e «*Em002-4 Guida Community*».
Lo **sviluppo ulteriore** (interno o esterno) con componenti open source esistenti **con effetto «copyleft»**, quando il software deve essere distribuito anche all'esterno (p. es. ai Cantoni).	**Sussiste** già **un obbligo di pubblicazione** della licenza «copyleft» (sull'effetto «copyleft» cfr. anche «*Em002-3 Guida Licenze OSS*»). Si devono osservare i documenti «*Em002-2 Istruzioni sulla pubblicazione di OSS*» e «*Em002-4 Guida Community*».
Il **contributo** a un progetto open source esistente.	**In generale sussiste l'obbligo di pubblicazione secondo la LMeCA.** **Si utilizzano la governance e la licenza del progetto.** Si devono osservare i documenti «*Em002-2 Istruzioni sulla pubblicazione di OSS*» e «*Em002-4 Guida Community*».
Lo **sviluppo congiunto** di un progetto nuovo o esistente con una community e la partecipazione ai costi.	**In generale sussiste l'obbligo di pubblicazione secondo la LMeCA.** Sulla base dei documenti «*Em002-2 Istruzioni sulla pubblicazione di OSS*», «*Em002-3 Guida Licenze OSS*» e «*Em002-4 Guida Community*», **si creano il progetto e la sua governance.**
Tutte le varianti di sviluppo/ulteriore sviluppo sono svolte da **un fornitore o da terzi incaricati**.	**In generale sussiste l'obbligo di pubblicazione secondo la LMeCA.** Sulla base dei documenti «*Em002-2 Istruzioni sulla pubblicazione di OSS*», «*Em002-3 Guida Licenze OSS*» e «*Em002-4 Guida Community*». Nel processo di acquisto si devono possibilmente creare i presupposti ed eventualmente anche trasferire i requisiti di rilascio ai fornitori/terzi.

4.2 Impiego di software a codice sorgente aperto senza l'apporto di modifiche

Nell'utilizzo di software a codice sorgente aperto in genere non è rilevante con quale licenza open source questi vengano pubblicati perché tutte le licenze di questo tipo (v. n. 2 «Definizioni») consentono l'utilizzo illimitato di software a codice sorgente aperto, indipendentemente dal numero di postazioni di lavoro, utenti collegati in contemporanea, server e processori ecc. che impiega il software.

Nell'impiego di software a codice sorgente aperto senza l'apporto di modifiche **non sussistono obblighi di pubblicazione,** nemmeno per i programmi legati a una licenza con un rigido effetto «copyleft» (ad eccezione della licenza AGPL, Affero General Public License).

Se il software a codice sorgente aperto è introdotto e gestito da fornitori di prestazioni interni alla Confederazione, può **venire meno la procedura di appalto pubblico**, poiché in genere la sola licenza OSS non costa niente al servizio che l'acquista e pertanto non è nemmeno di

per sé rilevante dl profilo degli acquisti.

Nel caso in cui un fornitore esterno di servizi sia incaricato della manutenzione, del supporto e di altri servizi per il software a codice sorgente aperto, si devono invece rispettare le disposizioni del diritto in materia di appalti pubblici. Il numero 8 descrive diverse varianti di supporto. Nell'acquisto di software a codice sorgente aperto è inoltre importante impiegare criteri di idoneità e di aggiudicazione adatti. Le relative basi sono illustrate al numero 8.

4.3 Sviluppo con componenti open source e rilascio del codice sorgente

In questo caso, per quanto riguarda l'obbligo di pubblicazione secondo l'articolo 9 LMeCA si dovrebbe procedere secondo il documento «*Em002-2 Istruzioni sulla pubblicazione di software a codice sorgente aperto*».
Se necessario, si devono consultare anche i documenti «*Em002-3 Guida Licenze OSS*» e «*Em002-4 Guida Community*». L'obiettivo è rilasciare il progetto dopo aver effettuato i controlli secondo le liste di controllo [Em002-2.1], [Em002-2.2], [Em002-2.3].

4.4 Collaborazione in progetti aperti (contributo)

Il numero 4.2 della guida BITKOM [BITCOM2024] espone possibili considerazioni sul contributo dell'Amministrazione federale a progetti aperti già esistenti[22]. In base all'importanza del progetto per l'Amministrazione federale occorre verificare il livello di responsabilità che l'ufficio interessato intende e può assumersi.

La collaborazione e lo sviluppo diretto in progetti aperti possono svolgersi anche sulla base dei documenti per il rilascio. In tal caso occorre porre l'accento sul documento «*Em002-4 Guida Community*» e sulla relativa lista di controllo [Em002-4.1].

5. Caratteristiche e selezione di licenze open source

Le varie licenze open source e le loro caratteristiche sono descritte nel documento «*Em002-3 Guida Licenze OSS*».

[22] Cfr. https://www.bitkom.org/Bitkom/Publikationen/Bitkom-Leitfaden-zu-Open-Source-Software-20.html, n. 4.2 (in tedesco).

6. Acquisto di OSS

Il presente capitolo e i seguenti si occupano dell'acquisto e dell'impiego di software a codice sorgente aperto senza l'apporto di modifiche. Il documento principale è il «*Merkblatt Beschaffung und EMBAG*»[23] (Promemoria acquisti e LMeCA), rilevante per i servizi IT e le opere. Altri documenti di interesse sono:

- «Anmeldung Ausschreibung» (Iscrizione bando)
- «Kickoff-Folien Ausschreibung» (Diapositive di avvio bando)
- Anche la «Em002-3 Guida Licenze OSS» può essere utile
- È possibile consultare anche la panoramica dei software a codice sorgente aperto impiegati, rilasciati e co-sviluppati

Per l'«enterprise readiness» dell'open source si può consultare anche il documento «Selection criteria for enterprise-ready open source software» [Gu2024][24] oppure anche il numero 3.3 della guida Bitkom[25] disponibile all'indirizzo https://www.bitkom.org/Bitkom/Publikationen/Bitkom-Leitfaden-zu-Open-Source-Software-20.html.

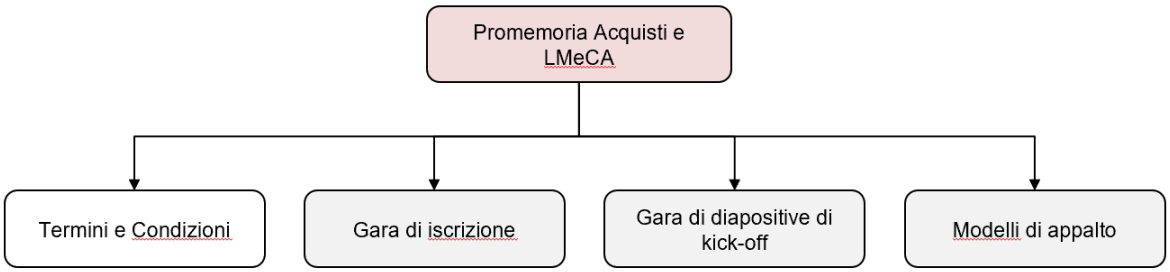

Figura 2: Documenti centrali sull'acquisto di OSS in relazione con l'articolo 9 LMeCA.

[23] I promemoria dell'UFCL destinati ai servizi richiedenti e agli acquisti sono disponibili nello strumentario corrispondente (in tedesco): https://intranet.bbl.admin.ch/bbl_kp/de/home/informatik/beschaffung-buerotechnik-informatik-des-bbl/werkzeugkasten.html.

[24] Cfr. [Gu2024]

[25] Cfr. https://www.bitkom.org/Bitkom/Publikationen/Bitkom-Leitfaden-zu-Open-Source-Software-20.html, n. 3.3 (in tedesco).

7. Reperibilità di alternative / soluzioni OSS

Grazie a diverse piattaforme, le organizzazioni interessate possono trovare e analizzare i software a codice sorgente aperto prima di impiegarli o acquistarli. I dati che ne risultano rappresentano un punto di partenza per la decisione relativa all'impiego o all'acquisto di software a codice sorgente aperto.

In ogni caso, dovrebbe fungere da base una compilazione minima dei requisiti ed eventualmente un'analisi del mercato.

7.1 alternativeTo

Uno strumento pratico per reperire alternative a un software è «alternativeTo – Crowdsourced software recommendations»[26]. Come suggerisce il nome, le alternative e le valutazioni di queste proposte derivano dal «crowdsourcing» («crowd» significa «folla» e «sourcing» significa «approvvigionamento»), ovvero nascono dagli input di molti singoli utenti. alternativeTo distingue quattro tipi di soluzioni software, tra cui solo la prima vale come software a codice sorgente aperto secondo l'articolo 9 LMeCA.

Free open source	Software pubblicato con una licenza open source.
Free	Software gratuito (freeware), il cui codice sorgente non è però liberamente accessibile e non può essere modificato.
Freemium	Questo tipo di software offre una soluzione gratuita e una premium, anche se tutte le funzionalità importanti possono già essere utilizzate nella versione gratuita. Non rientrano in questa categoria i software che possono essere testati gratuitamente per un certo periodo (p. es. con una versione di prova di 30 giorni) che corrispondono invece alla categoria «commercial».
Commercial	Si tratta di software proprietari il cui utilizzo prevede il pagamento di costi di licenza.

7.2 Repository di open source

Attualmente, la piattaforma di sviluppo open source più popolare al mondo, con oltre 30 milioni di utenti e oltre 100 milioni di directory di codici sorgenti[27] è GitHub. Oggi sono presenti su GitHub praticamente tutte le aziende di informatica, in particolare anche i produttori di software proprietari, nonché ogni altro tipo di aziende e organizzazioni. Le imprese pubblicano sulla piattaforma applicazioni open source, strumenti, librerie di programmazione ecc. Anche sempre più autorità – quelle svizzere sono oltre una decina[28] – pubblicano i propri software a codice sorgente aperto su «GitHub and Government» alla pagina https://government.github.com. Questo processo può essere eseguito anche con altri repository.

Altri repository sono, per esempio:

- GitLab
- BitBucket
- Sourceforge
- Launchpad

[26] https://alternativeto.net/
[27] https://octoverse.github.com
[28] https://government.github.com/community/#switzerland

Nella sezione «Insights» di GitHub si possono leggere numerose statistiche interessanti di un progetto open source:

Pulse	Panoramica delle attività più recenti di un progetto open source su GitHub: riepilogo delle informazioni più importanti su intensità di sviluppo, eterogeneità della community, segnalazioni in sospeso, miglioramenti («pull request») ecc.
Contributors	Rappresentazione di quali sviluppatori sono stati attivi e in che momento: un indicatore rilevante per comprendere se tutto dipende da una sola persona oppure da una community più ampia.
Commits	Visualizzazione di quali contributi sono stati apportati al progetto open source in questione e in quale arco di tempo.
Code frequency	Visualizzazione di quanto del codice sorgente è stato aggiunto o rimosso e in quale momento.
Dependency graph	Dipendenze del progetto open source da altri software a codice sorgente aperto (p. es. rilevante per l'identificazione delle lacune in materia di sicurezza e degli aggiornamenti).
Network	Rappresentazione di quale sviluppatore ha apportato un contributo a un certo ramo di sviluppo e in quale momento.
Forks	Elenco di tutte le copie di progetti open source su GitHub: indicatore della popolarità e della diffusione del software a codice sorgente aperto.

7.3 Open Hub

Per la raccolta di informazioni su una soluzione open source che potrebbero non essere state sviluppate su GitHub è possibile ricorrere a Open Hub dell'azienda Black Duck[29], che comprende in maniera organizzata una grande quantità di informazioni provenienti da circa mezzo milione di progetti open source.

Project Summary	Una breve descrizione del progetto open source.
In a Nutshell	I fatti più importanti di un progetto open source, come il numero di «commit», di «contributor» e linee di codice nonché il linguaggio di programmazione più utilizzato, il momento in cui sono stati inseriti i primi «commit» e le ultime modifiche. Inoltre è riportata anche una valutazione relativa alla base del codice e alle dimensioni del team di sviluppatori (p. es. «Mozilla Firefox has a well-established, mature codebase maintained by a very large development team with stable Y-O-Y commits»).
Quick Reference	Contiene informazioni sull'organizzazione, link al progetto e al codice nonché rimandi a progetti simili.
Licenses	Mostra la licenza o le licenze legate al progetto open source e le relative conseguenze.
Project Security	Fornisce informazioni sulla sicurezza e sulla vulnerabilità del progetto open source.
Code	Il numero di linee di codice nel corso del tempo viene visualizzato in un grafico, diviso secondo i linguaggi di programmazione.

[29] https://www.openhub.net/

Activity	Il numero di «commit» mensili è rappresentato in un grafico. Inoltre è presente anche un riepilogo degli ultimi 30 giorni e 12 mesi.
Community	Il numero di «contributor» attivi ogni mese è rappresentato in un grafico ed è presente anche una scala su cinque stelle con un rating dei progetti.

In aggiunta, Open Hub offre anche la possibilità di confrontare fra loro diversi progetti open source[30], creando così una rapida panoramica di quale progetto dispone della community più attiva, della durata maggiore di ulteriore sviluppo o della licenza adatta.

7.4 Repository speciali di codici

Le grandi organizzazioni dispongono di propri repository di codici. Un esempio particolare da evidenziare è il repository tedesco dedicato alle autorità opencode.de.[31]. Proprio i repository di altre amministrazioni pubbliche sono interessanti in questo contesto.

7.5 Directory OSS

Talvolta, organizzazioni open source gestiscono interessanti directory di software a codice sorgente aperto e aziende che possono supportarlo.
In Svizzera è il caso della «OSS Directory» di CH Open[32].

[30] https://www.openhub.net/p/_compare
[31] https://opencode.de/de
[32] https://www.ossdirectory.com/it/home

8. Modelli di supporto nell'impiego di OSS

L'impiego di software a codice sorgente aperto già presenti sul mercato può avvenire in sostanza in tre modi:

1. impiego senza supporto professionale;
2. impiego con supporto interno;
3. impiego con supporto a cura di fornitori esterni.

Di seguito vengono brevemente illustrati questi tre tipi di impiego insieme ai rispettivi vantaggi e svantaggi. Occorre decidere in base alla situazione quale sia lo scenario più ragionevole fra questi per il singolo caso. Il modello di supporto adatto dipende dalla rilevanza strategica del software a codice sorgente aperto, dall'integrazione tecnica e dalle risorse di personale disponibili.

Nel caso di software critici, si deve garantire in ogni caso il supporto professionale, interno o esterno. Nella pianificazione del supporto si dovrebbe considerare anche il ciclo di vita programmato. È possibile anche che il supporto venga offerto da più fornitori.

8.1 Impiego senza supporto professionale

Il software a codice sorgente aperto viene scaricato gratuitamente da Internet, installato e impiegato così com'è.

Vantaggi	- Costi contenuti - Messa in atto rapida
Svantaggi	- Nessuna garanzia di supporto - Nessuna pretesa di responsabilità
Rischio e copertura	Rischio elevato: non sono presenti contratti di supporto o garanzie e l'organizzazione possiede uno scarso o nessun know-how di sviluppatori.
Ambito d'impiego tipico	Software open source standard che può essere aggiornato indipendentemente da altri sistemi.

8.2 Impiego con supporto interno

Un'impresa o un'istituzione pubblica crea know-how o risorse per determinate soluzioni open source al fine di impiegarle nel lungo termine. Questa procedura si applica in particolare nei settori di attività critiche.

Vantaggi	- Elevata flessibilità grazie a know-how interno - Nessuna dipendenza da fornitori
Svantaggi	- Investimenti elevati e grande dispendio di tempo per la creazione di know-how - Costi fissi interni più elevati per il personale
Rischio e copertura	Rischio medio: il supporto dipende dal know-how e dalla disponibilità dell'IT interna.
Ambito d'impiego tipico	Sviluppi «in-house» e software a codice sorgente aperto a impiego strategico per cui è presente un know-how approfondito.

8.3 Impiego con supporto a cura di fornitori esterni

Viene coinvolto un fornitore esterno di open source che accompagna in modo professionale l'introduzione e la manutenzione del software a codice sorgente aperto. Questa procedura si applica in particolare anche nei settori di attività critiche in cui deve da subito essere presente un know-how approfondito del software. Può anche comprendere delle parti del rilascio o il rilascio complessivo.

Vantaggi	- Accesso diretto al know-how degli sviluppatori open source - Correzioni e ulteriore sviluppo in base al mandato - Scelta fra diversi fornitori di open source - Carattere vincolante, copertura dei rischi di compliance
Svantaggi	- Costi esterni dovuti ai fornitori di open source - Dipendenza dal fornitore di open source a livello di know-how
Rischio e copertura	Rischio basso: garanzia secondo la descrizione dell'incarico o il service level agreement.
Ambito d'impiego tipico	Software a codice sorgente aperto per attività critiche per cui è presente uno scarso o nessun know-how interno di sviluppatori.

9. Punto di contatto

Nell'Amministrazione federale non è presente un punto di contatto centrale dedicato alla diffusione di informazioni sulla tematica dell'open source. In linea di massima, la responsabilità dell'attuazione spetta agli uffici stessi.

Richieste di carattere generale sugli ausili OSS del set di documenti Em002 possono essere rivolte al settore TDT della Cancelleria federale: info.dti@bk.admin.ch.

Appendice

A. Modifiche rispetto alla versione precedente

- Eliminati i riferimenti all'OSIC.
- Il documento è stato rielaborato e riorganizzato rispetto alla versione precedente. Nelle situazioni descritte viene affrontato anche il tema dell'elaborazione e della modifica dei software a codice sorgente aperto.
- Molte parti sono state spostate negli ulteriori ausili del set di documenti Em002: p. es. licenze, glossario, FAQ.
- La parte relativa all'acquisto di software senza l'apporto di modifiche è stata ripresa dal promemoria UFCL del 2015.
- I riferimenti sono stati spostati nella guida strategica.
- Le parti relative a potenziale e sfide, reperibilità di possibili software a codice sorgente aperto, modelli di supporto e modelli commerciali con OSS sono rimaste nella guida pratica.

B. Riferimenti

I riferimenti al set di documenti Em002 sono elencati nella «Guida strategica» [Em002].

C. Abbreviazioni

L'elenco delle abbreviazioni è disponibile nel documento principale Em002.
Un glossario è presente nel documento sulle domande frequenti «Em002-6 FAQ».

D. Modelli commerciali con OSS

I software a codice sorgente aperto non costituiscono di per sé un modello commerciale poiché, diversamente dai software proprietari, vendere un software sotto una licenza open source non è un'opzione. Tuttavia le imprese hanno diverse possibilità di gestire modelli commerciali sulla base di software a codice sorgente aperto, perché quando p. es. per una soluzione di questo tipo si deve ricorrere al supporto di professionisti esterni, diventa necessario acquistare le rispettive prestazioni di servizi commerciali. Di seguito vengono illustrati i quattro modelli commerciali più frequenti legati ai software a codice sorgente aperto. Ulteriori aspetti di questi e altri modelli commerciali sono descritti nella guida BITKOM [BITKOM2023].

D.1 Servizi e prodotti basati su OSS

Sulla base di software a codice sorgente aperto, le imprese possono offrire servizi commerciali (come il «web hosting» o il «cloud computing») che con i software proprietari avrebbero costi notevolmente maggiori. Oggi la maggior parte delle start-up, dei portali online e dei fornitori di e-commerce basano le proprie piattaforme su software a codice sorgente aperto. Anche altre aziende del settore della tecnologia, come aziende di telecomunicazioni, fornitori di streaming o anche produttori di software proprietari, integrano dei software a codice sorgente aperto nei propri prodotti software e hardware e nei servizi online. Ciò consente alle imprese di offrire costantemente prestazioni innovative in un modo che sarebbe difficilmente possibile con i software proprietari.

D.2 Servizi per OSS

I cosiddetti fornitori di open source forniscono servizi esplicitamente per determinati software a codice sorgente aperto. Questi fornitori dispongono di esperti sviluppatori open source e possono pertanto offrire prestazioni di supporto, manutenzione, gestione, ulteriore sviluppo, consulenza, formazione e altri servizi per i software a codice sorgente aperto, di cui il richiedente può usufruire in base a un mandato o a un contratto d'appalto. Questi servizi per software a codice sorgente aperto possono essere messi pubblicamente a concorso in quanto non sussistono dipendenze dal rispettivo fornitore («vendor lock-in»). In questi bandi è importante prendere in considerazione i criteri corrispondenti in modo da selezionare i fornitori di servizi effettivamente competenti (cfr. numeri 4 e 8, [BITKOM2023] numero 3.3 e [Gu2024]).

D.3 Subscription

I servizi per software a codice sorgente aperto che vengono erogati in forma standardizzata e ricorrente come una specie di «service level agreement» (SLA) vengono denominati «subscription» (abbonamenti). Nel quadro di queste subscription, le aziende garantiscono per esempio aggiornamenti di sicurezza regolari, supporto via e-mail o telefono, compatibilità con altri prodotti software e hardware per mezzo di certificazioni, prestazioni di manutenzione a lungo termine e coperture contro pretese a livello legale (copyright, brevetti). In cambio i clienti pagano abbonamenti basati sul numero di postazioni di lavoro o CPU, in maniera analoga al pagamento dei diritti di licenza o di utilizzo dei software proprietari. Diversamente rispetto ai software proprietari, le subscription per i software a codice sorgente aperto non rappresentano una condizione per l'utilizzo del software, ma fungono da pagamento del valore aggiunto apportato dalle prestazioni del fornitore di open source.

D.4 Dual licensing

Se un’azienda dispone della proprietà intellettuale di una soluzione software e tutti i componenti open source integrati sono legati a una licenza permissiva, si può applicare una «dual licensing» (licenza duale) o anche licenza multipla. Da un lato, questa consente all’azienda sviluppatrice di pubblicare il software con una licenza open source «copyleft» e, dall’altro, di venderlo con una licenza proprietaria. Questa variante commerciale viene spesso denominata anche versione «enterprise» e tipicamente comprende certe funzionalità aggiuntive, come integrazioni esclusive di interfacce o il permesso per l’acquirente di integrare il software nei suoi prodotti proprietari. A seconda dell’ampiezza delle limitazioni della variante open source, nell’utilizzo di software con licenza duale è richiesta particolare attenzione poiché può essere indispensabile l’acquisto della versione «enterprise» e pertanto la dipendenza dal produttore è elevata analogamente al caso dei normali software proprietari.