

Juni 2021

9. Jahrg.

84364

Seite 57–120

# InTeR

Zeitschrift zum Innovations- und Technikrecht

# 2

## Herausgegeben von

Jürgen Ensthaler

Stefan Müller

Dagmar Gesmann-

Nuissl

## Herausgeberbeirat

Wilhelm-Albr. Achilles

Hans-Jürgen Ahrens

Udo di Fabio

Lars Funk

Thomas Klindt

Roman Reiss

Philipp Reusch

Franz Jürgen Säcker

Klaus Schülke

Christian Steinberger

Walther C. Zimmerli

Klaus J. Zink

## Schriftleitung

Lehrstuhl für

Wirtschafts-,

Unternehmens- und

Technikrecht an der

Technischen

Universität Berlin

## In Verbindung mit

VDI – Verein Deutscher Ingenieure e. V.

- Prof. Dr. Dr. Jürgen Ensthaler*
- 57 **Änderungen im Patentgesetz – Einbeziehung technischen Wissens in den Verletzungsprozess und Einschränkung des Unterlassungsanspruchs**
- Dr. Thomas M. Bossmeyer und Dr. Manuela Schlund*
- 58 **Produkt-Compliance**
- Dr. Gerhard Wiebe*
- 66 **IT-sicherheitsbezogene Pflichten von Herstellern smarter Produkte**
- Dr. Dr. Fabian Teichmann, LL.M. und Jan Koch*
- 70 **Wirtschaftsstrafrecht und Digitalisierung – Chancen und Herausforderungen**
- Jacob Schwartz, LL.M.*
- 77 **Betriebsgefahr und Unabwendbarkeit bei selbstfahrenden Fahrzeugen**
- Marco Deppe*
- 83 **Dynamische Typzuordnung von Elektroleichtfahrzeugen als Baustein multimodaler Mobilität**
- Dipl.-Jur. Marvin Gülker*
- 90 **Der urheberrechtliche Schutz von Schnittstellen, insbesondere von APIs – Teil 2**
- Prof. Dr. Dagmar Gesmann-Nuissl*
- 97 **Rechtsprechungsreport „Innovations- und Technikrecht“**
- 118 **InTeRessantes**

Typgenehmigung auch bei technischen Abweichungen gegenüber den EG-Typklassen hin, die eingangs vom KBA bewertet, und über die bei vorläufiger Genehmigung durch das KBA dann letztlich durch die EU-Kommission entschieden wird.<sup>27</sup> Dabei wären konkrete Vorgaben und Richtlinien zu prüfen, die diese Entscheidungsfindung anleiten würden. Ohne die Erlangung einer Ausnahmegenehmigung oder anderweitigen Sondererlaubnis sind de lege lata keine rechtlichen Möglichkeiten für eine verkehrswegübergreifende

Mobilität im Sinne der Vision des ‚Kamäleon‘ ersichtlich. Es verbleibt mit einer Einordnung in die EG-Typklasse L2e als reines Straßenfahrzeug ohne Nutzungsrechte von Fuß- und Radwegen. Gesetzgeberische Tätigkeit wäre erforderlich, um diese Vision multimodaler Mobilität im Sinne einer Verkehrswende realisieren zu können.

27 KBA, Merkblatt zu Genehmigungen für ‚Neue Techniken oder Konzepte‘ (MTK), 2011.

Dipl.-Jur. Marvin Gülker, Passau\*

## Der urheberrechtliche Schutz von Schnittstellen, insbesondere von APIs – Teil 2

*Schluss aus InTeR 2021, 27. Nachdem der erste Teil die technischen Grundlagen gelegt und festgestellt hat, dass klassische, quellcodebasierte APIs nicht mit den neueren Web-APIs vermischt werden dürfen, geht es im zweiten Teil um die urheberrechtliche Schutzfähigkeit dieser beiden speziellen Schnittstellenarten. Es wird gezeigt, dass klassische APIs als Computerprogramme schutzfähig sind, Web-APIs dagegen nicht. Als zentraler Prüfstein für die rechtliche Zulässigkeit der Übernahme geschützter APIs im Angesicht der Interoperabilität ist § 24 UrhG heranzuziehen und nicht die sog. „Merger Doctrine“. Das gilt auch nach der Urheberrechtsreform.*

### VI. APIs im Besonderen

Die Vielfalt der Schnittstellen setzt sich auch bei der Untergruppe der APIs fort. Wie die technischen Erörterungen zu Anfang gezeigt haben, muss zumindest zwischen klassischen und Web-APIs unterschieden werden. Wegen der ganz unterschiedlichen Funktionsweise können keine pauschalen Aussagen über „den“ Schutz von APIs getroffen werden.<sup>1</sup> Eine Einzelfallbetrachtung tut not, die nachfolgend zunächst für die klassischen APIs (unten 1.), sodann für die Web-APIs (unten 2.) durchgeführt werden soll.

#### 1. Klassische APIs

Streitgegenständlich im Java-Verfahren sind klassische, auf Quellcode beruhende APIs. Ihre Besonderheit gegenüber anderen Schnittstellenarten lässt sich am besten so zusammenfassen: ihre „Schnittstellenspezifikation“ liegt als Quellcode vor, den jeder, der Interoperabilität mit dieser Schnittstelle erreichen will, zwingend übernehmen muss. Das verkompliziert ihre Handhabung beträchtlich, denn die zwei von der CPRL verfolgten Ziele – Schutz von Computerprogrammen und Gewährleistung von Interoperabilität – geraten bei dieser Kategorie von APIs miteinander in Konflikt.

##### a) Klassische APIs als Teil des Computerprogramms

Dem wäre nicht so, wenn man klassischen APIs die Eigenart als (Teil eines) Computerprogramms absprechen könnte, indem man sie zu den nicht schutzfähigen Ideen und

Grundsätzen gem. § 69a Abs. 2 S. 2 UrhG zählte. Klassische APIs enthalten fast ausschließlich Deklarationen von Datenstrukturen, weshalb man auf den Gedanken verfallen kann, sie nicht im Sinne der Computerprogramm-Definition als steuernde Befehle einzustufen, sondern sie den Daten zuzuordnen.<sup>2</sup> In der Tat lösen derartige Deklarationen für sich genommen keine Effekte aus.<sup>3</sup> Jedoch sind die Deklarationen ein zwingender Bestandteil nach der formalen Grammatik der eingesetzten Programmiersprache und ohne sie könnte das Computerprogramm die eigentlichen Verarbeitungsbefehle nicht ausführen.<sup>4</sup> Dieser richtigen Aussage ist hinzuzufügen, dass die Deklarationen in ISO/IEC 2382:2015 Ziff. 2121372 ausdrücklich als Teil des Computerprogramms genannt werden. Nicht überzeugend ist dagegen das Argument, die Deklarationen steuern das Verhalten des Computerprogramms bei Typverletzungen,<sup>5</sup> denn in vielen Programmiersprachen werden derartige Fehler nicht zur Laufzeit des Computerprogramms, sondern noch beim Entwickler bei der Übersetzung von der Hoch- in die Maschinsprache (Kompilierung) durch das Übersetzungsprogramm (Compiler) überprüft. Dass der Compiler ein Computerprogramm ist, hat aber niemand bestritten.

Mit Blick auf das weitere Erfordernis des Vorliegens einer Ausdrucksform (§ 69a Abs. 2 S. 1 UrhG) hat *Marly* ausgeführt, APIs würden keine Vervielfältigung des Programms ermöglichen und seien deshalb nach den Vorgaben des EuGHs keine Ausdrucksform<sup>6</sup>. Das verkennt, dass klassische APIs stets als Quelltext vorliegen und ihre Kopie entsprechend sehr wohl eine Vervielfältigung eben dieses Programmteils darstellen kann. Demnach lässt sich den

\* Der Aufsatz basiert auf einer Seminararbeit des Verf., die er am Lehrstuhl für Bürgerliches Recht, Internationales Privatrecht und Rechtsvergleichung, Handels- und Wirtschaftsrecht (Prof. Dr. Renate Schaub, LL.M.) an der Ruhr-Universität Bochum geschrieben hat. Mehr über den Autor erfahren Sie auf Seite III.

1 So aber *Falkner*, in: Taeger (Hrsg.), Die Macht der Daten und der Algorithmen, 2019, S. 595, 599 ff.; *Marly*, Praxishandbuch Software-recht, 7. Aufl. 2018, Rn. 109.

2 Angedacht von *Nebel/Stiemerling*, CR 2016, 61, 62.

3 *Nebel/Stiemerling*, CR 2016, 61, 62.

4 *Nebel/Stiemerling*, CR 2016, 61, 63.

5 *Nebel/Stiemerling*, CR 2016, 61, 63.

6 *Marly*, GRUR 2012, 773, 779; *Marly* (Fn. 1), Rn. 101.

klassischen APIs ein urheberrechtlicher Schutz gem. § 69a UrhG nicht per se absprechen.

#### b) Kein enger Vervielfältigungsbegriff

Erwägen ließe sich aber, im Dienste der Interoperabilität den Begriff der „Vervielfältigung“ in § 69c Nr. 1 UrhG eng auszulegen und eine API-Kopie für Interoperabilitätsw Zwecke nicht darunter zu fassen. Der BGH vertritt nämlich die Meinung, nicht jeder technische Kopiervorgang bei Computerprogrammen sei auch eine „Vervielfältigung“<sup>7</sup>. Allerdings verbirgt sich hinter dieser Aussage eine Diskussion aus der Frühzeit der CPRL darüber, ob die temporäre Kopie des Computerprogramms im Arbeitsspeicher, die beim Laden eines Computerprogramms zwingend entsteht, als urheberrechtliche „Vervielfältigung“ gelten soll.<sup>8</sup> Man hatte vor diesem Hintergrund vertreten, dass der in § 69c Nr. 1 UrhG enthaltene Vervielfältigungsbegriff von demjenigen in § 16 UrhG verschieden sei.<sup>9</sup> Der BGH hat sich vor einiger Zeit für die Erfassung entschieden,<sup>10</sup> im Übrigen ist die Diskussion durch Änderung des § 16 UrhG gegenstandslos geworden, weil § 16 UrhG und § 69c Nr. 1 UrhG jetzt beide davon sprechen, dass „vorübergehende“ Handlungen Vervielfältigungen seien.<sup>11</sup> Der Streit sollte heute besser bei der Frage der Anwendbarkeit des § 44a Nr. 2 UrhG auf Computerprogramme geführt werden.<sup>12</sup> Im Sinne der Einheit der Rechtsordnung wird damit erreicht, dass der Begriff der „Vervielfältigung“ in §§ 16, 69c Nr. 1 UrhG den gleichen Bedeutungsgehalt hat. Davon sollte nicht wegen eines Problems, das mit der ursprünglichen Diskussion um die Kopie im Arbeitsspeicher nichts zu tun hat, wieder abgewichen werden.

#### c) Merger Doctrine und fehlende Schöpfungshöhe

Hauptsächlich wird die Interoperabilitätsfrage offenbar der Schöpfungshöhe zugeordnet.<sup>13</sup> Hier lassen sich zwei Aspekte unterscheiden: Zum einen kann man den bloß aus Deklarationen bestehenden klassischen APIs per se das Erreichen der Schöpfungshöhe absprechen (unten aa), oder man argumentiert mit der sog. „Merger Doctrine“ (unten bb).

#### aa) Allgemein fehlende Schöpfungshöhe

Forderungen, klassische APIs generell als nicht schutzfähige Banalität abzutun,<sup>14</sup> ist die Gefolgschaft zu verweigern. Da klassische APIs die Computerprogrammdefinition erfüllen, ist der Maßstab der Schöpfungshöhe kraft gesetzlicher Anordnung § 69a Abs. 3 UrhG und nicht § 2 Abs. 2 UrhG. § 69a Abs. 3 UrhG lässt zwar die genauen Kriterien der Beurteilung im Dunklen,<sup>15</sup> aber aus der Formulierung und der Entstehungsgeschichte ist zumindest klar, dass die hohen Anforderungen der früheren Rechtsprechung des BGH, die erheblich über dem Schaffen eines Durchschnittsprogrammierers liegende schöpferische Eigenheiten forderte,<sup>16</sup> abgesenkt werden sollten auf das Niveau der „kleinen Münze“.<sup>17</sup> Damit ist es unvereinbar, im Namen der Interoperabilität diese bewusste Entscheidung des Gesetzgebers durch den Richter revidieren zu lassen und eine besonders hohe Schöpfungshöhe zu fordern. Auch ist nicht ersichtlich, wie das Kriterium der Schöpfungshöhe das hiervon ganz unabhängige Ziel der Interoperabilität effektiv gewährleisten soll, denn egal wie hoch man diese Hürde ansetzt, wird sie doch von einigen APIs genommen werden. Für diese würde dann das Ziel der Interoperabilität nicht gelten. Das überzeugt mangels Anhaltspunkte für eine solche Wertung in der CPRL nicht; Interoperabilität muss unabhängig von der

Schöpfungshöhe gewährleistet sein. Dass dies keine akademischen Überlegungen sind, zeigt sich beim Blick in die technische Literatur. Dort wird das Design guter (klassischer) APIs ausdrücklich als schwierig bezeichnet.<sup>18</sup> Es sei einfach, APIs schlecht zu gestalten und deshalb gebe es viele schlechte, d.h. für Programmierer schwer zu benutzende APIs mit Kosten verursachenden Konsequenzen.<sup>19</sup> Jedenfalls bei klassischen APIs kann also keine Rede davon sein, es gebe technische Zwänge, die „einem Programmierer praktisch kaum einen kreativen Freiraum“<sup>20</sup> eröffnen. Man wird im Gegenteil nicht umhin kommen, gut durchdachten APIs grundsätzlich das Erreichen der Schöpfungshöhe auch dann bescheinigen zu müssen, wenn man diese hoch ansetzt.

#### bb) Merger Doctrine

Die Merger Doctrine ist eine dogmatische Figur des US-amerikanischen Urheberrechts, deren Anwendung zur Gewährleistung von Interoperabilität verschiedentlich vorgeschlagen wurde.<sup>21</sup> Sie wird in das europäische und deutsche Recht durch Verneinung der Schöpfungshöhe eingeführt.<sup>22</sup> Ob eine Übernahme der Merger Doctrine in das europäische Recht gewollt war, ist zwar streitig, richtigerweise aber zu bejahen. Art. 1 Abs. 3 des ursprünglichen Kommissionsentwurfs<sup>23</sup> sollte die Doktrin übernehmen.<sup>24</sup> In Ziff. 3.13 der dem Entwurf beigefügten Begründung wird das Konzept sogar ausdrücklich angesprochen. Die Gegenansicht, die sich auf spätere Ausführungen der Kommission<sup>25</sup> stützt<sup>26</sup>, übersieht, dass die Kommission an der genannten Stelle nicht über die Merger Doctrine spricht, sondern über die Abgrenzung von Idee und Ausdruck im All-

7 BGH, 3.2.2011 – I ZR 129/08, WRP 2011, 480, 482 – UsedSoft; BGH, 20.1.1994 – I ZR 267/91, WRP 1994, 299, 300 f. – Holzhandelsprogramm; BGH, 4.10.1990 – I ZR 139/89, BB-Beil. 18/1991, 2, 6 – Betriebssystem.

8 Zum Streit noch Redeker, IT-Recht, 7. Aufl. 2020, Rn. 47 ff.

9 Dreier, CR 1991, 577, 579; Lehmann, NJW 1991, 2112; Ohst, Computerprogramm und Datenbank, 2003, 21.

10 BGH, 3.2.2011 – I ZR 129/08, WRP 2011, 480, 482 – UsedSoft.

11 Vgl. Heerma, in: Wandtke/Bullinger (Hrsg.), UrhR, 5. Aufl. 2019, § 16 Rn. 18.

12 Vgl. einerseits Wiebe, in: Spindler/Schuster (Hrsg.), Recht der elektronischen Medien, 4. Aufl. 2019, § 44a Rn. 2, andererseits v. Welsler, in: Wandtke/Bullinger (Fn. 11), § 44a Rn. 23. Offengelassen in BGH, 3.2.2011 – I ZR 129/08, WRP 2011, 480, 482 – UsedSoft.

13 Beispielsweise bei Grützmaker, CR 2016, 485, 494; Dreier, CR 1991, 577, 583; Marly, GRUR 2012, 773, 779.

14 Grützmaker, CR 2016, 485, 494; Dreier, CR 1991, 577, 583. Beide aber pauschal für alle Schnittstellen (also auch klassische APIs).

15 Zuletzt hierzu Straker/Wehkamp, CR 2018, 699, 699 ff.; Hoeren/Wehkamp, CR 2018, 1, 1 ff.; früher etwa Lesshaft/Ulmer, CR 1993, 607, 608.

16 BGH, 9.5.1985 – I ZR 52/83, BGHZ 94, 276, 287 – Inkasso-Programm.

17 Grützmaker, in: Wandtke/Bullinger (Fn. 11), § 69a Rn. 35 m. w. N.; Lehmann, NJW 1991, 2112, 2113; BGH, 3.3.2005 – I ZR 111/02, WRP 2005, 1263, 1264 – Fash 2000.

18 Henning, acmqueue 4/5 (2007), 24, 25. Ebenso auch Hoeren/Wehkamp, CR 2018, 1, 6 f.

19 Henning, acmqueue 4/5 (2007), 24, 25 u. 29.

20 Lehmann, CR 1989, 1057, 1059.

21 Etwa bei Czarnota/Hart, Legal Protection of Computer Programs in Europe, 1991, 38; Schulte, CR 1992, 648, 650; Marly, GRUR 2012, 773, 778; Vinje, GRUR Int 1992, 250, 260. Vgl. auch KOM(1988) 816 endg., § 3.13.

22 Ohst (Fn. 9), 27 f.; Marly, GRUR 2012, 773, 778; Schulte, CR 1992, 648, 650; Dreier, CR 1991, 577, 583.

23 KOM(1988) 816 endg.

24 Czarnota/Hart (Fn. 21), 39.

25 SEK(1991) 87 endg., § 4.2.a.

26 Schulte, CR 1992, 648, 649. Widersprüchlich Wiebe, BB 1993, 1094, 1096, der einerseits von eigenem europäischem Computerurheberrecht ausgeht, andererseits aber die Merger Doctrine anwenden will.

gemeinen. Der EuGH hat die Merger Doctrine anerkannt,<sup>27</sup> wenn auch nur in Bezug auf graphische Benutzeroberflächen. Es steht zu erwarten, dass er für klassische APIs nicht anders entscheiden würde.

Nach der Merger Doctrine ist ein Ausdruck nicht schutzfähig, wenn es nur sehr wenige Möglichkeiten gibt, um eine Idee auszudrücken.<sup>28</sup> Idee und Ausdruck fallen in einem solchen Fall zusammen.<sup>29</sup> Das hat man bei der Übernahme klassischer APIs bisher mit der Begründung angenommen, dass der Wettbewerber zur Herstellung von Interoperabilität gezwungen ist, die Deklarationen des klassischen APIs 1:1 zu übernehmen.<sup>30</sup> Die fortschreitende Standardisierung auf gerade diese Schnittstelle sei ein entsprechender äußerer Faktor im Sinne der Merger Doctrine.<sup>31</sup> Dieser so auch in den USA geführten Argumentation ist der Court of Appeals for the Federal Circuit im Java-Verfahren überzeugend entgegengetreten.<sup>32</sup> Er hat ausgeführt, dass es für die Frage des Zusammenfallens von Idee und Ausdruck auf den Zeitpunkt der Schöpfung des Erstwerkes ankomme und nicht auf den Zeitpunkt der Übernahme.<sup>33</sup> Diese Aussage gilt uneingeschränkt auch für das europäische Recht. Auch hier ist die Frage der Schöpfungshöhe – an der man die Merger Doctrine ja festmacht – stets zum Schöpfungszeitpunkt zu beurteilen.<sup>34</sup> Andernfalls würde mit dem Auftreten eines Wettbewerbers *nachträglich* die Schöpfungshöhe eines Werkes entfallen. Solch eine Annahme ist mit der einheitlichen Dauer des Urheberrechts gem. § 64 UrhG unvereinbar. Daher kann mit dem Argument fortschreitender Standardisierung das Vorliegen einer Verschmelzung nicht begründet werden. Als das nun zu kopierende API erstmals geschaffen wurde, bestand ein entsprechender äußerer Zwang gerade noch nicht, denn zum (*de-facto*-)Standard hat es sich erst später entwickelt.<sup>35</sup> Nimmt man statt eines *de-facto*-Standards ein normiertes klassisches API an (z.B. POSIX.1-2008 oder den C-Standard ISO/IEC 9899:2011, soweit er die Standardbibliothek betrifft), ändert sich an dieser Beurteilung nichts, denn Erstschöpfung ist in diesem Fall dann die Norm, die ihrerseits keinen äußeren Zwängen unterlag (sie bildet sie ja erst selbst).

Damit ist nicht gesagt, dass der Merger Doctrine für APIs im speziellen oder gar Schnittstellen im Allgemeinen keine Bedeutung mehr zukäme; das stünde ja auch im Widerspruch zu den Annahmen der Kommission. Die Merger Doctrine spielt dort ihre Rolle, wo eine Schnittstelle (klassische APIs eingeschlossen) schon zum Zeitpunkt der Erstschöpfung von äußeren Zwängen bestimmt ist. Denkbar ist das etwa bei der oft erheblichen Speicherknappheit eingebetteter Systeme: so mag es eine Hardware-Architektur unmöglich machen, Funktionen mit mehr als zwei Parametern aufzurufen. In einem solchen System kann kein Schutz dafür verlangt werden, dass alle Funktionen nur maximal zwei Parameter besitzen. Idee und Ausdruck sind eins. Bei den meisten klassischen APIs, auch bei denen des Java-Verfahrens, liegt der Fall so aber nicht. Ob man eine Funktion, die die größte mehrerer Zahlen ermittelt, nun „Math.max“, „Math.maximum“ oder „Arith.larger“ nennt, kann der Erstschöpfer frei entscheiden,<sup>36</sup> ebenso wie die Frage, ob er nur zwei oder noch mehr Parameter für die Funktion zulässt. Dass die vielen Gestaltungsmöglichkeiten für klassische APIs sogar die Gefahr bergen, für Programmierer schwer handhabbare Software zu erstellen,<sup>37</sup> wurde bereits dargelegt. Weiter ist zu beachten, dass dem hier gewählten Beispiel isoliert tatsächlich keine Schöpfungs-

höhe zukommen mag (da beim Kopieren von Teilen eines Werks der kopierte Teil für sich genommen schutzfähig sein muss,<sup>38</sup> was auch bei Computerprogrammen nicht anders sein kann,<sup>39</sup> und die Benennung dieser Funktion als „max“ für jeden Programmierer die naheliegende erste Wahl wäre), es jedoch die Gesamtheit der Anordnung und Kombination aller für sich genommen banalen Elemente ist, die urheberrechtlichen Schutz genießt.<sup>40</sup>

Dieser Befund wird bestätigt durch die Erwägung, dass der Zweck, aus dem ein Werk geschaffen wird, für das Vorliegen der Schöpfungshöhe seit jeher als unbeachtlich eingestuft wird.<sup>41</sup> Daher kann die Absicht, Interoperabilität herzustellen, bei der Feststellung der Schöpfungshöhe im Rahmen der Merger Doctrine nicht berücksichtigt werden. Es zeigt sich damit, dass die Merger Doctrine, obwohl anwendbar, für mit klassischen APIs interoperablen Computerprogrammen in aller Regel von wenig Nutzen sein dürfte.

#### d) Schranken des Urheberrechts

Im Java-Streit hat der Court of Appeals für eine Lösung auf Schrankenebene in Form der „Fair-Use“-Schranke des US-Rechts votiert.<sup>42</sup> Das europäische oder deutsche Urheberrecht kennt eine solche generalklauselhafte Schranke des Urheberrechts nicht. Von den in § 69d UrhG vorgesehenen speziellen Schrankenregelungen für das Urheberrecht an Computerprogrammen greift für die API-Übernahme keine ein. Es stellt sich daher die Frage, ob nicht eine Lösung auf

27 EuGH, 22.12.2010 – C-393/09, K&R 2011, 105, Rn. 49 f. – BSA/Kulturministerium; folgend *Marly*, GRUR 2012, 773, 778.

28 Fed. Cir., 9.5.2014 – 2013-1021, GRUR Int 2014, 711, 716 – Oracle v. Google; *Wiebe*, BB 1993, 1094, 1096; *Czarnota/Hart* (Fn. 21), 38.

29 EuGH, 22.12.2010 – C-393/09, K&R 2011, 105, Rn. 49 – BSA/Kulturministerium; KOM(1988) 816 endg., § 3.13.

30 *Czarnota/Hart* (Fn. 21), 38; *Vinje*, GRUR Int 1992, 250, 259 f.; vorsichtiger *Schulte*, CR 1992, 648, 650.

31 *Marly*, GRUR 2011, 204, 208; *Marly*, GRUR 2012, 773, 779; *Marly* (Fn. 6), Rn. 101; *Spindler*, in: Schricker/Loewenheim (Begr.), UrhR, 6. Aufl. 2020, § 69a Rn. 13; *Lehmann*, in: Lehmann (Hrsg.), Die Europäische Richtlinie über den Schutz von Computerprogrammen, 2. Aufl. 1993, Kap. I.A, Rn. 7.

32 Das Urteil des US Supreme Court, 5.4.2021 – 18-956 – Google v. Oracle, erging nach Abfassung dieses Aufsatzes und konnte keine Berücksichtigung mehr finden. Es beschränkt sich allerdings auch auf und bejaht die Anwendung der in Deutschland nicht relevanten „Fair-Use“-Schranke. Beachte aber das abl. Sondervotum von Richter *Thomas*, das dem hier vertretenen Ansatz nahesteht und ein besseres technisches Verständnis erkennen lässt als die Mehrheitsmeinung.

33 Fed. Cir., 9.5.2014 – 2013-1021, GRUR Int 2014, 711, 716 – Oracle v. Google.

34 *Schulze*, in: Dreier/Schulze (Hrsg.), UrhG, 6. Aufl. 2018, § 2 Rn. 35; BGH, 27.5.1981 – I ZR 102/79, GRUR 1981, 820, 822 – Stahlrohrstuhl II.

35 Vgl. Fed. Cir., 9.5.2014 – 2013-1021, GRUR Int 2014, 711, 719 – Oracle v. Google.

36 Fed. Cir., 9.5.2014 – 2013-1021, GRUR Int 2014, 711, 716 – Oracle v. Google.

37 *Henning*, *acmqueue* 4/5 (2007), 24, 25 ff.; *Straker/Wehkamp*, CR 2018, 699, 702; vgl. auch Fed. Cir., 9.5.2014 – 2013-1021, GRUR Int 2014, 711, 715 – Oracle v. Google.

38 EuGH, 16.7.2009 – C-5/08, K&R 2009, 707, Rn. 38 f. – Infopaq/DDF.

39 BGH, 20.9.2012 – I ZR 90/09, WRP 2013, 808, Rn. 30 – UniBasic-IDOS; *Dreier*, in: Dreier/Schulze (Fn. 34), § 69a Rn. 23; *Schippel*, MMR 2020, 445, 446.

40 Für das US-amerik. Urheberrecht Fed. Cir., 9.5.2014 – 2013-1021, GRUR Int 2014, 711, 716 f. – Oracle v. Google; für das europ. Recht EuGH, 16.7.2009 – C-5/08, K&R 2009, 707, Rn. 45 ff. – Infopaq/DDF; speziell für Computerprogramme BGH, 20.9.2012 – I ZR 90/09, WRP 2013, 808, Rn. 30 – UniBasic-IDOS.

41 BGH, 27.2.1961 – I ZR 127/59, GRUR 1961, 635, 638 – Stahlrohrstuhl; *Bullinger*, in: Wandtke/Bullinger (Fn. 11), § 2 Rn. 29. Beide m. w. N. bis zum RG.

42 Fed. Cir., 9.5.2014 – 2013-1021, GRUR Int 2014, 711, 718 – Oracle v. Google.

Ebene der allgemeinen Schrankenregelungen der §§ 44a ff. UrhG möglich ist.

Die Anwendbarkeit der §§ 44a ff. UrhG auf Computerprogramme wird vor allem mit dem Argument bestritten, dass §§ 69d, 69 e UrhG abschließende *leges speciales* seien, was man wiederum neben der systematischen Stellung in einem eigenen Abschnitt auch aus Erwägungsgrund 19 CPRL a.F. (= Erwägungsgrund 15 CPRL n.F.) entnehmen könne.<sup>43</sup> Die systematische Auslegung ist aber angesichts der Existenz von § 69a Abs. 4 UrhG, der ausdrücklich den Zugriff auf allgemeine Regelungen gestattet, nicht so klar wie vorgebracht. Erwägungsgrund 19 a.F. = 15 n.F. gibt für die Frage ebenfalls nichts her, denn im Gegenteil kann man Erwägungsgrund 28 a.F. = 19 n.F. entnehmen, dass die Mitgliedsstaaten, solange sie sich im Rahmen der RBÜ halten, eigenen Spielraum genießen.<sup>44</sup> Die Äußerung der Kommission, die Einfügung des Kompromissvorschlages zum Reverse Engineering habe „deutlich gemacht, dass die Handlungen der Vervielfältigung und Übersetzung nicht für andere mehr allgemeine Zwecke wie Untersuchung, Forschung, oder private Verwendung [...] vorgenommen werden dürfen“,<sup>45</sup> spricht allerdings angesichts des Ausschlusses „allgemeine[r] Zwecke“ für eine Verdrängung.<sup>46</sup> Diese Äußerung fiel aber in engem Zusammenhang mit der Regelung zum Reverse Engineering (Dekompilierung) und kann besonders wegen des Gebrauchs des Begriffs der „Übersetzung“ auch so verstanden werden, dass nur die Ausnahme für Dekompilierung nicht für andere als die in ihr genannten Zwecken herangezogen werden darf. Über eine Sperre für Ausnahmen zu anderen Verwertungsrechten als Vervielfältigung und Übersetzung schließlich sagt die Äußerung gar nichts aus, was eher für als gegen eine Anwendbarkeit der allgemeineren Schrankenregelungen spricht.

Deshalb ist nicht von einer Totalsperre der §§ 44a ff. UrhG auszugehen. Vielmehr ist nach Sinn und Zweck der jeweils betroffenen Regelungen zu fragen.<sup>47</sup> Unmittelbar einleuchtend ist etwa, dass wegen der vollständig identischen Interessenlage im Vergleich zu „normalen“ Schutzgegenständen die Schranke des § 45 UrhG anwendbar sein muss.<sup>48</sup> Selbst wenn man den Ausführungen der Kommission entscheidende Bedeutung beimisst, wird man diese Vorschrift deshalb zumindest analog anwenden müssen. Ebenso klar ist, dass wegen der detaillierten Regelung des privaten Bereichs in § 69d Abs. 1 und Abs. 2 UrhG die Schranke des § 53 UrhG vollständig verdrängt wird.<sup>49</sup>

Die Übernahme eines klassischen APIs zwecks Interoperabilität könnte allenfalls unter das Zitatrecht aus § 51 UrhG fallen. Dessen Anwendbarkeit auf Computerprogramme ist umstritten. Man hat sie mit der nicht näher begründeten Aussage verneint, § 51 UrhG passe „seinem ganzen Sinn und Zweck nach nicht für Computerprogramme“<sup>50</sup>. Das verengt die Perspektive auf die des Endnutzers. Programmierer haben andere Bedürfnisse. Sinn und Zweck des § 51 UrhG ist es, die (kritische) Diskussion über Schutzgegenstände zu ermöglichen, damit über die Auseinandersetzung kultureller Fortschritt möglich wird.<sup>51</sup> Das gilt auch für Computerprogramme. Es muss möglich sein, schlechten Programmierstil anzuprangern, Beispiele guten und schlechten Code-Designs zu diskutieren und in diesem Zusammenhang Verbesserungsvorschläge zu machen. Das passiert auch praktisch.<sup>52</sup> § 51 UrhG ist deshalb auf Computerprogramme anwendbar. Nur: die Übernahme von klassischen APIs verfolgt keinen diskursiven Zweck. Sie

ist rein wirtschaftlich orientiert, weshalb sie von § 51 UrhG nicht erfasst wird.

#### e) Freie Benutzung

Nach hiesiger Auffassung ist die Übernahme zum Zwecke der Interoperabilität an § 24 UrhG (freie Benutzung) zu messen. Die Norm ist systematisch getrennt von den Schrankenregelungen untergebracht (im vierten und nicht wie die Schrankenregelungen im sechsten Abschnitt des ersten Teils des UrhG), sodass man selbst dann von seiner Anwendbarkeit auf Computerprogramme ausgehen muss, wenn man im Übrigen meint, dass die §§ 69d, 69 e UrhG abschließendes *lex specialis* zu §§ 44a ff. UrhG seien. Entsprechend wird die Anwendbarkeit von § 24 UrhG auf Computerprogramme allgemein für möglich gehalten.<sup>53</sup> Auch das Europarecht steht seiner Anwendung nicht entgegen, denn die CPRL regelt das Übernahmeproblem nicht ausdrücklich und überlässt es den Mitgliedsstaaten, wie sie das Ziel der Interoperabilität in diesen Fällen erreichen wollen.<sup>54</sup>

§ 24 UrhG fordert ein selbstständiges Werk, das in freier Benutzung des Werkes eines anderen geschaffen worden ist. Das setzt zunächst in aller Regel das Erreichen der Schöpfungshöhe durch das neue Werk voraus,<sup>55</sup> die bei Computerprogrammen aufgrund von § 69a Abs. 3 UrhG ausdrücklich auf das Niveau der „kleinen Münze“ reduziert ist und die folglich von den allermeisten Computerprogrammen erfüllt werden wird. Mit Blick auf den Java-Streit wird man sicher sagen können, dass ein großes und umfangreiches Computerprogramm wie Dalvik (Androids Java-Implementierung) diese Hürde nimmt. Maßgeblich für die Einstufung als frei ist, ob das neue Werk hinreichende Distanz zum alten Werk einhält, sodass dieses dahinter „verblasst“.<sup>56</sup> Die bestimmenden, eigenpersönlichen Züge

43 *Haberstumpf*, in: Lehmann (Hrsg.), Der urheberrechtliche Schutz von Computerprogrammen, 2. Aufl. 1993, Kap. II, Rn. 148.

44 *Dreier*, GRUR 1993, 781, 784; *Grützmaker*, in: Wandtke/Bullinger (Fn. 11), § 69a Rn. 84.

45 KOM(1990) 509 endg., Begründung S. 4 – insoweit nicht in ABl. C Nr. 320 v. 20.12.1990, 22 ff. abgedruckt.

46 *Haberstumpf*, in: Lehmann (Fn. 43), Kap. II, Rn. 148; *Czarnota/Hart* (Fn. 21), 31.

47 *Dreier*, in: *Dreier/Schulze* (Fn. 34), § 69a Rn. 32; *Spindler*, in: *Schricker/Loewenheim* (Fn. 31), § 69a Rn. 25i.

48 *Dreier*, in: *Dreier/Schulze* (Fn. 34), § 69a Rn. 34; *Spindler*, in: *Schricker/Loewenheim* (Fn. 31), § 69a Rn. 25i.

49 Amtl. Begr. BT-Drs. 12/4022, 8 f.; *Dreier*, GRUR 1993, 781, 784; *Spindler*, in: *Schricker/Loewenheim* (Fn. 31), § 69a Rn. 25i.

50 *Grützmaker*, in: Wandtke/Bullinger (Fn. 11), § 69a Rn. 85.

51 *Dreier*, in: *Dreier/Schulze* (Fn. 34), § 51 Rn. 1; *Spindler*, in: *Schricker/Loewenheim* (Fn. 31), § 51 Rn. 10.

52 Der gesamte Aufsatz von *Henning*, *acmqueue* 4/5 (2007), 24 verfolgt genau diesen Zweck und nutzt dafür auch Code-Zitate. Es gibt keinen Grund, ihm für den Fall, dass der von ihm zitierte Code geschützt ist, die Berufung auf § 51 UrhG zu versagen.

53 *Dreier*, in: *Dreier/Schulze* (Fn. 34), § 69a Rn. 34; *Spindler*, in: *Schricker/Loewenheim* (Fn. 31), § 69a Rn. 24c; *Czychowski*, in: *Fromm/Nordemann* (Begr.), UrhR, 12. Aufl. 2018, § 69a Rn. 40 a.E.; *Haberstumpf*, in: Lehmann (Fn. 43), Kap. II, Rn. 141.

54 *Vinje*, GRUR Int 1992, 250, 259.

55 *Bullinger*, in: Wandtke/Bullinger (Fn. 11), § 24 Rn. 2; *Schulze*, in: *Dreier/Schulze* (Fn. 34), § 24 Rn. 5; BGH, 23.6.1961 – IZR 105/59, GRUR 1961, 631, 632 – Fernsprehbuch; anders bei Parodien wegen europarechtlichen Einflusses, s. BGH, 28.7.2016 – IZR 9/15, WRP 2016, 1260, Rn. 28 – auf fett getrimmt. Vgl. zu letzterem auch *Bullinger*, in: Wandtke/Bullinger (Fn. 11), § 24 Rn. 2.

56 St. Rspr., s. nur BGH, 20.3.2003 – IZR 117/00, WRP 2003, 1235, 1237 – Gies-Adler; BGH, 11.3.1993 – IZR 264/91, GRUR 1994, 191, 193 – Asterix-Persiflagen; BGH, 21.11.1980 – IZR 106/78, GRUR 1981, 352, 353 – Staatsexamensarbeit. Für die Literatur *Schulze*, in: *Dreier/Schulze* (Fn. 34), § 24 Rn. 8.

des benutzten Werks müssen zurücktreten.<sup>57</sup> Dabei kann man auch auf den Umfang der Übernahme zurückgreifen.<sup>58</sup> Bei Computerprogrammen machen die klassischen APIs nur einen kleinen Teil des Gesamtprogramms aus. Im Java-Streit war von etwa 3 % die Rede.<sup>59</sup> Derart kleine Anteile am Code sind ungeeignet, das Gesamtprogramm zu prägen; in einem anderen Fall ging der BGH noch bei 5 % geänderten Codes sogar davon aus, dass nicht einmal eine Bearbeitung, sondern noch eine Vervielfältigung vorlag.<sup>60</sup> Vielmehr zeigt sich die Eigentümlichkeit eines Computerprogramms vor allem in der Implementation, also der konkreten Formulierung von Anweisungen, der Konstruktion von Programmverzweigungen, der Anordnung, Reihenfolge und Benennung von Variablen, den Einsatz von Unterprogrammen und deren Anordnung und ähnliches, wozu insbesondere auch die Begehung von Programmierfehlern („Bugs“) gehört.<sup>61</sup> Die Übernahme bloß der „äußeren Fassade“, die ein klassisches API bildet, ist deshalb nicht geeignet, Eigentümlichkeiten des benutzten Computerprogramms nachzubilden.

§ 24 UrhG fordert wegen der vollständigen Freistellung von der Zustimmung des Ersturhebers einen strengen, am Zweck der Vorschrift – Verwendung geschützter Werke als Anregung im Dienste der kulturellen Interessen, da niemand „im luftleeren Raum“ arbeitet<sup>62</sup> – orientierten Beurteilungsmaßstab.<sup>63</sup> Zu diesen kulturellen Interessen ist bei Computerprogrammen aufgrund der Wertungen der CPRL die Interoperabilität zwischen Computerprogrammen zu zählen, sodass eine Benutzung (nur) dann „frei“ im Sinne von § 24 UrhG sein kann, wenn sie auch das Ziel der Interoperabilität verfolgt. § 24 UrhG soll es dem Zweiturheber gerade nicht ermöglichen, sich schlicht eigenen Aufwand zu ersparen.<sup>64</sup> Verfolgt er indes das Ziel der Interoperabilität, muss auch die unveränderte Übernahme des klassischen APIs von § 24 UrhG erfasst werden, da technisch bedingt eine Interoperabilität anders nicht hergestellt werden kann. An dieser Hürde muss der Versuch, § 24 UrhG auf Googles Übernahme der Java-APIs anzuwenden, scheitern, denn Google verfolgte gerade nicht das Ziel der Interoperabilität, sondern wollte lediglich die Bekanntheit von Java ausnutzen, um auf diese Weise an Java gewöhnte Programmierer zur Programmierung für Dalvik zu motivieren.<sup>65</sup> Bequemlichkeit gehört weder zu den Zielen von § 24 UrhG noch zu denen der CPRL.

Der zur Zeit der Abfassung dieses Aufsatzes noch im Gesetzgebungsverfahren befindliche Entwurf eines Gesetzes zur Anpassung des Urheberrechts an die Erfordernisse des digitalen Binnenmarktes<sup>66</sup> sieht in Art. 1 Nr. 5 vor, § 24 UrhG aufzuheben. Nach jüngster Rechtsprechung des EuGHs<sup>67</sup> ist für diesen eigenartigen Hybrid aus Schutzbereichsbestimmung und Schranke im Anwendungsbereich der InfoSocRL<sup>68</sup> kein Raum mehr.<sup>69</sup> Die bisherige Rechtsprechung zur „inneren“ Distanz zu einem Werk<sup>70</sup> bei Parodien und künstlerischer Auseinandersetzung soll in eine eigenständige Schranke ausgelagert werden,<sup>71</sup> die aber für Computerprogramme unabhängig vom Streit um die Anwendbarkeit von Schrankenbestimmungen vorliegend ohnehin nicht in Betracht kommt. Bei der Herstellung von Interoperabilität zwischen Computerprogrammen geht es wie dargestellt um einen Fall „äußerer“ Distanz zum benutzten Werk und damit um die Schutzbereichsbegrenzung. Es ist daher erfreulich, dass die Schutzbereichsbegrenzung nicht wie teils gefordert<sup>72</sup> gänzlich verworfen,

sondern vielmehr inhaltsgleich nach § 23 UrhG verschoben werden soll.<sup>73</sup> Da gemäß Art. 1 Abs. 2 Buchst. a InfoSocRL die Regelungen der CPRL unberührt bleiben, ändert sich für die Beurteilung der Übernahme klassischer APIs zum Zwecke der Herstellung der von der CPRL gewollten Interoperabilität zwischen Computerprogrammen auch dann nichts, wenn der Gesetzentwurf so beschlossen werden sollte. Man darf nicht blind dafür sein, dass nach wie vor Urheberrecht außerhalb der InfoSocRL existiert.

## 2. Web-APIs

### a) Kommunikationsprotokolle als Ideen und Grundsätze

Da es sich bei den Web-APIs um einen Spezialfall der Kommunikationsprotokolle handelt, richtet sich ihre Beurteilung nach derjenigen für die Kommunikationsprotokolle. Anhand der oben angestellten teleologischen Erwägungen ist daher auch bei dieser Kategorie von APIs zu prüfen, was der Programmierer eines interoperablen Computerprogramms bei der Nutzung von Web-APIs benötigt und ob ein Zielkonflikt vorliegt. Da der Datenaustausch über ein Netzwerk erfolgt, ist ein Zugriff auf den das Kommunikationsprotokoll (Web-API) umsetzenden Quellcode des Zielprogramms nicht nötig. Es genügt, das für den Datenaustausch genutzte Kommunikationsprotokoll zu kennen, d. h. welche Zeichenfolgen in welcher Reihenfolge zu welchem Zeitpunkt zu übermitteln sind. In diesem Fall ist demnach der diese Regeln umsetzende Code als Ausdruck und das Kommunikationsprotokoll als Idee zu begreifen, wodurch die beiden Ziele der CPRL zugleich vollständig erfüllt werden können. Dieses Ergebnis ist auch interessengerecht,

57 Schulze, in: Dreier/Schulze (Fn. 34), § 24 Rn. 8; BGH, 11.3.1993 – IZR 264/91, GRUR 1994, 191, 193 – Asterix-Persiflagen; BGH, 26.3.1971 – IZR 77/69, GRUR 1971, 588, 589 – Disney-Parodie.

58 Vgl. BGH, 11.3.1993 – IZR 263/91, BGHZ 122, 53, 60 – Alcolix.

59 Fed. Cir., 9.5.2014 – 2013-1021, GRUR Int 2014, 711, 714, das Eingangsgericht zitierend, spricht davon, Google hätte 97 % des Codes selbst geschrieben, der Rest seien die unverändert übernommenen API-Deklarationen gewesen – Oracle v. Google.

60 BGH, 26.10.1989 – IZR 216/87, CR 1990, 188, 189 – Programmübernahme.

61 Vgl. die gut verständliche Auflistung bei OLG Frankfurt a.M., 6.11.1984 – 14 U 188/81, GRUR 1985, 1049, 1051 – Baustatikprogramm. Ähnl. Grützmacher, in: Wandtke/Bullinger (Fn. 11), § 69c Rn. 14 und Straker/Wehkamp, CR 2018, 699, 702 ff.

62 Schulze, in: Dreier/Schulze (Fn. 34), § 24 Rn. 1.

63 BGH, 26.9.1980 – IZR 17/78, GRUR 1981, 267, 269 – Dirlada; BGH, 11.3.1993 – IZR 263/91, BGHZ 122, 53, 60 – Alcolix; Bullinger, in: Wandtke/Bullinger (Fn. 11), § 24 Rn. 9; Schulze, in: Dreier/Schulze (Fn. 34), § 24 Rn. 9.

64 BGH, 26.9.1980 – IZR 17/78, GRUR 1981, 267, 269 – Dirlada; Bullinger, in: Wandtke/Bullinger (Fn. 11), § 24 Rn. 12; Schulze, in: Dreier/Schulze (Fn. 34), § 24 Rn. 9.

65 Fed. Cir., 9.5.2014 – 2013-1021, GRUR Int 2014, 711, 719 – Oracle v. Google.

66 BR-Drs. 142/21 = BT-Drs. 19/27426 (letztere derzeit nur als Vorabfassung verfügbar).

67 EuGH, 29.7.2019 – C-476/17, WRP 2019, 1156, Rn. 57 ff. – Metall auf Metall III.

68 Richtlinie 2001/29/EG des Europäischen Parlaments und des Rates zur Harmonisierung bestimmter Aspekte des Urheberrechts und der verwandten Schutzrechte in der Informationsgesellschaft (ABL. L Nr. 167 v. 22.6.2001, S. 10).

69 Ohly, GRUR 2017, 964, 967.

70 Zu Parodien BGH, 28.7.2016 – IZR 9/15, WRP 2016, 1260, Rn. 24 – auf fett getrimmt; BGH, 11.3.1993 – IZR 263/91, BGHZ 122, 53, 60 f. – Alcolix; BGH, 11.3.1993 – IZR 264/91, GRUR 1994, 191, 193 – Asterix-Persiflagen. Zur Kunstfreiheit beim sog. Sampling BVerfG, 31.5.2016 – 1 BvR 1585/13, WRP 2016, 822, Rn. 76 ff. – Metall auf Metall.

71 BR-Drs. 142/21, S. 54 und S. 83 = BT-Drs. 19/27426.

72 Ohly, GRUR 2017, 964, 968 f.

73 BR-Drs. 142/21, 54 und 83 = BT-Drs. 19/27426.

denn auf diese Weise verbleibt dem Hersteller des ursprünglichen Computerprogramms sein Urheberrecht hieran und Konkurrenz wird trotzdem (vorbehaltlich eines Schutzes auch des Kommunikationsprotokolls, dazu unten b) möglich. Keine der beiden Interessengruppen wird somit übervorteilt. Man könnte zwar einwenden, dass schon in den Entwurf des Protokolls, d.h. in den Ablauf und die Reihenfolge der Kommunikation erheblicher Forschungsaufwand geflossen sein kann, doch würde man damit letztlich ein Ausschließlichkeitsrecht am Lösungsweg fordern. Dieser wird in der Informatik als Algorithmus bezeichnet<sup>74</sup> und ist nach überwiegender Meinung nicht urheberrechtlich schutzfähig,<sup>75</sup> sodass dieses Argument ins Leere gehen muss.<sup>76</sup>

Allerdings wollte im europäischen Gesetzgebungsverfahren der Rechtsausschuss genau eine solche Zuordnung in den Erwägungsgründen festschreiben,<sup>77</sup> was vom Parlament aber abgelehnt wurde. Bei dieser Ablehnung handelte es sich indessen nicht um inhaltliche Opposition. Man muss sich diesbezüglich vor Augen halten, dass zu diesem Zeitpunkt im Gesetzgebungsverfahren (2. Lesung im europäischen Parlament) ein Richtlinienvorschlag vorlag, der einen unter langwierigen Querelen ausgehandelten Kompromiss an anderer Stelle, nämlich bei der Regelung über die Dekompilierung (in Deutschland heute § 69e UrhG) enthielt, den man bei Annahme von Änderungsanträgen unweigerlich wieder aufgeschnürt hätte, was niemand wollte.<sup>78</sup> Auch keine inhaltliche Opposition lag in dem Argument, technisches Vokabular wie dieses gehöre nicht in Gesetzestexte.<sup>79</sup> Der Rechtsausschuss selbst begriff alle eingebrachten Anträge ohnehin nur als „Klarstellungen“<sup>80</sup> ohne inhaltliche Änderungen gegenüber den bereits angenommenen Anträgen aus erster Lesung. Schon der ursprüngliche Kommissionsentwurf ließ eine entsprechende Trennung von Kommunikationsprotokoll und Implementation erkennen, als er angab, zur Herstellung von Interoperabilität sei eine Wiedergabe des ursprünglichen Codes nicht nötig.<sup>81</sup> Aus der Gesetzgebungsgeschichte wird man daher eine Schutzfähigkeit der Kommunikationsprotokolle als Computerprogramm nicht herleiten können. Die weitere Rechtspraxis bestätigt diesen Befund: Als die Kommission gegen Microsoft wegen dessen marktbeherrschender Position im Bereich Arbeitsgruppenserver vorging, verpflichtete sie Microsoft ausdrücklich nicht etwa zur Offenlegung von Quellcode, sondern zur Herausgabe von Informationen zum genutzten Kommunikationsprotokoll „Active Directory“.<sup>82</sup>

#### b) Schutzfähigkeit der Kommunikationsprotokolle außerhalb von §§ 69a ff. UrhG

Kommunikationsprotokolle unterfallen damit als „Schnittstellenspezifikation“ den nicht schutzfähigen Ideen und Grundsätzen eines Computerprogramms gem. § 69a Abs. 2 S. 2 UrhG. Das schließt logisch aus, dass sie zugleich selbst Schutz als Computerprogramm genießen können, denn ein solches Ergebnis verstieße gegen den Satz vom ausgeschlossenen Dritten. Sie erfüllen auch, um eine positive Begründung anzuführen, weder die Definition des Computerprogramms (denn es handelt sich bestenfalls um eine Sammlung, nicht aber eine Folge von Befehlen; und ohne richtige Umsetzung in Quellcode kann man einen Computer nicht zu einer Ausführung bewegen) noch die Anforderungen des EuGH an eine Ausdrucksform<sup>83</sup> (weder handelt es

sich um Quell-, noch um Objekt- oder vergleichbaren Code). Im Übrigen hat der EuGH diese Trennung zwischen Implementation und „Schnittstellenspezifikation“ für die mit den Kommunikationsprotokollen verwandten Dateiformate (es geht in beiden Fällen um Datenaustausch) bereits bestätigt.<sup>84</sup> Es ist nur konsequent, diesen Gedanken auch auf die Kommunikationsprotokolle und Web-APIs zu erstrecken.

Der EuGH hat allerdings angedeutet, dass er einen Schutz der Dateiformate und damit wohl auch der mit ihnen verwandten Kommunikationsprotokolle als anderes Werk (also nicht als Computerprogramm) für möglich hält.<sup>85</sup> Demnach bestünde grundsätzlich etwa die Möglichkeit, Protokollbestandteile wie die Wörter „GET“ und „POST“ in HTTP urheberrechtlich über § 2 Abs. 1 Nr. 1 UrhG als Sprachwerk zu schützen. Eine derartige Einordnung erscheint auf den ersten Blick unintuitiv. Immerhin enthält dieser Ansatz notwendigerweise die Aussage, dass die Protokolle bei der Betrachtung von § 69a UrhG den gem. § 69a Abs. 2 S. 2 UrhG ungeschützten Ideen unterfallen, bei der Betrachtung des allgemeineren § 2 UrhG dagegen einen schutzfähigen Ausdruck darstellen. Das kann man für widersprüchlich halten.<sup>86</sup> Zwingend ist dieser Schluss aber nicht, denn § 69a UrhG, insb. § 69a Abs. 2 S. 2 UrhG, ist *lex specialis* zu § 2 UrhG und kann daher inhaltlich von dieser Norm abweichen.

Nicht erklärt hat sich der EuGH zu der Frage, welche Auswirkungen dieser „gespaltene“ Schutz der in einem Dateiformat oder Kommunikationsprotokoll enthaltenen Werke haben soll. Er hat stattdessen darauf verwiesen, dass derartige Werke die Schöpfungshöhe zu erfüllen hätten.<sup>87</sup> Weil der Schutz von Kommunikationsprotokollen und ihren Elementen das Potential hat, das erklärte Ziel der CPRL, Interoperabilität zu ermöglichen, in erheblichem Maße zu gefährden, muss man in diesem Fall ein deutlich über die „kleine Münze“ hinausgehendes kreatives Schaffen fordern.<sup>88</sup> § 69a Abs. 3 UrhG steht dem nicht entgegen, denn es geht bei den Kommunikationsprotokollen gerade nicht um den Schutz für Computerprogramme nach den

74 Pomberger/Dobler, Algorithmen und Datenstrukturen, 2008, 33.

75 Grützmaker, in: Wandtke/Bullinger (Fn. 11), § 69a Rn. 29; Ohst (Fn. 9), 36 ff.; Dreier, in: Dreier/Schulze (Fn. 34), § 69a Rn. 22; Spindler, in: Schrickler/Loewenheim (Fn. 31), § 69a Rn. 12a f. Unklar Czyszowski, in: Fromm/Nordemann (Fn. 53), § 69a Rn. 30.

76 Zum patentrechtlichen Schutz von Computerprogrammen und Algorithmen ausführlich Gülker, Der Jurist 2018, 143.

77 EP-Sitzungsdokument PE 146.106 endg., S. 4 Änderungsantrag 1.

78 Vgl. die Äußerung von Kommissions-Vizepräsident Bangemann, EP-Plenarprot. 3-404 v. 16.4.1991, 68 f.

79 Redebeitrag des Abgeordneten Cox, EP-Plenarprot. Nr. 3-404 v. 16.4.1991, 66.

80 Redebeitrag von Berichterstatterin Salema, EP-Plenarprot. Nr. 3-404 v. 16.4.1991, 62 f.

81 KOM(1988) 816 endg., § 3.11.

82 Kommissionsentscheidung COMP/C-3/37.792 = 2007/53/EG, zit. nach ABl. L Nr. 32 v. 6.2.2007, S. 23, Ziff. 18, 30 ff.

83 EuGH, 22.12.2010 – C-393/09, K&R 2011, 105, Rn. 34 f. – BSA/Kulturministerium; EuGH, 2.5.2012 – C-406/10, WRP 2012, 802, Rn. 38 – SAS Institute.

84 EuGH, 2.5.2012 – C-406/10, WRP 2012, 802, Rn. 41 ff. – SAS Institute.

85 EuGH, 2.5.2012 – C-406/10, WRP 2012, 802, Rn. 45 – SAS Institute.

86 So Grützmaker, in: Wandtke/Bullinger (Fn. 11), § 69a Rn. 31, freilich ohne die hier gegebene Begründung.

87 EuGH, 2.5.2012 – C-406/10, WRP 2012, 802, Rn. 45 – SAS Institute.

88 Grützmaker, CR 2016, 485, 494 unter dem Begriff der Monopolisierung von Ideen.



§§ 69a ff. UrhG, sondern um allgemeinen Schutz gem. § 2 UrhG.

Gesteigerte Anforderungen an die Schöpfungshöhe genügen aber nicht, um das Ziel der Interoperabilität in jedem Falle zu erreichen, denn häufig wird es vorkommen, dass auch diese gesteigerten Anforderungen erfüllt werden. Ohne weitere Kriterien würde es unmöglich, den betreffenden Teil der Kommunikationsprotokolls (Web-APIs) zu replizieren und damit Interoperabilität herzustellen. Man hat in diesem Kontext das provokante Beispiel in die Diskussion eingebracht, bei dem ganz bewusst ein nach § 2 UrhG unstrittig geschütztes Werk (das könnte z.B. ein kunstvoll arrangiertes Foto sein) als Bestandteil eines Web-APIs (oder auch allgemeiner eines Kommunikationsprotokolls) genutzt wird, um so faktisch urheberrechtlichen Schutz auch für das Web-API bzw. Kommunikationsprotokoll selbst zu erlangen, weil dieses Werk nicht ohne Zustimmung des Rechteinhabers vervielfältigt werden dürfe.<sup>89</sup> Da hinter einem solchen, technisch – anders als es bei den klassischen APIs der Fall ist – nicht indizierten Vorgehen die kaum verhüllte Absicht steht, das von der CPRL vorgegebene Interessengleichgewicht zwischen Rechteinhabern und Wettbewerbern gezielt einseitig zulasten der letzteren zu verschieben, kann dem der Einwand des Rechtsmissbrauchs entgegen gehalten werden (§ 242 BGB). Nicht immer wird das aber so deutlich anzunehmen sein wie in diesem Fall. Dazu muss man sich nur vergegenwärtigen, dass die Rechtsprechung schon isolierten Wortfolgen wie „esoterische Räuberpistole“ die erforderliche Schöpfungshöhe zuerkannt hat.<sup>90</sup> Es lässt sich zumindest nicht ausschließen, dass damit auch Protokollelemente wie der unter Technikern beliebte, als Teil eines Kommunikationsprotokolls namens HTCPCP veröffentlichte Aprilscherz „418 I'm a teapot“<sup>91</sup> als hinreichend kreativ angesehen würden. Missbrauchsabsichten wird man den Gestaltern von HTCPCP kaum vorwerfen können.

Ohnehin ist es nicht ausgemacht, dass der EuGH sich der Forderung nach einer über der „kleinen Münze“ liegenden Schöpfungshöhe anschließen würde. Folgt man dem vom EuGH erwogenen Ansatz des „gespaltenen“ Schutzes für Werke in Dateiformaten und Kommunikationsprotokollen, dann sollte man auch bei den Kommunikationsprotokollen beachten, dass die Schöpfungshöhe kein geeignetes Kriterium zur Gewährleistung von Interoperabilität ist. Ähnlich wie bei den klassischen APIs dürfte es sich am ehesten anbieten, mithilfe von § 24 UrhG zu einer freien Benutzung des geschützten Werks zu gelangen. Während man allerdings bei den klassischen APIs darauf zurückgreifen kann, dass diese ja Teil des Computerprogramms sind und diese ihre schöpferische Eigentümlichkeit vor allem aus den konkreten Ablaufanweisungen beziehen, ist ein solcher Rückgriff bei den Kommunikationsprotokollen und Web-APIs gerade nicht möglich. Prägende Züge dieser Schutzgegenstände könnten z.B. auch besonders kreative Befehlsnamen sein, deren Übernahme eine Anwendung des § 24 UrhG eigentlich ausschließt. Das europarechtlich gebotene Ziel der Gewährleistung von Interoperabilität wird man aber nur erreichen können, wenn man § 24 UrhG (bzw. seine gemäß dem o.g. Gesetzentwurf zu erwartende Nachfolge-regelung in § 23 UrhG n.F.) richtlinienkonform dahingehend auslegt, dass er die Übernahme gestatte. Mit dem Wortlaut der Norm ist das jedenfalls vereinbar.

### c) Schutz der Protokolldokumentation

Unberührt von alledem bleibt die Beurteilung der mit der juristisch als „Schnittstellenspezifikation“ bezeichneten Funktionsweise nicht zu verwechselnden Protokollspezifikation oder besser -dokumentation. Dabei handelt es sich um ein für Menschen (Programmierer) geschriebenes Dokument, das die Funktionsweise des Protokolls erläutert. Beispielsweise erklärt der Standard RFC 2616 die Funktionsweise des Protokolls HTTP. Dokumente wie diese sind als Sprachwerk gem. § 2 Abs. 1 Nr. 1 UrhG zweifellos geschützt.<sup>92</sup>

## VII. Ergebnis

Die Untersuchung hat ergeben, dass die CPRL zwei Ziele verfolgt: den urheberrechtlichen Schutz von Computerprogrammen und die Gewährleistung von Interoperabilität zwischen Computerprogrammen verschiedener Hersteller. Die Frage, welche Teile einer Schnittstelle als Computerprogramm schutzfähig sind und welche den gem. § 69a Abs. 2 S. 2 UrhG ungeschützten Ideen und Grundsätzen (sog. „Schnittstellenspezifikation“) unterfallen, ist anhand dieser beiden Ziele für jeden Schnittstellentyp einzeln unter Heranziehung der nach wie vor zutreffenden Computerprogramm-Definition zu beantworten. Bei der Diskussion ist auf sprachliche Genauigkeit zu achten, da in technischer Hinsicht „klassische“, d. h. quellcodebasierte APIs ganz anders funktionieren als die neueren, auf Kommunikationsprotokollen basierenden Web-APIs. Eine weitere sprachliche Hürde liegt darin, dass der Begriff der „Schnittstelle“ im Gesetz anders als der technische Schnittstellenbegriff die Implementation meint.

Die beiden Ziele der CPRL können im Einzelfall miteinander in Konflikt geraten, was namentlich bei den klassischen APIs der Fall ist. In diesen Fällen ist § 24 UrhG heranzuziehen, dessen für die vorliegende Untersuchung maßgeblicher Gehalt als Schutzbereichsbegrenzung nach Abschluss der Urheberrechtsreform aller Voraussicht nach zukünftig unverändert in § 23 UrhG enthalten sein wird. Die Norm deckt die Übernahme allerdings nur dann, wenn sie auch der Interoperabilität dient. Kommunikationsprotokolle und Web-APIs sind keine Computerprogramme, nach der kritikwürdigen Meinung des EuGHs allerdings als sonstige Werke durchaus schutzfähig. Erreichen sie die Schöpfungshöhe, muss in richtlinienkonformer Auslegung erneut über § 24 UrhG die Herstellung von Interoperabilität gewährleistet werden. Stets gem. § 2 Abs. 1 Nr. 1 UrhG schutzfähig ist eine etwaige an Programmierer gerichtete Schnittstellendokumentation wie z.B. RFC 2616; die in ihr enthaltenen Ideen und Grundsätze sind mit denjenigen einer Implementation des Kommunikationsprotokolls identisch.

89 Falker, in: Taeger (Fn. 1), 603 f. Nicht richtig ist seine Annahme, eine Schutzfähigkeit folge aus der Rechtsprechung des EuGHs zu sog. komplexen Werken. Denn komplexe Werke sind solche, die, wie z.B. Computerspiele, neben einem Computerprogramm noch ein anderes Werk enthalten, EuGH, 19.12.2019 – C-263/18, WRP 2020, 185, Rn. 59 – Tom Kabinet; EuGH, 23.1.2014 – C-355/12, WRP 2014, 301, Rn. 23 f. – Nintendo/PC Box und 9Net. Kommunikationsprotokolle und Web-APIs sind aber wie dargestellt überhaupt keine Computerprogramme.

90 OLG Frankfurt a.M., 1.11.2011 – 11 U 75/06, NJOZ 2012, 848, 850 – Perlentauher II.

91 RFC 2324, § 2.3.2 – Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0).

92 Czarnota/Hart (Fn. 21), 38.