

Alesch Staehelin

Aktuelle Entwicklungen im IT-Recht

Beim folgenden Beitrag handelt es sich um eine angepasste Version des am 23. Juni 2022 anlässlich der INGRES-Tagung «Praxis des Immaterialgüterrechts in der Schweiz 2022» gehaltenen Referats.¹ Wer den ersten Teil des Aufsatzes liest, wird u.a. erfahren, was die Frage nach der urheberrechtlichen Schützbarkeit von Softwareschnittstellen (Application Programming Interfaces, APIs) mit dem Abtreibungsverbot in den USA zu tun hat.

L'article qui suit est une version adaptée de l'exposé² présenté le 23 juin 2022 à l'occasion du colloque de l'INGRES «Pratique en matière de droit de la propriété intellectuelle en Suisse 2022». En lisant la première partie de l'article, vous apprendrez notamment quel est le lien entre la question de la protection des interfaces de programmation d'applications (API) par le droit d'auteur et l'interdiction de l'avortement aux États-Unis.

I. Google LLC v. Oracle America, Inc. – wie wäre das in der Schweiz?

1. Worum ging es in dem Rechtsstreit zwischen Oracle und Google?
2. Der Supreme Court und die «Fair Use Doctrine»
3. Der 4-Faktoren-Test der Fair-Use-Doktrin
4. Supreme Court: «Fair Use» vorliegend gegeben
5. Die Kernfrage liess der Supreme Court leider offen
6. Interessanter Nebenfakt
7. Wie ist die Rechtslage in der Schweiz?
8. Die Rechtslage in Deutschland und gemäss der Rechtsprechung des EuGH

II. Judikatur: Softwareschutz & Urheberrecht

1. «Schüttgefässwaage 1 II» (BGer vom 2. Dezember 2019)
2. «Schüttgefässwaage 2 I» (HGer SG vom 24. Januar 2020)
3. «Schüttgefässwaage 2 II» (BGer vom 22. September 2020)

I. Google LLC v. Oracle America, Inc. – wie wäre das in der Schweiz?

Kommen wir gleich zum wohl spektakulärsten und wichtigsten IT-Gerichtsfall der letzten 15 bis 20 Jahre: Darf man eigentlich Softwareschnittstellen (Application Programming Interfaces, APIs) kopieren?

Im Bereich der Softwaretechnik und -entwicklung ist es weit verbreitet, von anderen geschriebene Softwareschnittstellen zu verwenden. In der Schweiz, in Deutschland, in vielen anderen europäischen Ländern und in den USA ist – auch durch eine gefestigte Rechtsprechung – allgemein anerkannt, dass Software-Quellcode als Ganzes meist urheberrechtlich geschützt ist.

Die Frage, inwieweit Softwareschnittstellen bzw. API-Code kopiert werden dürfen, blieb aber bisher von den Gerichten unbeantwortet.

Der Oberste Gerichtshof der USA, der US Supreme Court, hat in seiner Entscheidung «Google LLC v. Oracle America, Inc.» vom 5. April 2021³ ein wenig Klarheit darüber geschaffen, in welchem Umfang der zur Erstellung einer Softwareschnittstelle verwendete Deklarationscode kopiert werden kann.

1. Worum ging es in dem Rechtsstreit zwischen Oracle und Google?

Java ist eine von Oracle entwickelte Computerprogrammiersprache, die als Teil der Java SE-Plattform verwendet wird. Oracle ist die Inhaberin des Urheberrechts an Java SE.

2005 erwarb Google Android und kopierte, um Programmierern die Arbeit mit Java zu ermöglichen, 37 Java-API-Codestellen (d.h. rund 11'500 Zeilen Code). Die Java-API von Oracle wird indes nicht nur von Google, sondern weltweit verbreitet von Programmierern genutzt. Oracle warf Google vor, dass dieses Kopieren von Computer-Code unberechtigt erfolge und das Verwenden der urheberrechtlich geschützten Java-SE-Computerplattform entsprechend unzulässig sei.

Oracle verklagte daraufhin Google wegen des Kopierens dieses API-Codes mit dem Argument, dass Google damit das Urheberrecht von Oracle verletze. Google wiederum argumentierte, dass ein solches Kopieren eine faire Nutzung (sog. «Fair Use») darstelle und es somit von der Beachtung des Urheberrechts befreit sei.

ALESCH STAHELIN, Dr. iur., LL.M. (UCLA), Rechtsanwalt, Zürich.

- 1 Die Teile über die Totalrevision des Schweizerischen Datenschutzgesetzes und über aktuelle gesetzgeberische Aktivitäten im Bereich der Künstlichen Intelligenz (KI) in der Schweiz und in der EU sind an dieser Stelle vollständig weggelassen.
- 2 Les parties concernant la révision totale de la loi suisse sur la protection des données et les activités législatives actuelles dans le domaine de l'intelligence artificielle (IA) en Suisse et dans l'UE ont été entièrement omises ici.
- 3 <www.supremecourt.gov/opinions/20pdf/18-956_d18f.pdf>, alle Links zuletzt abgerufen am 1. August 2023.

Nach einer Reihe gerichtlicher Auseinandersetzungen befand die untere Instanz, dass sowohl der API-Code als auch die Struktur von Oracle urheberrechtlich geschützt werden könnten und dass die Verwendung des API-Codes durch Google keine faire Nutzung darstelle.

2. Der Supreme Court und die «Fair Use Doctrine»

Schliesslich hatte das höchste US-amerikanische Gericht, der Supreme Court, zu beurteilen, ob die Nutzung von Oracles Java-API-Codes durch Google eine Urheberrechtsverletzung darstellt und ob allenfalls die «Fair Use Doctrine» die Nutzung rechtfertigt.

Der Oberste Gerichtshof der USA hob am Ende das Urteil der unteren Instanz auf. Bei den von Google kopierten API-Codezeilen handle es sich um Teile von Benutzerschnittstellen. Durch diese APIs können Programmierer vorgefertigte Rechenaufgaben zur Verwendung in ihren eigenen Programmen aufrufen. Da Computerprogramme also primär eine Funktion erfüllen, sei es schwierig, traditionelle Urheberrechtskonzepte für diese Technologien anzuwenden. Justice STEPHEN GERALD BREYER, der für den Gerichtshof schrieb, wies darauf hin, dass die Fair-Use-Doktrin insbesondere bei der Computerprogrammierung die Funktionalität des fraglichen Werks berücksichtigen müsse. «Fair Use» biete also ein Korrektiv gegen die potenzielle Übermonopolisierung durch die Gewährung von Urheberrechten für Computerprogramme. Aus prozessökonomischen Gründen ging der Supreme Court hypothetisch von einem Szenario aus, in dem der fragliche API-Code urheberrechtlich geschützt ist, und widmete sich sodann eingehend dem 4-Faktoren-Test der Fair-Use-Doktrin:

3. Der 4-Faktoren-Test der Fair-Use-Doktrin

a) Art des Werks

Beim Prüfen des ersten der vier Fair-Use-Faktoren, geht es um die Art des Werks. Gemäss dem Supreme Court ist ein API zwar ein Code, aber kein Computerprogramm, da ein API keine Aufgabe ausführt. Als Teil einer Schnittstelle sind die von Google kopierten Zeilen naturgemäss mit Ideen betreffend die Gesamtorganisation bzw. Struktur der API verbunden. Dies, so das Gericht, spreche für eine faire Nutzung. Ideen seien urheberrechtlich nicht geschützt – erst die Konkretisierung der Idee, also deren Ausdruck, sei urheberrechtlich schützbar.

b) Zweck und Charakter bzw. die Art der Nutzung

Beim zweiten Faktor befasste sich der Supreme Court mit dem Zweck und Charakter bzw. der Art der Nutzung. Das Gericht befand, dass das Kopieren vorliegend keinen Selbstzweck darstelle. Die APIs sollten lediglich genutzt werden, um einen neuen kreativen Ausdruck, nämlich den von Google unabhängig geschriebenen, für die Entwicklung der Android-Plattform benötigten Code, zu schaffen. Der

Supreme Court stufte somit den Zweck des Werks als umgestaltend ein.

c) Menge und Umfang sowie Wesentlichkeit des verwendeten Teils im Verhältnis zum urheberrechtlich geschützten Werk

Zum dritten Faktor, mit dem die Menge und der Umfang sowie die Wesentlichkeit des verwendeten Teils im Verhältnis zum urheberrechtlich geschützten Werk beurteilt werden, hielt der Supreme Court fest, dass die 11'500 kopierten Zeilen bloss 0.4% des API ausmachen. Die Menge des kopierten Codes ist folglich bloss ein kleiner Teil eines wesentlich grösseren Ganzen – weniger als 3% der Gesamtmenge des Codes in der Java-API, die sich auf 2,86 Mio. Zeilen belief.

d) Auswirkungen der Nutzung auf dem potenziellen Markt

Der vierte und letzte Fair-Use-Faktor analysiert die Auswirkungen der Nutzung auf dem potenziellen Markt. Die neue Android-Smartphone-Plattform von Google stelle gemäss dem Supreme Court keinen Marktersatz für Java SE dar. Oracle würde sogar von der Neuimplementierung ihrer Schnittstelle profitieren.

4. Supreme Court: «Fair Use» vorliegend gegeben

Das Gericht verneinte also das Vorliegen einer Urheberrechtsverletzung auf der Basis der Fair-Use-Doktrin und fügte an, dass ein Durchsetzen des Urheberrechts auf der Grundlage dieser Tatsachen die Gefahr bergen würde, dass der Öffentlichkeit «kreativitätsbezogene Schäden» entstünden.

Aus der Kombination der vier Fair-Use-Faktoren ergab sich für das Gericht, dass Google sich mit dem Kopieren des API-Codes durchaus im Rahmen der Fair-Use-Doktrin – und damit in zulässiger Weise – bewegte. Der Oberste Gerichtshof der USA wies darauf hin, dass diese Entscheidung keine Änderung gegenüber seinen anderen Urteilen zur Fair-Use-Lehre darstelle. Vielmehr biete diese einen Rahmen und eine Analyseanleitung für die besonderen und einzigartigen Fragen und Umstände, die sich ergeben, wenn ein urheberrechtlich geschützter Computercode von anderen Entwicklern und Unternehmen verwendet wird.

5. Die Kernfrage liess der Supreme Court leider offen

Wichtig ist, dass das Gericht nicht entschieden hat, ob der deklarierende Code für sich allein urheberrechtsfähig ist. Diese Frage wurde offengelassen. Entschieden wurde bloss, dass im Zusammenhang mit Google das Verwenden eines kleinen Prozentsatzes des Codes zur Erstellung der Android-API zulässig war. Das Gerichts fällt die Entscheidung mit einer Quote von 6:2.

Ein Durchsetzen von Oracles Urheberrechten hätte sicherlich Konsequenzen auf globaler Ebene gehabt, denn es wird vermutet, dass rund 70% der Smartphones davon be-

troffen gewesen wären: Dies hätte unweigerlich zu enormen Schadenersatzklagen geführt.

6. Interessanter Nebenfact

Die «Dissenting Opinion» stammt von Justice CLARENCE THOMAS, dem erzkonservativen Abtreibungsgegner (der auch im April 2023 wieder in die Schlagzeilen geriet). Justice THOMAS war der Meinung, dass Oracles API-Code urheberrechtlich geschützt sei und drei der vier Fair-Use-Faktoren gegen Google sprächen, insbesondere seien sowohl qualitativ wie auch quantitativ in erheblichem Mass Code-Teile kopiert worden. Ihn dürfte wohl das Prinzip «Thou shalt not steal» aus den zehn Geboten beeinflusst haben.

7. Wie ist die Rechtslage in der Schweiz?

Das Thema ist aktuell, APIs sind in unserer digitalisierten Welt überall anzutreffen. Ein eigentliches Pendant zur Fair-Use-Doktrin gibt es in der Schweiz nicht. Ein Gericht könnte sich also nicht (wie der US Supreme Court) um die Beantwortung der Kernfrage, ob APIs urheberrechtlich schützenswert sind, drücken, um dann sogleich die kopierende Partei zu entlasten, indem dieser faire Nutzung («Fair Use») attestiert wird.

- a) Grundsatz auch bei Computerprogrammen: Es braucht Gestaltungsspielraum, Individualität und Schöpfungshöhe

Software-Schnittstellen können in der Schweiz grundsätzlich selbständig urheberrechtlich schützenswert sein, soweit man den Programmierern einen gewissen Gestaltungsspielraum bei deren Schaffung zugesteht.⁴ Für den Urheberrechtsschutz braucht es also auch bei einer Schnittstelle eine gewisse Individualität in der Kreation und die nötige Schöpfungshöhe.

In diesem Zusammenhang lohnt sich ein Blick auf die Anforderungen für den Urheberrechtsschutz bei Computerprogrammen: Ungeachtet eines bestehenden Gestaltungsspielraums fehlt einem Computerprogramm dann die notwendige Individualität, wenn es als banal bzw. trivial zu qualifizieren ist,⁵ weil sein Inhalt eine blosser Aneinanderreihung von bekanntem, zum Gemeingut gehörendem Material darstellt, oder wenn es vollständig auf rein alltäglicher, standardisierter Programmierarbeit beruht.⁶

Was bereits bei einem Computerprogramm gilt, muss daher umso mehr für eine Schnittstelle gelten: Massgebend für die Feststellung der Werkqualität einer Schnittstelle ist folglich allein, ob sie eine ausreichende schöpferische Individualität aufweist oder nicht. Prinzipiell wird diese Voraussetzung bei Schnittstellen in geringerem Masse als bei Computerprogrammen gegeben sein, weil der Gestaltungsspielraum eines Programmierers aufgrund der Vorgaben zur jeweiligen Datenübermittlung der einzelnen Computerprogramm-Komponenten noch enger ist. Zudem werden für den Datentransfer vielfach bestimmte Standards durch Re-

gulierungsorganisationen, Branchenverbände oder die in diesem Bereich tätigen Unternehmen festgelegt, wodurch das Geltendmachen eines Urheberrechts durch einen einzelnen Hersteller einer Computerprogrammkomponente ausscheidet.⁷

Was bräuchte es denn, damit eine Softwareschnittstelle in der Schweiz überhaupt urheberrechtlich geschützt sein könnte? Die notwendige Voraussetzung für einen urheberrechtlichen Schutz von Schnittstellen besteht in jedem Fall erstmal darin, dass die jeweilige Schnittstelle gemäss Art. 2 URG auch als geschütztes Werk zu qualifizieren ist.

Um diese Möglichkeit auszuloten, bietet sich zunächst einmal ein Blick darauf an, wann denn ein Computerprogramm Urheberrechtsschutz genießt und wann nicht:

Voraussetzung für die Anerkennung von Computerprogrammen als urheberrechtliches Werk bildet gemäss Art. 2 Abs. 1 URG das Vorhandensein einer geistigen Schöpfung mit individuellem Charakter.⁸

Geschützt ist ein Werk, wenn es sich als individuelle Schöpfung von den tatsächlichen oder natürlichen Vorbedingungen im Rahmen der Zweckbestimmung abhebt.⁹ Eine geistige Schöpfung liegt dann vor, wenn das Werk die Äusserung einer gedanklichen Tätigkeit eines Menschen darstellt und damit auf einer zumindest geringen geistigen Leistung beruht.¹⁰ Der individuelle Charakter eines Werks spiegelt sich darin wider, dass ein Dritter bei gleicher Aufgabenstellung nicht das gleiche oder im Wesentlichen das gleiche Werk schaffen wird, und grenzt sich gegenüber der Banalität oder einer routinemässigen Arbeit ab.¹¹

Die Schutzfähigkeit von Computerprogrammen hängt somit davon ab, ob dem Programmierer angesichts der Aufgabenstellung und der übrigen Rahmenbedingungen ein genügender Spielraum für eine persönliche Gestaltung in Auswahl, Sammlung, Anordnung und Einteilung der Informationen und Befehle zur Verfügung stand und ob er diesen Spielraum auch entsprechend genutzt hat.¹²

Ungeachtet eines bestehenden Gestaltungsspielraums fehlt einem Computerprogramm allerdings dann die not-

4 E. NEFF/M. ARN, Urheberrechtlicher Schutz der Software, in: R. von Büren/L. David (Hg.), Schweizerisches Immaterialgüter- und Wettbewerbsrecht (SIWR), II/2, Urheberrecht im EDV-Bereich, Basel 1998, 146.

5 Vgl. Botschaft URG, 523.

6 NEFF/ARN (Fn. 4), 132.

7 BVGer vom 18. Dezember 2018, B-831/2011, E. 576 (<https://jurispub.admin.ch/publiw/pub/cache.jsf?displayName=B-831/2011&decisionDate=2018-12-18>). Dieser Entscheid wurde im Übrigen in BGER vom 2. November 2022, 2C_596/2019 bestätigt.

8 Vgl. NEFF/ARN (Fn. 4), 131; I. CHERPILLOD, in: B. K. Müller/R. Oertli (Hg.), Stämpfli's Handkommentar zum Urheberrechtsgesetz (URG), 2. Aufl., Bern 2012, URG 2 N 1; G. RAUBER, Computersoftware, in: M. Streuli-Youssef (Hg.), Urhebervertragsrecht, Zürich 2006, 124.

9 Vgl. BGE 143 III 373 ff. E. 2.1; BGE 125 III 328 ff. E. 4b.

10 Vgl. BGE 130 III 168 ff. E. 4.5; D. BARRELET/W. EGLÖFF, Das neue Urheberrecht, 4. Aufl. Bern 2020, URG 2 N 5; CHERPILLOD (Fn. 8), URG 2 N 9.

11 Vgl. BGE 143 III 373 ff. E. 2.1; BGE 134 III 166 ff. E. 2.3.1; CHERPILLOD (Fn. 8), URG 2 N 20.

12 Vgl. NEFF/ARN (Fn. 4), 132; F. H. THOMANN, Grundriss des Software-schutzes, Zürich 1992, 30.

wendige Individualität, wenn es als banal bzw. trivial zu qualifizieren ist,¹³ weil sein Inhalt eine bloss Aneinanderreihung bekannten, zum Gemeingut gehörenden Materials darstellt, oder wenn es vollständig auf rein alltäglicher, standardisierter Programmierarbeit beruht.¹⁴

Im Zusammenhang mit der Banalität bzw. der Trivialität ist darauf hinzuweisen, dass vieles, was sog. «statistisch einmalig» ist, dennoch trivial bzw. banal ist – und deswegen keinen Urheberrechtsschutz genießt. Hinsichtlich der statistischen Einmaligkeit gilt nämlich dort, wo die Natur des Werks nur wenig Platz für eine persönliche Auswahl zulässt, dass es schwieriger ist, sich von der Einmaligkeit des Werks zu überzeugen: Die Auswahl muss überraschend und ungewöhnlich erscheinen, um Urheberrechtsschutz zu erlangen, ansonsten verlässt das Werk den Bereich des Banalen nicht.¹⁵

Eine triviale Kombination banaler Elemente könnte statistisch einmalig sein, wenn auch bloss, weil sie eine gewisse Anzahl Elemente kombiniert: Gemäss den Gesetzen der Wahrscheinlichkeit wird eine Kombination mit zahlreichen Elementen kaum noch einmal identisch kreiert. Eine strikte Anwendung dieses Kriteriums bei der Beurteilung des Urheberrechtsschutzes würde also darauf hinauslaufen, banale, triviale Schöpfungen zu schützen, soweit sie sich durch eine bestimmte Komplexität auszeichnen bzw. aus genügend vielen Elementen bestehen. Nicht nur die Lehre,¹⁶ sondern auch das schweizerische Bundesgericht¹⁷ sind daher der Meinung, dass die reine statistische Einmaligkeit allein für den Urheberrechtsschutz nicht genügt.

b) Softwareschnittstellen sind im Kern Ideen bzw. Grundsätze

Nun also zu den Schnittstellen und deren Spezifikationen bzw. Definitionen:

Unabhängig davon, ob Schnittstellen als selbständige Computerprogramme oder als Teil eines Computerprogramms ausgestaltet werden, kommt ihnen nur dann ein urheberrechtlicher Schutz zu, wenn sie auch selbst eine ausreichende schöpferische Individualität aufweisen. Als selbständige Computerprogramme ergibt sich dies unmittelbar aus Art. 2 Abs. 1 URG. Als Teile eines Computerprogramms weist ihnen Art. 2 Abs. 4 URG nur dann einen urheberrechtlichen Schutz zu, wenn sie selbst über eine ausreichende schöpferische Individualität verfügen.¹⁸

Nach überwiegender Ansicht ist bei Schnittstellen jedoch eine ausreichende schöpferische Individualität im Regelfall nicht gegeben.¹⁹ Ganz allgemein ist anerkannt, dass Schnittstellenspezifikationen nicht schutzfähig sind, da es sich dabei bloss um Ideen und Grundsätze der Interoperabilität handelt, denen der Urheberrechtsschutz verwehrt ist.²⁰ Zwar gehören die Schnittstellendefinitionen zum Computerprogramm, doch sind sie in der Regel wegen des fehlenden Gestaltungsspielraums²¹ bzw. mangels Schutzhöhe²² urheberrechtlich nicht geschützt.

c) Schnittstellen sind nicht direkt ausführbar

Auch wird in der Literatur ausgeführt, dass Schnittstellendefinitionen schon daher grundsätzlich nicht als Computerprogramme geschützt sind, weil sie nicht direkt ausführbar sind.²³ Es ist nämlich auch (bzw. insbesondere) im Bereich des Softwareschutzes zwischen dem Ausführbaren (hier: durch einen Computer ausführbare Software) und den statischen Daten zu differenzieren: Im Gegensatz zu Computerprogrammen haben Daten den Charakter isolierter Partikel (z.B. ja/nein, false/true, Zahlen), die erst vom Computerprogramm in einen Zusammenhang eingebettet werden.²⁴ Die speziellen Bestimmungen über den Softwareschutz gelten daher nicht für statische elektronische Dokumente,²⁵ weil diese nicht die für Computerprogramme typische unmittelbare Funktionalität aufweisen.²⁶ Die Softwareschutzbestimmungen sind vielmehr auf direkt ausführbare Werke zugeschnitten.²⁷ Mit anderen Worten: Auf der Ebene der statischen Daten greift noch kein rechtlicher Schutz – schon gar nicht ein urheberrechtlicher Schutz.

Wie aufgezeigt, gibt es im Zusammenhang mit Schnittstellenspezifikation kaum Gestaltungsspielraum, und wenn auch die eine oder andere Formulierung in der Schnittstel-

13 Vgl. Botschaft URG, 523.

14 Vgl. NEFF/ARN (Fn. 4), 132.

15 CHERPILLOD (Fn. 8), URG 2 N 29.

16 Vgl. die Zusammenstellung bei CHERPILLOD (Fn. 8), URG 2 N 30.

17 BGE 134 III 166 ff.

18 Vgl. NEFF/ARN (Fn. 4), 303; vgl. allgemein Botschaft URG, 523; BARRELET/EGLOFF (Fn. 10), URG 2 N 27; CHERPILLOD (Fn. 8), URG 2 N 67; R. M. HILTY, Urheberrecht, Bern 2011, 121 a.E.; NEFF/ARN (Fn. 4), 113.

19 BVGer vom 18. Dezember 2018,, B-831/2011, E. 578; S. BRÄNDLI, Die Flexibilität urheberrechtlicher Systeme, Bern 2017, Rn. 351; NEFF/ARN (Fn. 4), 146; R. OERTLI, in: B. K. Müller/R. Oertli (Hg.), Stämpflis Handkommentar zum Urheberrechtsgesetz (URG), 2. Aufl., Bern 2012, URG 21 N 11.

20 W. STRAUB, Softwareschutz, Zürich 2011, Rz. 249 ff., U. LOEWENHEIM, in: G. Schrickler/U. Loewenheim (Hg.), Urheberrecht, 6. Aufl., München 2020, DE-UrhG 69a N 13; sinngemäss auch OGer ZH, Teilurteil vom 24. Januar 2013, LK100006-O/U (www.gerichte-zh.ch/fileadmin/user_upload/entscheide/oeffentlich/LK100006-O14.pdf), E. 6.5, 6.8 und 6.12.

21 G. FRÖHLICH-BLEULER, Softwareverträge, 2. Aufl., Bern 2014, Rz. 80; Verfügung der Schweizerischen Wettbewerbskommission vom 29. November 2010 – Untersuchungsnummer 32-0205, RPW 2011/1; STRAUB (Fn. 19), Rz. 250; F.-X. STIRNIMANN, Urheberkartellrecht, Zürich 2004, 137 Fn. 480; G. FRÖHLICH-BLEULER, Urheberrechtliche Nutzungsbefugnisse des EDV-Anwenders, AJP 1995, 576; W. STRAUB, Der Sourcecode von Computerprogrammen im schweizerischen Recht und in der EU-Richtlinie über den Rechtsschutz von Computerprogrammen, UFITA 2001, 814; EU-Software-Richtlinie, E. 11.

22 FRÖHLICH-BLEULER (Fn. 21), Softwareverträge, Rz. 121; STRAUB (Fn. 20), Rz. 250; M. LEHMANN, Die Europäische Richtlinie über den Schutz von Computerprogrammen, in: M. Lehmann (Hg.), Rechtsschutz und Verwertung von Computerprogrammen, Köln 1993, Rz. 7.

23 STRAUB (Fn. 20), Rz. 250.

24 STRAUB (Fn. 20), Rz. 46.

25 Siehe hierzu auch KGer SG vom 17. Juli 2007, ZZ. 2006.36, E. 3a/cc/aaa und E. 3b/cc (<https://publikationen.sg.ch/rechtsprechung-gerichte-detail/4076/>).

26 M. BERGER, Schutz von Software – Überblick über die Rechtslage in der Schweiz, in: H. R. Trüeb (Hg.), Softwareverträge: Referate der Tagung der Stiftung für juristische Weiterbildung Zürich vom 11. November 2003, Zürich 2004, 25 ff., 45.

27 STRAUB (Fn. 20), Rz. 50.

lenspezifikation statistisch einmalig sein sollte, so würde der Urheberrechtsschutz an der Trivialität, Banalität bzw. Alltäglichkeit solcher Formulierungen scheitern. Dem «Beschrieb» einer Schnittstelle fehlt es somit in aller Regel an jeglicher schöpferischen Individualität – der Grundvoraussetzung für Urheberrechtsschutz nach Art. 2 Abs. 1 URG. Es handelt sich schlicht um eine standardisierte bzw. funktionsbestimmte – und damit nicht schutzfähige – Spezifikation.²⁸ Eine Entwicklungsdokumentation ist in aller Regel standardisiert bzw. funktionsbestimmt – und damit nicht schutzfähig.²⁹ Müssen Schnittstellen oder Teile davon aus technischen Gründen identisch oder praktisch identisch übernommen werden, damit sie überhaupt ihren Zweck erfüllen, dann ist dies zulässig.³⁰

d) Die Merger-Doktrin: das Verschmelzen von Idee und Ausdruck

Gemäss der sog. «Merger Doctrine»³¹ liegt im Übrigen dann keine Rechtsverletzung vor, wenn die «Idee» und der «Ausdruck» derart ineinander verschmolzen sind, dass sich die Übernahme der Ausdrucksformen nicht vermeiden lässt. Ein Ausdruck ist nicht schutzfähig, wenn es nur sehr wenige Möglichkeiten gibt, um eine Idee auszudrücken. Idee und Ausdruck fallen in einem solchen Fall zusammen.³² Auch dies spricht gegen einen Urheberrechtsschutz für Schnittstellen.

e) Interoperabilität und Dekompilierung

Interessant sind folgende Ausführungen des Kantonsgerichts St. Gallen:³³ «Für die Kommunikation unter Programmen ist die Kenntnis von sogenannten Schnittstelleninformationen von Bedeutung, da die Schnittstellen zur Herstellung von interoperablen Programmen benötigt werden. Schnittstellen sind Verbindungsstellen zwischen verschiedenen Systemen. Verbindungsstellen in diesem Sinne bestehen aus Informationen über die Art, wie an einer bestimmten Stelle Daten bereitgestellt werden müssen oder die Aufrufe für Programme zu erfolgen haben, damit die Zusammenarbeit zwischen verschiedenen Systemen möglich ist.³⁴ In diesem Bereich wurde der Schutz des Urheberrechts bewusst gelockert. Unter dem Kapitel «Schranken des Urheberrechts» regelt Art. 21 URG die sogenannte Dekompilierung bzw. Entschlüsselung eines Computerprogramms mit Blick auf seine Schnittstellen und deren Ausnutzung für die Herstellung interoperabler Computerprogramme. Wer das Recht hat, ein Computerprogramm zu gebrauchen, darf sich die erforderlichen Informationen über Schnittstellen zu unabhängig entwickelten Programmen durch Entschlüsselung des Programmcodes beschaffen oder durch Drittpersonen beschaffen lassen (Art. 21 Abs. 1 URG). Es ist dabei auch erlaubt, die Schnittstelleninformationen für ein Konkurrenzprodukt auszunutzen.³⁵»

f) Der Einfluss der Standardisierung

In der IT-Praxis lässt sich häufig beobachten, dass eine Schnittstelle und deren Beschreibung sich im Lauf der Zeit zu einem Standard oder zumindest zu einem Quasi-Standard bzw. faktischen Standard auf dem jeweiligen Markt etabliert. Werden für den Datentransfer bestimmte Standards durch Regulierungsorganisationen, Branchenverbände oder die in diesem Bereich tätigen Unternehmen festgelegt, scheidet das Geltendmachen eines Urheberrechts durch einen einzelnen Hersteller einer Computerprogrammkomponente aus.³⁶

g) Schnittstellen sind kaum je Sprachwerke

Bei Schnittstellen bzw. deren Spezifikationen dürfte schliesslich in aller Regel auch ein Schutz als Sprachwerk gemäss Art. 2 Abs. 2 lit. a URG nur schon am fehlenden Gestaltungsspielraum scheitern.³⁷ Keinen Urberschutz geniessen nämlich Sprachwerke, denen es an der individuellen Gestaltung fehlt, also z.B. Gebrauchsanweisungen, Produkteinformationen,³⁸ und dies immer auch dann, wenn der konkrete Text zwar statistisch einmalig ist, insgesamt aber doch als banale Zusammenstellung von Alltagsredewendungen oder als durch die Sachlogik vorgegeben erscheint.³⁹ Vergleichbar ist die Schnittstellenspezifikation mit den Charakteristiken einer Nachricht. Nachrichten sind Tatsachen, übermittelt in einer alltäglichen, nüchternen und schnörkellosen Sprache. Sie sind urheberrechtlich nicht schutzbar.⁴⁰

h) Fazit: Tendenziell kein selbständiger Urheberrechtsschutz für APIs in der Schweiz

Es finden sich also ganz viele Argumente, die im Zusammenhang mit der Rechtslage in der Schweiz gegen den Urheberrechtsschutz für eine Softwareschnittstelle bzw. deren

28 LOEWENHEIM (Fn. 20), DE-UrhG 69a N 19.

29 LOEWENHEIM (Fn. 20), DE-UrhG 69a N 19.

30 FRÖHLICH-BLEULER (Fn. 21), Softwareverträge, Rz. 121.

31 Diese gilt für die Schweiz via Art. 2 der WIPO Copyright Convention.

32 O. STAFFELBACH, Die Dekompilierung von Computerprogrammen gemäss Art. 21 URG, Bern 2003, 131; I. CHERPILLOD, Protection des logiciels et des bases de données: la révision du droit d'auteur en Suisse, SMI 1993, 63 Fn. 20; ausführlich zur sog. «Idea/expression dichotomy» im Urheberrecht R. A. GORMAN/J. C. GINSBURG, Copyright: Cases and Materials, 9. Aufl. 2017, 90 ff.

33 KGer SG vom 24. Mai 2005, DZ. 2002.3, E. 6.d (https://entscheid.suche.ch/kantone/sg_allent/dz_2002_3.html).

34 STAFFELBACH (Fn. 32), 70.

35 NEFF/ARN (Fn. 4), 304 ff.; I. CHERPILLOD, Schranken des Urheberrechts, in: SIWR II/1, 254 ff.

36 BVGer vom 18. Dezember 2018,, B-831/2011, E. 576.

37 STRAUB (Fn. 20), Rz. 250.

38 BGE 134 III 166 ff.

39 BARRELET/EGLOFF (Fn. 10), URG 21 N 21; CHERPILLOD (Fn. 8), URG 2 N 30.

40 R. VON BÜREN/M. A. MEER, Der Werkbegriff, in: R. von Büren/L. David (Hg.), Schweizerisches Immaterialgüter- und Wettbewerbsrecht (SIWR), II/1, Urheberrecht und verwandte Schutzrechte, 3. Aufl., Basel 2014, 93; T. CALAME, Elektronische Pressespiegel und Urheberrecht, recht 2002, 181 m.H.

Beschreibung (Spezifikation) sprechen. Selbstverständlich kann die Frage nach der Schützbarkeit einer API jedoch nur jeweils im Einzelfall zuverlässig analysiert werden.

8. Die Rechtslage in Deutschland und gemäss der Rechtsprechung des EuGH

Nicht fehlen sollte schliesslich bei einer Betrachtung der Möglichkeiten des Urheberrechtsschutzes von Software-schnittstellen auch die Perspektive des deutschen Rechts und gemäss der Rechtsprechung des EuGH:

a) Der Blick auf das deutsche Urheberrecht

Auch im deutschen Urheberrecht gilt, dass die «den Schnittstellen zugrundeliegenden Ideen und Grundsätze»⁴¹ nicht geschützt sind. Diese – nicht schutzfähigen – Elemente werden als «Schnittstellenspezifikation» (bzw. als «Schnittstellenbeschreibung») bezeichnet.⁴² Die überwiegende Mehrheit der Autoren in Deutschland fordert – wohl zu Recht – eine Trennung zwischen der Schnittstelle und ihrer «Spezifikation», also zwischen Ausdruck und Idee bzw. Form und Inhalt. Nicht urheberrechtlich geschützt bleibt damit stets die Schnittstellenspezifikation bzw. -beschreibung, also die blossen Idee. In Deutschland gehören Schnittstellenspezifikationen zu den nicht schutzfähigen Ideen und Grundsätzen eines Computerprogramms.⁴³ Das schliesst logisch aus, dass sie zugleich selbst Schutz als Computerprogramm geniessen können. Schnittstellenspezifikationen erfüllen 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 (denn es handelt sich nicht um Quellcode, Objektcode oder um einen vergleichbaren Code).⁴⁴ Diverse deutsche Autoren gehen noch einen Schritt weiter und sprechen sogar den Schnittstellen selbst jeglichen Urheberrechtsschutz ab.⁴⁵

Das LG Frankfurt hat in einem interessanten Fall entschieden,⁴⁶ dass einer XML-Datei die urheberrechtliche Schutzfähigkeit fehlt: Dabei wurde insbesondere befunden, dass es sich bei der fraglichen XML-Datei und den darin enthaltenen Regelsätzen nicht um ein Sprachwerk handelt, da vorliegend wenig bis gar kein geistiger Inhalt durch das Mittel der Sprache zum Ausdruck komme. Die Leistung müsse aus dem Werk erkennbar sein. Schriftgut, das Gebrauchszwecken dient und von ihnen weitgehend vorgegeben ist, müsse vergleichbare alltägliche Schriften deutlich übersteigen. Die Frage des Eigentümlichkeitsgrads bemesse sich dabei nach dem geistig-schöpferischen Gesamteindruck, und zwar im Gesamtvergleich gegenüber vorbestehenden Gestaltungen.

Weiter hat das LG Frankfurt festgestellt, dass die Gesamtheit der Regelsätze, die in der XML-Datei enthalten sind, die Voraussetzungen eines urheberrechtlich geschützten Werks nicht erfüllt, dass weder die XML-Datei, noch die

darin enthaltenen Regelsätze ein Computerprogramm darstellen und dass es sich bei der XML-Datei und den darin enthaltenen Regelsätzen um eine Schrift handle, die Gebrauchszwecken dient, nämlich dem Import in eine Datenbank-anwendung, wodurch diese Datenbank-anwendung auf die jeweiligen Bedürfnisse des Anwenders zugeschnitten wird. Dem Gericht erschliesse sich kein geistiger Inhalt der Regelsätze. Diese seien vielmehr bloss eine Aneinanderreihung von Zeichen, und selbst wenn ein geistiger Gehalt der Regelsätze erkennbar wäre, so sei nicht erstellt, dass diese Regelsätze bzw. der in ihnen verkörperte geistige Inhalt, also die geistige Schöpfung, sich deutlich vom Alltäglichen abheben.

b) Die Rechtsprechung des EuGH

Der EuGH hat die Trennung zwischen Implementation und Schnittstellenspezifikation für die mit den Kommunikationsprotokollen verwandten Dateiformate (es geht in beiden Fällen um Datenaustausch) bereits bestätigt.⁴⁷ Es ist nur konsequent, diesen Gedanken auch auf Kommunikationsprotokolle zu erstrecken.

41 Siehe § 69a Abs. 2 Satz 2 DE-UrhG.

42 M. LEHMANN, Freie Schnittstellen («interfaces») und freier Zugang zu den Ideen («reverse engineering»). Schranken des Urheberrechtsschutzes von Software, CR 1989, 1057, 1059; M. GRÜTZMACHER, Dateneigentum – ein Flickenteppich, CR 2016, 485, 494; T.C. VINJE, Die EG-Richtlinie zum Schutz von Computerprogrammen und die Frage der Interoperabilität, GRUR Int 1992, 250, 254; F. FALKER, in: J. Taeger (Hg.), Die Macht der Daten und der Algorithmen, 2019, 600; C. CZYCHOWSKI, in: A. Fromm/J.B. Nordemann (Hg.), Kommentar zum Urheberrechtsgesetz, 12. Aufl., Stuttgart 2018, DE-UrhG 69a N 32; B. CZARNOTA/R.J. HART, Legal Protection of Computer Programs in Europe, 1991, 38; T. HOEREN, IT-Recht, Stand: Oktober 2020, 59 f. (www.itm.nrw/wp-content/uploads/Skript-IT-Recht-Stand-Juni-2020-Stand-29.06.pdf, abgerufen am 21. November 2021), 16, 53, 58, 95, 98.

43 Siehe § 69a Abs. 2 Satz 2 DE-UrhG.

44 EuGH vom 22. Dezember 2010, C-393/09, K & R 2011, 105, Rn. 49, BSA/Kulturministerium; EuGH vom 2. Mai 2012, C406/10, WRP 2012, 802, Rn. 38, SAS Institute.

45 LEHMANN (M. LEHMANN, Die Europäische Richtlinie über den Schutz von Computerprogrammen, GRUR Int, 1991, 329) begründet dies mit der fortschreitenden Normierung und Standardisierung von Schnittstellen. MARLY (J. MARLY, Der Schutzgegenstand des urheberrechtlichen Softwareschutzes, GRUR 2012, 773, 779; J. MARLY, Praxishandbuch Softwarerecht, 7. Aufl., München 2018, Rn. 101) macht dies mit Blick auf das urheberrechtliche Erfordernis des Vorliegens einer Ausdrucksform (§ 69a Abs. 2 Satz 1 DE-UrhG) und führt in diesem Zusammenhang aus, «APIs» (also Schnittstellen) würden keine Vervielfältigung des Programms ermöglichen und seien deshalb nach den Vorgaben des EuGH keine Ausdrucksform. Auch seien Schnittstellen stets eine nicht schutzfähige Banalität (T. DREIER, Rechtsschutz von Computerprogrammen, CR 1991, 577, 583; GRÜTZMACHER, 494), und es gebe bei APIs technische Zwänge, die «einem Programmierer praktisch kaum einen kreativen Freiraum» (LEHMANN, 1059) eröffnen würden. GRÜTZMACHER (494) und KARL (C. KARL, Der urheberrechtliche Schutzbereich von Computerprogrammen, München 2009, 238) leiten den fehlenden Urheberrechtsschutz für Schnittstellen auch aus wettbewerbsrechtlichen Gründen ab, da ansonsten die Interoperabilität gefährdet sei.

46 LG Frankfurt a.M. vom 8. November 2012, 2–3 O 269/12.

47 EuGH vom 2. Mai 2012, C406/10, WRP 2012, 802, Rn. 38, SAS Institute.

Anfang Mai 2012 hat der EuGH erstmals entschieden,⁴⁸ dass Programmiersprachen keinen urheberrechtlichen Schutz genießen, denn sie sind in rechtlicher Hinsicht nicht mit Software-, Sprach- oder auch Musikwerken zu vergleichen.

Die Entscheidung verträgt sich gut mit der bisherigen schweizerischen Rechtsprechung zu schutzfähigen Werken.

Gleichzeitig regt sie dazu an, sich mit ähnlich gelagerten Fragen näher zu befassen, nämlich nach der urheberrechtlichen Einordnung von Dateiformaten, Protokollen und Benutzeroberflächen:

Das Urheberrecht ist im Kern der Schutz des Schöpferischen, d.h. dort wo es am Schöpferischen, an der Individualität bzw. der Originalität fehlt, da kann kein urheberrechtlicher Schutz geltend gemacht werden. Nicht geschützt sind – wie bereits erwähnt – Ideen, Konzepte und Grundsätze. Eine bloße Funktion, wie z.B. die Funktion, eine bestimmte Skriptsprache interpretieren zu können, ist keine Ausdrucksform. Diese Funktion – und damit letztendlich die Programmiersprache als solche – unterliegt daher nicht dem urheberrechtlichen Schutz. Wenn man bereits die Funktion (eines Computerprogramms) schützen wollte, so der EuGH in seiner letztlich wohlfahrtsökonomisch geprägten Argumentation, würde dies dem technischen Fortschritt und der industriellen Entwicklung schaden. Zudem würde es die Möglichkeit eröffnen, bloße Ideen zu monopolisieren.⁴⁹ Der Generalanwalt hatte in seiner Urteilempfehlung an den EuGH treffend angemerkt, dass eine Programmiersprache mit der Sprache eines Romanautors vergleichbar sei: Beide sind bloße Mittel zum Ausdruck, aber keine – urheberrechtlich schützbar – konkrete Ausdrucksform. Eine Programmiersprache könnte zwar theoretisch nach den allgemeinen Regeln als Sprachwerk urheberrechtlich geschützt sein. Das würde jedoch voraussetzen, dass die Auswahl der Schlüsselwörter und Operatoren sowie die Syntax, die von einer Programmiersprache vorgegeben werden, an eine geistige Schöpfung heranreichen. Das ist jedoch reine Theorie, denn die Programmiersprache wurde mit dem Ziel konzipiert, gemessen an ihrem Zweck möglichst effizient und leicht erlernbar zu sein. Die Gestaltungskriterien sind somit vorrangig funktional. Was seine Gestaltung aber nur einer rein technischen Zweckbestimmung verdankt, stellt eben nicht das Ergebnis eines schöpferischen Prozesses dar und ist damit einem urheberrechtlichen Schutz nach den allgemeinen Grundsätzen nicht zugänglich.

Diese vom EuGH entwickelten Grundsätze lassen sich nun auch im Hinblick auf Dateiformate, Protokolle, Benutzeroberflächen und Schnittstellenbeschreibungen anwenden:

Bei der Interpretation eines bestimmten Dateiformats handelt es sich nur um eine andere Art des Inputs als bei der Verarbeitung bestimmter Skripte. Hinzu kommt die Frage, ob die Output-seitige Verwendung eines Dateiformats (also die Fähigkeit, Dateien in einem bestimmten Format schreiben zu können) eine andere rechtliche Beurteilung erfordert. Tatsächlich musste der EuGH im Rahmen seiner Entscheidung auch genau diese Frage im Zusammen-

hang mit dem Dateiformat beantworten. Konsequenterweise stellte er fest, dass die Funktion, Dateien in einem bestimmten Format schreiben zu können, ebenso wenig geschützt ist, wie die des Einlesens. Im Ergebnis sind also Dateiformate wie Programmiersprachen zu beurteilen und daher grundsätzlich keinem urheberrechtlichen Schutz zugänglich.

Ein verwandter Bereich betrifft Netzwerkprotokolle: Die für Skripte und Dateiformate geltenden Grundsätze lassen sich auch auf solche technischen Definitionen anwenden. Ähnlich wie Dateiformate geben sie lediglich vor, welches Format die über eine Schnittstelle einzulesenden oder auszugebenden Daten haben müssen. Die Funktion, ein bestimmtes Protokoll verstehen und ausgeben zu können, genießt daher ebenso wenig urheberrechtlichen Schutz. Wie Programmiersprachen bieten auch Benutzerschnittstellen die Möglichkeit, mit einer Software zu kommunizieren. Das kann über Oberflächen unterschiedlicher Art geschehen: Die Interaktion kann grafisch, akustisch oder textorientiert erfolgen, Nutzer-Input kann eingetippt, ausgesprochen, per Klick oder per Geste vermittelt werden. Auf textorientierte Benutzerschnittstellen, die etwa mit einer Befehlszeile oder einfachen Menüs arbeiten, lässt sich das zu Protokollen und Dateiformaten Geschriebene ebenfalls anwenden.

c) Fazit nach deutschem Recht und der EuGH-Rechtsprechung

Auch nach deutschem Recht und im Licht der einschlägigen Rechtsprechung des EuGH dürften somit Softwareschnittstellen bzw. deren Beschreibungen (Spezifikationen) urheberrechtlich nicht schützbar sein.

II. Judikatur: Softwareschutz & Urheberrecht

Kommen wir nun zu den «Schüttgefässwaagen-Fällen», wo die Klägerin sich der Hauptherausforderung zu stellen hatte, wie sich substantiieren und beweisen lässt, dass sie überhaupt über die Urheberrechte an der durch die Beklagte übernommenen Software oder an den übernommenen Softwarekomponenten verfügt.

Schüttgefässwaagen bestehen aus interoperierenden Hard- und Softwarekomponenten. In allen vier Gerichtsfallen geht es insbesondere um Software zur elektronischen Steuerung von Waagen und Umwandlung analoger Messdaten in digitale Messwerte. Diese Software war von einem Mitarbeiter der Klägerin und von einem externen Softwareentwickler geschaffen worden. Die Klägerin beanspruchte, Eigentümerin der Urheberrechte an der Software zu sein. Der von der Klägerin beauftragte Drittentwickler hatte die Software ehemaligen Mitarbeitenden der Klägerin für die

48 EuGH vom 2. Mai 2012, C-406/10 (<https://dejure.org/dienste/vernetzung/rechtsprechung?Gericht=EuGH&Datum=02.05.2012&Aktenzeichen=C-406/10>).

49 W.M. LANDES/R.A. POSNER, *The Economic Structure of Intellectual Property Law*, 2003, 91 ff.

Entwicklung eines Konkurrenzprodukts zur Verfügung gestellt.

In allen vier Gerichtsfällen ging es im Kern stets um die drei Fragen, was für Elemente denn konkret übernommen worden waren, ob die übernommenen Elemente urheberrechtlich geschützt seien und ob die Urheberrechte an diesen Elementen der Klägerin zustehen.

Die Klägerin legte jeweils ein Gutachten vor, um aufzuzeigen, dass acht Funktionalitäten der Originalsoftware praktisch unverändert bzw. bloss geringfügig angepasst im Konkurrenzprodukt integriert waren. Dieses Gutachten wurde anhand einer Analyse der kompilierten Maschinen-codes (Original und Übernahme) erstellt.

Gerichtlich nicht entschieden wurde, ob die allenfalls bestehenden Urheberrechte der Klägerin zustehen. Die entsprechenden Massnahmengesuche gegen die Konkurrentin und den Drittentwickler wurden abgewiesen, da die Klägerin nicht substantiiert hatte, worin die angeblich übernommenen Funktionalitäten bestehen. Auch hatte sie nicht aufgezeigt, und dass die übernommenen Funktionalitäten angesichts der Vorgaben aus dem Stand der Technik überhaupt Individualität aufweisen.

Aus den Entscheiden lässt sich folgern, dass die Individualität übernommener Software(-Komponenten) in der Rechtsschrift selbst substantiiert und im Bestreitungsfall bewiesen werden muss – idealerweise mit den Quellcodes. Privatgutachten können zwar nützlich sein, sie sind aber als reine Parteibehauptungen zu werten und daher für sich nie ausreichend.

1. «Schüttgefässwaage 1 II» (BGer vom 2. Dezember 2019)

Das Schweizerische Bundesgericht in Lausanne befasste sich in seinem Entscheid «*Schüttgefässwaage 1 II*» vom 2. Dezember 2019⁵⁰ mit dem Fall der von der X. AG neu auf den Markt gebrachten, mit Druckluft betriebenen Schüttgefässwaagen. Die Y. AG, ein Technologieunternehmen, hatte ihrerseits bereits 2013–2017 solche Schüttgefässwaagen entwickelt und ging davon aus, dass ihre acht ehemaligen Mitarbeitenden, die inzwischen alle bei der X. AG tätig waren, das Konkurrenzprodukt durch unrechtmässige Verwendung von Arbeits- und Geschäftsgeheimnissen hergestellt hatten. Das Bundesgericht hiess die Beschwerde der Y. AG gegen den offensichtlich mangelhaften Entscheid der Vorinstanz, des Kantonsgerichts Appenzell Innerrhoden, gut und wies den Fall zur nochmaligen Beurteilung an die Vorinstanz zurück. Diese müsse nicht nur beurteilen, ob den angeblich rechtswidrig verwendeten Konstruktionsplänen, Leiterplatten und Dateien Geheimnischarakter zukomme, sondern v.a. auch, ob sie als Arbeitsergebnisse oder als urheberrechtlich geschützte Werke zu qualifizieren seien. Auch ohne Konkurrenzverbotsklausel i.S.v. Art. 340 des Schweizerischen Obligationenrechts (OR) bestehe gemäss der arbeitsrechtlichen Bestimmung von Art. 321a OR eine allgemeine Treuepflicht des Arbeitnehmers, aus der eine über die Beendigung des Arbeitsvertrags andauernde Geheimhaltungspflicht folge. Das Kantonsgericht Appenzell Innerrho-

den hatte daher fälschlicherweise befunden, dass hier die während eines Arbeitsverhältnisses erworbenen spezifischen Branchenkenntnisse frei verwendbar sind und nicht unter Art. 6 des Bundesgesetzes gegen den unlauteren Wettbewerb (UWG) fielen.

2. «Schüttgefässwaage 2 I» (HGer SG vom 24. Januar 2020)

Im Massnahmeentscheid «*Schüttgefässwaage 2 I*» des Handelsgerichts St. Gallen vom 24. Januar 2020⁵¹ hatte sich das Gericht im Zusammenhang mit vorsorglichen Massnahmen (einstweilige Verfügung) mit den Substantiierungsanforderungen beim Geltendmachen einer Urheberverletzung an einem Softwareprogramm für die Steuerung einer Waage für Mehl und Futtermittel zu befassen. Der Gerichtspräsident hielt zunächst fest, dass Computerprogramme mit gewisser Komplexität i.d.R. urheberrechtlich geschützt seien. Werde die unerlaubte Übernahme von Programmteilen behauptet, so sei die Rechtsinhaberschaft am originären Code und die Übernahme konkreter Codeteile glaubhaft zu machen. Der Gerichtspräsident wies in der Folge die von der Gesuchstellerin (wiederum der Y. AG) geltend gemachten Ansprüche ab, da die Y. AG nicht dargelegt habe, «welche Funktionen die angeblich übereinstimmenden Dateien konkret beinhalten. Nachdem die Gesuchstellerin [...] weder darlegt, welchen Inhalt die einzelnen angeblich übereinstimmenden Programmfunktionen aufweisen, noch Ausführungen darüber macht, welche schöpferischen Eigenleistung XY bei der Programmierung dieser Programmfunktionen erbracht haben soll, kann nicht geprüft werden, ob sich diese Programmfunktionen von bereits vorbestehenden Programmfunktionen unterscheiden, einen individuellen Charakter aufweisen und somit urheberrechtlich geschützt sind. Eine Verletzung eines Urheberrechts ist damit nicht glaubhaft gemacht.»⁵²

3. «Schüttgefässwaage 2 II» (BGer vom 22. September 2020)

Das Schweizerische Bundesgericht wies die gegen den Massnahmeentscheid «*Schüttgefässwaage 2 I*» des Handelsgerichts St. Gallen vom 24. Januar 2020 erhobene Beschwerde der Y. AG in seinem Entscheid «*Schüttgefässwaage 2 II*» vom 22. September 2020 ab, soweit darauf eingetreten wurde.⁵³ Wird in einem Massnahmeverfahren das unerlaubte Verwenden und die Weitergabe neu entwickelter Software geltend gemacht, so sei davon auszugehen, dass ein nicht wieder gutzumachender Nachteil gemäss Art. 93 Abs. 1 lit. a des Bundesgesetzes über das Bundesgericht (BGG) glaubhaft ist, «zumal es sich bei den angeblich vom Beschwerdegegner verwendeten Programmfunktionen der strittigen Software

50 BGer vom 2. Dezember 2019, 4A_381/2019.

51 HGer SG vom 24. Januar 2020, HG.2019.32-HGP.

52 HGer SG vom 24. Januar 2020, HG.2019.32-HGP, E. 2.3.

53 BGer vom 22. September 2020, 4A_115/2020.

um Bestandteile einer Neuentwicklung handelt, für die Umsatz- und Gewinnzahlen fehlen». Zudem sei «mit der Beschwerdeführerin davon auszugehen, dass das Motiv der Kundschaft für die Wahl von Konkurrenzprodukten, die unter Verwendung geschützter Softwarebestandteile hergestellt

wurden, nicht nachweisbar sein dürfte». Unter diesen Umständen sei «entgegen der Ansicht des Beschwerdegegners nicht zu erwarten, dass der drohende Nachteil durch Schadenersatz, Genugtuung oder Gewinnherausgabe zu beseitigt ist.»⁵⁴

Zusammenfassung

Im Bereich der Softwaretechnik und -entwicklung ist es weit verbreitet, von anderen geschriebene Softwareschnittstellen zu verwenden. In der Schweiz, in Deutschland, in vielen anderen europäischen Ländern und in den USA ist – auch durch eine gefestigte Rechtsprechung – allgemein anerkannt, dass Software-Quellcode als Ganzes meist urheberrechtlich geschützt ist. Die Frage, inwieweit Softwareschnittstellen bzw. API-Code kopiert werden dürfen, blieb aber bisher von den Gerichten unbeantwortet.

Der Oberste Gerichtshof der USA, der US Supreme Court, hat in seiner Entscheidung «Google LLC v. Oracle America, Inc.» vom 5. April 2021 ein wenig Klarheit darüber geschaffen, in welchem Umfang der zur Erstellung einer Softwareschnittstelle verwendete Deklarationscode kopiert werden kann. Das Gericht verneinte das Vorliegen einer Urheberrechtsverletzung auf der Basis der sog. Fair-Use-Doktrin und fügte an, dass ein Durchsetzen des Urheberrechts auf der Grundlage dieser Tatsachen die Gefahr bergen würde, dass der Öffentlichkeit «kreativitätsbezogene Schäden» entstünden. Aus der Kombination der vier Fair-Use-Faktoren ergab sich für das Gericht, dass Google sich mit dem Kopieren des API-Codes durchaus im Rahmen der Fair-Use-Doktrin – und damit in zulässiger Weise – bewegte. Der Oberste Gerichtshof der USA wies darauf hin, dass seine Entscheidung keine Änderung gegenüber seinen anderen Urteilen zur Fair-Use-Lehre darstelle. Vielmehr biete diese einen Rahmen und eine Analyseanleitung für die besonderen und einzigartigen Fragen und Umstände, die sich ergeben, wenn ein urheberrechtlich geschützter Computercode von anderen Entwicklern und Unternehmen verwendet wird. Wichtig ist, dass das Gericht nicht entschieden hat, ob der deklarierende Code für sich allein urheberrechtlich ist. Diese Frage wurde offengelassen. Entschieden wurde bloss, dass im Zusammenhang mit Google das Verwenden eines kleinen Prozentsatzes des Codes zur Erstellung der Android-API zulässig war. Ein Durchsetzen von Oracles Urheberrechten hätte sicherlich Konsequenzen auf globaler Ebene gehabt, denn es wird vermutet, dass rund 70% der Smartphones davon betroffen gewesen wären: Dies hätte unweigerlich zu enormen Schadenersatzklagen geführt.

Résumé

Dans le domaine de la technique et du développement de logiciels, il est très répandu d'utiliser des interfaces logicielles écrites par des tiers. En Suisse, en Allemagne, dans de nombreux autres pays européens et aux États-Unis, il est communément admis, notamment par une jurisprudence bien établie, que le code source d'un logiciel dans son ensemble est généralement protégé par le droit d'auteur. Cependant, la question de savoir dans quelle mesure les interfaces logicielles ou le code API peuvent être copiés est restée jusqu'à présent sans réponse de la part des tribunaux.

Dans sa décision «Google LLC v. Oracle America, Inc.» du 5 avril 2021, la Cour suprême des États-Unis a apporté un peu de clarté quant à la mesure dans laquelle le code de déclaration utilisé pour créer une interface logicielle pouvait être copié. Le Tribunal a nié l'existence d'une violation du droit d'auteur sur la base du principe de l'usage loyal (*fair use doctrine*) et a ajouté que l'application du droit d'auteur à ces faits risquerait de causer au public des «dommages en matière de créativité». La combinaison des quatre facteurs déterminant l'usage loyal a permis à la Cour de conclure que Google, en copiant le code API, se situait bien dans le cadre du principe de l'usage loyal – et avait donc agi de manière licite. La Cour suprême des États-Unis a souligné que sa décision ne constituait pas un changement par rapport à ses autres jugements concernés par le principe de l'usage loyal. Au contraire, elle a estimé qu'elle fournissait ainsi un cadre et un modèle d'analyse pour les questions qui se posent et les circonstances particulières et uniques qui surviennent lorsqu'un code informatique protégé par des droits d'auteur est utilisé par d'autres développeurs et entreprises. Il est important de noter que le Tribunal n'a pas déterminé si le code dont il était question était en soi soumis au droit d'auteur; cette question a été laissée en suspens. Il a seulement été stipulé que, dans le contexte de Google, l'utilisation d'un petit pourcentage du code pour la création de l'API Android était licite. L'application des droits d'auteur d'Oracle aurait certainement entraîné des conséquences au niveau mondial, car on estime qu'environ 70% des smartphones auraient été concernés: cela aurait inévitablement conduit à d'énormes actions en dommages et intérêts.

54 BGer vom 22. September 2020, 4A_115/2020, E. 1.3.

Nach überwiegender Ansicht ist in der Schweiz bei Schnittstellen eine ausreichende schöpferische Individualität im Regelfall nicht gegeben. Es ist anerkannt, dass Schnittstellenspezifikationen nicht schutzfähig sind, da es sich dabei bloss um Ideen und Grundsätze der Interoperabilität handelt, denen der Urheberrechtsschutz verwehrt ist. Zwar gehören die Schnittstellendefinitionen zum Computerprogramm, doch sind sie in der Regel wegen des fehlenden Gestaltungsspielraums bzw. mangels Schutzhöhe urheberrechtlich nicht geschützt. Wenn auch die eine oder andere Formulierung in der Schnittstellenspezifikation statistisch einmalig sein sollte, so würde der Urheberrechtsschutz an der Trivialität, Banalität bzw. Alltäglichkeit solcher Formulierungen scheitern. Dem «Beschrieb» einer Schnittstelle fehlt es somit in aller Regel an jeglicher schöpferischen Individualität – der Grundvoraussetzung für Urheberrechtsschutz nach Art. 2 Abs. 1 URG. Es handelt sich schlicht um eine standardisierte bzw. funktionsbestimmte – und damit nicht schutzfähige – Spezifikation. Eine Entwicklungsdokumentation ist in aller Regel standardisiert bzw. funktionsbestimmt – und damit nicht schutzfähig. Auch wird in der Literatur ausgeführt, dass Schnittstellendefinitionen schon daher grundsätzlich nicht als Computerprogramme geschützt sind, weil sie nicht direkt ausführbar sind. Es ist nämlich auch (bzw. insbesondere) im Bereich des Softwareschutzes zwischen dem Ausführbaren (hier: durch einen Computer ausführbare Software) und den statischen Daten zu differenzieren: Im Gegensatz zu Computerprogrammen haben Daten den Charakter isolierter Partikel (z.B. ja/nein, false/true, Zahlen), die erst vom Computerprogramm in einen Zusammenhang eingebettet werden. Die speziellen Bestimmungen über den Softwareschutz gelten daher nicht für statische elektronische Dokumente, weil diese nicht die für Computerprogramme typische unmittelbare Funktionalität aufweisen. Die Softwareschutzbestimmungen sind vielmehr auf direkt ausführbare Werke zugeschnitten. Mit anderen Worten: Auf der Ebene der statischen Daten greift noch kein rechtlicher Schutz – schon gar nicht ein urheberrechtlicher Schutz. Müssen schliesslich Schnittstellen oder Teile davon aus technischen Gründen identisch oder praktisch identisch übernommen werden, damit sie überhaupt ihren Zweck erfüllen, dann ist dies zulässig.

Selon l'opinion dominante en Suisse, les conditions suffisantes pour définir un caractère individuel créatif ne sont généralement pas réunies en ce qui concerne les interfaces. Il est admis que les spécifications d'interface ne peuvent pas être protégées, car il s'agit uniquement d'idées et de principes d'interopérabilité qui ne sont pas protégés par le droit d'auteur. Les définitions d'interface font certes partie d'un programme informatique, mais elles ne sont généralement pas protégées par le droit d'auteur en raison du manque de liberté de conception, et donc du champ de protection. Même si l'une ou l'autre formulation dans la spécification d'une interface devait être statistiquement unique, la protection par le droit d'auteur échouerait en raison de l'aspect trivial, banal ou commun de telles formulations. La «description» d'une interface est donc généralement dépourvue de tout caractère individuel créatif, condition de base pour jouir de la protection du droit d'auteur selon l'art. 2 al. 1 LDA. Il s'agit simplement d'une spécification standardisée et déterminée par la fonction – et donc non protégeable –, comme c'est en général le cas de la documentation de développement. La littérature explique également que les définitions d'interface ne sont en principe pas protégées en tant que programmes informatiques parce qu'elles ne sont pas directement exécutables. En effet, dans le domaine de la protection des logiciels, il convient également (voire surtout) de faire la distinction entre ce qui est exécutable (en l'espèce: un logiciel exécutable par un ordinateur) et les données statiques: contrairement aux programmes informatiques, les données ont le caractère de particules isolées (p.ex. oui/non, faux/vrai, chiffres) qui ne sont intégrées dans un contexte que par le biais du programme informatique. Les dispositions spécifiques relatives à la protection des logiciels ne s'appliquent donc pas aux documents électroniques statiques, car ceux-ci ne présentent pas la fonctionnalité immédiate caractéristique des programmes informatiques. Les dispositions relatives à la protection des logiciels sont plutôt conçues pour les œuvres directement exécutables. En d'autres termes, aucune protection juridique n'intervient encore au niveau des données statiques – et en tout cas pas une protection par le droit d'auteur. Enfin, la reprise à l'identique ou pratiquement à l'identique d'interfaces ou de parties d'interfaces est licite si elle est nécessaire pour des raisons techniques, afin que celles-ci puissent remplir leur fonction.