

Nina Locher

## Urheber- und vertragsrechtliche Aspekte der Container-Virtualisierung

---

Längst hat sich die Virtualisierung als zentraler Innovationsmotor im heutigen Geschäftsleben etabliert. Laufend werden neue Lösungen zur Ressourceneinsparung und Steigerung der Systemverfügbarkeit entwickelt. Während die herkömmlichen virtuellen Maschinen von der juristischen Lehre bereits aufgegriffen wurden, ist die neuere «leichtgewichtige» Virtualisierungslösung des Containers bis anhin weitgehend unbehandelt geblieben. Nachfolgend soll aufgezeigt werden, wie der Einsatz der Container-Technologie urheber- und vertragsrechtlich zu beurteilen ist. Vertragsklauseln können in diesem Kontext auf beiden Ebenen Wirkungen zeitigen.

---

Beitragsart: Beiträge

Region: Schweiz

Rechtsgebiete: IT-Recht; Immaterialgüterrecht; IP-Recht; Urheberrecht; Vertragsrecht

Zitiervorschlag: Nina Locher, Urheber- und vertragsrechtliche Aspekte der Container-Virtualisierung, in: Jusletter IT 30. Juni 2021

## Inhaltsübersicht

1. Einleitung
2. Technische Grundlagen
  - 2.1. Hintergrund: Virtualisierung
    - 2.1.1. Grundlegendes
    - 2.1.2. Virtualisierungsgebiete und -technologien
  - 2.2. Container-basierte Virtualisierung
    - 2.2.1. Funktionsweise
    - 2.2.2. Vergleich virtueller Maschinen mit Containern
3. Softwareschutzrechtliche Beurteilung
  - 3.1. Urheberrechtlicher Schutz von Computerprogrammen
    - 3.1.1. Allgemeines
    - 3.1.2. Inhalt des Urheberrechts
  - 3.2. Gesetzliches Gebrauchs- und Weiterveräußerungsrecht
    - 3.2.1. Regelungsgehalt und Anwendungsbereich
    - 3.2.2. Sachlicher Umfang der gesetzlichen Gebrauchsbefugnis
  - 3.3. Softwareeinsatz in containerisierten Umgebungen
    - 3.3.1. Ablufenlassen
    - 3.3.2. Nutzung in mehreren Instanzen
    - 3.3.3. Sicherungskopien
    - 3.3.4. Modifikationen
4. Vertragliche Nutzungsbestimmungen
  - 4.1. Softwareüberlassungsvertrag
    - 4.1.1. Qualifikation
    - 4.1.2. Rechtlicher Rahmen
      - 4.1.2.1. Kerngehalt des Gebrauchsrechts
      - 4.1.2.2. Gültige Vereinbarung allgemeiner Lizenzbedingungen
  - 4.2. Lizenzbestimmungen im Bereich virtualisierter Umgebungen
    - 4.2.1. Relevante Lizenzformen
      - 4.2.1.1. Einfach- und Mehrfachlizenzen
      - 4.2.1.2. CPU- und Core-Lizenzen
    - 4.2.2. Einzelne Regelungen verschiedener Softwareanbieter
      - 4.2.2.1. Instanzenbasierte Lizenzierung
      - 4.2.2.2. Hardwarebasierte Lizenzierung
5. Verhältnis zwischen vertrags- und urheberrechtlicher Ebene
  - 5.1. Urheberrechtliche Wirkung vertraglicher Nutzungsbestimmungen
  - 5.2. Rechtsfolgen bei Lizenzvertragsverletzungen
    - 5.2.1. Gesetzliche Sanktionen
    - 5.2.2. Vertragliche Sanktionen
  - 5.3. Wirkung von Lizenzklauseln im Bereich der (Container-)Virtualisierung
    - 5.3.1. Regelungen nach Zahl der Instanzen
    - 5.3.2. Hardwarebezogene Beschränkungen
6. Fazit

## 1. Einleitung

[1] Der fortlaufende technische Wandel wirft bei der Entwicklung und beim Einsatz von Computerprogrammen aus urheberrechtlicher wie auch aus vertragsrechtlicher Perspektive verschiedene neue Fragestellungen auf. Im Zentrum der rechtlichen Einordnungen dieser Arbeit steht das Gebiet der Virtualisierung – ein IT-Trend, welcher spätestens seit anfangs des 21. Jahrhunderts nicht mehr zu bremsen ist und in dessen Zuge ständig neue Lösungen entwickelt werden. Um Ressourcen zu sparen und die Verfügbarkeit von Systemen zu steigern, werden Hardware-

oder Softwarekomponenten abstrahiert und auf virtueller Ebene bereitgestellt. Während die herkömmlichen virtuellen Maschinen, welche einen kompletten Rechner virtualisieren, in der juristischen Lehre schon teilweise Eingang gefunden haben, ist die neuere «leichtgewichtige» Virtualisierungslösung des Containers bis anhin noch weitgehend unbehandelt. Bei diesem Ansatz wird das Betriebssystem des Hosts genutzt und es werden lediglich die Anwendungen voneinander isoliert. Aus der informationstechnologischen Sicht unterscheiden sich beide Architekturen erheblich, doch rechtlich gesehen wird selten differenziert, weshalb bei der juristischen Betrachtung der Container-Virtualisierung stets auch die Situation bei virtuellen Maschinen berücksichtigt werden muss. Um eine technische Basis für die Arbeit zu schaffen, werden zuerst die Hintergründe und Grundlagen beider Technologien dargelegt (2.). Anschliessend stellt sich im Falle der Softwareüberlassung auf Dauer die Frage, wie der Einsatz eines Programms in solch virtualisierten Umgebungen urheberrechtlich (3.) zu beurteilen ist, falls kein oder kein gültiger Vertrag vorliegt. Sodann ist die vertragsrechtliche Ebene (4.) bei Vorliegen eines Softwarevertrages darzulegen: Welche Vereinbarungen werden im Kontext von (container-)virtualisierten Umgebungen getroffen und welcher rechtliche Rahmen ist dabei zu berücksichtigen? Zur Diskussion stehen hiernach die Modelle grosser Softwareanbieter auf dem Markt sowie die Probleme, die in der Praxis auftreten und die die Vorteile einer Virtualisierung nicht selten auch zunichtemachen können. Schliesslich soll die Wechselwirkung zwischen der urheberrechtlichen und der vertragsrechtlichen Ebene (5.) genauer betrachtet werden, da Lizenzvereinbarungen durchaus verschiedene Wirkungen zukommen können, was zu unterschiedlichen Rechtsfolgen führt. Auf dieser Basis ist zu untersuchen, welche Rechtsnatur die spezifischen Vereinbarungen im Bereich virtualisierter bzw. containerisierter Umgebungen haben und inwiefern sie auch unwirksam sein können.

## 2. Technische Grundlagen

### 2.1. Hintergrund: Virtualisierung

#### 2.1.1. Grundlegendes

[2] Die *Virtualisierung* ist als zentraler Innovationsmotor aus dem heutigen Geschäftsleben nicht mehr wegzudenken. Die Virtualisierungswelle hat bereits Mitte der 60er-Jahren begonnen, als IBM Versuchssysteme zur virtuellen Speicherverwaltung entwickelte und somit die Grundlagen für ein hardwareunabhängigeres Design von Rechnersystemen legte. Es gibt keine allgemeingültige Definition des Virtualisierungsbegriffs.<sup>1</sup> In der Informatik ist mit Virtualisierung prinzipiell die Abstraktion einer (physikalischen) Ressource (z.B. Prozessoren, Speicher, etc.) mithilfe eines Stückes Software gemeint.<sup>2</sup> Neben physikalischen Einheiten können auch Softwareobjekte abstrahiert werden.<sup>3</sup> Für die unterschiedlichen Anwendungsszenarien wurden fortlaufend viele

---

<sup>1</sup> CHRISTOPH ARNOLD/MICHEL RODE/JAN SPERLING, KVM Best Practices: Virtualisierungslösungen für den Enterprise-Bereich, Heidelberg 2012, S. 1 f.

<sup>2</sup> CHRISTOPH MEINEL/CHRISTIAN WILLEMS/SEBASTIAN ROSCHKE/MAXIM SCHNJAKIN, Virtualisierung und Cloud Computing: Konzepte, Technologiestudie, Marktübersicht, Potsdam 2011, S. 10.

<sup>3</sup> Vgl. STEFAN LUBER/FLORIAN KARLSTETTER, Was ist Virtualisierung?, <https://www.cloudcomputing-insider.de/was-ist-virtualisierung-a-756279/> (alle Internetquellen wurden am 12. Januar 2021 zuletzt besucht).

verschiedene Technologien, Produkte und Geschäftsmodelle entwickelt.<sup>4</sup> Es braucht dabei immer ein Hostsystem, welches als Basis für die virtuellen Komponenten dient und sodann mit Guests, also mit virtuellen Gastsystemen bestückt wird.<sup>5</sup> Im allgemeinen Sprachgebrauch wird mit der Virtualisierung meist die herkömmliche Hardware-Virtualisierung gemeint, bei welcher physikalische Computer zu virtuellen Maschinen (sog. VM), abstrahiert werden.<sup>6</sup>

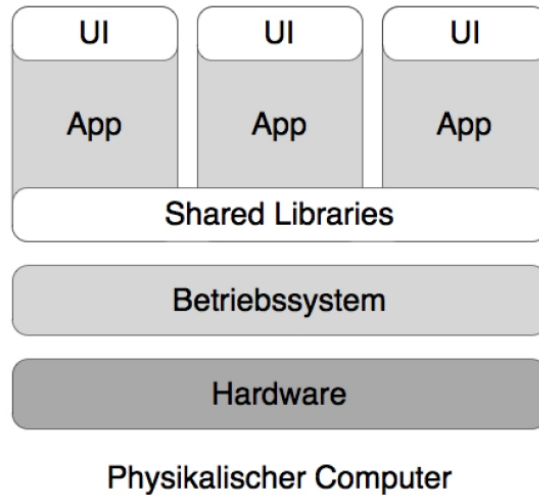


Abbildung 1: System ohne Virtualisierung<sup>7</sup>

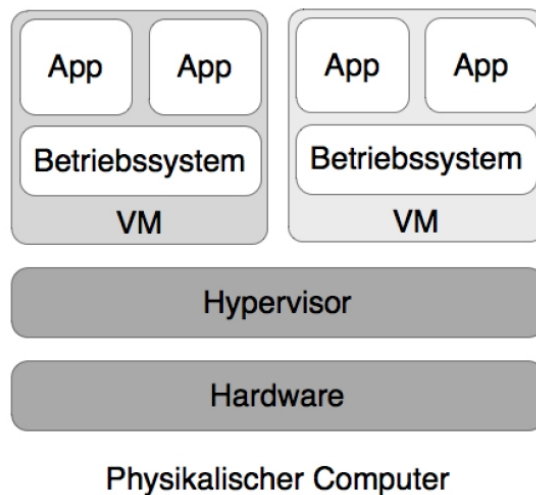


Abbildung 2: System mit Virtualisierung (Hardwarevirtualisierung)<sup>8</sup>

<sup>4</sup> MEINEL/WILLEMS/ROSCHKE/SCHNJAKIN (Fr. 2), S. 7.

<sup>5</sup> Vgl. JANINA WALKER/DANIEL TH. GERTSCH, Virtual Book – Grundlagenwerk über die Virtualisierung, Frick 2015, S. 34.

<sup>6</sup> MEINEL/WILLEMS/ROSCHKE/SCHNJAKIN (Fr. 2), S. 13.

<sup>7</sup> MEINEL/WILLEMS/ROSCHKE/SCHNJAKIN (Fr. 2), S. 10.

<sup>8</sup> Ebd.

[3] Die Mehrheit der virtuellen Systemumgebungen haben den gleichen Zweck: vorhandene Systemressourcen effizient und sicher nutzen.<sup>9</sup> Ursprüngliche Serversysteme verbrauchten viel Strom und Platz, waren aber alles andere als ausgelastet und zudem auch nicht kostengünstig in der Anschaffung. Dank hocheffizienter Virtualisierungstechnologie kann heutzutage über nahezu 100 % der Ressourcen leistungsfähiger Hardware verfügt werden. Als Teil von Green IT trägt die Virtualisierung dank Stromeinsparung zur Verbesserung der Energieeffizienz bei und reduziert die weltweite Hardwareproduktion.<sup>10</sup> Von Vorteil ist auch die Steigerung der Verfügbarkeit von Systemen durch deren einfache Wiederherstellung oder Verschiebung.<sup>11</sup>

### 2.1.2. Virtualisierungsgebiete und -technologien

[4] Generell wird zwischen den Gebieten der Hard- und Software-Virtualisierung unterschieden. Erstere stellt virtuelle Hardware-Ressourcen (z.B. CPU oder Speicher) bereit, zweite hingegen Anwendungen oder Betriebssysteme.<sup>12</sup> Die Hardware-Virtualisierung findet hauptsächlich im Serverbereich Einsatz. Zugleich ist dieses Konzept auch auf den Gebieten der Speicher- und der Netzwerkvirtualisierung verbreitet, welche in dieser Arbeit nicht weiter berücksichtigt werden.<sup>13</sup> Die Hardware-Virtualisierung ermöglicht auf einem «echten Rechner» das gleichzeitige Betreiben mehrerer virtuellen Maschinen, die sich die Ressourcen teilen.<sup>14</sup> Die dabei angewendete Technologie und logische Schicht zwischen Host- und virtuellem Gastsystem nennt sich *Hypervisor* bzw. *Virtual Machine Monitor*.<sup>15</sup> Dabei wird zwischen zwei Typen unterschieden: solchen, die direkt auf der Hardware – also *bare metal* – laufen und solchen, die als Applikation innerhalb eines Betriebssystems agieren.<sup>16</sup> Dem virtuellen System wird durch den Hypervisor vorgetäuscht, es sei der alleinige Nutzer der Hardware-Ressourcen.<sup>17</sup>

[5] Klassische Einsatzfelder der Software-Virtualisierung sind die Desktop- (auch Präsentations-) sowie die Anwendungsvirtualisierung. Bei der erstgenannten wird üblicherweise mittels *thin clients* auf den Desktop oder auf Anwendungen eines entfernten Servers zugegriffen (auch *Terminal Computing* genannt).<sup>18</sup> Für die folgende Darstellung zentral ist jedoch vor allem die Anwendungsvirtualisierung. Deren Ziel ist die Vermeidung von Konflikten zwischen Anwendungen untereinander oder zwischen Anwendung und Betriebssystem.<sup>19</sup> Während dies traditionell u.a. durch Techniken wie *Streaming* erreicht wurde, erfolgt die Umsetzung heutzutage auch mit-

---

<sup>9</sup> Vgl. ARNOLD/RODE/SPERLING (Fn. 1), S. 2.

<sup>10</sup> Vgl. zum Ganzen WALKER/GERTSCH (Fn. 5), S. 35 ff. und 42 ff.

<sup>11</sup> Vgl. MALTE GRÜTZMACHER, Software-Urheberrecht und Virtualisierung, in: Helmut Redeker/Peter Hopfen (Hrsg.): DGRI Jahrbuch 2011, Köln 2012, S. 169–177, S. 170 (zit. GRÜTZMACHER, Virtualisierung).

<sup>12</sup> LUBER/KARLSTETTER (Fn. 3), Was ist Virtualisierung?

<sup>13</sup> MEINEL/WILLEMS/ROSCHKE/SCHNJAKIN (Fn. 2), S. 12.

<sup>14</sup> Vgl. MEINEL/WILLEMS/ROSCHKE/SCHNJAKIN (Fn. 2), S. 13.

<sup>15</sup> ARNOLD/RODE/SPERLING (Fn. 1), S. 3.

<sup>16</sup> Vgl. MEINEL/WILLEMS/ROSCHKE/SCHNJAKIN (Fn. 2), S. 13.

<sup>17</sup> Vgl. ARNOLD/RODE/SPERLING (Fn. 1), S. 3.

<sup>18</sup> KONRAD MEIER, Infrastrukturkonzepte für virtualisierte wissenschaftliche Forschungsumgebungen, Diss., Freiburg im Breisgau 2017, S. 15.

<sup>19</sup> MEIER (Fn. 18), S. 16.

tels virtueller Isolation durch die folglich relevante *Containerisierung* von Applikationen (dazu Kap. 2.2.). Zudem ist es auch möglich, beide Ansätze zu kombinieren.<sup>20</sup>

## 2.2. Container-basierte Virtualisierung

### 2.2.1. Funktionsweise

[6] Gängige Hypervisor-Technologien wurden in den letzten Jahren fortlaufend durch neue Virtualisierungsansätze auf Betriebssystemebene verdrängt bzw. ergänzt. Im Vordergrund steht dabei die leichtgewichtigere Container-Virtualisierung.<sup>21</sup> Man spricht in diesem Zusammenhang nicht mehr von virtuellen Maschinen, sondern von *Containern* oder *Jails*. Bei Containern handelt es sich um voneinander abgetrennte Laufzeitumgebungen, die sich den gleichen Betriebssystemkern (*Kernel*) teilen. Die darauf laufenden einzelnen Applikationen wissen nichts von den anderen Containern oder deren Ressourcen.<sup>22</sup>

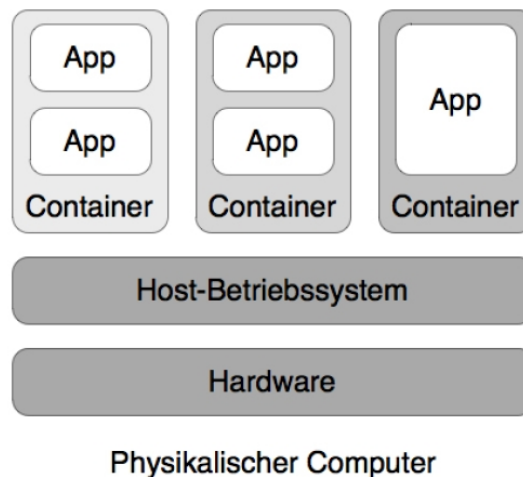


Abbildung 3: Container-basierte Virtualisierung<sup>23</sup>

[7] Durch die Mitbenutzung des Betriebssystemkerns des Hosts können Systemressourcen wie Prozessor, Netzwerk oder Speicher effizient genutzt werden und Applikationen über Systeme hinweg verschoben werden, ohne dabei das komplette Betriebssystem mit zu migrieren.<sup>24</sup> Verbreitete Containertechnologien sind z.B. Linux Containers (LXC), Docker, Rocket (rkt) oder OpenVZ, wobei Docker am meisten verbreitet ist. Insbesondere der Cloud-Computing-Markt wurde durch das Aufkommen der Container-Virtualisierung stark geprägt.<sup>25</sup> Aufgrund der unterschiedlichen

<sup>20</sup> Vgl. ROBERT SHELDON, Applikationscontainer-Technologie oder Virtualisierung?, <https://tinyurl.com/yd2xfqp7>.

<sup>21</sup> Vgl. TOM NOLLE, Container oder VMs: Welche Technologie eignet sich besser für die Cloud?, <https://tinyurl.com/yae3m7kc>.

<sup>22</sup> Zum Ganzen MEINEL/WILLEMS/ROSCHKE/SCHNJAKIN (Fn. 2), S. 17.

<sup>23</sup> Ebd.

<sup>24</sup> CHRISTOPH HAAR/ERIK BUCHMANN, IT-Grundschutz für die Container-Virtualisierung mit dem neuen BSI-Baustein SYS. 1.6., in: INFORMATIK 2019, Lecture Notes in Informatic (LNI), Gesellschaft für Informatik, Bonn 2019, S. 479–491, S. 481.

<sup>25</sup> Vgl. SONG WU/HAI JIN/HANG HUANG/BOWEN RUAN, A Performance Study of Containers in Cloud Environment, in: 10th Asia-Pacific Services Computing Conference, APSCC 2016: Advances in Services Computing, Zhangjiajie 2016, S. 343–356, S. 344.

Ausgestaltung und Anwendung können die Containerprodukte weiter in *application containers* und *system containers* unterteilt werden. Applikations-Container (z.B. Docker und rkt) erwirken die Abkapselung einer einzigen Anwendung oder eines Prozesses in einem *container image*.<sup>26</sup> Mit Docker kann man Applikationen und all deren Abhängigkeiten (Bibliotheken, Hilfsprogramme und Konfigurationsdateien) in einen Container packen.<sup>27</sup> Die extreme Simplifizierung einer Anwendung birgt vor allem bei Mikroservice-Architekturen grosse Vorteile. Systemcontainer (z.B. LXC oder OpenVZ) stellen dagegen ähnlich wie virtuelle Maschinen ein vollwertiges Betriebssystem bereit.<sup>28</sup> Innovative und cloudbasierte Anwendungen können durch Container auf eine agile und kosteneffiziente Weise umgesetzt, ausgeliefert und gewartet werden.<sup>29</sup>

### 2.2.2. Vergleich virtueller Maschinen mit Containern

[8] Der Hauptunterschied zwischen den beiden Virtualisierungslösungen liegt beim Betriebssystemkern.<sup>30</sup> Container nutzen alle denselben Kernel und das bestehende Betriebssystem wird lediglich modifiziert, um weitere Isolation bereitzustellen, während bei Hypervisor-Technologien ein oder mehrere vollständige Gast-Betriebssysteme betrieben werden.<sup>31</sup> Abbildung 4 soll beide Architekturen illustrieren.

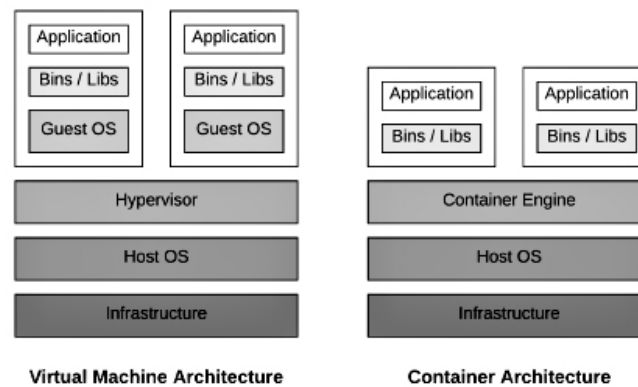


Abbildung 4: Vergleich der beiden Architekturen (virtuelle Maschine und Container)<sup>32</sup>

[9] Im Gegensatz zu Containern, die sich das Betriebssystem teilen, benutzen virtuelle Maschinen eine Ebene weiter unten dieselbe physische Hardware.<sup>33</sup> Ein Hypervisor betreibt oft verschiedene Host-Betriebssysteme und benötigt grosse Mengen an Hardware-Ressourcen.<sup>34</sup> Der Hypervisor fällt bei Containern weg, wodurch ein geringerer Management-Aufwand verursacht wird.

<sup>26</sup> Ebd.; vgl. auch WES FELTER/ALEXANDRE FERREIRA/RAM RAJAMONY/JUAN RUBIO, An updated performance comparison of virtual machines and linux containers, IBM Research Report, Austin 2014, S. 1–11, S. 3.

<sup>27</sup> Vgl. MATHIJS J. SCHEEPERS, Virtualization and Containerization of Application Infrastructure: A Comparison, in: 21st twente student conference on IT, Bd. 1, Enschede 2014, S. 1–7, S. 1.

<sup>28</sup> Vgl. WU/JIN/HUANG/RUAN (Fn. 25), S. 344.

<sup>29</sup> HAAR/BUCHMANN (Fn. 24), S. 479.

<sup>30</sup> WU/JIN/HUANG/RUAN (Fn. 25), S. 346.

<sup>31</sup> Vgl. FELTER/FERREIRA/RAJAMONY/RUBIO (Fn. 26), S. 2.

<sup>32</sup> WU/JIN/HUANG/RUAN (Fn. 25), S. 345.

<sup>33</sup> Vgl. NOLLE (Fn. 21), Container oder VMs.

<sup>34</sup> Vgl. SCHEEPERS (Fn. 27), S. 1.

Der reduzierte *Overhead* ermöglicht den Betrieb einer grösseren Anzahl von Anwendungen als dies bei virtuellen Maschinen möglich ist.<sup>35</sup> Ausserdem sind Container um Einiges performanter, denn sie brauchen weniger Speicherressourcen und lassen sich schneller starten als virtuelle Maschinen.<sup>36</sup> Als schwieriger erweist sich die zuverlässige Isolation der Container voneinander, was zu schwerwiegenden Sicherheitslücken führen kann.<sup>37</sup> Beide Virtualisierungsansätze bezwecken Portabilität und Isolation sowie Nutzungsoptimierung der Hardware-Ressourcen, sind aber für unterschiedliche Anwendungsfälle geeignet.<sup>38</sup> Ausserdem ist auch eine Kombination möglich, was insbesondere in Cloud-Umgebungen nicht selten vorkommt, aber letztendlich trotzdem wieder unnötigen *Overhead* generiert und redundant ist betreffend die Performanz.<sup>39</sup> Je nach Cloud-Typ eignen sich Container (v.a. bei Private Clouds) oder virtuelle Maschinen besser.<sup>40</sup>

### 3. Softwareschutzrechtliche Beurteilung

[10] Auf urheberrechtlicher Ebene stellt sich die Frage, wie der Einsatz eines urheberrechtlich geschützten Computerprogramms in virtualisierten Umgebungen zu beurteilen ist. Wie weit gehen die Ausschliesslichkeitsrechte des Urhebers und in welchen Fällen schränken gesetzliche Gebrauchsrechte des Erwerbers diese ein?

#### 3.1. Urheberrechtlicher Schutz von Computerprogrammen

[11] Im Folgenden wird primär auf den urheberrechtlichen Schutz in der Schweiz Bezug genommen, dessen rechtliche Grundlagen vor allem das URG und die URV bilden. Weiteren Schutz bieten die Sonderbestimmungen über Computerprogramme in Art. 10 Ziff. 1 und Art. 11 des TRIPS-Abkommens sowie Art. 4 und 7 des WIPO-Urheberrechtsvertrags.<sup>41</sup> Für die Auslegung des URG ist ausserdem die Softwareschutz-Richtlinie<sup>42</sup> bedeutsam, da der schweizerische Gesetzgeber sich an dieser orientiert hatte.<sup>43</sup> Erwähnenswert ist, dass auch das Patent-, Design- und das Lauterkeitsrecht Computerprogrammen in gewissen Fällen Schutz bieten.<sup>44</sup>

---

<sup>35</sup> Vgl. NOLLE (Fn. 21), Container oder VMs.

<sup>36</sup> Vgl. WU/JIN/HUANG/RUAN (Fn. 25), S. 345; vgl. SCHEEPERS (Fn. 27), S. 7.

<sup>37</sup> Vgl. HAAR/BUCHMANN (Fn. 24), S. 481 und 487 f.

<sup>38</sup> SCHEEPERS (Fn. 27), S. 7.

<sup>39</sup> Vgl. FELTER/FERREIRA/RAJAMONY/RUBIO (Fn. 26), S. 1; WU/JIN/HUANG/RUAN (Fn. 25), S. 344.

<sup>40</sup> NOLLE (Fn. 21), Container oder VMs.

<sup>41</sup> GEORG RAUBER, Use Restrictions in Softwareverträgen, in: Florian S. Jörg/Olivier Arter (Hrsg.): Internet-Recht und IT-Verträge, 2. Aufl., Bern 2009, S. 139–177, S. 141 (zit. RAUBER, Use Restrictions).

<sup>42</sup> Richtlinie 91/250/EWG des Rates vom 14. Mai 1991 über den Rechtsschutz von Computerprogrammen (heute kodifiziert als Richtlinie 2009/24/EG) (nachfolgend «Softwareschutz-RL»).

<sup>43</sup> Botschaft zu einem Bundesgesetz über das Urheberrecht und verwandte Schutzrechte (Urheberrechtsgesetz, URG), zu einem Bundesgesetz über den Schutz von Topographien von integrierten Schaltungen (Topographiengesetz, ToG) sowie zu einem Bundesbeschluss über verschiedene völkerrechtliche Verträge auf dem Gebiete des Urheberrechts und der verwandten Schutzrechte vom 19. Juni 1989, BBl. 1989 III 477 ff., S. 609.

<sup>44</sup> WOLFGANG STRAUB/PHILIPP RÜFENACHT, Rechtlicher Schutz von Softwareentwicklungen, in: Conrad Weinmann/Peter Münch/Jürg Herren (Hrsg.): Schweizer IP-Handbuch, Basel 2021, 2. Aufl., S. 503–552, Rn. 0.2; STRAUB, Softwareschutz, Zürich 2011, Rn. 506 ff.



### 3.1.1. Allgemeines

[12] Zu den *Schutzgegenständen* des URG gehören grundsätzlich Werke der Literatur und Kunst gemäss Art. 2 Abs. 1 URG, die sodann in Abs. 2 beispielhaft aufgeführt werden (literarische, musikalische Werke etc.). Der nachfolgende Abs. 3 gewährt auch Computerprogrammen urheberrechtlichen Schutz, indem diese im Sinne des Gesetzes als Werke angesehen werden. Im Gesetz finden sich zahlreiche weitere Sonderbestimmungen zu Computerprogrammen, da die generell geltenden Regeln, welche ursprünglich auf Schriftsteller, Künstler etc. zugeschnitten worden waren, sich vielfach als ungeeignet erwiesen.<sup>45</sup>

[13] Computerprogramme umfassen alle in einer Programmiersprache verfassten vollständigen Verfahren zur Lösung einer bestimmten Aufgabe. Schutz geniessen der Quellcode und der Objektcode.<sup>46</sup> Die allgemeinen urheberrechtlichen *Schutzvoraussetzungen* sind gleichermassen auch beim Schutz von Computerprogrammen zu erfüllen. Es muss sich also gemäss Art. 2 Abs. 1 URG um eine geistige Schöpfung handeln, die über einen individuellen Charakter verfügt.<sup>47</sup> Der vorgegebene Zweck lässt bei der Programmierung je nach Aufgabenstellung oft nicht allzu grossen Gestaltungsspielraum zu, weshalb nach h.L. keine hohen Anforderungen an die Individualität zu stellen sind.<sup>48</sup> Das zweite Schutzkriterium, das Vorliegen einer geistigen Schöpfung, ist dann erfüllt, wenn es sich um ein von Menschen entwickeltes Computerprogramm handelt.<sup>49</sup> Nach dem Schöpferprinzip, dem das URG folgt, ist der *Inhaber der Rechte* gemäss Art. 6 URG grundsätzlich der Urheber, d.h. die natürliche Person, die das Werk geschaffen hat.<sup>50</sup> Der *Schutzumfang* ist gleichzustellen mit dem Umfang der Individualität. Bei einem Computerprogramm sind also alle Sequenzen einer oder mehreren Programmzeilen geschützt, die das Erfordernis der Individualität erfüllen.<sup>51</sup>

### 3.1.2. Inhalt des Urheberrechts

[14] Was den *Inhalt des Urheberrechts* anbelangt, steht dem Rechtsinhaber ein umfassendes bzw. ausschliessliches Recht zu, über die Verwendung seines Werks zu entscheiden. Davon erfasst sind einerseits die nicht übertragbaren Urheberpersönlichkeitsrechte und die (in casu relevanten) frei übertragbaren Urhebervermögensrechte.<sup>52</sup> Im Softwarebereich sind letztere in die folgenden selbständig verwertbaren ausschliesslichen Nutzungsrechte zu unterteilen: Vervielfältigungs-, Verbreitungs-, Vorführungs-, Bearbeitungs- und Vermietungsrechte.<sup>53</sup> Ebenfalls dazugezählt werden auch das Recht auf Dekompilation von Computerprogrammen und die Befugnis zur Nutzung

---

<sup>45</sup> Vgl. STRAUB/RÜFENACHT (Fn. 44), Rn. 0.1.

<sup>46</sup> Zum Objektcode BGE 125 III 243; zum Sourcecode OGer. ZH, sic! 2011, 230; vgl. EUGEN MARBACH/PATRIC DUCREY/GREGOR WILD, Immaterialgüter- und Wettbewerbsrecht, 4. Aufl., Bern 2017, Rn. 278.

<sup>47</sup> STRAUB/RÜFENACHT (Fn. 44), Rn. 4.

<sup>48</sup> BGE 125 III 328 E. 4b; vgl. STRAUB/RÜFENACHT (Fn. 4), Rn. 4.2; EMIL F. NEFF/MATTHIAS ARN, Urheberrechtlicher Schutz der Software, in: SIWR II/2, Basel 1998, S. 71; FELIX H. THOMANN, Softwareschutz durch das Urheberrecht, in: Felix H. Thomann/Georg Rauber (Hrsg.): Softwareschutz, Bern 1998, S. 1–58, S. 13.

<sup>49</sup> Vgl. GEORG RAUBER, Computersoftware, in: Magda Streuli-Youssef (Hrsg.): Urhebervertragsrecht, Zürich 2006, S. 119–260, S. 130 f. (zit. RAUBER, Computersoftware).

<sup>50</sup> MARBACH/DUCREY/WILD (Fn. 46), Rn. 288.

<sup>51</sup> Vgl. STRAUB/RÜFENACHT (Fn. 44), Rn. 6.

<sup>52</sup> Vgl. STRAUB (Fn. 44), Rn. 105.

<sup>53</sup> CLARA-ANN GORDON, Handel mit Secondhand-Volumenlizenzen – auch ohne Zustimmung des Urhebers zulässig?, sic! 2008, S. 758–764, S. 760.

innerhalb von Netzwerken, Software as a Service etc.<sup>54</sup> Die vorliegende Darstellung befasst sich vorwiegend mit dem Vervielfältigungsrecht nach Art. 10 Abs. 2 lit. a URG und daneben auch mit dem Änderungsrecht gemäss Art. 11 Abs. 1 URG.

[15] Das *Vervielfältigungsrecht* (Recht zur Herstellung von Werkexemplaren) als wichtigste Einzelbefugnis erstreckt sich auf jede mögliche Art der Vervielfältigung, d.h. sowohl auf die Erstellung permanenter Kopien wie auch auf die Erstellung flüchtiger Kopien. Bei jeder Vervielfältigung, die nicht durch die bestimmungsgemässe Verwendung (dazu Kap. 3.2.2.) gedeckt wird, bedarf es der Zustimmung des Rechtsinhabers.<sup>55</sup>

[16] Weiter hat der Urheber auch das ausschliessliche *Änderungs- bzw. Bearbeitungsrecht*. Ihm steht zu, jegliche Änderungen seines Werks zu verbieten. Übliche Bearbeitungsarten bei Software sind die Übersetzung aus einer Programmiersprache in eine andere und die Wartung, welche in der Regel mit einer Änderung des Programms, insbesondere mit einer Fehlerkorrektur einhergeht.<sup>56</sup> Falls der Rechtsinhaber die notwendigen Programmänderungen zur Fehlerkorrektur nicht unter angemessenen Konditionen anbietet, ist der Erwerber dazu ggf. aus seinem Gebrauchsrecht berechtigt (dazu Kap. 3.2.2.).<sup>57</sup>

### 3.2. Gesetzliches Gebrauchs- und Weiterveräusserungsrecht

[17] Neben der im vorherigen Kapitel behandelten positiven Umschreibung der urheberrechtlichen Kontrollrechte enthält das Gesetz zwecks Ausgleich der vielfältigen Interessen der Softwarenutzer auch Schrankenregelungen, die die Kontrolle des Rechtsinhabers beschränken.<sup>58</sup> Die zentrale softwarerechtliche Schranke bildet das *gesetzliche Gebrauchs- und Weiterveräusserungsrecht* nach Art. 12 Abs. 2 URG.<sup>59</sup> Der rechtmässige Erwerber eines Computerprogramms hat demnach das Recht, das Werkexemplar zu gebrauchen und weiterzuveräussern. Die Tragweite dieses Artikels ist in der Lehre zum Teil umstritten.<sup>60</sup> In dieser Arbeit spielt Art. 12 Abs. 2 URG eine wichtige Rolle, da es u.a. vom Gebrauchsrecht abhängt, inwieweit der Erwerber die Software in virtualisierten Umgebungen einsetzen darf und in welchen Situationen der Urheber einschreiten darf. Weiter darf der Softwareerwerber zur Schaffung der Interoperabilität eines Programms gemäss Art. 21 URG i.V.m. Art. 17 Abs. 2 und 3 URV dessen Programmcode entschlüsseln und hat nach Art. 24 Abs. 2 URG das Recht, eine Sicherheitskopie herzustellen.<sup>61</sup> Nach der weiteren Schrankenbestimmung in Art. 24a URG ist überdies die bloss flüchtige Kopie eines Werks erlaubt, wenn sie der rechtmässigen Nutzung dient und ihr keine eigenständige wirtschaftliche Bedeutung zukommt.<sup>62</sup> Sonstige urheberrechtliche Schranken wie z.B. das Zitatrecht oder die Parodiefreiheit spielen im Softwarebereich eine untergeordnete Rolle.<sup>63</sup> Die im URG sonst geltenden Befugnisse

---

<sup>54</sup> STRAUB/RÜFENACHT (Fn. 44), Rn. 10.1.

<sup>55</sup> Vgl. zum Ganzen THOMANN (Fn. 48), S. 25.

<sup>56</sup> Vgl. THOMANN (Fn. 48), S. 30 f.

<sup>57</sup> STRAUB/RÜFENACHT (Fn. 44), Rn. 12.1; RAUBER (Fn. 41), Use Restrictions, S. 154.

<sup>58</sup> Vgl. RAUBER (Fn. 41), Use Restrictions, S. 143.

<sup>59</sup> STRAUB (Fn. 44), Rn. 142; vgl. GIANNI FRÖHLICH-BLEULER, *Softwareverträge*, 2. Aufl., Bern 2014, Rn. 103.

<sup>60</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 126

<sup>61</sup> Vgl. RAUBER (Fn. 41), Use Restrictions, S. 144.

<sup>62</sup> Ebd.

<sup>63</sup> Vgl. STRAUB (Fn. 44), Rn. 145 f.

des Erwerbers zum Eigengebrauch und zur Vermietung gelangen nach Art. 19 Abs. 4 URG und Art. 13 Abs. 4 URG bei Software nicht zu Anwendung.<sup>64</sup>

### 3.2.1. Regelungsinhalt und Anwendungsbereich

[18] Voraussetzung für die Anwendung von Art. 12 Abs. 2 URG ist die Erschöpfung.<sup>65</sup> Mit der Veräusserung des Werkexemplars oder der Zustimmung zu einer solchen verliert der Urheber das Verbreitungsrecht für das betreffende Exemplar, nicht aber das Urheberrecht. Es erschöpft sich also lediglich ein Element des Urheberrechts.<sup>66</sup> Dem Erwerber des Exemplars werden zwei Befugnisse zugestanden: einerseits das *Recht auf Weiterveräusserung*, welches vor allem beim Vertrieb von Gebrauchtssoftware wichtig ist, und andererseits das vorliegend entscheidende *Gebrauchsrecht*.<sup>67</sup> Es soll damit der Werkgenuss des Erwerbers abgesichert werden.<sup>68</sup> Anders als bei anderen Werkkategorien wird dem Softwareerwerber kein Recht auf Weitervermietung vermittelt.<sup>69</sup> Damit die Erschöpfung eines Programmexemplars eintritt, muss es mit der Zustimmung des Berechtigten veräussert werden.<sup>70</sup> Generell wird bejaht, dass die Regelung auch bei online übertragener Software greift.<sup>71</sup> Nicht bei allen Software-Übertragungen ist das Merkmal der Veräusserung erfüllt (zum Softwareüberlassungsvertrag Kap. 4.1.1.). Der Veräusserungsbegriff von Art. 12 Abs. 2 URG ist mit demjenigen des Obligationen- oder Sachenrechts nicht deckungsgleich, sondern geht weiter.<sup>72</sup> Ausschlaggebend ist das definitive Entlassen der Programmkopien aus dem Herrschaftsbereich des Rechtsinhabers zum zeitlich unlimitierten Gebrauch. Ferner ist die Kopie gegen Einmalentgelt, ggf. Ratenzahlungen oder gratis zu überlassen.<sup>73</sup> Demgemäss findet das Gebrauchs- und Weiterveräusserungsrecht auch auf Softwareüberlassungen auf Dauer (sog. unechte Lizenzen) Anwendung. Nicht anwendbar ist dies nach h.L. bei Softwareüberlassungen auf Zeit (sog. echte Lizenzen), bei welchen wiederkehrende Nutzungsgebühren die Norm sind und somit mangels Veräusserung keine Erschöpfung eintritt.<sup>74</sup> Eine andere Ansicht vertritt RAUBER, welcher beim *Gebrauchsrecht* den Anwendungsbereich auch auf echte Lizenzen ausdehnen möchte, da in Art. 5 Abs. 1 der Softwareschutz-RL klar auch die zeitlich limitierte Überlassung als Anwendungsfall konstituiert wird und beim Gebrauchsrecht nicht an die Erschöpfung angeknüpft

---

<sup>64</sup> RAUBER (Fn. 41), Use Restrictions, S. 144.

<sup>65</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 126 und 140.

<sup>66</sup> Vgl. DENIS BARRELET/WILLI EGLOFF, Das neue Urheberrecht, Kommentar zum Bundesgesetz über das Urheberrecht und verwandte Schutzrechte, 4. Aufl., Bern 2020 (zit. URG Komm. – Bearbeiter, Art. ... N ...), URG Komm. – BARRELET/EGLOFF, Art. 12 N 1.

<sup>67</sup> Vgl. RAUBER (Fn. 41), Use Restrictions, 145; FRÖHLICH-BLEULER (Fn. 59), Rn. 140, 146 ff. und 183 ff.

<sup>68</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 140.

<sup>69</sup> Vgl. URG Komm. – BARRELET/EGLOFF, Art. 12 N 21; vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 175.

<sup>70</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 141.

<sup>71</sup> STRAUB (Fn. 44), Rn. 157; GORDON (Fn. 53), S. 761; FRÖHLICH-BLEULER (Fn. 59), Rn. 144.

<sup>72</sup> LUKAS MORSCHER/LARA DORIGO, Software-Lizenzverträge, Erschöpfung bei Computerprogrammen und Gebrauchthandel mit Softwarelizenzen, in: Florian S. Jörg/Olivier Arter (Hrsg.): Internet-Recht und IT-Verträge, 2. Aufl., Bern 2009, S. 17–72, S. 40.

<sup>73</sup> STRAUB/RÜFENACHT (Fn. 44), Rn. 18.1 und 19; STRAUB (Fn. 44), Rn. 153 ff.; JOSEF GELLIS, Softwarelizenz: Die Stellung des Lizenznehmers bei Veräusserung des Schutzrechts durch den Lizenzgeber oder bei dessen Konkurs, sic! 2005, S. 439–452, S. 440; MORSCHER/DORIGO (Fn. 72), 40 f.

<sup>74</sup> Vgl. STRAUB/RÜFENACHT (Fn. 44), Rn. 18 ff.; GELLIS (Fn. 73), S. 441; FRÖHLICH-BLEULER (Fn. 59), Rn. 143 und 150; NEFF/ARN (Fn. 48), S. 253; MORSCHER/DORIGO (Fn. 72), S. 41.

wird.<sup>75</sup> Diese Arbeit folgt jedoch der h.L. und konzentriert sich nur auf die Vereinbarungen zur Softwareüberlassung auf Dauer (Kap. 4.).

### 3.2.2. Sachlicher Umfang der gesetzlichen Gebrauchsbefugnis

[19] Um den Umfang der gesetzlichen Gebrauchsbefugnis zu eruieren, ist eine Interessenabwägung zwischen den Interessen des Erwerbers und denjenigen des Rechtsinhabers vorzunehmen.<sup>76</sup> Anders als bei der «Konsumierung» anderer Werke ist der Softwareerwerber auf gewisse zum Gebrauch notwendige Kopiervorgänge angewiesen.<sup>77</sup> Art. 12 Abs. 2 URG äussert sich nicht konkret zum sachlichen Umfang (bspw. wie viele virtuelle Kopien gleichzeitig im System genutzt werden dürfen).<sup>78</sup> Massgebend für dessen Bestimmung ist in erster Linie der Softwarelizenzvertrag zwischen Rechtsinhaber und Ersterwerber.<sup>79</sup> Zu berücksichtigende Faktoren sind der Vertragsinhalt und die Vertragsumstände, die Charakteristika und Beschreibung der Software wie auch Branchenusancen und die in Art. 17 URV aufgezählten Nutzungshandlungen, die als Mindestinhalt dienen.<sup>80</sup> Fehlen spezifische Regelungen im Softwarevertrag, sind die Präzisierungen in Art. 17 Abs. 1 lit. a URV zu beachten, die die Umsetzung der Softwareschutz-RL<sup>81</sup> bezwecken und das Laden, Anzeigen, Ablaufenlassen, Übertragen oder Speichern sowie die Herstellung dafür erforderlicher Kopien als bestimmungsgemässe Verwendung eines Programms bezeichnen.<sup>82</sup> Dazu gehört insbesondere die zum Gebrauch notwendige Vervielfältigung im Arbeitsspeicher.<sup>83</sup> Vertragliche Nutzungsbeschränkungen müssen mit urheberrechtlicher Wirkung ausgestattet sein, um das Gebrauchsrecht zu definieren und damit auch für alle nachfolgenden Erwerber bindend zu sein.<sup>84</sup> Dem gesetzlichen Gebrauchs- und Weiterveräusserungsrecht wird zudem ein zwingender Kerngehalt zugeschrieben, welcher vertraglich nicht einschränkbar ist.<sup>85</sup> Neben Vervielfältigungen soll das Gebrauchsrecht, obwohl dies in Art. 17 Abs. 1 lit. a URV nicht explizit erwähnt wird, ebenfalls ein Recht auf Bearbeitung der Software im Rahmen des bestimmungsmässigen Gebrauchs vermitteln, zumal auch Art. 5 Abs. 1 i.V.m. Art. 4 lit. b der Softwareschutz-RL ein solches vermittelt.<sup>86</sup> Das setzt voraus, dass die Änderungen zur bestimmungsgemässen Verwendung des Programms notwendig sind, was nicht der Fall ist, wenn der Rechtsinhaber die Fehlerbehebung zu angemessenen Konditionen anbietet.<sup>87</sup>

---

<sup>75</sup> Zum Ganzen RAUBER (Fn. 41), Use Restrictions, S. 148 ff.; RAUBER (Fn. 49), Computersoftware, S. 171 ff.; kritisch auch THOMAS SEMADENI, Erschöpfungsgrundsatz im Urheberrecht, Diss., Bern 2004, S. 64 ff.

<sup>76</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 154.; vgl. STRAUB (Fn. 44), Rn. 171.

<sup>77</sup> Vgl. STRAUB (Fn. 44), Rn. 151.

<sup>78</sup> STRAUB (Fn. 44), Rn. 171.

<sup>79</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 154; vgl. BERNARD HEUSLER/ROLAND MATHYS, IT-Vertragsrecht, Zürich 2004, S. 156; RAUBER (Fn. 41), Use Restrictions, S. 153.

<sup>80</sup> Vgl. STRAUB/RÜFENACHT (Fn. 44), Rn. 20; STRAUB (Fn. 44), Rn. 172 ff.

<sup>81</sup> Vgl. Art. 4 und Art. 5 CRRL.

<sup>82</sup> Vgl. RAUBER (Fn. 41), Use Restrictions, S. 154; FRÖHLICH-BLEULER (Fn. 59), Rn. 157.

<sup>83</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 158.

<sup>84</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 155; siehe zur urheber- und vertragsrechtlichen Wechselwirkung Kap. 5.1.

<sup>85</sup> STRAUB/RÜFENACHT (Fn. 44), Rn. 21.5; STRAUB (Fn. 44), Rn. 165 und 183; RAUBER (Fn. 49), Computersoftware, S. 166 f.; DERS. (Fn. 41), Use Restrictions, S. 153 f. und 158; siehe zum zwingenden Kern Kap. 4.1.2.1.

<sup>86</sup> Vgl. STRAUB (Fn. 44), Rn. 170 und 193; RAUBER (Fn. 49), Computersoftware, S. 182; RAUBER (Fn. 41), Use Restrictions, S. 154 f.

<sup>87</sup> Vgl. STRAUB (Fn. 44), Rn. 196; RAUBER (Fn. 49), Computersoftware, S. 183.

### 3.3. Softwareeinsatz in containerisierten Umgebungen

[20] Art. 12 Abs. 2 URG kommt vor allem dann zur Anwendung, wenn kein direkter Vertrag zwischen Rechtsinhaber und Programmierer vorliegt (bspw. bei nicht rechtsgültig akzeptierten AGB) oder einer vorliegt, dieser sich aber aus anderen Gründen nicht als wirksam herausstellt.<sup>88</sup> Ausserdem kommt es auch oft vor, dass in den wirksamen Verträgen konkrete Regelungen zur Virtualisierungsproblematik fehlen, was ebenfalls zu einer Anwendung von besagtem Artikel führt. Es ist konkret zu klären, ob und in welchem Umfang der Softwareerwerber bei fehlender oder fehlender zusätzlicher Vereinbarung zum Einsatz der Software in containerisierten bzw. virtualisierten Umgebungen berechtigt ist. Braucht es bei den Nutzungen im Zusammenhang mit dem typischen Betrieb von containerisierter Software eine Erlaubnis des Rechtsinhabers, womit eine gesonderte Lizenzpflicht bestünde oder kann sich der Erwerber auf die gesetzliche Gebrauchsbefugnis berufen? Wie weit der bestimmungsgemässe Gebrauch geht, ist immer anhand des konkreten Falls zu ermitteln.<sup>89</sup>

#### 3.3.1. Ablaufenlassen

[21] Zuerst zu beurteilen ist das blosses Ablaufenlassen einer Anwendung in einem Container. Container stellen eine stabile Laufzeitumgebung für eine Software bereit, da wie gesagt das Programm samt benötigter Abhängigkeiten zur Laufzeit integriert werden kann. Die reine Installation einer Anwendung in einem Container oder in einer virtuellen Maschine unterscheidet sich nicht im Wesentlichen von der direkten Installation auf einem physischen Rechner. Physikalische und virtuelle Server sind aus wirtschaftlich funktionaler Sicht äquivalent.<sup>90</sup> Wenn im Lizenzierungsumfeld von einer «Instanz» gesprochen wird, ist dieser Begriff gleichzustellen mit der virtuellen Maschine selbst (bzw. deren Betriebssystem) oder mit dem Container.<sup>91</sup> Die Ausführung der Anwendung in Docker und seinen Pendanten deckt sich mit dem in Art. 17 Abs. 1 lit. a URV definierten Ablaufenlassen eines Programms. Vorliegend geht es um den Betrieb *einer* einzigen Programmkopie im Container (analog in einer virtuellen Maschine), bei welchem es zu einer urheberrechtlichen Vervielfältigung im Arbeitsspeicher kommt, die aber für den Gebrauch notwendig ist und somit von Art. 12 Abs. 2 URG umfasst wird.<sup>92</sup> Es fehlt an einer wirtschaftlich eigenständigen Bedeutung der Nutzung in virtualisierten Systemen, die eine gesonderte Lizenzierung dafür rechtfertigen würde.<sup>93</sup> Es wird schliesslich nach wie vor nur eine Kopie genutzt und darüber hinaus werden auch keine neuen Nutzerkreise erschlossen. Allein der Umstand, dass auch nicht speziell dafür konzipierte Programme in einer virtuellen Umgebung betrieben werden, deutet auf eine fehlende eigenständige Nutzungsart hin.<sup>94</sup> STRAUB und auch FRÖHLICH-BLEULER gehen davon aus, dass im Zweifel von einer Erlaubnis dazu auszugehen ist, sofern keine Beschränkungen auf bestimmte Technologien vorliegen.<sup>95</sup> Ob das allgemeine Ablaufenlassen einer Kopie in

---

<sup>88</sup> Vgl. STRAUB (Fn. 44), Rn. 152.

<sup>89</sup> Vgl. STRAUB (Fn. 44), Rn. 163.

<sup>90</sup> Vgl. JOCHEN MARLY, Praxishandbuch Softwarerecht, 7. Aufl., München 2018, Rn. 1674.

<sup>91</sup> Vgl. Server-Virtualisierung, <https://www.itwissen.info/Servervirtualisierung-server-virtualization.html>.

<sup>92</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 1706 und 1711; vgl. GRÜTZMACHER (Fn. 11), Virtualisierung, S. 173.

<sup>93</sup> Vgl. GRÜTZMACHER (Fn. 11), Virtualisierung, S. 174.

<sup>94</sup> Vgl. zum Ganzen FRÖHLICH-BLEULER (Fn. 59), Rn. 1712.

<sup>95</sup> STRAUB (Fn. 44), Rn. 300; vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 1712.

virtualisierten Umgebungen dem Kerngehalt des Gebrauchsrechts zuzuordnen ist, ist in der Lehre umstritten (dazu Kap. 5.3.1.).<sup>96</sup>

[22] Oftmals stellen Anbieter ihre Software ihrerseits gleichzeitig auch als Container bereit in Form einer Downloadversion auf Plattformen wie z.B. Docker Hub. Docker Hub ist ein Online-dienst, welcher eine Registry für Docker-Images und Repositories zur Verfügung stellt. Nutzer können dort selbst erstellte Images hochladen, öffentlich und auch nicht-öffentlich, d.h. nur für einen bestimmten Adressatenkreis.<sup>97</sup> Die Bereitstellung deutet darauf hin, dass die Containerisierung einer Anwendung bereits durch die Beschreibung und die Charakteristika der Software vorgesehen und somit als bestimmungsgemässe Verwendung zu qualifizieren ist. Dabei handelt es sich um weitere Vertragsumstände, die auf den bestimmungsgemässen Gebrauch hinweisen. Das blosses Ablaufenlassen einer einzigen Programmkopie im Container und allgemein in virtualisierten Umgebungen braucht also keine explizite Erlaubnis und ist nicht gesondert lizenzpflichtig.

### 3.3.2. Nutzung in mehreren Instanzen

[23] In der Praxis kommt es selten vor, dass nur eine einzige Containerinstanz mit der betreffenden Programmkopie im Einsatz ist, gehören doch gerade die Skalierung und die Hochverfügbarkeit zu den grössten Vorteilen der Container-Technologie und der Virtualisierung. Meist betreibt man ein Set-Up mit zahlreichen Containern und Hosts, ein sog. «Container-Cluster». Skalierbarkeit und Redundanz sind im Bereich der Container nur im Zusammenspiel mit Orchestrierungswerkzeugen wie Kubernetes, Docker Swarm etc. erreichbar. Bei Problemen mit einem Server oder Container übernimmt ein anderer Node (virtuelle oder physische Ressource für Container) blitzschnell die notleidenden Workloads.<sup>98</sup> Container sind vergleichsweise noch skalierbarer als virtuelle Maschinen. Es ist einfacher, sie zu installieren und zu starten.<sup>99</sup> Bei Kubernetes bspw. sollen immer hinreichend viele Container für den durchgängigen und reibungslosen Betrieb zur Verfügung stehen, dies auch auf verschiedenen Servern, um die Ausfallsicherheit zu gewährleisten. Es ist möglich und sogar oft der Fall, dass Container in Private oder Public Clouds betrieben werden.<sup>100</sup> Auf die Einzelheiten der Clouds wird aufgrund des ausufernden Themenbereichs nicht weiter eingegangen. Betreffend die urheberrechtlich relevanten Vervielfältigungen vertritt die h.L. die Ansicht, dass bereits eine Parallelinstallation urheberrechtlich unzulässig sei.<sup>101</sup> Dafür spricht die enge Umschreibung des gesetzlichen Gebrauchsrechts. Folglich müsste eine parallele Installation, wie sie in redundanten Systemen üblich ist, ausdrücklich vereinbart werden, auch wenn keine zeitgleiche Nutzung erfolgt.<sup>102</sup> FRÖHLICH-BLEULER ist der Meinung, dass wenn die Mehrfachinstallation des Programms für den bestimmungsgemässen Gebrauch aus technischen

---

<sup>96</sup> Sich dafür aussprechend GRÜTZMACHER (Fn. 11), Virtualisierung, S. 177; a.A. tendenziell FRANK A. KOCH, Computer-Vertragsrecht, 7. Aufl., Freiburg im Breisgau 2009, S. 917.

<sup>97</sup> Docker, [https://de.wikipedia.org/wiki/Docker\\_\(Software\)](https://de.wikipedia.org/wiki/Docker_(Software)).

<sup>98</sup> Vgl. zum Ganzen SANDER VAN VUGT, Welche Containertechnologie braucht mein Rechenzentrum?, <https://tinyurl.com/yyskn75g>.

<sup>99</sup> SHELDON (Fn. 20), Applikationscontainer-Technologie.

<sup>100</sup> Vgl. zum Ganzen VAN VUGT (Fn. 98), Welche Containertechnologie braucht mein Rechenzentrum?

<sup>101</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 1704; HEUSLER/MATHYS (Fn. 79), S. 159; MARKUS WANG, Software-Überlassungsverträge: Nutzungsbeschränkungen, Leistungsstörungen, Gewährleistung und Haftung, in: Hans Rudolf Trüb (Hrsg.): Softwareverträge, Zürich 2004, S. 105–128, S. 109.

<sup>102</sup> HEUSLER/MATHYS (Fn. 79), S. 159.

Gründen erfolgt, wie es bei der Spiegelung des EDV-Systems zur Erhöhung der Verfügbarkeit zutrifft, dies zum gesetzlich erlaubten Gebrauch zähle.<sup>103</sup> Eine andere Lehrmeinung geht davon aus, dass die mehrfache Installation auf dem Computer allein noch keine Parallelnutzung darstelle, sondern erst das Ausführen des Programms wirtschaftlich relevant sei. Gemäss dem Prinzip der «sequentiellen Singulärnutzung» dürfe die Programmkopie auf unterschiedlichen Maschinen im Einsatz sein, jedoch nur sequentiell, d.h. nicht gleichzeitig.<sup>104</sup> Einigkeit besteht darin, dass beim *parallelen* Einsatz mehrerer Programmkopien auf einer oder unterschiedlichen Maschinen die Zustimmung des Rechtsinhabers erforderlich ist.<sup>105</sup> Die Nutzung in mehreren Instanzen führt dazu, dass die Software zumindest teilweise mehrfach im Arbeitsspeicher vorhanden ist.<sup>106</sup> Nur der Rechtsinhaber ist dazu berechtigt, die zeitgleiche Mehrfachnutzung des Computerprogramms in Containern wie auch in virtuellen Maschinen zu erlauben und vorzugeben, bis zu welcher Anzahl von Instanzen seine Software lizenziert werden soll, wobei bei Fehlen einer weitergehenden Erlaubnis die Instanzenanzahl auf diese vorgegebene Anzahl beschränkt ist.<sup>107</sup> Das mehrfache Betreiben einer Software in einer virtuellen Umgebung gehört ohne weitere Vereinbarung nicht mehr zum bestimmungsgemässen Gebrauch nach Art. 12 Abs. 2 URG. Dem Rechtsinhaber könnten sonst potenziell Einnahmemöglichkeiten verloren gehen.<sup>108</sup> Berücksichtigt man das Prinzip der sequentiellen Singulärnutzung, könnten mehrere Container mit der gleichen Konfiguration (inkl. Programmkopie) parallel erstellt werden, aber nur einer dürfte ohne weitere Vereinbarung tatsächlich in Betrieb sein. In der Praxis spielt es eine untergeordnete Rolle, welcher Ansicht gefolgt wird, da bei Container-Umgebungen die Gefahr einer Übernutzung erheblich ist, weshalb die mehrfache Installation oft vertraglich eingeschränkt oder zusätzlich vergütet wird. Die genannten Orchestrierungslösungen können helfen, unerlaubte Mehrfachnutzungen ausfindig zu machen und sie zu verhindern, denn sie koordinieren den Containerbetrieb. Parallele Nutzungen erfolgen in der Praxis oft unabsichtlich.

### 3.3.3. Sicherungskopien

[24] Es ist fraglich, wie es sich mit virtuellen Umgebungen verhält, die als Sicherungskopie gespeichert werden, um sie bei Nutzungsverlusten abzurufen. Art. 24 Abs. 2 URG statuiert grundsätzlich die zwingende Erlaubnis zur Herstellung einer Sicherungskopie des Computerprogramms. Gemäss h.L. handelt es sich hier um einen Anwendungsfall des gesetzlichen Gebrauchsrechts, welcher zwar bei Software-Überlassungen auf Dauer gelten soll, nicht aber bei echten Lizenzen.<sup>109</sup> Dem Anwender sollte das Recht zustehen, seine Sicherheitsarchitektur selbst zu definieren und zu implementieren.<sup>110</sup> Das kann je nachdem bedeuten, dass auch mehrere Sicherungskopien hergestellt werden dürfen, was entweder durch eine weite Auslegung von Art. 24

---

<sup>103</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 1704.

<sup>104</sup> Zum Ganzen STRAUB (Fn. 44), Rn. 178, 189 und 192

<sup>105</sup> Vgl. STRAUB (Fn. 44), Rn. 178 und 184.

<sup>106</sup> GRÜTZMACHER (Fn. 11), Virtualisierung, S. 173.

<sup>107</sup> Vgl. GRÜTZMACHER (Fn. 11), Virtualisierung, S. 173 f.

<sup>108</sup> Ebd.

<sup>109</sup> Vgl. STRAUB (Fn. 44), Rn. 232; HEUSLER/MATHYS (Fn. 79), S. 162; NEFF/ARN (Fn. 48), S. 307 f.; a.A. FRÖHLICH-BLEULER (Fn. 59), Rn. 105.

<sup>110</sup> RAUBER (Fn. 49), Computersoftware, S. 233.

Abs. 2 URG oder durch den bestimmungsgemässen Gebrauch gedeckt ist.<sup>111</sup> Der Anwender muss jedoch aufpassen, dass die überlassene Software durch die Sicherheitsarchitektur nicht übernutzt wird.<sup>112</sup> Zu berücksichtigen sind Partizipations-, Kontroll- und Geheimhaltungsinteressen des Urhebers.<sup>113</sup> Erlaubt ist das Sichern der gesamten Installation einer virtuellen Maschine als Backup, bei welchem nicht zwischen Daten und Software unterschieden wird, d.h. Softwarekomponenten automatisch mitgesichert werden.<sup>114</sup> Bei echten Backups, die erst später zur Nutzung zurückgespielt werden, stellt das kein Problem dar, denn die Gefahr einer parallelen Nutzung besteht nicht.<sup>115</sup> Derartige Backups sind vorwiegend bei virtuellen Maschinen relevant, deren gesamte Konfiguration gesichert wird. Container können zwar auch Sicherungszwecken dienen, doch wäre hier eher von einer *warm* oder *hot-Standby*-Situation auszugehen. Dabei installiert der Anwender die Sicherungskopie je Software bereits auf einem Ausweich-Rechner (oder in diesem Fall Container) und fährt sie im Notfall hoch (*warmes Backup*) oder behält sie dauernd in paralleler, synchroner Bereitschafts-Verfügung bereit (*heisses Backup*).<sup>116</sup> Sicherungskopien auf solchen Standby-Systemen sind nicht mehr vom gesetzlichen Privileg gedeckt, da es in diesem Fall nicht bloss um die Sicherung der Daten geht, sondern auch um eine Leistungsverbesserung.<sup>117</sup> Der Hersteller wird auch kaum dazu fähig sein, zu prüfen, ob die Software wirklich als Backup oder eher parallel genutzt wird. Dasselbe gilt für Kopien, die im Netzwerk gespeichert werden und jederzeit zum Gebrauch bereitstehen, denn sie bringen eine hohe Gefahr der operativen Nutzung mit sich.<sup>118</sup> Eine Version liegt auch «heiss» vor, wenn sie virtuell an einem anderen Ort vorfindbar ist, beim *Backup* in der Cloud etwa.<sup>119</sup> Für solche vom Gebrauchsrecht nicht gedeckte Situationen wären gesonderte Lizenzen erforderlich. Handelt es sich jedoch um eine technische Modalität und müssen die Anwendungen notwendigerweise ständig verfügbar sein (bspw. bei einer Handelssoftware bei Banken) kann das aktive, permanente Verhalten im Rahmen von Hochverfügbarkeits-Clustern durchaus als bestimmungsgemäss gelten.<sup>120</sup>

### 3.3.4. Modifikationen

[25] Eine weitere relevante Nutzungshandlung im Zusammenhang mit der Container-Virtualisierung ist die Modifikation bzw. Bearbeitung von Software. Je nach Technologie beinhaltet ein Container mehrere Layer. Dateien auf einem Layer lassen sich ausblenden, ergänzen oder überlagern, was als Bearbeitung im lizenzrechtlichen Kontext zu verstehen ist.<sup>121</sup> Um die Grösse und den Ressourcenbedarf eines Containers noch mehr zu reduzieren, werden nicht betriebsrelevante Infor-

---

<sup>111</sup> Für eine weite Auslegung von Art. 24 Abs. 2 URG siehe RAUBER (Fn. 49), *Computersoftware*, S. 191 f. und 233; DERS. (Fn. 41), *Use Restrictions*, S. 144 f.; für die Zuordnung zum bestimmungsgemässen Gebrauch vgl. STRAUB (Fn. 44), Rn. 237; FRÖHLICH-BLEULER (Fn. 59), Rn. 106.

<sup>112</sup> Vgl. RAUBER (Fn. 49), *Computersoftware*, S. 233.

<sup>113</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 106.

<sup>114</sup> Vgl. STRAUB (Fn. 44), Rn. 237.

<sup>115</sup> Vgl. GRÜTZMACHER (Fn. 11), *Virtualisierung*, S. 175.

<sup>116</sup> Vgl. JOCHEN SCHNEIDER, *Handbuch EDV-Recht*, 5. Aufl., Köln 2017 (zit. SCHNEIDER, Teil ... Rn. ...), Teil M Rn. 449.

<sup>117</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 108; zum deutschen Recht (analog) auch SCHNEIDER (Fn. 116), Teil M Rn. 449.

<sup>118</sup> GRÜTZMACHER (Fn. 11), *Virtualisierung*, S. 174 f.

<sup>119</sup> SCHNEIDER (Fn. 116), Teil M Rn. 451.

<sup>120</sup> Vgl. MALTE GRÜTZMACHER, *Lizenzgestaltung für neue Nutzungsformen im Lichte von § 69d UrhG (Teil 2)*, CR 2011, S. 697–705 (zit. GRÜTZMACHER, *Nutzungsformen 2*), S. 702.

<sup>121</sup> KARSTEN KLEIN/THOMAS SCHULTE, *Container-Technologie – Segen und Fluch*, <https://tinyurl.com/y5uej2ff>.



mationen der containerisierten Software gelegentlich bewusst entfernt, z.B. Logos, Dokumentation oder gar Programmteile wie nicht verwendete Treiber/Module. Dieses «Entschlacken», welches in der Praxis vorkommt, gilt auch als Modifikation und stellt eine ggf. unerlaubte Nutzungshandlung dar. Obwohl in Art. 17 Abs. 1 lit. a URV nicht erwähnt, gehört auch ein Bearbeitungs- und Fehlerbehebungsrecht zum bestimmungsgemässen Gebrauch, insofern dazu Notwendigkeit vorliegt.<sup>122</sup> Dazu gehören bspw. die Behebung blockierender Fehler oder die Anpassung von Installationsskripten.<sup>123</sup> Somit bleibt dem Softwareerwerber mangels anderer vertraglicher Absprache lediglich das Recht, die Software bei entsprechender Notwendigkeit für die Installation in einem Container zu «modifizieren». Nicht vom Gebrauchsrecht umfasst sind Anpassungen des Programms an individuelle Bedürfnisse, Programmerweiterungen- oder verbesserungen.<sup>124</sup> Auf diese Punkte ist bei Modifikationen im Kontext des Containereinsatzes acht zu geben.

## 4. Vertragliche Nutzungsbestimmungen

[26] Nachdem soeben der Umfang des gesetzlichen Gebrauchsrechts erläutert und die Situation bei fehlenden vertraglichen Regelungen bezüglich des Softwareeinsatzes in containerisierten bzw. generell virtualisierten Umgebungen untersucht worden ist, soll es nun um die Situation bei Vorliegen vertraglicher Bestimmungen gehen. Vorab sind einige Ausführungen zur strittigen Qualifikation des Softwareüberlassungsvertrags und dessen rechtlichen Rahmen vorzunehmen. Danach werden Lizenzformen und spezifische Lizenzklauseln, die sich auf den Softwareeinsatz in (container-)virtualisierten Systeme beziehen anhand der Regelwerke bestimmter Softwareanbieter thematisiert. Durch diese beispielhaften Vereinbarungen sollen auch deren Probleme in der IT-Praxis, insb. im Lizenzmanagement verdeutlicht werden.

### 4.1. Softwareüberlassungsvertrag

#### 4.1.1. Qualifikation

[27] Das Gebiet der Softwareüberlassung zeichnet sich durch eine Vielfalt möglicher Rechtsgeschäfte aus.<sup>125</sup> Dem Vertragspartner wird hierbei durch den Berechtigten das Recht zur Nutzung der Software eingeräumt, wobei die Qualifikation derartiger Verträge umstritten ist. Umgangssprachlich ist meist von einem «Softwarelizenzvertrag» die Rede, dessen Ausgestaltung unterschiedlich sein kann, was Auswirkungen auf die rechtliche Qualifikation hat.<sup>126</sup> Davon zu unterscheiden ist der Übergang absoluter Rechte am Immaterialgut, welcher beim Softwarelizenzvertrag als rein schuldrechtliche Konzeption nicht zutrifft.<sup>127</sup> Gemäss h.L. und Rechtsprechung handelt es sich beim (Software-)Lizenzvertrag um einen Innominatvertrag, welcher je nach Ausgestaltung miet- und pachtrechtliche, teilweise aber auch teilweise kauf- oder gesellschafts-

---

<sup>122</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 159; RAUBER (Fn. 49), Computersoftware, S. 182; STRAUB (Fn. 44), Rn. 193.

<sup>123</sup> STRAUB (Fn. 44), Rn. 193.

<sup>124</sup> Vgl. RAUBER (Fn. 41), Use Restrictions, S. 156; FRÖHLICH-BLEULER (Fn. 59), Rn. 159.

<sup>125</sup> MORSCHER/DORIGO (Fn. 72), S. 21.

<sup>126</sup> Vgl. GELLIS (Fn. 73), S. 439.

<sup>127</sup> Vgl. RETO M. HILTY, Urheberrecht, 2. Aufl., Bern 2020, Rn. 693.

rechtliche Elemente beinhalten kann.<sup>128</sup> Der Eingrenzung des Themas halber soll Gegenstand folgender Erläuterungen vor allem Standardsoftware sein, die auf Dauer überlassen wird. Beim Vertragsgegenstand handelt es sich demnach um Programme, die nicht speziell für den Anwender hergestellt werden.<sup>129</sup> Die Berücksichtigung der Überlassung von Individualsoftware, welche spezifisch auf die Bedürfnisse des Erwerbers zugeschnitten wird, würde den Rahmen dieser Arbeit sprengen, da es sich in diesem Falle i.d.R. um einen Werkvertrag handelt.<sup>130</sup> Eine weitere zentrale Weichenstellung liegt darin, dass zeitlich begrenzte Nutzungsrechte im Sinne der – bereits in Kap. 3.2.1. behandelten – «echten» Lizenz (vorherrschend bei Individualsoftware) im weiteren Verlauf ebenfalls nicht Thema sein werden. Bei solchen Konstellationen wird meist ein periodisches Entgelt entrichtet und das Element der Gebrauchsüberlassung überwiegt.<sup>131</sup> Hierbei wären ggf. die Regeln der Miete oder diejenigen der Pacht anzuwenden.<sup>132</sup> Nicht jeder Vertrag mit dem Titel «Softwarelizenz» bezieht sich aber auf die Gewährung einer «echten» Lizenz.<sup>133</sup> In der Lehre herrschen unterschiedliche Meinungen über die Rechtsnatur des vorliegend relevanten Vertrags zur Überlassung von Standardsoftware auf Dauer (auch: definitive Softwareüberlassung), bei welchem die Nutzungsbefugnis zeitlich unbeschränkt übertragen wird und der Vertragsgegenstand meist Standardsoftware darstellt.<sup>134</sup> Einige Autoren möchten darauf direkt oder zumindest analog Kaufvertragsrecht anwenden<sup>135</sup>, andere beabsichtigen Kaufrecht anzuwenden, da das Rechtsgeschäft über das Werkexemplar einen Kaufvertrag darstelle<sup>136</sup> und wiederum eine dritte Meinung besagt, dass es sich um einen Lizenzvertrag<sup>137</sup> bzw. Innominatkontrakt handle, auf welchen die allgemeinen Bestimmungen des OR anzuwenden sind.<sup>138</sup> Nach der Ansicht der Autoren STRAUB und FRÖHLICH-BLEULER, welcher ich mich anschliesse, ist auf das Vertragsverhältnis Kaufrecht analog anzuwenden, wenn es sich beim Vertragsgegenstand um Standardsoftware handelt, die Nutzungsbefugnis auf Dauer erteilt wird und der Anwender für die Nutzung nur eine Einmal-Lizenzgebühr schuldet.<sup>139</sup> STRAUB braucht bei solchen Situationen den – bereits im Zusammenhang mit dem Anwendungsbereich des Gebrauchsrechts behandelten – Begriff des «unechten Lizenzvertrags».<sup>140</sup> Aufgrund dessen wird weiterhin von «Lizenzklauseln» oder «Lizenzbedingungen» die Rede sein, obwohl es sich aufgrund des fehlenden Dauerschuldverhältnisses eher um einen atypischen Lizenzvertrag handelt.<sup>141</sup> Es ist letztendlich festzuhalten, dass bei

---

<sup>128</sup> BGE 92 II 299, Pra 1967, 173; GELLIS (Fn. 73), S. 440; vgl. RETO M. HILTY, Lizenzvertragsrecht, Habil., Bern 2001, S. 159 ff., DERS. (Fn. 127), Rn. 630.

<sup>129</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 1639.

<sup>130</sup> Vgl. THOMANN (Fn. 48), S. 36.

<sup>131</sup> GELLIS (Fn. 73), S. 440; vgl. NEFF/ARN (Fn. 48), S. 272; SEMADENI (Fn. 75), S. 50.

<sup>132</sup> MORSCHER/DORIGO (Fn. 72), S. 30; FRÖHLICH-BLEULER (Fn. 59), Rn. 2395; WANG (Fn. 101), S. 117.

<sup>133</sup> MORSCHER/DORIGO (Fn. 72), S. 55.

<sup>134</sup> GELLIS (Fn. 73), S. 440 f.

<sup>135</sup> Etwa HEUSLER/MATHYS (Fn. 79), S. 43; MORSCHER/DORIGO (Fn. 72), S. 29; WANG (Fn. 101), S. 116 f.

<sup>136</sup> THOMANN (Fn. 48), S. 35; NEFF/ARN (Fn. 48), S. 272 (bei Massensoftware).

<sup>137</sup> HILTY (Fn. 127), Rn. 708, geht bei Einmallyzenzgebühren immer noch von einem Dauerschuldverhältnis und folglich von einer Lizenzkonstruktion aus.

<sup>138</sup> ROGER STAUB, IP-Klauseln/Datenschutz in IT-Verträgen, in: Florian S. Jörg/Olivier Arter (Hrsg.): Internet-Recht und IT-Verträge, 2. Aufl., Bern 2009, S. 179–214, S. 191.

<sup>139</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 1648 ff.; STRAUB (Fn. 44), Rn. 153 ff.

<sup>140</sup> STRAUB (Fn. 44), Rn. 153.

<sup>141</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 1653; a.A. STAUB (Fn. 138), S. 191.

dieser Konstellation nach Rechtsprechung und h.L. primär Kaufvertragsrecht angewendet wird, ungeachtet deren Qualifikation.<sup>142</sup>

#### 4.1.2. Rechtlicher Rahmen

[28] Softwarevertragliche Nutzungsvereinbarungen sind nur bei gültiger Vereinbarung wirksam, weshalb der rechtliche Rahmen zu berücksichtigen ist. Wenn AGB Vertragsbestandteil sein sollen, stellen sich bereits Konsensfragen. Weiter können der bisher nur angetönte zwingende Kern von Art. 12 Abs. 2 URG sowie kartell- und lauterkeitsrechtliche Schranken zu einer Unwirksamkeit oder Nichtigkeit führen. Auf die übrigen urheberrechtlichen Schranken bei Software wurde bereits eingegangen (Kap. 3.2.). Im weiteren Verlauf dieser Arbeit werden die kartellrechtlichen Bestimmungen nur am Rande behandelt.

##### 4.1.2.1. Kerngehalt des Gebrauchsrechts

[29] Die h.L. in der Schweiz ordnet dem Gebrauchsrecht einen zwingenden Kern gemäss Art. 12 Abs. 2 URG i.V.m. Art. 17 Abs. 1 lit. a URV zu, welcher in Abweichung vom Grundsatz der Vertragsautonomie vertraglich nicht eingeschränkt werden kann.<sup>143</sup> Zwar hält Art. 12 Abs. 2 URG diesen nicht explizit fest, doch bei der Gesetzesauslegung ist die Softwareschutz-RL mitzuberücksichtigen, und diese enthält einen solchen Kerngehalt.<sup>144</sup> Dieser beinhaltet die mit der Installation verbundenen notwendigen Vervielfältigungen und Speicherungen sowie den operativen Lauf des Programms. Dasselbe gilt für Fehlerbehebungen, die die bestimmungsgemässe Nutzung des Programms sicherstellen, insofern der Rechtsinhaber selbst keine solche anbietet.<sup>145</sup> Der zwingende Kern von Art. 12 Abs. 2 URG stellt auf gebrauchtsrechtlicher Ebene den minimalen Sockel dar, von welchem die Parteien auch auf vertraglicher Ebene nicht abweichen dürfen.<sup>146</sup> Vertragliche Einschränkungen, die gegen den zwingenden Kern verstossen, gelten als unwirksam.<sup>147</sup>

##### 4.1.2.2. Gültige Vereinbarung allgemeiner Lizenzbedingungen

[30] Bei Vorliegen allgemeiner Lizenzbedingungen (oder: AGB), muss stets geprüft werden, ob diese überhaupt Vertragsbestandteil geworden sind. Vorausgesetzt ist ein Konsens zwischen den Vertragsparteien betreffend den Nutzungseinschränkungen.<sup>148</sup> Um ein direktes Vertragsverhältnis zum Endabnehmer, welcher die Software über Zwischenhändler erwirbt, herzustellen, bedienen sich Hersteller oft vorformulierter Allgemeiner Vertragsbedingungen (*End User License Agreements*, EULA). Das Ziel ist die Einschränkung des Umfangs des Gebrauchsrechts und der

---

<sup>142</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 1653; GELLIS (Fn. 73), S. 441; HEUSLER/MATHYS (Fn. 79), S. 43; BGE 124 III 456, E. 4bb; BGer. 4A\_446/2015 vom 3. März 2016 (BGer. wendet direkt Kaufrecht an).

<sup>143</sup> STRAUB (Fn. 44), Rn. 165; RAUBER (Fn. 41), Use Restrictions, S. 153 und 158.

<sup>144</sup> Vgl. RAUBER (Fn. 41), Use Restrictions, S. 153.

<sup>145</sup> Vgl. RAUBER (Fn. 41), Use Restrictions, S. 153; DERS. (Fn. 49), Computersoftware, S. 184; FRÖHLICH-BLEULER (Fn. 59), Rn. 164.

<sup>146</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 166 f.

<sup>147</sup> Vgl. STRAUB (Fn. 44), Rn. 165.

<sup>148</sup> Vgl. RAUBER (Fn. 41), Use Restrictions, S. 157.

Haftungsansprüche.<sup>149</sup> Entsprechende Bedingungen müssen aber, um vom Konsens umfasst und damit Vertragsbestandteil zu werden, vor dem Vertragsschluss zur Kenntnis genommen werden können. Dies trifft z.B. bei den sog. *Shrink Wrap Licenses* nicht zu, denn die Nutzungsbedingungen können erst nach Softwarekauf durch das Öffnen der Schutzhülle (mittels Hinweis auf der Verpackung) zur Kenntnis genommen werden.<sup>150</sup> Konstellationen, bei welchen die EULA vor dem Eigentumserwerb (bspw. vor dem Download) akzeptiert werden, führen zu einer gültigen Vereinbarung.<sup>151</sup> Containerspezifisch ist hierbei anzufügen, dass in der Praxis den EULA bereits durch integrierte Parameter im Ausführungsbefehl, den Container zu starten, zugestimmt wird. Inwieweit Vertragsbedingungen so tatsächlich gültig vereinbart werden ist fraglich und wird in casu offengelassen. Sind die vorformulierten Nutzungsbedingungen einmal gültig vereinbart worden, kann ggf. immer noch ein Anwendungsfall der Ungewöhnlichkeitsregel vorliegen. Demnach ist ein Erwerber nach Treu und Glauben nicht an global übernommene vorformulierte Vertragsbedingungen gebunden, bei welchen er zu seinen Ungunsten auf das verzichtet, worauf er sich vernünftigerweise verlassen darf.<sup>152</sup> Davon ist bei Klauseln mit ungewöhnlichem Inhalt oder intransparenter Form auszugehen. Derartige Bestimmungen sind als nichtig zu qualifizieren.<sup>153</sup> Unter Umständen liegen in diesem Zusammenhang auch lauterkeitsrechtliche Verletzungen vor, dies einerseits bei irreführenden Inhalten in AGB nach der Generalklausel in Art. 2 UWG (praktisch von geringer Bedeutung) und bei missbräuchlichen Konditionen in Konsumentenverträgen nach Art. 8 UWG, welcher auch eine Inhaltskontrolle beinhaltet.<sup>154</sup> Im Gegensatz zur Schweiz, bildete sich in Deutschland eine detaillierte Lehre und Rechtsprechung zu AGB-rechtlich unwirksamen Nutzungsbeschränkungen und anderen Bestimmungen in Softwareüberlassungsverträgen. Hierzulande muss der Hinweis auf das Korrektiv bei ungewöhnlichen Nutzungsbeschränkungen genügen.<sup>155</sup>

## 4.2. Lizenzbestimmungen im Bereich virtualisierter Umgebungen

[31] Das Gebiet der Container-Virtualisierung ist verhältnismässig jung und in der juristischen Lehre noch wenig behandelt. In der Praxis sind Lizenzmodelle häufig obsolet und noch auf physische Rechner ausgerichtet, m.a.W. noch nicht an dynamische virtualisierte Umgebungen angepasst.<sup>156</sup> Diese Problematik trifft bei Containern wie auch bei virtuellen Maschinen zu. Je nach Hersteller kann die Ausgestaltung der Lizenzen und v.a. deren Metrik, also die dazugehörigen Vergütungssysteme, stark variieren. Eine Systematisierung ist angesichts der stetigen technischen Entwicklung komplex.<sup>157</sup> In der Praxis erweisen sich Lizenzmodelle vielfach als extrem umfangreich und geradezu unübersichtlich.

---

<sup>149</sup> Vgl. zum Ganzen STRAUB (Fn. 44), Rn. 263.

<sup>150</sup> Vgl. zum Ganzen RAUBER (Fn. 41), Use Restrictions, S. 157.

<sup>151</sup> Vgl. STRAUB (Fn. 44), Rn. 270.

<sup>152</sup> RAUBER (Fn. 41), Use Restrictions, S. 157.

<sup>153</sup> STRAUB (Fn. 44), Rn. 273 und 279.

<sup>154</sup> Vgl. STRAUB (Fn. 44), Rn. 278.

<sup>155</sup> Zum Ganzen RAUBER (Fn. 49), Computersoftware, S. 227.

<sup>156</sup> ANDREJ RADONIC, Lizenzierung von Anwendungen, <https://www.computerwoche.de/a/vorsicht-virtualisierung,2514557,8> (zit. RADONIC, Anwendungen).

<sup>157</sup> Vgl. SCHNEIDER (Fn. 116), Teil R Rn. 86.

#### 4.2.1. Relevante Lizenzformen

[32] Es gibt zahlreiche Lizenzformen mit unterschiedlichen – bspw. räumlichen, zeitlichen und inhaltlichen – Beschränkungsmöglichkeiten. Im Folgenden stehen inhaltliche Beschränkungen der Nutzungsrechte und Lizenzmetriken im Vordergrund.

##### 4.2.1.1. Einfach- und Mehrfachlizenzen

[33] Softwareüberlassungsverträge beinhalten oft Klauseln, die den Umfang der Nutzung klar festlegen. Nutzungsrechte an Software werden üblicherweise als *Einfach-* oder als *Mehrfachlizenz* eingeräumt. Bei ersterer erhält der Anwender das Recht, das Programm auf einem einzigen Computer zu nutzen, wohingegen bei der Mehrfachlizenz das Recht auf die gleichzeitige Nutzung auf mehreren Geräten eingeräumt wird.<sup>158</sup> Sie kann sich anstatt auf die Anzahl der Rechner auch auf die Anzahl der Anwender beziehen, die die Software gleichzeitig nutzen (Concurrent User).<sup>159</sup> Um allfällig komplexe hardwarebasierte Lizenzmetriken zu umgehen, entscheiden sich Anwender in der Praxis gelegentlich für das Named User Modell. Dabei wird die maximale Anzahl der Nutzer festgelegt und mit einem registrierten, namentlich eingetragenen Zugang auf das Programm ausgestattet.<sup>160</sup> Eine bekannte Zwischenform der Einfach- und Mehrfachlizenz ist die sog. Volumenlizenz, bei welcher mehrere Einfachlizenzen im Paket mit einem Volumenrabatt veräußert werden.<sup>161</sup> Im Hinblick auf den Softwareeinsatz in virtualisierten Umgebungen stehen maschinenbezogene Einfach- und Mehrfachlizenzen im Vordergrund und es muss erörtert werden, wie Container (oder virtuelle Maschinen) einzuordnen sind. Einfach- und Mehrfachlizenzen gestatten mitunter den Einsatz des Programms auf einer bestimmten Höchstzahl von Rechnern. Virtuelle Maschinen sind indessen dazu fähig, komplette Rechner zu simulieren und somit hunderte von Betriebssystemen nebeneinander völlig unabhängig voneinander ablaufen zu lassen. Angesichts dieses technologischen Fortschritts stellt sich die Frage, wie eine solche Recheneinheit definiert werden soll, wobei die Qualifikation einer virtuellen Maschine als eigener Rechner oft vorherrschend ist.<sup>162</sup> Bei den üblichen Applikationscontainern werden nicht mehrere Betriebssysteme nebeneinander betrieben, sondern es wird auf der Ebene des Betriebssystems für zusätzliche Isolation gesorgt. Besonders von Herstellern werden Containerinstanzen aber oft auch als eigene Rechner im Vertragssinne aufgefasst (Kap. 4.2.2.). In der Praxis erfolgt bei Containern – und analog auch bei virtuellen Maschinen – vielfach eine Lizenzierung nach der Zahl der Instanzen, auf denen die Applikation läuft. Die Instanzenanzahl definiert dann den Lizenzbedarf. Die Einzellizenz erfasst den Einsatz auf einer Instanz, während der Betrieb auf mehreren Instanzen gesondert lizenzpflichtig ist.<sup>163</sup>

---

<sup>158</sup> MORSCHER/DORIGO (Fn. 72), S. 53.

<sup>159</sup> MARLY (Fn. 90), Rn. 1194.

<sup>160</sup> Ebd.

<sup>161</sup> MORSCHER/DORIGO (Fn. 72), S. 53.

<sup>162</sup> Vgl. zum Ganzen ASTRID AUER-REINSDORFF/ISABELL CONRAD, Handbuch IT- und Datenschutzrecht, 3. Aufl., München 2019 (zit. AUER-REINSDORFF/CONRAD, § ... Rn. ...), § 12 Rn. 83 ff.

<sup>163</sup> Vgl. GRÜTZMACHER (Fn. 11), Virtualisierung, S. 173 f.

#### 4.2.1.2. CPU- und Core-Lizenzen

[34] Um eine Nutzungsbeschränkung (und damit gelegentlich auch Vergütungsansprüche) an der Leistungsfähigkeit der Hardware auszurichten, bedienen sich Softwareanbieter seit jeher sog. «CPU-Klauseln», heutzutage vermehrt aber auch Klauseln, die auf eine bestimmte Anzahl Kerne in einer Mehrkern-CPU abstellen.<sup>164</sup> Mitunter ist der Übergang von der Begrenzung der Mehrfachnutzung hin zu einer Beschränkung der Nutzungsintensität fließend.<sup>165</sup> Manche CPU-Lizenzen sind in der Praxis mit der Vergütungsregelung gekoppelt.<sup>166</sup> Die urheberrechtliche und insb. die schuldrechtliche Wirkung solcher Klauseln ist hierzulande – noch mehr aber in Deutschland<sup>167</sup> – umstritten (Kap. 5.3.2.).

[35] Unter dem Begriff «CPU-Klauseln» versteht man Klauseln, die die Software-Nutzung an eine bestimmte Zentraleinheit der Hardware (Hauptspeicher und Prozessor<sup>168</sup>) oder an eine bestimmte Prozessorfamilie binden.<sup>169</sup> Der deutsche Autor MARLY ist der Ansicht, diese Bezeichnung sei unglücklich, da die jüngeren Mehrprozessorsysteme oder Mehrkern-CPU's einer Anbindung an lediglich eine CPU entgegenstehen und weil virtuelle Systeme keinen eigenen Prozessor haben, aber dennoch mitumfasst werden. Aus diesem Grund bevorzugt er den Begriff «Systemvereinbarungen».<sup>170</sup> Die Beschränkungen können konkreter oder abstrakter Natur sowie mit oder ohne Upgrade-Recht ausgestattet sein.<sup>171</sup> Ein solches wird bei einem Wechsel auf eine Hardware mit höherer Ablaufgeschwindigkeit ausgeübt, unter der Bedingung einer zusätzlichen Lizenzgebühr.<sup>172</sup> Ebenfalls unter den Überbegriff der «CPU-Klausel» fallen Beschränkungen der Softwarenutzung auf Systeme mit einer bestimmten Anzahl an Prozessoren oder Prozessorkernen.<sup>173,174</sup> Modelle, die auf den Prozessor bzw. die Zahl der Prozessoren, Cores o.Ä. abstellen, nennt man auch «Prozessor-Modelle». Es werden damit nutzungsintensitätsabhängige Vergütungen erzielt.<sup>175</sup>

[36] Im Laufe der Zeit haben technische Innovationen bei solchen Bestimmungen zu Unklarheiten geführt. Systeme verfügen heutzutage meist über mehrere Prozessoren. In einen Prozessorchip können ausserdem seit 2005 auch mehrere Prozessorkerne integriert werden (sog. Mehrkernprozessoren, englisch *Multicoreprocessors*), um eine erhöhte Rechenleistung zu erreichen. Die dadurch erwünschte Steigerung der Leistungsfähigkeit einer Software hängt dann aber vor allem von deren Parallelisierungsgrad ab.<sup>176</sup> Ältere Lizenzgebührenmodelle stellten im Gegensatz zu

---

<sup>164</sup> GRÜTZMACHER (Fn. 120), Nutzungsformen 2, S. 698.

<sup>165</sup> AUER-REINSDORFF/CONRAD (Fn. 162), § 12 Rn. 80.

<sup>166</sup> Vgl. SCHNEIDER (Fn. 116), Teil R Rn. 115 f., welcher von «einer Mehrfachvergütung in Abhängigkeit von der Nutzungsintensität» spricht; vgl. GRÜTZMACHER (Fn. 120), Nutzungsformen 2, S. 698.

<sup>167</sup> Dazu ausführlich MARLY (Fn. 90), Rn. 1675 ff.

<sup>168</sup> Die Begriffe «Prozessor» und «CPU» (Central Processing Unit) werden im Folgenden als Synonyme verwendet. Gemeint ist damit in diesem Kontext der Hauptprozessor, d.h. die zentrale Verarbeitungseinheit in einem Computer. Siehe dazu CPU, <https://www.elektronik-kompodium.de/sites/com/0309161.htm>.

<sup>169</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 1727 und 1732.

<sup>170</sup> Zum Ganzen MARLY (Fn. 90), Rn. 1665.

<sup>171</sup> GRÜTZMACHER (Fn. 120), Nutzungsformen 2, S. 698.

<sup>172</sup> AUER-REINSDORFF/CONRAD (Fn. 162), § 12 Rn. 93.

<sup>173</sup> Mit dem Prozessorkern (auch Core) ist der zentrale Teil einer CPU gemeint. Siehe dazu Prozessorkern, <https://de.wikipedia.org/wiki/Prozessorkern>.

<sup>174</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 1735.

<sup>175</sup> Vgl. SCHNEIDER (Fn. 116), Teil R Rn. 89.

<sup>176</sup> Siehe dazu Mehrkernprozessor, <https://de.wikipedia.org/wiki/Mehrkernprozessor>.

heute oft auf die Anzahl der Prozessoren ab, ohne die Anzahl der Prozessorkerne zu berücksichtigen.<sup>177</sup> Ist heute von «Prozessoren» die Rede, sind darunter im Zweifel die Prozessorchips und nicht die Prozessorkerne zu verstehen.<sup>178</sup> Stellt der Vertrag auf eine bestimmte Anzahl CPU ab, ohne eine spezifische Regelung zum Einsatz in virtualisierten Umgebungen zu enthalten, darf dem entsprechenden Programm im virtualisierten System nur so viel Prozessorleistung zustehen, wie es der Anzahl vertraglich definierter CPU entspricht. Den massgebenden Nutzungsanteil zu eruieren gestaltet sich in der Praxis schwierig, da virtualisierte Umgebungen dynamisch über mehrere Hardwareeinheiten verteilt werden.<sup>179</sup> Bei virtualisierten Umgebungen sind zudem oft virtuelle Prozessorkerne im Einsatz, auf welchen weitere Programme betrieben werden können. Manche Softwareanbieter beschränken die Nutzung ihrer Programme sowohl auf eine bestimmte Anzahl physischer als auch virtueller Prozessoren.<sup>180</sup> Ausserdem wird in Lizenzmodellen inzwischen auch nach sog. Hardwarethreads lizenziert, die den Besonderheiten von virtuellen Maschinen oder Containern Rechnung tragen (dazu Kap. 4.2.2.2.). Grundsätzlich ist immer fraglich, wie gut sich solche Lizenzmetriken für virtuelle Systeme eignen, denn diese benutzen die Hardware-Ressourcen zeitlich nur partiell, teilen sie mit weiteren «Guests» und werden ggf. zwischen den Servern hin- und hergeschoben oder je nach Bedarf gestartet. Sie nehmen allesamt nur einen Teil der verfügbaren Gesamtprozessorleistung in Anspruch und es wird ihnen keine eigene Hardware zugewiesen, denn sie laufen dynamisch auf mehreren physischen Einheiten.<sup>181</sup> Es kann überdies sinnvoll sein, gleich die Gesamtheit aller physikalischen oder virtuellen Computer (analog auch Containerinstanzen) zu lizenzieren, was durch einige Anbieter mittels Cluster-Lizenzen angeboten wird.<sup>182</sup> Cluster können unterschiedliche Zwecke verfolgen und deren Ausgestaltung variiert nach Hersteller. Teils resultiert die Höhe der Vergütung aus dem Umfang der Nutzungsintensität.<sup>183</sup>

#### 4.2.2. Einzelne Regelungen verschiedener Softwareanbieter

[37] Softwareanbieter, die ihre Software in Online-Verzeichnissen wie «Docker Hub» anbieten, geben im besten Fall die spezifischen Lizenzbedingungen der containerisierten Software an, wobei es oft keine grossen Unterschiede zu den herkömmlichen Lizenzmetriken für virtuelle Maschinen gibt. Ein einheitliches, auf die Virtualisierung abgestimmtes und aus Kostenperspektive faires Modell hat sich noch nicht durchgesetzt. In der Softwarebranche wird momentan mit dem SaaS-Ansatz der Abrechnung nach tatsächlicher Softwarenutzung «*pay per use*» (auch «*on demand*») experimentiert, wobei die Nutzung in der Cloud erfolgt.<sup>184</sup> Beim CaaS-Ansatz bietet der Cloud-Provider sogar ganze Container-Infrastrukturen zur Nutzung an.<sup>185</sup> Aufgrund der star-

---

<sup>177</sup> Vgl. AUER-REINSORFF/CONRAD (Fn. 162), § 12 Rn. 100 (Praxistipp).

<sup>178</sup> STRAUB (Fn. 44), Rn. 295.

<sup>179</sup> Zum Ganzen STRAUB (Fn. 44), Rn. 299 und 301 f.

<sup>180</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 1738.

<sup>181</sup> Vgl. STRAUB (Fn. 44), Rn. 297 f.

<sup>182</sup> Vgl. AUER-REINSORFF/CONRAD (Fn. 162), § 12 Rn. 99.

<sup>183</sup> Vgl. SCHNEIDER (Fn. 116), Teil R Rn. 118 und 122.

<sup>184</sup> Vgl. ANDREJ RADONIC, Softwarelizenzen und Virtualisierung; Pay-per-Use, <https://www.itwissen.info/Pay-per-Use-pay-per-use.html> (zit. RADONIC, Softwarelizenzen und Virtualisierung).

<sup>185</sup> Vgl. TRENT ALLGOOD, Virtual Containers; ITAM Best Practices and Licensing Considerations – Part II, <https://www.anglepoint.com/virtual-containers-pt2>.

ken Abweichung von der klassischen Softwareüberlassung und des begrenzten Umfangs dieser Arbeit wird vorliegend nicht weiter darauf eingegangen.

#### 4.2.2.1. Instanzenbasierte Lizenzierung

[38] Als prominentes Beispiel für die instanzenbasierte Lizenzierung können die Lizenzbestimmungen von SQL Server 2019, einer bekannten Datenplattform aus dem Hause Microsoft, genannt werden. Die Standard Edition kann über Server- und Zugriffslizenzen lizenziert werden (Server+CAL *licensing model*), wobei es sich um Volumenlizenzen handelt. Beim Client/Server Modell geht es typischerweise darum, ein Nutzungsrecht anhand der technischen Definition der beteiligten Server und der Clients, auf welchen die Nutzer die Software nutzen, einzuräumen.<sup>186</sup> In casu benötigt jede Betriebssysteminstanz, egal ob physisch oder virtuell, auf welcher die SQL 2019 Standard Software läuft, eine Server-Lizenz. So steht in den Licensing Terms: *«For Products under the Server/CAL License Model, customer may use one Running Instance of server software in either a Physical OSE or Virtual OSE on a Licensed Server for each License it acquires.»*<sup>187</sup> Um den Einsatz von Containerlösungen zu regeln, werden diese einer virtuellen Betriebssysteminstanz gleichgestellt: *«For purposes of licensing use of SQL Server software running within a container on a container runtime such as docker, cri-o or containerd, (i) a container is considered to be a Virtual OSE, [...]»*.<sup>188</sup> Folglich braucht auch jeder Container eine Server-Lizenz. Der zusätzliche Licensing Guide von Microsoft zeigt zwar ausführlich die strukturellen Unterschiede einer virtuellen Maschine und eines Containers auf, hält dann aber fest: *«[...] Containers and virtual machines are structured differently, but they are considered the same from a licensing perspective.»*<sup>189</sup> Demnach werden Container gleich wie virtuelle Maschinen behandelt, auch wenn sie nicht ein komplettes Betriebssystem virtualisieren. Es sind dann aber noch Zugriffslizenzen (Nutzer oder Gerät) nötig.<sup>190</sup> Die Hardware und die virtuellen Ressourcen können sodann ohne zusätzliche Kosten im Rahmen der Lizenzbestimmungen frei skaliert werden.<sup>191</sup>

[39] Ein Beispiel für eine Einfachlizenz ist in den Lizenzbestimmungen vom kleineren Softwareanbieter ColorLogic zu finden, der im Bereich von Farbmanagement-Technologien entsprechende Lösungen anbietet. Dort steht explizit: *«[...] Sie sind berechtigt, die Software auf nur einem Computer gleichzeitig zu benutzen. Virtuelle Maschinen und Docker-Container werden hierbei als separate Computer gewertet. [...]»*.<sup>192</sup> Docker-Container und virtuelle Maschinen werden einander wieder explizit gleichgestellt und als eigener Rechner definiert. Das Betreiben der Software ist demgemäss nur auf *einer* Container-Instanz bzw. *einer* Instanz einer virtuellen Maschine von der Einzellizenz erfasst. Je nach Anzahl Container kann eine Lizenzierung nach Containerinstanzen aus der Sicht des Anwenders ein teures Szenario darstellen. Bei der Lizenzierung von Instanzen gibt es verschiedene Möglichkeiten. Lizenzen können die Anzahl der Instanzen beschränken bzw.

---

<sup>186</sup> Vgl. AUER-REINSDORFF/CONRAD, § 12 Rn. 97.

<sup>187</sup> Microsoft, Licensing Terms, <https://www.microsoft.com/licensing/terms/productoffering/SQLServer/MCA>.

<sup>188</sup> Microsoft, Licensing Terms (Fn. 187).

<sup>189</sup> Microsoft, Licensing Guide, <https://download.microsoft.com/download/6/6/0/66078040-86d8-4f6e-b0c5-e9919bbcb537/SQL%20Server%202019%20Licensing%20guide.pdf>, S. 23.

<sup>190</sup> Vgl. Microsoft, Licensing Guide (Fn. 189), S. 25.

<sup>191</sup> Microsoft SQL Server 2019 Lizenzierung, <https://www.software-express.de/hersteller/microsoft/sql-server/lizenzierung/>.

<sup>192</sup> ColorLogic GmbH, Lizenzvereinbarung, <https://colorlogic.de/lizenzvereinbarung/>.



sich an ihnen orientieren oder auch für den Softwareeinsatz auf einem Server eine unbeschränkte Zahl von virtuellen Lizenzen vorsehen.<sup>193</sup> Eine rechtliche Einordnung folgt in Kap. 5.3.1.

#### 4.2.2.2. Hardwarebasierte Lizenzierung

[40] Traditionelle Lizenzmodelle orientieren sich oft an Hardwarekomponenten und stossen beim Softwareeinsatz in virtuellen Umgebungen, sei es in virtuellen Maschinen oder auch in Containern, an ihre Grenzen. Bei den virtuellen Maschinen hat sich die Frage schon etwas früher, vor dem grossen Aufkommen der Container, gestellt. Einsparungen, die das Unternehmen durch Virtualisierungstechnologien erzielt hatte, wurden wieder ausgeglichen durch hohe Softwarekosten, welche durch herkömmliche Lizenzmodelle, die für die Preisbestimmung v.a. auf die Anzahl Prozessoren im virtuellen System abstellten, verursacht.<sup>194</sup>

[41] Oracle, einer der grössten Softwareanbieter, stellt ein breit ausgearbeitetes und verschachteltes Vertragswerk mit vielen verschiedenen AGB und Anlagen bereit. Bei softwarebasierenden virtuellen Maschinen wird von einer «Soft-Partitionierung» ausgegangen, da Ressourcen flexibel verteilt werden.<sup>195</sup> Oracle und andere Softwareanbieter sehen für den Fall der Applikationsvirtualisierung vor, dass ihre Anwendung für den gesamten Serververbund lizenziert wird und nicht nur für die Prozessoren, denen die Applikation zugewiesen ist.<sup>196</sup> Gängige Containerprodukte wie Docker sind ebenfalls nicht bei der «Hard-Partitionierung» aufgeführt, was *e contrario* bedeutet, dass auf der Basis einer «Soft-Partitionierung» lizenziert wird.<sup>197</sup> In der Oracle Partitioning Policy, welche definiert, unter welchen Konditionen Lizenzen verlangt werden, wird aufgeführt: «Once a container image (e.g. a Docker image) containing Oracle Programs has been pulled to a host, or to a Kubernetes node in a Kubernetes cluster, (either a virtual machine or a physical machine), that host or Kubernetes node must be licensed for the Oracle Programs for the number of processors on that host or Kubernetes node.»<sup>198</sup> Alle im Server vorhandenen physischen Prozessoren und deren Kerne müssen also lizenziert werden, unabhängig davon, wie viele Ressourcen die virtuelle Maschine (oder der Container) tatsächlich adressiert und nutzt.<sup>199</sup> Bei der Standard Edition<sup>200</sup> bspw. benötigt man bei einem Virtualisierungs-Cluster, welcher aus fünf Servern mit je zwei Prozessoren besteht, zehn Lizenzen. Die Diskrepanz zwischen genutzter und zu lizenzierender Leistung kann erheblich sein.<sup>201</sup> Einige wenige Ausnahmen sind von diesem «Hardine-Approach» ausgenommen. Sobald aber die gesamte Infrastruktur lizenziert ist, kann eine unbeschränkte Anzahl an

---

<sup>193</sup> Vgl. GRÜTZMACHER (Fn. 11), Virtualisierung, S. 174.

<sup>194</sup> Vgl. LUDGER SCHMITZ, Lizenzen bremsen Virtualisierung aus, <https://tinyurl.com/yb4wmpjj> (zit. SCHMITZ, Lizenzen bremsen Virtualisierung aus).

<sup>195</sup> Vgl. Oracle, Partitioning Policy, <https://www.oracle.com/assets/partitioning-070609.pdf>, S. 1 f.

<sup>196</sup> Vgl. GRÜTZMACHER (Fn. 11), Virtualisierung, S. 176.

<sup>197</sup> Oracle, Partitioning Policy (Fn. 195), S. 1; Running and Licensing Oracle Programs in Containers and Kubernetes, <https://www.oracle.com/a/tech/docs/running-and-licensing-programs-in-containers-and-kubernetes.pdf>, S. 5.

<sup>198</sup> Oracle, Partitioning Policy (Fn. 195), S. 5.

<sup>199</sup> RADONIC (Fn. 156), Anwendungen.

<sup>200</sup> Jedes Oracle-Produkt, das «Standard» im Produktnamen hat, wird so lizenziert (bspw. Datenbanken, Web-Logic und BI Suite), siehe dazu JOCHEN KUTSCHERUK, Oracle-Lizenzierung in Zeiten der Virtualisierung, DOAG/SOUG News 5-2015, S. 30–33, S. 30.

<sup>201</sup> Vgl. zum Ganzen KUTSCHERUK (Fn. 200), S. 31.

Containern in Betrieb genommen werden. Da oft zahlreiche Container – üblicherweise mehr als virtuelle Maschinen – in Betrieb sind, birgt diese Regelung je nach Situation auch Vorteile.<sup>202</sup>

[42] Bei der Lizenzierung des SQL Servers 2019 (bereits behandelt für die Standardausgabe in Kap. 4.2.2.1.) entschied sich *Microsoft* bei der Enterpriseversion, welche sich eher an grössere Organisationen richtet, nach Prozessorkernen zu lizenzieren (Per-Core-Modell).<sup>203</sup> Container werden auch bei der Core-Lizenzierung wieder den virtuellen Maschinen aus Lizenzierungsperspektive gleichgestellt.<sup>204</sup> Das Core-Lizenzmodell von *Microsoft* ermöglicht einerseits die maximale Virtualisierung, bei welcher u.U. das Ausführen einer unbegrenzten Anzahl an virtuellen Maschinen oder Containern erlaubt ist, wenn alle physischen Prozessorkerne des Hostservers lizenziert werden.<sup>205</sup> Um nicht das ganze physische Cluster zu lizenzieren, können andererseits auch die individuellen Container oder die einzelnen virtuellen Maschinen lizenziert werden, dies gemäss den ihnen zugewiesenen v-Cores (oder v-CPU's, virtuelle Threads).<sup>206</sup> Diese virtuellen Einheiten stellen – vereinfacht gesagt – die Pendanten der physischen Einheiten in der virtualisierten Umgebung dar und werden tatsächlich aus *time slots* über alle physischen Cores gebildet.<sup>207</sup> Bei Containern wird nicht nach v-Cores lizenziert, sondern nach Hardwarethreads: «[...] (ii) the Physical or Virtual Cores available to that container are considered to be Hardware Threads.»<sup>208</sup>, sodann: «[...] If hyperthreading is enabled and Customer is licensing use under the Per Core License Model, Customer must assign a Core License for each Hardware Thread mapped to a container, subject to a minimum of four Licenses.»<sup>209</sup>. Ein Hardwarethread eines Cores bildet (aus Lizenzierungsperspektive) einen v-Core ab.<sup>210</sup> Ein v-Core kann aber auch mehrere Hardwarethreads repräsentieren.<sup>211</sup> Bei mehreren Hardwarethreads, die einen v-Core unterstützen, müssen alle Threads (aller v-Cores) lizenziert werden, die der virtuellen Maschine oder dem Container zugewiesen sind.<sup>212</sup> Für die Zuweisung von v-Cores/Hardwarethreads an Container braucht es – anders als bei virtuellen Maschinen – eine spezielle Konfiguration, ansonsten werden standardmässig alle physischen Ressourcen in Anspruch genommen.<sup>213</sup> Insbesondere bei Containerlösungen ist eine Lizenzierung auf Basis der Anzahl Cores fragwürdig. Der Container benötigt tatsächlich oft nur einen Bruchteil eines (lizenzierten) Cores, was in casu bei Berücksichtigung der Mindestlizenzierung von vier Cores je nach Szenario zu einer teuren Angelegenheit werden kann. Durch die massiven Skalierungsmöglichkeiten von Containern ergeben sich völlig andere Nutzungsszenarien, für welche erst wenige Softwarehäuser angemessene Lizenzmodelle entwickelt haben. Bei einer Container-Orchestrierung arbeitet man mit Pods, kleinteiligen Plattformen, die eine Gruppe von Applikationscontainern beinhalten. Der tatsächliche Core wird dabei oft so segmentiert, dass bei

---

<sup>202</sup> Vgl. zum Ganzen ALLGOOD (Fn. 185), Virtual Containers.

<sup>203</sup> *Microsoft SQL Server 2019 Lizenzierung* (Fn. 191).

<sup>204</sup> Vgl. Kap. 4.1.2.1.; vgl. *Microsoft, Licensing Guide* (Fn. 189), S. 23.

<sup>205</sup> Vgl. *Microsoft, Licensing Guide* (Fn. 189), S. 19 und 21.

<sup>206</sup> Vgl. *Microsoft, Licensing Guide* (Fn. 189), S. 19 und 23.

<sup>207</sup> v-CPU, <https://www.software-express.de/glossar/vcpu-virtual-central-processing-uni/>.

<sup>208</sup> *Microsoft, Licensing Terms* (Fn. 187).

<sup>209</sup> Ebd.

<sup>210</sup> *Microsoft, Licensing Guide* (Fn. 189), S. 23.

<sup>211</sup> *Microsoft, Licensing Guide* (Fn. 189), S. 12.

<sup>212</sup> FAQ, <https://www.software-express.de/info33/sql-server-lizenzierung-faq/>.

<sup>213</sup> Vgl. ALLGOOD (Fn. 185), Virtual Containers.

OpenShift<sup>214</sup> z.B. 1000 Millicores (entsprechen zusammen einem Core) ressourcentechnisch auf die verschiedenen Pods verteilt werden können.<sup>215</sup> Dabei den ganzen Core oder sogar mehrere Cores zu lizenzieren, obwohl nur ein Tausendstel tatsächlich genutzt wird, ist fragwürdig und macht die Anwendung von Containern kostspielig.

[43] Schliesslich ist noch auf die Lizenzbestimmungen von IBM, einem weiteren grossen Softwareanbieter, einzugehen, welcher wie seine Konkurrenten über ein vielschichtiges Regelwerk mit zahlreichen Anlagen verfügt. Es handelt sich um ein ähnliches Modell wie dasjenige von Oracle, jedoch wird bei der «Soft-Partitionierung» anhand des IBM License Metric Tools die maximale tatsächliche Prozessornutzung erörtert und nur diese lizenziert.<sup>216</sup> Mittels der spezifischen Lizenzmetrik des «Processor Value Units» (PVU) wird die Leistungsfähigkeit der Prozessoren gemessen.<sup>217</sup> Das Gesagte gilt primär für virtuelle Maschinen und es müssen bestimmte Sub-Capacity-Lizenzierungsbedingungen erfüllt sein.<sup>218</sup> In den Vertragsbestimmungen ist folgende Nutzungsbeschränkung ersichtlich: «Vor einer Erweiterung der Virtualisierungskapazität eines berechtigten Sub-Capacity-Produkts muss der Kunde zuerst ausreichende Lizenzen [...] zur Abdeckung der Erweiterung erwerben.»<sup>219</sup> Anders als andere Anbieter unterscheidet IBM zwischen Containern und virtuellen Maschinen. Erst gerade kürzlich, im Oktober 2020, hat das Unternehmen neue Lizenzbestimmungen zum Einsatz von IBM Software in containerisierten Unternehmungen veröffentlicht, vor allem in Hinblick auf die zukunftsrelevanten (Hybrid-)Clouds und den damit einhergehenden grossen Container-Orchestrierungen. Den kleinteiligen Pods, worauf Container laufen, soll mit einer neuen, granulareren Zählweise bei der Messung der v-CPU Kapazität besser Rechnung getragen werden.<sup>220</sup> Sind die spezifischen Bedingungen für die Containerlizenzierung<sup>221</sup> erfüllt, hält der «Anhang – Spezielles Angebot für Containerlizenzierung» fest, der Kunde müsse: «[...] die Berechtigungen für die Gesamtzahl der Kerne (Cores) erwerben, die zur Kapazität aller Container beitragen, die für das Berechtigte Containerprodukt verfügbar sind.»<sup>222</sup>, ansonsten «[...] muss der Kunde die Gesamtzahl der physischen Prozessorkerne lizenzieren, die zur Nutzung auf allen Servern, auf denen das Berechtigte Produkt bereitgestellt wird, aktiviert und verfügbar sind (Volle Kapazität).»<sup>223</sup>. Es ist also nicht wie bei anderen Anbietern nötig, der ganze Container-Cluster zu lizenzieren, sondern gesamthaft gesehen nur die addierten v-CPU Kapazitäten der

---

<sup>214</sup> Die OpenShift Container Platform bietet Kubernetes-Umgebungen für Unternehmen an, welche zum Erstellen, Bereitstellen und Verwalten von containerbasierten Anwendungen auf jedem öffentlichen oder privaten Rechenzentrum dienen, auf denen Red Hat Enterprise Linux unterstützt wird. Siehe dazu Openshift, <https://de.wikipedia.org/wiki/OpenShift>.

<sup>215</sup> Vgl. zum Ganzen HANSGEORG LANGHORST, Lizenzmanagement in Zeiten von Cloud und DevOps – Klar Schiff im Containerterminal?, <https://tinyurl.com/yanllevc> (zit. LANGHORST, Lizenzmanagement in Zeiten von Cloud und DevOps).

<sup>216</sup> RADONIC (Fn.156), Anwendungen.

<sup>217</sup> Zu den «PVU» siehe MARLY (Fn. 90), S. 1194.

<sup>218</sup> Vgl. Sup-Capacity Licensing, <https://www.ibm.com/software/passportadvantage/subcaplicensing.html>.

<sup>219</sup> IBM, International Passport Advantage Vertrag, [http://public.dhe.ibm.com/software/passportadvantage/PA\\_Agreements/PA\\_Agreement\\_German.pdf](http://public.dhe.ibm.com/software/passportadvantage/PA_Agreements/PA_Agreement_German.pdf), Ziff. 1.13.

<sup>220</sup> Vgl. zum Ganzen TOM ESPENSEN, IBM Container Licensing – another Hybrid Cloud milestone for Big Blue, <https://tinyurl.com/y9ohvdkz> (zit. ESPENSEN, IBM Container Licensing).

<sup>221</sup> Vgl. IBM, Container Licenses, <https://www.ibm.com/software/passportadvantage/containerlicenses.html>.

<sup>222</sup> IBM, Anhang – Spezielles Angebot für Containerlizenzierung, [http://public.dhe.ibm.com/software/passportadvantage/Containers/Addendum\\_to\\_Quotation\\_Terms\\_Special\\_Option\\_for\\_Container\\_Licensing\\_de\\_DE.pdf](http://public.dhe.ibm.com/software/passportadvantage/Containers/Addendum_to_Quotation_Terms_Special_Option_for_Container_Licensing_de_DE.pdf), Ziff. 1.

<sup>223</sup> IBM, Anhang – Spezielles Angebot für Containerlizenzierung (Fn. 222), Ziff. 2.

containerisierten Applikationen.<sup>224</sup> Wenn es sich um Kapazitäten handelt, die weniger als einem v-Core entsprechen, ist der betreffende Bruchteil massgeblich.<sup>225</sup> Alle diese Bruchteile werden dann addiert und auf die nächste ganze Zahl gerundet, die für die Lizenzierung relevant ist, und nicht die aufgerundete Kapazität jedes einzelnen Pods/Nodes. Im Fokus steht dabei der Einsatz auf Kubernetes-Orchestrierungen.<sup>226</sup> Für die Kunden von IBM birgt diese granulare Zählweise beim Einsatz von Containern grosse Kostenvorteile, wenn nicht die ganzen v-CPUs lizenziert werden müssen.<sup>227</sup> Eine zentrale Voraussetzung für die IBM-Containerlizenzierung ist der Besitz des Zählprogramms «IBM License Service».<sup>228</sup> M.E. legt IBM mit der Einführung dieser Lizenzmetrik einen Meilenstein für die Lizenzierung von Containern, da deren ressourcenschonende, kleinteilige Struktur berücksichtigt wird, was bei grossen Container-Orchestrierungen zu einer angemessenen Vergütung führt.

[44] Im Gespräch mit Praktikern hat sich herausgestellt, dass es sich zum Teil als schwierig erweist, in diesem Durcheinander von Lizenzbedingungen den Durchblick zu bewahren und die Lizenz-Compliance sicherzustellen. Fundierte Kenntnisse der Regelwerke sind notwendig, um einerseits Überlizenzierungen zu verhindern, andererseits auch Sanktionen als Folge einer Unterlizenzierung zu vermeiden. Auf dem Markt gibt es zahlreiche Programme, die hierbei Abhilfe verschaffen können. Gängige Vorteile der Container-Technologie wie das «Scale-Up»<sup>229</sup> und das «Scale-Out»<sup>230</sup> können in der Praxis durch die verbreiteten Lizenzmodelle schnell zum Kostenfaktor werden. Lizenzen, die sich auf fixe Hardwarekomponenten oder Instanzenanzahl beziehen, eignen sich aus der Kostenperspektive nicht für Container-Landschaften, bei welchen man fliegend Ressourcen anpassen und für eine bestimmte Zeit dazu- oder abschalten kann.

## 5. Verhältnis zwischen vertrags- und urheberrechtlicher Ebene

[45] Im letzten Teil dieser Arbeit soll auf die Wechselwirkung zwischen dem Urheber- und dem Vertragsrecht eingegangen werden. Nach den generellen Grundlagen sollen kurz die Rechtsfolgen einer Vertragsverletzung sowie einer kumulativen Urheberrechtsverletzung skizziert werden, um danach die urheberrechtliche Tragweite bzw. die bloss schuldrechtliche Natur der spezifischen Lizenzvereinbarungen im Bereich der Virtualisierung zu erörtern.

---

<sup>224</sup> IBM, Container Licenses FAQ, <https://www.ibm.com/software/passportadvantage/containerfaqov.html>, Ziff. 2.

<sup>225</sup> Vgl. IBM, Container Licenses (Fn. 221), «Pods with vCPU capacity less than one core will be counted as fractions.».

<sup>226</sup> IBM, Container Licenses FAQ (Fn. 224), Ziff. 1 f.

<sup>227</sup> Vgl. ESPENSEN (Fn. 220), IBM Container Licensing.

<sup>228</sup> Vgl. IBM, Container Licenses (Fn. 221).

<sup>229</sup> Mit dem «Scale-Up» (vertikale Skalierung) ist das Steigern der Leistung durch das Hinzufügen von Ressourcen zu einem Knoten/Rechner des Systems gemeint. Siehe dazu Skalierbarkeit, <https://de.wikipedia.org/wiki/Skalierbarkeit>.

<sup>230</sup> «Scale-Out» (horizontale Skalierung) steht für die Leistungssteigerung eines Systems durch das Hinzufügen zusätzlicher Rechner/Knoten, unabhängig von der Hardware. Siehe dazu Wikipedia, Skalierbarkeit (Fn. 229).

## 5.1. Urheberrechtliche Wirkung vertraglicher Nutzungsbestimmungen

[46] Nutzungsbeschränkungen im Lizenzvertrag sind zunächst vertragliche Bestimmungen, doch es bestehen auch enge und vielschichtige Zusammenhänge zum Urheberrecht an der Software.<sup>231</sup> Es ist daran zu erinnern, dass am Bestand des Urheberrechts nichts geändert wird und dem Anwender lediglich ein obligatorisch wirkender Anspruch auf Nutzung der Software eingeräumt wird.<sup>232</sup> Das gesetzliche Gebrauchsrecht und der Überlassungsvertrag hängen insofern zusammen, als dass der Umfang des gesetzlichen Gebrauchsrechts oft vertraglich definiert wird, wobei aber bei Begrenzungen auf gebrauchrechtlicher Ebene *urheberrechtlich wirkende Nutzungsbeschränkungen* vorausgesetzt werden. Solche urheberrechtlich wirkenden Klauseln sind dann auch gegenüber nachfolgenden Erwerbern bindend.<sup>233</sup> Wichtig in der Praxis ist dies vor allem im Falle unwirksamer AGB des Herstellers beim Softwareerwerb über einen Zwischenhändler.<sup>234</sup> Es gibt aber auch rein *vertraglich (wirkende) Nutzungsbefugnisse*, die das gesetzliche Gebrauchsrecht erweitern, einschränken oder konkretisieren können, wobei diese dann einzig zwischen den Parteien – also *inter partes* – wirken und bei einer Weiterveräußerung der Software entfallen.<sup>235</sup> Bei solchen schuldrechtlichen Beschränkungen hängt die Durchsetzbarkeit lediglich davon ab, dass keine kartell- und lauterkeitsrechtlichen Unwirksamkeitsgründe vorliegen.<sup>236</sup> Der zwingende Kerngehalt des Gebrauchsrechts nach Art. 12 Abs. 2 URG setzt der Vertragsfreiheit ebenfalls Grenzen.<sup>237</sup>

[47] Auf Ebene des Gebrauchsrechts wird dem Programmiererwerber erlaubt, sonst nur dem Rechteinhaber vorbehaltene Nutzungshandlungen gemäss Art. 10 URG vorzunehmen, wobei je nach dem eine Begrenzung auf einzelne davon stattfindet.<sup>238</sup> Für *urheberrechtlich (oder auch absolut) wirkende* Beschränkungen auf Ebene des Gebrauchsrechts wird vorausgesetzt, dass es sich um wirtschaftlich selbständige Verwertungs- und Vertriebsformen bzw. wirtschaftlich-technisch selbständige Nutzungsarten handelt.<sup>239</sup> Andersartige Nutzungsbeschränkungen hätten nur *vertragliche Wirkung*. Aus Herstellersicht bezwecken urheberrechtliche Beschränkungen die Sicherung seiner Marktstellung sowie seiner Erträge aus dem Softwarevertrieb.<sup>240</sup> Nach der – nach wie vor noch nicht ganz einheitlichen – Verkehrsauffassung deutet die Erschliessung eines neuen Verbraucher- oder Nutzerkreises auf eine selbständige Nutzungsart hin. Darunter fällt bspw. die Regelung, die die Softwarenutzung nur auf einer bestimmten Anzahl von Arbeitsplätzen erlaubt.<sup>241</sup> Von urheberrechtlicher Bedeutung sind ausserdem nur solche Vereinbarungen, die dem Partizipationsinteresse des Herstellers dienen, ansonsten wären sie rein schuldrechtlicher Natur.<sup>242</sup> Letztendlich muss das Programmexemplar im Sinne des zwingenden Kerns aber in jedem

---

<sup>231</sup> Vgl. RAUBER (Fn. 41), Use Restrictions, S. 140.

<sup>232</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 1694.

<sup>233</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 154 f. und 1691; vgl. STRAUB (Fn. 44), Rn. 179 und 181.

<sup>234</sup> STRAUB (Fn. 44), Rn. 181.

<sup>235</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 1692.

<sup>236</sup> RAUBER (Fn. 41), Use Restrictions, S. 140.

<sup>237</sup> STRAUB (Fn. 44), Rn. 165 und 262; vgl. RAUBER (Fn. 41), Use Restrictions, S. 141.

<sup>238</sup> Vgl. STRAUB (Fn. 44), Rn. 183; vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 1695.

<sup>239</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 155 und 1695.

<sup>240</sup> RAUBER (Fn. 41), Use Restrictions, S. 140.

<sup>241</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 1696; MARLY (Fn. 90), Rz. 1698.

<sup>242</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 1697.

Fall gebraucht werden können, was die Installation und den Ablauf des installierten Programms sowie ggf. die Fehlerbehebung umfasst.<sup>243</sup>

[48] Aus den vorgängigen Erläuterungen ist zu folgern, dass eine Nutzung über den vertraglich vereinbarten Umfang des gesetzlichen Gebrauchsrechts hinaus nicht nur eine *Vertragsverletzung*, sondern zugleich eine *Urheberrechtsverletzung* darstellt.<sup>244</sup> Die Verletzung einer rein schuldrechtlich wirkenden Vereinbarung stellt lediglich eine Vertragsverletzung dar.<sup>245</sup>

[49] Die schweizerischen Gerichte mussten sich dazu bisher nicht konkret äussern. Dem EuGH wurden dagegen in jüngster Zeit vom Cour d'appel de Paris einige Fragen zum Verhältnis des Vertragsrechts zum Urheberrecht vorgelegt.<sup>246</sup> Hintergrund bildet ein Rechtsstreit in Frankreich, bei welchem es darum ging, dass der Lizenznehmer «Free Mobile» die vom Lizenzgeber «IT Development» lizenzierte Software angeblich unrechtmässig bearbeitet und somit gegen das entsprechende vertragliche Verbot verstossen hat. Ob tatsächlich eine Vertrags- oder Urheberrechtsverletzung (in casu Verletzung der aus Art. 4 Software-RL vermittelten Rechte) vorliegt, ist fraglich<sup>247</sup>, stand aber auch nicht zur Debatte, vielmehr ging es darum, ob aus einer Vertragsverletzung auch eine Urheberrechtsverletzung erwachsen kann. Laut dem deutschen Autor GRÜTZMACHER ist diese Fragestellung unglücklich, «denn die Frage, ob eine Urheberrechtsverletzung vorliegt oder nicht, richtet sich nicht danach, ob gegen eine vertragliche Regelung verstossen wird oder nicht, sondern danach, ob in urheberrechtsrelevanter Weise in das Urheberrecht am Computerprogramm eingegriffen wird.»<sup>248</sup> Der EuGH hält *summa summarum* fest, dass die Verletzung einer Klausel eines Lizenzvertrags, insofern sie die Urheberrechte des Inhabers betrifft, auch eine Urheberrechtsverletzung darstellt.<sup>249</sup> Das bekannte CPU-Urteil des BGH<sup>250</sup> ist in diesem Bereich ebenfalls wegweisend. Aus den Ausführungen des Gerichts lässt sich schliessen, dass bei Lizenzbegrenzungen, die sich mit eigenständigen Nutzungsarten decken, auch urheberrechtliche Beschränkungen vorliegen.<sup>251</sup> In der deutschen Lehre wird diese urheberrechtliche Wirkung bestimmter Lizenzklauseln bzw. -beschränkungen vielfach auch als (urheberrechtlich)-*dinglicher* Charakter bezeichnet.<sup>252</sup> Dieses Attribut sollte zurückhaltend angewendet werden, denn aufgrund der gegenwärtigen Konzeption des schweizerischen Rechts erweisen sich die Rechtsregeln zum dinglichen Eigentum beim Urheberrecht, welches das geistige Eigentum betrifft, nicht immer als passend. Bei immateriellen Gütern fehlt es erstens am Besitz, welcher nicht mit dem Besitz an einem Werkexemplar verwechselt werden darf, und zweitens ist auch kein Register (wie z.B. das Grundbuch)

---

<sup>243</sup> Vgl. RAUBER (Fn. 49), *Computersoftware*, S. 184 f.

<sup>244</sup> STRAUB (Fn. 44), Rn. 179.

<sup>245</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 1697.

<sup>246</sup> Urteil des EuGH vom 18. Dezember 2019 Rs. C-666/18 *IT Development gegen Free Mobile*.

<sup>247</sup> Kritisch dazu MALTE GRÜTZMACHER, Wenn die Urheberrechtsverletzung zur Vertragsverletzung wird – Anmerkung zu EuGH, Urteil vom 18.12.2019 – C-666/18 – IT Development/Free Mobile, ZUM 2020, S. 245–248 (zit. GRÜTZMACHER, EuGH), S. 246.

<sup>248</sup> GRÜTZMACHER (Fn. 247), EuGH, S. 246.

<sup>249</sup> Urteil des EuGH vom 18. Dezember 2019 Rs. C-666/18 *IT Development gegen Free Mobile*, Rn. 42.

<sup>250</sup> Urteil des BGH vom 24. Oktober 2002 (I ZR 3/00), *CPU-Klausel*, CR 2003, S. 323.

<sup>251</sup> Urteil des BGH vom 24. Oktober 2002 (I ZR 3/00), *CPU-Klausel*, CR 2003, S. 323 (325); vgl. MALTE GRÜTZMACHER, Lizenzmetriken und Copyright – ein Widerspruch?, ITRB 2017, S. 141–147 (zit. GRÜTZMACHER, Lizenzmetriken), S. 141.

<sup>252</sup> Etwa GRÜTZMACHER (Fn. 251), Lizenzmetriken, S. 141; DERS., Lizenzgestaltung für neue Nutzungsformen im Lichte von § 69d UrhG (Teil 1), CR 2011, S. 485–491 (zit. GRÜTZMACHER, Nutzungsformen 1), S. 486 f.; AUER-REINSDORFF/CONRAD (Fn. 162), § 12 Rn. 72.

vorhanden, das als Besitzsubstitut dienen könnte. Während beim Urheberrecht gar kein solches existiert, werden andere Immaterialgüterrechte (Marken-, Patent- und Designrechte) zwar in einem Register eingetragen, doch kommt diesem nicht eine vergleichbare Bedeutung des öffentlichen Glaubens wie etwa beim Grundbuch zu.<sup>253</sup> Beim geistigen Eigentum ist mitunter auch von «quasi-dinglichen» Rechten die Rede<sup>254</sup>, was m.E. besser passt.

## 5.2. Rechtsfolgen bei Lizenzvertragsverletzungen

[50] Wird eine Verletzung des Softwareüberlassungsvertrags z.B. im Rahmen eines Software-Audits festgestellt, kann der Anbieter je nach urheberrechtlicher Tragweite der verletzen Vereinbarung ggf. gesetzliche sowie vertragliche Sanktionen herbeiführen.<sup>255</sup> Betreffend den Schadenersatz im Besonderen ist dieser zwar bei einer Anspruchskonkurrenz wirtschaftlich nur einmal geschuldet, dennoch bestehen die Ansprüche aus vertraglicher und ausservertraglicher Haftung selbständig. Die Folge des Normkonkurrenzverhältnisses zwischen Art. 97 und Art. 41 OR ist dementsprechend nicht die Alternativität, sondern die Kumulation.<sup>256</sup> Bei einer Urheberrechtsverletzung käme zudem auch der durch Art. 67 ff. URG vermittelte strafrechtliche Rechtsschutz in Frage, der vorliegend nicht weiter thematisiert wird.

### 5.2.1. Gesetzliche Sanktionen

[51] Bei einem Verstoß gegen eine Klausel mit urheberrechtlicher Wirkung stehen dem Rechteinhaber aufgrund seiner absolut geschützten Rechtsposition von Gesetzes wegen verschiedene Rechtsbehelfe zur Verfügung. Er kann sie nicht nur gegenüber dem betreffenden Vertragspartner geltend machen, sondern gegenüber jedermann (erga omnes), der derartige Handlungen begeht.<sup>257</sup> Im Bereich des Softwareschutzes relevant sind dabei die Ansprüche auf Verbot einer drohenden Verletzung und Beseitigung einer bestehenden Verletzung nach Art. 62 Abs. 1 lit. a und b URG sowie nach Art. 62 Abs. 2 URG Ansprüche auf Schadenersatz (Art. 41 ff. OR) und Gewinnherausgabe (aus einer GoA gemäss Art. 423 Abs. 1 OR oder aus ungerechtfertigter Bereicherung gemäss Art. 62 OR).<sup>258</sup> Bei widerrechtlich hergestellter Software gibt es zudem Auskunftsansprüche gemäss Art. 62 Abs. 1 lit. c URG. Es käme grundsätzlich auch eine Genugtuung bei Verletzungen des Urheberpersönlichkeitsrechts nach Art. 62 Abs. 2 URG i.V.m. Art. 49 OR in Frage, doch diese ist im Bereich des Softwareschutzes von untergeordneter Bedeutung.<sup>259</sup>

---

<sup>253</sup> Vgl. zum Ganzen HILTY (Fn. 127), Urheberrecht, Rz. 551.

<sup>254</sup> Vgl. OLIVER STÖCKL/ANSELM BRANDI-DOHRN, Der dingliche Charakter von Lizenzen, CR 2011, S. 553–560, S. 554.

<sup>255</sup> HEUSLER/MATHYS (Fn. 79), S. 171 f.

<sup>256</sup> Vgl. PETER GAUCH/WALTER R. SCHLUEP/JÖRG SCHMID/SUSAN EMMENEGGER, Schweizerisches Obligationenrecht Allgemeiner Teil, Band II, 11. Aufl., Zürich 2020, Rn. 2938 und 2940.

<sup>257</sup> HEUSLER/MATHYS (Fn. 79), S. 171 f.

<sup>258</sup> Vgl. URG Komm. – EGLOFF/HEINZMANN (Fn. 66), Art. 62 N 16 f. und 21.

<sup>259</sup> Vgl. STRAUB (Fn. 44), Rn. 545.

### 5.2.2. Vertragliche Sanktionen

[52] Daneben kann der Rechtsinhaber aufgrund der Verletzung des Vertrages Ansprüche auf Schadenersatz gemäss Art. 97 Abs. 1 OR und auf Schadensbeseitigung gemäss Art. 98 Abs. 3 OR geltend machen. Vertraglich werden meistens weitere Sanktionen an eine Urheberrechtsverletzung geknüpft, bspw. die Nachzahlung im Umfang der Übernutzung, der Verfall einer Konventionalstrafe sowie der Entzug der Nutzungsberechtigung, wobei sich letztere bei einer Veräusserung der Software wohl nicht mit dem zwingenden Kern von Art. 12 Abs. 2 URG vereinbaren lässt.<sup>260</sup>

### 5.3. Wirkung von Lizenzklauseln im Bereich der (Container-)Virtualisierung

[53] Bei der folgenden Analyse wird die Konsensbildung zwischen den Parteien vorangestellt. Es wird davon ausgegangen, dass allfällige AGB gültig Vertragsbestandteil geworden sind, gemäss der Darlegung in Kap. 4.1.2.2.

#### 5.3.1. Regelungen nach Zahl der Instanzen

[54] Vertragliche Vereinbarungen, die den Einsatz auf einen Rechner und analog auf eine Instanz (virtuelle Maschine bzw. Container) beschränken, wie sie in den Lizenzbedingungen der Color-Logic vorliegen, wären nicht notwendig. Dennoch sind solche Klauseln aus der Sicht des Anbieters empfehlenswert, um Diskussionen mit den Anwendern zu vermeiden.<sup>261</sup> Die Beschränkung auf eine Instanz ist technisch-wirtschaftlich klar abgrenzbar. Es geht darum, dass dem Anbieter potenzielle Einnahmen weiterer Einzelplatzlizenzen nicht verloren gehen. Bei der Installation mehrerer Instanzen führt das bei der Nutzung pro Container oder pro virtueller Maschine zu einer urheberrechtlichen Vervielfältigung im Arbeitsspeicher.<sup>262</sup> Zwar ist eine solche Vervielfältigung beim Betreiben einer Programmkopie von Art. 12 Abs. 2 URG erfasst, doch kann bei einem mehrfachen Vorhandensein im Arbeitsspeicher (ggf. Storage) nicht mehr von einer bestimmungsgemässen Nutzung ausgegangen werden.<sup>263</sup> Die Beschränkung auf eine bestimmte Anzahl Instanzen bzw. die Lizenzierung nach Instanzenanzahl knüpft direkt an die zusätzlichen Vervielfältigungsstücke an. Es handelt sich hierbei um eine selbständige Nutzungsart und Verwertungsform. Zusätzliche Installationen auf mehreren Instanzen erweitern den Verbraucherkreis. Der Rechtsinhaber ist wie erwähnt berechtigt dazu, die zeitgleiche Mehrfachnutzung seiner Software zu verbieten bzw. zu kontrollieren.<sup>264</sup> Es kann gefolgert werden, dass die Regelungen zur Instanzenanzahl nicht nur rein schuldrechtlicher Natur sind, sondern auch urheberrechtlich wirken. Für die gesetzliche Gebrauchsbefugnis sind solche Regelungen also konstitutiv.<sup>265</sup>

---

<sup>260</sup> Vgl. zum Ganzen HEUSLER/MATHYS (Fn. 79), S. 172.

<sup>261</sup> GRÜTZMACHER (Fn. 11), Virtualisierung, S. 177.

<sup>262</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 1711; GRÜTZMACHER (Fn. 11), Virtualisierung, S. 173.

<sup>263</sup> Analog zum deutschen Recht GRÜTZMACHER (Fn. 11), Virtualisierung, S. 173.

<sup>264</sup> RAUBER (Fn. 44), Use Restrictions, S. 165.

<sup>265</sup> Ebd.



[55] Urheberrechtlich unzulässig im Sinne eines Verstosses gegen den Kerngehalt der Gebrauchsbefugnis nach Art. 12 Abs. 2 URG i.V.m. Art. 17 Abs. 1 lit. a URV wäre in diesem Zusammenhang – in der Lehre nicht unumstritten – vor allem ein generelles Verbot der Nutzung in einer virtuellen Umgebung.<sup>266</sup> Da sich die Installation einer Software in einer virtuellen Instanz nicht entscheidend von derjenigen auf einem physikalischen Rechner unterscheidet und die Installation einer Anwendung in jedem Falle gewährleistet sein muss, schliesse ich mich der oben erwähnten Lehrmeinung an und sehe das allgemeine Verbot des Softwareeinsatzes in einem Container oder einer virtuellen Maschine als Verstoß gegen den zwingenden Kern der Gebrauchsbefugnis an. Demnach entfaltet es weder urheberrechtliche noch vertragliche Wirkungen. Beschränkungen auf bestimmte Technologien sind jedoch denkbar.<sup>267</sup>

[56] Aus kartellrechtlicher Perspektive könnte ein missbräuchliches Verhalten etwa die zusätzlichen Gebühren oder den Erwerb weiterer Lizenzen z.B. bei der Erhöhung der Anzahl User (oder in diesem Kontext auch die Erhöhung der Anzahl Instanzen) betreffen.<sup>268</sup> Bei *unangemessen hohen* Preisen solcher Nachlizenzierungen, die durch marktbeherrschende Unternehmen festgesetzt werden, ist eine Korrektur nach Art. 7 Abs. 2 lit. c KG i.V.m. Art. 13 lit. b KG möglich.

### 5.3.2. Hardwarebezogene Beschränkungen

[57] Es ist zu prüfen, ob vertragliche Beschränkungen durch Lizenzmetriken oder übrige hardwarebezogene Beschränkungen für die Softwarenutzung in containerisierten oder virtualisierten Umgebungen auch urheberrechtliche Wirkung entfalten. Bei CPU-Klauseln ist je nach Typ zu differenzieren. Bei Beschränkungen auf eine *bestimmte* einzelne Zentraleinheit oder einen *bestimmten* Prozessortypen, ist der generelle Konsens dahingehend, dass von einer rein vertraglichen Wirkung ausgegangen wird.<sup>269</sup> Begründet wird dies damit, dass der Urheber kein Partizipationsinteresse am Einsatz auf einer anderen Hardwareeinheit hat, da keine zusätzlichen Verbraucherkreise erschlossen werden.<sup>270</sup> Es fehlt an einer eigenständigen Nutzungsart, da nur die Art der Ausübung des Nutzungsrechts betroffen ist.<sup>271</sup> Die Zahl der Vervielfältigungen ist nicht tangiert.<sup>272</sup>

[58] Umstrittener ist die Thematik bei Klauseln, die den Gebrauch des Programms auf eine bestimmte Anzahl Prozessoren(kerne) oder auf andere nutzungsintensitätsabhängige Lizenzmetriken beschränken, die im Bereich der Virtualisierung oft anzutreffen sind (vgl. Kap. 4.2.2.2.). Erstens ist das Partizipationsinteresse zweifelhaft. Das Nutzungsrecht des Anwenders darf nämlich nicht auf eine bestimmte Ausübungsart beschränkt werden, wenn keine neuen Verbraucherkreise daraus erfolgen, da dies nicht mehr dem Urheberrecht entspricht.<sup>273</sup> Im vorliegenden Fall hat der Einsatz von Software in Containern oder virtuellen Maschinen auf Systemen mit unterschiedlich starker Leistungsfähigkeit m.E. keinen Einfluss auf die Erfassung möglicher Verbraucherkreise. Dennoch bringen die Softwareanbieter oft das Argument, dass durch leistungsfähigere Systeme,

---

<sup>266</sup> Analog zum deutschen Recht GRÜTZMACHER (Fn. 11), Virtualisierung, S. 177; a.A. indirekt KOCH (Fn. 96), S. 917.

<sup>267</sup> Vgl. STRAUB (Fn. 44), Rn. 300.

<sup>268</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 302.

<sup>269</sup> Etwa FRÖHLICH-BLEULER (Fn. 59), Rn. 1729 und 1733; RAUBER (Fn. 49), Computersoftware, S. 232 f.; DERS. (Fn. 41), Use Restrictions, S. 169; WANG (Fn. 101), S. 110 f.; STAUB (Fn. 138), IP-Klauseln, S. 187; STRAUB (Fn. 44), Rn. 289.

<sup>270</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 1729 und 1733.

<sup>271</sup> Vgl. WANG (Fn. 101), S. 111.

<sup>272</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 1735 f.; GRÜTZMACHER (Fn. 120), Nutzungsarten 2, S. 699.

<sup>273</sup> Vgl. MARLY (Fn. 90), Rn. 1671.

die Software auch intensiver genutzt werden könne, was eine Entgelterhöhung rechtfertige.<sup>274</sup> Das hat m.E. nichts mit dem Partizipationsinteresse zu tun, denn die Programmnutzung in Containern oder virtuellen Maschinen wurde schon zu Beginn mit der Lizenzgebühr vergütet und es ergeben sich nach wie vor keine neuen Verbraucherkreise durch die intensivere Nutzung auf leistungsstärkeren virtuellen oder physischen Ressourcen. Die Hersteller möchten finanziell von der Mehrnutzung des Programms aufgrund der höheren Prozessorenleistung profitieren. Es soll jedoch gesagt sein, dass eine solche u.a. davon abhängt, ob eine Software auch für den Parallelbetrieb entwickelt wurde.<sup>275</sup> Technisch gesehen ist hierbei anzumerken, dass eine Prozessoraufrüstung insb. bei virtualisierten Systemen nicht unbedingt zu einer intensiveren Nutzung führt. Es werden zwar absichtlich leistungsfähige Prozessoren genutzt, um eine Virtualisierung zu betreiben, doch letztlich teilt sich die Software, die in einer virtuellen Maschine oder einem Container läuft, die Ressourcen mit den anderen Gästen (und deren Applikationen).<sup>276</sup>

[59] Nach all dem Gesagten, stellt sich höchstens die Frage, ob allfällige urheberrechtlich relevante Vervielfältigungen auf eine «quasi-dingliche» Tragweite der Klauseln hinweisen könnte. Generell wird jedoch angenommen, dass eine Anknüpfung an CPU sich nicht an der Anzahl der Vervielfältigungen orientiert.<sup>277</sup> Bezüglich den Mehrkernprozessoren sollten aber einige Besonderheiten in Betracht gezogen werden. Vorab kommt es bei Programmen, die nicht für den Parallelbetrieb geschrieben wurden, zu keinen zusätzlichen urheberrechtlichen Vervielfältigungen.<sup>278</sup> Ist die Software aber parallelisierbar, wird ihre Abarbeitung mittels Multi-Threading auf die verschiedenen mehrfädigen Cores aufgeteilt, was zu einer erhöhten Geschwindigkeit führen kann, aber nicht muss (je nach Programmierung der Applikation oder Unterstützung des Betriebssystems). Gegebenenfalls werden die Daten hier in den verschiedenen Puffer-Speicher (Prozessorcaches) der Cores zwischengespeichert, was auf eine Vervielfältigung hindeuten könnte, die aber gemäss GRÜTZMACHER fraglich ist, denn die Programmteile werden bereits vorab auf die Cores aufgeteilt und parallel abgearbeitet.<sup>279</sup> Geht man aber von einer Vervielfältigung aus, wäre diese m.E. sowieso, wenn nicht von Art. 12 Abs. 2 URG, sicherlich von Art. 24a URG gedeckt, da es sich bloss um eine der rechtmässigen Nutzung dienende, flüchtige Kopie ohne eigenständige wirtschaftliche Bedeutung handeln würde. Die Anzahl der urheberrechtlich relevanten Kopien erhöht sich aus meiner Sicht auch nicht durch die Anzahl mehrerer *virtueller Prozessoren* oder die Anzahl *Hardwarethreads*, die zu einem Container führen. Es geht immer noch um die Abarbeitung *einer* Programmkopie, die auf verschiedene Teile aufgeteilt wird. Ausserdem werden durch die Zusammenarbeit von v-CPU, v-Cores und Hardwarethreads auch keine neuen Verbraucherkreise erschlossen. Eine technisch-wirtschaftlich klar abgrenzbare Nutzungsform ist nicht ersichtlich.

[60] Beschränkungen durch CPU-, Core- und sonstiger nutzungsabhängiger Lizenzmetriken, ungeachtet der Anknüpfung an physikalische oder virtuelle Einheiten, kommt lediglich eine schuldrechtliche Wirkung zu und sie sind nicht urheberrechtlich durchsetzbar. Um von der gegebenenfalls höheren Leistungsfähigkeit zu profitieren, bleibt dem Hersteller also nur die Möglichkeit, die Höhe der Vergütung von der Anzahl der Prozessoren mit vertraglicher Wirkung abhängig

---

<sup>274</sup> Vgl. MARLY (Fn. 90), Rn. 1672.

<sup>275</sup> Vgl. zum Ganzen FRÖHLICH-BLEULER (Fn. 59), Rn. 1735 f.; GRÜTZMACHER (Fn. 120), Nutzungsarten 2, S. 699 f.

<sup>276</sup> Vgl. zum Ganzen, GRÜTZMACHER (Fn. 120), Nutzungsformen 2, S. 698.

<sup>277</sup> Vgl. GRÜTZMACHER (Fn. 120), Nutzungsformen 2, S. 698.

<sup>278</sup> FRÖHLICH-BLEULER (Fn. 59), Rn. 1736.

<sup>279</sup> Vgl. zum Ganzen GRÜTZMACHER (Fn. 120), Nutzungsformen 2, S. 699 f.

zu machen.<sup>280</sup> Weiter könnte man lizenzrechtliche Beschränkungen von blossen Vergütungsregelungen unterscheiden.<sup>281</sup> Im vorliegenden Fall ändert sich ungeachtet dieser Unterscheidung nichts an der bloss vertraglichen Rechtsnatur.

[61] Fraglich ist, ob bei CPU-Klauseln neben der urheberrechtlichen Unwirksamkeit auch ein Verstoss gegen den zwingenden Kerngehalt des gesetzlichen Gebrauchsrechts vorliegt, wonach entsprechende Bestimmungen nicht nur urheberrechtlich, sondern auch vertraglich nicht durchsetzbar wären. Klauseln, die auf Dauer den Wechsel der Software auf eine andere Hardwareeinheit verbieten, stellen ein Verbot der Installation zum bestimmungsgemässen Gebrauch dar. Schliessen solche Klauseln den Hardwarewechsel völlig aus oder machen ihn von der willkürlichen Zustimmung des Anbieters abhängig, greifen sie in den zwingenden Kern von Art. 12 Abs. 2 URG ein.<sup>282</sup> Die Ersetzung der Hardwareeinheit muss im Einzelfall erlaubt sein.<sup>283</sup> Den Hardwarewechsel von einer zusätzlichen Vergütung abhängig zu machen, ist zulässig.<sup>284</sup> Beschränkungen auf genau bestimmte Zentraleinheiten oder Prozessortypen sind im Zusammenhang mit container- oder VM-spezifischen Lizenzmetriken selten. Im Rahmen der Recherche wurden in casu keine Verstösse gegen den zwingenden Kerngehalt ausfindig gemacht. Der Wechsel auf leistungsfähigere Hardwareeinheiten ist in diesem Bereich kaum verboten, wird jedoch oft mit einer zusätzlichen Vergütung gekoppelt, was nicht gegen den zwingenden Kern verstösst.

[62] In Ausnahmefällen können AGB- oder kartellrechtliche Nichtigkeitsgründe vorliegen.<sup>285</sup> Individualvertraglich bereiten hardwarebezogene Lizenzbeschränkungen normalerweise keine Probleme.<sup>286</sup> Es kann aber bei einer allfälligen Regelung der Beschränkungen in AGB zur Korrektur ungewöhnlicher Nutzungsbeschränkungen kommen (Kap. 4.1.2.2.) kommen. Die schweizerischen Gerichte mussten bis anhin noch keine solchen Fälle im Bereich der Nutzungsbeschränkungen in Softwareüberlassungsverträgen beurteilen. Es fehlt an einer systematischen Aufarbeitung dieser Thematik. Hingegen hat sich im deutschen Recht eine detaillierte Lehre und Rechtsprechung zu AGB-rechtlich unwirksamen Nutzungsbeschränkungen herausgebildet.<sup>287</sup> Gemäss dieser AGB-Rechtsprechung werden CPU-Klauseln in Softwareveräusserungsverträgen oft als unwirksam qualifiziert.<sup>288</sup>

[63] Hardware-bindende Nutzungsbeschränkungen können ausserdem gegen das Wettbewerbsrecht verstossen, wenn unangemessene Bedingungen bei Konfigurationsänderungen durch einen marktmächtigen Partner erzwungen werden.<sup>289</sup> Missbräuchliches Verhalten kann bspw. bei der Bündelung von Leistungen und Produkten aus dem Angebot marktbeherrschender Hardware- oder Softwareanbieter vorliegen, da dadurch u.U. der Wettbewerb mit Drittanbietern beschränkt wird.<sup>290</sup> Ebenfalls unzulässig können durch marktbeherrschende Unternehmen festgesetzte un-

---

<sup>280</sup> Vgl. KOCH (Fn. 96), S. 916.

<sup>281</sup> GRÜTZMACHER (Fn. 251), Lizenzmetriken, S. 142.

<sup>282</sup> Zum Ganzen RAUBER (Fn. 41), Use Restrictions, S. 169; DERS. (Fn. 49), Computersoftware, S. 232.

<sup>283</sup> RAUBER (Fn. 49), Computersoftware, S. 232; FRÖHLICH-BLEULER (Fn. 59), Rn. 1730.

<sup>284</sup> RAUBER (Fn. 41), Use Restrictions, S. 169; DERS. (Fn. 49), Computersoftware, S. 232.

<sup>285</sup> RAUBER (Fn. 49), Computersoftware, S. 233.

<sup>286</sup> Vgl. MARLY (Fn. 90), Rn. 1675.

<sup>287</sup> Zum Ganzen RAUBER (Fn. 49), Computersoftware, S. 227.

<sup>288</sup> Etwa Urteil des OLG Frankfurt vom 10. März 1994 (6 U 18/93), CR 1994, S. 398.

<sup>289</sup> STRAUB (Fn. 44), Rn. 290.

<sup>290</sup> RAUBER (Fn. 49), Computersoftware, S. 233.

angemessen hohe Upgrade-Gebühren sein, die in keinem Verhältnis zur wirtschaftlichen Gegenleistung stehen.<sup>291</sup> Mögliche Folgen von missbräuchlichen Bestimmungen sind die Unwirksamkeit gemäss Art. 20 OR<sup>292</sup> sowie die Korrektur missbräuchlicher Konditionen nach Art. 7 Abs. 2 lit. c i.V.m. Art. 13 lit. b KG und kartellrechtliche Bussen nach Art. 49a ff. KG.<sup>293</sup>

## 6. Fazit

[64] Die Container-Technologie hat in den letzten Jahren herkömmliche Virtualisierungsansätze ergänzt oder teilweise sogar abgelöst. Aufgrund der zahlreichen Vorteile interessieren sich immer mehr Unternehmen für die «leichtgewichtige» und vermeintlich kostengünstige Virtualisierungslösung. Werden urheberrechtlich geschützte Programme in Containern betrieben, muss der Softwareerwerber deren Schutz weiter beachten. Auch ohne explizite Nutzungseinräumung darf er aber aufgrund des gesetzlichen Gebrauchsrechts nach Art. 12 Abs. 2 URG bei Software-Überlassungen auf Dauer ein Programmexemplar in einem Container (oder in einer virtuellen Maschine) betreiben, denn die hierfür vorgenommene gesetzlich garantierte Installation unterscheidet sich nicht entscheidend von derjenigen auf einem physischen Rechner. Der Betrieb auf mehreren Instanzen, der in einer Mehrfachnutzung resultiert, sowie nicht für den Gebrauch notwendige Modifikationen sind nicht mehr von Art. 12 Abs. 2 URG erfasst. Der mehrfache Containerbetrieb als Teil der Sicherheitsarchitektur ist zwar von Art. 24 Abs. 2 URG gedeckt, aber nur solange nicht mehrere Programmkopien produktiv im Einsatz sind, um tatsächlich eine Leistungssteigerung zu erreichen. In der Praxis beinhalten die Regelwerke der Softwareanbieter meistens vertragliche Bestimmungen bzw. Beschränkungen zu den entsprechenden Sachverhalten, dies oft in AGB oder EULA, bei welchen jeweils geprüft werden muss, ob sie überhaupt gültiger Vertragsbestandteil werden. Um die Mehrfachnutzung abzugelten, werden Container sowie virtuelle Maschinen vielfach einem physischen Rechner gleichgestellt und es wird nach der Anzahl der Instanzen lizenziert. Zudem knüpfen Softwareanbieter bei Nutzungsbeschränkungen oder Vergütungsansprüchen mittels CPU- oder Coreklauseln an die Leistungsfähigkeit der Hardware an. Umgesetzt wird dies mittels umfangreicher und teilweise verwirrender Regelwerke sowie schwer deutbarer Lizenzmetriken, die mitunter auf die Anzahl Prozessoren oder Cores des physischen Hosts abstellen. Container (oder auch virtuelle Maschinen) werden aber meist zwischen Servern hin- und hergeschoben, nur nach Bedarf gestartet und teilen sich untereinander die Ressourcen. Pro Instanz alle Prozessoren/Cores zu lizenzieren, stellt eine fragwürdige Anknüpfung dar. Insbesondere bei Containern, die nur einen Bruchteil der Ressourcen benötigen und von denen oft eine Vielzahl in Betrieb ist, führt dies zur Lizenzierung von tatsächlich nicht gebrauchten Ressourcen. Manche Softwareanbieter haben inzwischen eine granularere Lizenzierung auf kleinteiliger Ebene eingeführt, was zu begrüßen ist. Letztlich muss immer zwischen Bestimmungen mit rein vertraglicher Wirkung und Beschränkungen mit urheberrechtlicher Tragweite, die auch gegenüber Zweiterwerb durchsetzbar sind, differenziert werden. Je nachdem ob absolute oder relative Rechte verletzt werden, unterscheiden sich auch die Rechtsfolgen. Bei der Orientierung an der Anzahl Instanzen handelt es sich um urheberrechtlich wirkende

---

<sup>291</sup> Vgl. FRÖHLICH-BLEULER (Fn. 59), Rn. 301.

<sup>292</sup> STRAUB (Fn. 44), Rn. 345.

<sup>293</sup> Vgl. RAUBER (Fn. 41), Use Restrictions, S. 158; FRÖHLICH-BLEULER (Fn. 59), Rn. 281.

Regelungen, denn sie knüpfen in wirtschaftlich-technisch abgrenzbarer Weise an die Anzahl Vervielfältigungen an. Ein allgemeines Verbot der Nutzung in virtualisierten Systemen würde jedoch gegen den zwingenden Kern von Art. 12 Abs. 2 URG verstossen. Wird an die Leistungsfähigkeit der Hardware angeknüpft, handelt es sich nicht um eine wirtschaftlich-technisch eigenständige Nutzungsform. Durch die Zusammenarbeit der Prozessoren oder Cores im Verbund kommt es zu keinen zusätzlichen urheberrechtlich relevanten Vervielfältigungen, und zwar ungeachtet, ob es sich um virtuelle oder physische Einheiten handelt oder ob es sich um parallelisierbare oder nicht-parallelisierbare Software handelt. Vertragsbestimmungen, die in einem Verbot des Hardwarewechsels resultieren und allenfalls gegen den zwingenden Kerngehalt von Art. 12 Abs. 2 URG verstossen, sind im Zusammenhang mit der Container-Virtualisierung im Rahmen dieser Recherche keine aufgefunden und dementsprechend untersucht worden. Immerhin müssen solche vertraglichen Bestimmungen, sofern sie bei allfälligen AGB überhaupt Vertragsbestandteil geworden sind, dann auch noch den kartell- und lauterkeitsrechtlichen Rahmen einhalten.

---

NINA LOCHER, MLaw, Legal Trainee bei Scout24.

Vorliegender Beitrag basiert auf der im Herbstsemester 2020 verfassten Masterarbeit der Autorin. Ein besonderer Dank gebührt dem Betreuer dieser Masterarbeit, Herrn Dr. iur. Wolfgang Straub, für die immer gewährte Unterstützung und seinen fachlichen Rat.