Copy

Hamburg District Court
File no: 310 O 89/15

Pronounced on 08.07.2016

Heinelt, Judicial Clerk
Registrar to the Court

Judgment

IN THE NAME OF THE PEOPLE

In the matter

Christoph Hellwig, Schidlachstraße 11, 6020 Innsbruck, Austria

- **Plaintiff** -

Counsel on the record:
Law firm **Jaschinski, Biere, Brexl,** Christinenstraße 18/19, 10119 Berlin, file no.: 14-1640

versus

VMware Global, Inc., Zweigniederlassung Deutschland, represented by Patrick Kevan Krysler, Freisinger Straße 3, 85716 Unterschleißheim

- **Defendant** -

Counsel on the record:
Law firm **Freshfields, Bruckhaus, Deringer,** Im Zollhafen 24, 50678 Cologne, file no.: DAC17877145

the 10th Civil Chamber of Hamburg District Court rules through Presiding District Court Judge Hartmann, District Court Judge Harders and District Court Judge Dr. Frantzen on the basis of the hearing held on 25.02.2016 that:

1.

The action is dismissed.

2.

The Plaintiff bears the costs of the legal dispute.

3.

The judgment is provisionally enforceable on payment of security at a rate of 110% of the respective amount to be enforced.

Findings of Fact

The Plaintiff is a software developer. He claims that he has been involved in work on the kernel of the well-known Linux operating system and that he has thus acquired adapter's copyrights. The Plaintiff claims further that the Defendant has used parts of his Linux code for a software product of its own that it offers for downloading. For this, according to the Plaintiff, the Defendant may not quote the Open Source License under which Linux is published, because it has failed to comply with the license terms; for this reason, licensing based on that specific resolutory condition may no longer be applied, meaning that a right of use does not currently exist for the Defendant in respect of said parts of the Linux code, and that the Plaintiff as a co-adapter of Linux may prohibit the Defendant from using its software that it has developed using Linux.

For the details:

According to the parties' submissions in these proceedings, the original version of the **Linux operating system** was initiated long before the year 2000 by the programmer Linus Torvalds, and it has since been further developed by numerous programmers independently from one another. Since 2006 the **version control system "git"** has been used for the Linux system, from which it can be verified which contributions have been made by which programmers (cf. extract from website, Exhibit K1).

The Linux software has been licensed throughout as free software under the terms of the **GNU General Public License, Version 2, also referred to as GPL-2.0** (text: Exhibit K 6). The basic concept of this licensing model is that every software developer who makes a contribution to Linux allows anyone generally not only to use Linux, including the respective contributions he or she has made, but also to further develop it – on the resolutory condition, however, that the further development is in turn likewise licensed under the GPL, i.e. is made available to all and sundry for free usage and further development. Thus users are meant to receive comprehensive rights of use in return for themselves granting all other users comprehensive rights of use on passing on the software. This includes amongst other things disclosing the source code of the modified version.

The **Plaintiff** is a self-employed software developer and has been involved in work on a large number of Open Source Software projects, for instance in the further development of Linux since the year 2000. Exactly which parts of Linux he has modified is disputed. What is not disputed however, is that the Plaintiff, whenever he has made modifications to the Linux software, has licensed them under the GPL-2.0 in accordance with the aforementioned concept.

In the case instant, the Plaintiff is asserting **rights in the kernel of the Linux operating system**, also and in particular in connection with the system for addressing external devices via device drivers. As regards the general terms "operating system", "device driver" and "interface" – irrespective of the programs at the centre of this dispute – the following is undisputed:

- According to information from the parties in the case instant, the **kernel of an operating system** is understood to be the core of an operating system, whereby this core is responsible for managing and addressing data storage devices (hard discs, optical drives, flash memory devices) and for managing the device drivers (Plaintiff, Statement of Claim, p. 8 and 9), and the task of the kernel – apart from other supervisory functions – is to manage, verify, organise and prioritise input/output requests from other software components and applications (Defendant, written pleading 01.07.2015 p. 6 = p. 68 of the annex).

- In order for the kernel of an operating system to communicate with an external device connected to the computer, a **device driver** is required. Without this being contested, the Defendant has argued here that basically a device driver is software which operates or controls a certain type of device that is connected with the computer, by translating requests from applications and from the operating system, depending on the particular requirements of the respective device's type, brand and model. If a computer program wants to communicate with a device, the driver has to issue the device with a command or series of commands (written pleading 01.07.2015 p. 7 = p. 61 of the annex).

- Interaction between the kernel and device drivers can be achieved by varying the structure of the operating system. Without this being contested, the Plaintiff has stated that in the structure of an operating system the device drivers can either be joined to the rest of the kernel to form a single binary file, or be exported as **kernel modules** that then have to be loaded dynamically by the system (Statement of Claim, p. 9).

- **Interface** is the term used for a component that enables other components to communicate; in hardware, it may be a certain plug-in connection; in software, it may involve certain programming requirements. An interface can be termed an **abstract or stable interface** if its configuration or requirements do not alter even if the communicating components are developed further; thus for Windows, for instance, Microsoft provides special public interfaces for device drivers, and these interfaces remain unaltered over a longer period (Statement of Claim p. 17). It is on this premise that the hardware manufacturers who produce the external devices having to be addressed, knowing the specification of the abstract interface for their respective device, normally develop and provide device drivers that fit the interface (cf. Defendant written pleading 01.07.2015 p. 7 = p. 61 of the annex).

In contrast to this, **Linux kernel modules** are not conventional drivers or other modules that can be loaded dynamically and are developed using an abstract interface definition; instead, they are fully integrated parts of the overall Linux program with full access to the internals of the Linux operating system, and they are only exported (if at all) in order to reduce usage of resources (Statement of Claim p. 11). As a matter of principle, **the drivers and the core** of the operating system cannot be separated in Linux, in contrast to other operating systems; in Linux, the internal interfaces and the drivers are always from the same originator. The **interfaces in Linux** are not stable, i.e. neither unchanging nor downwards-compatible. Instead, interfaces and drivers are constantly adjusted to one another. The operating system core and the drivers are interdependent and their development is both mutual and reciprocal (Statement of Claim p. 17).

The **Defendant** is the manufacturer of the **software ESXi;** the Defendant itself refers to this software as its “main product”. ESXi is so-called **virtualisation software**, which enables the user to divide one computer into several virtual machines. ESXi consists of several different programs and components. In particular, ESXi contains a component called “**Hypervisor**”, which is a special kind of operating system that enables several different operating systems to run parallel on one and the same computer. Like other operating systems, ESXi also contains a kernel: it is called “**vmkernel**” (whereby the abbreviation “vm” stands for “virtual machine”). If ESXi is installed on a computer, the user can instruct the “vmkernel” to start a virtual machine and then install an operating system on that virtual machine. The user can also start several virtual machines and thus install whatever number of operating systems it wants on the virtual machines created by the “vmkernel”. In this manner, with ESXi the Linux operating system can also be installed on a virtual machine.

It is undisputed between the parties that in terms of structure, the **architecture of ESXi** is such that the operating system kernel is separate from the kernel modules:

- The actual **kernel** of ESXi is the “**vmkernel**”. It only exists in binary or object code, i.e. it cannot be modified by a programmer unless it is translated into source code. The Defendant has not disclosed the source code for the “vmkernel”; nor has it offered any licensing for the “vmkernel”, in particular licensing under the GPL-2.0. The Plaintiff does not claim that the “vmkernel” itself as such also contains Linux code.

- In addition to this is a **module** which the Defendant calls “**vmlinux**”. It is not disputed between the parties that this module contains parts of modified code of the Linux operating system (the dispute is above all about which parts of the code are involved and whether they originate from the Plaintiff). It is undisputed that the Defendant has disclosed the source code of this module on its website where it is offered for downloading, and to this extent licensed it under the GPL-2.0.

- Finally, ESXi contains **further modules,** in which **device drivers** are to be found, some of which are modified device drivers.

According to the Plaintiff’s pleading – denied outright by the Defendant – the Defendant offers the software ESXi in the version specified in Petition no. 1) as an executable program, i.e. in object code, through its website for downloading via the Internet also in Germany. The Plaintiff regards this as an infringement of his rights in the Linux code that is used, and issued the Defendant with a warning in the letter dated 20.08.2014 (Exhibit K 8). The Defendant replied without issuing a negative covenant (letter dated 19.09.2014, Exhibit K 9). The Plaintiff sent an email on 14.11.2014 (Exhibit K 10). Subsequent talks between the parties failed to lead to an agreement.

The Plaintiff claims as follows:

He, the Plaintiff, has **modified substantial parts of the Linux kernel** and acquired adapter’s copyrights in this respect. The source code developed by the Plaintiff can be accessed by the general public in the **git repository** at <https://git.kernel.org/cgit/linux/kernel/> (written pleading 25.09.2015 p. 3 = p. 147 of the annex). Each contribution towards development which the Plaintiff has made can be verified and is documented in detail in the repositories of the main-line kernel (i.e. the “official” version of the kernel) (Statement of Claim p. 15).

Apart from this, the Plaintiff has presented on the **CD-ROM K 12** (folder: **history.tgz**) what he states to be the full source code of the latest stage of development of the Linux program with the source-code management system Bitkeeper (**Linux Version 2.6.12-rc2**) as an archive (history.tgz) and the entire changelog of the version hitherto in git format (written pleading 25.09.2015 p. 10 = p. 154 of the annex). Moreover, the Plaintiff specifically refers to the folder "**linux.tgz**" which CD-ROM K 12 contains, stating that he is thus submitting the archive that contains both the Linux kernel in Version 2.6.18 and the entire changelog of the version from Linux 2.6.12-rc to Linux 2.6.18 in git format (written pleading 25.09.2015 p. 10 = p. 154 of the annex). The Plaintiff states further that **Version 2.6.18** of the Linux kernel is the one most similar to the version used by the Defendant (written pleading 25.09.2015 p. 10 = p. 154 of the annex; cf. also written pleading 29.04.2016 p. 10 = p. 274 of the annex); that the CD-ROM K 12 thus provides the source code that the Plaintiff has contributed to Linux and turns up again in the Defendant's "vmklinux" (written pleading 25.09.2015 p. 10 = p. 154 of the annex); and that the **"…blame" files** likewise submitted on the CD-ROM K 12 show which code originates from the Plaintiff and which changes the Defendant has made (for the details: written pleading 25.09.2015 p. 3 = p. 147 of the annex).

The Plaintiff claims further that in the pre-litigation **email K 10** – the existence and wording of which are undisputed – he gave the Defendant examples of those parts of named source-code files (scsi_error.c, scsi_lib.c, scsi_proc.c scsi.c und host.c) which he had developed (written pleading 25.09.2015 p. 2 = p. 146 of the annex).

The Plaintiff then submits the 3-page Exhibit **K 15**, stating that from the information it contains it can be gathered which of the contributions that he made have been taken over by the Defendant (written pleading 25.09.2015 p. 11 = p. 155 of the annex). The Plaintiff states that the code he developed is to be found in ESXi to an extent that is relevant in copyright terms, and that it can be tracked by means of the source-code comparison attached in Exhibit K 15; said Exhibit also describes how the comparison can be carried out; a full comparison of the source codes has been done using the Linux code in K 15 and the source code of "vmklinux" which the Defendant offers (written pleading 25.09.2015 p. 11 = p. 155 of the annex).

In addition to this, two files for which the Defendant has disclosed the source code contain **headers with references to copyrights** held by the Plaintiff amongst others (for details: Statement of Claim p. 14 = p. 14 of the annex). The Plaintiff refers in this context to Exhibit K 3.

The Plaintiff argues that he was involved above all in work on **the Linux kernel's SCSI subsystem**. This code has been developed as part of the Linux kernel since 1992 (without his involvement at that early stage) and has always been published together with the kernel as part of a uniform program. He states that it is indeed possible to export the SCSI subsystem to a separate kernel module, but that this is only seldom done in practice. The SCSI subsystem interacts directly with the memory management, the file systems and the scheduler (for controlling the operating system's processes) and is not called via abstract interfaces (Statement of Claim p. 12). During the period from 2000 – 2004, the Plaintiff himself was one of the most active developers working on the SCSI subsystem of the Linux kernel, which by all means involved programming that was complex and thus merited protection (written pleading 25.09.2016 p. 8 = p. 152 of the annex). Other developers of the Linux kernel could bear witness to the fact that the Plaintiff had made material and complex contributions towards development of the SCSI subsystem (written pleading 25.09.2015 S. 10 = p. 154).

The Plaintiff continues by pleading that there are indications that part of the SCSI functionality is also to be found **in the "vmkernel**", but that without anything else to go on it has to be assumed that they are the Defendant's own developments (written pleading 29.04.2016 p. 6-7 = p. 270 f.).

The Plaintiff maintains that the “**SCSI Hotplug**” is one of the functions which ESXi uses (written pleading 25.09.2015 p. 8 = p. 152 of the annex). Within the SCSI subsystem there is a SCSI Hotplug functionality, i.e. the dynamic in- and exclusion of hardware during on-going operation, one of the most important functions (written pleading 25.09.2015, p. 8 = p. 152 of the annex). In its own right, the programming underlying this functionality enjoys copyright protection, because it is an example of a complex functionality and its implementation, leading to the *de facto* assumption that it has a sufficient degree of creative originality (op cit. p. 9 = p. 153 of the annex). This code was added to the Linux program in many steps and spreads over several individual functions (op cit.). The Plaintiff states that he made the **19 individual contributions (patches)** named in his written pleading 25.09.2015 p. 9 (p. 153 of the annex), and also refers to copies of these patches on the CD-ROM Exhibit K 12. An expert could confirm that complex programming is involved here (op cit.). Insofar as the Defendant has denied that the Plaintiff is the originator of individual patches, the Plaintiff has entered a submission on those parts of the modifications that he has claimed (for details: written pleading 25.09.2015 p. 6-7 under 11.4.a)-c) = p. 150 f. of the annex). The Plaintiff pleads further that on 21.02.2003 he contributed the **patch “scsi_scan.c restructuring for ieee1394 hotplugging**” to SCSI, which was adopted in Linux on 23.03.2003. The fact that his functionality has been adopted in “vmklinux” can be seen from certain callings of the functions “scsi_add_device" and “SCSl_remove_device” in all “vmklinux” SAS drivers and in the Libata module (written pleading 29.04.2016 p. 11= p. 275).

The Plaintiff makes further statements about parts of the SCSI functionality that are allegedly in “vmklinux”, are needed for multiple hardware drivers (so-called “**midlayer code**”), and have so-to-speak been brought before the court; this applies e.g. for the error handling function; another part of the SCSI functionality is to be found in the **hardware drivers** that are connected to “vmklinux” (written pleading 29.04.2016 p. 7 = p. 271 of the annex).

Finally, the Plaintiff states with reference to the SCSI that one function, namely “**scsi_remove_single_device**” has been adopted by the Defendant identically: it is to be found in the Linux kernel 2.5.64 in the file “drivers/scsi/scsi_proc.c” in lines 423 ff. (K 25), and in VWware ESXi 5.5 Update 2 in the file “vmdrivers/src_92/vmklinux_92/linux/scsi/scsi_proc.c” in lines 250 ff. (K 26); these are 99% identical (K 27) (written pleading 29.04.2016 p. 12-13 = p. 276-277).

The Plaintiff claims further that he also contributed substantially to work on the **Radix Trees**. Radix Trees are a scalable implementation of a special tree structure that is used in particular in the management of file-system buffers (caches) and also in many other areas of application. As a central part of the core of the Linux operating system, this functionality was never intended for exporting to modules, nor has it been exported to kernel modules by any users other than the Defendant (Statement of Claim p. 12). Finally, the Plaintiff pleads that the Linux Radix Tree is efficient implementation of a data structure for finding objects by means of an index. The data structure was invented by the programmer Velikov. Its implementation was the result of work done jointly by Velikov and the Plaintiff. The Plaintiff submitted the implementation for inclusion in Linux on 09.04.2002. The Defendant adopted a newer version of the code dated 2012 and made minimal modifications to it (written pleading 29.04.2016 p. 13-15 with reference to K 29-31). Other developers of the Linux kernel can testify that the Plaintiff made substantial complex contributions towards the development of the Radix Trees (written pleading 25.09.2015 p. 10 = p. 154).

Similarly, analysis has shown that the Defendant has used **many other parts of the Linux operating system**, for which the Plaintiff contributed smaller modifications of the source code (Statement of Claim p. 15 = p. 15 of the annex).

The Plaintiff alleges that the Defendant's use of the Linux code in "vmklinux" is illegal because the **license terms of the GPL-2.0 have not been complied with**. The GPL-2.0 requires in Item 2.b) – and to this extent the issue is undisputed – that the user *"must cause any work [...], **that in whole or in part contains or is derived** from the Program or any part thereof, to be licensed as a whole [...] under the terms of this License"*. The Plaintiff does indeed concede that the Defendant has disclosed the source code of the "vmklinux" module and licensed it under the GPL-2.0; but he states that this alone fails to meet the requirements under the GPL-2.0 for using the Linux code, and that licensing in accordance with Item 4 of the GPL-2.0 therefore no longer applies. This is because in this particular case, the clause quoted above has to be applied to reflect the fact that **"vmklinux" and "vmkernel" are to be regarded as a uniform work** and that they therefore have to be licensed collectively under the GPL-2.0, i.e. **the source code for "vmkernel" would also have to be disclosed by the Defendant** in order to satisfy the requirements of the GPL-2.0. The need to regard "vmklinux" and "vmkernel" as an entity ensues from the following features of the Defendant's program that are alleged by the Plaintiff:

- The architecture of the Defendant's software in this respect can be gathered from Exhibit K 2.

- The Linux code is loaded dynamically into the "vmkernel" component. This applies also and in particular for the SCSI subsystem, which contains the Plaintiff's code – whereby the code is executed in the kernel space, i.e. not in the user space, but as part of the operating system (Statement of Claim p. 15 = p. 15 of the annex).

- Both "vmkernel" and "vmklinux" are executed in the same address space, i.e. functions in "vmkernel" call functions in "vmklinux", which then in turn use functions in "vmkernel". The prevailing general understanding is that this is a typical feature of a uniform program, because it is typical for independent programs to be executed in separate address rooms, whereby each implements its own functionality which does not involve any far-reaching interaction between the components (Statement of Claim p. 15-16 = p. 15-16 of the annex).

- Unless the "vmklinux" module is used, the "vmkernel" cannot run with the modified Linux code (Statement of Claim p. 29). In the Defendant's ESXi program, the "vmkernel" cannot run on its own and additional software components are imperative if it is to be able to address any hardware. These software components are to be found in the kernel modules exported from the kernel, and in particular in the "vmklinux" module. Thus the "vmkernel" cannot run without the software exported to the modules (Statement of Claim p. 8).

- "vmklinux" cannot run without "vmkernel". The Defendant's aim was to be able to use the substantially modified Linux code in close integration with the "vmkernel". For this, it was not sufficient for the Defendant to use the adopted Linux modules in object code and merely compile them via an interface; instead, they had to be actually integrated in the kernel of ESXi as well, which meant specifically modifying and individually adjusting them in each case (Statement of Claim p. 13). Consequently, "vmklinux" cannot be used as part of Linux. In particular, the Defendant's "vmklinux" cannot be loaded into the Linux kernel (Statement of Claim p. 29) or used with any other operating system (Statement of Claim p. 13).

It is from this – i.e. the need for "vmkernel" and "vmklinux" to be regarded as an entity on the one hand, and the fact that parts of the code of the Radix Trees and the SCSI subsystem have been adopted in "vmklinux" on the other – that it follows in the Plaintiff's assessment that the VMware ESXi software has to be regarded as a "combined work" or "derivative work", i.e. as an adaptation of the Plaintiff's programming achievement (Statement of Claim p. 29).

The Plaintiff petitions

that the court order the Defendant as follows:

1.

On pain of an administrative fine of up to EUR 250,000 to be fixed by the Court in each single instance of violation, and – in the event that such fine cannot be collected – on pain of detention or of imprisonment for up to six months to be enforced on its permanent representative, the Defendant is to cease and desist making the kernel (i.e. including the components “vmkernel”, “vmklinux” and “VMK API” of the software Hypervisor vSphere VMware ESXi 5.5.0 (including Update 1 and Update 2) accessible to the public, unless at the same time, in accordance with the license terms of the GNU General Public License, Version 2,

o the complete corresponding source code of the kernel of the software Hypervisor vSphere VMware ESXi 5.5.0 (including Update 1 and Update 2) is made accessible royalty-free, and

o the kernel of the software Hypervisor vSphere VMware ESXi 5.5.0 (including Update 1 and Update 2) is offered under the license terms of the GNU General Public License, Version 2.

2.

The Defendant is ordered to pay the Plaintiff EUR 4,046 plus interest on that amount at a rate of five per cent over and above the respective base rate as from *lis pendens*.

The Defendant petitions

that the action be dismissed.

The Defendant pleads as follows:

The Plaintiff is trying to force the Defendant to *de facto* waive protection of its intellectual property in the product ESXi, by insisting that ESXi be made the object of an Open Source software license, the sole reason being that the central component of ESXi, the “vmkernel”, although it does not itself contain any Linux code, communicates with a separate module which in turn uses Linux code. The Plaintiff is not entitled to any such claim. In particular, the Plaintiff has neither shown to the requisite standard of proof that he is at all entitled to parts of the Linux code – let alone which ones – that have been used in “vmklinux”, nor is the Plaintiff’s argument for classifying the “vmkernel” as a “derived work” based on Linux within the meaning of the GPL-2.0 correct in either factual or legal terms. For the details:

The **action is not admissible,** quite simply because the petition is too imprecise and the Plaintiff has no interest in it being filed (for the details: Statement of Defence p. 48 f. = p. 102 f. of the annex; written pleading 15.04.2016 p. 9 ff. = p. 259 ff. of the annex). On top of which it is also **unfounded**.

The Plaintiff's pleading does **not establish a software work that merits protection**, let alone that any such work or **exactly which one has been created by him**. The reference to "linux.tgz-Archiv" and the "git repository" do not suffice either; the Plaintiff does not even make the effort to identify the individual code, so that it can only be guessed which lines of the code the Plaintiff wants to use for substantiating his claim (written pleading 05.02.2016 p. 7-8 = p. 196-197).

Insofar as the Plaintiff points to the **SCSI subsystem** for proof of his authorship, this pleading is unsubstantiated because the Plaintiff fails to explain which specific files of the Linux kernel in his opinion constitute the SCSI system (Statement of Defence p. 11 = p. 68 of the annex). It is also unclear exactly what the Plaintiff contributed to it (Statement of Defence p. 49 f. = p. 103 f. of the annex).

When the Plaintiff refers to the **Radix Trees** for proving his authorship, this pleading is unsubstantiated as well. Radix Trees are decade-old file structures for rapidly calling files. The Plaintiff has not explained which code he has written, or that this code is a sufficiently creative work to merit copyright protection (Statement of Defence p. 11= p. 68 of the annex). Moreover, as the Plaintiff himself has stated, Radix Trees are merely structures; but according to jurisdiction by the ECJ, mere file structures are not computer programs that are protected by copyright (Statement of Defence p. 50 = p. 104 of the annex).

The Plaintiff's reference to a **repository** to prove his authorship of a certain code is likewise insufficient: the Plaintiff's pleading that he "contributed" to or was "involved" in the code does not state sufficiently specifically what the Plaintiff is meant to have created. On the contrary, the Defendant's analyses have revealed that in actual fact other programmers wrote the entire code or at least parts of the code for which the Plaintiff is claiming authorship (Statement of Defence p. 13 f. = p. 67 f. of the annex with examples and details).

The Defendant **denies the Plaintiff's alleged contributions**, pleading ignorance (Statement of Defence p. 26 = p. 80 of the annex). It also denies that any such contributions **merit copyright protection** (written pleading 05.02.2016 p. 8 = p. 197 of the annex).

But even if the Plaintiff did create parts of the Linux kernel, he can have acquired **only adapter's copyright** in that respect, and his rights as a holder of adapter's copyright would then be limited to the parts he modified (Statement of Defence p. 53 = p. 107 of the annex). The Plaintiff cannot assert any rights that go beyond his own contributions (written pleading 05.02.2016 p. 12 f. = p. 201 f. of the annex)

The Plaintiff's **pleading fails to establish an infringing use of code**, namely that the Defendant's software does in fact contain original and/or derived works or parts of works in which the Plaintiff has rights, or that "vmkernel" is derived from the Plaintiff's parts of Linux that merit protection (Statement of Defence p. 54 = p. 108 of the annex).

The Defendant's product **ESXi** is not a single executable software program, instead of which it is a software product that comprises numerous works as well as components, functions and features; the Plaintiff has not explained that or how the Defendant has offered ESXi for downloading via the Internet (Statement of Defence p. 29 = p. 83 of the annex).

Insofar as the Plaintiff alleges that the Defendant has modified the Linux code so as to be able to use it more closely integrated with the proprietary "vmkernel" component, this pleading lacks substantiation; in addition to which, the Defendant used Linux kernel code in order to develop a technology that has an entirely different purpose from that of the Linux kernel, namely in order to create an interoperability module, which is allowed under the GPL (Statement of Defence p. 35 = p. 89

of the annex). Any SCSI code in “vmklinux” serves a different purpose from that of the **SCSI subsystem** in Linux (Statement of Defence p. 34 = p. 88 of the annex).

Also insofar as the Plaintiff refers to the **SCSI hotplug functionality**, the ability of a system to exchange components without having to shut down the system has been neither invented nor created by the Plaintiff, nor is it a feature that is implemented in a single function. The **patches** to which the Plaintiff refers in this context are not relevant pointers, because they relate to the Linux kernel but not to “vmklinux”, and so it cannot be concluded from them whether the code that the Plaintiff claims to have written is something meriting protection that has been used by the Defendant. In actual fact, the functions in “vmklinux” that relate to the hotplug functionality do not contain any essential code originating from the Plaintiff (written pleading 05.02.2016 p. 11= p. 200 of the annex).

The Defendant continues by stating that in all other respects, it may be assumed that any of the Plaintiff’s contributions to the Linux code, insofar as they are claimed to have been used in “vmklinux” and thus in ESXi, are so small in scope that they fade so far into the background that in relation to the entirety of “vmklinux”, “vmkernel” and “ESXi” this is **not an adaptation or modification**, but **only – if anything – free use** (Statement of Defence p. Item 14 = p. 68 of the annex; written pleading 05.02.2016 p. 20 ff. = p. 209 ff.). The Defendant has made substantial modifications to the Linux code (Statement of Defence p. 15 = p. 69 of the annex):

- On analysing the information presented by the Plaintiff, the Defendant has managed to identify code from “vmklinux” that the Plaintiff claims to have written in about 798 lines. “vmklinux” in itself comprises about 214,000 lines of code (i.e. 269 times as much); “vmkernel” comprises 1.3 million lines of code (i.e. 1,654 time as much) (written pleading 05.02.2016 p. 4 = BL. 193 of the annex).
- Of the 798 lines of code however, 396 lines have not been used in “vmklinux”, instead of which they have been commented out (i.e. they only appear in human-readable source code) (written pleading 05.02.2016 p. 6 = p. 46 of the annex).
- Of the remaining 402 lines of code, only a little less than half – namely 185 lines – are code that does not originate from the Plaintiff, but which he has only slightly modified or moved to another place (written pleading 05.02.2016 p. 6 = p. 46 of the annex).
- Then of the remaining 217 lines of code, 68 are mere “comments” (e.g. explanations of the functionality of the code) that are not compiled in the final machine-readable and executable code either (written pleading 05.02.2016 p. 6 = p. 46 of the annex).
- Thus this **ultimately leaves just 149 lines that may possibly have originated from the Plaintiff** and reached the end user (written pleading 05.02.2016 p. 6 f. = p. 46 f. of the annex, also for the following). Just the three “vmklinux” files to which the Plaintiff refers already contain 6895 lines of code, to which the Plaintiff has therefore contributed less than 2.2% in terms of volume. Thus the Plaintiff has then only contributed 0.07% to “vmklinux” as a whole with its 214,000 lines of code. If “vmkernel” and “vmklinux” were to be regarded as an entity, which the Defendant denies, the Plaintiff would have contributed **less than 0.012% to “vmkernel”** with its 1.3 million lines.

There has been **no infringement of the license terms of the GPL-2.0** (Statement of Defence p. 56 ff. = p. 110 of the annex). The Defendant has (indisputably) licensed the source code of “vmklinux” as such in compliance with the terms of the GPL (Statement of Defence p. 9 = p. 63 of the annex). Under these terms, the source code of “vmkernel” does not have to be disclosed. The terms do indeed stipulate that the source code of a “derivative work”, i.e. a work derived from a freely licensed work, has to be disclosed. But **“vmkernel” is not a work derived from Linux**.

“vmkernel” is an independently developed work, the development of which the Defendant commenced over 15 years ago (Statement of Defence p. 2 = p. 56 of the annex) and for which it has made substantial investments. Time-wise, “vmkernel” as such was (indisputably) developed by the Defendant before the Plaintiff created his alleged software work (Statement of Defence p. 10, 16 = p. 64, 70 of the annex). The Defendant denies that “vmkernel” cannot run on its own, instead needing additional software components, and that these have been taken from Linux (Statement of Defence p. 29 = p. 83 of the annex). Apart from which, no operating system kernel can run or has any practical use without device drivers; from the fact that every kernel needs device drivers, it cannot therefore be concluded that it is then automatically a derived work within the meaning of the GPL (Statement of Defence p. 64 f. = p. 118 f. of the annex).

“vmklinux” is an independent component likewise developed by the Defendant; it enables communication between “vmkernel” and device drivers compatible with Linux, such that third-party hardware manufacturers could now continue to use the device drivers with the “vmkernel” which they had originally developed for the Linux kernel (Statement of Defence p. 3 = p. 57 of the annex). Like other device drivers (both ones compatible with Linux and others) that communicate with the “vmkernel”, “vmklinux” is an **independent module separate from “vmkernel”** (Statement of Defence p. 7 = p. 61 of the annex); it has deliberately been developed by the Defendant as a **device driver interoperability module** (for details: Statement of Defence p. 15 = p. 69 of the annex). To state that “vmklinux” cannot run without “vmkernel” is both irrelevant and misleading (Statement of Defence p. 36 = p. 90 of the annex). Whether a component can run on its own is not decisive; thus application programs for instance depend on operating systems. When the Plaintiff states that the general assumption is that kernel modules have to be regarded as part of the kernel if they are not capable of running – also in unaltered form – with other Unix-type operating systems, this is a non-binding subsequent interpretation of the GPL; whether it applies for the Linux kernel and Linux kernel modules is irrelevant, it does not at any rate apply for “vmkernel” (Statement of Defence p. 39-40 = p. 93-94 of the annex). Apart from which, Linux kernel modules do not form a uniform work together with the Linux kernel either (Statement of Defence p. 32 = p. 86 of the annex).

Thus **“vmkernel” and “vmklinux” are not a uniform program.** Instead, they both exist (and to this extent this is not disputed) as two separate binary files. “vmkernel” is the core component of ESXi, whereas “vmklinux” is a much smaller separate module designed to enable device manufacturers to continue using their Linux-compatible device drivers with ESXi.

Thus “vmkernel” by all means communicates with the “vmklinux” module. “vmkernel” has an entirely different architecture from that of the Linux kernel, because device drivers are not integrated in the kernel (as is the case with the Linux kernel), instead of which they have to be addressed through a stable documented interface, the “VMK API” (Statement of Defence p. 17 = p. 71 of the annex). Just the mere fact that “vmkernel” communicates with the independent “vmklinux” module and this module contains Linux code does not mean that “vmkernel” also has to be regarded as being derived from Linux (Statement of Defence p. 4 = p. 58 of the annex).

This **communication between “vmkernel” and “vmklinux”** does not take place via instable internal kernel interfaces (Statement of Defence p. 64 = p. 118 of the annex), but via the **“VMK API” interface**, which (other than is maintained by the Plaintiff) is a **stable and documented** interface (Statement of Defence p. 3 = p. 57 of the annex). “VMK API” is an interface used by numerous components and device drivers; it is stable and **downwards-compatible** as a matter of principle; currently a total of about 122 device drivers and other modules have been developed by 21 businesses which all communicate with “vmkernel” via “VMK API” (Statement of Defence p. 15 f. = p. 69 f. of the annex). Thus “vmkernel” can address via “VMK API” and native drivers, without the communication going via “vmklinux” (e.g. Hewlett Packard printers); to this extent, “vmkernel” is not dependent on “vmklinux” (written pleading 15.04.2016 p. 4 = p. 254 of the annex). Thus “VMK API” is a genuine interface because it is not only used for communication between “vmklinux” and “vmkernel”, but **also used by hundreds of other modules** that have been developed by third parties as well as by the Defendant; in addition, “VMK API” has the typical features of a software interface (written pleading 05.02.2016 p. 29 ff. = p. 218 ff. of the annex). The device drivers which communicate with “vmkernel” are separate components (Statement of Defence p. 31 = p. 85 of the annex). “VMK API” was developed by the Defendant as early as 2004 and has been marketed by it since 2006, whereas the version of the Defendant’s product which in the Plaintiff’s view contains the alleged “SCSI subsystem” has only been marketed since May 2009, and the version which allegedly contains the Radix Tree code has only been marketed since 2013 (Statement of Defence p. 16 = p. 70 of the annex). “VMK API” thus provides a **clear dividing line** between the proprietary software of the “vmkernel” and the “vmklinux” module (Statement of Defence p. 3 = p. 57 of the annex). Especially when a connection is made through such an interface as this, separate programs and thus separate works are to be presumed (written pleading 05.02.2016 p. 21 = p. 2010 of the annex). If this were to be seen differently, it would lead to the absurd situation that all owners of third-party modules that communicate with “vmkernel” through the interface would likewise be in a position to claim copyrights in “vmkernel” (written pleading 05.02.2016 p. 2 = p. 192 of the annex).

The Plaintiff’s argument that the interface divides “vmkernel” and “vmklinux” artificially is to no effect, because “vmklinux” has a specific functionality that is quite separate from that of “vmkernel”, namely to enable Linux-compatible device drivers to communicate with “vmkernel” (written pleading 05.02.2016 p. 27 = p. 216 of the annex). The Plaintiff also confuses the interfaces he is speaking about: the fact is that apart from the “VMK API” interface (through which “vmkernel” and “vmklinux” communicate) there is **another interface**; however, this is the one **through which “vmklinux” communicates with the device drivers**, whereby this interface is not instable either in the sense maintained by the Plaintiff (written pleading 15.04.2016 p. 5 = p. 255 of the annex).

The additional fact that **“vmklinux” is executed in the kernel space** does not mean that “vmklinux” and “vmkernel” have to be regarded as uniform software, because other kernel extensions and device drivers are likewise executed in this space (Statement of Defence p. 19 = p. 73 of the annex). Nor does any prevailing opinion exist to this effect (Statement of Defence p. 31 = p. 85 of the annex). Whether two software components are executed in the same memory space or in different memory spaces [aus dem – missing word(s)] is a technical matter of no import for an assessment in terms of copyright law (Statement of Defence p. 38 = p. 92 of the annex).

Amongst other things, the Defendant also pleads **forfeiture**; after all, the Plaintiff waited several years before bringing legal action, despite his assumed awareness of the facts (Statement of Defence p. 67 = p. 121 of the annex; written pleading 13.07.2015 p. 3 = p. 132 of the annex; written pleading 05.02.2016 p. 43 = p. 232 of the annex).

For further details regarding the facts of the matter and the status of the dispute, reference is made to the written pleadings exchanged by the parties, insofar as they were declared the subject-matter of the hearing concluded on 25.02.2016, and to the subsequently admitted procedural documents that were received by 15.04.2016 and – following extension of the deadline – by 29.04.2016. Besides this, the Defendant also submitted a written statement dated 10.05.2016 for the file, but this was not subsequently admitted to the proceedings.

Reasons for the Decision

In the final analysis, the action is to no avail.

A.

The action is however **admissible**.

Hamburg District Court has **international and local jurisdiction** pursuant to Code of Civil Procedure § 32, because the Plaintiff pleads that the program that is at the centre of the dispute has been offered by the Defendant via the Internet for downloading in Germany (doubly pertinent fact).

The action is also sufficiently **precise within the meaning of Code of Civil Procedure § 253**. The admissibility of the action is not affected by the Defendant's objection in this respect, namely that the Plaintiff already has to specify in the petition exactly which parts of the Defendant's program the Defendant should not use. For the details:

As discussed at the hearing, the only property rights from which the Plaintiff might possibly derive the right to claim forbearance are adapter's copyrights, because the Plaintiff states that he successively made independent contributions to the Linux kernel. It follows from Copyright Act § 69c No. 2 clause 2 in conjunction with § 3 that adapter's copyright can also be created when software is modified. Whether any of said contributions was created in collaboration with others does nothing to alter this fact, because then the community of originators would also be entitled to a mere adapter's copyright and the Plaintiff thus to "co-adapter's copyright".

The fact that the Plaintiff's status as a holder of adapter's copyright may only relate to part of the Linux kernel does not in principle conflict with the assumption that he holds his own adapter's copyright for each of these parts respectively. *"Whereby parts of computer programs can also merit protection if they in turn meet the criteria for copyright protection [...], which may be the case if they are of not only minor importance for the running of the program and have their own command structure, which to the extent of given creative freedom is an expression of individual creative work."* (Hamburg Higher District Court, ruling dated 11 January 2001 - 3 U 120/00 -, text item 37, juris). Thus adapter's copyrights may also relate to parts of software, if the programmer's adaptation work in turn meets the aforementioned criteria.

However, on the premise initially assumed here – namely that the Plaintiff acquired relevant adapter's copyrights in parts of the Linux kernel – the Plaintiff's legal position would be limited. Thus it has been stated, e.g. in Schicker/Loewenheim, Copyright Law, 4th ed. 2010, § 69c text item 20: "*The object of protection under legislation on adapter's copyright is only the modification as such; the adapter does not acquire any rights whatsoever in the original program."* As far as can be seen, there is no controversy about this in jurisdiction and in relevant literature. This is what the Plaintiff himself also ultimately assumed, when he says in the forum posting dated 2006 which he quotes (Exhibit K 19) that unfortunately he did not have sufficient copyrights in the Linux version as it stood at the time to be able to take action against the Defendant.

Nonetheless, a holder of adapter's copyright can still demand forbearance of the use of any such other computer program, if it avails itself in a non-free manner of the parts protected for said rightsholder within the meaning of Copyright Act § 69c Nr. 2 (adaptation) in conjunction with § 23 (modification). The petition for a cease-and-desist order that has to be formulated in court proceedings is thus sufficiently precise if the program version incorporating the infringement of rights is sufficiently precisely named. In the case of individually developed programs that are not traded commercially, instead of which they are only used by a single client and are continuously adjusted, the wording frequently poses difficulties that make it necessary to already specify the essence of the infringement in greater detail in the wording of the petition. That problem does not arise here though, because the Defendant's program – and it does not deny this – can normally be purchased. The Plaintiff has sufficiently precisely named the version that is meant to incorporate the infringement of rights, meaning that in any enforcement procedure it could be verified whether marketing of the version specified (or a version assessed as having the same core) has continued despite the prohibition. Thus the criteria for sufficient precision have been met.

The additional fact that any legal right to an injunction on the Plaintiff's part can likewise only be limited does not conflict with the sufficient precision of the wording of the petition for a cease-and-desist order. It is indeed conceivable that after a prohibition was issued, the Defendant might alter its program and could then claim that the new version of the program was no longer covered by the enforceable prohibition. However, this is not a question of the precision of the wording of the petition for a cease-and-desist order, but a question of its reach according to the core theory, which states that the enforceable prohibition also covers those infringing acts, which reflect the characteristically infringing aspect of the infringement that was examined in the procedure leading up to issuance of the prohibition and formed the basis for the prohibition order. Which aspect is characteristic can and where necessary must be determined by means of interpretation; for this, recourse can and where necessary must be had to the explanatory remarks contained in the reasons for the decision.

A plaintiff who fails to state with sufficient precision the substantive essence of the injunction he is pursuing, may run the risk of the court being unable to establish any infringing act at all from his insufficiently substantiated pleading (in which case the action is admissible but unfounded), or only being able to establish one out of possibly several infringing acts (in which case the action is admissible and well-founded, but the focal point of the prohibition is narrowly defined because it centres solely on the one infringing act that could be established).

Whether a plaintiff also restricts the petition for an injunction right from the start to just one infringing aspect already specified in the petition, thus possibly delimiting the disputed issue from more far-reaching grounds, remains up to the party. And whether, for the sake of the ruling's clarity, a restriction has to be included in the operative part of the decision in order to describe the prohibition more exactly, although no such restriction has been made in the petition, is a matter which has to be considered by the court on an individual basis.

All this does nothing to alter the fact however, that the prohibition being sought in the case instant has been stated by the Plaintiff with sufficient precision by naming the Plaintiff's program.

B

The action is unfounded nonetheless. The Plaintiff is not entitled to the injunction being claimed in Petition no. 1) pursuant to Copyright Act § 97 I in conjunction with § 69c No. 2. For this reason, the claim to reimbursement of costs for issuing a warning plus interest pursuant to Copyright Act § 97a and Code of Civil Procedure § 288 and § 291, which is asserted in Petition no. 2), does not exist.

1.

As has already been stated with regard to the action's admissibility, the Plaintiff's legal position in the case instant can only be based on adapter's copyrights, because the Plaintiff claims that he has successively made his own contributions to the Linux kernel. The fact that adapter's copyright can also be brought about in a software modification follows – as has already been stated – from Copyright Act § 69c No. 2 clause 2 in conjunction with § 3. As a starting point, this is not legally disputed by the parties to the litigation either.

On the premise that the Plaintiff acquired relevant adapter's copyrights in parts of the Linux kernel, the Plaintiff's legal position – as has likewise already been stated – would however be limited. When considering the merits of the claim, the court therefore has to examine whether the Plaintiff's pleading is sufficiently substantiated to establish the cause of action and whether the Plaintiff has proved where necessary:

- which parts of the Linux program he claims to have modified, and in what manner;
- to what extent these modifications meet the criteria for adapter's copyright pursuant to Copyright Act § 69c No. 2 clause 2 in conjunction with § 3; and
- to what extent the Plaintiff pleads and where necessary proves that the Defendant has in turn adopted (and possibly further modified) those adapted parts of the program that substantiate his claim to protection.

Then – but only if it can be established that code parts have been used which are protected for the Plaintiff's benefit – further steps would have to follow, in particular to examine the following:

- Can “vmklinux” and “vmkernel” be regarded as a uniform work (and what legal criteria are to be applied for determining this uniformity?) For if “vmklinux” has to be regarded as a work that is separate from “vmkernel”, the parties concur that the Defendant was only obliged to disclose the “vmklinux” code in order to comply with the GPL-2.0 requirements and that it fulfilled this condition.

- If “vmklinux” and “vmkernel” could be regarded as a uniform work, there would finally have to be an examination of the significance for the “vmkernel+vmklinux” entity then having to be assumed of the Plaintiff's parts of the code adopted in “vmklinux”, and whether their use – seen from this perspective – still constitutes an adaptation or modification within the meaning of Copyright Act. § 69 No. 2 in conjunction with § 23, or whether it already constitutes free usage within the meaning of Copyright Act § 24 (and what legal criteria would have to be applied for determining this). Particular attention would have to be paid here to the fact that the Plaintiff's legal position were only that of a holder of adapter's copyright, and that the Defendant has strongly argued that in terms of quantity, the scope of any of the Plaintiff's code it may have adopted is extremely small in relation to the total volume of “vmkernel+vmklinux”, without it being possible to deduce from the Plaintiff's pleadings that the dimensions claimed by the Defendant might be entirely inaccurate.

Nonetheless, these questions (on which the legal interest of the parties and their counsel presumably focus) can and must remain unanswered. This is because the very first requirement for conducting an examination, namely that code possibly protected for the Plaintiff as a holder of adapter's copyright has been used in the Defendant's product, cannot be established. This is still true even after taking into account the Plaintiff's subsequently admitted procedural document dated 29.04.2016, in which he (after the deadline had been extended) had a further opportunity to enter a pleading on the reservations in this respect which the court had already expressed at the hearing.

1.

The **Plaintiff's general remarks about his involvement in work on the Linux kernel** and the possibility of researching this in publicly accessible sources do not satisfy the procedural requirements for a plaintiff's statement of the cause of action.

a)

Thus the Plaintiff has pointed out that in a **git repository** the parts of the Linux code which originated from him, the Plaintiff, could be publicly viewed; moreover (on p. 15 of the Statement of Claim), that in the repositories of the mainline kernel (i.e. the official version of the kernel) details of each of the Plaintiff's development contributions could be verified and documented. A broad reference such as this to the possibility of investigating in the Internet the facts that have been submitted does not constitute an admissible pleading in court procedure.

b)

Nor does presenting the allegedly complete **source code of Linux** Version 2.6.12-rc2 together with the changelog (written pleading 25.09.2015 p. 10) satisfy the requirements for a procedurally verifiable pleading on those parts that are pertinent here to the Plaintiff's adapter's copyright; this is because it is not the job of the court or of the opposing party to pick out from an entire source code any parts of code that might originate from the Plaintiff, and to judge for themselves to what extent and for which parts and related issues the Plaintiff might be seeking protection under copyright law.

The Plaintiff's broad reference to the CD-ROM **K 12**, stating that it contains those parts of his code which turn up again in "vmklinux" (written pleading 25.09.2015 p. 10), does not suffice either as a statement of the infringement of rights. The Plaintiff would have to specifically name those parts that have been used. It would also be possible for him to do so, because it is undisputed that the Defendant has disclosed the source code for the "vmklinux" module. Submission of the file "**linux.tgz**" with an archive on the CD-ROM K 12 is not sufficient in this context either, because here again the court and the opposing party would have to find out for themselves which parts are meant to originate from the Plaintiff.

When the Plaintiff maintains that **version 2.6.18 of the Linux kernel** is the one "most similar to the version used by the Defendant" (Statement of Claim 25.09.2015 p. 10), this pleading again fails to establish use of the Plaintiff's parts of the code as the cause of action, because it neither specifies individually which constituent parts are meant, nor states exactly where in the Defendant's code these parts turn up; the version "used" by the Defendant evidently means a Linux version, not a version of "vmklinux". Nor is this pleading made more specific in the procedural document dated 29.04.2016, top of p. 10 (p. 274 of the annex), where the Plaintiff merely refers to the details given in the Statement of Claim.

When the Plaintiff sums up by stating that with K 12 he has submitted the code he contributed to Linux that turns up again in "vmklinux", it does not suffice for pleading his adapter's copyright in Linux to submit the entire code of this program and merely state that it also contains the modified parts which originate from the Plaintiff; for the court and the opposing party this is not a verifiable pleading as to the constituent parts in which the Plaintiff is seeking to assert adapter's rights.

c)

Insofar as the Plaintiff has maintained in his written submissions that the **"blame" files** (or the PDFs of the same content) submitted with CD-ROM **K 12** show which code originates from the Plaintiff and which modifications the Defendant has made in “vmklinux”, that is not entirely accurate and verifiable, as is already evident from the Plaintiff’s pre-litigation email K 10.

However, the folder “email dated 14.11.2014” on the CD-ROM K 12 contains several PDF files, which according to K 10 correspond to the “blame” files that are likewise attached. The “blame” files allegedly specify which parts of the Linux code originate from the Plaintiff; this latter claim appears to be correct, because when the PDFs are opened lists with lines of code appear, whereby the respective originator is named for each line of code. But this can only be understood to mean that the adapter’s copyrights of all the various originators in Linux are involved (including the Plaintiff who is also listed). Whatever the case, no comparison is made in these lists with the “vmklinux” code from the Plaintiff’s program.

A comparison to this effect is to be found instead in the html-files likewise mentioned in the **email K 10**. With reference to them, it is stated in K 10 that these files show the result of a comparison between the “Linux file (Version 2.6.18) and your client’s file”; what is apparently meant is a comparison between the Linux program and the Defendant’s program. Moreover, K 10 reads, *“The files that have been compared can be gathered from the file names. In the files you will find code marked black, green and red. The black code is code that is contained without any modification both in the original Linux version and in your client’s modified version. The code added by your client is marked green, whilst the code marked red is Linux code that no longer exists in your client’s version. "scsi_error.c" and "scsi_proc.c" have substantial matches. They appear less in the other files, because code from several Linux files has been merged into one file.”* This is verifiable to the extent that there are indeed lines of code marked black, green and red in the html-files. The Plaintiff claims that the code marked black and (with modifications) green has been used. However, the html-files do not show from which Linux programmer the Linux code that is marked black and green actually originates. In particular, there is no indication as to which parts of the code marked black and green in the html-files are meant to originate from the Plaintiff. Thus the reader would have to go about finding this out for himself, by comparing the black and green parts of the html-files with the PDFs containing the “blame” files. This at any rate is not a procedurally verifiable pleading, particularly since not even a mere allegation indicating the cause of action, to the effect that “vmklinux” incorporates certain lines of code from the Plaintiff, emerges from a pleading thus made.

d)

When the Plaintiff maintains further that a full **comparison of the source codes has been conducted, which can be verified e.g. from Exhibit K 15,** the court can only judge the examples provided with K 15; beyond this, the pleading with regard to adopted elements is unsubstantiated, quite simply because the Plaintiff fails to submit the result of the alleged comparison and offers no statement on it.

But the Plaintiff has not managed to prove any infringement of adapter’s copyright in respect of the parts of the code asserted with K 15 either. The Defendant has convincingly argued that ultimately, only eight specific and separable functions of the SCSI subsystem can be detected from K 15; the Plaintiff has not contested this.

The Defendant has argued moreover that one of these eight functions has been “commented out” in “vmklinux”, i.e. it is not used in “vmklinux”; and that in the remaining seven functions, more than half the lines of code have either been written by another originator (without any contribution from the Plaintiff) or have simply been modified or at the most shifted by the Plaintiff (written pleading 05.02.2016 p. 14 = p. 203).

In this respect therefore, the Plaintiff should have specified more exactly his shares in the functions claimed with K 15 and furnished proof of his adapter's copyright. The Plaintiff has failed to do this in an adequate manner:

- No evidence in this respect is furnished in his direct written submission on K 15 (written pleading 25.09.2015 p. 11 = p. 155 of the annex).

- K 15 itself is not a printout of an extract of the code or a published changelog of an adaptation; instead, it is an overview – presumably compiled by the Plaintiff himself – of the eight individual functions being claimed in it; thus Exhibit K 15 acquires no evidential value at all as regards the Plaintiff's authorship.

- Insofar as the Plaintiff has pointed out the general possibility of verifying the history of the adaptation of Linux in the Internet, this pleading lacks sufficient precision as regards the questions of exactly where the changelog of the adaptation of the eight specific functions has been published, and what information is given there in relation to the Plaintiff's specific shares in these eight functions.

- This also applies however for the files presented by the Plaintiff with K 12, which are meant to contain the history of the respective version (history.tgz; linux.tgz). The Plaintiff thus asserts certain of these files' statements regarding the eight functions concerned, but since he does not do so in a sufficiently verifiable manner, their value as an indicator of the Plaintiff's authorship cannot be assessed.

- But even if the changelogs were to name the Plaintiff as the most recent adapter, the Defendant has correctly argued that changelogs are not copies of the computer program concerned, so that no presumption of authorship pursuant to Copyright Act § 10 can result from them. Nonetheless, any indicative value that might still be assumed would have to be substantiated by the Plaintiff and pleaded in relation to each respective function. The Plaintiff has not done this.

e)

When the Plaintiff pleads in his Statement of Claim (p. 14) that two files from "vmklinux", for which the Defendant has disclosed the source code, contain **headers with references to copyrights** held amongst others by the Plaintiff, it cannot be deduced from this pleading alone that parts of the Plaintiff's code have been used that merit being protected separately under (adapter's) copyright. This cannot simply be assumed merely because the Defendant itself has named the Plaintiff as a possible modifier of Linux; instead, it must be established by the Plaintiff.

2.

Similarly, an infringement of the Plaintiff's adapter's copyrights cannot be ascertained from the Plaintiff's statements maintaining his **involvement in work on the SCSI subsystem.**

a)

When the Plaintiff pleaded that he had indications that part of the SCSI functionality was also to be found **in the "vmkernel"**, he at the same time conceded himself that since he had no further clues, he would have to assume that they were the Defendant's own developments; thus according to his own pleading, the Plaintiff expressly is not claiming here that his own Linux code had been adopted.

In this respect as well therefore, as has already been said, any examination can be restricted to the question of code used in “vmklinux”.

b)

Insofar as the Plaintiff initially maintains **generally** that during the period from 2000 – 2004 he was one of the most active developers doing work on the SCSI subsystem of the Linux kernel, by all means involving complex programming that merits protection (written pleading 25.09.2016 p. 8 = p. 152 of the annex), this general claim does not suffice to satisfactorily plead achievements by the Plaintiff that merit protection under copyright law.

The Plaintiff cannot refer to the complexity of the overall SCSI subsystem for arguing the protectability of his own contributions, because he concedes that he only helped to adapt this, too; so since he only asserts adapter’s copyright here as well, he needs to specifically name those programming achievements for which he seeks protection.

The offer to question other kernel developers about the Plaintiff’s contributions to the Linux kernel serves the purpose of investigating facts; it does not constitute a verifiable pleading on those parts.

c)

When the Plaintiff claims in particular that the **SCSI Hotplug** is one of the most important functions used by ESXi (written pleading 25.09.2015 p. 8 = p. 152 of the annex), the question as to whether the Plaintiff has satisfactorily pleaded his performance as adapter-originator by submitting the PDFs for **19 patches on CD-ROM K 12**, can remain unanswered. What is not satisfactory at any rate is his pleading on possible use of his parts of the code in “vmklinux”. The Defendant has expressly denied using relevant code of the Plaintiff in connection with the Hotplug functionality (written pleading 05.02.2016, p. 11= p. 200 of the annex).

On this issue, the Plaintiff’s procedural document dated 25.09.2015 merely states on p. 8 that the Defendant’s program ESXi uses the “function” of the “SCSI Hotplug”. Thus only a corresponding functionality is claimed, but not that this functionality is also based on identical program code. Even less can it be deduced from this pleading that ESXi, and in particular “vmklinux”, contains that program code which incorporates those very same 19 contributions by the Plaintiff that can be seen in the “Patches” folder on CD-ROM K 12 and for which he seeks adapter’s copyright – and especially not these specific contributions in an unmodified form. But a claim to this effect would have been necessary in order for an infringement, especially of the Plaintiff’s adapter’s copyright, to be established as the cause of action. The fact that submission of Exhibit K 15 is insufficient here as well can be correspondingly deduced from the deliberations made in Item 1.d) above.

The Plaintiff’s statement (written pleading 29.04.2016 p.11) on the **patch „scsi_scan.c restructuring for ieee1394 hotplugging”** (= Patch No. 11 in written pleading 25.09.2015 p. 9 = p. 153 of the annex) is not sufficient either. For one thing it is not clear to what extent this patch involves complex programming indicative of creative originality. Whatever the case though, it is not stated here either that it was precisely the Plaintiff’s programming that was adopted in “vmklinux”; instead, talk is of calling up two separate functions “SCSl-add_device” and “SCSl-remove_device” from which it is meant to be possible to recognise the code used. That is not sufficient. The Plaintiff should have specifically pleaded which code in “vmklinux” is supposed to originate from himself.

When the Plaintiff in his written pleading dated 25.09.2015 (p. 6-7 under 11.4.a)-c)) states that some of the lines of code discussed there did indeed originate from other adapters, without altering his own copyright work, the pleading on the Plaintiff’s own achievement remains unverifiable. He claims in Item a) of said procedural document that he created the contribution **(patch) “fixes and cleanups [etc.]”** (= Patch No. 10 on p. 9, op cit.), without setting forth this contribution in any further detail. In Item b), the Plaintiff concedes that he merely modified third-party code at the place concerned in order to be able to use another interface, something that is relevant in terms of copyright law; but he neither submits the code he modified, nor does he explain to what extent adapting the code to an interface exceeded average programming work. In Item c), the Plaintiff claims that he created the **contribution (patch) “scsi_device refcounting [etc.]"** (= Patch No. 19 on p. 9, op cit.); but there are no explanatory comments on its protectability under copyright law either. Merely as a precaution, it should also be pointed out that the Plaintiff does not expressly claim – either at this stage or later – that it is these very parts of code being discussed at this juncture that are also contained in the Defendant’s “vmklinux”.

d)

The Plaintiff’s pleading on the **midlayer code** within the SCSI functionality (written pleading 29.04.2016 p. 7) is not enough to establish the Plaintiff’s rights either, because the Plaintiff does not claim that he was the originator of this midlayer code.

The same applies to his statements (op cit.) on the SCSI functionality in **hardware drivers** connected to “vmklinux”.

e)

Insofar as the Plaintiff finally comments on an SCSI function **“scsi_remove_single_device”**, this pleading differs from the Plaintiff’s other submissions on the SCSI subsystem discussed hitherto, in that here the Plaintiff verifiably claims the origin of specific code and its use in “vmklinux”, namely by listing specific places in files where it can be found (in the Linux kernel 2.5.64 – in the file “drivers/scsi/scsi_proc.c” in lines 423 ff. (K 25); in VWware ESXi 5.5 Update 2 – in the file “vmdrivers/src_92/vmklinux_92/linux/scsi/scsi_proc.c” in lines 250 ff. (K 26); cf. (written pleading 29.04.2016 p. 12-13); this proves that it is indeed possible for the Plaintiff to make such a pleading.

On the other hand, the Plaintiff’s pleading on the copyright protectability of the Linux SCSI code that he claims to have created is insufficient. The Plaintiff fails to explain the function either as such or within the SCSI subsystem, let alone within Linux, nor does he do so for the function within “vmklinux”. The Plaintiff does not state to what extent the programming or adaptation are distinct from purely manual programming work or are meant to be sufficiently complex in their own right. And the expert evidence he offers merely serves the purpose of investigating facts.

3.

Also when the Plaintiff claims that he helped to **develop the Radix Trees**, this pleading for a start fails to establish whether this is even a computer program within the meaning of Copyright Act § 69a I. No verifiable pleading has been submitted at any rate regarding any parts of the Radix Tree code that merit protection for the Plaintiff and have been taken over by the Defendant.

The understanding of the term “computer program” in Copyright Act § 69a calls for an interpretation in conformity with EU directives. Ruling on the interpretation of the Computer Programs Directive (Council Directive 91/250/EEC), the ECJ has decided (ECJ ruling dated 2 May 2012 - C-406/10 - = GRUR 2012, 814 - SAS – quoted from juris, text item 46) that Art. 1 Para. 2 of the Directive is to be interpreted to the effect that neither the functionality of a computer program, nor the programming language or file format used in a computer program in order to use certain functions of that program, are a manifestation of the program, and that they are not therefore covered by the copyright protection afforded to computer programs within the meaning of said Directive. This dictum is preceded by the ECJ making the following distinction (quoted from juris, text items 42 and 43):

> *42. As far as the programming language and the file format are concerned that are used in a computer program in order to interpret and execute the application programs written by the users or in order to read and write data in a certain data format, these are elements of the program by means of which the users use certain functions of the program.*
>
> *43. If a third party were to obtain that part of the source or object code that relates to the programming language or file format that are used in a computer program, and if such third party were to create similar components in its own computer program with the help of that code, such conduct possibly involves reproduction in part within the meaning of Art. 4 (a) of Council Directive 91/250.*

This distinction can be applied to the question of the protection of file tree structures, insofar as these structures as such are likewise only functions of the program, but not a manifestation of the program. Accordingly, the Plaintiff's mere reference to tree structures does not suffice in order to substantiate any adapter's copyright on his part or any infringement of such copyright.

In his Statement of Claim, the Plaintiff only states that Radix Trees are "a scalable implementation of a special tree structure, which is used for managing caches in particular, but in many other areas as well." It is not evident from this that they involve a specific program code, which can be implemented itself or is a protectable part of an implementable program code.

On the contrary, the Plaintiff's pleading can only be understood to mean that it is about a "structure", i.e. the manner in which something is arranged. Even if this structure were to relate to the manner (order) in which specific program code (as specific commands) can be arranged most appropriately, most effectively, most rationally in terms of memory space, and so that it can be implemented most rapidly, etc., this would only involve a general design principle for computer programs, but not a program as such. However, a mere design principle as such, which – as the Plaintiff himself states – can be used in the most varied contexts, cannot be protected under copyright law.

b)

The Plaintiff's other pleadings regarding the use of any code in connection with Radix Trees are not sufficient to establish the cause of action either.

Insofar as it is stated on p. 15 of the Statement of Claim that an analysis he conducted revealed that (quote) "*the 'Radix Tree' functionality was used*" in the Defendant's source code, with evidence for this being offered: "*In the event of dispute: submission of the source code files*", it cannot be verified from this statement for one thing that actual lines of code have been used *at all* ("functionality"), and for another which lines of code are supposed to be involved, and finally the offer of evidence is not only aimed at investigation but is also far too vague in itself.

When the Plaintiff then makes statements about “Radix Trees” on p. 13-15 of his written pleading dated 29.04.2016, Exhibit K 31 that is submitted with code from the Defendant’s “vmklinux” indeed suggests that Radix Trees might involve not only a structure, but also specific lines of code. However, the Plaintiff fails to plead which parts listed in K 31 (the essence of the infringement) are meant to have been taken from Linux, and which of those parts he in turn claims to have created himself (the essence of the action). On the contrary, the Plaintiff concedes that he did not invent the structure, stating that he merely helped the inventor Velikov with its “implementation”. This means however, that the part of the code which the Plaintiff claims to have created himself cannot be deduced from the Plaintiff’s pleading.

Nor does K 30 provide any clarification in this respect. Insofar as p. 1 of that exhibit appears to indicate that parts of the code might originate from the Plaintiff, it remains unclear exactly which lines this refers to and to what extent these particular lines are claimed to have also been used in “vmklinux”. All the more so, because the Plaintiff himself states that the Defendant did not use the version dated 2002 that is given in K 30, but a later version dating from 2012, on which to base “vmklinux”; but which parts of the Radix Trees version dated 2012 are meant to have been created by the Plaintiff cannot be deduced from the Plaintiff’s pleading either. To this extent, the expert evidence he offers serves the purpose of investigating facts.

c)

The offer for other kernel developers to be heard on the Plaintiff’s contributions to the Radix Trees likewise serves the purpose of investigating facts and is not a verifiable pleading on those contributions, or on the question of whether the Radix Trees merit protection.

4.

Insofar as the Plaintiff maintains further on p. 15 of the Statement of Claim that his analysis has shown that **many other parts of the Linux operating system have been used by the Defendant**, to which he contributed smaller modifications of the source code, and here also offers “submission of source code files” by way of evidence, this pleading again fails in more than one respect to state the cause of action, because it makes clear neither which “parts” of Linux are supposed to be involved, nor which modifications the Plaintiff made in that respect, and it does not state which parts of code specifically originating from him are supposed to have been used in “vmklinux”.

C.

The decision on costs ensues from Code of Civil Procedure § 91.

The decision on enforceability ensues from Code of Civil Procedure § 709 clauses 1 and 2.