

Open Source versus Open Standards

Gegensätzliche Konzepte und die Bedeutung von Interoperabilität

Inhalt

Inhalt.....	2
Einführung.....	3
Offene Standards als „Rezepte“	3
Im Mittelpunkt stehen Kompatibilität und Interoperabilität	3
Offene Standards.....	4
Definition von Standards.....	4
Standards zur Förderung von Handel und Innovation	4
Proprietäre vs. offene Standards.....	4
Definition von offenen Standards	4
Open Source	6
Vier Aspekte von Open Source	6
„Ideologen“ und „Pragmatiker“.....	6
Verschiedene OSS-Lizenzbestimmungen.....	6
Unterscheidung von Open-Source-Software und Open Standards	8
Prozess vs. Implementierung.....	8
Open Source ist unabhängig von Standards	8
Konsens als Basis für Standards.....	8
Interoperabilität	9
OSS-Lizenzbestimmungen können Interoperabilität verhindern.....	9
Unterschiedliche Ziele von OSS und offenen Standards	9
Motive zur Integration offener Standards	9
Warum einige Unternehmen OSS-Produkte fördern	9
Die Frage nach der Verantwortlichkeit	10
Probleme bei Tests und Gewährleistung.....	10
Interoperabilität als Basis für Marktakzeptanz.....	10
Fazit	12

Einführung

In der Softwareindustrie und unter Endanwendern bestehen erhebliche Unklarheiten hinsichtlich der Begriffe „Open Source“ und „Open Standards“ (offene Standards) sowie der damit verbundenen Konzepte, Policies und Lizenzmodelle. Dies beginnt bereits bei den grundlegenden Bedeutungen, die heftig umstritten sind. Ursache des Problems ist eine allzu großzügige und häufig falsche Verwendung der Begriffe – teils unbeabsichtigt, teils vorsätzlich. Die konzeptionellen Rahmenwerke hinter den Begriffen werden hierdurch verschleiert und Bestrebungen zur Formalisierung des Vokabulars zunichte gemacht. Durch die unscharfe Terminologie kommt es oftmals zu einer falschen Verknüpfung des Begriffs „Open Source“ mit dem dazu sehr verschiedenen Konzept der „Open Standards“.

Offene Standards als „Rezepte“

Auf dem Markplatz der vielen miteinander konkurrierenden Implementierungen sorgen offene Standards für Interoperabilität und die Erfüllung bestimmter Minimalanforderungen. Offene Standards sind unabhängig vom Entwicklungsmodell, das für die Implementierung dieser Standards verwendet wurde. Innerhalb des Software-Entwicklungsmodells können offene Standards gleichermaßen in proprietären Lösungen wie in Open-Source-Produkten verwirklicht werden. Darüber hinaus ist es richtig, dass die Software-Entwicklung nicht notwendigerweise auf Standards basieren muss, und die meiste Open-Source-Software tut dies auch nicht. Genau betrachtet handelt es sich bei offenen Standards um Spezifikationen und technologische Rahmenwerke oder „Rezepte“. Im Gegensatz dazu benennt Open Source den konkreten Softwarecode, in dem offene Standards implementiert sein können – oder auch nicht. Im übertragenen Sinne entspricht also die Open-Source-Implementierung offener Standards einem Kuchen, der nach den Vorgaben eines Rezeptes gebacken wurde.

Im Mittelpunkt stehen Kompatibilität und Interoperabilität

Dieses Dokument soll zu einer klaren Definition und Unterscheidung der beiden Konzepte beitragen. Nach der Begriffsbestimmung folgt darüber hinaus eine Analyse der Beziehungen zwischen Open-Source-Software und dem Open-Standards-Prozess. Im Mittelpunkt stehen dabei die Fragen, in welchem Ausmaß die beiden Konzepte kompatibel sind, wo sie kollidieren und welche Auswirkungen sie jeweils auf die Interoperabilität von Hardware und Software haben.

Offene Standards

Definition von Standards

Abhängig vom Kontext kann der Begriff „Standard“ eine Vielzahl verschiedener Bedeutungen annehmen. Die International Organization for Standards (ISO) verwendet folgende Definition: „Ein Standard ist ein Dokument, das auf einem Konsens basiert und Regeln, Richtlinien oder Merkmale für Aktivitäten oder deren Resultate beinhaltet.“

Standards nehmen verschiedene Formen an, wie zum Beispiel technische Spezifikationen, Sicherheitsprozeduren, Herstellungsprozesse und Zugangsvoraussetzungen, um nur einige wenige zu nennen. Als Diskussionsgrundlage ist in diesem Dokument folgende Definition maßgeblich:

Ein Standard ist eine *technische Spezifikation*, die zur allgemeinen Verwendung in der Industrie vorgesehen ist oder diesen Status bereits erreicht hat.

Zweites entscheidendes Merkmal eines Standards ist die Fähigkeit, zwischen verschiedenen Produkten oder Services *Interoperabilität* herzustellen.

Standards sorgen folglich für Interoperabilität und die Erfüllung bestimmter Minimalanforderungen. Sie sind unabhängig von dem Entwicklungsmodell, das für die Implementierung dieser Standards verwendet wurde. So legt der ISO-Standard zu metrischen Schraubgewinden die Abmessungen eines zwei Millimeter großen Gewindes fest, aber nicht den Produktionsprozess oder die zu verwendenden Materialien.

Standards zur Förderung von Handel und Innovation

Das Konzept von Standards ist bereits sehr alt. Jedes moderne ökonomische System erfordert die Einführung von Standards für Gewichte, Maße, Zeit und Währung. Wenn eine Gesellschaft im Laufe ihrer Entwicklung komplexere Technologien hervorbringt und verwendet, benötigt sie aus Effizienzgründen ein stetig wachsendes System von Spezifikationen. Gleichzeitig werden auch die Prozeduren für die Einführung neuer Standards formalisiert. Standards fungieren als Benchmark oder Referenzpunkt, an dem konkurrierende Implementierungen gemessen werden können, sodass die Erfüllung der fundamentalen Anforderungen sichergestellt bleibt. Da Standards die Interoperabilität zwischen den Technologien der vielen verschiedenen Computer- und Elektronikhersteller ermöglichen, werden sie allgemein als wirksame und effiziente Methode zur Förderung von Handel und Innovation betrachtet.

Proprietäre vs. offene Standards

Es existieren mehrere Wege zur Beschreibung und Kategorisierung von Standards. Eine Art der Klassifizierung unterscheidet Standards aufgrund der Kriterien *proprietär* (entwickelt nach Maßgabe einer oder mehrerer privater Organisationen) und *offen* (entwickelt in Zusammenarbeit und auf Grundlage eines Konsens innerhalb offener Foren oder Prozesse). Technologieunternehmen entwickeln proprietäre Standards, damit Geschäftspartner, Kunden und andere Interessierte eigene Produkte und Services realisieren können, die mit den Lösungen des Herstellers nahtlos zusammenarbeiten. Solche Standards werden „proprietär“ genannt, da sie speziell auf die Produkte und Services eines bestimmten Herstellers zugeschnitten sind und nicht das Ziel einer universellen oder produktübergreifenden Interoperabilität verfolgen.

Definition von offenen Standards

Im Gegensatz dazu sind offene, Konsens-basierte Standards unabhängig von einzelnen Produkten. Dies bedeutet, dass offene Standards nicht an die Produkte und Services eines

speziellen Anbieters gebunden sind. Ebenso wenig verhindern oder beschränken die entsprechenden Lizenzbestimmungen der offenen Standards eine Implementierung in bestimmten Hardware- oder Softwareprodukten. Unglücklicherweise wird „offener Standard“ häufig im Marketing für einige Produkte und Services unzutreffend eingesetzt. Dem Begriff an sich fehlt die nötige Klarheit, deshalb wird in diesem Dokument als Grundlage für weitere Überlegungen die folgende Definition verwendet:

Ein offener Standard ist eine technische Spezifikation, die in Zusammenarbeit und auf Basis eines Konsenses entwickelt und gepflegt wird. Die entsprechenden Lizenzbestimmungen erlauben es allen interessierten Personen und Organisationen, den offenen Standard in eigene Produkte und Services zu integrieren, sodass diese mit anderen Implementierungen der Spezifikation interoperabel sind.

Open Source

Vier Aspekte von Open Source

Der Begriff „Open-Source-Software“ (OSS) kann verwirren, da er in einer Vielzahl von Zusammenhängen verwendet wird. Er bezieht sich auf mindestens einen von vier verwandten Aspekten oder deren Kombination:

- das OSS-Entwicklungsmodell
- OSS-Philosophien
- OSS-Lizenzierungssysteme
- und OSS-Geschäftsmodelle.

Die Modelle für Lizenzierung und Distribution zählen zu den Merkmalen, die OSS und kommerzielle Software hinsichtlich der Einhaltung von Standards unterscheiden (siehe weiter unten). Die anderen Aspekte sind zwar grundsätzlich davon unabhängig, beeinflussen und definieren aber trotzdem die OSS-Lizenzierungs- und Distributionsmodelle. In ihrer Gesamtheit legen die vier Aspekte die Basis für eine Charakterisierung der so genannten OSS-Community oder OSS-Bewegung.

„Ideologen“ und „Pragmatiker“

OSS stellt eine Alternative zum traditionellen kommerziellen Softwaremodell dar, in dem Unternehmen die Entwicklung von Softwareprodukten vorantreiben. Distribution und Lizenzierung der fertigen Produkte sind feste Bestandteile dieses Modells, denn Unternehmen müssen einen Return on Investment generieren und ihre Entwicklungsvorleistungen finanzieren. Im Gegensatz zu den wirtschaftlichen Zielen, die das kommerzielle Modell kennzeichnen, steht im Mittelpunkt des OSS-Modells oft die Lösung bestimmter Probleme, von denen IT-Experten oder Applikationsentwickler allgemein betroffen sind. Deshalb stellen OSS-Entwickler industrieweit ihre Arbeitszeit kostenlos und freiwillig zur Verfügung, um diese konkreten Probleme zu lösen. Da die Software in einem Gemeinschaftsprozess entwickelt wird, kann die fertige Problemlösung auch von jedermann frei genutzt und weitergegeben werden. Nichtsdestotrotz haben sich in der OSS-Community verschiedene Auffassungen über die freie Verwendung etabliert, denn es gibt keine einheitliche OSS-Philosophie. Innerhalb der Gemeinschaft besteht ganz im Gegenteil eine große Uneinigkeit über die Lizenzbestimmungen, die Distribution und Einsatz von OSS regeln. Die Verfechter der rivalisierenden OSS-Philosophien können dabei in zwei Gruppen eingeteilt werden:

- **Ideologen:** Diese Gruppe glaubt, dass Software (sowohl Quell- als auch Objektcode) für jedermann kostenlos verfügbar sein sollte und niemand Software kommerziell verwerten sollte. Schöpferische Eigentumsrechte am Softwarecode werden eingesetzt, um weitestgehend sicherzustellen, dass weder OSS noch andere Software (soweit möglich) kommerziell angeboten wird.
- **Pragmatiker:** Diese Gruppe unterstützt die Koexistenz von kommerzieller Software und OSS. Gleichzeitig glauben Pragmatiker aber an die Überlegenheit des OSS-Entwicklungsmodells, das als tragfähige Basis für eine „gesunde“ Softwareindustrie gesehen wird.

Verschiedene OSS-Lizenzbestimmungen

Ergebnis dieser unterschiedlichen Philosophien ist die Distribution von OSS unter einer Vielzahl verschiedener Lizenzbestimmungen. Die Free Software Foundation (FSF), Autor und Erfinder der GNU General Public License (GPL), kann dem ideologischen Ansatz zugerechnet werden. Unter der GNU erfolgt die Distribution von Linux, dem populärsten OSS-Produkt. Zu den Pragmatikern zählt die Open Source Initiative (OSI), die eine große

Liste der verschiedenen anerkannten Lizenzbestimmungen verwaltet, inklusive der GPL. Die einzelnen Lizenzmodelle unterscheiden sich sogar in Schlüsselaspekten erheblich, aber die große Mehrheit der Open-Source-Lizenzen entspricht den OSI-Definitionen. Von den neun essentiellen OSI-Kriterien haben drei Relevanz für dieses Dokument:

- Open-Source-Lizenzen erfordern, dass der Quellcode für die Öffentlichkeit zugänglich ist.
- Open-Source-Lizenzen erlauben jedermann die Weitergabe des Quellcode ohne Einschränkungen.
- Open-Source-Lizenzen erlauben jedermann die Modifikation des Codes sowie die freie Weitergabe des veränderten Codes.

Innerhalb dieser weit gefassten Parameter differieren OSS-Lizenzen stark voneinander, vor allem bei Art und Umfang der Beschränkungen, die sie Lizenznehmern auferlegen. Einige OSS-Lizenzen sind extrem freizügig, wie zum Beispiel die Berkeley Software Distribution (BSD). Dabei handelt es sich um eine ganz einfache und schrankenlose Lizenz für freie Software, die lediglich die Zuschreibung von Copyrights sowie Reproduktionen der Lizenz regelt. Andere Lizenzen enthalten moderate Restriktionen, so die Apple Public Source License (APSL) oder die Mozilla Public License (MPL). Beide legen fest, dass eine Distribution bestimmter Dateien mit integriertem APSL/MPL-Code nur als Quellcode und nur unter den Bestimmungen der APSL/MPL erfolgen darf. Andere Dateien, die Teil eines größeren Systems sind und keinen APSL/MPL-Code enthalten, dürfen hingegen unter jeder Lizenz veröffentlicht werden. Darüber hinaus gibt es auch sehr restriktive Modelle, wie die bereits erwähnte GNU GPL. Jede Software, die in irgendeiner Form GPL-Code enthält, darf nur unter den Bedingungen der GPL weitergegeben werden.

Unterscheidung von Open-Source-Software und Open Standards

Prozess vs. Implementierung

Die Unterscheidung zwischen den beiden sehr unterschiedlichen Begriffen „Open-Source-Software“ und „Open Standards“ fällt nicht schwer. Im weitesten Sinne beschreibt „Open Standard“ das Resultat eines *Prozesses* zur Etablierung einheitlicher technischer Spezifikationen. „Open Source“ bezieht sich hingegen auf die Implementierung, das heißt ein spezielles *Modell für die Software-Entwicklung, Lizenzierung und Distribution*. Selbstverständlich kann die Veröffentlichung von Quellcodes nicht mit offenen Standards gleichgesetzt werden. Der Prozess zur Generierung offener Standards bleibt im Bezug auf die Software-Entwicklung neutral: Bei der Suche nach der besten Lösung wird niemand bevorzugt oder benachteiligt.

Open Source ist unabhängig von Standards

OSS-Entwickler und kommerzielle Entwickler stehen gleichermaßen vor der Wahl, bestimmte Standards in ihren Softwarecodes zu verwenden oder auch nicht. Natürlich existieren unzählige OSS-Implementierungen offener Standards, aber das Merkmal „Open Source“ ist grundsätzlich vollkommen *unabhängig* von Eigenschaften wie „Standard-basiert“ oder „Standard-konform“. Es gibt keine inhärente Verbindung zwischen dem Modell, unter dem eine bestimmte Software entwickelt, lizenziert und vertrieben wird, und der Implementierung eines offenen Standards. Die Behauptung, dass „Open Source“ *und* „Open Standards“ identisch seien, ist offensichtlich unrichtig. Fälschlicherweise wird zur Begründung oftmals die Entwicklungsgeschichte von World Wide Web und Internet angeführt, aber diese beiden Konzepte müssen getrennt voneinander betrachtet werden. Internet und World Wide Web sind die Ergebnisse der Anstrengungen von staatlichen und akademischen Institutionen, Wirtschaftsunternehmen und einzelnen Personen. Im Rahmen von Organisationen zur Standardisierung, wie IETF und W3C, wurden von allen Beteiligten die Standards entwickelt, auf denen das Internet heute basiert (IP, TCP, DNS, PPP, HTTP, SMTP, POP und andere). Viele Implementierungen dieser Technologien existieren sowohl in kommerziellen als auch in Open-Source-Entwicklungsmodellen.

Konsens als Basis für Standards

Kritische Differenzen bestehen darüber hinaus bei der Frage nach dem Konsens. Offene Standards werden in einem Prozess auf Basis eines Konsenses geschaffen. Hauptsächliche Aufgabe einer unabhängigen Standardisierungsinstanz ist die Bereitstellung eines Forums, in dem zu bestimmten Punkten ein Konsens erzielt werden kann. Um dies zu gewährleisten, stellen die Institutionen umfassende Regeln zu Beteiligung, Konstruktion, Umsetzung und Erweiterung auf. Darüber hinaus definieren sie die Struktur und den Ablauf der Besprechungen und sorgen somit für einen fairen Diskurs, in dem Einzelinteressen nicht die Oberhand gewinnen können. Die einfache Implementierung einer technischen Spezifikation als Open Source stellt hingegen keinen Konsens her und macht aus der Technologie auch keinen Standard. Tatsächlich fehlt bei der Entwicklung von Open-Source-Software jegliche Form formeller Richtlinien und festgeschriebener Konventionen. Jedes Projekt hat zudem eine eigene kulturelle Identität und Struktur, wobei die Kontrolle des Endergebnisses meist bei einer Kerngruppe oder einem selbsternannten Projektverantwortlichen liegt. Für die Entwicklung von Open-Source-Projekten sind diese Faktoren von großer Bedeutung, gleichzeitig unterstreichen sie aber auch die fundamentalen Unterschiede zwischen Open Source und dem Konsens-basierten Prozess zur Etablierung offener Standards.

Interoperabilität

OSS-Lizenzbestimmungen können Interoperabilität verhindern

Open-Source-Befürworter argumentieren oftmals, dass die universelle Zugänglichkeit von OSS-Code notwendigerweise auch die Interoperabilität fördert. Richtig ist, dass jeder Entwickler Standards in seine Projekte implementieren und unter der OSS-Lizenz veröffentlichen kann, wenn dies von den entsprechenden Lizenzbestimmungen erlaubt wird. Wie jede Implementierung eines Standards in kommerzieller Software trägt dies tatsächlich zur Etablierung und allgemeinen Verbreitung dieses Standards bei. Einige Merkmale von OSS lassen es aber fraglich erscheinen, ob OSS wirklich zwingend die Verwendung von offenen Standards begünstigt und Interoperabilität fördert. Es existieren zum Beispiel mehrere offene Standards (MPEG-, GSM- und 3GPP-Spezifikationen, MHP, IEEE 1394 etc.), die unter herkömmlichen kommerziellen Softwarelizenzen problemlos implementiert werden können. In vielen Fällen ist die Implementierung und Distribution als Open-Source-Software hingegen nicht möglich, weil die OSS-Lizenzbestimmungen dies verhindern.

Unterschiedliche Ziele von OSS und offenen Standards

Offene Standards und OSS unterscheiden sich grundlegend in ihren Zielen. Die fundamentale Aufgabe eines Standards ist die Förderung von Interoperabilität. Bei offenen Standards geht es darüber hinaus um Interoperabilität zwischen den verschiedensten Hardware- und Softwareprodukten sowie Services. Dies begünstigt den Wettbewerb unter den Anbietern und veranlasst sie zu Innovationen, um die eigene Implementierung positiv vom Umfeld abzuheben. Im Kontrast dazu steht das grundlegende Ziel der OSS-Philosophie: Software-Anwender sollen Quellcodes nach Belieben modifizieren und die veränderten Versionen weitergeben können. Alle OSS-Codes sind frei manipulierbar, somit darf ursprünglich Standard-konforme OSS von jedem Anwender verändert werden – auch wenn die Software danach keine Standards mehr unterstützt und zu den Versionen anderer Nutzer inkompatibel ist. Die Freiheit zur Modifikation des Quellcode beinhaltet auch das Recht zur Zerstörung der Interoperabilität.

Motive zur Integration offener Standards

Ein weiterer wichtiger Punkt ist die Motivation von Entwicklern bei der Implementierung von Standards. Kommerzielle Software-Entwickler (CSDs) sind zur Umsatzgenerierung grundsätzlich auf den Verkauf von Softwarelizenzen angewiesen. Der Grad der Interoperabilität ihrer Produkte mit der Hardware und Software anderer Hersteller wirkt sich direkt auf die Nachfrage und das Interesse der Anwender aus. CSDs haben daher einen starken und unmittelbaren Anreiz zur Implementierung offener Standards und der Realisierung von Interoperabilität. Dementsprechend stellen CSDs umfangreiche Ressourcen für die Entwicklung Standard-konformer und interoperabler Software bereit.

Warum einige Unternehmen OSS-Produkte fördern

Einige Unternehmen fördern die Verbreitung von OSS als Teil ihres eigenen Geschäftsmodells, obwohl die Lizenzierung der OSS-Produkte kein Umsatzfaktor ist. Im Fokus steht vielmehr die Umsatzgenerierung durch Support, Services, Hardware, Dokumentation sowie den Verkauf von ergänzender kommerzieller Software für OSS-Plattformen und -Applikationen. Daher haben die Unternehmen ein Interesse an der Interoperabilität ihrer OSS-Produkte und Services. Idealerweise würde diese Gruppe von Anbietern also dafür sorgen, dass die von ihnen vermarktete OSS im gleichen Maße anerkannten Standards entspricht wie kommerzielle Software. In der Realität investieren diese Unternehmen aber meist nicht in die Entwicklung von OSS-Implementierungen und sind auch

nicht am Management der Projekte beteiligt. Als Konsequenz haben sie somit auch nur einen geringen Einfluss auf die Entwicklung des OSS-Codes und die Implementierung von Standards zur Interoperabilität.

Die Frage nach der Verantwortlichkeit

Damit eng verbunden ist die Frage nach der Verantwortlichkeit und dem „Besitz“ am Quellcode. Ein CSD besitzt den von ihm entwickelten Code und sein Ansehen beim Kunden hängt maßgeblich von der Leistungsfähigkeit und Interoperabilität seiner Produkte ab. Deshalb verwenden CSDs umfangreiche Ressourcen auf Tests ihres Codes zur Überprüfung dieser Kriterien. Im Gegensatz dazu kann OSS-Code aus mehreren Tausend voneinander unabhängigen Beiträgen bestehen, die in den verschiedensten kontrollierten und unkontrollierten Umgebungen entwickelt wurden. Intensive Tests der Interoperabilität und Übereinstimmung mit Standards sind daher bei OSS-Software ungleich schwieriger und aufwändiger. Da Anwender die Freiheit haben, den OSS-Code ihren jeweiligen Aufgaben anzupassen, verlagert sich der Besitz am Code vom Hersteller auf den Anwender – und damit auch die Verantwortung für die Interoperabilität. Die meisten Anwender verfügen aber nicht über die nötigen Kenntnisse und Ressourcen, um die Interoperabilität einer Software mit anderen Produkten zu gewährleisten. Jede Modifikation durch den Anwender kann somit die Software zu anderen Versionen derselben Software inkompatibel machen. Besonders problematisch ist dies bei Software, die zur Interaktion über das Internet entwickelt wurde.

Probleme bei Tests und Gewährleistung

Im Rahmen der OSS-Implementierung eines offenen Standards kann eine Reihe von Komplikationen auftreten. Aufgrund der unklaren Verantwortlichkeit für den Code und des Fehlens einer definierten Zertifizierungsinstanz haben sich Tests der Interoperabilität von OSS als Problem erwiesen. Jede Modifikation am Code durch Anwender, die zeitlich nach einem Test vorgenommen wird, führt zur Entwertung des vergebenen Zertifikats. Da Tests zudem mit hohen Kosten verbunden sind, können sie realistischweise auch nicht nach jeder Änderung des Codes wiederholt werden. Über die Zeit vergrößert sich das Problem zusätzlich durch veränderte Versionen der Standards. Da offene Standards in einem kontinuierlichen Prozess weiterentwickelt und gepflegt werden (Implementierung neuer Features oder Bugfixes), muss die Kontrolle dieser Evolution bei einer Standardisierungsinstanz liegen. Zwischen OSS-Entwicklern und potenziellen Anwendern ihrer Software besteht normalerweise keine eindeutig definierte Beziehung. Deshalb gibt es auch kein universell einsetzbares Verfahren, um bestehende Implementierungen zuverlässig auf neue Versionen der verwendeten Standards zu aktualisieren. Einigen Anwendern gelingt es möglicherweise, zeitnah auf neue Versionen umzusteigen, aber die Verantwortung hierfür lastet allein auf dem Kunden. Vor ähnlichen Schwierigkeiten stehen auch Tests der Interoperabilität von OSS in Verbindung mit Referenz-Implementierungen oder den Produkten anderer Hersteller. Sobald der getestete Code modifiziert wird, kann keine Gewährleistung mehr für die Interoperabilität übernommen werden.

Interoperabilität als Basis für Marktakzeptanz

Interoperabilität ist das zentrale Merkmal jedes erfolgreichen offenen Standards. Wenn Produkte, in denen offene Standards implementiert sind, sich als nicht ausreichend interoperabel erweisen, besteht keine Aussicht auf Akzeptanz im Markt. In Folge hat auch der offene Standard keine Möglichkeit mehr, sich industrieweit zu etablieren. Aus all diesen hier angeführten Gründen kann OSS dazu führen, dass Hersteller für die verschiedenen Implementierungen eines Standards keine Interoperabilität mehr garantieren können. Vor diesem Hintergrund erweist sich, dass OSS grundsätzlich nicht besonders geeignet ist, um die

Verbreitung offener Standards und Interoperabilität zu fördern. OSS zeigt im Bezug auf die Implementierung offener Standards vielmehr ein erhebliches Konfliktpotenzial, dass durch die mangelnde Verantwortlichkeit gegenüber dem Markt noch verstärkt wird.

Fazit

Die ungenaue Abgrenzung zwischen OSS und offenen Standards, sei sie vorsätzlich oder ungewollt, hat zu einer allgemeinen Verwirrung geführt. Glücklicherweise kann ein Missverständnis der Begriffe einfach aufgelöst werden. Bei eingehender Betrachtung der beiden Konzepte stellt sich ganz klar heraus, dass sie nicht nur unterschiedlich sind, sondern sich sogar in einigen Aspekten widersprechen.

Insbesondere sind es offene Standards, die Interoperabilität zwischen den Technologien der vielen verschiedenen Hersteller garantieren. Dies wiederum wird allgemein als wirksame und effiziente Methode zur Förderung von Handel und Innovation betrachtet.

Die in diesem Dokument enthaltenen Informationen stellen die behandelten Themen aus der Sicht der Microsoft Corporation zum Zeitpunkt der Veröffentlichung dar. Da Microsoft auf sich ändernde Marktanforderungen reagieren muss, stellt dies keine Verpflichtung seitens Microsoft dar, und Microsoft kann die Richtigkeit der hier dargelegten Informationen nach dem Zeitpunkt der Veröffentlichung nicht garantieren.

Dieses Whitepaper dient nur zu Informationszwecken. MICROSOFT SCHLIESST FÜR DIE INFORMATIONEN IN DIESEM DOKUMENT JEDE GEWÄHRLEISTUNG AUS, SEI SIE AUSDRÜCKLICH ODER KONKLUDENT.

Die Benutzer/innen sind verpflichtet, sich an alle anwendbaren Urheberrechtsgesetze zu halten. Unabhängig von der Anwendbarkeit der entsprechenden Urheberrechtsgesetze darf ohne ausdrückliche schriftliche Erlaubnis der Microsoft Corporation kein Teil dieses Dokuments für irgendwelche Zwecke vervielfältigt oder in einem Datenempfangssystem gespeichert oder darin eingelesen werden, unabhängig davon, auf welche Art und Weise oder mit welchen Mitteln (elektronisch, mechanisch, durch Fotokopieren, Aufzeichnen, usw.) dies geschieht.

Microsoft kann Inhaber von Patenten oder Patentanträgen, Marken, Urheberrechten oder anderen gewerblichen Schutzrechten sein, die den Inhalt dieses Dokuments betreffen. Die Bereitstellung dieses Dokuments gewährt keinerlei Lizenzrechte an diesen Patenten, Marken, Urheberrechten oder anderen gewerblichen Schutzrechten, es sei denn, dies wurde ausdrücklich durch einen schriftlichen Lizenzvertrag mit der Microsoft Corporation vereinbart.

© 2003 Microsoft Corporation. Alle Rechte vorbehalten.

Die in diesem Dokument aufgeführten Produkt- und Firmennamen sind möglicherweise Marken der jeweiligen Eigentümer.