



Open Source-Software

Univ. Lektor Dr. Ingmar Vinzenz

Email: ingmar.vinzenz@tuwien.ac.at

- Open Source Software, was ist das?
 - Quelloffene Software, der Source-Code ist verfügbar
 - Die unter bestimmten Lizenzen stehen
 - Prinzipielle Philosophie: freie und unbeschränkten Weiterverbreitung der Software, die Verfügbarkeit des Quellcodes
 - Ist grundsätzlich „freie Software“
 - Grundsätzlich „kostenfrei“ erhältlich
 - Kann grundsätzlich frei verbreitet werden
 - Achtung, auf Details!
 - Verschiedenste Erscheinungsformen:
 - Libraries, Plug-Ins, Tools,...
 - Einzelne Codeteile
 - Komplette Anwendungssoftware
- Ziel von OSS generell ist:
 - Source-Code soll eingesehen und verwendet werden können
 - Kann bearbeitet, vervielfältigt, verwertet

- Unterschied Open Source Software und...
 - Freeware?
 - Anwendung des Programmes ist kostenlos
 - Source-Code nicht verfügbar
 - Keine Bearbeitung oder Weiterentwicklung
 - Shareware, Expireware, Crippleware?
 - Meist: Software kann für einen bestimmten Zeitraum getestet werden
 - Oft: Eingeschränkter Nutzungsumfang
 - Auch hier kein Source-Code und keine Bearbeitung/Weiterentwicklung
 - Software in Public Domain?
 - Software unterliegt nicht oder nicht mehr dem Urheberrecht
 - Wegen Zeitablauf: Bei Software noch nicht der Fall!
 - Urheber bleibt Urheber (Kontinentaleuropa), dieser gewährt „jedermann“ umfassende Rechte zur Bearbeitung, Weiterentwicklung, Vervielfältigung, Verwertung...
 - Im anglo-amerikanischen Recht ist expliziter Verzicht auf das Copyright möglich
 - Ohne Bedingungen und kostenlos!
 - Beispiele?

- Charakteristikum von Open Source Software liegt im Lizenzmodell!
 - Achtung: Begriff Lizenz passt nicht 100% auf Urheberrecht!
- Arten von OSS
 - Copyleft-Lizenzen, zB. GPL
 - Lizenzen mit beschränkten Copyleft-Effekt
 - Freizügige (BSD-artige) Lizenzen ohne Copyleft-Effekt, zB BSD
 - Lizenzen mit Wahlmöglichkeiten
 - Lizenzen mit Sonderrechten
 - Sonstige Lizenzen, zB. Creative Commons

- Wichtigster Vertreter:
 - General Public License (GPL)
- Copyleft?
 - Bearbeitungen einer unter der GPL laufenden Software wiederum nur unter Geltung der GPL verbreitet werden
 - Ziel ist es, dass Software nach einer Bearbeitung auch weiter frei bleibt
 - Rückführung in den proprietären Bereich soll verhindert werden
 - “The code and the freedoms become legally inseparable.”
 - Infektion („viraler Effekt“) von Bearbeitungen und Ergänzungen: Der gesamte neue Code ist unter die Lizenz zu stellen!
 - Von der Ansteckung ausgenommen ist Software, die unabhängig von der OSS ausführbar ist und dabei eigene Funktionalitäten besitzt
 - Probleme bereits bei Bündelung (in einer Installationsdatei)
 - Verbindung eigene Software mit Copyleft-Software:
 - Eigene Software wird „frei“ und wird selbst zur Copyleft-Software

- Pflichten bei Copyleft-Lizenzen (GPL v 3) – Auszug:
 - Hinweis auf Bearbeitung und Datum
 - Beibehaltung der „Copyright“-Hinweise, nicht zu verändern oder zu erweitern
 - Haftungs- und Gewährleistungsausschlüsse übernehmen!
 - Den Source-Code mitliefern oder ein drei Jahre lang gültiges Angebot dazu abgeben
- Verwendung kann ein hohe Risiko für Softwarehersteller bedeuten!
 - Verlust der Kontrolle über eigene Software durch „Infektion“
 - Source-Code ist herauszugeben
 - Verstoß führt zu Verlust des Nutzungsrechts der lizenzierten Teile!
 - „Problem:“ Hersteller muss Gewährleistung und Haftung ausschließen, was oft nicht möglich ist

Zulässigkeit des Viralen Effekts?

- Judikatur in Deutschland:
 - Fall WLAN-Router: Ein Hersteller (A) verwendete für seinen Router Software unter GPL v2, ein anderer Hersteller (B) übernahm diese Software. Klage des A gegen B scheiterte, weil die gesamte Software unter der Lizenz steht und B sie daher nutzen durfte
- In Österreich: Kein Fall ersichtlich
- Rechtsnatur des Viralen Effekts weitgehend ungeklärt:
 - Verpflichtung dinglich-urheberrechtlicher Natur?
 - Schuldrechtliche Verpflichtung?
 - Sind diese Lizenzen AGB?
 - AGB-Kontrolle?
 - Sittenwidrigkeit?
 - Wie kann Automatismus des viralen Effekts erklärt werden?
 - Ist dieser überhaupt möglich?
 - Wirkung von einseitigen Erklärungen, dass Copyleft-Effekt nicht eintreten soll?

Zulässigkeit des Viralen Effekts?

- Faktisch scheint der virale Effekt „zu funktionieren“!
- Wenn man schon nicht die Finger von „Copyleft“-Software lassen möchte:
 - Eigenständige, lauffähige Programme schaffen!
 - Sonst kein abgeleitetes Werk entstanden!
 - Achtung bei Einsatz von Compilern: Wird Copyleft-Code in die neue Software eingefügt, untersteht diese auch wieder diesen Nutzungsbedingungen
 - Problematisch und im Einzelfall zu überprüfen:
 - Schnittstellen: Findet Austausch durch Kommunikationsmitteln statt, die üblicherweise bei der Kommunikation und Zusammenarbeit von Programmen verwendet werden?

Beispiele für Lizenzen mit Copyleft (Auswahl)

- GNU General Public License (GPL)
- Honest Public License
- IBM Public License
- European Union Public License
- Open Software License
- RedHat eCos Public License
- Eclipse Public License
- Deutsche Freie Softwarelizenz

- Wichtigster Vertreter:
 - GNU Lesser General Public License (LGPL)
- Kernpunkte und Unterschiede zum starken Copyleft:
 - Pflichten im Wesentlichen wie bei starkem Copyleft
 - Lizenztext und Copyright-Vermerk beibehalten
 - Haftungs- und Gewährleistungsausschüsse belassen
 - ...
 - LGPL für Libraries entwickeln, um freie Standard-Libraries auch im kommerziellem Softwarebereich verwenden zu können
 - Ein Programm, das LGPL-Code zusammen mit eigenem proprietärem Code verwendet:
 - So aufgebaut, dass jeder Nutzer den quelloffenen in das endgültige Programm (selbstständig) verbinden kann.
 - Nutzer müssen den LGPL-Teil ändern und neu linken können

Beispiele für Lizenzen schwachem Copyleft-Effekt

- GNU Lesser General Public License (LGPL)
- Mozilla Public License (MPL)
 - Etwas schwächerer Copyleft-Effekt
 - Source-Code muss grundsätzlich unter MPL bleiben
 - Dieser darf jedoch zusammen mit proprietärem Code für ein Programm verwendet werden.
- NASA Open Source Agreement
- Apple Public Source License

Geschäftsmodelle mit OSS mit Copyleft-Effekt?

- Kann zB. GPL von einem SW-Hersteller kommerziell nutzbar gemacht werden?
- Artikel 4 GPL v3:
 - "You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee."
 - Klingt doch gut?
- Lösungen:
 - Harte Trennung zwischen GPL/LGPL, abgetrennten Teil kommerzialisieren
 - Unabhängige Programme schaffen
 - Dual Licensing?
 - Womit wird Geld verdient?
 - Bsp.: Magnolia CMS

- Wichtige Vertreter:
 - BSD License (Berkeley Software Distribution)
 - Apache Software License
 - MIT License
- Relativ große Freiheiten:
 - Bearbeitung, Verbreitung, Verknüpfung
 - Überführung in proprietäre Software zu eigenen Lizenzbedingungen möglich
 - Keine Verpflichtung, den Quellcode zu veröffentlichen

■ Lizenzen mit Wahlmöglichkeiten

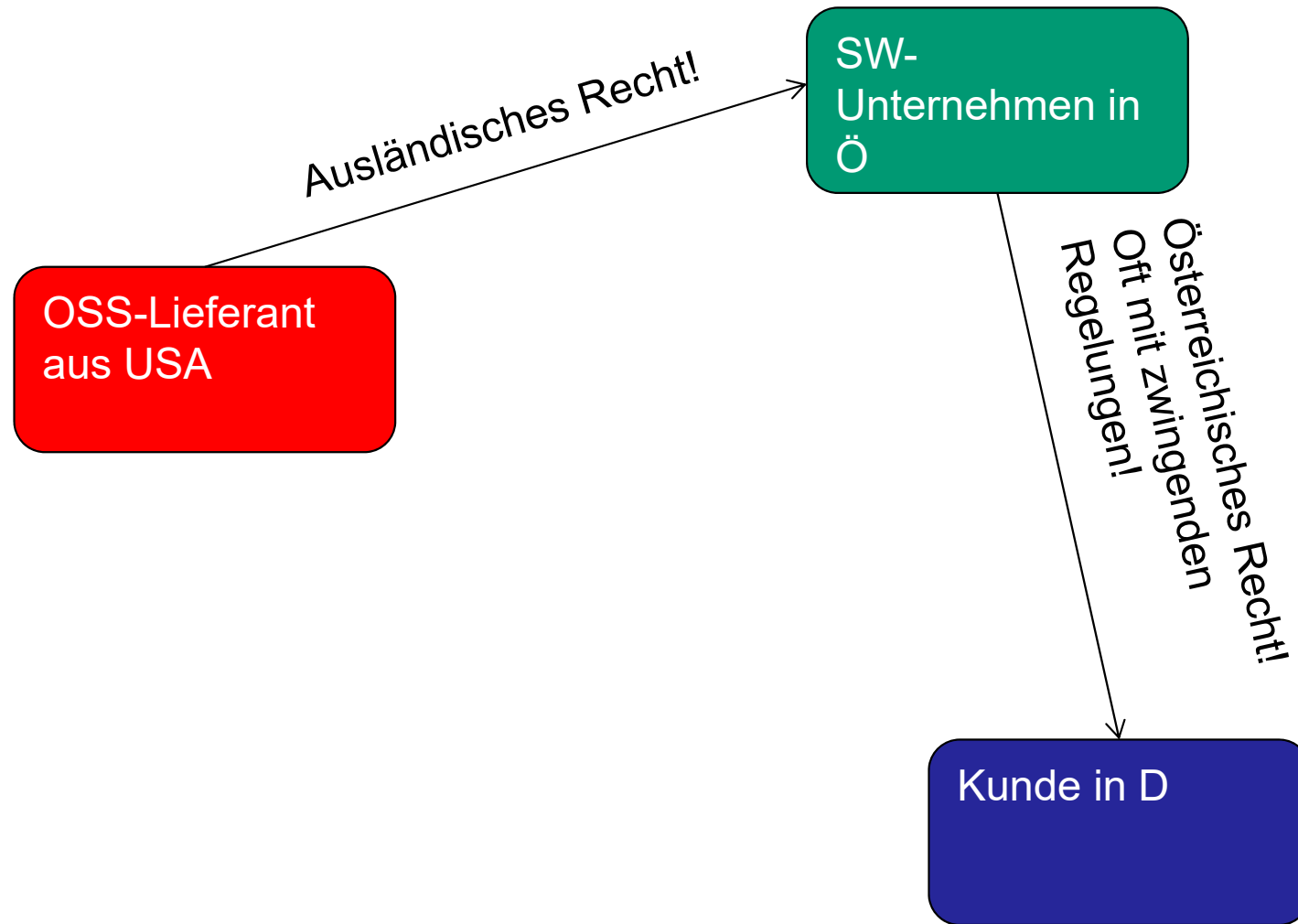
- Unterschiedliche Rechtsfolgen, je nach Modifikation
- Wahlmöglichkeiten für den Nutzer, wie Änderungen weiterverbreitet werden können
- Bsp:
 - Artistic License
 - LaTeX Project Public License
 - Apple Public Source License

■ Lizenzen mit Sonderrechten

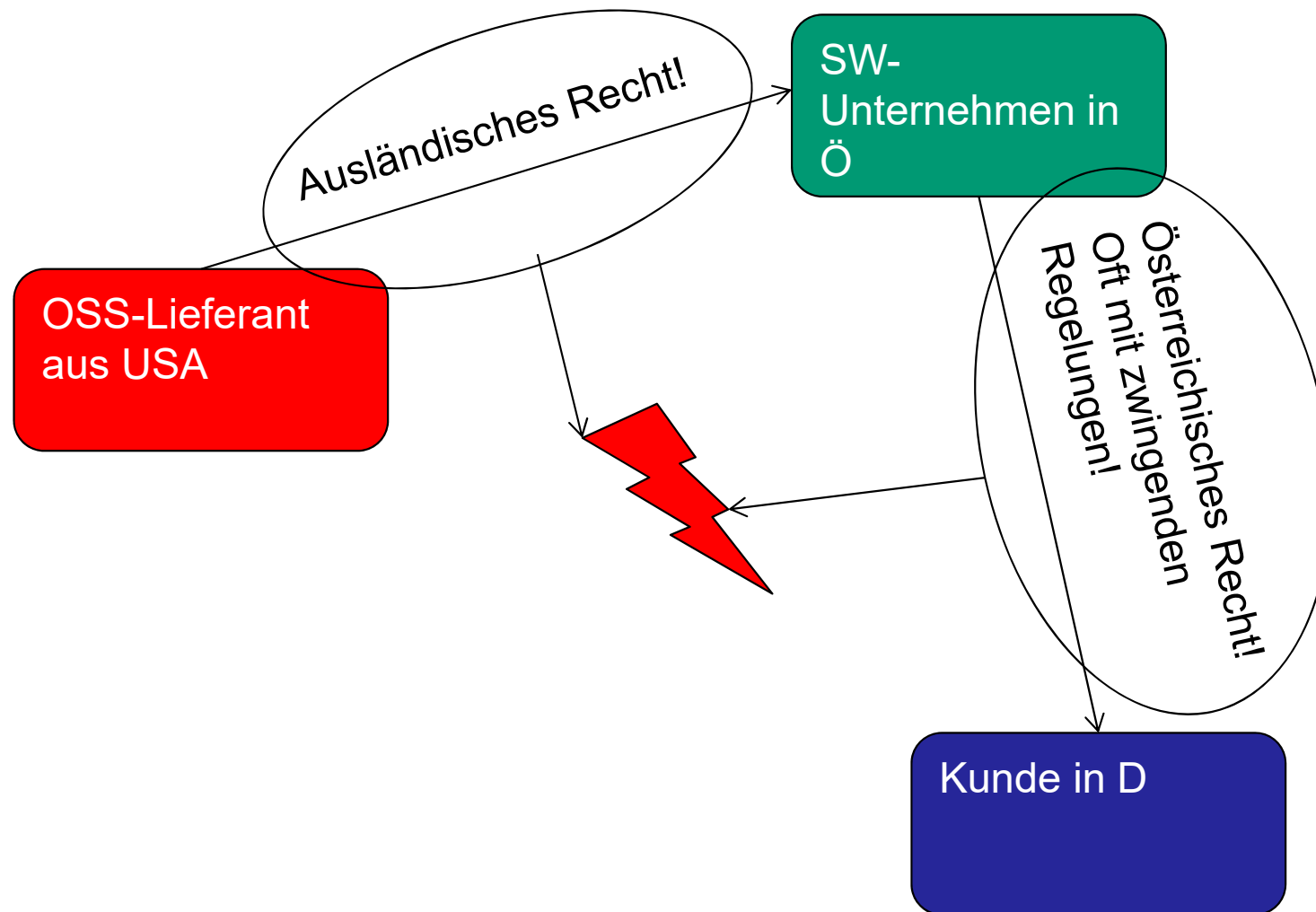
- Lizenzgeber sichert sich bestimmte Sonderrechte (Privilegien), wenn durch einen Nutzer Software weiterentwickelt und weitergegeben wird
- Oft Große Unternehmen, die proprietäre Software später offengelegt haben
- Erhalten oft nicht ausschließliche, weitgehende Nutzungsrechte, können oft Änderungen des Nutzers auch unter anderer Lizenz weitergeben
- Bsp:
 - Apple Public Source License
 - Netscape Public License

- Lizenzbedingungen von Open Source Software regeln und betreffen primär das Verhältnis zwischen
 - OSS Hersteller (Lizenzgeber) und 1. Nutzer (SW-Unternehmen = Lizenznehmer)
 - Interner Einsatz von OSS, ohne dass Software modifiziert oder weitergegeben wird, meist ohne rechtliche Hürden
 - Vertrag unterliegt oft ausländischem Recht!
 - Kann im Zweipersonalverhältnis noch überschaubar und handhabbar sein!
 - Anwendbares Recht?
 - Rechtswahl?
 - Was wird im Lizenzvertrag geregelt?
 - Achtung auf versteckte Bedingungen!
- Was passiert bei einer Weitergabe?
 - SW-Unternehmen schließt mit Drittem einen Vertrag ab
 - Nach österreichischem Recht

Open Source Software im mehrpersonalen Verhältnis



Open Source Software im mehrpersonalen Verhältnis



- Problematisches Aufeinandertreffen von Lizenzen!
 - Haftungsausschlüsse nach ausländischem Recht oft einfach und großzügig möglich, nach österreichischem Recht oft problematisch
 - OSS-Lizenzbedingungen können Haftungs- und Gewährleistungsausschlüsse vorgeben, die der SW-Hersteller schlecht oder gar nicht durchreichen kann
 - SW-Hersteller haftet und leistet Endkunden (können auch Konsumenten sein!) Gewähr, ohne dass er auf den OSS-Lizenzgeber zurückgreifen kann
- Welche Möglichkeit hat nun ein SW-Hersteller, der OSS in einem Projekt einsetzen möchte?

????????

- Hängt von der Frage ab: Wie OSS und eigene Leistung weitergegeben wird:
 - 1. Untrennbare Verbindung:
 - OSS ist untrennbarer Teil der Software
 - SW-Hersteller kann im Streitfall zwischen 2 Rechtsordnungen aufgerufen werden
 - Risikoabwägung für Einsatz von OSS erforderlich!
 - SW-Hersteller wird meist auf bekannte, gut betreute OSS mit aktiver Community setzen
 - Oft wird Risiko als überschaubar angesehen
 - 2. Rechtliche Trennung:
 - Leistung so erbringen, dass eigene Leistung eine Ergänzung der OSS darstellt
 - Vertraglich darstellen, dass Käufer sich OSS „selbst beschaffen“ muss
 - Vertragsgegenstand konkretisieren und genau beschreiben!
 - Darauf achten, dass der Kunde eine eigene OSS-Lizenz nimmt und direkter Vertragspartner mit OSS-Lieferant wird
 - Bei der Rechtsgewährung uU. OSS extra erwähnen

- Erklärung der Wertschöpfung und der Vorteile für den Kunden
 - Source Code ist verfügbar, kann eingesehen werden
 - Konkrete OSS hat sich in Praxis bereits bewährt
 - Aktive Community
- Nutzungsrechte
 - Vom Kunden geforderte Nutzungsrechte sind oft mit OSS nicht möglich
 - Oft sehr restriktiv: Ausschreibungen, die oft keinen Verhandlungsspielraum bieten
 - Teilweise bereits Umdenken, Möglichkeit des Einsatzes von OSS oft nicht mehr ausgeschlossen
- OSS-Risikomanagement
 - Dokumentation, was, wo, wofür, in welcher Fassung eingesetzt wird
 - Überwachung von Sicherheitslücken und neuen Versionen