



# BUNDESPATENTGERICHT

IM NAMEN DES VOLKES

URTEIL

2 Ni 13/12 (EP)

---

(Aktenzeichen)

An Verkündungs Statt  
zugestellt  
27. November 2014

...

In der Patentnichtigkeitssache

...

**betreffend das europäische Patent 1 233 343**  
**(DE 602 14 059)**

hat der 2. Senat (Nichtigkeitssenat) des Bundespatentgerichts auf Grund der mündlichen Verhandlung vom 24. Juli 2014 unter Mitwirkung der Vorsitzenden Richterin Sredl sowie der Richter Merzbach, Dipl.-Phys. Dr. rer. nat. Forkel, Dipl.-Phys. Dr. rer. nat. Schwengelbeck und Dipl.-Ing. Altvater

für Recht erkannt:

- I. Das europäische Patent 1 233 343 wird mit Wirkung für das Hoheitsgebiet der Bundesrepublik Deutschland für nichtig erklärt.
- II. Die Kosten des Rechtsstreits trägt die Beklagte.
- III. Das Urteil ist gegen Sicherheitsleistung in Höhe von 120 % des zu vollstreckenden Betrages vorläufig vollstreckbar.

**Tatbestand**

Die Beklagte ist eingetragene Inhaberin des am 15. Februar 2002 unter Inanspruchnahme einer US-Priorität (US 788 317) vom 16. Februar 2001 in englischer Sprache angemeldeten und mit Datum 23. August 2006 veröffentlichten europäischen Patents EP 1 233 343 B1 mit der Bezeichnung „Method and radio interface layer comprising a set of application programming interfaces (APIs)“, welches vom Deutschen Patent- und Markenamt unter der Nummer DE 602 14 059 T2 mit der deutschen Bezeichnung „Verfahren und Funkschnittstellschicht bestehend aus einer Menge von Anwendungsprogrammierungsschnittstellen (APIs)“ geführt wird.

In der Verfahrenssprache Englisch haben die erteilten nebengeordneten Patentansprüche 1, 7 und 31 folgenden Wortlaut:

- „1. A method for communicating with a telephony radio hardware (245, 485), **characterized in that** the method comprises the steps of:

interfacing a computer (20, 49) to the telephony radio hardware by an abstraction layer comprising a proxy layer (235) and a driver layer (240);

calling one of a set of APIs comprised in said abstraction layer, wherein the set of APIs correspond to call control functions and are for abstracting out multiple radio hardware technologies without knowledge of the telephony radio or cellular network;

receiving said API call at a first interface of the proxy layer;

transforming the API call to a command understood by the driver layer and sending said command to the driver layer at a second interface;

receiving said command at the second interface;

determining at least one standard telephony radio hardware command corresponding to the called API by said driver layer; and

sending, by said driver layer, the telephony radio hardware command to the telephony radio hardware at a third interface.

7. A method of operating a telephone (200, 400) for facilitating communications between an application program module and a radio hardware (245, 485), **characterized in that** the method comprises the steps of:

calling (305), by an application (205, 210, 215, 220, 225, 405), an API to perform a particular function;

receiving said API call at a first interface of a proxy layer (235);

transforming the API to an IOCTL by the proxy layer (235) comprised

in a radio interface layer (430), wherein the proxy layer is for communicating with the application at the first interface and a driver layer at a second interface and the driver layer communicates with the proxy layer at said second interface and the radio hardware at a third interface;

sending (320) the IOCTL by the proxy layer to the driver layer (240) at said second interface;

receiving the IOCTL by the driver layer at said second interface;

transforming, by the driver layer, the IOCTL into a command understood by the radio hardware to perform the particular function;

and

sending said radio command to the radio hardware at said third interface.

31. A telephone apparatus (200, 400), adapted to perform the method of one of claims 1 to 30.“

In deutscher Übersetzung lauten die erteilten Ansprüche 1, 7 und 31:

- „1. Verfahren zur Kommunikation mit einer Telefoniefunkhardware (245; 485), **dadurch gekennzeichnet, dass** das Verfahren folgende Schritte umfasst:

Koppeln (interfacing) eines Computers(20; 49) an die Telefoniefunkhardware durch eine Abstraktionsschicht, welche eine Proxyschicht (235) und eine Treiberschicht (240) umfasst;

Aufrufen eines aus einem Satz in der Abstraktionsschicht enthaltener APIs, wobei der Satz von APIs Anrufsteuerfunktionen entsprechen und der Abstraktion aus mehreren Funkhardwaretechnologien ohne Kenntnis des Telefoniefunks oder des Zellularnetzes dienen;

Empfangen des API-Aufrufs an einer ersten Schnittstelle der Proxyschicht;

Transformieren des API-Aufrufs in einen Befehl, der von der Treiberschicht verstanden wird, und Senden des Befehls an die Treiberschicht an einer zweiten Schnittstelle;  
Empfangen des Befehls an der zweiten Schnittstelle;  
Bestimmen mindestens eines Standardtelefoniefunkhardwarebefehls, der dem aufgerufenen API entspricht, durch die Treiberschicht; und  
Senden des Telefoniefunkhardwarebefehls an die Telefoniefunkhardware an einer dritten Schnittstelle durch die Treiberschicht.

7. Verfahren zum Betrieb eines Telefons( 200; 400) zur Ermöglichung von Kommunikationen zwischen einem Anwendungsprogrammmodul und einer Funkhardware (245; 485), **dadurch gekennzeichnet, dass** das Verfahren folgende Schritte umfasst:

Aufrufen (305) eines API durch eine Anwendung (205, 210, 215, 220, 225, 405), um eine bestimmte Funktion durchzuführen;  
Empfangen des API-Aufrufs an einer ersten Schnittstelle einer Proxyschicht (235);  
Transformieren des APIs in einen IOCTL durch die Proxyschicht (235), welche in einer Funkschnittstellenschicht (430) enthalten ist, wobei die Proxyschicht der Kommunikation mit der Anwendung an der ersten Schnittstelle und einer Treiberschicht an einer zweiten Schnittstelle dient und die Treiberschicht mit der Proxyschicht an der zweiten Schnittstelle und mit der Funkhardware an einer dritten Schnittstelle kommuniziert;  
Senden (320) des IOCTL durch die Proxyschicht an die Treiberschicht (240) an der zweiten Schnittstelle;  
Empfangen des IOCTL durch die Treiberschicht an der zweiten Schnittstelle;  
Transformieren des IOCTL in einen Befehl, der durch die Funkhardware verstanden wird, um eine bestimmte Funktion

auszuführen, durch die Treiberschicht; und

Senden des Funkbefehls an die Funkhardware an der dritten Schnittstelle.

31. Telefongerät (200; 400), das dazu angepasst ist, das Verfahren nach einem der Ansprüche 1 bis 30 durchzuführen.“

Hinsichtlich des Wortlauts der auf den erteilten Patentanspruch 1 direkt oder indirekt rückbezogenen Unteransprüche 2 bis 6 sowie der auf den erteilten Patentanspruch 7 direkt oder indirekt rückbezogenen Unteransprüche 8 bis 30 wird auf die Streitpatentschrift Bezug genommen.

Die Klägerin macht unter Berufung auf die von ihr vorgelegten Druckschriften und Unterlagen

**D1:** US 5 490 275 A,

**D2:** Bill Robinson: *Software Radio: The Standards Perspective*; laut Klägerin veröffentlicht im Mai 1997,

**D3:** Enrico Buracchini: *The Software Radio Concept*. In: IEEE Communications Magazine, September 2000,

**D4:** Christian Herrero Verón, Marta Sacchini, Irene Yera Pemán: *Software interface for a multi technology system phone*; Master Thesis, laut Klägerin als „Doktorarbeit“ veröffentlicht zwischen 24. März 2000 und 6. April 2000,

**D5:** US 5 327 558 A,

**D6:** Auszüge aus: Chris Sells: *Windows Telephony Programming - A Developer's Guide to TAPI*, Addison-Wesley: Reading, Massachusetts, 1998

sowie die mit Schriftsatz vom 20. September 2013 vorgelegten Anlagen

**NK6:** Internet-Archiv-Ausdruck der CORDIS-Webseite

zum Beleg, dass die Druckschrift D2 (Bill Robinson: *Software Radio: The Standards Perspective*; laut Klägerin veröffentlicht im Mai 1997) vorveröffentlicht sei,

**NK7:** Stellungnahme von Prof. Per Runeson vom 18. Mai 2011

als Beleg, dass die Druckschrift D4 (Christian Herrero Verón, Marta Sacchini, Irene Yera Pernán: *Software interface for a multi technology system phone*; Master Thesis, laut Klägerin als „Doktorarbeit“ veröffentlicht zwischen 24. März 2000 und 6. April 2000.) tatsächlich vorveröffentlicht sei; und

**NK8:** vollständige Abschrift der Druckschrift **D6**, Chris Sells: *Windows Telephony Programming – A Developer’s Guide to TAPI*, Addison-Wesley: Reading, Massachusetts, 1998 (zuvor hatte die Klägerin nur Auszüge der **D6** eingereicht)

geltend, dass der Gegenstand des Streitpatents gegenüber dem Stand der Technik nicht patentfähig sei. Er sei gegenüber den aufgeführten Druckschriften nicht neu, beruhe aber jedenfalls nicht auf einer erfinderischen Tätigkeit (Art. 52 Abs. 1 EPÜ i. V. m. Art. II, § 6 Abs. 1 Nr. 1 IntPatÜG und Art. 54 Abs. 1, 2 EPÜ und Art. 56 i. V. m. Art. 54 Abs. 2 EPÜ). Zudem sei er auch nach Art. 52 (2) Buchst. c / (3) EPÜ vom Patentschutz ausgeschlossen, da die beanspruchte Lehre keine Anweisungen enthalte, die der Lösung eines konkreten technischen Problems mit technischen Mitteln diene. Ferner sei das Streitpatent unzulässig erweitert, da das Merkmal „Transformieren des API-Aufrufs in einen Befehl“ im Anspruch 1 über den Inhalt der Anmeldung in der ursprünglichen Fassung hinausgehe (Art. 138 Abs. 1 lit. c) EPÜ i. V. m. Art. II § 6 Abs. 1 Nr. 3 IntPatÜbkG).

In der Streitpatentschrift werden ferner folgende Druckschriften abgehandelt:

**D7:** EP 0 994 614 A2 und

**D8:** WO 96/06393 A1.

Im Prüfungsverfahren vor dem EPA wurden neben den Druckschriften D7 und D8 noch folgende Druckschriften berücksichtigt:

**D9:** US 6 018 571 A,

**D10:** US 6 141 564 A,

**D11:** M.M. Tso [u. a.]: *Always On, Always Connected Mobile Computing*. In: 5th IEEE International Conference on, 1996, S. 918-924, und

**D12:** H. Steeman: *Wireless Application Protocol (WAP)*. In: *Elektro Electronics*, Vol. 26, No. 289, 2000, S. 56-58.

Die Klägerin beantragt,

das europäische Patent EP 1 233 343 mit Wirkung für das Hoheitsgebiet der Bundesrepublik Deutschland für nichtig zu erklären.

Die Beklagte beantragt,

die Klage abzuweisen;

hilfsweise beantragt sie, dem Streitpatent eine der Fassungen der in der Verfahrenssprache Englisch abgefassten Hilfsanträge 1-9, vorgelegt mit Schriftsatz vom 29. Oktober 2013, Bl. 342-445 d. A., zu geben, wobei in den Hilfsanträgen 4 und 9 jeweils in Patentanspruch 1 die Formulierung des Verfahrensschritts „transforming the by the proxy layer instance“ geändert wird in die Formulierung „transforming, by the proxy layer instance“ und wobei in den Hilfsanträgen 1-9 die unabhängigen



Patentansprüche jeweils gesondert auf ihre Patentfähigkeit geprüft werden sollen.

### **Hilfsantrag 1**

Die unabhängigen Patentansprüche 1 und 7 in der Fassung des **Hilfsantrags 1** lauten (mit markierten Änderungen gegenüber den Patentansprüchen 1 und 7 in der erteilten Fassung):

- „1. A method for communicating with a telephony radio hardware (245, 485), characterized in that the method comprises the steps of:  
  
interfacing a computer (20, 49) to the telephony radio hardware by an abstraction layer comprising a proxy layer (235) and a driver layer (240), wherein the proxy layer is hardware independent and the driver layer is hardware specific;  
calling one of a set of APIs comprised in said abstraction layer, wherein the set of APIs correspond to call control functions and are for abstracting out multiple radio hardware technologies without knowledge of the telephony radio or cellular network;  
receiving said API call at a first interface of the proxy layer;  
transforming the API call to a command understood by the driver layer and sending said command to the driver layer at a second interface;  
receiving said command at the second interface;  
determining at least one standard telephony radio hardware command corresponding to the called API by said driver layer; and  
sending, by said driver layer, the telephony radio hardware command to the telephony radio hardware at a third interface.
  
7. A method of operating a telephone (200, 400) for facilitating communications between an application program module and a radio

hardware (245, 485), characterized in that the method comprises the steps of:

calling (305), by an application (205, 210, 215, 220, 225, 405), an API to perform a particular function;

receiving said API call at a first interface of a proxy layer (235);

transforming the API to an IOCTL by the proxy layer (235) comprised in a radio interface layer (430), wherein the proxy layer is for communicating with the application at the first interface and a driver layer at a second interface and the driver layer communicates with the proxy layer at said second interface and the radio hardware at a third interface; wherein the proxy layer is hardware independent and the driver layer is hardware specific

sending (320) the IOCTL by the proxy layer to the driver layer (240) at said second interface;

receiving the IOCTL by the driver layer at said second interface;

transforming, by the driver layer, the IOCTL into a command understood by the radio hardware to perform the particular function;

and

sending said radio command to the radio hardware at said third interface.”

Die weiteren Patentansprüche gemäß Hilfsantrag 1 entsprechen den erteilten Unteransprüchen 2 bis 6 und 8 bis 30 sowie dem nebengeordneten Patentanspruch 31 in der erteilten Fassung.

## **Hilfsantrag 2**

Die unabhängigen Patentansprüche 1 und 7 in der Fassung des **Hilfsantrags 2** lauten (mit markierten Änderungen gegenüber den Patentansprüchen 1 und 7 in der erteilten Fassung):

- „1. A method for communicating with a telephony radio hardware (245, 485), characterized in that the method comprises the steps of:

interfacing a computer (20, 49) to the telephony radio hardware by an abstraction layer comprising a proxy layer (235) and a driver layer (240), wherein the proxy layer is hardware independent and the driver layer is hardware specific;

calling one of a set of APIs comprised in said abstraction layer, wherein the set of APIs correspond to call control functions and are for abstracting out multiple radio hardware technologies without knowledge of the telephony radio or cellular network;

receiving said API call at a first interface of the proxy layer;

transforming the API call to a command understood by the driver layer and sending said command to the driver layer at a second interface;

receiving said command at the second interface;

determining at least one standard telephony radio hardware command corresponding to the called API by said driver layer; ~~and~~

sending, by said driver layer, the telephony radio hardware command to the telephony radio hardware at a third interface;

generating an API call at one of a plurality of applications (205, 210, 215, 220, 225, 405) to perform the particular function;

sending the API call to a radio driver;

at the radio driver, converting the API call to a command understood by the radio hardware;

transmitting the command to the radio hardware;

performing the particular function at the radio hardware; and

in response to successfully performing the particular function, sending a success code to the one of the plurality of applications that generated the API call;

wherein

the API, command and success code are associated with an identifier linking them together and linking them to the one of the plurality of applications that generated the API call;  
the radio driver receives the success code;  
using the identifier, the radio driver matches the success code with the one of the plurality of applications that generated the API call;  
and  
the radio driver sends the success code to the one of the plurality of applications that generated the API call.

7. A method of operating a telephone (200, 400) for facilitating communications between an application program module and a radio hardware (245, 485), characterized in that the method comprises the steps of:

calling (305), by an application (205, 210, 215, 220, 225, 405), an API to perform a particular function;  
receiving said API call at a first interface of a proxy layer (235);  
transforming the API to an IOCTL by the proxy layer (235) comprised in a radio interface layer (430), wherein the proxy layer is for communicating with the application at the first interface and a driver layer at a second interface and the driver layer communicates with the proxy layer at said second interface and the radio hardware at a third interface; wherein the proxy layer is hardware independent and the driver layer is hardware specific  
sending (320) the IOCTL by the proxy layer to the driver layer (240) at said second interface;  
receiving the IOCTL by the driver layer at said second interface;  
transforming, by the driver layer, the IOCTL into a command understood by the radio hardware to perform the particular function;  
sending said radio command to the radio hardware at said third interface;

receiving, by the driver layer, communications from the radio hardware indicating that the particular function has been performed and sending, by the driver layer, a success code to the proxy layer indicating that the particular function has been performed;  
wherein the telephone comprises the proxy layer, the driver layer, the application and the radio hardware;  
the application calls the API in the proxy layer;  
the API is associated with the function to be performed by the radio hardware;  
the command understood by the radio hardware corresponds to the function; and  
the method further comprises the steps of:  
sending (325) the command to the radio hardware;  
generating in the driver layer a unique ID associated with the API,  
waiting (335) for a response from the radio hardware; and  
when received, calling back (335) the calling application with the response and the unique ID returned from the API call.”

Die weiteren Patentansprüche gemäß Hilfsantrag 2 entsprechen den erteilten Unteransprüchen 2 bis 6, den neu nummerierten und mit entsprechend geänderten Rückbezügen versehenen erteilten Unteransprüchen 10, 11 (nunmehr 8 und 9), 14 bis 30 (nunmehr 10 bis 26) sowie dem nebengeordneten Patentanspruch 31 in der erteilten Fassung (nunmehr 27).

### **Hilfsantrag 3**

Die unabhängigen Patentansprüche 1 und 7 in der Fassung des **Hilfsantrags 3** lauten (mit markierten Änderungen gegenüber den Patentansprüchen 1 und 7 in der erteilten Fassung):

- „1. A method for communicating with a telephony radio hardware (245, 485), characterized in that the method comprises the steps of:

interfacing a computer (20, 49) to the telephony radio hardware by an abstraction layer comprising a proxy layer (235) and a driver layer (240), wherein the proxy layer is hardware independent and the driver layer is hardware specific;

calling, by each of plural applications (205, 210, 215, 220, 225, 405), one of a set of APIs comprised in said abstraction layer, wherein the set of APIs correspond to call control functions and are for abstracting out multiple radio hardware technologies without knowledge of the telephony radio or cellular network;

creating a separate proxy layer instance (235) in said proxy layer for each of the calling applications by the calling applications linking to a proxy library, wherein said driver layer (240) is shared among the proxy layer instances (235), and wherein the proxy layer instances and the driver layer use different address spaces;

receiving, from one of said calling applications, said API call at a first interface of the proxy layer instance (235) associated to said one calling application;

transforming, by the proxy layer instance (235) associated to said one calling application, the API call to a command understood by the shared driver layer; and

sending, by the proxy layer instance (235) associated to said one calling application, said command to the shared driver layer at a second interface;

receiving, by said shared driver layer and from the proxy layer instance (235) associated to said one calling application, said command at the second interface;

determining at least one standard telephony radio hardware command corresponding to the called API by said shared driver layer; and

sending, by said shared driver layer, the telephony radio hardware command to the telephony radio hardware at a third interface.

7. A method of operating a telephone (200, 400) for facilitating communications between an application program module and a radio hardware (245, 485), characterized in that the method comprises the steps of:

calling (305), by each of a plurality of applications (205, 210, 215, 220, 225, 405), an API to perform a particular function;

creating, in a proxy layer comprised in a radio interface layer (430), a separate proxy layer instance (235) for each of the calling applications by the calling applications linking to a proxy library, wherein the proxy layer is for communicating with the applications at a first interface and a driver layer at a second interface and the driver layer communicates with the proxy layer at said second interface and the radio hardware at a third interface, and wherein the driver layer (240) is shared by the proxy layer instances (235), and the proxy layer instances (235) and the driver layer use different address spaces;

receiving, from one of said calling applications, said API call at a first interface of the proxy layer instance (235) associated to said one calling application;

transforming the API to an IOCTL by said proxy layer instance (235) associated to said one calling application; ~~comprised in a radio interface layer (430), wherein the proxy layer is for communicating with the application at the first interface and a driver layer at a second interface and the driver layer communicates with the proxy layer at said second interface and the radio hardware at a third interface;~~

sending (320) the IOCTL by the said proxy layer instance associated to said one calling application to the shared driver layer (240) at said second interface;

receiving the IOCTL by the shared driver layer at said second interface;

transforming, by the shared driver layer, the IOCTL into a command understood by the radio hardware to perform the particular function;  
and  
sending said radio command to the radio hardware at said third interface;  
wherein the proxy layer is hardware independent and the driver layer is hardware specific.”

Die weiteren Patentansprüche gemäß Hilfsantrag 3 entsprechen den erteilten Unteransprüchen 2 bis 6 und 8 bis 30 sowie dem nebengeordneten Patentanspruch 31 in der erteilten Fassung.

#### **Hilfsantrag 4**

Die unabhängigen Patentansprüche 1 und 7 in der Fassung des Hilfsantrags 4 lauten (mit markierten Änderungen gegenüber den Patentansprüchen 1 und 7 in der erteilten Fassung):

- „1. A method for communicating with a telephony radio hardware (245, 485), characterized in that the method comprises the steps of:

interfacing a computer (20, 49) to the telephony radio hardware by an abstraction layer comprising a proxy layer (235) and a driver layer (240), wherein the proxy layer is hardware independent and the driver layer is hardware specific;

calling, by each of plural applications (205, 210, 215, 220, 225, 405), one of a set of APIs comprised in said abstraction layer, wherein the set of APIs correspond to call control functions and are for abstracting out multiple radio hardware technologies without knowledge of the telephony radio or cellular network;

creating a separate proxy layer instance (235) in said proxy layer for each of the calling applications by the calling applications linking to a



proxy library, wherein said driver layer (240) is shared among the proxy layer instances (235), and wherein the proxy layer instances and the driver layer use different address spaces;

receiving, from one of said calling applications, said API call to perform a particular call control function at a first interface of the proxy layer instance (235) associated to said one calling application;

transforming, by the proxy layer instance (235) associated to said one calling application, the API call to a command corresponding to the API call to perform the particular call control function understood by the shared driver layer; ~~and~~

sending, by the proxy layer instance (235) associated to said one calling application, said command corresponding to the API call to perform the particular call control function to the shared driver layer at a second interface;

receiving, by said shared driver layer and from the proxy layer instance (235) associated to said one calling application, said command corresponding to the API call to perform the particular call control function at the second interface;

determining at least one standard telephony radio hardware command corresponding to the ~~called~~ API call to perform the particular call control function by said shared driver layer;

sending, by said shared driver layer, the telephony radio hardware command corresponding to the API call to perform the the particular call control function to the telephony radio hardware at a third interface;

performing the particular function at the radio hardware; and

in response to successfully performing the particular function at the radio hardware, sending a success code to the one of the plurality of applications that generated the API call;

wherein the API call, command and success code are associated with an identifier linking them together and linking them to the one of the plurality of applications that generated the API call;

the driver layer receives the success code;

using the identifier, the driver layer matches the success code with the one of the plurality of applications that generated the API call;  
and

the driver layer sends the success code to the one of the plurality of applications that generated the API call.

7. A method of operating a telephone (200, 400) for facilitating communications between an application program module and a radio hardware (245, 485), characterized in that the method comprises the steps of:

calling (305), by each of a plurality of applications (205, 210, 215, 220, 225, 405), an API in a proxy layer to perform a particular function to be performed by the radio hardware;

creating, in a proxy layer comprised in a radio interface layer (430), a separate proxy layer instance (235) for each of the calling applications by the calling applications linking to a proxy library, wherein the proxy layer is for communicating with the applications at a first interface and a driver layer at a second interface and the driver layer communicates with the proxy layer at said second interface and the radio hardware at a third interface, and wherein the driver layer (240) is shared by the proxy layer instances (235), and the proxy layer instances (235) and the driver layer use different address spaces;

receiving, from one of said calling applications, said API call at a first interface of the proxy layer instance (235) associated to said one calling application;

transforming the API to an IOCTL by said proxy layer instance (235) associated to said one calling application; ~~comprised in a radio interface layer (430), wherein the proxy layer is for communicating with the application at the first interface and a driver layer at a second in-~~

~~terface and the driver layer communicates with the proxy layer at said second interface and the radio hardware at a third interface;~~  
sending (320) the IOCTL by the said proxy layer instance associated to said one calling application to the shared driver layer (240) at said second interface;  
receiving the IOCTL by the shared driver layer at said second interface;  
transforming, by the shared driver layer, the IOCTL into a command understood by the radio hardware to perform the particular function, wherein the command understood by the radio hardware corresponds to the particular function; ~~and~~  
sending said radio command to the radio hardware at said third interface;  
receiving, by the driver layer, communications from the radio hardware indicating that the particular function has been performed;  
sending, by the driver layer, a success code to the proxy layer instance from which the IOCTL has been received, the success code indicating that the particular function has been performed;  
generating in the driver layer a unique ID associated with the API;  
waiting (335) for a response from the radio hardware; and  
when received, calling back (335) said one calling application with the response and the unique ID returned from the API call;  
wherein the proxy layer instances are hardware independent and the driver layer is hardware specific.”

Die weiteren Patentansprüche gemäß Hilfsantrag 4 entsprechen den erteilten Unteransprüchen 2 bis 6, den neu nummerierten und mit entsprechend geänderten Rückbezügen versehenen erteilten Unteransprüchen 10, 11 (nunmehr 8, 9), 14 bis 30 (nunmehr 10 bis 26) sowie dem nebengeordneten Patentanspruch 31 in der erteilten Fassung (nunmehr 27).

### **Hilfsantrag 5**

Die unabhängigen Patentansprüche 1 und 7 in der Fassung des Hilfsantrags 5 lauten (mit markierten Änderungen gegenüber den Patentansprüchen 1 und 7 in der erteilten Fassung):

- „1. A method for communicating with a telephony radio hardware (245, 485), characterized in that the method comprises the steps of:

interfacing a computer (20, 49) to the telephony radio hardware by an abstraction layer comprising a proxy layer (235) and a driver (240);  
calling one of a set of APIs comprised in said abstraction layer, wherein the set of APIs correspond to call control functions and are for abstracting out multiple radio hardware technologies without knowledge of the telephony radio or cellular network;  
receiving said API call from a TSP or TAPI interface at a first interface of the proxy layer;  
transforming the API call to a command understood by the driver layer and sending said command to the driver layer at a second interface;  
receiving said command at the second interface;  
determining at least one standard telephony radio hardware command corresponding to the called API by said driver layer; and  
sending, by said driver layer, the telephony radio hardware command to the telephony radio hardware at a third interface.

7. A method of operating a telephone (200, 400) for facilitating communications between an application program module and a radio hardware (245, 485), characterized in that the method comprises the steps of:

calling (305), by an application (205, 210, 215, 220, 225, 405), an API to perform a particular function, wherein the application is a TSP or a TAPI interface;

receiving said API call at a first interface of a proxy layer (235);

transforming the API to an IOCTL by the proxy layer (235) comprised in a radio interface layer (430), wherein the proxy layer is for communicating with the application at the first interface and a driver layer at a second interface and the driver layer communicates with the proxy layer at said second interface and the radio hardware at a third interface;

sending (320) the IOCTL by the proxy layer to the driver layer (240) at said second interface;

receiving the IOCTL by the driver layer at said second interface;

transforming, by the driver layer, the IOCTL into a command understood by the radio hardware to perform the particular function;

and

sending said radio command to the radio hardware at said third interface.”

Die weiteren Patentansprüche gemäß Hilfsantrag 5 entsprechen den erteilten Unteransprüchen 2 bis 6 und 8 bis 30 sowie dem nebengeordneten Patentanspruch 31 in der erteilten Fassung.

### **Hilfsantrag 6**

Anspruch 1 nach Hilfsantrag 6 entspricht dem Anspruch 1 nach Hilfsantrag 1 unter Änderung des Merkmals (Änderung markiert):

„receiving said API call from a TSP or TAPI interface at a first interface of the proxy layer;“

Der nebengeordnete Anspruch 7 nach Hilfsantrag 6 entspricht dem Anspruch 7 nach Hilfsantrag 1 unter folgender Änderung des Merkmals (Änderung markiert):

„calling (305), by an application (205, 210, 215, 220, 225, 405), an API to perform a particular function, wherein the application is a TSP or TAPI interface;”

Die unabhängigen Patentansprüche 1 und 7 in der Fassung des **Hilfsantrags 6** lauten damit (Änderungen gegenüber den Patentansprüchen 1 und 7 nach Hilfsantrag 1 markiert):

- „1. A method for communicating with a telephony radio hardware (245, 485), characterized in that the method comprises the steps of:

interfacing a computer (20, 49) to the telephony radio hardware by an abstraction layer comprising a proxy layer (235) and a driver layer (240); wherein the proxy layer is hardware independent and the driver layer ist hardware specific;

calling one of a set of APIs comprised in said abstraction layer, wherein the set of APIs correspond to call control functions and are for abstracting out multiple radio hardware technologies without knowledge of the telephony radio or cellular network;

receiving said API call from a TSP or TAPI interface at a first interface of the proxy layer;

transforming the API call to a command understood by the driver layer and sending said command to the driver layer at a second interface;

receiving said command at the second interface;

determining at least one standard telephony radio hardware command corresponding to the called API by said driver layer; and

sending, by said driver layer, the telephony radio hardware command to the telephony radio hardware at a third interface.

7. A method of operating a telephone (200, 400) for facilitating communications between an application program module and a radio hardware (245, 485), characterized in that the method comprises the steps of:

calling (305), by an application (205, 210, 215, 220, 225, 405), an API to perform a particular function, wherein the application is a TSP or TAPI interface;

receiving said API call at a first interface of a proxy layer (235);

transforming the API to an IOCTL by the proxy layer (235) comprised in a radio interface layer (430), wherein the proxy layer is for communicating with the application at the first interface and a driver layer at a second interface and the driver layer communicates with the proxy layer at said second interface and the radio hardware at a third interface; wherein the proxy layer is hardware independent and the driver layer is hardware specific;

sending (320) the IOCTL by the proxy layer to the driver layer (240) at said second interface;

receiving the IOCTL by the driver layer at said second interface;

transforming, by the driver layer, the IOCTL into a command understood by the radio hardware to perform the particular function;

and

sending said radio command to the radio hardware at said third interface.“

Die weiteren Patentansprüche gemäß Hilfsantrag 6 entsprechen den erteilten Unteransprüchen 2 bis 6 und 8 bis 30 sowie dem nebengeordneten Patentanspruch 31 in der erteilten Fassung.

### Hilfsantrag 7

Anspruch 1 nach Hilfsantrag 7 entspricht dem Anspruch 1 nach Hilfsantrag 2 unter Änderung des Merkmals (Änderung markiert):

„generating an API call at one of a plurality of applications (205, 210, 215, 220, 225, 405) to perform the particular function, wherein said one calling application is a TSP or a TAPI interface;”

Der nebengeordnete Anspruch 7 nach Hilfsantrag 7 entspricht dem Anspruch 7 nach Hilfsantrag 2 unter folgender Änderung des Merkmals (Änderung gekennzeichnet):

„calling (305), by an application (205, 210, 215, 220, 225, 405), an API to perform a particular function; wherein said calling application is a TSP or a TAPI interface;”

Die unabhängigen Patentansprüche 1 und 7 in der Fassung des Hilfsantrags 7 lauten damit (Änderungen gegenüber den Patentansprüchen 1 und 7 gemäß Hilfsantrag 2 markiert):

- „1. A method for communicating with a telephony radio hardware (245, 485), characterized in that the method comprises the steps of:

interfacing a computer (20, 49) to the telephony radio hardware by an abstraction layer comprising a proxy layer (235) and a driver layer (240); wherein the proxy layer is hardware independent and the driver layer ist hardware specific;

calling one of a set of APIs comprised in said abstraction layer, wherein the set of APIs correspond to call control functions and are for abstracting out multiple radio hardware technologies without knowledge of the telephony radio or cellular network;



receiving said API call at a first interface of the proxy layer;  
transforming the API call to a command understood by the driver layer and sending said command to the driver layer at a second interface;  
receiving said command at the second interface;  
determining at least one standard telephony radio hardware command corresponding to the called API by said driver layer;  
sending, by said driver layer, the telephony radio hardware command to the telephony radio hardware at a third interface;  
generating an API call at one of a plurality of applications (205, 210, 215, 220, 225, 405) to perform the particular function, wherein said one calling application is a TSP or a TAPI interface;  
sending the API call to a radio driver;  
at the radio driver, converting the API call to a command understood by the radio hardware;  
transmitting the command to the radio hardware;  
performing the particular function at the radio hardware; and  
in response to successfully performing the particular function, sending a success code to the one of the plurality of applications that generated the API call;  
wherein  
the API, command and success code are associated with an identifier linking them together and linking them to the one of the plurality of applications that generated the API call;  
the radio driver receives the success code;  
using the identifier, the radio driver matches the success code with the one of the plurality of applications that generated the API call;  
and  
the radio driver sends the success code to the one of the plurality of applications that generated the API call.

7. A method of operating a telephone (200, 400) for facilitating communications between an application program module and a radio hardware (245, 485), characterized in that the method comprises the steps of:

calling (305), by an application (205, 210, 215, 220, 225, 405), an API to perform a particular function, wherein said calling application is a TSP or a TAPI interface;

receiving said API call at a first interface of a proxy layer (235);

transforming the API to an IOCTL by the proxy layer (235) comprised in a radio interface layer (430), wherein the proxy layer is for communicating with the application at the first interface and a driver layer at a second interface and the driver layer communicates with the proxy layer at said second interface and the radio hardware at a third interface; wherein the proxy layer is hardware independent and the driver layer is hardware specific;

sending (320) the IOCTL by the proxy layer to the driver layer (240) at said second interface;

receiving the IOCTL by the driver layer at said second interface;

transforming, by the driver layer, the IOCTL into a command understood by the radio hardware to perform the particular function; and

sending said radio command to the radio hardware at said third interface;

receiving, by the driver layer, communications from the radio hardware indicating that the particular function has been performed and

sending, by the driver layer, a success code to the proxy layer indicating that the particular function has been performed;

wherein the telephone comprises the proxy layer, the driver layer, the application and the radio hardware;

the application calls the API in the proxy layer;

the API is associated with the function to be performed by the radio hardware;  
the command understood by the radio hardware corresponds to the function; and  
the method further comprises the steps of:  
sending (325) the command to the radio hardware;  
generating in the driver layer a unique ID associated with the API.  
waiting (335) for a response from the radio hardware; and  
when received, calling back (335) the calling application with the response and the unique ID returned from the API call.”

Die weiteren Patentansprüche gemäß Hilfsantrag 7 entsprechen den erteilten Unteransprüchen 2 bis 6, den neu nummerierten und mit entsprechend geänderten Rückbezügen versehenen erteilten Unteransprüchen 10, 11 (nunmehr 8, 9), 14 bis 22 (nunmehr 10 bis 18), 29, 30 (nunmehr 19, 20) sowie dem nebengeordneten Patentanspruch 31 in der erteilten Fassung (nunmehr 21).

### **Hilfsantrag 8**

Anspruch 1 nach Hilfsantrag 8 entspricht dem Anspruch 1 nach Hilfsantrag 3 unter Änderung des Merkmals (Änderung markiert):

„receiving, from one of said calling applications, said API call at a first interface of the proxy layer instance (235) associated to said one calling application, wherein said one calling application is a TSP or TAPI interface;”

Der nebengeordnete Anspruch 7 nach Hilfsantrag 8 entspricht dem Anspruch 7 nach Hilfsantrag 3 unter Änderung des Merkmals (Änderung markiert):

„receiving, from one of said calling applications, said API call at a first interface of the proxy layer instance (235) associated to said one calling application, wherein said one calling application is a TSP or TAPI interface;”

Die unabhängigen Patentansprüche 1 und 7 in der Fassung des Hilfsantrags 8 lauten (mit markierten Änderungen gegenüber den Patentansprüchen 1 und 7 gemäß Hilfsantrag 3):

- „1. A method for communicating with a telephony radio hardware (245, 485), characterized in that the method comprises the steps of:

interfacing a computer (20, 49) to the telephony radio hardware by an abstraction layer comprising a proxy layer (235) and a driver layer (240), wherein the proxy layer is hardware independent and the driver layer is hardware specific;

calling, by each of plural applications (205, 210, 215, 220, 225, 405), one of a set of APIs comprised in said abstraction layer, wherein the set of APIs correspond to call control functions and are for abstracting out multiple radio hardware technologies without knowledge of the telephony radio or cellular network;

creating a separate proxy layer instance (235) in said proxy layer for each of the calling applications by the calling applications linking to a proxy library, wherein said driver layer (240) is shared among the proxy layer instances (235), and wherein the proxy layer instances and the driver layer use different address spaces;

receiving, from one of said calling applications, said API call at a first interface of the proxy layer instance (235) associated to said one calling application, wherein said one calling application is a TSP or TAPI interface;

transforming, by the proxy layer instance (235) associated to said one calling application, the API call to a command understood by the shared driver layer;

sending, by the proxy layer instance (235) associated to said one calling application, said command to the shared driver layer at a second interface;

receiving, by said shared driver layer and from the proxy layer instance (235) associated to said one calling application, said command at the second interface;  
determining at least one standard telephony radio hardware command corresponding to the called API by said shared driver layer; and  
sending, by said shared driver layer, the telephony radio hardware command to the telephony radio hardware at a third interface.

7. A method of operating a telephone (200, 400) for facilitating communications between an application program module and a radio hardware (245, 485), characterized in that the method comprises the steps of:

calling (305), by each of a plurality of applications (205, 210, 215, 220, 225, 405), an API to perform a particular function;  
creating, in a proxy layer comprised in a radio interface layer (430), a separate proxy layer instance (235) for each of the calling applications by the calling applications linking to a proxy library, wherein the proxy layer is for communicating with the applications at a first interface and a driver layer at a second interface and the driver layer communicates with the proxy layer at said second interface and the radio hardware at a third interface, and wherein the driver layer (240) is shared by the proxy layer instances (235), and the proxy layer instances (235) and the driver layer use different address spaces;  
receiving, from one of said calling applications, said API call at a first interface of the proxy layer instance (235) associated to said one calling application, wherein said one calling application is a TSP or TAPI interface;  
transforming the API to an IOCTL by said proxy layer instance (235) associated to said one calling application;

sending (320) the IOCTL by said proxy layer instance associated to said one calling application to the shared driver layer (240) at said second interface;  
receiving the IOCTL by the shared driver layer at said second interface;  
transforming, by the shared driver layer, the IOCTL into a command understood by the radio hardware to perform the particular function;  
and  
sending said radio command to the radio hardware at said third interface;  
wherein the proxy layer is hardware independent and the driver layer is hardware specific.”

Die weiteren Patentansprüche gemäß Hilfsantrag 8 entsprechen den erteilten Unteransprüchen 2 bis 6 und 8 bis 30 sowie dem nebengeordneten Patentanspruch 31 in der erteilten Fassung.

### **Hilfsantrag 9**

Anspruch 1 nach Hilfsantrag 9 entspricht dem Anspruch 1 nach Hilfsantrag 4 unter Änderung des Merkmals (Änderung markiert):

„receiving, from one of said calling applications, said API call to perform a particular call control function at a first interface of the proxy layer instance (235) associated to said one calling application, wherein said one calling application is a TSP or a TAPI interface;”

Der nebengeordnete Anspruch 7 nach Hilfsantrag 9 entspricht dem Anspruch 7 nach Hilfsantrag 4 unter Änderung des Merkmals (Änderung gekennzeichnet):

„receiving, from one of said calling applications, said API call at a first interface of the proxy layer instance (235) associated to said one calling application, wherein said one calling application is a TSP or a TAPI interface;”

Die unabhängigen Patentansprüche 1 und 7 in der Fassung des Hilfsantrags 9 lauten (mit markierten Änderungen gegenüber den Patentansprüchen 1 und 7 gemäß Hilfsantrag 4):

„1. A method for communicating with a telephony radio hardware (245, 485), characterized in that the method comprises the steps of:

interfacing a computer (20, 49) to the telephony radio hardware by an abstraction layer comprising a proxy layer (235) and a driver layer (240), wherein the proxy layer is hardware independent and the driver layer is hardware specific;

calling, by each of plural applications (205, 210, 215, 220, 225, 405), one of a set of APIs comprised in said abstraction layer, wherein the set of APIs correspond to call control functions and are for abstracting out multiple radio hardware technologies without knowledge of the telephony radio or cellular network;

creating a separate proxy layer instance (235) in said proxy layer for each of the calling applications by the calling applications linking to a proxy library, wherein said driver layer (240) is shared among the proxy layer instances (235), and wherein the proxy layer instances and the driver layer use different address spaces;

receiving, from one of said calling applications, said API call to perform a particular call control function at a first interface of the proxy layer instance (235) associated to said one calling application, wherein said one calling application is a TSP or a TAPI interface;

transforming, by the proxy layer instance (235) associated to said one calling application, the API call to a command corresponding to

the API call to perform the particular call control function understood by the shared driver layer;

sending, by the proxy layer instance (235) associated to said one calling application, said command corresponding to the API call to perform the particular call control function to the shared driver layer at a second interface;

receiving, by said shared driver layer and from the proxy layer instance (235) associated to said one calling application, said command corresponding to the API call to perform the particular call control function at the second interface;

determining at least one standard telephony radio hardware command corresponding to API call to perform the particular call control function by said shared driver layer; and

sending, by said shared driver layer, the telephony radio hardware command corresponding to the API call to perform the the particular call control function to the telephony radio hardware at a third interface;

performing the particular function at the radio hardware; and

in response to successfully performing the particular function, sending a success code to the one of the plurality of applications that generated the API call;

wherein the API call, command and success code are associated with an identifier linking them together and linking them to the one of the plurality of applications that generated the API call;

the driver receives the success code;

using the identifier, the driver layer matches the success code with the one of the plurality of applications that generated the API call;

and

the driver layer sends the success code to the one of the plurality of applications that generated the API call.



7. A method of operating a telephone (200, 400) for facilitating communications between an application program module and a radio hardware (245, 485), characterized in that the method comprises the steps of:
  - calling (305), by each of a plurality of applications (205, 210, 215, 220, 225, 405), an API in a proxy layer to perform a particular function to be performed by the radio hardware;
  - creating, in a proxy layer comprised in a radio interface layer (430), a separate proxy layer instance (235) for each of the calling applications by the calling applications linking to a proxy library, wherein the proxy layer is for communicating with the applications at a first interface and a driver layer at a second interface and the driver layer communicates with the proxy layer at said second interface and the radio hardware at a third interface, and wherein the driver layer (240) is shared by the proxy layer instances (235), and the proxy layer instances (235) and the driver layer use different address spaces;
  - receiving, from one of said calling applications, said API call at a first interface of the proxy layer instance (235) associated to said one calling application, wherein said one calling application is a TSP or a TAPI interface;
  - transforming the API to an IOCTL by said proxy layer instance (235) associated to said one calling application;
  - sending (320) the IOCTL by said proxy layer instance associated to said one calling application to the driver layer (240) at said second interface;
  - receiving the IOCTL by the shared driver layer at said second interface;
  - transforming, by the shared driver layer, the IOCTL into a command understood by the radio hardware to perform the particular function, wherein the command understood by the radio hardware corresponds to the particular function; ~~and~~

sending said radio command to the radio hardware at said third interface;  
receiving, by the driver layer, communications from the radio hardware indicating that the particular function has been performed;  
sending, by the driver layer, a success code to the proxy layer instance from which the IOCTL has been received, the success code indicating that the particular function has been performed;  
generating in the driver layer a unique ID associated with the API;  
waiting (335) for a response from the radio hardware; and  
when received, calling back (335) said one calling application with the response and the unique ID returned from the API call;  
wherein the proxy layer instances are hardware independent and the driver layer is hardware specific.“

Die weiteren Patentansprüche gemäß Hilfsantrag 9 entsprechen den erteilten Unteransprüchen 2 bis 6, den neu nummerierten und mit entsprechend geänderten Rückbezügen versehenen erteilten Unteransprüchen 10, 11 (nunmehr 8, 9), 14 bis 30 (nunmehr 10 bis 26) sowie dem nebengeordneten Patentanspruch 31 in der erteilten Fassung (nunmehr 27).

Die Beklagte tritt den Ausführungen der Klägerin in allen Punkten entgegen. Das Streitpatent unterliege keinem Patentierungsausschluss und sei auch im Übrigen patentfähig, jedenfalls in einer der Fassungen der Hilfsanträge. Es sei weder unzulässig erweitert, noch mangle es ihm gegenüber dem Stand der Technik an Neuheit bzw. erfinderischer Tätigkeit. In Bezug auf die Druckschrift D2 bestreitet die Beklagte mit Nichtwissen, dass diese im September 2000 veröffentlicht worden sei.

Zum Nachweis ihres Verständnisses des Begriffes „IOCTL“ überreicht die Beklagte mit dem Schriftsatz vom 29. Oktober 2013 zudem einen Auszug aus einem Fachbuch:

**NK12:**W. Mauerer: *Linux Kernelarchitektur: Konzepte, Strukturen und Algorithmen von Kernel 2.6.*, München: Carl Hanser Verlag GmbH & Co. KG, 2004.

Die Klägerinnen machen weiterhin geltend, dass der Gegenstand des Streitpatents auch in der Fassung der Hilfsanträge gegenüber dem Stand der Technik mangels Neuheit bzw. fehlender erfinderische Tätigkeit nicht patentfähig sei.

Wegen der weiteren Einzelheiten des Vorbringens der Beteiligten wird auf den Akteninhalt verwiesen.

### **Entscheidungsgründe**

Die Klage, mit der u. a. der Nichtigkeitsgrund der fehlenden Patentfähigkeit nach Artikel II § 6 Absatz 1 Nr. 1 IntPatÜG, Artikel 138 Abs. 1 lit. a EPÜ i. V. m. Artikel 54 Absatz 1, 2 und Artikel 56 EPÜ geltend gemacht wird, ist zulässig. Sie ist auch begründet, denn das Streitpatent hat weder in der erteilten Fassung noch in der Fassung einer der Hilfsanträge Bestand, da ihm der vorgenannte Nichtigkeitsgrund der fehlenden Patentfähigkeit entgegensteht. Es bedarf daher keiner abschließenden Entscheidung, ob der Gegenstand des Patents unzulässig erweitert ist (Art. II § 6 IntPatÜG, Art. 138 Abs. 1 lit. c) EPÜ) bzw. ob die Lehre des Streitpatents unter den Patentierungsausschluss nach Art. 52 Abs. 2 lit. c, Abs. 3 EPÜ fällt.

#### **I.**

- 1)** Ausweislich der erteilten Beschreibung betrifft das Streitpatent im Allgemeinen Anwendungsprogrammierungsschnittstellen (APIs) und im Besonderen eine Funkschnittschicht im Rahmen eines Schichtenmodells bzw. einer Schichtenarchitektur, die einen Satz von APIs umfasst (vgl. Streitpatent, Abs. [0001] bzw. die entsprechende deutsche Übersetzung DE 602 14 059 T2).

- a) Gemäß der Beschreibungseinleitung des Streitpatents sollten Mobiltelefone (Zellulartelefone) idealerweise dieselbe Funktionalität wie Personalcomputer oder Handheld-PDAs aufweisen. Um derartige Anwendungen in eine Mobil- bzw. Zellulartelefonumgebung zu implementieren, müssten Anwendungsentwickler ihre Software zur Verwendung in einem Mobilfunktelefon (Zellulartelefon) entwickeln oder anpassen. Das Anpassen oder Entwickeln von Software zur Verwendung in einem Mobiltelefon eines Originalgeräteherstellers (OEM) sei jedoch nicht notwendigerweise ein Garant dafür, dass die Softwareanwendung auf dem Mobil-/ Zellulartelefon eines anderen Herstellers funktioniere, und zwar aufgrund der verschiedenen Funkhardwareimplementierungen der verschiedenen OEMs und aufgrund der Unterschiede bezüglich der verschiedenen Mobilfunkumgebungen (vgl. Streitpatent, Abs. [0002] bzw. die entsprechende deutsche Übersetzung DE 602 14 059 T2). Um eine Softwarelösung zu konzipieren, die an mehrere unterschiedliche Mobilfunksysteme und Funkhardware angepasst werden kann, bestehe die Notwendigkeit einer Hardwareanpassungsschicht, das heißt, einer Schicht, die die Besonderheiten eines bestimmten Mobilfunksystems bzw. einer bestimmten Hardware von dem Großteil des Softwaresystems isoliert. Des Weiteren bestehe die Notwendigkeit einer vordefinierten Schnittstelle, die von den Softwarekomponenten verwendet wird. Darüber hinaus bestehe die Notwendigkeit, dass es die Schicht den Hardwareherstellern ermöglichen sollte, die Implementierung der Hardwareschnittstelle zu ersetzen bzw. zu modifizieren, um ihre spezifische Hardware anzupassen (vgl. Streitpatent, Abs. [0003] bzw. die entsprechende deutsche Übersetzung DE 602 14 059 T2). Eine derartige Schicht (TAPI) existiere bereits für die Verwendung bei der Entwicklung eines allgemeinen Telefoniesystems. Die TAPI habe jedoch zwei Nachteile, die die Verwendung in einer Mobilfunkumgebung (zellularen Umgebung) erschweren: ein erheblicher Teil an Mobilfunk-spezifischer Funktionalität würde nicht durch die TAPI-Schnittstelle bereitgestellt, und es sei ziemlich schwierig, TAPI-Dienstanbieter (TSPs / TAPI Service Providers) zu

implementieren, wodurch es problematisch sei, das Softwaresystem an unterschiedliche Hardwaretypen anzupassen (vgl. Streitpatent, Abs. [0004] bzw. die entsprechende deutsche Übersetzung DE 602 14 059 T2). Aus diesem Grund bestehe die Notwendigkeit einer neuen Hardwareanpassungsschicht, die insbesondere besser an die Mobilfunkumgebung angepasst werden kann und die die Aufgabe erleichtert, die Schicht an unterschiedliche Hardwaretypen anzupassen (vgl. Streitpatent, Abs. [0007] bzw. die entsprechende deutsche Übersetzung DE 602 14 059 T2).

- b) Dem Streitpatent liegt daher die **Aufgabe** bzw. das technische Problem zugrunde, es den Anwendungen innerhalb einer Funkkommunikationsvorrichtung zu ermöglichen, mit der Funktelefoniehardware auf eine solche Art und Weise zu kommunizieren, dass die Anwendungen vollständig unabhängig von der Funktelefoniehardware oder einem Mobilfunknetz (Zellulernetz) programmiert werden können (vgl. deutsche Übersetzung des Streitpatents, Abs. [0007]).
- c) Die genannte Aufgabe wird gemäß Streitpatentschrift durch die in den unabhängigen Patentansprüchen 1 und 7 beschriebenen Verfahren gelöst. Der erteilte Patentanspruch 1 (Hauptantrag) hat in der Verfahrenssprache Englisch folgenden Wortlaut (Merkmalsgliederung hinzugefügt):

„1. A method for communicating with a telephony radio hardware (245,485), **characterized in that** the method comprises the steps of:

**M1** interfacing a computer (20, 49) to the telephony radio hardware by an abstraction layer comprising

**M1.1** a proxy layer (235) and

**M1.2** a driver layer (240);

**M2** calling one of a set of APIs comprised in said abstraction layer, wherein

**M2.1** the set of APIs correspond to call control functions and

**M2.2** are for abstracting out multiple radio hardware technologies without knowledge of the telephony radio or cellular network;

**M3** receiving said API call at a first interface of the proxy layer;

**M4** transforming the API call to a command understood by the driver layer and

**M5** sending said command to the driver layer at a second interface;

**M6** receiving said command at the second interface;

**M7** determining at least one standard telephony radio hardware command corresponding to the called API by said driver layer; and

**M8** sending, by said driver layer, the telephony radio hardware command to the telephony radio hardware at a third interface.”

Der erteilte unabhängige Anspruch 7 lautet (Merkmalsgliederung hinzugefügt):

„7. A method of operating a telephone (200, 400) for facilitating communications between an application program module and a radio hardware (245, 485), **characterized in that** the method comprises the steps of:

- N1** calling (305), by an application (205, 210, 215, 220, 225, 405), an API to perform a particular function;
- N2** receiving said API call at a first interface of a proxy layer (235);
- N3** transforming the API to an IOCTL by the proxy layer (235) comprised in a radio interface layer (430), wherein
  - N3.1** the proxy layer is for communicating with the application at the first interface and a driver layer at a second interface and
  - N3.2** the driver layer communicates with the proxy layer at said second interface and the radio hardware at a third interface;
- N4** sending (320) the IOCTL by the proxy layer to the driver layer (240) at said second interface;
- N5** receiving the IOCTL by the driver layer at said second interface;
- N6** transforming, by the driver layer, the IOCTL into a command understood by the radio hardware to perform the particular function; and
- N7** sending said radio command to the radio hardware at said third interface.”

In deutscher Übersetzung lauten die erteilten Ansprüche 1 und 7 (Merkmalsgliederung hinzugefügt):

- „1. Verfahren zur Kommunikation mit einer Telefoniefunkhardware (245; 485), **dadurch gekennzeichnet, dass** das Verfahren folgende Schritte umfasst:

- M1** Koppeln (interfacing) eines Computers (20; 49) an die Telefoniefunkhardware durch eine Abstraktionsschicht, welche
- M1.1** eine Proxyschicht (235) und
- M1.2** eine Treiberschicht (240) umfasst;
- M2** Aufrufen eines aus einem Satz in der Abstraktionsschicht enthaltener APIs, wobei
- M2.1** der Satz von APIs Anrufsteuerfunktionen entsprechen und
- M2.2** der Abstraktion aus mehreren Funkhardwaretechnologien ohne Kenntnis des Telefoniefunks oder des Zellularnetzes dienen;
- M3** Empfangen des API-Aufrufs an einer ersten Schnittstelle der Proxyschicht;
- M4** Transformieren des API-Aufrufs in einen Befehl, der von der Treiberschicht verstanden wird, und
- M5** Senden des Befehls an die Treiberschicht an einer zweiten Schnittstelle;
- M6** Empfangen des Befehls an der zweiten Schnittstelle;
- M7** Bestimmen mindestens eines Standardtelefoniefunkhardwarebefehls, der dem aufgerufenen API entspricht, durch die Treiberschicht; und
- M8** Senden des Telefoniefunkhardwarebefehls an die Telefoniefunkhardware an einer dritten Schnittstelle durch die Treiberschicht.“



- „7. Verfahren zum Betrieb eines Telefons (200; 400) zur Ermöglichung von Kommunikationen zwischen einem Anwendungsprogrammmodul und einer Funkhardware (245; 485), **dadurch gekennzeichnet, dass** das Verfahren folgende Schritte umfasst:
- N1** Aufrufen (305) eines API durch eine Anwendung (205, 210, 215, 220, 225, 405), um eine bestimmte Funktion durchzuführen;
- N2** *Empfangen des API-Aufrufs an einer ersten Schnittstelle einer Proxyschicht (235);*
- N3** Transformieren des APIs in einen IOCTL durch die Proxyschicht (235), welche in einer Funkschnittstellenschicht (430) enthalten ist, wobei
- N3.1** die Proxyschicht der Kommunikation mit der Anwendung an der ersten Schnittstelle und einer Treiberschicht an einer zweiten Schnittstelle dient und
- N3.2** die Treiberschicht mit der Proxyschicht an der zweiten Schnittstelle und mit der Funkhardware an einer dritten Schnittstelle kommuniziert;
- N4** Senden (320) des IOCTL durch die Proxyschicht an die Treiberschicht (240) an der zweiten Schnittstelle;
- N5** Empfangen des IOCTL durch die Treiberschicht an der zweiten Schnittstelle;
- N6** Transformieren des IOCTL in einen Befehl, der durch die Funkhardware verstanden wird, um eine bestimmte Funktion auszuführen, durch die Treiberschicht; und

**N7** Senden des Funkbefehls an die Funkhardware an der dritten Schnittstelle.“

Anspruch 1 nach Hilfsantrag 1 weist die Merkmale des Anspruchs 1 nach Hauptantrag auf unter Hinzufügung eines Merkmals im Anschluss an das Merkmal M1.2, wobei das hinzugefügte Merkmal in deutscher Übersetzung wie folgt lautet:

**M1.3** „wobei die Proxyschicht Hardware-unabhängig und die Treiberschicht Hardware-spezifisch ist“

Der unabhängige Anspruch 7 nach Hilfsantrag 1 weist die Merkmale des Anspruchs 7 nach Hauptantrag auf unter Hinzufügung eines Merkmals im Anschluss an das Merkmal N3.2, wobei das hinzugefügte Merkmal in deutscher Übersetzung wie folgt lautet:

**N3.3** „wobei die Proxyschicht Hardware-unabhängig und die Treiberschicht Hardware-spezifisch ist;“

Anspruch 1 nach Hilfsantrag 2 weist die Merkmale des Anspruchs 1 nach Hilfsantrag 1 auf unter Hinzufügung von Merkmalen, die in deutscher Übersetzung wie folgt lauten (Merkmalsgliederung hinzugefügt):

**M9** „Erzeugen eines API-Aufrufs in einer von einer Vielzahl von Anwendungen (205, 210, 215, 220, 225, 405), um die bestimmte Funktion durchzuführen;

**M10** Senden des API-Aufrufs an einen Funktreiber;

**M11** Konvertieren des API-Aufrufs in einen Befehl, der von der Funkhardware verstanden wird, im Funktreiber;

**M12** Übertragen des Befehls an die Funkhardware;

**M13** Durchführen der bestimmten Funktion durch die Funkhardware; und

**M14** in Erwiderung auf eine erfolgreiche Durchführung der bestimmten Funktion, Senden eines Erfolgscodes an die eine der Vielzahl von Anwendungen, die den API-Aufruf erzeugt hat;

wobei

**M15** das API, der Befehl und der Erfolgscode mit einem Identifikator assoziiert sind, der sie miteinander verbindet und der sie mit der einen der Vielzahl von Anwendungen verbindet, die den API-Aufruf erzeugt hat;

**M16** der Funktreiber den Erfolgscode empfängt;

**M17** der Funktreiber den Erfolgscode mit der einen der Vielzahl von Anwendungen, die den API-Aufruf erzeugt hat, abgleicht, indem er den Identifikator verwendet; und

**M18** der Funktreiber den Erfolgscode an die eine der Vielzahl von Anwendungen, die den API-Aufruf erzeugt hat, sendet.“

Anspruch 7 nach Hilfsantrag 2 weist die Merkmale des Anspruchs 7 nach Hilfsantrag 1 auf unter Hinzufügung von Merkmalen, die in deutscher Übersetzung wie folgt lauten (Merkmalsgliederung hinzugefügt):

**N8** „Empfangen von Kommunikationen von der Funkhardware durch die Treiberschicht, wobei die Kommunikationen angeben, dass die bestimmte Funktion durchgeführt wurde; und

**N9** Senden eines Erfolgscodes an die Proxyschicht, der angibt, dass die bestimmte Funktion durchgeführt wurde, durch die Treiberschicht;

- N10** wobei das Telefon die Proxyschicht, die Treiberschicht, die Anwendung und die Funkhardware umfasst;
- N11** die Anwendung das API in der Proxyschicht aufruft;
- N12** das API mit der durch die Funkhardware durchzuführenden Funktion assoziiert ist;
- N13** der Befehl, der von der Funkhardware verstanden wird, der Funktion entspricht;  
und das Verfahren ferner die Schritte umfasst:
- N14** Senden (325) des Befehls an die Funkhardware;
- N15** Erzeugen eines eindeutigen mit dem API assoziierten Identifikators in der Treiberschicht;
- N16** Warten (335) auf eine Antwort von der Funkhardware; und
- N17** wenn diese empfangen wird, Zurückrufen (335) der aufrufenden Anwendung mit der von dem Aufruf zurückgegebenen Antwort und eindeutigem Identifikator.“

Anspruch 1 nach Hilfsantrag 3 weist die Merkmale M1 und M1.1 bis M1.3 des Anspruchs 1 nach Hilfsantrag 1 auf unter Hinzufügung von Merkmalen, die in deutscher Übersetzung wie folgt lauten (Merkmalsgliederung hinzugefügt; Änderungen gegenüber den Merkmalen des Anspruchs 1 nach Hilfsantrag 1 markiert):

- M2\*** „Aufrufen eines aus einem Satz in der Abstraktionsschicht enthaltener APIs aus jeder einer Vielzahl von Anwendungen (205, 210, 215, 220, 225, 405), wobei

- M2.1\*** der Satz von APIs Anrufsteuerfunktionen entsprechen und
- M2.2\*** der Abstraktion aus mehreren Funkhardwaretechnologien ohne Kenntnis des Telefoniefunks oder des Zellulernetzes dienen;
- M2.3\*** Erzeugen einer separaten Proxy-Instanz (235) in der Proxyschicht für jede der aufrufenden Anwendungen indem die aufrufenden Anwendungen eine Proxybibliothek linken, wobei die Proxyschicht-Instanzen die Treiberschicht (240) gemeinsam benutzen und wobei die Proxyschicht-Instanzen und die Treiberschicht verschiedene Adressräume nutzen;
- M3\*** Empfangen des API-Aufrufs von einer der aufrufenden Anwendungen an einer ersten Schnittstelle der Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert ist;
- M4\*** Transformieren des API-Aufrufs durch die Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert ist in einen Befehl, der von der Treiberschicht verstanden wird, ~~und~~
- M5\*** Senden des Befehls durch die Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert ist, an die gemeinsam benutzte Treiberschicht an einer zweiten Schnittstelle;
- M6\*** Empfangen des Befehls von der Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert ist an der zweiten Schnittstelle durch die gemeinsam benutzte Treiberschicht;
- M7\*** Bestimmen mindestens eines Standardtelefoniefunkhardwarebefehls, der dem aufgerufenen API entspricht, durch die gemeinsam benutzte Treiberschicht; und

**M8\*** Senden des Telefoniefunkhardwarebefehls an die Telefoniefunkhardware an einer dritten Schnittstelle durch die gemeinsam benutzte Treiberschicht.“

Anspruch 7 nach Hilfsantrag 3 lautet in deutscher Übersetzung (Merkmalsgliederung hinzugefügt; Änderungen gegenüber den Merkmalen des Anspruchs 7 nach Hilfsantrag 1 markiert):

„Verfahren zum Betrieb eines Telefons (200; 400) zur Ermöglichung von Kommunikationen zwischen einem Anwendungsprogrammmodul und einer Funkhardware (245; 485), dadurch gekennzeichnet, dass das Verfahren folgende Schritte umfasst:

**N1\*** Aufrufen (305) eines API durch eine jede einer Vielzahl von Anwendungen (205, 210, 215, 220, 225, 405), um eine bestimmte Funktion durchzuführen;

**N1.1\*** Erzeugen einer separaten Proxy-Instanz (235) in der Proxyschicht einer Funkschnittstellenschicht (430) für jede der aufrufenden Anwendungen indem die aufrufenden Anwendungen eine Proxybibliothek laden, wobei die Proxyschicht der Kommunikation mit den Anwendungen an der ersten Schnittstelle und einer Treiberschicht an einer zweiten Schnittstelle dient und die Treiberschicht mit der Proxyschicht an der zweiten Schnittstelle und mit der Funkhardware an einer dritten Schnittstelle kommuniziert, wobei die Proxyschicht-Instanzen (235) die Treiberschicht (240) gemeinsam benutzen und wobei die Proxyschicht-Instanzen (235) und die Treiberschicht verschiedene Adressräume nutzen;

- N2\*** Empfangen des API-Aufrufs einer der aufrufenden Anwendungen an der ersten Schnittstelle der Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert ist;
- N3\*** Transformieren des APIs in einen IOCTL durch die Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert ist, ~~welche in einer Funkschnittstellenschicht (430) enthalten ist,~~ wobei
- ~~**N3.1\*** die Proxyschicht der Kommunikation mit der Anwendung an der ersten Schnittstelle und einer Treiberschicht an einer zweiten Schnittstelle dient und~~
- ~~**N3.2\*** die Treiberschicht mit der Proxyschicht an der zweiten Schnittstelle und mit der Funkhardware an einer dritten Schnittstelle kommuniziert;~~
- ~~**N3.3\*** wobei die Proxyschicht Hardware-unabhängig und die Treiberschicht Hardware-spezifisch ist;~~
- N4\*** Senden (320) des IOCTL durch die Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert ist an die gemeinsam benutzte Treiberschicht (240) an der zweiten Schnittstelle;
- N5\*** Empfangen des IOCTL durch die gemeinsam benutzte Treiberschicht an der zweiten Schnittstelle;
- N6\*** Transformieren des IOCTL in einen Befehl, der durch die Funkhardware verstanden wird, um eine bestimmte Funktion auszuführen, durch die gemeinsam benutzte Treiberschicht; und

**N7\*** Senden des Funkbefehls an die Funkhardware an der dritten Schnittstelle.

**N8\*** wobei die Proxyschicht Hardware-unabhängig und die Treiberschicht Hardware-spezifisch ist.

Anspruch 1 nach Hilfsantrag 4 weist die Merkmale M1 bis M2.3\* des Anspruchs 1 nach Hilfsantrag 3 auf unter Hinzufügung von Merkmalen, die in deutscher Übersetzung wie folgt lauten (Merkmalsgliederung hinzugefügt; Änderungen gegenüber den Merkmalen des Anspruchs 1 nach Hilfsantrag 3 markiert):

**M3\*\*** „Empfangen des API-Aufrufs, eine bestimmte Anrufsteuerfunktion auszuführen, von einer der aufrufenden Anwendungen an einer ersten Schnittstelle der Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert ist;

**M4\*\*** Transformieren des API-Aufrufs durch die Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert ist in einen Befehl, der dem API-Aufruf eine bestimmte Anrufsteuerfunktion auszuführen entspricht und der von der gemeinsam genutzten Treiberschicht verstanden wird,

**M5\*\*** Senden des Befehls, der dem API-Aufruf eine bestimmte Anrufsteuerfunktion auszuführen entspricht, durch die Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert ist, an die gemeinsam benutzte Treiberschicht an einer zweiten Schnittstelle;

**M6\*\*** Empfangen des Befehls, der dem API-Aufruf eine bestimmte Anrufsteuerfunktion auszuführen entspricht von Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert



ist, an der zweiten Schnittstelle durch die gemeinsam benutzte Treiberschicht;

- M7\*\*** Bestimmen mindestens eines Standardtelefoniefunkhardwarebefehls, der dem aufgerufenen API-Aufruf eine bestimmte Anrufsteuerfunktion auszuführen entspricht, durch die gemeinsam benutzte Treiberschicht; und
- M8\*\*** Senden des Telefoniefunkhardwarebefehls, der dem API-Aufruf eine bestimmte Anrufsteuerfunktion auszuführen entspricht, an die Telefoniefunkhardware an einer dritten Schnittstelle durch die gemeinsam benutzte Treiberschicht.
- M9\*\*** Durchführen der bestimmten Anrufsteuerfunktion durch die Funkhardware;
- M10\*\*** in Erwiderung auf eine erfolgreiche Durchführung der bestimmten Anrufsteuerfunktion durch die Funkhardware, Senden eines Erfolgscodes an die eine der Vielzahl von Anwendungen, die den API-Aufruf erzeugt hat; wobei
- M11\*\*** der API-Aufruf, der Befehl und der Erfolgscode mit einem Identifikator assoziiert sind, der sie miteinander verbindet und der sie mit der einen der Vielzahl von Anwendungen verbindet, die den API-Aufruf erzeugt hat;
- M12\*\*** der Funktreiber den Erfolgscode empfängt;
- M13\*\*** der Funktreiber den Erfolgscode mit der einen der Vielzahl von Anwendungen, die den API-Aufruf erzeugt hat, abgleicht, indem er den Identifikator verwendet; und

**M14\*\*** der Funktreiber den Erfolgscode an die eine der Vielzahl von Anwendungen, die den API-Aufruf erzeugt hat, sendet.

Der nebengeordnete Anspruch 7 nach Hilfsantrag 4 lautet in der deutschen Übersetzung (Merkmalsgliederung hinzugefügt; Änderungen gegenüber dem Anspruch 7 nach Hilfsantrag 3 markiert):

„7. Verfahren zum Betrieb eines Telefons (200; 400) zur Ermöglichung von Kommunikationen zwischen einem Anwendungsprogrammmodul und einer Funkhardware (245; 485), dadurch gekennzeichnet, dass das Verfahren folgende Schritte umfasst:

**N1\*\*** Aufrufen (305) eines API in einer Proxyschicht durch eine jede einer Vielzahl von Anwendungen (205, 210, 215, 220, 225, 405), um eine bestimmte Funktion durchzuführen, die von der Funkhardware ausgeführt werden soll;

**N1.1\*\*** Erzeugen einer separaten Proxy-Instanz (235) in der Proxyschicht einer Funkschnittstellenschicht (430) für jede der aufrufenden Anwendungen indem die aufrufenden Anwendungen eine Proxybibliothek laden, wobei die Proxyschicht der Kommunikation mit den Anwendungen an der ersten Schnittstelle und einer Treiberschicht an einer zweiten Schnittstelle dient und die Treiberschicht mit der Proxyschicht an der zweiten Schnittstelle und mit der Funkhardware an einer dritten Schnittstelle kommuniziert, wobei die Proxyschicht-Instanzen (235) die Treiberschicht (240) gemeinsam benutzen und wobei die Proxyschicht-Instanzen (235) und die Treiberschicht verschiedene Adressräume nutzen;

**N2\*\*** Empfangen des API-Aufrufs einer der aufrufenden Anwendungen an der ersten Schnittstelle der Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert ist;

- N3\*\*** Transformieren des APIs in einen IOCTL durch die Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert ist,
- N4\*\*** Senden (320) des IOCTL durch die Proxyschicht-Instanz (235), die mit der einen aufrufenden Anwendung assoziiert ist an die gemeinsam benutzte Treiberschicht (240) an der zweiten Schnittstelle;
- N5\*\*** Empfangen des IOCTL durch die gemeinsam benutzte Treiberschicht an der zweiten Schnittstelle;
- N6\*\*** Transformieren des IOCTL in einen Befehl, der durch die Funkhardware verstanden wird, um eine bestimmte Funktion auszuführen, durch die gemeinsam benutzte Treiberschicht, wobei der Befehl, der durch die Funkhardware verstanden wird, der einen bestimmten Funktion entspricht; und
- N7\*\*** Senden des Funkbefehls an die Funkhardware an der dritten Schnittstelle.
- N8\*\*** wobei die Proxyschicht Hardware-unabhängig und die Treiberschicht Hardware-spezifisch ist;
- N9\*\*** Empfangen von Kommunikationen von der Funkhardware, die angeben, dass die bestimmte Funktion durchgeführt wurde, durch die Treiberschicht;
- N10\*\*** Senden eines Erfolgscodes an die Proxyschicht-Instanz, von der der IOCTL empfangen wurde, wobei der Erfolgscode angibt, dass die bestimmte Funktion durchgeführt wurde, durch die Treiberschicht;

**N11\*\*** Erzeugen eines eindeutigen mit dem API assoziierten Identifikators in der Treiberschicht;

**N12\*\*** Warten (335) auf eine Antwort von der Funkhardware; und

**N13\*\*** wenn diese empfangen wird, Zurückrufen (335) der aufrufenden Anwendung mit der von dem Aufruf zurückgegebenen Antwort und eindeutigem Identifikator;

**N14\*\*** wobei die Proxyschicht Hardware-unabhängig und die Treiberschicht Hardware-spezifisch ist.“

Die jeweiligen Ansprüche 1 nach den Hilfsanträgen 5 und 6 entsprechen dem Anspruch 1 nach Hauptantrag bzw. dem Anspruch 1 nach Hilfsantrag 1 unter Änderung des jeweiligen Merkmals M3 wie folgt (in deutscher Übersetzung; hinzugefügtes Teilmerkmal markiert):

**M3\*\*\*** „Empfangen des API-Aufrufs von einem TSP oder einer TAPI-Schnittstelle an einer ersten Schnittstelle der Proxyschicht;“

Die jeweiligen nebengeordneten Ansprüche 7 nach den Hilfsanträgen 5 und 6 entsprechen den jeweiligen Ansprüchen 7 nach Hauptantrag bzw. Hilfsantrag 1 unter Änderung des Merkmals N1 (in deutscher Übersetzung; hinzugefügtes Teilmerkmal markiert):

**N1\*\*\*** „Aufrufen (305) eines API durch eine Anwendung (205, 210, 215, 220, 225, 405), um eine bestimmte Funktion durchzuführen, wobei die Anwendung ein TSP oder eine TAPI-Schnittstelle ist;“

Die jeweiligen nebengeordneten Ansprüche 1 nach Hilfsanträgen 7 bis 9 entsprechen den jeweiligen Ansprüchen 1 nach Hilfsanträgen 2 bis 4 unter Hinzufügung des folgenden Merkmals im Anschluss an das Merkmal M9

gemäß Hilfsantrag 2 bzw. an das Merkmal M3\* gemäß Hilfsantrag 3 bzw. an das Merkmal M3\*\* gemäß Hilfsantrag 4:

„wobei die eine aufrufende Anwendung ein TSP oder eine TAPI-Schnittstelle ist“

Die jeweiligen nebengeordneten Ansprüche 7 nach Hilfsanträgen 7 bis 9 entsprechen den jeweiligen Ansprüchen 7 nach Hilfsanträgen 2 bis 4 unter Hinzufügung des vorstehend aufgeführten Merkmals („wobei die eine aufrufende Anwendung ein TSP oder eine TAPI-Schnittstelle ist“) im Anschluss an das Merkmal N1 gemäß Hilfsantrag 2 bzw. an das Merkmal N1\* gemäß Hilfsantrag 3 bzw. an das Merkmal N1\*\* gemäß Hilfsantrag 4.

- 2) Als zuständiger **Fachmann** ist ein Elektrotechnik-Ingenieur oder Informatiker mit Fachhochschulabschluss und mehrjähriger Erfahrung in der Entwicklung von Funkschnittstellen im Bereich Telekommunikation anzusehen.
  
- 3) Der in den Ansprüchen aufgeführte Begriff „Proxy layer“ (bzw. der Begriff „Proxyschicht“ in der deutschen Übersetzung des Streitpatents) bedarf der Erläuterung. Laut Streitpatent bildet die Proxyschicht eine Schicht, die durch mehrere andere Schichten der Kernarchitektur (wie beispielsweise eine TSP-Schicht, eine ExtAPI-Schicht und einen SIM-Manager) unter Verwendung der plattformspezifischen Befehle dieser Kernarchitekturen aufgerufen werden kann. Gemäß einem Ausführungsbeispiel des Streitpatents handelt es sich bei der Proxyschicht um eine dynamische Verbindungsbibliothek bzw. dynamische Programmibibliothek (Dynamic Link Library / DLL), die Rückruf-Mitteilungen und Funktions-Aufrufe zwischen Prozessen verwaltet und dabei die spezifischen Befehle der Kernarchitektur in API-Aufrufe umwandelt, die von der Treiberschicht verstanden werden (vgl. Abs. [0026] der deutschen Übersetzung des Streitpatents). In Übereinstimmung mit der Beschreibungseinleitung des Streitpatents (vgl. Abs. [0004]) versteht der Fachmann unter dem im Zusammenhang mit der Proxyschicht bzw. einer Abstraktionsschicht

aufgeführten TSP (TAPI Service Provider / Telephony Service Provider) einen Dienst für computergestützte Telefonie im Rahmen eines Schichtenmodells.

## II.

Das Streitpatent erweist sich sowohl in der erteilten Fassung als auch im Umfang der Hilfsanträge 1 bis 9 nicht als patentfähig, da sich die darin beanspruchte Lehre für den Fachmann in naheliegender Weise aus dem Stand der Technik ergibt (Artikel II § 6 Absatz 1 Nr. 1 IntPatÜG, Artikel 138 Abs. 1 lit a EPÜ i. V. m. Artikel 56 EPÜ).

### 1) Hauptantrag

Mit der nach Hauptantrag beanspruchten Fassung der Patentansprüche kann das angegriffene Patent nicht erfolgreich verteidigt werden.

- a) Das Verfahren gemäß Anspruch 1 nach Hauptantrag ergibt sich für den Fachmann in nahe liegender Weise aus der Kenntnis der Druckschrift **D6**. Der Gegenstand des Anspruchs 1 nach Hauptantrag beruht daher nicht auf einer erfinderischen Tätigkeit. Der Anspruch 1 nach Hauptantrag ist somit nicht patentfähig.

Denn die Druckschrift **D6** offenbart ein Verfahren zur Kommunikation mit einer Telefoniehardware („telephony hardware“; vgl. Fig. 1.2 und den zugehörigen Text auf S. 6 bis S. 8 sowie Einleitungsseite xix, letzter Abs. und S. 13, letzter Abs. bis S. 14, erster Abs.), wobei in der Einleitungsseite xix im letzten Absatz auch auf drahtlose Telefon-Kommunikation und auf Mobiltelefonie als Anwendungsgebiet hingewiesen wird („telephone network communication [...] wireless, cellular“). Daher entnimmt der Fachmann der Druckschrift **D6** aufgrund seines Fachwissens über Mobiltelefonie bzw. Funktelefonie (Telefoniefunk), dass es sich bei der im Weiteren in der Druckschrift **D6** aufgeführten Telefoniehardware ebenfalls um

Funktelefoniehardware (Telefoniefunkhardware) handelt (**einleitendes Merkmal des Anspruchs 1**). Das in der Druckschrift **D6** offenbarte Verfahren umfasst dabei das Koppeln eines Computers an die vorstehend genannte Telefoniehardware bzw. Funktelefoniehardware (Telefoniefunkhardware) durch eine Abstraktionsschicht („layer of abstraction“, vgl. S. 14, erster Abs. im Abschnitt „Connection-model Independence“ i. V. m. S. 7, Fig. 1.2 / **Merkmal M1**). Diese Abstraktionsschicht weist eine Proxyschicht in Form einer sogenannten TAPI-Implementierung („TAPI implementation“) für eine Telefonie-Anwendungsprogrammierschnittstelle („Telephony API“) auf (**Merkmal M1.1**) und eine TSP-Treiberschicht („Telephony Service Provider TSP“), die u. a. Treibersätze für klassische Telefon-Hardware (PBX, POTS), ISDN-Hardware und Fax/Modem-Hardware sowie einen „Unimodem“-Treibersatz umfasst (vgl. Fig. 1.2 und den zugehörigen Text auf S. 6, letzter Abs. und S. 7; sowie Fig. 7.1 mit Beschreibung S. 181, zweiter. Abs., bis S. 182, zweiter Abs. / **Merkmal M1.2**).

Des Weiteren ist der Druckschrift **D6** im Zusammenhang mit einer Anwendungsprogrammierschnittstelle („TAPI“) für Telefonie-Anwendungen auch das Aufrufen der vorstehend genannten Anwendungsprogrammierschnittstelle („Telephony API“) zu entnehmen, die in der Abstraktionsschicht bzw. TAPI-Implementierung enthaltenen ist (vgl. S. 181, zweiter Abs.: „[...] function call to API requesting some service on a telephony device“, und Fig. 1.2 sowie den Text auf S. 7, dritter Abs.: „the goals of the TAPI: [...] 2. Access to data via existing standard APIs“ / **Merkmal M2**<sub>teilweise</sub>, ohne explizite Nennung eines API aus einem Satz APIs). Dabei ist der Druckschrift **D6** bereits zu entnehmen, dass die „APIs“ (vgl. S. 7, dritter Abs. Punkt 2) mit zugehörigen Anrufsteuerfunktionen (u. a. „Fax“, „Dialer“) verbunden sind bzw. diesen Funktionen entsprechen (vgl. die in Fig. 1.2 über der „TAPI“ dargestellte Schicht mit Telefonie-Anwendungen sowie den zugehörigen Text auf S. 7 i. V. m. S. 41, letzter Abs. / **Merkmal 2.1**). Weiterhin entnimmt der Fachmann den vorstehend genannten Zitatstellen in der Druckschrift **D6**, dass die Anwendungsprogrammierschnittstellen

(APIs) der Plattform- bzw. der Hardwareunabhängigkeit und folglich der Abstraktion bzgl. mehrerer Funkhardwaretechnologien ohne Kenntnis der Funktelefonie oder des Mobilfunk- bzw. Zellulernetzes dienen (vgl. S. 7, erster bis dritter Abs.: „Applications are programmed using TAPI [...] Each TSP then uses whatever interface is appropriate to control its telephony hardware [...] This layered approach makes it possible for an application to be developed without the developers having to worry about the specific hardware provided on a particular machine“, i. V. m. S. 13, letzter Abs., in dem der Fachmann auf die Anwendungsmöglichkeit bzgl. Funktelefoniehardware bzw. Mobiltelefonie („wireless, cellular“) hingewiesen wird / **Merkmal M2.2**).

Der Fachmann entnimmt der Druckschrift **D6** dabei aufgrund seines Fachwissens über Schichtenmodelle im Zusammenhang mit dem in der Fig. 1.2 und dem zugehörigen Text beschriebenen Modell zur Kommunikation mit Telefoniehardware mittels einer Schnittstelle für Telefonieanwendungen („TAPI“), einer Treiberschicht („TSP“) und einer TSP-Schnittstelle („TSPI“) auch folgende Verfahrensschritte:

- das Empfangen des zuvor im Zusammenhang mit Merkmal M2 genannten API-Aufrufs an einer ersten Schnittstelle („TAPI“) der Proxyschicht (vgl. die in Fig. 1.2 zwischen den Telefonieanwendungen und der Proxyschicht bzw. „TAPI Implementation“ eingezeichneten (ersten) Schnittstelle „TAPI“ / **Merkmal M3**),
- ein Transformieren des API-Aufrufs durch die Proxyschicht („TAPI Implementation“) in einen Befehl, der von der Treiberschicht („Telephony Service Provider“ / „TSP“) verstanden wird (vgl. Fig. 1.2 und dem zugehörigen Text auf S. 7, zweiter Abs., sowie Beispiel zu „Unimodem“, S. 182, erster vollständiger Satz, und zweiter Abs.: „For example, the Unimodem TSP [...] maps its requests to Hayes AT commands that are sent to the modem via a serial port“ / **Merkmal M4**),



- ein Senden des Befehls an die Treiberschicht („Telephony Service Provider“ / „TSP“) an einer zweiten Schnittstelle (Telephony Service Provider Interface „TSPI“; vgl. Beispiel zu „Unimodem“, S. 182, erster vollständiger Satz, und zweiter Abs.: „[...] maps its requests to Hayes AT commands that are sent to the modem via a serial port [...] To access the TSP’s functionality, TAPI requires each TSP be packaged as a DLL with a set of well known entry points“ / **Merkmal M5**),
- ein Empfangen des gesendeten Befehls an der zweiten Schnittstelle „TSPI“ (implizit aus dem „Unimodem“-Beispiel, S. 182, erster vollständiger Satz, und zweiter Abs.: „[...] sent to the modem via a serial port“; hieraus folgt auch zwangsläufig der Empfang durch das Interface der Treiberschicht / **Merkmal M6**),
- ein Bestimmen von Standardtelefoniefunkhardwarebefehlen, die den Standard-Anwendungsprogrammierschnittstellen („existing standard APIs“) entsprechen; dies liest der Fachmann in der Druckschrift **D6** im Zusammenhang mit den „AT Commands“ als Standardtelefoniehardwarebefehle zum Betrieb eines Modems (vgl. „Unimodem“-Beispiel S. 182, erster vollständiger Satz, und zweiter Abs.) in Verbindung mit den weiteren vorgesehenen Treibersätzen für Telefoniehardware (vgl. Fig. 1.2 mit zugehörigem Text auf S. 7, dritter Abs.) und dem Hinweis auf drahtlose und Mobiltelefonie als konkretes Anwendungsgebiet (vgl. S. 13, letzter Abs.) mit (**Merkmal M7**), sowie
- ein Senden des vorstehend genannten Telefoniefunkhardwarebefehls an die Telefoniefunkhardware an einer dritten Schnittstelle, die zwischen der in Fig. 1.2 dargestellten Treiberschicht („TSP“) und der darunter dargestellten Hardware (u. a. „Fax/Modem Hardware“) ausgebildet ist (vgl. „Unimodem“-Beispiel, S. 182, erster vollständiger Satz, und zweiter Abs.: „[...] sent to the modem via a

serial port“, wobei der Fachmann aufgrund der vorstehend genannten Hinweise hier in Analogie Befehle für eine Standardfunktelefoniehardware mitliest / **Merkmal M8**).

Damit unterscheidet sich das im Anspruch 1 aufgeführte Verfahren von der Lehre der Druckschrift **D6** darin, dass im Merkmal M2 explizit ein Satz in der Abstraktionsschicht enthaltener APIs aufgeführt wird. Wie bereits zuvor im Hinblick auf das Merkmal M2 ausgeführt, lehrt die Druckschrift **D6** jedoch bereits, dass die in der Abstraktionsschicht enthaltene TAPI-Implementierung dazu dient, auf Daten mittels existierender Standard-Anwendungsprogrammierungsschnittstellen („existing standard APIs“) zuzugreifen (vgl. Fig. 1.2 und den zugehörigen Text auf S. 7, letzter Abs.: „the goals of TAPI: [...] 2. Access to data via existing standard APIs“). Aufgrund dieser Lehre der Druckschrift **D6**, mittels „TAPI“ auf existierende Standard-Anwendungsprogrammierschnittstellen („APIs“) zuzugreifen, hat der Fachmann hinreichend Veranlassung, die vorstehend genannte Abstraktionsschicht so auszubilden, dass diese nicht nur eine einzige Anwendungsprogrammierschnittstelle / API enthält, sondern einen Satz von APIs, der den vorstehend genannten Mehrzahl an Anrufsteuerfunktionen (u. a. „Fax“, „Dialer“) entspricht (**Merkmal M2<sub>Rest</sub>**).

Den Ausführungen der Patentinhaberin in Bezug auf die im Anspruch 1 aufgeführte Telefoniefunkhardware und die Abstraktionsschicht, dass eine API- bzw. eine TAPI-Schnittstelle, wie sie in der Druckschrift **D6** aufgeführt sind, auf Festnetztelefonie – und nicht auf drahtlose Telefonie bzw. Mobilfunk – ausgerichtet seien, wobei die in der Figur 1.2 dargestellten Schichten auch nicht der im Streitpatent beanspruchten Abstraktionsschicht mit einer Proxy- und einer Treiberschicht entsprechen würden, kann nicht beigetreten werden. Denn der Fachmann entnimmt der Druckschrift **D6** – wie vorstehend dargelegt –, dass es sich hier ebenso um drahtlose Telefonie handelt („telephone network communication [...] wireless, cellular“, S. 13, letzter Satz) mit entsprechender Hardware in Form einer

Funktelefoniehardware (Telefoniefunkhardware) sowie um eine damit zusammenhängende Abstraktionsschicht („layer of abstraction“), die auch eine Proxyschicht und eine Treiberschicht in Form von dynamischen Verbindungsbibliotheken (DLLs) beinhaltet, wie es auch im Streitpatent aufgeführt ist (vgl. u. a. Abs. [0025] und Abs. [0026] der deutschen Übersetzung des Streitpatents). Auch den Ausführungen der Patentinhaberin in der mündlichen Verhandlung, dass die Druckschrift **D6** nicht notwendigerweise die Transformation eines API-Aufrufs offenbare, da auf den Seiten 181 bis 183 der Druckschrift **D6** im Zusammenhang mit der Figur 7.2 lediglich ein „Mapping“ bzw. „to map“ aufgeführt sei, kann nicht zugestimmt werden. Denn das Wort „transforming“, welches im Merkmal M4 des Anspruchs 1 in nicht weiter spezifizierter Form in Bezug auf einen API-Aufruf aufgeführt wird, bedeutet ganz allgemein ein geeignetes Überführen eines API-Aufrufs, also auch eine Abbildung, wie es auf den Seiten 181 bis 183 der Druckschrift **D6** im Zusammenhang mit der Ansteuerung der Hardware und einem „Mapping“ von Aufrufen („calls“) in Bezug auf passende Einsprungspunkte der TSP-Treiberschicht offenbart ist („to map the call to the proper TSP’s entry points“). Bereits aus der Hardwareunabhängigkeit der TAPI-Schnittstelle (Proxyschicht) und der Hardwareabhängigkeit der Treiber, welche dem Aufrufen spezifischer Hardwarefunktionen dienen, ist ersichtlich, dass sich beide Schnittstellen unterscheiden und das „Mapping“ der Aufrufe gemäß Druckschrift **D6** zwangsläufig ein Überführen einer hardwareunabhängigen in eine hardwareabhängige Form – also auch ein „Transformieren“ des Aufrufs – bedeutet. Im Übrigen wird hierzu auch die vorherigen Ausführungen zum API-Aufruf und zur Hardwareunabhängigkeit verwiesen (vgl. die Ausführungen zu Merkmal M2.2), die hier in gleicher Weise gelten. Auch die von der Patentinhaberin vorgebrachte Argumentation, ein Standardtelefoniefunkhardwarebefehl, wie er in den Merkmalen M7 und M8 des Anspruchs 1 aufgeführt werde, sei kein AT-Befehl, wie er in der Figur 1.2 der Druckschrift **D6** offenbart sei, kann nicht zugestimmt werden. Denn, wie vorstehend ausgeführt, entnimmt der Fachmann der Druckschrift

**D6**, dass es sich bei der dort offenbarten Hardware um Funktelefoniehardware bzw. Mobiltelefonie-Hardware („wireless, cellular“) handelt – in diesem Zusammenhang ist es für den Fachmann selbstverständlich, dass die vorstehend zitierten AT-Befehle („AT Commands“) als Standardtelefoniehardwarebefehle für Modems entsprechend der weiter in Druckschrift **D6** beschriebenen Telefonie- und Mobiltelefoniehardware beispielgebend für entsprechende Telefoniehardwarebefehle und Funktelefoniehardwarebefehle (oder Telefoniefunkhardwarebefehle) stehen, zumal zum Prioritätszeitpunkt des Streitpatents Modem-Einrichtungen im Zusammenhang mit drahtloser Telefonie bzw. Mobiltelefonie – entsprechend der vorveröffentlichten Druckschrift **D6** – bekannt und gebräuchlich waren.

- b) Bei dieser Sachlage bedarf es keiner Entscheidung, ob der Gegenstand des Patentanspruchs 1 nach Hauptantrag über den Inhalt der Anmeldung in der ursprünglich eingereichten Fassung hinausgeht.
  
- c) Auch der Gegenstand des nebengeordneten Anspruchs 7 nach Hauptantrag ergibt sich für den Fachmann in nahe liegender Weise aus der Lehre der Druckschrift **D6**. Denn – wie bereits zum Anspruch 1 nach Hauptantrag ausgeführt – offenbart die Druckschrift **D6** ein Verfahren zur Kommunikation mit einer Telefoniehardware (vgl. S. xix, letzter Abs., S. 13, letzter Abs. bis S. 14, erster Abs. sowie Fig. 1.2 und den zugehörigen Text auf S. 6 bis S. 8), wobei in der der Druckschrift **D6**, Einleitungsseite xix, letzter Absatz, von drahtloser Telefon-Kommunikation bzw. Mobilfunk („telephone network communication [...] wireless, cellular“) die Rede ist. Dabei entnimmt der Fachmann der Druckschrift **D6** auch, dass es sich um ein Verfahren zum Betrieb eines Telefons zur Ermöglichung von Kommunikationen zwischen einem Anwendungsmodul im Zusammenhang mit einer Anwendungsprogrammierschnittstelle („API“) und einer Telefoniehardware handelt, wobei es sich bei der Telefoniehardware aufgrund des Hinweises auf drahtlose Telefon-Kommunikation auch um

Telefoniefunkhardware handelt (**einleitendes Merkmal des Anspruchs 7**). Des Weiteren entnimmt der Fachmann der Druckschrift **D6** im Zusammenhang mit dem in der Figur 1.2 und dem zugehörigen Text beschriebenen Modell zur Kommunikation mit Telefoniehardware mittels einer TAPI-Schnittstelle, einer TSP-Treiberschicht und einer TSPI-Schnittstelle folgende damit verbundene Verfahrensschritte (vgl. die Ausführungen zum Anspruch 1):

- ein Aufrufen einer Anwendungsprogrammierschnittstelle („API“) durch eine Anwendung („Telephony Application (Dialer)“), um dementsprechend eine bestimmte Funktion, beispielsweise in Form eines Wählvorgangs durchzuführen (**Merkmal N1**),
- ein Empfangen des API-Aufrufs an einer ersten Schnittstelle („TAPI“) der Proxyschicht in Form einer „TAPI Implementation“ (vgl. die in Fig. 1.2 zwischen den Telefonieanwendungen und der „TAPI implementation“ eingezeichnete „TAPI“-Schnittstelle sowie die entsprechenden Ausführungen zum Anspruch 1 / **Merkmal N2**),
- ein Transformieren des API-Aufrufs durch die Proxyschicht („TAPI Implementation“) in einen Ein-/ Ausgabe-Steuerbefehl, der von der Treiberschicht (Telephony Service Provider“ / „TSP“) verstanden wird, wobei es sich bei dem im nebengeordneten Anspruch 7 genannten „Transformieren des API“ nach Angaben der Patentinhaberin in der mündlichen Verhandlung für den Fachmann offensichtlich um nichts anderes als ein Transformieren eines API-Aufrufs handelt (vgl. Figur 1.2 und dem zugehörigen Text auf S. 7, zweiter Abs. / **Merkmal N3**<sub>teilweise</sub>, ohne explizite Nennung eines „IOCTL“-Eingabe-/ Ausgabebefehls), wobei
- die Proxyschicht („TAPI Implementation“) der Kommunikation mit einer der vorstehend genannten Anwendungen an der ersten

Schnittstelle („TAPI“) und einer TSP-Treiberschicht („Telephony Service Provider“) an einer zweiten Schnittstelle („TSPI“) dient (**Merkmal N3.1**), wobei die Treiberschicht mit der zweiten Schnittstelle und mit der zuvor genannten Funktelefoniehardware (Telefoniefunkhardware) an einer dritten Schnittstelle zwischen der in der Figur 1.2 unten dargestellten Hardware und der TSP-Treiberschicht verbunden ist, und die dritte Schnittstelle für die Übergabe von entsprechenden Befehlen, die von der Hardware verstanden werden, an die Hardware dient (**Merkmal N3.2**).

Der Fachmann entnimmt der Druckschrift **D6** dabei ebenfalls im Zusammenhang mit dem in Figur 1.2 sowie Fig. 7.1 und dem jeweils zugehörigen Text beschriebenen Modell zur Kommunikation mit Telefoniehardware mittels einer TAPI-Schnittstelle, einer TSP-Treiberschicht und einer TSPI-Schnittstelle folgende Verfahrensschritte:

- ein Senden des Ein-/Ausgabe-Steuerbefehls an die TSP-Treiberschicht an der zweiten Schnittstelle (Telephony Service Provider Interface „TSPI“; vgl. Fig. 1.2 und Fig. 7.1 sowie den jeweils zugehörigen Text / **Merkmal N4<sub>teilweise</sub>**, ohne explizite Nennung eines „IOCTL“-Eingabe-/ Ausgabebefehls),
- ein Empfangen des gesendeten Ein-/ Ausgabe-Steuerbefehls durch die TSP-Treiberschicht an der zweiten Schnittstelle („TSPI“ / **Merkmal N5<sub>teilweise</sub>**, ohne explizite Nennung eines „IOCTL“-Eingabe-/ Ausgabebefehls),
- ein Transformieren des gesendeten Ein-/Ausgabe-Steuerbefehls in einen Befehl in Form eines AT-Commands, der durch die vorstehend genannte Hardware resp. Funkhardware verstanden wird, um eine bestimmte Funktion auszuführen, durch die TSP-Treiberschicht (vgl. Fig. 1.2 sowie Fig. 7.1 und den jeweils zugehörigen Text / **Merkmal**

**N6**teilweise, ohne explizite Nennung eines „IOCTL“-Eingabe-/Ausgabebefehls), sowie

- ein Senden des transformierten Befehls resp. Funkbefehls an die Funktelefoniehardware (Telefoniefunkhardware) an der dritten Schnittstelle, die – wie vorstehend dargelegt – zwischen der in der Fig. 1.2 dargestellten Treiberschicht („TSP“) und der darunter dargestellten Hardware angeordnet ist (**Merkmal N7**).

In der Druckschrift **D6** werden damit im Unterschied zum Anspruch 7 des Streitpatents explizit keine IOCTL-Befehle als Ein-/Ausgabe-Steuerbefehle genannt (vgl. die Ausführungen zu den Merkmalen N3 und N4 bis N6). Die im Rahmen der Kommunikation zwischen einem Anwendungsprogrammmodul und der Hardware eingesetzten IOCTL-Steuerbefehle sollen dabei gemäß Streitpatent „Input/Output-Control Codes“ bzw. Eingabe-/Ausgabe-Steuerbefehle bezeichnen, mit denen Informationen zu einem Treiber gesendet werden (vgl. Abs. [0035] der deutschen Übersetzung DE 602 14 059 T2 des Streitpatents). Bei dem der Druckschrift **D6** entnehmbaren transformierten API-Aufruf handelt es sich dabei für den Fachmann ohne weiteres erkennbar auch um Eingabe/Ausgabe-Steuerbefehle bzw. Input/Output-Control-Codes, was bezüglich der technischen Funktion äquivalent zu den – im Streitpatent nicht weiter spezifizierten – IOCTL-Befehlen ist, da sie der Steuerung der Kommunikation mittels der Telefoniehardware dienen (vgl. in **D6** beispielsweise Modem-Befehle / Hayes AT commands; S. 181 bis 182, seitenübergreifender Abs.). Dem Fachmann verbleibt somit ausgehend von der Druckschrift **D6** allenfalls, einen entsprechenden bekannten Befehlssatz zu wählen (**Merkmale N3<sub>Rest</sub>, N4<sub>Rest</sub>, N5<sub>Rest</sub> und N6<sub>Rest</sub>**). Damit kann die namentliche Nennung eines IOCTL-Steuerbefehls im nebengeordneten Anspruch 7 entgegen der Auffassung der Patentinhaberin keine erfinderische Tätigkeit gegenüber der Lehre der Druckschrift **D6** begründen. Der nebengeordnete Anspruch 7 nach Hauptantrag ist somit ebenfalls nicht patentfähig.

d) Die Patentinhaberin verteidigt ihr Patent explizit mit unabhängigen Ansprüchen 1 und 7 nach Hauptantrag sowie entsprechenden unabhängigen Ansprüchen nach den Hilfsanträgen 1 bis 9. Eine darüber hinausgehende Verteidigung des Patents im Hinblick auf einzelne abhängige Ansprüche ist vorliegend wiederum nicht ersichtlich. Mit den nicht patentfähigen Ansprüchen 1 und 7 nach Hauptantrag sind daher auch die Unteransprüche 2 bis 6 bzw. die Unteransprüche 8 bis 30 und der formal nebengeordnete, auf ein Telefongerät gerichtete Patentanspruch 31 nach Hauptantrag, der auf ein „Verfahren nach einem der Ansprüche 1 bis 30“ zurückbezogen ist, nicht schutzfähig. Weder hat sich die Patentinhaberin auf die eigenständige Patentfähigkeit der Unteransprüche berufen, noch ist diese für den Senat ersichtlich (vgl. BGH, GRUR 2007, 862 Leitsatz – „Informationsübermittlungsverfahren II“ und **BGH X ZR 109/08** 1. Leitsatz – „Sensoranordnung“).

## 2) Hilfsantrag 1

Mit der nach Hilfsantrag 1 beanspruchten Fassung der Patentansprüche kann das angegriffene Patent ebenfalls nicht erfolgreich verteidigt werden.

a) Anspruch 1 nach Hilfsantrag 1 weist gegenüber Anspruch 1 nach Hauptantrag das zusätzliche Merkmal auf, dass die Proxyschicht Hardware-unabhängig und die Treiberschicht Hardware-abhängig ist (vgl. Merkmal M1.3).

Auch dieses zusätzliche Merkmal kann keine erfinderische Tätigkeit gegenüber dem Stand der Technik gemäß Druckschrift **D6** begründen. Denn der Druckschrift **D6** ist auch bereits zu entnehmen, dass durch das in der Figur 1.2 und dem zugehörigen Text beschriebene Schichtenmodell ermöglicht wird, Anwendungen unabhängig von der Hardware und damit auch unabhängig von den zur Hardware zugehörigen Hardware-Treibern zu



entwickeln (vgl. S. 7, zweiter Abs.). Das bedeutet für den Fachmann nichts anderes, als dass die in der Figur 1.2 dargestellte Proxyschicht in Form der vorstehend zitierten TAPI-Implementierung („TAPI Implementation“), die an der Schnittstelle zur Anwendung („Telephony Application“) ausgebildet ist, Hardware-unabhängig ist, während die zur Steuerung der Hardware zugehörige Treiberschicht („TSP“) – ihrem Zweck der unmittelbaren Hardware-Steuerung durch die Bereitstellung von hardwarenahen Funktionen entsprechend – Hardware-abhängig ist (**Merkmal M1.3**). Bezüglich der weiteren Merkmale des Anspruchs 1 nach Hilfsantrag 1 wird auf die Ausführungen zum Anspruch 1 nach Hauptantrag verwiesen, die hier in gleicher Weise gelten. Das Verfahren gemäß Anspruch 1 nach Hilfsantrag 1 ergibt sich damit für den Fachmann ebenfalls in naheliegender Weise aus der Kenntnis der Druckschrift **D6**.

- b)** Anspruch 7 nach Hilfsantrag 1 weist gegenüber Anspruch 7 nach Hauptantrag ebenfalls das zusätzliche Merkmal auf, dass die Proxyschicht Hardware-unabhängig und die Treiberschicht Hardware-abhängig ist (vgl. Merkmal N2.3). Dieses zusätzliche Merkmal kann auch hier keine erfinderische Tätigkeit gegenüber dem Stand der Technik gemäß Druckschrift **D6** begründen. Denn – wie vorstehend ausgeführt – entnimmt der Fachmann der Druckschrift **D6**, dass die in der Figur 1.2 dargestellte Proxyschicht in Form der TAPI-Implementierung („TAPI Implementation“), Hardware-unabhängig ist, während die zur Steuerung der Hardware dienende zugehörige Treiberschicht („TSP“) zwangsläufig – d.h. ihrem Zweck der unmittelbaren Hardware-Steuerung entsprechend – Hardware-abhängig ist (**Merkmal N2.3**). Bezüglich der weiteren Merkmale des Anspruchs 7 nach Hilfsantrag 1, die inhaltsgleich zu den Merkmalen des Anspruchs 7 nach Hauptantrag sind, wird auf die diesbezüglichen Ausführungen zum Hauptantrag verwiesen. Auch das Verfahren gemäß Anspruch 7 nach Hilfsantrag 1 ergibt sich damit für den Fachmann in naheliegender Weise aus der Kenntnis der Druckschrift **D6**.

- c) Mit den nicht patentfähigen Ansprüchen 1 und 7 nach Hilfsantrag 1 sind auch die Unteransprüche 2 bis 6 bzw. die Unteransprüche 8 bis 30 und der formal nebengeordnete, auf ein Telefongerät gerichtete Patentanspruch 31 nach Hilfsantrag 1, der auf ein „Verfahren nach einem der Ansprüche 1 bis 30“ zurückbezogen ist, nicht schutzfähig, da diese Ansprüche nicht selbständig patentfähig verteidigt worden sind. Für den Senat ist dies auch nicht ersichtlich.

### 3) Hilfsantrag 2

Auch mit der nach Hilfsantrag 2 beanspruchten Fassung der Patentansprüche kann das angegriffene Patent nicht erfolgreich verteidigt werden.

- a) Anspruch 1 nach Hilfsantrag 2 weist gegenüber Anspruch 1 nach Hilfsantrag 1 zusätzliche Merkmale auf, die das programmtechnische Erzeugen, Senden und Konvertieren eines API-Aufrufs sowie das Übertragen und Durchführen von Befehlen inklusive einer Erwiderung mit einem Erfolgscode betreffen (Merkmale M9 bis M18).

Auch diese zusätzlichen Merkmale können keine erfinderische Tätigkeit gegenüber dem Stand der Technik gemäß Druckschrift **D6** begründen. Denn der Fachmann entnimmt der Druckschrift **D6** im Zusammenhang mit dem vorstehend in Bezug auf die Ansprüche 1 nach Haupt- und Hilfsantrag 1 genannten API-Aufruf auch (vgl. Fig. 7.1 mitsamt Beschreibung, S. 181-182 und die weiteren Zitatstellen a. a. O.)

- das Erzeugen des API-Aufrufs in einer von einer Vielzahl von Anwendungen („Telephony Application (PIM)“, „Telephony Application (Fax)“, „Telephony Application (Dialer)“), um eine bestimmte Funktion, beispielsweise in Form eines Wählvorgangs mittels „Dialer“ durchzuführen (vgl. Fig. 1.2 und den zugehörigen Text auf S. 7, sowie Kapitel 3, S. 41-46, Abschnitt „Initialization“ / **Merkmal M9**),

- das Senden des API-Aufrufs an einen Treiber der „TSP“-Treiber-schicht (vgl. Fig. 1.2 und den zugehörigen Text auf S. 7, dritter Abs., i. V. m. dem Text auf S. 13, letzter Abs., in dem die Anwendung in Bezug auf drahtlose Telefonie bzw. Mobiltelefonie („wireless, cellular“) beschreiben wird, was für den Fachmann nichts anderes als die Anwendung in Bezug auf Telefoniefunkhardware und einen entsprechenden Funktreiber bedeutet / **Merkmal M10**) und
- das Konvertieren des API-Aufrufs in einen Befehl, der von der Telefoniefunkhardware verstanden wird, an den vorstehend genannten Treiber bzw. Funktreiber (**Merkmal M11**) mitsamt einem Übertragen des Befehls an die Hardware bzw. Telefoniefunkhardware und dem Durchführen der bestimmten Funktion (z. B. Wählvorgang mittels „Dialer“) durch diese Hardware (vgl. Fig. 1.2 und den zugehörigen Text auf S. 7, dritter Abs., i. V. m. dem Text auf S. 13, letzter Abs. / **Merkmale M12 und M13**).

Dabei ist der Druckschrift **D6** ebenfalls zu entnehmen, dass in Erwiderung auf eine erfolgreiche Durchführung der bestimmten Funktion ein Senden eines Erfolgscodes an die eine der Vielzahl von Anwendungen erfolgt, die den API-Aufruf erzeugt hat, wobei in Druckschrift **D6** die „line callback function“ der Weitergabe des Ergebnisses an die aufrufende Anwendung dient (vgl. S. 43, zweiter Abs.: „[...] the TAPI window receives the message and turns it into a call to the application callback function“ i. V. m. S. 201, erster Abs.: „When the operatin has eitehr completed successfully or failed, it’s the TSP’s job to notify TAPI using the proper request ID“ i. V. m. S. 47, zweiter Abs.: „When the request has been completed, a LINE\_REPLY event will be sent to the line callback function“ und S. 43, zweiter Abs.: „[...] the TAPI window receives the message and turns it into a call to the application callback function“ / **Merkmal M14**). Druckschrift **D6** ist mit dem auf S. 47 im ersten Satz genannten „request identifier“ ebenfalls ein Identifikator zu entnehmen, mit dem der Erfolgscodes mit der einen, den zugehörigen Befehl aufrufenden Anwendung verbunden ist. Es kann dabei dahinstehen, ob ein

Identifizierer, der das API, den Befehl und den Erfolgscode mit einem Identifikator assoziiert und der sie mit der einen der Vielzahl von Anwendungen verbindet, nicht zwangsläufig in einem asynchronen System notwendig ist, bei dem ein Erfolgscode nach erfolgreicher Befehlsausführung in der Hardware an den Aufrufenden gesendet werden soll (vgl. „request identifier“, Abschnitt „TAPI Function Return“, S. 47, erster und zweiter Abs. und Abschnitt „Asynchronous Functions“, S. 201, erster und zweiter Abs. / **Merkmal M15**). Dies bedeutet für den Fachmann im Zusammenhang mit der vorstehend zitierten Treiberschicht („TSP“) und dem damit verbundenen Treiber (beispielsweise dem mit dem für Funktelefonie bzw. Mobilfunk („wireless, cellular“) verbundenen TSP-Funktreiber), dass der Funktreiber den Erfolgscode auch empfängt (**Merkmal M16**) und den Erfolgscode mit der Anwendung, die den API-Aufruf erzeugt hat, abgleicht, indem er den vorstehend genannten Identifikator verwendet (vgl. hierzu die vorherigen Ausführungen zum Erfolgscode S. 46 bis 47, Abschnitt „TAPI Function Returns“ und S. 201, Abschnitt „Asynchronous Functions“ / **Merkmal M17**). Der Fachmann entnimmt diesen Textstellen zur Durchführung asynchroner Funktionsaufrufe der Druckschrift **D6** weiter, dass die Treiber (bspw. der Funktreiber) den Erfolgscode über die Proxyschicht („TAPI Implementation“) an die Anwendung senden (vgl. einleitende Ausführungen zu Merkmal M14 hinsichtlich der „callback function“ / **Merkmal M18**). Bezüglich der weiteren Merkmale des Anspruchs 1 nach Hilfsantrag 2, die inhaltsgleich zu den Merkmalen des Anspruchs 1 nach Hilfsantrag 1 sind, wird auf die diesbezüglichen Ausführungen zum Hilfsantrag 1 verwiesen. Auch das Verfahren gemäß Anspruch 1 nach Hilfsantrag 2 ergibt sich damit für den Fachmann in naheliegender Weise aus der Kenntnis der Druckschrift **D6**.

- b)** Anspruch 7 nach Hilfsantrag 2 weist gegenüber Anspruch 7 nach Hilfsantrag 1 zusätzliche Merkmale auf, die ebenfalls das Empfangen von Kommunikationen von der Funkhardware, das Senden eines Erfolgscodes, das Erzeugen eines Identifikators und ein Zurückrufen der aufrufenden Anwendung betreffen (Merkmale N8 bis N17).

Auch diese zusätzlichen Merkmale können keine erfinderische Tätigkeit gegenüber dem Stand der Technik gemäß Druckschrift **D6** begründen. Denn der Fachmann entnimmt der Druckschrift **D6** auch folgende Schritte:

- das Empfangen von Kommunikationen von der Funkhardware durch die vorgenannte „TSP“-Treiberschicht, wobei die Kommunikationen angeben, dass die bestimmte Funktion durchgeführt wurde (vgl. S. 43, zweiter Abs.: „[...] a TSP is often called back in interrupt context, that is when an event is generated by some piece of telephony hardware [...] the TAPI window receives the message and turns it into a call to the application callback function“ / **Merkmal N8**) und
- das Senden eines Erfolgscodes („[...] positive request identifier“) an die Proxyschicht in Form der TAPI-Implementierung (vgl. den Text auf S. 47, erster Abs. i. V. m. den vorstehend genannten Zitatstellen / **Merkmal N9**).

Dabei umfasst die Architektur des Telefons – wie vorstehend zu den Ansprüchen 1 und 7 nach Hauptantrag bzw. Hilfsantrag 1 ausgeführt – eine Proxyschicht („TAPI Implementation“), eine Treiberschicht („TSP“), eine Anwendung („Telephony Application“) und eine Funkhardware („Hardware“ / „wireless, cellular“ / **Merkmal N10**), wobei die Anwendung das API, welches mit der durch die Hardware bzw. Funkhardware durchzuführenden Funktion assoziiert ist, in der Proxyschicht aufruft (**Merkmale N11 und N12**) und die zugehörigen Steuerbefehle („AT commands“), die von der vorstehend genannten Hardware verstanden werden, dieser Funktion entsprechen (vgl. den Text auf S. 7, zweiter Abs., sowie die Ausführungen zum Hilfsantrag 1, die hier in gleicher Weise gelten / **Merkmal N13**). Der Fachmann entnimmt der Druckschrift **D6** dabei auch – wie auch schon vorstehend ausgeführt – das Senden des Befehls an die Hardware bzw. Funkhardware (vgl. S. 181-182 und weitere Zitatstellen a. a. O. / **Merkmal N14**) und das

Erzeugen eines eindeutigen mit dem API assoziierten Identifikators („The request identifier uniquely identifies [...]“; vgl. S. 47, erster Abs.). Im Unterschied zu Druckschrift **D6** wird der Identifikator durch die Proxy-, nicht die Treiberschicht („TSP“) festgelegt (vgl. S. 47, erster Abs.). In welcher Schicht diese Festlegung erfolgt, ist aber technisch unerheblich, da jeder asynchrone Aufruf, der einen Erfolgscode erwartet, immer über TAPI und TSP an die Hardware geleitet wird, wobei beide Schichten den Identifikator zur Zuordnung zur Anwendung bei der Rückgabe des Erfolgscodes benötigen und damit in gleicher Weise zu dessen Erzeugung geeignet sind (**Merkmal N15**). Des Weiteren entnimmt der Fachmann der Druckschrift **D6** ebenfalls das Warten auf eine entsprechende Antwort von der Hardware resp. Funkhardware (vgl. Verwendung des „request identifier“, Abschnitt „TAPI Function Return“, S. 47, erster und zweiter Abs. und Abschnitt „Asynchronous Functions“, S. 201, erster und zweiter Abs., i. V. m. „callback function“ S. 43 und S. 47, jeweils zweiter Abs. / **Merkmal N16**), wobei bei einem Empfang der Antwort auch ein Zurückrufen der aufrufenden Anwendung mit der von dem Aufruf zurückgegebenen Antwort und dem eindeutigen Identifikator erfolgt (vgl. zu Merkmal N16 genannte Textstellen / **Merkmal N17**). Bezüglich der weiteren Merkmale des Anspruchs 7 nach Hilfsantrag 2, die inhaltsgleich zu den Merkmalen des Anspruchs 7 nach Hilfsantrag 1 sind, wird auf die entsprechenden Ausführungen zum Hilfsantrag 1 bzw. Hauptantrag verwiesen. Das Verfahren gemäß Anspruch 7 nach Hilfsantrag 2 ergibt sich damit für den Fachmann ebenfalls in naheliegender Weise aus der Kenntnis der Druckschrift **D6**.

- c) Mit den nicht patentfähigen Ansprüchen 1 und 7 nach Hilfsantrag 2 sind auch die Unteransprüche 2 bis 6 bzw. die Unteransprüche 8 bis 26 und der formal nebengeordnete, auf ein Telefongerät gerichtete Patentanspruch 27 nach Hilfsantrag 2, der auf ein „Verfahren nach einem der Ansprüche 1 bis 26“ zurückbezogen ist, nicht schutzfähig, da diese Ansprüche nicht als eigenständig patentfähig verteidigt worden sind. Für den Senat ist dies auch nicht ersichtlich.

#### 4) Hilfsantrag 3

Auch mit der nach Hilfsantrag 3 beanspruchten Fassung der Patentansprüche kann das angegriffene Patent nicht erfolgreich verteidigt werden.

- a) Die im Verfahren gemäß Anspruch 1 nach Hilfsantrag 3 gegenüber dem Anspruch 1 nach Hilfsantrag 1 hinzugefügten Merkmale können ebenfalls keine erfinderische Tätigkeit begründen. Denn der Fachmann entnimmt der Druckschrift **D6** ebenso, dass es sich beim Aufrufen eines aus einem Satz in der Abstraktionsschicht enthaltener APIs (vgl. die Ausführungen zum Merkmal M2.1 nach Hilfsantrag 1 bzw. nach Hauptantrag, die hier in gleicher Weise gelten) um ein Aufrufen aus jeder einer Vielzahl von Anwendungen handelt, da in der Druckschrift **D6** in diesem Zusammenhang verschiedene „Applications“ aufgeführt sind (vgl. Figur 1.2, „Telephony Application (PIM)“, „Telephony Application (Fax)“, „Telephony Application (Dialer)“, sowie dem zugehörigen Text auf S. 7: „Applications are programmed using TAPI [...]“ / **Merkmal M2\***). Des Weiteren ist der Druckschrift **D6** zu entnehmen, dass die Proxyschicht mittels einer TAPI-Implementierung im Zusammenhang mit einer Programmbibliothek in Form einer dynamischen Verbindungsbibliothek bzw. „DLL“ realisiert ist („TAPI32.DLL“; vgl. insbesondere Fig. 1.2 auf S. 7 mit zugehörigem Text sowie S. 182, zweiter und dritter Abs.). Dem Fachmann ist geläufig, dass wenn ein Prozess eine dynamische Verbindungsbibliothek anfordert, diese in den Speicher geladen und in den privaten Adressbereich des Prozesses projiziert wird. Jeder andere Prozess, der die Funktionalität der dynamischen Verbindungsbibliothek nutzen möchte, erhält eine Kopie der geladenen Bibliothek in seinem privaten Adressraum. Aufgrund dieser programmtechnischen Voraussetzungen ist dem Fachmann bekannt, dass für jede der aufrufenden Anwendungen, d. h. für jeden aufrufenden Prozess, eine separate Proxy-Instanz erzeugt wird, wobei die „TAPI32.DLL“ dynamisch gebunden wird und die Proxy-Instanzen die Treiberschicht in Form von „TSP“ gemeinsam nutzen. Für die einzelnen Prozesse und damit

auch für die Proxy-Instanzen und die Treiberschicht eigene private und damit verschiedene Adressräume vorzusehen, ist hierbei ein zwingendes Erfordernis, um Kollisionen beim Zugriff auf Daten im Adressraum zu vermeiden (**Merkmal M2.3\***). Hinsichtlich den Formulierungszusätzen in den **Merkmalen M4\* bis M8\***, in denen gegenüber den Formulierungen der Merkmale M4 bis M8 des Anspruchs 1 nach Hilfsantrag 1 nochmals zum Ausdruck kommt, dass es sich um eine der aufrufenden Anwendungen i. V. m. einer Proxyschicht-Instanz sowie um eine gemeinsam benutzte Treiberschicht handelt, wird auf die entsprechenden vorherigen Ausführungen verwiesen, die hier in gleicher Weise gelten. Bezüglich der weiteren Merkmale des Anspruchs 1 nach Hilfsantrag 3, die inhaltsgleich zu den Merkmalen des Anspruchs 1 nach Hilfsantrag 1 bzw. nach Hauptantrag sind, wird auf die entsprechenden Ausführungen zum Hilfsantrag 1 bzw. zum Hauptantrag verwiesen. Das Verfahren gemäß Anspruch 1 nach Hilfsantrag 3 ergibt sich damit für den Fachmann ebenfalls in naheliegender Weise aus der Kenntnis der Druckschrift **D6**.

- b)** Die im Verfahren gemäß Anspruch 7 nach Hilfsantrag 3 gegenüber dem Anspruch 7 nach Hilfsantrag 1 hinzugefügten einzelnen Merkmalsteile in den Merkmalen N1\* bis N3\*, N4\* bis N6\* und N8\*, die inhaltlich den vorstehend genannten Formulierungen im Anspruch 1 nach Hilfsantrag 3 entsprechen, können in gleicher Weise keine erfinderische Tätigkeit begründen, da diese Merkmale bereits der Druckschrift **D6** entnehmbar sind bzw. für den Fachmann aufgrund des Offenbarungsgehalts der Druckschrift **D6** und seines Fachwissens über diesbezügliche programmtechnische Maßnahmen nahegelegt sind (vgl. die Ausführungen zum Anspruch 1 nach Hilfsantrag 3, die hier in gleicher Weise gelten). Bezüglich der übrigen Merkmale des Anspruchs 7 nach Hilfsantrag 3, die inhaltsgleich zu den Merkmalen des Anspruchs 7 nach Hilfsantrag 1 sind, wird wiederum auf die entsprechenden Ausführungen zum Hilfsantrag 1 bzw. Hauptantrag verwiesen.



- c) Mit den nicht patentfähigen Ansprüchen 1 und 7 nach Hilfsantrag 3 sind auch die Unteransprüche 2 bis 6 bzw. die Unteransprüche 8 bis 26 und der formal nebengeordnete, auf ein Telefongerät gerichtete Patentanspruch 31 nach Hilfsantrag 3, der auf ein „Verfahren nach einem der Ansprüche 1 bis 30“ zurückbezogen ist, nicht schutzfähig. Die Beklagte hat diese Ansprüche auch nicht als selbständig patentfähig verteidigt. Für den Senat war dies ebenso nicht ersichtlich.

## 5) Hilfsantrag 4

Mit der nach Hilfsantrag 4 beanspruchten Fassung der Patentansprüche kann das angegriffene Patent nicht erfolgreich verteidigt werden.

- a) Die im Verfahren gemäß Anspruch 1 nach Hilfsantrag 4 gegenüber dem Anspruch 1 nach Hilfsantrag 3 hinzugefügten Merkmale können ebenfalls keine erfinderische Tätigkeit begründen. Denn der Druckschrift **D6** ist auch bereits zu entnehmen, dass das Empfangen eines API-Aufrufs im Zusammenhang mit der Programmierschnittstelle für Telefonieanwendungen („TAPI“) führt dazu, eine bestimmte Anrufsteuerfunktion (z. B. eine Wählfunktion mittels „Dialer“) auszuführen (vgl. Fig. 1.2 auf S. 7 und entsprechenden Ausführungen zum Hauptantrag und Hilfsantrag 3, die hier in gleicher Weise gelten / **Merkmal M3\*\***). Des Weiteren ist der Druckschrift **D6** auch bereits zu entnehmen, dass der per Transformation in der Proxyschicht („TAPI Implementation (TAPI32.DLL)“) generierte Befehl bzw. Standardtelefoniebefehl dem API-Aufruf entspricht, welcher dazu dient, die entsprechend bestimmte Funktion im Zusammenhang mit der vorstehend genannten Anrufsteuerfunktion auszuführen (vgl. die Ausführungen zum Hauptantrag und zum Hilfsantrag 3, die hier in gleicher Weise gelten / **Merkmale M4\*\* bis M8\*\***). Darüber hinaus sind die im Anspruch 1 nach Hilfsantrag 4 aufgeführten **Merkmale M9\*\* bis M14\*\***, die einen Erfolgscode und einen entsprechenden Identifikator betreffen, inhaltsgleich zu den im Anspruch 1 nach Hilfsantrag 2 aufgeführten Merkmalen M13 bis

M18, die der Druckschrift **D6** ebenfalls zu entnehmen sind (vgl. die diesbezüglichen Ausführungen zum Anspruch 1 nach Hilfsantrag 2 sowie weitere Zitatstellen a. a. O., die hier inhaltlich in gleicher Weise gelten). Diese im Anspruch 1 nach Hilfsantrag 4 nochmals aufgeführten Merkmale in Kombination mit den vorstehend abgehandelten Merkmalszusätzen bzgl. dem API-Aufruf i. V. m. einer bestimmten Anrufsteuerfunktion (Merkmale M4\*\* bis M8\*\*) sind damit auch bereits der Druckschrift **D6** entnehmbar, so dass sich das hier beanspruchte Verfahren für den Fachmann ebenfalls in naheliegender Weise aus der in der Druckschrift **D6** ergibt (vgl. die vorstehenden Ausführungen, die hier inhaltlich in gleicher Weise gelten).

- b) Die im Verfahren gemäß Anspruch 7 nach Hilfsantrag 4 gegenüber dem Anspruch 7 nach Hilfsantrag 3 hinzugefügten Merkmale können ebenfalls keine erfinderische Tätigkeit begründen. Denn der Druckschrift **D6** ist auch bereits zu entnehmen, dass das API in der Proxyschicht („TAPI Implementation“) aufgerufen wird, um eine Funktion aufzurufen, die von der Hardware verstanden wird, wobei es sich bei der Hardware aufgrund der in der Druckschrift **D6** aufgeführten drahtlosen Telefonie („wireless, celular“) auch um eine Funkhardware handelt (vgl. S. 13, letzter Abs. sowie die Ausführungen zum Anspruch 7 nach Hauptantrag bzw. zu den jeweiligen Ansprüchen 7 nach den Hilfsanträgen 1, 2 und 3, die hier in gleicher Weise gelten / **Merkmal N1.1\*\***). Dabei sind auch die weiteren im Anspruch 7 nach Hilfsantrag 4 aufgeführten Merkmale N2\*\* bis N9\*\* sowie die Merkmale N10\*\* bis N14\*\*, die insbesondere einen Erfolgscode und einen entsprechenden Identifikator betreffen, inhaltsgleich zu den im Anspruch 7 nach Hilfsantrag 2 aufgeführten Merkmalen, die der Druckschrift **D6** ebenfalls zu entnehmen sind (vgl. die diesbezüglichen Ausführungen zum Anspruch 7 nach Hilfsantrag 2, die hier inhaltlich in gleicher Weise gelten / **Merkmal N2\*\* bis N14\*\***). Diese im Anspruch 7 nach Hilfsantrag 4 aufgeführten Merkmale in Kombination mit den vorstehend abgehandelten Merkmalen bzgl. dem API-Aufruf i. V. m. einer Anrufsteuerfunktion sind damit wiederum der Druckschrift **D6** entnehmbar, so dass sich das mit

Anspruch 7 nach Hilfsantrag 4 beanspruchte Verfahren für den Fachmann ebenfalls in naheliegender Weise aus der in der Druckschrift **D6** ergibt.

- c) Mit den nicht patentfähigen Ansprüchen 1 und 7 nach Hilfsantrag 4 sind auch die Unteransprüche 2 bis 6 bzw. die Unteransprüche 8 bis 26 und der formal nebengeordnete, auf ein Telefongerät gerichtete Patentanspruch 27 nach Hilfsantrag 4, der auf ein „Verfahren nach einem der Ansprüche 1 bis 26“ zurückbezogen ist, nicht schutzfähig, da diese Ansprüche nicht als selbständig patentfähig verteidigt worden sind und dies für den Senat auch nicht ersichtlich war.

## 6) Hilfsantrag 5

Mit der nach Hilfsantrag 5 beanspruchten Fassung der Patentansprüche kann das angegriffene Patent ebenfalls nicht erfolgreich verteidigt werden.

- a) Das Verfahren gemäß Anspruch 1 nach Hilfsantrag 5 entspricht inhaltlich dem Anspruch 1 nach Hauptantrag unter Hinzufügung eines Zusatzes, wonach der API-Aufruf der Proxyschicht einer Hardware Abstraktionsschicht durch eine TSP- bzw. eine TAPI-Schnittstelle erfolgen soll (Merkmal M3\*\*\*). Dies bedeutet nichts anderes, als dass die Anwendung, welche die Abstraktionsschicht aufruft, selbst als ein TSP („TAPI Service Provider“ bzw. „Telephony Service Provider“) und demnach als ein Dienst für computergestützte Telefonie im Rahmen eines abstrakten Schichtenmodells mit einer TAPI-Schnittstelle definiert wird. Die Definition bzw. inhaltliche Bestimmung einer Anwendung, die mittels einer separaten, aus Druckschrift **D6** bekannten Abstraktionsschicht mit der Telefonie-Hardware kommunizieren soll (vgl. hierzu auch Druckschrift **D6**, S. 7, Fig. 1.2 und die Ausführungen zum Hauptantrag), ist im Hinblick auf die Lösung des technischen Problems bzw. der gestellten Aufgabe als nicht weiter relevant anzusehen. Denn eine solche Definition bzw. inhaltliche Bestimmung einer aufrufenden Anwendung als TSP-Dienst für

computergestützte Telefonie bestimmt oder beeinflusst nicht weiter die technische Lösung der Aufgabe, es Hardware-unabhängigen Anwendungen zu ermöglichen, mit einer Telefoniehardware zu kommunizieren (vgl. BGH, GRUR 2011, 125, 2. Leitsatz – Wiedergabe topografischer Informationen; BGH, GRUR 2013, 909, 910 ff – Fahrzeugnavigationssystem).

Den Ausführungen der Patentinhaberin, dass aus dem Aufruf der Abstraktionsschicht durch einen TSP oder eine TAPI-Schnittstelle zwangsläufig folge, dass es sich bei der anspruchsgemäßen Abstraktionsschicht nicht um eine TSP/TAPI-Hardwareabstraktionsschicht im Sinne der Druckschrift **D6** handeln könne, kann nicht beigetreten werden. Wesentlich ist dabei nicht die Bezeichnung der Schicht im gewählten Schichtenmodell, sondern allein deren Funktion – hier also der Schaffung einer Hardwareabstraktionsschicht mit Hardware-unabhängiger Anwendungsschnittstelle (Proxyschicht).

Bezüglich der weiteren Merkmale des Anspruchs 1 nach Hilfsantrag 5 wird auf die entsprechenden Ausführungen zum Hauptantrag verwiesen, die hier inhaltlich in gleicher Weise gelten. Auch die zusätzliche Definition bzw. inhaltliche Bestimmung einer Anwendung entsprechend Merkmal M3\*\*\* des Anspruchs 1 nach Hilfsantrag 5 kann damit keine erfinderische Tätigkeit gegenüber dem aus der Druckschrift **D6** bekannten Stand der Technik begründen.

- b)** Die vorstehenden Ausführungen zum Anspruch 1 nach Hilfsantrag 5 gelten in gleicher Weise in Bezug auf den nebengeordneten Anspruch 7 nach Hilfsantrag 5, in dem eine Anwendung, die das API aufruft, ebenfalls als TSP- oder TAPI-Schnittstelle definiert wird (Merkmal N1\*\*\*). Diese zusätzliche Definition bzw. inhaltliche Bestimmung einer Anwendung kann in gleicher Weise keine erfinderische Tätigkeit gegenüber dem aus der Druckschrift **D6** bekannten Stand der Technik begründen (zur Vermeidung

von Wiederholungen wird auf die Ausführungen zum Anspruch 1 nach Hilfsantrag 5 verwiesen).

- c) Mit den nicht patentfähigen Ansprüchen 1 und 7 nach Hilfsantrag 5 sind auch die Unteransprüche 2 bis 6 bzw. die Unteransprüche 8 bis 30 und der formal nebengeordnete, auf ein Telefongerät gerichtete Patentanspruch 30 nach Hilfsantrag 5, der auf ein „Verfahren nach einem der Ansprüche 1 bis 30“ zurückbezogen ist, nicht schutzfähig. Die Patentinhaberin hat diese Ansprüche auch nicht als selbständig patentfähig verteidigt. Dies war für den Senat auch nicht ersichtlich.

## **7) Hilfsantrag 6**

Auch mit der nach Hilfsantrag 6 beanspruchten Fassung der Patentansprüche kann das angegriffene Patent nicht erfolgreich verteidigt werden.

- a) Das Verfahren gemäß Anspruch 1 nach Hilfsantrag 6 entspricht inhaltlich dem Anspruch 1 nach Hilfsantrag 1 unter Hinzufügung eines Zusatzes, wonach der API-Aufruf von einer Anwendung in Form eines TSP oder einer TAPI-Schnittstelle empfangen werden soll (vgl. Merkmal M3\*\*\*). Auch hier bedeutet das nichts anderes, als dass eine Anwendung, die das API aufruft, einen TSP-Dienst („Telephony Service Provider“) und somit einen Dienst für computergestützte Telefonie im Rahmen eines abstrakten Schichtenmodells darstellen soll. Wie bereits zum Anspruch 1 nach Hilfsantrag 5 dargelegt, ist es in Bezug auf die technische Lehre zur Lösung der gestellten Aufgabe als nicht weiter relevant anzusehen, wie eine Hardware-unabhängige Anwendung, die mit der Hardware kommuniziert, definiert bzw. inhaltlich bestimmt sind, so dass dies hier ebenfalls keine erfinderische Tätigkeit begründen kann (vgl. die Ausführungen zum Anspruch 1 nach Hilfsantrag 5, die hier in gleicher Weise gelten). Für die weiteren, unverändert aus Ansprüchen 1 nach Hilfsantrag 1 übernommenen Merkmale gelten die vorstehenden Ausführungen zum Hilfsantrag 1.

- b)** Die vorstehenden Ausführungen zum Anspruch 1 nach Hilfsantrag 6 gelten in gleicher Weise in Bezug auf den nebengeordneten Anspruch 7 nach Hilfsantrag 6, in dem ebenfalls zusätzlich aufgeführt ist, dass eine Anwendung, die das API aufruft, ein TSP bzw. eine TAPI-Schnittstelle darstellen soll (vgl. wiederum Merkmal N1\*\*\*). Auch hier kann die Definition bzw. inhaltliche Bestimmung einer Anwendung als TSP-Dienst für computergestützte Telefonie mit einer TAPI-Schnittstelle keine erfinderische Tätigkeit gegenüber dem aus der Druckschrift **D6** bekannten Stand der Technik begründen (vgl. die vorstehenden Ausführungen, die hier in gleicher Weise gelten).
- c)** Mit den nicht patentfähigen Ansprüchen 1 und 7 nach Hilfsantrag 6 sind auch die Unteransprüche 2 bis 6 bzw. die Unteransprüche 8 bis 30 und der formal nebengeordnete, auf ein Telefongerät gerichtete Patentanspruch 30 nach Hilfsantrag 6, der auf ein „Verfahren nach einem der Ansprüche 1 bis 30“ zurückbezogen ist, nicht schutzfähig. Die Patentinhaberin hat sich zudem nicht darauf berufen, dass diesen Unteransprüchen ein eigenständig patentfähiger Gehalt innewohnt. Dies war für den Senat auch nicht ersichtlich.

## **8) Hilfsanträge 7 bis 9**

Mit den nach den Hilfsanträgen 7 bis 9 beanspruchten Fassungen der Patentansprüche kann das angegriffene Patent ebenfalls nicht erfolgreich verteidigt werden.

- a)** Die jeweiligen Verfahren gemäß den Ansprüchen 1 nach Hilfsantrag 7 bzw. nach den Hilfsanträgen 8 und 9 entsprechen inhaltlich den jeweiligen Verfahren gemäß den Ansprüchen 1 nach Hilfsantrag 2 bzw. nach Hilfsantrag 3 bzw. nach Hilfsantrag 4 unter jeweiliger Ergänzung des Merkmals, dass die eine aufrufende Anwendung ein TSP bzw. eine TAPI-Schnittstelle darstellen soll (ebenfalls Merkmal M3\*\*\*). Wie bereits

vorstehend zu den Hilfsanträgen 5 und 6 dargelegt, bedeutet das nichts anderes, als dass eine Anwendung, die das API aufruft, wiederum als TSP-Dienst bzw. als entsprechende TAPI-Schnittstelle und damit inhaltlich als Dienst für computergestützte Telefonie definiert wird. Welche Hardware-unabhängige Anwendung dabei über die Abstraktionsschicht – die entsprechend der Lehre der Druckschrift **D6** bereits eine TSP-Schicht als Treiberschicht und eine TAPI-Implementierung als Proxyschicht mitsamt zugehöriger Schnittstelle umfasst – mit der Hardware kommuniziert, ist auch hier im Hinblick auf die Lösung des technischen Problems bzw. die gestellte Aufgabe als nicht weiter relevant anzusehen. Denn die Definition einer Anwendung als TSP-Dienst für computergestützte Telefonie bestimmt oder beeinflusst nicht weiter die Lösung der Aufgabe, es Hardware-unabhängigen Anwendungen zu ermöglichen, mit einer Telefoniehardware zu kommunizieren. Bezüglich der weiteren Merkmale der jeweiligen Ansprüche 1 nach den Hilfsanträgen 7 bis 9 wird auf die entsprechenden Ausführungen zu den Hilfsanträgen 2 bis 4 verwiesen, die hier in gleicher Weise gelten. Die zusätzliche Definition bzw. inhaltliche Bestimmung einer Anwendung als Dienst für computergestützte Telefonie kann hier somit auch hier keine erfinderische Tätigkeit gegenüber dem aus der Druckschrift **D6** bekannten Stand der Technik begründen.

- b) Dies gilt ebenso bezüglich der jeweiligen nebengeordneten Ansprüche 7 nach Hilfsantrag 7 bzw. nach Hilfsantrag 8 und nach Hilfsantrag 9, in denen ebenfalls ergänzt ist, dass die eine aufrufende Anwendung ein TSP oder eine TAPI-Schnittstelle darstellt. Auch hier kann eine solche Ergänzung keine erfinderische Tätigkeit gegenüber der Lehre der Druckschrift **D6** begründen, da eine solche Definition einer Anwendung die Lösung des technischen Problems bzw. der Aufgabe nicht weiter bestimmt oder beeinflusst (vgl. die entsprechenden Ausführungen zu den jeweiligen Ansprüchen 1 nach den Hilfsanträgen 7 bis 9, die hier in gleicher Weise gelten).

- c) Mit den nicht patentfähigen Ansprüchen 1 und 7 nach den Hilfsanträgen 7 bis 9 sind auch die darauf rückbezogenen Unteransprüche und die formal nebengeordneten, auf ein Telefongerät gerichteten Patentansprüche 21 (Hilfsantrag 7) bzw. 31 (Hilfsanträge 8) bzw. 27 (Hilfsantrag 9) nicht schutzfähig, da auf diese Ansprüche kein eigenständiges Patentbegehren gerichtet war.

### III.

Die Kostenentscheidung beruht auf § 84 Abs. 2 PatG i. V. m. § 91 Abs. 1 Satz 1 ZPO. Die Entscheidung über die vorläufige Vollstreckbarkeit folgt aus § 99 Abs. 1 PatG, § 709 Satz 1 und 2 ZPO.

### IV.

Gegen dieses Urteil kann das Rechtsmittel der Berufung gemäß § 110 PatG eingelegt werden.

Die Berufung ist innerhalb eines Monats nach Zustellung des in vollständiger Form abgefassten Urteils – spätestens nach Ablauf von fünf Monaten nach Verkündung – durch einen in der Bundesrepublik Deutschland zugelassenen Rechtsanwalt oder Patentanwalt schriftlich zum Bundesgerichtshof, Herrenstraße 45a, 76133 Karlsruhe, einzulegen.

Die Berufungsschrift muss

- die Bezeichnung des Urteils, gegen das die Berufung gerichtet ist, sowie
- die Erklärung, dass gegen dieses Urteil Berufung eingelegt werde,

enthalten. Mit der Berufungsschrift soll eine Ausfertigung oder beglaubigte Abschrift des angefochtenen Urteils vorgelegt werden.



Auf die Möglichkeit, die Berufung nach § 125a PatG in Verbindung mit § 2 der Verordnung über den elektronischen Rechtsverkehr beim Bundesgerichtshof und Bundespatentgericht (BGH/BPatGERVV) auf elektronischem Weg zum Bundesgerichtshof einzulegen, wird hingewiesen (s. [www.bundesgerichtshof.de/erv.html](http://www.bundesgerichtshof.de/erv.html))

Sredl

Merzbach

Dr. Schwengelbeck

Dr. Forkel

Altvater

Pr