Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Cancelleria federale CaF
Trasformazione digitale e governance delle TIC TDT

14.09.2026

Em002-1 Guida pratica per il software open source nell’Amministrazione federale

Versione 2.0-0be65ca

Indice

1. 1. Utilizzo della presente guida 3
1.1. Intento / parti interessate 3
2. Definizioni 5
3. Potenziale e sfide 7
3.1. Potenziale del software open source 7
3.2. Sfide nella gestione dell’OSS 9
4. Costellazioni per l’utilizzo e il lavoro con l’OSS 11
4.1. Livelli di maturità per l’OSS nell’Amministrazione federale 11
4.2. Utilizzo di software open source non modificato 13
4.3. Sviluppo con componenti open source e pubblicazione del codice sorgente 13
4.4. Collaborazione in progetti aperti (contribution) 13
5. Caratteristiche e scelta delle licenze open source 14
6. Acquisto di OSS 14
7. Reperibilità delle soluzioni OSS 14
7.1. Catalogo OSS 14
7.2. OSS Directory 14
7.3. alternativeTo 15
7.4. Repository open source 15
7.5. Open Hub 16
7.6. Repository di codice particolari 17
8. Modelli di supporto per l’utilizzo dell’OSS 17
8.1. Utilizzo senza supporto professionale 17
8.2. Utilizzo con supporto interno 18
8.3. Utilizzo con supporto di un fornitore esterno 18
9. Punto di contatto 19
10. Allegati 19
10.1. Modifiche rispetto alla versione precedente 19
10.2. Riferimenti 19
10.3. Abbreviazioni 19

10.4. Modelli di business con l’OSS 19

10.4.1. Servizi e prodotti basati sull’OSS 19

10.4.2. Servizi per l’OSS 19

10.4.3. Abbonamenti 20

10.4.4. Doppia licenza 20

> **NOTA** Il presente documento è una bozza in continua evoluzione e fa parte degli strumenti destinati a sostenere l’Amministrazione federale nella pubblicazione di codice open source. Per maggiori informazioni si veda il README principale.

1. 1. Utilizzo della presente guida

Secondo l’articolo 9 della legge federale sull’impiego di mezzi elettronici per l’adempimento dei compiti delle autorità (LMeCA), la Confederazione è tenuta a pubblicare il codice sorgente del software che sviluppa o fa sviluppare per adempiere i suoi compiti. [1] Il settore TDT della Cancelleria federale ha dedotto da questo mandato legale obiettivi generali, esposti nelle Linee guida strategiche per il software open source.

Il presente documento funge da guida pratica alle prescrizioni giuridiche e strategiche. Intende sostenere le unità amministrative della Confederazione nell’utilizzo, nell’acquisto e nella pubblicazione di software open source. Offre a tutte le persone interessate una panoramica completa del tema e rinvia a informazioni approfondite contenute negli altri strumenti.

A seconda delle conoscenze pregresse e delle esigenze potete tuttavia leggere le guide in modo selettivo prima di passare agli altri strumenti. La panoramica seguente vi aiuta a individuare il punto di partenza adeguato. In caso di dubbio si raccomanda di leggere dapprima la guida.

1.1. Intento / parti interessate

Introduzione generale al software open source

- Capitolo Definizioni
- Capitolo Potenziale e sfide
- Capitolo Reperibilità delle soluzioni OSS
- Em002-5 Promemoria Strumenti OSS
- Em002-3 Guida alle licenze OSS
- Em002-6 FAQ sull’OSS

Introduzione generale alla LMeCA per i decisori

- Em002-5 Promemoria Strumenti OSS
- Em002 Linee guida strategiche per il software open source nell’Amministrazione federale

Semplice acquisto di software destinato all’uso

- Capitolo Livelli di maturità per l’OSS nell’Amministrazione federale
- Capitolo Utilizzo di software open source non modificato
- Em002-7 Aspetti strategici degli acquisti e OSS
- Promemoria Acquisto di software e art. 9 LMeCA [KBB-MB]
- Direttive OSS negli acquisti [BBL-WL]

Nuovo sviluppo informatico e pubblicazione

1. RS 172.019: https://www.fedlex.admin.ch/eli/cc/2023/682/it

- Capitolo Sviluppo con componenti open source e pubblicazione del codice sorgente
- Capitolo Collaborazione in progetti aperti (contribution)
- Em002-2 Istruzioni per la pubblicazione di OSS
- Em002-2.1 Lista di controllo Valutazione preliminare OSS
- Em002-2.2 Lista di controllo Analisi e preparazione OSS
- Em002-2.3 Lista di controllo Rilascio e pubblicazione OSS
- Em002-4 Guida alle community OSS
- Em002-4.1 Lista di controllo Community OSS

Acquisto di software da realizzare (prestazione di servizi o appalto)

- Capitolo Livelli di maturità per l’OSS nell’Amministrazione federale
- Capitolo Sviluppo con componenti open source e pubblicazione del codice sorgente
- Capitolo Collaborazione in progetti aperti (contribution)
- Em002-7 Aspetti strategici degli acquisti e OSS
- Cassetta degli attrezzi dell’UFCL, comprese le direttive Open Source negli acquisti
- Piattaforma di apprendimento e modelli dell’amministrazione pubblica del CCAP (www.perimap.admin.ch), compreso Acquisto di software e art. 9 LMeCA [KBB-MB]
- Direttive OSS negli acquisti [BBL-WL]
- Em002-2 Istruzioni per la pubblicazione di OSS
- Em002-2.1 Lista di controllo Valutazione preliminare OSS
- Em002-2.2 Lista di controllo Analisi e preparazione OSS
- Em002-2.3 Lista di controllo Rilascio e pubblicazione OSS
- Em002-4 Guida alle community OSS
- Em002-4.1 Lista di controllo Community OSS

Accertamenti giuridici in vista della pubblicazione

- Em002-6 Domande frequenti sulla pubblicazione di OSS (FAQ-OSS)
- Em002-3 Guida alle licenze OSS

Principi di lavoro per i capiprogetto

- Documenti HERMES
- Em002-5 Promemoria Strumenti OSS

Contributi di software a terzi e collaborazioni

- Em002-4 Guida alle community OSS

- Em002-4.1 Lista di controllo Community OSS
- Em002-7 Aspetti strategici degli acquisti e OSS
- Em002-3 Guida alle licenze OSS

Il grafico seguente offre una panoramica dei documenti rilevanti per l’OSS nell’Amministrazione federale.

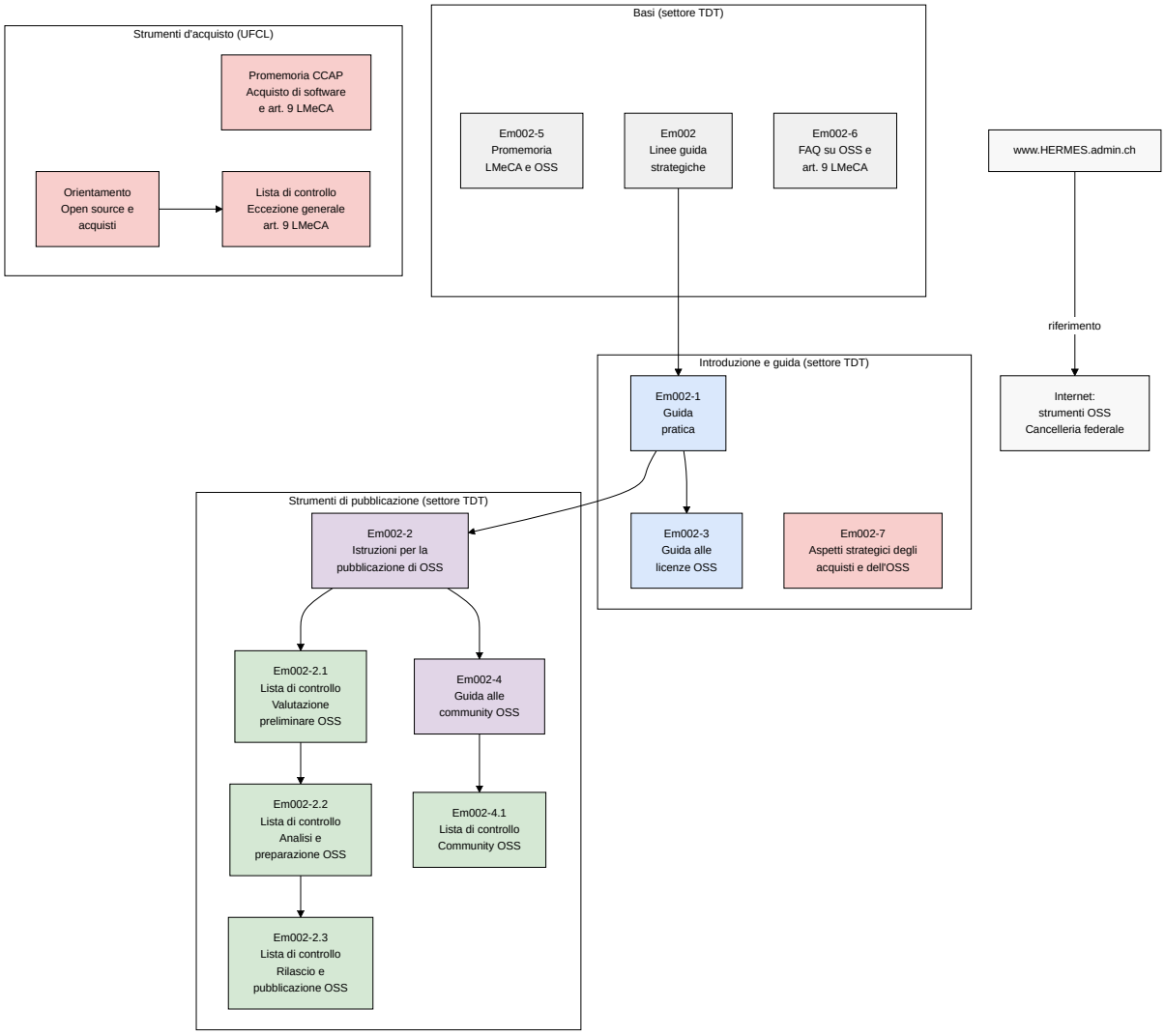


Figura 1: Panoramica degli strumenti OSS in relazione all’art. 9 LMeCA

2. Definizioni

Software open source

Un software è considerato software open source (OSS) quando è pubblicato sotto una delle circa 80 licenze riconosciute dall’Open Source Initiative OSI *[OSI2024]*. Concretamente, la definizione di open source stabilisce dieci criteri che tutte le licenze open source devono soddisfare *[Pe1999]*. Essi possono essere sostanzialmente riassunti in quattro punti:

1. Il software può essere utilizzato per qualsiasi scopo.
2. Il codice sorgente del software è liberamente accessibile.
3. Il software può essere copiato e diffuso senza restrizioni.

4. Il software può essere modificato e diffuso in forma modificata a determinate condizioni.

Software libero
Prima del termine «open source», negli anni Ottanta la Free Software Foundation (FSF) ha introdotto il concetto di «software libero». Nella sostanza il software libero soddisfa le condizioni del software open source, ma mira a garantire che il software resti sempre liberamente accessibile e, per quanto possibile, non venga integrato in software proprietario. [2]

Software proprietario
Nel caso del software proprietario un fornitore sviluppa il software e vende all'utente una licenza d'uso. Di regola l'utente non conosce il codice del software e non può né ridistribuirlo né modificarlo. Può utilizzare il software soltanto secondo le condizioni di licenza (p. es. End-User Licence Agreement, EULA) dietro pagamento di tasse di licenza. Il software può ad esempio essere autorizzato soltanto per un determinato numero di utenti o un determinato numero di processori. [3] In cambio delle spese di manutenzione, di regola annuali, il fornitore si impegna a correggere gli errori entro un termine adeguato e a sviluppare continuamente il software.

Licenza
Una licenza è un documento che contiene prescrizioni vincolanti per l'utilizzo e la diffusione di un software. Nell'ambito open source si utilizzano di regola licenze conformi all'OSI. [4]

Community
Il concetto di community nel contesto del software open source è definito in senso ampio. Può trattarsi di un ecosistema informale o di una struttura dotata di governance che sviluppa il software. Il documento Em002-4 Guida alle community OSS tratta questo tema. I membri della community possono gestire congiuntamente il prodotto, svilupparlo, testarlo, tradurlo, fornire riscontri o semplicemente utilizzare il software.

Diritti di terzi
Anche per il software open source possono esistere diritti di protezione (diritto d'autore, diritto dei marchi, diritto dei brevetti), che possono essere fatti valere da terzi, p. es. utilizzando codice sorgente creato da terzi.

Eccezioni secondo la LMeCA
Le eccezioni secondo l'art. 9 cpv. 1 LMeCA sono i diritti di terzi e i motivi rilevanti per la sicurezza. Entrambi sono trattati nel capitolo 3 di Em002-2 Istruzioni per la pubblicazione di software open source.

Codice sorgente
In informatica il codice sorgente (o testo sorgente) è il testo leggibile dall'essere umano di un programma per computer, redatto in un linguaggio di programmazione. [5]

Pubblicazione
Per pubblicazione si intende in primo luogo la messa a disposizione del codice sorgente sotto una licenza aperta conforme all'OSI. Per ragioni pratiche essa comprende di regola anche la documentazione e le istruzioni automatizzate per costruire l'applicazione a partire dal codice sorgente. Anche i test automatizzati e i documenti di test ne fanno solitamente parte. Affinché una community possa svilupparsi si utilizza una piattaforma che assume poi diverse funzioni di una piattaforma di sviluppo (build, test, supporto).

Sovranità digitale
Utilizziamo la definizione contenuta nel rapporto del DFAE sulla sovranità digitale [St2024]: «La sovranità digitale di uno Stato o di un'organizzazione comprende necessariamente il controllo completo dei dati memorizzati ed elaborati nonché la decisione autonoma su chi possa accedervi.

---

2. Software libero — Wikipedia
3. Software proprietario — Wikipedia
4. Open Source Initiative
5. Codice sorgente — Wikipedia

Essa comprende inoltre la capacità di sviluppare, modificare, controllare e completare autonomamente componenti e sistemi tecnologici.»

Repository
Un repository è un luogo di archiviazione utilizzato da uno strumento di gestione delle versioni per i file e i metadati relativi alla base di codice. I repository consentono a più collaboratori di lavorare sugli stessi file e di archiviare diverse versioni. La maggior parte consente inoltre l'issue tracking, la gestione delle release, le build automatizzate e la documentazione.

Fork
Un fork [6] di un progetto open source è il processo di avviare uno sviluppo indipendente del progetto originario.

Contributor Licence Agreement (CLA)
Un Contributor Licence Agreement [7] è un documento che descrive le condizioni alle quali è possibile apportare proprietà intellettuale a un progetto o a un'iniziativa.

Source code management (SCM)
La gestione del codice sorgente [8], detta anche gestione delle versioni, è uno strumento che gestisce efficacemente modifiche e numeri di versione, in particolare nello sviluppo software. Si è affermato il software libero Git [9] per la gestione di SCM distribuiti.

Open source software development (OSSD)
Nello sviluppo di software open source (OSSD) [10], oltre alla pubblicazione del codice sorgente, l'intero processo di sviluppo si svolge pubblicamente. Tutto, dai requisiti (issue) al codice sorgente, è trasparente. L'OSSD è supportato da strumenti di comunicazione pubblici (mailing list, forum ecc.), da un sistema di gestione delle versioni (Git), da elenchi di errori e funzionalità, da una roadmap e da strumenti per sviluppatori.

Software Package Data Exchange (SPDX)
Software Package Data Exchange (SPDX) [11] descrive un formato standard per la 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 soggetto a una licenza con clausola di copyleft, ogni modifica o estensione del codice sorgente deve essere nuovamente pubblicata sotto la licenza del software open source modificato (cfr. anche Em002-3 Guida alle licenze OSS).

3. Potenziale e sfide

Nell'odierno mondo informatico il software open source presenta un potenziale enorme. Al tempo stesso l'impiego di soluzioni open source comporta anche sfide. Questi due aspetti sono illustrati più in dettaglio nella sezione seguente. Le indicazioni si basano tra l'altro sui risultati dello Studio Open Source Svizzera 2018, 2021 e 2024 dell'Università di Berna, nel quale gli interpellati hanno fornito informazioni sui vantaggi e sugli svantaggi riconosciuti del software open source.

3.1. Potenziale del software open source

I punti seguenti descrivono il potenziale che può essere realizzato impiegando e pubblicando software open source.

6. Fork (sviluppo software) — Wikipedia
7. Contributor License Agreement — Wikipedia
8. Controllo versione — Wikipedia
9. Git — Wikipedia
10. Open-source software development — Wikipedia
11. Software Package Data Exchange — Wikipedia

1. Sovranità digitale
Si tratta della capacità di utilizzare e controllare i servizi digitali. La sovranità digitale si estende all’intero ciclo di vita di un sistema digitale. Il software open source promuove la sovranità digitale, poiché garantisce intercambiabilità, libertà di configurazione e influenza. [12]

2. Nessuna tassa di licenza
Per l’utilizzo di software open source non sorgono costi di licenza. Nell’acquisire pacchetti standard open source complessi può tuttavia essere opportuno acquistare abbonamenti di supporto a pagamento.

3. Risparmi grazie alla cooperazione con altri utenti
Poiché un software soggetto a una licenza open source può essere utilizzato e sviluppato senza restrizioni, i costi degli sviluppi ulteriori possono essere condivisi oppure è possibile riprendere sviluppi aggiuntivi esistenti di altre unità amministrative secondo il principio «sviluppare una volta — utilizzare più volte». Al tempo stesso si può beneficiare dell’esperienza e degli sviluppi altrui.

4. Costituzione di community e condivisione delle conoscenze
Il software open source facilita la costituzione di community e la condivisione delle conoscenze, per esempio tra i diversi livelli federali in Svizzera. Il software può essere sviluppato congiuntamente, gli errori possono essere corretti e le esperienze individuali condivise. L’accresciuto scambio di conoscenze tra diverse unità amministrative porta a una migliore comprensione di ciò a cui lavorano gli altri, così da evitare doppioni e permettere alle soluzioni migliori di diffondersi.

5. Minore dipendenza dai fabbricanti
Il vendor lock-in (dipendenza dai fornitori di software) è considerato molto elevato in informatica. L’impiego di software soggetto a licenza open source riduce il vendor lock-in, poiché esercizio, manutenzione, supporto, sviluppo ulteriore e altre prestazioni per il software open source possono essere messi a concorso pubblicamente. Invece di riacquistare integralmente un intero sistema a causa di un vendor lock-in, è possibile mettere a concorso separatamente anche estensioni o prestazioni del ciclo di vita per una soluzione OSS. Ciò consente di risparmiare notevoli costi di migrazione.

6. Standard aperti e interoperabilità
Con il software open source la compatibilità con altre soluzioni software e sistemi informatici (interoperabilità) è di regola maggiore rispetto al software proprietario. Le soluzioni open source utilizzano inoltre quasi esclusivamente formati di dati aperti, per cui possono essere facilmente scambiate con altri sistemi.

7. Trasparenza sulla struttura del software
Poiché il software è disponibile anche sotto forma di codice sorgente, la sua qualità può essere verificata, per esempio mediante revisioni esterne. Inoltre, sulla base del codice sorgente è possibile creare documentazione (p. es. in vista di nuovi bandi per prestazioni di sviluppo o della sostituzione di software open source al termine della sua vita utile).

8. Sicurezza e fiducia grazie alla trasparenza
Data l’apertura del codice sorgente, la sicurezza del software open source può essere superiore a quella del software proprietario. È inoltre molto più difficile inserire backdoor e altre falle nel software open source, il che accresce la fiducia nel software.

9. Maggiore qualità e modularità del codice
La qualità del software open source può essere superiore a quella del software proprietario, poiché la motivazione a scrivere codice di qualità è verosimilmente maggiore quando gli sviluppatori sanno che il loro codice sorgente sarà pubblicato. Inoltre le soluzioni open source tendono a essere fortemente modulari, per cui singoli moduli possono essere facilmente sostituiti e i restanti

---

12. Utilizziamo la definizione di *[St2024]*: «La sovranità digitale di uno Stato o di un’organizzazione comprende necessariamente il controllo completo dei dati memorizzati ed elaborati nonché la decisione autonoma su chi possa accedervi. Essa comprende inoltre la capacità di sviluppare, modificare, controllare e completare autonomamente componenti e sistemi tecnologici.»

continuare a essere utilizzati. [13]

10. Adattamento semplice
L’accesso al codice sorgente consente agli utenti di effettuare sviluppi ulteriori autonomamente o tramite fornitori esterni. Il software può così essere adattato rapidamente alle proprie esigenze.

11. Innovazione e integrazione rapide
Il software open source viene sviluppato rapidamente e continuamente dalla community. Nuove tecnologie e nuovi standard di dati vengono per esempio spesso pubblicati come librerie di programmazione open source. Ciò consente la rapida realizzazione di soluzioni software innovative.

12. Attrattiva del datore di lavoro
L’impiego di moderne tecnologie open source e la collaborazione informale con community internazionali favoriscono la motivazione dei collaboratori e accrescono così l’attrattiva del datore di lavoro, il che agevola a sua volta il reclutamento di giovani professionisti qualificati.

3.2. Sfide nella gestione dell’OSS

Di seguito sono descritte le sfide tipiche che si incontrano nella pratica con il software open source e sono delineate alcune possibili soluzioni. Non trattiamo in questa sede le sfide generali che riguardano tutti i software (p. es. la necessità di verificare la cibersicurezza e di assicurare un supporto adeguato per il software critico).

1. Elevati costi di passaggio dovuti a dipendenze esistenti
La sostituzione di software proprietario con soluzioni open source può risultare molto onerosa se l’applicazione è strettamente integrata nel sistema informatico esistente mediante interfacce o altre dipendenze. Questi costi di passaggio fanno sì che l’introduzione di software open source sia spesso conveniente soltanto al termine del ciclo di vita del software proprietario.

2. Funzionalità mancanti o assenza di una soluzione open source adeguata
Nell’acquisto di software standard può accadere che non esista alcuna alternativa open source alla soluzione proprietaria (precedente), oppure soltanto una funzionalmente insufficiente. Come possibile soluzione vi è la facoltà di sviluppare le funzionalità mancanti insieme ad altri utenti del software open source e di aggiungerle al progetto open source. [14]

3. Maggiori costi d’integrazione
I risparmi a breve termine derivanti dalle licenze non dovute nell’impiego di software open source sono spesso compensati da maggiori costi d’integrazione. È quindi importante riconoscere che l’impiego professionale di software open source non è «gratuito», ma genera effettivamente costi interni o esterni.

4. Responsabilità e supporto poco chiari
Il software open source è spesso criticato per la mancanza di supporto e manutenzione da parte di imprese esterne. Esiste tuttavia ormai una molteplicità di fornitori che offrono prestazioni commerciali (supporto a lungo termine, manutenzione continua, sviluppo ulteriore, responsabilità e garanzia ecc.) per le soluzioni open source. I diversi modelli di business con il software open source sono illustrati nell’allegato D e i diversi modelli di supporto nel capitolo 8. Una panoramica

13. Un ulteriore aspetto di sicurezza è che le librerie utilizzate (comprese le dipendenze) sono più facili da documentare rispetto al software proprietario. Ciò significa a sua volta che, quando emergono vulnerabilità, esse possono essere individuate e corrette rapidamente (cfr. anche SBOM).
14. Per tali nuove funzionalità sono competenti i responsabili di progetto o di applicazione. Le modalità precise sono stabilite dal progetto. Em002-2 ed Em002-4 aiutano nelle decisioni di dettaglio. A seconda dell’entità della funzionalità può non essere opportuno percorrere integralmente le istruzioni. Vanno osservati gli aspetti del diritto degli acquisti pubblici relativi alle risorse.

aggiornata dei fornitori di servizi open source e delle loro prestazioni figura nell'Open Source Directory (https://www.ossdirectory.com) o nello Studio OSS 2024 [15]*[OSS2024]*.

5. Mercato ristretto con pochi fornitori

Poiché determinate applicazioni open source, come quelle desktop, richiedono di regola poco o nessun supporto e manutenzione, il mercato dei fornitori open source di supporto e manutenzione può essere ristretto. In tal caso un bando per prestazioni di servizi relative a software open source può contribuire a creare un mercato che accresca la concorrenza tra i fornitori. L'esistenza di un'ampia community di sviluppo per il software in questione e la qualità della documentazione di sviluppo rivestono importanza pratica nel mettere a concorso tali prestazioni.

6. Scarsa visibilità

Le community dei progetti open source si concentrano soprattutto sullo sviluppo del prodotto e raramente sul marketing. Per contro i fabbricanti di software proprietario investono massicciamente nella commercializzazione e nella vendita dei loro prodotti. Questa mancanza di pubblicità per il software open source crea spesso la falsa impressione che non esistano alternative ai prodotti proprietari. Vi sono però piattaforme come alternativeTo [16] che offrono una panoramica delle soluzioni per compiti specifici (maggiori informazioni sulle alternative OSS nel capitolo «Reperibilità delle soluzioni OSS»).

7. Mancanza di accettazione da parte degli utenti finali

A causa di interfacce utente differenti, funzioni mancanti, scarsa facilità d'uso e poca pubblicità per il software open source, determinati prodotti open source soffrono di una mancanza di accettazione da parte degli utenti finali. Adeguate offerte di comunicazione, documentazione e formazione possono contrastare questa sfida. Anche il fatto che il software open source possa essere sviluppato secondo le esigenze degli utenti può portare a una maggiore accettazione.

8. Competenze interne scarse o inesistenti

Le soluzioni open source evolvono rapidamente e richiedono al tempo stesso una profonda comprensione tecnica. Può quindi accadere che i collaboratori interni dispongano di poche o nessuna competenza su determinati software open source. Il know-how OSS interno può essere sviluppato mediante il perfezionamento e possibilità di autoapprendimento tramite fonti online, la creazione di wiki interni ecc.

9. Difficile valutazione del futuro

Per chi è esterno a una community open source è spesso difficile riconoscere come il progetto si svilupperà in futuro. È quindi importante poter valutare realisticamente lo sviluppo futuro di una soluzione open source. Il capitolo «Open Hub» presenta a tal fine la piattaforma Open Hub, che consente di valutare l'attività e l'eterogeneità della community di sviluppo. Ciò permette di apprezzare meglio lo sviluppo futuro di un progetto open source.

10. Incertezze giuridiche

La molteplicità di licenze open source diverse e il numero finora esiguo di sentenze giudiziarie su questioni interpretative possono talvolta generare incertezze giuridiche con il software open source. La presente guida pratica intende offrire una panoramica delle principali licenze open source nonché delle loro caratteristiche e compatibilità. Fonti per risposte approfondite a questioni giuridiche figurano nei documenti Em002-3 Guida alle licenze OSS e Em002-6 FAQ sull'OSS e l'art. 9 LMeCA.

11. Offerta troppo ampia di software open source

Il numero di prodotti software open source è fortemente aumentato negli ultimi anni. I potenziali utenti di software open source lamentano pertanto la molteplicità di soluzioni open source presenti sul mercato. Per questo motivo il capitolo «Reperibilità di…» presenta due piattaforme, alternativeTo e Open Hub, che consentono di confrontare le soluzioni open source.

15. Studio OSS
16. https://alternativeto.net/

12. Onere di coordinamento nello sviluppo congiunto
Quando l’Amministrazione federale partecipa a progetti comuni e vi contribuisce, ne deriva un onere di coordinamento all’interno di tali collaborazioni, ma anche quando occorre coinvolgere una community. Idealmente questo onere supplementare dovrebbe comunque risultare conveniente grazie a una soluzione migliore, a una maggiore accettazione o a una partecipazione accresciuta di terzi.

4. Costellazioni per l’utilizzo e il lavoro con l’OSS

4.1. Livelli di maturità per l’OSS nell’Amministrazione federale

Nell’impiego e nello sviluppo di OSS nell’Amministrazione federale esistono diverse costellazioni riguardo agli obblighi connessi alle licenze open source. Un tipico modello di maturità [17] per l’introduzione dell’open source nelle organizzazioni può presentarsi così:

1. Denial — nessun impiego di software open source, o impiego inconsapevole
2. Consumption — impiego passivo di software open source
3. Participation — impegno nelle community open source
4. Contribution — contributi pragmatici a progetti open source
5. Leadership — impegno strategico con l’open source per creare valore [18]

Costellazione	Conseguenza
Il semplice **utilizzo** di software open source esistente (senza modifica del codice).	Non sussiste **alcun obbligo** di condividere il codice sorgente.
Lo **sviluppo interamente nuovo** di software destinato a essere pubblicato sotto una licenza open source.	**In tal caso sussiste piena libertà quanto alla licenza sotto la quale il software viene pubblicato.** Per la Confederazione va consultato il documento Em002-3 Guida alle licenze OSS. Vanno osservati i documenti Em002-2 Istruzioni per la pubblicazione di OSS ed Em002-4 Guida alle community OSS.

17. Modello di maturità ispirato a https://todogroup.org/resources/guides/a-guide-to-outbound-open-source-software/#maturity-levels
18. La creazione non è menzionata, poiché l’art. 9 LMeCA la esige direttamente.

Costellazione	Conseguenza
Sviluppo ulteriore (interno o esterno) con componenti open source esistenti (con o senza effetto copyleft [19]), fintanto che il software è utilizzato **esclusivamente all'interno della stessa organizzazione** e non è ridistribuito.	I componenti OSS possono essere combinati a piacimento con componenti di terzi. Dalla licenza non deriva in linea di principio alcun obbligo di pubblicazione, fintanto che il software è utilizzato esclusivamente all'interno della stessa organizzazione e non è diffuso (eccezione: AGPL). Poiché non è giuridicamente chiarito in modo definitivo fin dove si estenda il concetto di «diffusione all'interno della stessa organizzazione», questo scenario si applica ai componenti OSS con effetto copyleft soltanto in casi eccezionali. In tali casi consultate il servizio giuridico della vostra organizzazione. Cfr. Em002-3 Guida alle licenze OSS ed Em002-6 FAQ sull'OSS e l'art. 9 LMeCA. Vanno osservati i documenti Em002-2 Istruzioni per la pubblicazione di OSS ed Em002-4 Guida alle community OSS. **Poiché secondo l'art. 9 LMeCA sussiste un obbligo di pubblicazione, le modifiche devono essere nuovamente pubblicate. Il modo più semplice è farlo come modifiche al progetto open source esistente.**
Sviluppo ulteriore (interno o esterno) **esclusivamente con** componenti open source esistenti **senza effetto copyleft**, se il software deve essere diffuso all'esterno (p. es. ai Cantoni).	I componenti OSS possono essere combinati a piacimento con componenti di terzi e dalle licenze non deriva alcun obbligo di pubblicazione (cfr. anche Em002-3 Guida alle licenze OSS). **Nel quadro dell'art. 9 LMeCA la pubblicazione è tuttavia obbligatoria. Idealmente essa dovrebbe avvenire tramite il progetto esistente (nessun fork).** Vanno osservati i documenti Em002-2 Istruzioni per la pubblicazione di OSS ed Em002-4 Guida alle community OSS.
Sviluppo ulteriore (interno o esterno) con componenti open source esistenti **con effetto copyleft**, se il software deve essere diffuso all'esterno (p. es. ai Cantoni).	**Sussiste un obbligo di pubblicazione** in virtù della licenza copyleft (cfr. anche Em002-3 Guida alle licenze OSS riguardo all'effetto copyleft). Vanno osservati i documenti Em002-2 Istruzioni per la pubblicazione di OSS ed Em002-4 Guida alle community OSS.
Contributo a un progetto open source esistente.	**Secondo la LMeCA sussiste di regola un obbligo di pubblicazione. Si applicano la governance e la licenza del progetto.** Vanno osservati i documenti Em002-2 Istruzioni per la pubblicazione di OSS ed Em002-4 Guida alle community OSS.
Sviluppo congiunto di un progetto nuovo o esistente con una community e ripartizione dei costi.	**Secondo la LMeCA sussiste di regola un obbligo di pubblicazione.** I documenti Em002-2 Istruzioni per la pubblicazione di OSS, Em002-3 Guida alle licenze OSS ed Em002-4 Guida alle community OSS servono **a impostare il progetto e la sua governance.**

19. Spiegazione del termine «copyleft» in Em002-3 Guida alle licenze OSS

Costellazione	Conseguenza
Tutte le varianti di sviluppo o di sviluppo ulteriore sono realizzate **tramite un fornitore o terzi incaricati.**	**Secondo la LMeCA sussiste di regola un obbligo di pubblicazione.** Sulla base di Em002-2 Istruzioni per la pubblicazione di OSS, Em002-3 Guida alle licenze OSS ed Em002-4 Guida alle community OSS. Se possibile, i presupposti per la pubblicazione andrebbero creati già nel processo d'acquisto e, se necessario, i requisiti per il rilascio del software andrebbero comunicati al fornitore o ai terzi.

4.2. Utilizzo di software open source non modificato

Nell'utilizzo di software open source non è in linea di principio determinante sapere sotto quale licenza open source esso sia pubblicato, poiché tutte le licenze open source (cfr. capitolo 2 «Definizioni») consentono l'uso illimitato del software open source, indipendentemente dal numero di postazioni di lavoro, di utenti collegati simultaneamente, di server e processori ecc. su cui il software è impiegato.

Nell'utilizzo di software open source non modificato non sorge **alcun obbligo di pubblicazione**, nemmeno per i programmi soggetti a una licenza con effetto copyleft rigoroso (eccezione: licenza AGPL, Affero General Public Licence).

Fintanto che il software open source è introdotto ed esercito da fornitori di prestazioni interni alla Confederazione, ciò può avvenire **senza procedura di acquisto pubblico**, poiché «la sola licenza OSS di regola non costa nulla al servizio acquirente e non è quindi di per sé rilevante sotto il profilo degli acquisti pubblici».

Se per la manutenzione, il supporto e altre prestazioni relative al software open source è incaricato un fornitore esterno, vanno osservate le prescrizioni del diritto degli acquisti pubblici. Diverse varianti di supporto sono descritte nel capitolo 9. È inoltre importante utilizzare criteri di idoneità e di aggiudicazione adeguati nell'acquisto di software open source. Le basi al riguardo sono illustrate nel capitolo 8.

4.3. Sviluppo con componenti open source e pubblicazione del codice sorgente

Con l'obbligo di pubblicazione secondo l'art. 9 LMeCA occorre qui seguire la procedura descritta in Em002-2 Istruzioni per la pubblicazione di OSS. Se necessario vanno consultati anche i documenti Em002-3 Guida alle licenze OSS ed Em002-4 Guida alle community OSS. L'obiettivo è pubblicare il progetto in modo controllato sulla base delle tre liste di controllo *Em002-2.1*, *Em002-2.2*, *Em002-2.3*.

4.4. Collaborazione in progetti aperti (contribution)

Le possibili riflessioni sul contributo dell'Amministrazione federale a progetti aperti già esistenti sono esposte nel capitolo 4.2 della guida BITKOM *[BITKOM2024]*. [20] A seconda dell'importanza del progetto per l'Amministrazione federale occorre valutare quanta responsabilità il rispettivo ufficio intenda e possa assumere.

La collaborazione in progetti aperti e lo sviluppo diretto in progetti aperti possono parimenti avvenire sulla base dei documenti per la pubblicazione. Al centro figurano Em002-4 Guida alle community OSS e la relativa lista di controllo *Em002-4.1*.

20. Cfr. https://www.bitkom.org/Bitkom/Publikationen/Bitkom-Leitfaden-zu-Open-Source-Software-20.html, capitolo 4.2.

5. Caratteristiche e scelta delle licenze open source

Le caratteristiche e la scelta delle licenze open source sono descritte nel documento Em002-3 Guida alle licenze OSS.

6. Acquisto di OSS

Le questioni strategiche relative agli acquisti e all'OSS sono trattate in Em002-7 Aspetti strategici degli acquisti e del software open source.

Il Centro di competenza per gli acquisti pubblici della Confederazione (CCAP) ha pubblicato un nuovo promemoria intitolato *Acquisto di software e art. 9 LMeCA [KBB-MB]* nonché due documenti «Criteri tipo — acquisto LMeCA e open source» e «Testi contrattuali tipo — sviluppo software».

Ulteriore sostegno è disponibile sull'intranet dell'UFCL (accessibile soltanto dalla rete federale) con le *direttive Open Source negli acquisti [BBL-WL]* e la *lista di controllo per l'eccezione forfettaria secondo l'art. 9 LMeCA [BBL-CL]*.

L'idoneità dell'open source all'uso aziendale è trattata nel documento *Selection Criteria for Enterprise-Ready Open Source Software [Gu2024]* [21] o nella guida Bitkom. [22]

7. Reperibilità delle soluzioni OSS

Diverse piattaforme consentono alle organizzazioni interessate di trovare e analizzare software open source prima dell'utilizzo o dell'acquisto. I dati così ottenuti costituiscono una base per decidere in merito all'impiego o all'acquisto di software open source.

Una raccolta di requisiti minimi e, se del caso, un'analisi di mercato dovrebbero in ogni caso essere d'ausilio.

7.1. Catalogo OSS

Le autorità federali sono tenute a pubblicare il codice sorgente del software che sviluppano o fanno sviluppare per adempiere i loro compiti. Il catalogo OSS [23] elenca tutto il software sviluppato dalle autorità federali dal 1° gennaio 2024 e pubblicato come open source. Offre una panoramica del software disponibile e agevola il reperimento e il riutilizzo di soluzioni open source.

Se viene creato un nuovo repository o una nuova organizzazione, ciò va segnalato per l'inserimento nel catalogo OSS all'indirizzo e-mail opensource@bk.admin.ch.

Catalogo OSS: https://www.opensource.admin.ch

7.2. OSS Directory

L'OSS Directory gestito da CH Open [24] agevola il reperimento di fornitori qualificati che offrono supporto professionale per il software open source.

I fornitori open source mostrano, mediante casi di successo, quali utenti (clienti) hanno aiutato a introdurre determinati prodotti open source. Il Directory propone inoltre notizie open source aggiornate provenienti da piattaforme esistenti, notizie di rilievo selezionate, eventi, community, offerte di lavoro e

21. Cfr. *[Gu2024]*
22. Cfr. Raccomandazioni pratiche per il software open source | Guida 2025 | Bitkom e. V., capitolo 3.3
23. https://www.opensource.admin.ch/
24. https://www.ossdirectory.com/it/home

articoli per la condivisione delle conoscenze. L'OSS Directory è disponibile in tedesco, inglese, francese e italiano. Tutte le voci possono essere aggiunte e aggiornate direttamente dagli utenti registrati.

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

7.3. alternativeTo

Uno strumento pratico per trovare alternative a un software è «alternativeTo - Crowdsourced software recommendations». [25] Come suggerisce il nome, le alternative e le relative valutazioni sono nate tramite crowdsourcing, ossia grazie ai contributi di molti singoli utenti. alternativeTo distingue quattro tipi di soluzioni software, di cui soltanto il primo è considerato software open source ai sensi dell'art. 9 LMeCA:

Free open source
: Software pubblicato sotto una licenza open source.

Free
: Software gratuito (freeware) il cui codice sorgente non è però liberamente disponibile e non può essere modificato.

Freemium
: Tale software offre una versione gratuita e una premium, con tutte le funzionalità importanti già disponibili nella versione gratuita. Non rientra in questa categoria il software che può essere testato gratuitamente per un periodo determinato (p. es. una versione di prova di 30 giorni), che ricade nella categoria «Commercial».

Commercial
: Si tratta di software proprietario per il cui utilizzo sono riscosse tasse di licenza.

7.4. Repository open source

Attualmente la piattaforma di sviluppo open source più popolare al mondo, con oltre 30 milioni di utenti e oltre 100 milioni di repository di codice sorgente, è GitHub. [26] Oggi praticamente tutte le imprese informatiche, in particolare i fabbricanti di software proprietario, nonché molti altri tipi di imprese e organizzazioni sono presenti su GitHub. Sempre più autorità pubbliche — una dozzina circa dalla Svizzera [27] — pubblicano inoltre il proprio software open source su «GitHub and Government» all'indirizzo https://government.github.com. Questa procedura può essere svolta anche con altri repository.

Esempi di altri repository sono:

- GitLab
- Bitbucket
- SourceForge
- LaunchPad

Su GitHub Insights è possibile consultare numerose statistiche rilevanti di un progetto open source su GitHub:

25. https://alternativeto.net/
26. https://octoverse.github.com
27. https://government.github.com/community/#switzerland

Pulse
Panoramica delle attività recenti di un progetto open source su GitHub: sintesi delle informazioni principali su intensità di sviluppo, eterogeneità della community, segnalazioni aperte, miglioramenti (pull request) ecc.

Contributors
Indicazione di quali sviluppatori sono stati attivi e quando. È un indicatore importante per sapere se tutto dipende da una sola persona o se dietro vi è una community più ampia.

Commits
Indicazione di quali contributi sono stati apportati a questo progetto open source e in quale periodo.

Code frequency
Visualizzazione di quanto codice sorgente è stato aggiunto o rimosso e quando.

Dependency graph
Dipendenze del progetto open source da altro software open source (p. es. rilevante per individuare vulnerabilità di sicurezza e aggiornamenti).

Network
Indicazione di quando quale sviluppatore ha contribuito a quale ramo di sviluppo.

Forks
Elenco di tutte le copie del progetto open source su GitHub. Indicatore della popolarità e della diffusione del software open source.

7.5. Open Hub

Se occorre raccogliere informazioni su una soluzione open source che non è necessariamente sviluppata su GitHub, Open Hub di Black Duck rappresenta una buona opzione. [28] Per circa mezzo milione di progetti open source vengono riassunte in modo chiaro numerose informazioni importanti:

Project Summary
Una breve descrizione del progetto open source.

In a Nutshell
I fatti più importanti su un progetto open source, quali il numero di commit, di collaboratori e di righe di codice, nonché il linguaggio di programmazione più utilizzato, la data del primo commit e dell’ultima modifica. È inoltre indicata una valutazione della base di codice e delle dimensioni del team di sviluppo (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, collegamenti al progetto e al codice nonché rinvii a progetti analoghi.

Licences
Indica sotto quale licenza o quali licenze il progetto open source è licenziato e quali conseguenze ne derivano.

Project Security
Fornisce indicazioni sulla sicurezza e sulle vulnerabilità del progetto open source.

Code
Un grafico che mostra il numero di righe di codice nel tempo, suddiviso per linguaggio di programmazione.

Activity
Un grafico che mostra il numero di commit al mese. È inoltre fornita una sintesi degli ultimi 30 giorni e 12 mesi.

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

Community
: Un grafico che mostra il numero di collaboratori attivi al mese. È inoltre visualizzata una valutazione del progetto su una scala a cinque stelle.

Open Hub consente anche di confrontare diversi progetti open source. [29] Ciò fornisce rapidamente una panoramica di quale progetto abbia la community più attiva, il tempo di sviluppo più lungo o la licenza più adeguata.

7.6. Repository di codice particolari

Le organizzazioni di maggiori dimensioni dispongono di propri repository di codice. Merita menzione in particolare il repository tedesco per le autorità opencode.de. [30] Rivestono qui particolare interesse i repository di altre amministrazioni pubbliche.

8. Modelli di supporto per l’utilizzo dell’OSS

Il software open source già disponibile sul mercato può essere utilizzato fondamentalmente in tre modi:

1. Utilizzo senza supporto professionale
2. Utilizzo con supporto interno
3. Utilizzo con supporto di un fornitore esterno.

Questi tre tipi di utilizzo e i relativi vantaggi e svantaggi sono illustrati brevemente qui di seguito. Quale di questi scenari sia il più opportuno in un caso concreto va deciso caso per caso. Il modello di supporto adeguato dipende dalla rilevanza strategica del software open source, dall’integrazione tecnica e dalle risorse di personale disponibili.

Per il software critico il supporto deve in ogni caso essere fornito in modo professionale, internamente o esternamente. Anche il ciclo di vita previsto dovrebbe avere un ruolo nella pianificazione del supporto. Può anche darsi che il supporto sia ottenuto da più fornitori.

8.1. Utilizzo senza supporto professionale

Il software open source è disponibile su Internet per essere scaricato, installato e utilizzato gratuitamente.

Vantaggi

- Costi contenuti
- Attuazione rapida

Svantaggi

- Nessun supporto garantito
- Nessuna pretesa di responsabilità

Rischio e copertura
: Rischio elevato: non esistono contratti di supporto né garanzie e nell’organizzazione vi sono poche o nessuna competenza di sviluppo.

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

Ambito d’applicazione tipico
Software open source standard che può essere aggiornato indipendentemente da altri sistemi.

8.2. Utilizzo con supporto interno

Un’impresa o un’organizzazione dell’ente pubblico sviluppa competenze e risorse su determinate soluzioni open source per un utilizzo a lungo termine. Questo approccio è particolarmente diffuso negli ambiti critici per l’attività.

Vantaggi

- Elevata flessibilità grazie al know-how interno
- Nessun lock-in del fornitore

Svantaggi

- Elevato investimento e tempo necessari per sviluppare le competenze
- Maggiori costi fissi interni per il personale

Rischio e copertura
Rischio medio: il supporto dipende dal know-how e dalla disponibilità dell’informatica interna.

Ambito d’applicazione tipico
Sviluppi propri, software open source impiegato strategicamente e per il quale è disponibile un know-how approfondito.

8.3. Utilizzo con supporto di un fornitore esterno

Viene coinvolto un fornitore open source esterno per accompagnare professionalmente l’introduzione e la manutenzione del software open source. Questo approccio è impiegato soprattutto negli ambiti critici per l’attività, in cui un know-how approfondito del software deve essere immediatamente disponibile. Ciò può comprendere anche parti della pubblicazione o l’intera pubblicazione.

Vantaggi

- Accesso diretto al know-how degli sviluppatori open source
- Correzioni ed estensioni su base contrattuale
- Scelta tra diversi fornitori open source
- Vincolatività, copertura contro i rischi di conformità

Svantaggi

- Costi esterni dovuti ai fornitori open source
- Dipendenza dal know-how del fornitore open source

Rischio e copertura
Rischio ridotto: la garanzia è prestata secondo il capitolato d’oneri o il service level agreement

Ambito d’applicazione tipico
Software open source critico per l’attività, per il quale sono disponibili poche o nessuna competenza interna di sviluppo

9. Punto di contatto

Nell’Amministrazione federale non esiste un punto di contatto unico che funga da Open Source Programme Office. In linea di principio gli uffici federali sono responsabili essi stessi dell’attuazione.

Le richieste generali relative agli strumenti OSS della serie di documenti Em002 possono essere rivolte al settore TDT della Cancelleria federale: opensource@bk.admin.ch.

10. Allegati

10.1. Modifiche rispetto alla versione precedente

- Capitolo 4.1: aggiunti i livelli di maturità
- Capitolo 6: acquisto di OSS completato e adeguato con i nuovi documenti dell’UFCL
- Capitolo 7.1: nuovo catalogo OSS della Confederazione (opensource.admin.ch)
- Ulteriori modifiche e miglioramenti redazionali minori

10.2. Riferimenti

I riferimenti alla serie di documenti Em002 figurano nelle Em002 Linee guida strategiche.

10.3. Abbreviazioni

Un elenco delle abbreviazioni figura nel documento principale Em002. Un glossario figura nel documento «Em002-6 Domande frequenti sulla pubblicazione di OSS (FAQ-OSS)».

10.4. Modelli di business con l’OSS

Il software open source non è di per sé un modello di business perché, diversamente dal software proprietario, non è possibile vendere un software soggetto a licenza open source. Esistono nondimeno diverse possibilità per le imprese di gestire modelli di business fondati sul software open source. Se ad esempio occorre ottenere un supporto esterno professionale per una soluzione open source, è necessario acquistare le corrispondenti prestazioni commerciali. I quattro modelli di business più diffusi per il software open source sono illustrati qui di seguito. Ulteriori aspetti di questi e di altri modelli di business sono esposti nella guida BITKOM *[BITKOM2023]*.

10.4.1. Servizi e prodotti basati sull’OSS

Le imprese possono offrire servizi commerciali quali il web hosting o il cloud computing sulla base di software open source, che con software proprietario sarebbero molto più costosi. Per questo motivo la maggior parte delle start-up, dei portali online e dei fornitori di commercio elettronico costruisce oggi le proprie piattaforme su software open source. Altre imprese tecnologiche, come le imprese di telecomunicazione, i fornitori di streaming o persino i fabbricanti di software proprietario, integrano software open source nei loro prodotti software e hardware nonché nei servizi online. Ciò consente alle imprese di continuare a offrire soluzioni innovative, difficilmente realizzabili con software proprietario.

10.4.2. Servizi per l’OSS

I fornitori open source erogano prestazioni per software open source selezionati. Dispongono di sviluppatori open source esperti e possono quindi offrire supporto, manutenzione, esercizio, sviluppo, consulenza, formazione e altre prestazioni per il software open source. Esse possono essere ottenute dal fornitore open source su mandato o nel quadro di un contratto d’appalto. Tali prestazioni per software open source possono essere messe a concorso pubblicamente, poiché non sussiste alcun

vendor lock-in. Ciò che conta in tali bandi è la considerazione di criteri adeguati, affinché siano selezionati fornitori di prestazioni effettivamente competenti (cfr. capitoli 0 e 8, *[BITKOM2023]* capitolo 3.3 e *[Gu2024]*).

10.4.3. Abbonamenti

Quando le prestazioni per software open source sono fornite in forma standardizzata e ricorrente, alla stregua di un service level agreement (SLA), si parla di abbonamenti. Nel quadro di tali abbonamenti le imprese garantiscono per esempio aggiornamenti di sicurezza continui, supporto via e-mail o telefono, compatibilità con altri prodotti software e hardware mediante certificazioni, prestazioni di manutenzione a lungo termine e coperture contro pretese giuridiche (diritto d’autore, brevetti). In cambio i clienti pagano abbonamenti per postazione di lavoro o per CPU, analogamente alle tasse di licenza o d’uso del software proprietario. Diversamente dal software proprietario, gli abbonamenti per il software open source non costituiscono tuttavia un presupposto per l’utilizzo del software, bensì soltanto un modo di remunerare il valore aggiunto delle prestazioni fornite dal fornitore open source.

10.4.4. Doppia licenza

Se un’impresa detiene la proprietà intellettuale di una soluzione software e tutti i componenti open source integrati sono soggetti a una licenza permissiva, è possibile applicare una doppia o multipla licenza. Ciò consente all’impresa che sviluppa di pubblicare il software sotto una licenza open source copyleft e al contempo di venderlo sotto una licenza proprietaria. Questa versione commerciale è spesso chiamata versione Enterprise e comprende tipicamente determinate funzionalità supplementari, quali integrazioni di interfacce esclusive o l’autorizzazione per gli acquirenti a integrare il software nei propri prodotti proprietari. A seconda dell’entità delle restrizioni della versione open source, è consigliabile prudenza nell’impiego di software a doppia licenza, poiché l’acquisto della versione Enterprise può risultare inevitabile, con un vendor lock-in pari a quello del tipico software proprietario.