



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 698 37 545 T2 2007.12.20**

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 004 068 B1**

(51) Int Cl.<sup>8</sup>: **G06F 1/00 (2006.01)**

(21) Deutsches Aktenzeichen: **698 37 545.9**

(86) PCT-Aktenzeichen: **PCT/US98/03623**

(96) Europäisches Aktenzeichen: **98 907 612.0**

(87) PCT-Veröffentlichungs-Nr.: **WO 1998/037481**

(86) PCT-Anmeldetag: **25.02.1998**

(87) Veröffentlichungstag

der PCT-Anmeldung: **27.08.1998**

(97) Erstveröffentlichung durch das EPA: **31.05.2000**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **11.04.2007**

(47) Veröffentlichungstag im Patentblatt: **20.12.2007**

(30) Unionspriorität:

**805804 25.02.1997 US**

(84) Benannte Vertragsstaaten:

**AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LI, LU, MC, NL, PT, SE**

(73) Patentinhaber:

**Intertrust Technologies Corp., Sunnyvale, Calif., US**

(72) Erfinder:

**HALL, Edwin J., San Jose, CA 95129, US; SHEAR, Victor H., Bethesda, MD 20814, US; TOMASELLO, Luke S., San Jose, CA 95123, US; VAN WIE, David M., Sunnyvale, CA 94086, US; WEBER, Robert P., Menlo Park, CA 94025, US; WORSENCROFT, Kim, Los Gatos, CA 95032, US; XU, Xuejun, Fremont, CA 94555, US**

(74) Vertreter:

**Schaumburg, Thoenes, Thurn, Landskron, 81679 München**

(54) Bezeichnung: **TECHNIKEN ZUR VERWENDUNG VON BESCHREIBUNGSHALTENDEN DATENSTRUKTUREN**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

## Beschreibung

**[0001]** Die vorliegende Erfindung betrifft Techniken zur Verwendung von deskriptiven Datenstrukturen.

**[0002]** Immer mehr Menschen verwenden sichere digitale Container zum sicheren und gesicherten Speichern und Transportieren von digitalem Inhalt. Ein sicheres digitales Containermodell ist der von Intertrust Technologies Corp. in Sunnyvale, Kalifornien, entwickelte Container "DigiBox™". Die US-Patente US 7095854 und WO97/43761 im Namen von Ginter et al. beschreiben Eigenschaften dieses DigiBox™-Containermodells – ein leistungsstarkes, flexibles und allgemeines Konstrukt, das geschützte, effiziente und interoperable elektronische Beschreibung und Regulierung von Beziehungen des elektronischen Kommerzes aller Arten ermöglicht, das die Schnittstelle für gesicherten Transport, Speicherung und Rechteverwaltung mit Objekten und digitalen Informationen in solchen Containern umfasst.

**[0003]** Kurzgefasst sind DigiBox-Container manipulationssichere digitale Container, mit denen man beliebige Arten von digitalen Informationen verpacken kann, wie zum Beispiel Text, Graphik, ausführbare Software, Audio und/oder Video. Die Rechteverwaltungsumgebung, in der DigiBox™-Container benutzt werden, ermöglicht es Kommerz-Teilnehmern, Regeln mit den digitalen Informationen (Inhalt) zu assoziieren. Die Rechteverwaltungsumgebung ermöglicht außerdem, Regeln (wobei hier Regeln und Parameterdatenkontrollen mit eingeschlossen sind) sicher mit anderen Rechteverwaltungsinformationen zu assoziieren, wie zum Beispiel Regeln, während der Benutzung der digitalen Informationen erzeugte Audit-Aufzeichnungen und mit der Aufrechterhaltung der ordnungsgemäßen Funktionsweise der Umgebung assoziierte administrative Informationen, einschließlich der Sicherstellung von Rechten und etwaiger Vereinbarungen zwischen Teilnehmern. Mit dem elektronischen DigiBox™-Container kann man digitale Informationen, diesbezügliche Regeln und andere Rechteverwaltungsinformationen sowie andere Objekte und/oder Daten in einer verteilten Rechteverwaltungsumgebung speichern, transportieren und eine Rechteverwaltungsschnittstelle zu ihnen bereitstellen. Mit dieser Anordnung kann man eine elektronisch durchgesetzte Kette der Abwicklung und Kontrolle bereitstellen, wobei die Rechteverwaltung weiter besteht, während ein Container von einer Entität zu einer anderen verlagert wird. Diese Fähigkeit hilft bei der Unterstützung einer Architektur der digitalen Rechteverwaltung, die es Inhaltsrechteinhabern (darunter etwaige Teilnehmer, die systemautorisierte Interessen in Bezug auf solchen Inhalt haben, wie zum Beispiel Inhaltsweiterveröffentlicher oder sogar Regierungsbehörden) ermöglicht, Inhalt, Ereignisse, Transaktionen, Regeln und Benutzungskonsequenzen, einschließlich einer etwaigen erforderlichen Be-

zahlung und/oder Benutzungsmeldung, sicher zu kontrollieren und zu verwalten. Diese sichere Kontrolle und Verwaltung setzt sich dauerhaft fort und schützt Rechte, während Inhalt an Verfasser, Distributoren, Zweckwiederverwerter, Konsumenten, Zahlungs-Disagregatoren und andere Wertkettenteilnehmer abgeliefert, von ihnen benutzt und zwischen ihnen übergeben wird.

**[0004]** Zum Beispiel kann ein Verfasser von Inhalt ein oder mehrere Elemente digitaler Informationen in einem sicheren DigiBox-Container mit einer Menge von Regeln verpacken – wobei sich solche Regeln variabel in einem oder mehreren Containern und/oder Client-Kontrollknoten befinden können – und den Container zu einem Distributor senden. Der Distributor kann die Regeln in dem Container innerhalb der vom Verfasser erlaubten Parameter ergänzen und/oder modifizieren. Der Distributor kann dann den Container durch eine beliebige erlaubte (oder nicht verbotene) Regel verteilen – indem er ihn zum Beispiel über ein elektronisches Netzwerk wie etwa das Internet übermittelt. Ein Konsument kann den Container herunterladen und den Inhalt gemäß den Regeln in dem Container benutzen. Auf dem lokalen Computer oder einem anderen InterTrust-bewussten Gerät wird der Container geöffnet und die Regeln werden durchgesetzt (durch Software, die InterTrust als einen InterTrust-Kommerz-Knoten bezeichnet). Der Konsument kann den Container (oder eine Kopie davon) zu anderen Konsumenten weiterleiten, die (wenn es die Regeln erlauben) den Inhalt gemäß denselben, unterschiedlichen oder anderen enthaltenen Regeln benutzen können – welche Regeln gelten, wird dabei durch benutzerverfügbare Rechte bestimmt, wie zum Beispiel die spezifische Identifikation des Benutzers, einschließlich einer etwaigen Klassenzugehörigkeit bzw. etwaiger Klassenzugehörigkeiten (z.B. ein Automobil-Club oder Anstellung bei einer bestimmten Universität). Gemäß solchen Regeln kann der Knoten Benutzungs- und/oder Zahlungsinformationen sammeln und zur Zahlungsbegleichung zu einem oder mehreren Clearinghäusern senden und denjenigen mit Rechten an ihrem Empfang Benutzungsinformationen übermitteln.

**[0005]** Das oben und in der Patentbeschreibung von Ginter et al. (zusammen mit ähnlichen anderen Modellen von DigiBox/VDE (Virtual Distribution Environment)) beschriebene Knoten- und Containermodell besitzt nahezu unbegrenzte Flexibilität. Es kann auf viele verschiedene Umstände und spezifische Implementierungen angewandt werden. Wenn man zum Beispiel [Fig. 1A](#) und [Fig. 1B](#) betrachtet, kann ein Zeitungsverleger eine Zeitung **102** in einem Container **100A** verteilen. Ein Verleger von Modejournalen **106** kann die Modejournale in einem anderen Container **100C** verteilen. Ähnlich kann zum Beispiel eine Großbankgeschäftsumgebung noch einen weiteren Container und ein elektronisches Handelssystem noch

einen anderen Container verwenden und so weiter.

**[0006]** Das DigiBox-Containermodell von InterTrust erlaubt und ermöglicht diese und andere verschiedene Container-Verwendungszwecke. Es ermöglicht eine detaillierte Container-Anpassung für verschiedene Verwendungszwecke, Benutzungsklassen und/oder Benutzer um verschiedenen Anforderungen und Geschäftsmodellen zu genügen. Diese Möglichkeit der Anpassbarkeit ist sehr wichtig, insbesondere bei Verwendung in Verbindung mit einer verteilten Vielzweck-Rechteverwaltungsumgebung wie zum Beispiel der in Ginter et al. beschriebenen. Eine solche Umgebung verlangt eine praktische Optimierung der Anpassbarkeit, einschließlich Anpassbarkeit und Transparenz für Containermodelle. Diese Anpassungsflexibilität hat mehrere Vorteile, wie zum Beispiel die Ermöglichung der Optimierung (z.B. maximale Effizienz, minimales Overhead) des detaillierten Containerentwurfs für jede konkrete Anwendung oder jeden konkreten Umstand, um es so zu erlauben, daß viele verschiedene Containerentwürfe für viele verschiedene Zwecke (z.B. Unternehmensmodelle) gleichzeitig existieren und von dem Rechtekontroll-Client (Knoten) auf einem elektronischen Benutzergerät, wie zum Beispiel einem Computer oder einem Unterhaltungsgerät, benutzt werden.

**[0007]** Obwohl die Unterstützung eines hohen Grades an Flexibilität große Vorteile hat, kann sie Schwierigkeiten für den durchschnittlichen Benutzer produzieren. Man bedenke zum Beispiel den Vorgang des Kreierens eines Gemäldes. Ein hervorragender Maler kreiert ein Gemälde aus einer leeren Leinwand. Da die Leinwand zu Anfang leer war, war der Maler völlig uneingeschränkt. Das Gemälde hätte eine Landschaft, ein Portrait, eine Seelandschaft oder ein beliebiges anderes Bild sein können – die Grenze ist nur die Vorstellungskraft des Malers. Diese Flexibilität erlaubt es einem hervorragenden Maler, ein Meisterwerk wie etwa die "Mona Lisa" zu kreieren. Es ist jedoch großes Können erforderlich, um ausgehend von einer leeren Leinwand ein ansprechendes Bild zu kreieren. Folglich kann man von einem unerfahrenen Maler nicht erwarten, daß er ein gutes Gemälde kreiert, wenn er von einer leeren Leinwand ausgeht.

**[0008]** Man betrachte nun einen gerade erst anfangenden Amateur-Maler. Diese Person verfügt nicht über das Können, eine leere Leinwand in ein ansprechendes Bild zu transformieren. Anstatt Jahre damit zu verbringen, zu versuchen, dieses Können zu erwerben, kann der Amateur einfach hergehen und ein Mal-Kit des Typs "Malen nach Zahlen" kaufen. Statt eine leere Leinwand zu verwenden, beginnt der Amateur-Maler mit einer vorbedruckten Leinwand, die das zu malende Bild definiert. Durch Befolgung von Anweisungen ("alle mit "12" bezeichneten Bereiche sollen dunkelrot gemalt werden", "alle mit "26" bezeich-

neten Bereiche sollen hellblau gemalt werden"), kann der Amateur mit relativ wenig Können ein Bild malen, das für das Auge relativ ansprechend ist. Hierzu muss der Amateur die vorgedruckten Anweisungen auf der Leinwand streng befolgen. Jegliche Abweichungen könnten bewirken, dass das Bild letztendlich schlecht herauskommt.

**[0009]** Probleme mit der leichten Benutzung auf dem Computergebiet können mit der Situation des "Malens nach Zahlen" verglichen werden. Wenn es wichtig ist, dass untrainierte und/oder unerfahrene Benutzer bestimmte Software benutzen, können die Systementwickler bestimmte Konstrukte vordefinieren und diese in das System hineinkonstruieren. Diese Technik erlaubt es unerfahrenen Benutzern, potentiell sehr komplizierte Entwürfe zu benutzen, ohne diese vollständig verstehen zu müssen. Die durch Verwendung des Programms verfügbare Funktionalität und Flexibilität wird dadurch normalerweise aber strikt definiert, d.h. stark eingeschränkt. Kreative Lösungen für Probleme sind folglich eingeschränkt, um praktischen Wert bereitzustellen. Außerdem kann sogar der erfahrene Benutzer aus der Verwendung zuvor implementierter Entwürfe großen Nutzen ziehen. Nur weil ein Benutzer ein komplexes Programm programmieren kann, heißt es zum Beispiel nicht, dass es angebracht oder effizient ist, ein Programm für einen spezifischen Zweck zu erstellen, auch wenn das zuvor implementierte Programm nicht ideal ist. Wenn die Erstellung eines neuen Programms mehr "kostet", d.h. zu viel Zeit oder finanzielle Ressourcen in Anspruch nimmt, wird der erfahrene Benutzer normalerweise ein zuvor implementiertes Programm benutzen, wenn es verfügbar ist. Der größte zu realisierende Gesamtwert in Bezug auf Anpassung besteht deshalb darin, in der Lage zu sein, mit großer Leichtigkeit und Effizienz anzupassen, so dass die Kosten der Anpassung den Nutzen nicht übersteigen.

**[0010]** Einheitlichkeit, Flexibilität, Kompatibilität und Interoperabilität sind weitere Gesichtspunkte, die auf dem Computergebiet insbesondere in Bezug auf Anpassung unterstützender Systeme ins Spiel kommen. In der Situation des Gemäldes kann das menschliche Auge Einzigartigkeit anerkennen, und die "einzigartige" Beschaffenheit eines Meisterwerks, wie zum Beispiel der Mona Lisa, ist ein großer Teil dessen, wodurch ein Gemälde so wertvoll wird. Im Gegensatz dazu ist es häufig wünschenswert, zumindest das Gesamt-Layout und Format von Dingen im Computergebiet zu uniformisieren. Es ist viel effizienter, wenn ein Computer im voraus weiß, wie Objekte zu behandeln und zu benutzen sind. Wenn der Computer zum Beispiel nicht im voraus weiß, wie ein Eingangsobjekt zu lesen oder zu behandeln ist, sagt man, dass der Computer und das Objekt "inkompatibel" sind, d.h. sie können nicht miteinander arbeiten. Man sagt, dass Computer "interoperabel" sind, wenn sie miteinander arbeiten können.

**[0011]** Inkompatibilitäts- und Interoperabilitätsprobleme können verhindern, dass ein Computer mit einem anderen Computer spricht und können einen davon abhalten, von jemand anderem erstellte Computerdaten zu benutzen.

**[0012]** Zum Beispiel könnte es außerhalb der Computerwelt ein Franzose mit wenig Kenntnissen von Englisch als zweite Sprache wesentlich sinnvoller und effizienter finden, ein komplexes Problem in seiner Muttersprache Französisch zu beschreiben. Wenn er jedoch mit einer zweiten Person, einem Engländer, spricht und der Engländer kein Französisch versteht, sind die beiden in Französisch nicht interoperabel und der Franzose muss auf die wesentlich weniger effiziente Option zurückgreifen, Englisch mit dem Engländer zu reden. Dies ist natürlich wesentlich besser, als wenn er versuchen würde, mit einem Deutschen zu sprechen, der weder Englisch noch Französisch versteht. Die beiden wären dann in Bezug auf die Besprechung des Problems nicht "interoperabel".

**[0013]** Da Rechteverwaltungs-Container potentiell von einer großen Anzahl verschiedener Benutzer, Gruppen und Organisationen ausgetauscht und für eine große Anzahl verschiedener Zwecke benutzt werden können, ist es ähnlich sehr wichtig, Kompatibilität und Interoperabilität bereitzustellen, wenn diese verschiedenen Teilnehmer, die jeweils an einem oder mehreren verschiedenen Rechteverwaltungsmodellen teilnehmen, effizient interoperieren sollen. Wenn zum Beispiel ein Rechteverwaltungs-Container zum Verteilen eines Rundbriefs verwendet wird und für diesen Zweck optimiert ist, muss jeder Leser des Rundbriefs ein Computersystem oder Software besitzen, das bzw. die "weiß", wie der Container und der darin enthaltene Rundbrief zu lesen sind. Da Kommerz, wie zum Beispiel das Verteilen von Rundbriefen, so effizient und kosteneffektiv wie machbar sein muss, ist es wichtig, Rechteverwaltungs-Container zu optimieren, d.h. anzupassen, um optimal die Anforderungen ihrer Modelle widerzuspiegeln und unnötige Merkmale für jede jeweilige Anwendung oder Klasse von Anwendung zu vermeiden, da unnötige Merkmale unnötiges Datenverarbeitungs-Overhead und/oder unnötigen Speicherplatz erfordern werden.

**[0014]** Verschiedene Rundbrief-Verleger können verschiedene Containerformate benutzen, die an ihrer eigenen konkreten Rundbriefe und/oder Inhaltstypen und/oder Formate angepasst sind. Ein Rundbriefleser, der an vielen verschiedenen Rundbriefen interessiert ist, muss möglicherweise in der Lage sein, um eine große Anzahl verschiedener Formate zu lesen. Es wird normalerweise nicht effizient (oder aufgrund von Sicherheitsproblemen nicht angemessen) sein, einfach bei der Lieferung die verschiedenen Container zu analysieren und zu "versuchen"

das konkrete verwendete Format "herauszufinden" oder anderweitig auszumachen.

**[0015]** Veröffentlichte Standards können dabei helfen, einen Grad der Interoperabilität und Standards für gegebene Arten von Anwendungen zu erzielen, es dauert aber im Allgemeinen lange, bis ein bestimmter Standard industrieweite Akzeptanz erreicht, und Standards werden zwischen Kategorien von Anwendungen stark variieren müssen. Darüber hinaus sind Datenstruktur- und andere Standards häufig für den kleinsten gemeinsamen Nenner ausgelegt – das heißt, sie tragen Felder und Anforderungen, die von einigen nicht benötigt werden, und es fehlen in ihnen andere Merkmale, die in bestimmten Fällen optimal sind. Es wird immer Anwendungen geben, die nicht in Bezug auf Effizienz und/oder Betrieb optimiert werden können, wenn unter Zwang, einen spezifischen Standard zu benutzen.

**[0016]** Kompromisse zwischen Flexibilität, Leichtigkeit der Benutzung und Inkompatibilität und Interoperabilität können weiter kompliziert werden, wenn Sicherheitsbetrachtungen ins Spiel kommen. Um in vielen elektronischen Kommerzanwendungen effektiv zu sein, sollten Entwürfe von elektronischen Containern manipulationssicher und gesichert sein. Man muß annehmen, daß jegliche weithin zur Erstellung und/oder Benutzung von Containern verfügbare Werkzeuge in die Hände derer fallen werden, die versuchen, die Container zu zerbrechen oder aufzuschlagen oder anderweitig digitale Informationen ohne Autorisierung zu benutzen. Die Container-Erstellungs- und -benutzungswerkzeuge müssen deshalb selbst in dem Sinne gesichert sein, daß sie bestimmte Einzelheiten über den Containerentwurf schützen müssen. Durch diese zusätzliche Sicherheitsanforderung kann es sogar noch schwieriger werden, Container leicht benutzbar zu machen und Interoperabilität bereitzustellen.

**[0017]** Die oben zitierte Patentschrift von Ginter et al. beschreibt als nicht erschöpfendes Beispiel "Vorlagen", die als eine Menge (oder Ansammlung von Mengen) von Kontrollanweisungen und/oder Daten für Objektkontrollsoftware wirken können. Siehe zum Beispiel Besprechungen der Verfahren "Objekterstellung und anfängliche Kontrollstrukturen", "Vorlagen und Klassen" und "Objektdefinitionsdatei", "Informationen" und "Inhalt" in der Schrift von Ginter et al.. Die beschriebenen Vorlagen sind mindestens in bestimmten Beispielen dazu fähig, Objekte in einem Prozess, der mit Benutzeranweisungen interagiert und Inhalt zur Erstellung eines Objekts bereitgestellt hat, zu erstellen (und/oder zu modifizieren). Ginter et al. beschreiben, dass Vorlagen zum Beispiel als Textdateien repräsentiert werden können, die spezifische Strukturen und/oder Komponentenbaugruppen definieren, und dass solche Vorlagen – mit ihren Strukturen und/oder Komponentenbaugruppen – als Objekt-

verfassungs- und/oder Objektkontrollanwendungen dienen können. Ginter et al. erwähnen, dass Vorlagen dabei helfen können, die im Kontext spezifischer Industrien und/oder Unternehmen und/oder Anwendungen naturgemäßen flexiblen und konfigurierbaren Fähigkeiten zu fokussieren, indem ein Rahmen für Betrieb und/oder Struktur bereitgestellt wird, um es existierenden Industrien und/oder Anwendungen und/oder Unternehmen zu erlauben, vertraute Konzepte in Bezug auf Inhaltstypen, Verteilungsansätze, Preisgebungsmechanismen, Benutzerinteraktionen mit Inhalt und/oder diesbezüglichen administrativen Aktivitäten, Budgets und dergleichen zu manipulieren. Dies ist nützlich bei der Suche nach optimierten Unternehmensmodellen und Wertketten, wodurch das richtige Gleichgewicht zwischen Effizienz, Transparenz, Produktivität usw. bereitgestellt wird.

**[0018]** Die vorliegende Erfindung erweitert diese Technologie, indem sie neben anderen Merkmalen eine deskriptive Datenstruktur bereitstellt.

**[0019]** Aus EP 0398645 ist ein System zur Kontrolle von Zugangsprivilegien zu Daten bekannt, bei dem über mehrere Objekte in einer objektorientierten Datenbank hinweg eine Zugangskontrollrichtlinie zugewiesen wird. Der Zugang zu der Datenbank wird auf der Basis der Zuweisung und eines Berechtigungsnachweises eines den Zugang anfordernden Benutzers kontrolliert.

**[0020]** Die vorliegende Erfindung wird in den unabhängigen Ansprüchen definiert, auf die der Leser nun verwiesen wird. Bevorzugte Merkmale der Erfindung werden in den abhängigen Ansprüchen angeführt.

**[0021]** In einem Beispiel kann eine maschinenlesbare deskriptive Datenstruktur eine abgekürzte abstrakte Repräsentation des Formats der Daten in einer die Rechteverwaltung betreffenden Datenstruktur umfassen. Mit der abstrakten Datenrepräsentation kann man eine einzelne Rechteverwaltungsdatenstruktur beschreiben, oder sie kann für eine Familie von Datenstrukturen generisch sein, die dem Format und/oder anderen Eigenschaften, die die abstrakte Repräsentation definiert, folgen. Mit der abstrakten Repräsentation kann man Rechteverwaltungsdatenstrukturen erstellen, anderen (darunter "andere" Rechteverwaltungsknoten automatisch) erlauben, solche Datenstrukturen zu lesen und zu verstehen, und bestimmte oder alle der Datenstrukturen manipulieren.

**[0022]** Die deskriptive Datenstruktur kann als "Vorlage" verwendet werden, um dabei zu helfen, Rechteverwaltungsdatenstrukturen zu erstellen und anderen Knoten zu beschreiben, einschließlich der Verwendung zur Hilfe beim Verständnis und bei der Manipulation solcher Rechteverwaltungsdatenstrukturen.

**[0023]** Bei einer besonders vorteilhaften Anordnung kann die maschinenlesbare deskriptive Datenstruktur mit einer oder einer Familie entsprechender Rechteverwaltungsdatenstrukturen assoziiert sein – und kann somit von jeglicher spezifischer bestimmter Rechteverwaltungsdatenstruktur-Benutzung unabhängig sein. Zum Beispiel kann man mit solchen Datenstrukturen eine Kopie der deskriptiven Datenstruktur behalten. Als Alternative kann man die gesamte deskriptive Datenstruktur oder einen Teil davon von einer anderen Stelle (z.B. einem Clearinghaus oder Repository) erhalten und unabhängig auf Bedarf liefern.

**[0024]** Gemäß einem Beispiel stellt die maschinenlesbare deskriptive Datenstruktur eine Beschreibung bereit, die entsprechende Struktur(en) in der Rechteverwaltungsdatenstruktur widerspiegelt und/oder definiert. Zum Beispiel kann die deskriptive Datenstruktur eine rekursive hierarchische Liste bereitstellen, die eine entsprechende rekursive hierarchische Struktur in der Rechteverwaltungsdatenstruktur widerspiegelt und/oder definiert. Bei anderen Beispielen können die durch die deskriptive Datenstruktur bereitgestellte Beschreibung bzw. Beschreibungen komplexen mehrdimensionalen Datenstrukturen mit 2, 3 oder n Dimensionen entsprechen. Die deskriptive Datenstruktur kann direkt und/oder indirekt spezifizieren, wo in einer assoziierten Rechteverwaltungsdatenstruktur entsprechende definierte Datentypen gefunden werden können. Die deskriptive Datenstruktur kann ferner Metadaten bereitstellen, die ein oder mehrere Attribute der entsprechenden Rechteverwaltungsdaten und/oder der Prozesse, mit denen sie erstellt und/oder benutzt werden, beschreiben. In einem Beispiel könnte die gesamte deskriptive Datenstruktur als solche Metadaten umfassend angesehen werden.

**[0025]** Die maschinenlesbare deskriptive Datenstruktur kann abhängig von der konkreten Anwendung ganz oder teilweise geschützt werden oder nicht. Bestimmte maschinenlesbare deskriptive Datenstrukturen können ganz oder teilweise verschlüsselt werden, während andere in "klarer" Form gehalten werden können, damit sie leicht zugänglich sind. Bestimmte maschinenlesbare Beschreibungsdatenstrukturen können, verschlüsselt oder nicht, ganz oder teilweise in Bezug auf Integrität durch einen kryptographischen Hash-Algorithmus in Kombination mit einem Geheimhaltungsalgorithmus geschützt werden, um ein kryptographisches Siegel zu bilden und/oder durch Verwendung anderer Schutztechniken (einschließlich Hardware, z.B. sichere Halbleiter- und/oder Hardware-Verpackungsschutzmittel). Die maschinenlesbaren deskriptiven Datenstrukturen können selbst in Rechteverwaltungsdatenstrukturen verpackt werden, und es können Regeln (z.B. Zulassungsaufzeichnungen) mit ihnen assoziiert werden, die den Zugang zu ihnen kontrollieren.

**[0026]** Gemäß einem Aspekt davon, wie deskriptive Datenstrukturen gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung vorteilhaft benutzt werden können, kann eine maschinenlesbare deskriptive Datenstruktur von einem Anbieter erstellt werden, um das Layout der konkreten Rechtsverwaltungsdatenstruktur(en), wie etwa sicherer Container, des Anbieters zu beschreiben. Diese Vorlagen der deskriptiven Datenstruktur ("DDS") können zur Erstellung von Containern benutzt werden. Eine Auswahl zwischen zwei oder mehr möglichen DDS kann auf einer oder mehreren Klassen basieren und/oder eine oder mehrere Klassen können auf Parameterdaten basieren. Die DDS kann geladen und als Layoutregeln für sichere Container, die erstellt werden, benutzt werden. Der Anbieter kann die DDS privat halten oder sie veröffentlichen, damit andere Anbieter auf der Basis derselben DDS kompatible und interoperable Container erstellen können.

**[0027]** Deskriptive Datenstrukturen können auch von einem Container-Viewer, -Browser, -Leser oder einer anderen Endbenutzeranwendung, die dafür ausgelegt ist, mit Containern zu arbeiten, benutzt werden. Es können wirklich generische Viewer oder andere Anwendungen geschrieben werden, die einen Container in einem beliebigen Format mindestens teilweise durch Nutzung deskriptiver Datenstrukturen verarbeiten können. Somit kann eine deskriptive Datenstruktur dazu verwendet werden, einen generischen Viewer (oder eine andere Anwendung) zumindest vorübergehend zu einem spezialisierten Viewer (oder einer anderen Anwendung) zu konvertieren und/oder anzupassen, der bzw. die um eine oder mehrere Klassen von Containern herum optimiert ist. Zusätzlich können spezialisierte Leser bereitgestellt werden, um deskriptive Datenstrukturen effizient zu verarbeiten, um Schlüsselmedienelemente zu finden (z.B. Titelseite, Inhaltsverzeichnis, Inseratenindex, Glossar, Artikel, ungeschützte Vorschau, Preis und/oder Rechteinformationen in Bezug auf Ansicht, Ausdrucken, elektronisches Abspeichern, Weiterverteilung, diesbezügliche Budgets und/oder andere Parameterinformationen usw.).

**[0028]** Solche spezialisierten Leser können dann nahtlos, transparent und automatisch verarbeiten, um dem Benutzer eine leicht zu benutzende Schnittstelle zu präsentieren (z.B. eine Symbolanzeige für jedes der Schlüsselmedienelemente), die für die spezifische Anwendung, den spezifischen Container und/oder den spezifischen Benutzer optimiert ist. Anders und/oder anders präsentiert können solche Elemente zum Beispiel auf der Basis der Identität des Benutzers und/oder des Benutzerknotens angezeigt oder anderweitig verwendet werden, wobei man zum Beispiel eines oder mehrere Klassenattribute, die sich auf solche automatisierte Verarbeitung auswirken können, berücksichtigen kann.

**[0029]** Es können außerdem zwei oder mehr DDS mit einem Container und/oder Containerinhalten sowie zum Beispiel einer oder mehreren Benutzer- und/oder Knotenklassen assoziiert werden. Eine Auswahl unter zwei oder mehr möglichen DDS für einen gegebenen Container und/oder eine gegebene Klasse von Containern und/oder Containerinhalten kann deshalb auf einer oder mehreren Klassen und/oder einer oder mehreren Klassen auf der Basis von Parameterdaten basieren. Insgesamt ist diese Möglichkeit, gespeicherte, optimierte und angepasste Containermodelle leicht zu charakterisieren und/oder wiederzuverwenden, und die nachfolgende Transparenz der Übersetzung aus solchen angepassten Containern (z.B. spezifischen DDS) in Vielzahl-Recherverwaltungsbenutzung besonders nützlich. Zum Beispiel, wenn solche angepassten DDS als Grundlage für die Erstellung von angepasster und optimierten Anzeige von Containerinhalt und/oder Kontrollinformation verwendet werden können, um die Leichtigkeit der Benutzung, die Effizienz, Transparenz und die Optimierung einer verteilten verallgemeinerten Recherverwaltungsumgebung wesentlich zu verbessern. In einer solchen Umgebung können zum Beispiel Benutzerknoten mit verschiedenen DDS interagieren, um sich automatisch auf die Anforderungen der mit solchen DDS assoziierten kommerziellen oder anderweitigen Rechtemodelle einzustellen.

**[0030]** Bestimmte Anbieter können beträchtliche Zeit mit dem Entwurf komplizierter deskriptiver Container-Datenstrukturen verbringen, die das Layout ihrer assoziierten Container beschreiben. Mit dieser Art von Investition in Struktur und Format wird die deskriptive Datenstruktur häufig durch ihre Wiederverwendung für dieselbe oder ähnliche Anwendungen signifikanten Wert besitzen. Entitäten können deskriptive Datenstrukturen hausintern benutzen, um einheitliche und hocheffiziente Erstellung von Containern sicherzustellen. Drittanbieter (d.h. ein von dem für die Erstellung deskriptiver Datenstrukturen verantwortlichen verschiedener Anbieter) können diese deskriptiven Datenstrukturen benutzen, wenn sie mit anderen Entitäten kompatible Container erstellen möchten. Ein Beispiel liegt vor, wenn der Verleger einer Zeitung mit großer Auflage eine deskriptive Datenstruktur zum Lesen seiner Zeitung entwickelt. Andere, kleinere Zeitungen möchten möglicherweise etwaige Viewer oder andere für die Verwendung mit der Zeitung mit großer Auflage eingerichtete Werkzeuge durch Verwendung desselben Containerformats nutzen. Deskriptive Datenstrukturen können urheberrechtlich geschützt werden und/oder sind anderweitig sowohl per Gesetz als auch durch das Recherverwaltungssystem selbst schützbar. Zum Beispiel können sie auch durch ihre eigenen Container und assoziierten Kontrollen geschützt werden, um sicherzustellen, daß Verfasser deskriptiver Datenstrukturen und/oder Distributoren und/oder ande-

re Benutzer solcher DDS faire, durch Rechtesystem verwaltete Einnahmen an ihrer Erstellung einer deskriptiven Datenstruktur und/oder benutzungsbezogenen Bemühungen erhalten.

**[0031]** Zusätzlich zu dem Obigen folgt eine Liste von Merkmalen und Vorteilen, die gemäß Aspekten der vorliegenden Erfindung entstehen:

- **Integritätseinschränkungen:** die deskriptive Datenstruktur ermöglicht es dem Anbieter, die Integrität eines Inhalts zu schützen, indem die Spezifikation von Integritätseinschränkungen ermöglicht wird. Integritätseinschränkungen geben die Möglichkeit, integritätsbezogene Regeln über den Inhalt anzugeben.

- **Anwendungserzeugung:** die deskriptive Datenstruktur kann zum Erzeugen eines oder mehrerer Teile von Softwareprogrammen, die Rechteverwaltungsstrukturen manipulieren, verwendet werden. Zum Beispiel könnte eine deskriptive Datenstruktur als "Anweisungen" dienen, die eine automatisierte Verpackungsanwendung für digitalen Inhalt und/oder einen automatisierten Leser von digitalem Inhalt steuern, wie zum Beispiel Anzeigeprioritäten und -organisation (z.B. Reihenfolge und/oder Layout).

- **Dynamische Benutzeroberflächen zur Erzeugung von Anwendungen:**

Anwendungen können eine deskriptive Datenstruktur lesen, um eine Oberfläche zu erzeugen, die für Datenerstellung, Editieren und/oder Komposition für ein spezifisches Modell optimiert ist, darunter Modelle, bei denen zum Beispiel komplexer Inhalt aus textlichen, Audio-Video- und interaktiven (z.B. Anfrage-)Elementen zusammengestellt wird. Die Daten können die Form eines Containers, einer Datenbank und/oder einer beliebigen anderen digitalen Informationsorganisation als ein beliebiges einfaches oder zusammengesetztes und komplexes Dateiformat annehmen. Anwendungen können auch eine diskriptive Datenstruktur lesen, um zu lernen, wie eine Oberfläche zum Sammeln und/oder Erstellen von Inhalt am besten anzuzeigen ist.

- **Dynamische Benutzeroberflächen für Anzeigenanwendungen:**

Anwendungen können auch eine deskriptive Datenstruktur lesen und eine für Datenanzeige geeignete Schnittstelle erzeugen. Bei diesen Daten kann es sich um einen Container, eine Datenbank oder ein beliebiges anderes zusammengesetztes komplexes Dateiformat handeln. Anwendungen können auch eine deskriptive Datenstruktur lesen, um zu lernen, wie eine Oberfläche zur Präsentation von Inhalt am besten anzuzeigen ist. Anwendungen können ferner eine deskriptive Datenstruktur lesen, um zu lernen, wie Anzeigefunktionen in Bezug auf Interaktion – zur Inhaltserstellung und/oder -verpackung und/oder für Benutzeranzeigezwecke, darunter Optimierung beliebiger

solcher Interaktionen – mit einer anderen einzelnen oder mehreren anderen Anwendungen, Smart Agents, Datenverarbeitungs Umgebungen, Identität (einschließlich etwaiger Klassenidentitäten) von Benutzer und/oder Benutzerknoten usw. zu verwalten sind. Zum Beispiel könnte eine Benutzeroberfläche für die Interaktion mit Folgendem unterschiedlich optimiert werden: ein Mitglied der U.S. Air Force im Gegensatz zu einem Fakultätsmitglied der Sozialwissenschaften an einer Universität; oder ein Mitglied eines Kiwanis-Clubs im Gegensatz zu einem Mitglied eines Clubs der protestantischen Kirche, ein Bürger der Vereinigten Staaten im Gegensatz zu einem Bürger von Saudi Arabien, einschließlich einer entsprechenden Anzeige von erwarteten Klassenzugehörigkeitssymbolen und diesbezüglicher entsprechender Organisation oder Unterdrückung angezeigter Informationen.

- **Möglichkeit zum automatischen Identifizieren und Finden von Datenfeldern:** Volltextsuche, Agenten, Web-Spider und dergleichen ziehen aus in einem oder mehreren Bereichen einer DDS enthaltenen Informationen Nutzen und können mit ihnen interagieren, wenn Bereiche in einer Datei bekanntermaßen potentiell interessante Informationen enthalten und solche Informationen in einem vordefinierten Format präsentiert werden.

- **Die Möglichkeit, benötigte oder gewünschte Daten ohne Erstkenntnis des Datenformats zu extrahieren:** Volltextsuche, Agenten, Web-Spider und dergleichen ziehen aus in einem oder mehreren Bereichen einer DDS enthaltenen Informationen Nutzen und können mit ihnen interagieren, wenn große Dateien beliebiger Komplexität und mit unbekanntem Ursprung ohne Spezialkenntnisse verarbeitet werden können.

- **Effiziente, maschinen- bzw. menschenlesbare Datenzusammenfassung:** die deskriptiven Datenstrukturen können optimal klein, zweckmäßig und kosteneffektiv zu verarbeiten, zu senden und/oder zu speichern sein.

- **Wiederverwendbar, skalierbar – unabhängig von tatsächlichen Daten:** deskriptive Datenstrukturen können beliebig komplex und deshalb potentiell zeitaufwändig zu konstruieren sein und erfordern bestimmte Expertise. Dadurch erhält die deskriptive Datenstruktur Wiederverkaufswert.

- **Definition und Umdefinition des Inhalts-Layout im Verlauf:** die Arbeit mit einem Layout-Werkzeug ermöglicht schnelle Iterationen (einschließlich Editieren und Modifikationen) eines Entwurfs (Layout), die zweckmäßiger und kosteneffektiver als die Erstellung eines solchen Layouts sein können, was auch relativ schwierig sein kann oder die Expertise vieler Benutzer übersteigt.

- **Deskriptive Datenstrukturattribute erlauben den Fall nicht gefundener Meta-Eigenschaften in tatsächlichen Daten:** da sowohl Erstellungs- als auch Nach-Erstellungsprozesse dieselbe deskrip-

tive Datenstruktur verarbeiten, können Meta-Informationen in der deskriptiven Datenstruktur abgelegt werden, die ansonsten in dem verpackten Inhalt nicht verfügbar wären. Ein Beispiel hierfür: ob die Anzeige bestimmter Felder "erforderlich" oder "verborgen" ist.

- Ermöglicht Entwurfsautomatisierung über "Wizards" der deskriptiven Datenstruktur: deskriptive Datenstrukturen ermöglichen selbst eine weitere Automatisierung über "Wizards". Es kann zum Beispiel deskriptive Datenstrukturen geben, die bei der Definition anderer deskriptiver Datenstrukturen helfen. Andere deskriptive Datenstrukturen definierende deskriptive Datenstrukturen könnten die unvollständige deskriptive Datenstruktur zum Beispiel für ein Buch oder Magazin repräsentieren. Der "Wizard" kann eine Reihe von Dialogkästen umfassen, die dem Benutzer angezeigt werden, um die fehlenden Informationen auszufüllen, damit sie eine abgeschlossene deskriptive Datenstruktur wird.

- Anwendungen außerhalb einer bestimmten Rechteverwaltungsarchitektur: polymorphe Anwendungen können zum Beispiel deskriptive Datenstrukturen benutzen, um bestimmte Datenvisualisierungsattribute und/oder -anforderungen zu bestimmen, wie zum Beispiel, welcher Look und Feel dem Benutzer angezeigt werden soll. Wenn zum Beispiel eine deskriptive Datenstruktur eine Textverarbeitungsdokumentreferenz enthält, könnte die polymorphe Anwendung eine für Anzeige und Editieren eines Dokuments geeignete Schnittstelle erzeugen. Wenn die deskriptive Datenstruktur Referenzen auf viele ausführbare Programme enthält, könnte die polymorphe Anwendung vom Benutzer erfragen, wo die Dateien abgespeichert werden sollen.

- Ermöglicht Schirm-Anwendungen, deskriptive Datenstrukturen zu verarbeiten und unbekannte Dateitypen und Prozesse zu deligieren:

Schirm-(oder polymorphe)Anwendungen können zum Beispiel im Wesentlichen als Operation für eine bestimmte Datei wirken. Diese Schirm-Anwendung kann die Dinge in der Datei, um die sie sich kümmert, extrahieren und verarbeiten, während sie die Dinge, die sie nicht versteht, ignoriert oder delegiert (zum Beispiel an Benutzer und/oder Wertkettenpartner (z.B. Distributor), um die Anzeige solcher Posten zu steuern).

- Laufzeit-Interpretation: es ist möglich, eine deskriptive Datenstruktur in der Laufzeit zu interpretieren, wodurch materiell vergrößerte Effizienzen und Pünktlichkeit bereitgestellt werden.

- Laufzeit-Anpassbarkeit: Systeme können sich an in Echtzeit ankommende dynamische Daten durch Verwendung von deskriptiven Datenstrukturen anpassen.

- Automatische Konvertierungsfähigkeit: deskriptive Datenstrukturen verwendet zum automatischen Konvertieren von einem Format in ein an-

deres.

- Vereinfachter Systementwurf: die Verwendung deskriptiver Datenstrukturen kann die Notwendigkeit einer sekundären "Wrapper"-Anwendungsprogrammierschnittstelle (API) oder einer anderen Anordnung zum sicheren "Eingrenzen" des Container-Erstellungsprozesses stark verringern. Eine solche "Wrapper"-API zum Steuern und anderweitigen Einschränken des Container-Erstellungsprozesses könnte ansonsten notwendig sein, um sicherzustellen, dass alle erstellten Container kompatibel sind, wodurch die Flexibilität und die Möglichkeit der Anpassung begrenzt werden.

- Objektorientierte Vorlagen-Programmierung: die Verwendung von anzeigebezogenen, interaktionsbezogenen und rechtebezogenen Konzeptobjekten, die durch Benutzeroberflächenauswahlen auf hoher Ebene und Priorisierungen und Spezifikation relevanter Parameterdaten ausgewählt werden können, wobei dies eine sehr leichte Erstellung bestimmter Vorlagenkategorien ermöglicht – wie etwa Konstruktion und Anzeige-Hinweisinformationen.

- Verwendung einer Vorlagesprache und eines Interpreters unter Einbindung von unterstützender Programmierung durch Verwendung von Sprachelementen und Interpretation solcher Sprache durch Knoten, wie in Ginter et al. beschrieben, wobei solche Sprachelemente umfasst, die Anzeige, Rechte und Programminteraktionselemente, Prioritäten und Parameterdaten beschreiben.

#### Kurze Beschreibung der Zeichnungen

**[0032]** Diese und weitere Merkmale und Vorteile von zur Zeit bevorzugten beispielhaften Ausführungsformen gemäß der Erfindung werden durch Bezugnahme auf die folgende ausführliche Beschreibung in Verbindung mit den Zeichnungen besser und vollständiger verständlich. Es zeigen:

**[0033]** [Fig. 1A](#) und [Fig. 1B](#) beispielhafte Inhaltscontainer;

**[0034]** [Fig. 2A](#) und [Fig. 2B](#) beispielhafte Inhaltscontainer, die mit beispielhaften deskriptiven Datenstrukturen assoziiert sind;

**[0035]** [Fig. 3](#) einen beispielhaften Erstellungs- und Benutzungsprozess für deskriptive Datenstrukturen;

**[0036]** [Fig. 4](#) einen weiteren beispielhaften Erstellungs- und Benutzungsprozess;

**[0037]** [Fig. 5](#) eine beispielhafte Systemarchitektur, die deskriptive Datenstrukturen verwendet;

**[0038]** [Fig. 5A](#) einen durch das System von [Fig. 5](#)



ausgeführten beispielhaften Prozess;

[0039] [Fig. 6](#) eine hierarchische Organisation deskriptiver Datenstrukturen;

[0040] [Fig. 6A](#) ein Beispiel dafür, wie deskriptive Datenstrukturen mit atomischen Transaktionsdaten benutzt werden können;

[0041] [Fig. 7](#) ein beispielhaftes Format deskriptiver Datenstrukturen;

[0042] [Fig. 8](#) eine beispielhafte graphische Oberfläche zur Erstellung deskriptiver Datenstrukturen;

[0043] [Fig. 9](#) einen beispielhaften Prozess zum Verfolgen von die Rechteverwaltung betreffenden Daten deskriptiver Datenstrukturen;

[0044] [Fig. 10A](#) eine beispielhafte Verwendung deskriptiver Datenstrukturen zur Bereitstellung von Interoperabilität zwischen Umgebungen; und

[0045] [Fig. 10B](#) weitere Einzelheiten darüber, wie die beispielhafte deskriptive Datenstruktur von [Fig. 10A](#) organisiert werden kann.

Ausführliche Beschreibung zur Zeit bevorzugter beispielhafter Ausführungsformen

[0046] [Fig. 2A](#) und [Fig. 2B](#) zeigen die beispielhaften Container **100a**, **100c** von [Fig. 1A](#), [Fig. 1B](#), die mit maschinenlesbaren deskriptiven Datenstrukturen **200** und **200'** assoziiert sind. Mit Bezug auf [Fig. 2A](#) ist eine deskriptive Datenstruktur **200** mit dem Inhaltscontainer **100a** assoziiert. Mit dieser deskriptiven Datenstruktur **200** kann man den Inhalt (und bestimmte andere Eigenschaften) des Containers **100a** definieren. In dem gezeigten Beispiel definiert die deskriptive Datenstruktur **200** eine Anzahl von Teilen von Inhalt **102** im Zeitungsstil, wie zum Beispiel die Schlagzeile (Deskriptor **202a**), das Ausgabedatum (Deskriptor **202b**), den Aufmacher (Deskriptor **202c**), neueste Nachrichten (Deskriptor **202d**), Bilder (Deskriptor **202e**), Anzeigen (Deskriptor **202f**) und Teil (Deskriptor **202g**).

[0047] Die Definitionen **202** der deskriptiven Datenstruktur enthalten oder spezifizieren in diesem Beispiel nicht den konkreten Inhalt entsprechender Teile der Zeitung **102**, sondern definieren stattdessen abstrakter ein generisches Format, das eine Publikation des Zeitungsstils benutzen könnte. Zum Beispiel spezifiziert die beispielhafte Schlagzeilendefinition **202a** der deskriptiven Datenstruktur von [Fig. 2A](#) keine konkrete Schlagzeile (z.B. "Yankees gewinnen Pennant!"), sondern definiert stattdessen die Stelle (zum Beispiel die logische oder anderweitige Offset-Adresse) in der Container-Datenstruktur **100a** (sowie bestimmte andere Eigenschaften), in der sol-

che Schlagzeileninformationen verankert sein können. Da die deskriptive Datenstruktur **200** generisch für eine Klasse oder Familie von Inhaltspublikationen des Zeitungsstils ist, kann sie wiederverwendet werden. Zum Beispiel könnte jede tägliche Ausgabe einer Zeitung unter Verwendung von und/oder in Assoziation mit derselben deskriptiven Datenstruktur **200** erstellt werden. Durch abstraktes Definieren des Datenformats und anderer Eigenschaften des Zeitungsstil-Inhalts **102** ermöglicht die deskriptive Datenstruktur **200** ein leichtes Erstellen, Benutzen und Manipulieren des Zeitungsstil-Inhalts **102**.

[0048] Mit Bezug auf [Fig. 2B](#) kann eine andere deskriptive Datenstruktur **200'** dazu benutzt werden, eine andere Klasse von Inhaltspublikationen **106**, wie zum Beispiel Modejournale, zu definieren. Die deskriptive Datenstruktur **200'** für diese Inhaltsklasse spiegelt im Vergleich zu der in [Fig. 2A](#) gezeigten deskriptiven Datenstruktur **200** ein anderes Format (und möglicherweise andere Eigenschaften) wider. Da Modejournale in der Regel keine Schlagzeilen oder neuestem Nachrichten enthalten, kann die beispielhafte deskriptive Datenstruktur **200'** zum Beispiel solche Formatierung nicht definieren. Stattdessen kann die deskriptive Datenstruktur **200'** zum Definieren einer Klasse von Modejournalinhalt das Ausgabedatum (Deskriptor **204a**), einen Journaltitel (Deskriptor **204b**), den Namen eines Photographen (Deskriptor **204c**) und assoziierte Bebilderungskennzeichnung (Deskriptor **204d**) definieren.

[0049] Die Beispiele von [Fig. 2A](#) und [Fig. 2B](#) zeigen deskriptive Datenstrukturen **200**, **200'**, die zusammen mit assoziiertem Inhalt **102**, **106** in Inhaltsobjektcontainern **100a**, **100c** abgeliefert werden. Es können jedoch auch andere Formen der Assoziation benutzt werden. Zum Beispiel kann die deskriptive Datenstruktur **200** zusammen mit assoziierten Regeln, die Zugriff darauf und/oder Verwendung kontrollieren, unabhängig in ihrem eigenen separaten Container abgeliefert werden. Als Alternative könnten die deskriptiven Datenstrukturen **200** in einer Bibliothek gespeichert und bei Bedarf abhängig von bestimmten Anforderungen in gesicherter oder ungesicherter Form abgeliefert werden.

[0050] Obwohl [Fig. 2A](#) und [Fig. 2B](#) Inhaltsbeispiele von gedruckten Publikationen sind, ist zusätzlich die Verwendung der deskriptiven Datenstrukturen **200** nicht darauf beschränkt. Im Gegenteil kann man die deskriptiven Datenstrukturen **200** zum Definieren des Formats und/oder anderer mit vielfältigen verschiedenen Arten von digitalen Informationen assoziierten Eigenschaften verwenden, darunter zum Beispiel:

- Bilder
- Ton
- Video
- Computerprogramme
- Methoden

- ausführbare Dateien
- interpretierbare Dateien
- Währungsobjekte
- Währungscontainer für Währungsobjekte
- Regeln
- beliebige Eingaben
- beliebige Computerausgaben
- andere deskriptive Datenstrukturen
- beliebige andere Informationen

#### Beispielhafter Prozess zum Erstellen und Benutzen deskriptiver Datenstrukturen

[0051] [Fig. 3](#) zeigt einen beispielhaften Prozess zum Erstellen und Benutzen von deskriptiven Datenstrukturen **200**. In diesem Beispiel wird ein Layout-Werkzeug **300** benutzt, um die deskriptive Datenstruktur **200** zu erstellen. Dieses Layout-Werkzeug **300** kann zum Beispiel ein softwaregesteuerter Prozess sein, der über eine graphische Benutzeroberfläche mit einem Menschen interagiert. Die resultierende deskriptive Datenstruktur **200** (die auf einem Massenspeichergerät oder einem anderen Speicher gespeichert werden kann) kann dann dazu verwendet werden, eine beliebige Anzahl anderer Prozesse zum Erstellen oder Interpretieren gespeicherter Daten zu ermöglichen. Zum Beispiel kann die deskriptive Datenstruktur in einem Erstellungsprozess **302** verwendet werden. Der Erstellungsprozess **302** kann die deskriptive Datenstruktur lesen und als Reaktion eine Ausgabedatei **400** mit einem vordefinierten Format erzeugen, wie zum Beispiel einen Container **100** entsprechend einem Format, das durch die deskriptive Datenstruktur **200** beschrieben wird. Ein Viewing-Prozess **304** kann die deskriptive Datenstruktur **200** dazu benutzen, wichtige Posten in der Ausgabedatei **400** zur Anzeige zu finden. Ein Browsing-Prozess **306** kann die deskriptive Datenstruktur **200** dazu benutzen, Posten in der gespeicherten Ausgabedatei **400** zu finden, wie zum Beispiel Schlüsselwörter oder anderen durchsuchbaren Text. Die deskriptive Datenstruktur **200** kann Integritätseinschränkungen oder Regeln zuführen, die die Integrität von entsprechendem Inhalt während der Benutzung und/oder des Zugriffs auf den Inhalt schützen.

[0052] [Fig. 4](#) zeigt einen ausführlicheren beispielhaften Erstellungs- und Benutzungsprozess deskriptiver Datenstrukturen. Bei diesem Beispiel kann das Layout-Werkzeug **300** über eine graphische Benutzeroberfläche **312** bereitgestellte Benutzereingaben **310** annehmen. Die Ausgabe des Layout-Werkzeugs **300** kann eine deskriptive Datenstruktur **200** zum Beispiel in Form einer Textdatei sein. Ein gesicherter Verpackungsprozess **302a** kann containerspezifische Daten als Eingabe annehmen und kann außerdem als Nulleseeingabe die deskriptive Datenstruktur **200** annehmen. Der Verpacker **302a** würde auf einer graphischen Benutzeroberfläche basieren und/oder könnte automatisiert werden. Der Verpa-

cker **302a** verpackt die containerspezifischen Daten **314** in einem sicheren Container **100**. Außerdem kann er gegebenenfalls auch die deskriptive Datenstruktur **200** in demselben Container **100** verpacken. Ein Viewer **304** kann Daten **314** mit Hilfe der deskriptiven Datenstruktur **200** und gemäß den mit dem Container verpackten für die Daten **314** geltenden Regeln **316** und/oder der deskriptiven Datenstruktur **200** ansehen.

#### Beispielhafte Architektur zur Verwendung deskriptiver Datenstrukturen

[0053] [Fig. 5](#) zeigt eine beispielhafte Architektur eines gesicherten Systems, die sich für die Verwendung mit der deskriptiven Datenstruktur **200** eignet. In diesem Beispiel kann ein elektronisches Gerät **500** des in der oben zitierten Patentschrift von Ginter et al. beschriebenen Typs innerhalb einer manipulationssicheren Barriere **502** bereitgestellt werden. Das elektronische Gerät **500** kann eine Anwendungsschnittstelle (API) **504** enthalten. Über die API **504** können eine oder mehrere Anwendungen **506** mit dem elektronischen Gerät **500** kommunizieren. Bei bestimmten Beispielen kann die Anwendung **506** auf dem gesicherten elektronischen Gerät **500** ausgeführt werden. Jede Anwendung **506** kann einen Interpreter **508** für deskriptive Datenstrukturen enthalten. Bei der Benutzung kann das elektronische Gerät **500** auf den sicheren Container **100** zugreifen und – gemäß den Regeln **316** – auf die deskriptive Datenstruktur **200** und den darin enthaltenen Inhalt **102** zugreifen und ihn der Anwendung **506** zuführen. Der Interpreter **508** in der Anwendung **506** kann dann seinerseits die deskriptive Datenstruktur **200** lesen und benutzen. Zusätzlich kann die Anwendung **506** in dem Sinne polymorph sein, daß sie eine Personalität oder ein Verhalten annehmen kann, die bzw. das mindestens teilweise durch die deskriptive Datenstruktur **200** definiert wird.

[0054] [Fig. 5A](#) zeigt einen von der beispielhaften gesicherten Systemarchitektur von [Fig. 5](#) ausgeführten beispielhaften ausführlichen Prozess. In diesem Beispiel fordert die Anwendung **506** das Gerät **500** auf, die deskriptive Datenstruktur **200** aus dem Container **100** abzurufen (Block **550**). Das elektronische Gerät **500** liest die deskriptive Datenstruktur **200** und führt unter den durch die assoziierten Regeln **316** spezifizierten Bedingungen die deskriptive Datenstruktur **200** der Anwendung **506** zu (Block **552**). Die Anwendung **506** fordert ihren Interpreter **508** dann auf, die deskriptive Datenstruktur **200** zu interpretieren (Block **554**). Der Interpreter **508** teilt der Anwendung **506** mit, was die deskriptive Datenstruktur **200** aussagt (Block **556**). Die Anwendung **506** extrahiert oder erhält die Informationen der deskriptiven Datenstruktur, die sie benötigt oder wünscht, von dem Interpreter **508** (Block **558**). Zum Beispiel nehme man an, die Anwendung **506** möchte die Informationen

"Schlagzeile" in dem in [Fig. 2A](#) gezeigten Zeitungsstil-Inhalt anzeigen. Die Anwendung **506** kann den Interpreter **508** auffordern, ihr die Informationen zu geben, die dabei helfen werden, diese "Schlagzeilen-Informationen" zu finden, zu lesen, zu formatieren und/oder anzuzeigen.

**[0055]** Als ein weiteres Beispiel kann der Interpreter **508** der Anwendung **506** eine Elementidentifikation (z.B. einen hexadezimalen Wert oder eine andere Kennung) geben, die den Schlagzeileninformationen in dem Zeitungsstil-Inhalt entspricht (Block **558**). Die Anwendung **506** kann dann das elektronische Gerät **500** auffordern, ihr die Schlagzeilen-(oder anderweitigen)Inhaltsinformationen **102** in dem Container **100** zuzuführen, indem dem elektronischen Gerät **500** über die API **504** entsprechende Inhaltsinformationen zugeführt werden (Block **560**). Zum Beispiel kann die Anwendung **506** dem elektronischen Gerät **500** die Element-ID übergeben, die der Interpreter **508** der Anwendung zugeführt hat. Obwohl die Anwendung **506** möglicherweise keine direkte Kenntnis darüber hat, was sich in dem Container **100** befindet (und möglicherweise nur durch einen durch das Gerät **500** bereitgestellten sicheren VDE-Knoten auf den Container **100** zugreifen kann), kann der Interpreter **508** (durch Betrachten der deskriptiven Datenstruktur **200**) der Anwendung **506** genug Informationen mitteilen, damit die Anwendung weiß, wie die von ihr gewünschten Informationen von dem elektronischen Gerät **500** anzufordern sind.

**[0056]** Das elektronische Gerät kann dann auf die Informationen **102** in dem Container **100** zugreifen und die angeforderten Informationen (gemäß den Regeln **316** in dem Container) an die Anwendung **506** abliefern (Block **562**). Die Anwendung **506** kann dann die ihr durch das elektronische Gerät **500** gegebenen Informationen mindestens teilweise auf der Basis des ihr von dem Interpreter **508** über die Inhaltsinformationen Mitgeteilten benutzen (Block **564**). Zum Beispiel kann die deskriptive Datenstruktur **200** Charakteristika darüber bereitstellen, wie die Anwendung **506** mit den Informationen **102** umgehen soll. Zum Beispiel kann die deskriptive Datenstruktur **200** der Anwendung **506** vorschreiben, immer ein bestimmtes Feld (z.B. das Autor- oder Copyright-Feld) anzuzeigen und niemals andere Informationen (z.B. Informationen, die vor den meisten Benutzern verborgen werden sollten) anzuzeigen. Die DDS **200** kann außerdem vollständige Präsentations- oder "Visualisierungs"-Informationen bereitstellen, so dass ein Informationsanbieter zum Beispiel Look und Feel der Informationen kontrollieren kann, wenn sie angezeigt oder anderweitig wiedergegeben werden. Die deskriptive Datenstruktur **200** kann Codierung anderer Charakteristika in Form von Metadaten bereitstellen, die auch während des Vorgangs des Erstellens, Benutzens oder Manipulierens des Containers **100** von der Anwendung **506** benutzt werden können. Die

DDS **200** kann zur Erzeugung eines Softwareprogramms zum Manipulieren von Rechteverwaltungsstrukturen verwendet werden. Zum Beispiel könnte eine DDS **200** als die "Anweisungen" dienen, die eine automatisierte Verpackungsanwendung für digitalen Inhalt oder einen automatisierten Leser von digitalem Inhalt steuern.

Durch die destruktive Datenstruktur bereitgestellte beispielhafte Beschreibung(en)

**[0057]** [Fig. 6](#) zeigt ein Beispiel dafür, wie eine deskriptive Datenstruktur **200** eine beliebig komplexe Informationsstruktur, wie zum Beispiel einen hierarchischen Container **100**, beschreiben und definieren kann. In diesem konkreten Beispiel enthält der Container **100** Eigenschaften **600(1)**, **600(2)**. Die Eigenschaft **600(1)** kann n Attribute **602(1)**, **602(2)** ... **602(n)** enthalten. Die Eigenschaft **600(2)** kann eine beliebige Anzahl von Attributen **604(1)**, **604(2)** enthalten und kann außerdem eine zusätzliche Eigenschaft **606** enthalten. Die Eigenschaft **606** kann ihrerseits ihre eigenen Attribute **608(1)**, **608(2)** ... enthalten. Die assoziierte deskriptive Datenstruktur **200** kann als Baumstrukturliste **250** organisiert werden, die eine rekursive Struktur zum Widerspiegeln der rekursiven Struktur der Inhalte des Containers **100** bereitstellt. Zum Beispiel kann die Liste **250** "Zweige" in Form von "Eigenschafts"-Deskriptoren **252(1)**, **252(2)** entsprechend den Eigenschaften **600(1)**, **600(2)** enthalten. Jeder Eigenschafts-Deskriptor **252** kann seinerseits eine Liste **254** von Attributen enthalten und kann zusätzliche Eigenschafts-Deskriptoren **256** in derselben rekursiven hierarchischen Anordnung enthalten, die die beispielhafte Inhaltscontainerstruktur widerspiegelt. Die DDS **200** kann zur Beschreibung von beliebig komplexen hierarchischen oder nichthierarchischen Datenstrukturen beliebiger Dimension (1 bis n) verwendet werden.

**[0058]** [Fig. 6A](#) zeigt, dass die deskriptive Datenstruktur **200** in Verbindung mit einer beliebigen Art von Informationen verwendet werden kann, wie zum Beispiel Ereignisse oder Methoden, die eine "atomische Transaktion", wie etwa eine Immobilien-Transaktion, definieren. In diesem Beispiel von [Fig. 6A](#) enthält ein Container **100** eine oder mehrere deskriptive Datenstrukturen **200** und assoziierte Steuerermenge(n) **316** in Bezug auf eine Sequenz von "Ereignissen" **700**, die eine Immobilien-Transaktion definieren. Die DDS **200** kann zum Beispiel eine Anzahl verschiedener Einträge **200A–200N** betreffend jedes verschiedene "Ereignis" innerhalb der Transaktion (z.B. "Angebot", "Akzeptanz", "Kauf/Verkauf", "Inspektion", "Hypothek" usw.) enthalten. Diese Einträge **200A–N** können zum Beispiel definieren, wo in dem Container **100** das Ereignis gefunden werden kann. Die Einträge **200A–200N** können auch Metadaten enthalten, die zusätzliche Charakteristika entsprechend dem Ereignis bereitstellen (zum Beispiel,

wie bestimmte Informationen in Bezug auf das Ereignis angezeigt werden sollen).

Beispielhafte Formatierung einer deskriptiven Datenstruktur

**[0059]** [Fig. 7](#) zeigt ein Beispiel dafür, wie die deskriptive Datenstruktur **200** formatiert werden kann. Wie bereits erwähnt, kann die deskriptive Datenstruktur **200** eine Liste, wie etwa eine Verbundliste, umfassen. Jeder Listeneintrag **260(1)**, **260(2)**, ... kann eine Anzahl von Datenfeldern enthalten, darunter zum Beispiel:

- ein Objektnamenfeld **262**,
- ein oder mehrere Metadatenfelder **264** (die Teil der deskriptiven Datenstruktur sein und/oder durch diese referenziert werden können); und
- Ortsinformationen **266** (mit denen dabei geholfen werden kann, die entsprechenden Informationen in der Containerdatenstruktur **100** zu identifizieren).

**[0060]** Das Objektnamenfeld **262** kann eine Konstante enthalten, die einer Art von Informationen entspricht oder diese beschreibt. Zum Beispiel kann das Objektnamenfeld **262** als "Handle" zu dem Inhalt oder den Daten wirken; es kann eine indirekte Referenz zu dem Inhalt oder den Daten sein; und/oder es kann zum Nachschlagen des Inhalts oder der Daten verwendet werden. Es folgen Beispiele für Objektnamen:

#### Vielzweck-Objektnamen

NUMMER  
KETTE  
DATUM  
TITEL  
BESCHREIBUNG  
AUTOR  
ANBIETER  
MIME\_TYP  
VERSION  
URL  
EMAIL  
NEWGROUP  
DATEI\_NAME  
SCHLÜSSELWÖRTER  
DATUM\_ERSTELLUNG  
DATUM\_MODIFIKATION  
DATUM\_LETZTER\_ZUGRIFF  
NATIVE\_PLATTFORM  
GRÖßE  
INHALT  
VORSCHAU  
THUMBNAIL  
TEXT  
BEBILDERUNG  
ILLUSTRATION  
UNBEKANNT

VORLAGE  
NAME RECHNUNG  
CONTAINER

#### Objektnamen des Buch-Stils

DATUM\_TERMIN  
TITEL\_SEITE  
PROLOG  
EINFÜHRUNG  
ZUSAMMENFASSUNG  
INHALTSVERZEICHNIS  
KAPITEL  
KAPITEL\_NUMMER  
INDEX

#### Objektnamen E-Mail-Stil

VON  
ZU  
CC  
BETREFF  
NACHRICHT\_HAUPTTEIL  
ANLAGE

#### Objektnamen Zeitungs-Stil

AUSGABE\_DATUM  
ARTIKEL  
KOLUMNE  
TITELBLATTARTIKEL  
AUFMACHER  
NEUESTE\_NACHRICHTEN  
ANZEIGEN  
TEIL  
REDAKTION

**[0061]** Die DDS **200** kann eine beliebige Art von Daten oder Metadaten enthalten oder referenzieren. In einem Beispiel benutzt die DDS **200** das Objektnamenfeld **262** zum Zeigen auf oder Referenzieren von Metadaten. Diese Metadaten können bestimmte mit dem Objektnamen assoziierte Eigenschaften definieren. Zum Beispiel können solche Metadaten Integritäts- oder andere Einschränkungen während des Erstellungs- und/oder Benutzungsprozesses auferlegen (z.B. "beim Erstellen eines Objekts müssen Sie diese Informationen angeben", oder "beim Anzeigen des Objekts müssen Sie diese Informationen anzeigen"). Die Metadaten **262** können auch ferner den assoziierten Objektnamen beschreiben oder anderweitig bestimmen.

**[0062]** Bei einem bevorzugten Beispiel verwendet die DDS **200** den Objektnamen **262**, um auf an anderer Stelle gespeicherte Metadaten zu verweisen – zum Beispiel in einem Container **100**. Diese Referenzierungstechnik hat mehrere Vorteile. Eine Situation, in der es nützlich wäre, die Metadaten getrennt von der DDS **200** in einem sicheren Container **100** zu

speichern, wäre zum Beispiel in Situationen, in denen es erwünscht ist, dass die DDS ohne weiteres einer Anwendung außerhalb zugänglich ist, aber die assoziierten Metadaten zu schützen. Zum Beispiel betrachte man den Fall des Behandelns von Web-Spider-Anfragen. Ein Web-Spider kann die DDS **200** nach einem bestimmten Objektnamen **262** befragen. Wenn der Objektname gefunden wird, kann der Web-Spider die entsprechenden Metadaten anfordern. Der Web-Spider kann bereits Zugang zu den Metadaten haben, ist aber möglicherweise nur in der Lage, unter entsprechenden Bedingungen, die durch ein entsprechendes gesichertes elektronisches Gerät **500** auf der Basis assoziierter Regeln **316** kontrolliert werden, auf die assoziierten Metadaten aus dem Container **100** zuzugreifen. Als ein weiteres Beispiel kann das Speichern von Metadaten getrennt von der DDS **200** eine Benutzung derselben DDS mit verschiedenen Metadaten in verschiedenen Kontexten erlauben. Man nehme zum Beispiel an, dass eine DDS **200** einen Objektnamen, zum Beispiel SCHLÜSSELWÖRTER, enthält. Wenn die DDS **200** mit dem Container **100A** assoziiert ist, verweist der DDS-Objektname SCHLÜSSELWÖRTER auf die SCHLÜSSELWÖRTER-Metadaten des Containers **100A**. Wenn umgekehrt später dieselbe DDS **200** mit einem anderen Container **100C** assoziiert (z.B. verpackt) wird, verweist der DDS-Objektname SCHLÜSSELWÖRTER auf die SCHLÜSSELWÖRTER-Daten des Containers **100B**.

**[0063]** Obwohl es bevorzugt wird, den Objektnamen **262** als Verweis auf an anderer Stelle gespeicherte Metadaten zu verwenden, kann es andere Fälle geben, bei denen es notwendig oder wünschenswert ist, Metadaten explizit in die DDS **200** aufzunehmen. Beispielsweise zeigt [Fig. 7](#) eine beispielhafte DDS **200**, die das Metadatenfeld **264** enthält und außerdem unter Verwendung des Objektnamens **262** auf Metadaten in einen Container **100** verweist. Es können eine der Techniken oder beide verwendet werden.

**[0064]** Die DDS **200** erlaubt es somit Wertkettenteilnehmern, die Integrität von Inhalt zu schützen, indem die Spezifikation von Integritätseinschränkungen ermöglicht wird. Integritätseinschränkungen der DDS **200** ergeben eine Möglichkeit, Regeln über den Inhalt anzugeben. Zum Beispiel kann die DDS **200** spezifizieren, dass ein Artikel einer Zeitung nicht angesehen werden kann, ohne dass ihre Schlagzeile angesehen wird. Die entsprechende Integritätseinschränkung kann die folgende Regel angeben "wenn es einen Artikel gibt, muss es auch eine Schlagzeile geben". Ein anderes Beispiel ist ein Foto, das Teil eines Magazins ist, und die damit verbundene Erwähnung. Die durch die DDS **200** bereitgestellte Integritätseinschränkungsregel könnte folgendermaßen lauten: "dieses Foto nicht ohne assoziierte Erwähnung darstellen".

**[0065]** Durch DDS-Integritätseinschränkungen erhalten Wertkettenteilnehmer ein Werkzeug zum Schutz der Benutzung der DDS **200**, wodurch sichergestellt wird, dass durch eine bestimmte DDS repräsentierter Inhalt alle wesentlichen Komponenten enthält – dass sie die DDS repräsentiert. Dadurch erhalten Anbieter eine Möglichkeit zum Festlegen von Konventionen und zum Durchsetzen von Benutzungsstandards. Es gibt viele mögliche Integritätseinschränkungen. Es folgen einige wenige Beispiele:

- Vorgeschrieben: a ist als Teil des Inhalts vorgeschrieben
- Optional: a ist eine optionale Komponente des Inhalts
- Vorgeschriebene Beziehung: wenn a vorliegt, muss b vorliegen, oder wenn a vorliegt, müssen b, c und d vorliegen. Umgekehrt darf, wenn b nicht vorliegt, a nicht vorliegen. Beziehungen in dieser Kategorie sind 1:m mit  $m > 0$ .
- Optionale Beziehung: wenn a vorliegt, darf b vorliegen oder nicht. Wenn b vorliegt, liegt a garantiert vor. Beziehungen in dieser Kategorie sind 1:n mit  $n \geq 0$ .
- Wiederholung: a muss n mal auftreten, mit  $n > 1$ . Dies könnte mit Wertebereichen usw. spezifiziert werden.
- Andere Regeln und/oder Anforderungen. Metadaten **264**

Beispielhafte graphische Oberfläche zum Erstellen von deskriptiven Datenstrukturen

**[0066]** [Fig. 8](#) zeigt eine beispielhafte graphische Benutzeroberfläche **312** zum Erstellen von deskriptiven Datenstrukturen. In diesem Beispiel kann die graphische Benutzeroberfläche **312** den Benutzer zur Eingabe des Objektnamens auffordern. Zusätzlich kann die graphische Benutzeroberfläche **312** Optionen für das Spezifizieren der assoziierten Metadaten **264** bereitstellen. Zu den in [Fig. 8](#) gezeigten Optionen können zum Beispiel die folgenden gehören:

- Metadaten des "Konstruktionstyps" (bei Objekt-konstruktion sind die Informationen erforderlich; bei Objekt-konstruktion soll das Objekterstellungswerkzeug immer oder niemals zur Eingabe der Informationen auffordern);
- Metadaten anzeigen (z.B. immer die assoziierten Informationen anzeigen (z.B. für Copyright-Vermerke, Autorennamen und dergleichen) oder die Informationen immer oder niemals sichtbar machen; und/oder
- Layout-"Hinweise" und Felddefinitionen (z.B. Text, Textblock, Integer, Datei, Bild oder ein anderer Datentyp).

**[0067]** Die obigen Metadatenbeschreibungen sind nichteinschränkende Beispiele. Es können andere Metadateneigenschaften und -attribute verwendet werden.

Beispielhafter Prozess, der deskriptive Datenstrukturen benutzt

[0068] [Fig. 9](#) zeigt eine beispielhafte Anordnung zur Benutzung der in der gleichzeitig anhängigen verwandten US-Patentanmeldung, laufende Nr. 08/699,712 (oben zitiert) für deskriptive Datenstrukturen **200** beschriebenen Infrastruktur. Die in [Fig. 9](#) gezeigte Anordnung kann in einer Anzahl verschiedener Kontexte nützlich sein. Zum Beispiel möchte ein Anbieter **600** deskriptiver Datenstrukturen **200** möglicherweise wissen, welche deskriptiven Datenstrukturen **200** seine Kunden am besten mögen, so dass er die Qualität seiner Produkte verbessern kann. Oder ein Anbieter **600** kann Kunden für die Benutzung deskriptiver Datenstrukturen **200** pro Benutzung oder auf anderer Grundlage Gebühren berechnen. Bei einem weiteren Beispiel können bestimmte deskriptive Datenstrukturen **200** oder Klassen von DDS **200** darauf beschränkt werden, nur von autorisierten Benutzern oder Klassen autorisierter Benutzer benutzt zu werden.

[0069] [Fig. 9](#) zeigt einen DDS-Anbieter **600**, der eine DDS **200** und eine assoziierte Kontrollmenge **316** an einen Wertkettenteilnehmer **602** abliefern. Die Kontrollen **316** können Regeln und assoziierte Konsequenzen zur Kontrolle oder anderweitigen Beeinflussung der Benutzung oder anderer Aspekte dessen, was der Wertkettenteilnehmer **602** mit der DDS **200** anfangen kann, bereitstellen. Die Kontrollen **316** und die DDS **200** können in einen Container **100** verpackt werden. Der Wertkettenteilnehmer **602** kann den Container **100**, der die DDS **200** enthält, direkt von dem DDS-Anbieter **600** erhalten; als Alternative kann der Anbieter ihm ein Rechte- und Zulassungs-Clearinghouse **604** und einen Teilnehmer **602** geben und ihn von dem Clearinghouse (oder einer anderen Stelle) erhalten (siehe Container **100B**).

[0070] Der Wertkettenteilnehmer **602** kann mit der DDS **200** Inhalt **102** verfassen. Der Teilnehmer **602** kann den Inhalt **102** mit assoziierten Kontrollen **316A** in einen Container **100A** verpacken. Der Teilnehmer **600** kann gegebenenfalls die DDS **200** und assoziierte Kontrollen **316a**, **316b** mit dem Inhalt **102** in denselben Container aufnehmen – oder sich auf den Anbieter **600** und/oder das Rechts- und Zulassungs-Clearinghouse **604** verlassen, die DDS und ihre Kontrollen unabhängig zum Beispiel in einem anderen Container **100c** zu Endbenutzern **606** abzuliefern.

[0071] Die Endbenutzer **606(1)**, ..., **606(n)** benutzen die DDS **200** (gemäß den Kontrollen **316**) in Verbindung mit dem Inhalt **102** (zum Beispiel zum Lesen, Browsen und anderweitigem Zugreifen auf den Containerinhalt). Die Kontrollen **316**, **316A** können von Benutzergeräten fordern, einem Benutzungs-Clearinghouse **612** Benutzungsdaten **610** zuzuführen.

Das Benutzungs-Clearinghouse **612** kann Benutzungsdaten **610A** in Bezug auf den Zugriff auf die DDS **200** und/oder ihre Benutzung dem DDS-Anbieter **600** zuführen und kann unabhängig dem Wertkettenteilnehmer **602** Benutzungsdaten **610B** in Bezug auf Zugang zu dem Inhalt **102** und/oder seiner Benutzung zuführen.

Deskriptive Datenstrukturen können benutzt werden, um einen Grad an Interoperabilität zwischen Rechtsverwaltungsumgebungen zu erzielen

[0072] Gemäß der vorliegenden Erfindung bereitgestellte deskriptive Datenstrukturen **200** können einen Grad an Interoperabilität zwischen Quellen- und Zielrechteverwaltungsumgebungen und/oder eine Brücke zur Erzielung mindestens eines gewissen Grades an Interoperabilität zwischen einer Rechteverwaltungsumgebung und der Außenwelt bereitstellen.

[0073] Verschiedene Rechteverwaltungsumgebungen können wesentlich inkompatible Mechanismen zum Definieren von ein Objekt betreffenden Rechten aufweisen. Die deskriptiven Datenstrukturen **200** können mindestens eine teilweise Brücke zur Erzielung eines Grads an Kompatibilität und Interoperabilität bereitstellen. Zum Beispiel kann ein Anbieter, der ein Objekt innerhalb einer Quellenrechteverwaltungsumgebung definiert, eine deskriptive Datenstruktur zur Verwendung durch Prozesse innerhalb einer oder mehrerer Ziel-Rechteverwaltungsumgebungen erzeugen. Zum Beispiel kann ein Objekterzeuger oder ein anderer Anbieter innerhalb einer deskriptiven Datenstruktur **200** bestimmte Regeln, Integritätseinschränkungen und/oder andere Eigenschaften spezifizieren, die auf das Objekt angewandt werden können oder sollten, nachdem es in eine Ziel-Rechteverwaltungsumgebung importiert wurde. Die Ziel-Rechteverwaltungsumgebung kann wählen, solche Regeln, Einschränkungen und/oder Eigenschaften abhängig von dem Grad, zu dem sie der Quellenumgebung vertrauen kann, selektiv durchzusetzen. Zum Beispiel können aus einem X.12-Sicherheit verwendenden EDI-System importierte Objekte vertrauenswürdiger als Objekte sein, die aus Umgebungen mit geringerer (oder keiner) Sicherheit präsentiert werden.

[0074] Bei einem anderen Beispiel kann ein Anbieter, der ein Objekt außerhalb jeglicher Rechteverwaltungsumgebung erstellt, eine deskriptive Datenstruktur **200** erstellen, die zu verwenden ist, falls und wenn das Objekt in eine oder mehrere Rechteverwaltungsumgebungen importiert wird. Die Ziel-Rechteverwaltungsumgebung(en) kann bzw. können solche deskriptive Datenstruktur(en) benutzen, um bei dem effizienten Verstehen und Handhaben des Objekts zu helfen. Eine innerhalb einer Rechteverwaltungsumgebung erstellte deskriptive Datenstruktur kann ferner in eine oder mehrere Anwendungen außerhalb der

Rechteverwaltungsumgebung exportiert und zur Unterstützung der Anwendung(en) beim Interpretieren von exportiertem Inhalt oder anderen Informationen verwendet werden.

[0075] [Fig. 10A](#) zeigt ein Beispiel dafür, wie deskriptive Datenstrukturen **200** benutzt werden können, um Interoperabilität bereitzustellen. Bei dem Beispiel von [Fig. 10A](#) erstellt ein DDS-Erstellungswerkzeug **800** eine DDS **200**, die einen oder mehrere Zieldatenblöcke **801** enthält. In einem Beispiel kann das DDS-Erstellungswerkzeug **800** auf bestimmten oder allen der Fähigkeiten des Layout-Werkzeugs **300** basieren und/oder diese enthalten und kann zusätzlich zu mit dem Layout-Werkzeug **300** assoziierten Merkmalen Interoperabilitätsfähigkeiten bereitstellen. Bei einem anderen Beispiel enthält das DDS-Erstellungswerkzeug **800** möglicherweise keine der Fähigkeiten des Layout-Werkzeugs **300** und kann die DDS **200** ausschließlich für Interoperabilitätszwecke erstellen. Das DDS-Erstellungswerkzeug **800** kann zum Beispiel ein Anwendungsprogramm mit einer graphischen Benutzeroberfläche sein, ein Hintergrundprozess, der eine Benutzeroberfläche nur anzeigt, wenn sie gerade durch einen Benutzer konfiguriert wird, ein Teil eines Betriebssystems, ein Teil der Firmware eines Computers, ein Serverprozess, der unabhängig oder als Teil eines oder dem gesamten "Gateway" zwischen einem System und einem anderen wirken kann (z.B. ein öffentliches Netzwerk und ein privates Netzwerk, zwei oder mehr private Netzwerke, ein lokales Netzwerk und ein großflächiges Netzwerk usw.) oder eine beliebige andere wünschenswerte Implementierung oder Integration.

[0076] Der Zieldatenblock **801** kann Informationen bereitstellen, mit denen Interoperabilität mit einer bestimmten Zielumgebung **850** bereitgestellt wird. Eine einzelne DDS **200** kann in einem Beispiel Interoperabilität mit N verschiedenen Zielumgebungen **850** bereitstellen, indem N Zieldatenblöcke **801(1)**, ... **801(N)** vorgesehen werden, die jeweils einer verschiedenen Zielumgebung **850(1)**, ... **850(N)** entsprechen.

[0077] In diesem Beispiel enthält jeder Zieldatenblock **801** Regel-(Kontroll-)informationen. Verschiedene Zieldatenblöcke **801** können verschiedene Regelinformationen für verschiedene Zielumgebungen **850** bereitstellen. Die Regelinformationen können zum Beispiel Operationen (Ereignisse) und/oder Konsequenzen von Anwendungsprogrammfunktionen **856** in der assoziierten Zielumgebung **850** betreffen, wie zum Beispiel das Spezifizieren von:

- zulässigen und/oder vorgeschriebenen Operationen;
- Beschaffenheit und/oder Ausmaß von zulässigen und/oder vorgeschriebenen Operationen; und/oder
- Konsequenzen des Ausführens von zulässigen

und/oder vorgeschriebenen Operationen.

[0078] Der Zieldatenblock **801** kann außerdem gegebenenfalls zusätzliche Informationen enthalten, die einen DDS-Parser **852** und/oder einen Übersetzer **854** innerhalb einer entsprechenden Zielumgebung **850** Anleitung gibt.

[0079] [Fig. 10B](#) zeigt ein ausführliches Beispiel dafür, wie Zielinformationen innerhalb der DDS **200** organisiert werden können. In diesem Beispiel erstellt das DDS-Erstellungswerkzeug **800** einen DDS-Kopfteil **805**, der einen oder mehrere Zielaufzeichnungskopfteile **807** referenziert. Der DDS-Kopfteil **805** kann zum Beispiel ein Feld "Anzahl der Ziele" **809** enthalten, das die Anzahl der Zieldatenblöcke **801** in der DDS **200** angibt, ein Feld "Offset zum ersten Zieldatenteil" **811**, das den Ort des ersten Zieldatenblocks **801(1)** in der DDS **200** bereitstellt, ein Quellennachrichtenfeld **812A**, das die Quellenumgebung identifiziert, und ein optionales Verfassersiegel **812B**, mit dem die Integrität und Authentizität der DDS **200** verifiziert werden kann. Das Quellennachrichtenfeld **812A** (das optional sein kann) kann eine Quellen-ID enthalten, mit der beim Verifizieren der Quellenumgebung der DDS **200** geholfen werden kann, und ein optionales Quellensiegel (das in der Quellennachricht vorhanden sein kann oder nicht). Jeder Zieldatenblock **801** in der DDS **200** kann mit einem Zielaufzeichnungskopfteil **807** beginnen, der ein Feld "Ziel-ID" **813**, ein Feld "Länge" **815**, ein Feld "Offset zum nächsten Zieldatenteil" **817**, ein optionales Verfassersiegel **819** und eine optionale Quellennachricht **821** enthält. Das Feld "Ziel-ID" **813** kann eine eindeutige Identifikationsnummer oder einen Wert spezifizieren, die bzw. der dem assoziierten Zieldatenblock **801** entspricht und/oder die beabsichtigte Zielumgebung(en) identifiziert, das Feld "Länge" **815** kann die Länge des Zieldatenblocks **801** spezifizieren, und das Feld "Offset" **817** kann den Ort (relativ oder absolut) des nächsten Zieldatenblocks **801** in der DDS **200** angeben (und kann einen Nullwert für den letzten Zieldatenblock annehmen).

[0080] Die optionalen Verfassersiegel **812B**, **819** (und Quellensiegel) können kryptographische Siegel sein, die dabei helfen, sicherzustellen, dass die DDS **200** bzw. die Zielaufzeichnungen **801** seit ihrer Erstellung nicht verändert wurden, sowie die Identität von Verfasser und/oder Quelle der DDS **200**. Die optionalen Quellennachrichten **812C** und **821** können Informationen sein, die dabei helfen, sicherzustellen, dass eine Zielumgebung weiß, welche Quellenumgebung die DDS **200** erstellt hat.

[0081] Wieder mit Bezug auf [Fig. 10A](#) kann das DDS-Erstellungswerkzeug **800** beim Erstellen der DDS **200** diese und jeden Zieldatenblock **801** zu Integritätszwecken unter Verwendung geeigneter kryptographischer Prozesse kryptographisch versiegeln,

indem zum Beispiel zuerst eine kryptographische Hash-Funktion (z.B. SHA, MD5 usw.) an den Daten ausgeführt und dann der resultierende Hash-Wert unter Verwendung eines privaten Schlüssels des DDS-Verfassers, der mit einem asymmetrischen Kryptosystem (z.B. RSA, El Gamal, usw.) assoziiert ist, verschlüsselt wird. Wenn Versiegelung verwendet wird, sollte der DDS-Verfasser vorzugsweise sicherstellen, dass der mit dem Verschlüsselungs-Privatschlüssel assoziierte öffentliche Schlüssel zertifiziert ist (z.B. mit einem Privatschlüssel einer zertifizierenden Autorität verschlüsselt) und für die Verwendung durch Zielumgebungen zum Validieren des Siegels verfügbar ist (z.B. durch Aufnahme eines Zertifikats in die DDS **200**, Veröffentlichens des Zertifikats auf einem öffentlichen Netzwerk usw.).

**[0082]** Wenn die Quellennachrichten **812C**, **821** benutzt werden, sollten sie vorzugsweise durch die Quellenumgebung bereitgestellte Informationen repräsentieren, die einer Zielumgebung helfen können, die Quellenumgebung zu identifizieren, und die ferner auch dabei helfen können, sicherzustellen, dass die DDS **200** tatsächlich von der Quellenumgebung erstellt wurde (und der deshalb zum Beispiel so weit vertraut werden kann, wie die Quellenumgebung vertrauenswürdig ist). Zum Beispiel kann eine Quellenumgebung eine geschützte Verarbeitungsumgebung (PPE) der in der oben zitierten Patentschrift von Ginter et al. beschriebenen Form aufweisen. Bestimmten solcher PPE können kryptographische Schlüssel (z.B. ein privater Schlüssel eines Paares aus einem öffentlichen Schlüssel und privaten Schlüssel) verfügbar sein, mit denen man ein kryptographisches Hash verschlüsseln kann, das je nach Fall von dem DDS-Kopfteil **805** oder dem Zielblockkopfteil **807** genommen wurde. Bei einem solchen Beispiel müsste eine Zielumgebung einen entsprechenden kryptographischen Schlüssel (z.B. einen öffentlichen Schlüssel eines Paares aus einem öffentlichen Schlüssel und einem privaten Schlüssel) unter Verwendung vertrauenswürdiger Techniken (z.B. Ablieferung in einem von einer vertrauenswürdigen zertifizierenden Autorität signierten Zertifikat) beschaffen, um eine solche Quellennachricht zu evaluieren. Bei einem anderen Beispiel wurden möglicherweise kryptographische Schlüssel bei der Herstellung mit dem DDS-Erstellungswerkzeug **800** mitgeliefert, und es kann diese kryptographischen Schlüssel anstelle von Schlüsseln von einer PPE benutzen, obwohl diese Technik im Allgemeinen gegenüber Manipulation durch einen erfahrenen Computerhacker anfälliger wäre und deshalb für Zielumgebungen etwas weniger vertrauenswürdig sein könnte.

**[0083]** Zusätzlich oder als Alternative (zum Beispiel wenn kryptographische Techniken nicht angemessen oder gewünscht sind), kann die Quellennachricht eine eindeutige Kennung enthalten, die der Quellenumgebung entspricht.

**[0084]** Das DDS-Erstellungswerkzeug **800** (siehe [Fig. 10A](#)) kann dann die resultierende DDS **200** zusammen mit einem assoziierten Objekt **830** in eine sicheren Container **100** verpacken. Bei einem anderen Beispiel kann das DDS-Erstellungswerkzeug **800** die DDS **200** in ein Objekt **830'**, das ein Verfahren zum Freigeben der DDS an den Zielumgebungs-Parser **852** bereitstellt, einbetten oder die DDS anderweitig damit assoziieren. Die DDS **200** und ihr assoziiertes Objekt **830** können dann zur Verarbeitung an eine oder mehrere Zielumgebungen **850** abgeliefert werden.

**[0085]** Der Zielumgebungs-Parser **852** (und/oder Übersetzer **854**) kann zum Beispiel Teil eines Anwendungsprogramms, Teil eines Betriebssystems oder Teil eines Hilfsprogramms sein, das von einem Anwendungsprogramm und/oder einem Betriebssystem oder in Verbindung damit benutzt wird. Der Zielumgebungs-Parser **852** empfängt die DDS **200** und analysiert sie, um den Zieldatenblock **801(k)** zu finden, der der Zielumgebung **850(k)** entspricht. Der Parser **852** kann dann aus dem entsprechenden Zieldatenblock **801** die in dem Zieldatenblock enthaltenen Regeln bestimmen. Der Parser **852** versteht vorzugsweise genug über die Struktur der DDS **200**, um (z.B. unter Verwendung in [Fig. 10B](#) gezeigten Kopfteilinformationen) den entsprechenden Zieldatenblock **801**, der ihm entspricht, zu finden und außerdem die Regeln in dem Zieldatenblock zu verstehen. Der Zielumgebungs-Parser **852** muss keinerlei zusätzliche Regeln **316** verstehen, die in den Container **100** verpackt oder anderweitig mit dem Objekt **830** abgeliefert werden, er kann aber etwaige solche zusätzlichen Regeln gegebenenfalls benutzen (z.B. wenn er keinen Zieldatenblock **801** in der DDS **200** für die konkrete Zielumgebung **850** findet (zum Beispiel, wenn er in der Lage ist, einen bestimmten anderen Zieldatenblock **801** zu verstehen, dessen Regeln auf einer veröffentlichten Spezifikation und/oder einem veröffentlichten Standard basieren)).

**[0086]** Der Zielumgebungs-Parser **852** kann geltende Zielregeln aus dem Zieldatenblock **801** erhalten und diese Regeln den Anwendungsprogrammfunktionen **856** zuführen. Die Anwendungsprogrammfunktionen **856** können eine beliebige das Objekt **830** betreffende Operation definieren, wie zum Beispiel:

- Ausschneiden
- Kopieren
- Drucken
- Einfügen
- Abspeichern
- Ändern
- Löschen
- Eine beliebige andere Operation

**[0087]** Die von dem Parser **852** bereitgestellten Zielregeln können zum Beispiel für folgendes verwendet werden: Zulassen, Vorschreiben und/oder Verhin-



dem bestimmter Operationen; Definieren des Ausmaßes, zu dem bestimmte Operationen ausgeführt werden können (z.B. Begrenzen der Anzahl von Kopien, Definieren des Ausmaßes des Ausschneidens, Regeln, die zum Ausschneiden von Informationen bei nachfolgender Benutzung angewandt werden sollen usw.); und/oder Definieren der Konsequenzen des Ausführens einer bestimmten Operation (z.B. dem Benutzer Gebühren dafür berechnen, ein gesamtes Objekt **830** oder einen Teil davon auszudrucken oder anderweitig zu benutzen und/oder darauf zuzugreifen, Aufzeichnungen von Zeit und/oder Anzahl solcher ausgeführten Operationen führen usw.).

**[0088]** Der Parser **852** kann außerdem oder als Alternative bestimmte oder alle der von ihm aus dem Zieldatenblock **801** erhaltenen Regeln anderen Anordnungen zuführen, um die Regeln anzuwenden, wie zum Beispiel den Block "andere Rechteverwaltungsfunktionen" **858**. Der Block **858** kann eine beliebige Art von Rechteverwaltungsfunktionen bereitstellen. Der Übersetzer **854** kann gegebenenfalls benutzt werden, um es den Anwendungsprogramm-funktionen **856** und/oder dem Block "andere Rechteverwaltung" **858** zu erlauben, die Regeln zu verstehen. Als ein Beispiel könnte man mit dem Übersetzer **854** die aus dem Zieldatenblock **801** erhaltenen Regelinformationen weiter ausführen, parametrisieren und/oder sichern, damit sie mehr oder völlig kompatibel mit dem Block "andere Rechteverwaltungsfunktionen" **858** sind.

**[0089]** Es wurde eine nützliche Datenstrukturdefinitions-methode und -anordnung in Verbindung mit ihren praktischsten und zur Zeit bevorzugten Ausführungsbeispielen beschrieben.

**[0090]** Bei einer bevorzugten Ausführungsform des Datenverarbeitungsverfahrens umfasst der Schritt des Benutzens den Schritt des Formatierens der Anzeige mindestens eines Teils mindestens einer Rechteverwaltungsdatenstruktur. Vorzugsweise umfasst der Schritt des Benutzens den Schritt des Formatierens des Lesens mindestens eines Teils mindestens einer Rechteverwaltungsdatenstruktur mindestens teilweise gemäß der deskriptiven Datenstruktur. Gemäß einer vorteilhaften Ausführungsform umfasst der Schritt des Benutzens den Schritt des mindestens teilweisen Anzeigens mindestens einer Rechteverwaltungsdatenstruktur mindestens teilweise auf der Basis der deskriptiven Datenstruktur.

**[0091]** Der Erstellungsschritt umfasst vorzugsweise den Schritt des Bereitstellens von Metadaten innerhalb der deskriptiven Datenstruktur, und der Anzeigeschritt umfasst das Anzeigen mindestens bestimmter Informationen aus der Rechteverwaltungsdatenstruktur mindestens teilweise gemäß den Metadaten. Bei einer weiteren bevorzugten Ausführungsform umfasst der Benutzungsschritt den Schritt des dyna-

mischen Erzeugens einer Benutzeroberfläche mindestens teilweise auf der Basis der deskriptiven Datenstruktur. Der Schritt des Benutzens umfasst vorzugsweise den Schritt des automatischen Identifizierens und/oder Findens mindestens eines Datenfelds mindestens teilweise auf der Basis der deskriptiven Datenstruktur.

**[0092]** Es ist vorteilhaft, dass der Benutzungsschritt den Schritt des automatischen Extrahierens von Daten innerhalb der Rechteverwaltungsdatenstruktur mindestens teilweise auf der Basis der deskriptiven Datenstruktur umfasst. Vorzugsweise umfasst der Erstellungsschritt das Erstellen einer deskriptiven Datenstruktur, die von jeglicher konkreter Rechteverwaltungsdatenstruktur unabhängig ist, aber abstrakt eine Klasse von Rechteverwaltungsdatenstrukturen beschreibt. Vorteilhafterweise umfasst der Erstellungsschritt den Schritt des Erstellens von Metadaten zum Definieren mindestens einer Eigenschaft des Benutzungsschrittes. Gemäß einer bevorzugten Ausführungsform umfasst der Erstellungsschritt den Schritt des Erstellens der abstrakten Repräsentation mindestens teilweise durch Verwendung eines Wizards, wobei die Funktionsweise des Wizards mindestens teilweise durch eine weitere deskriptive Datenstruktur definiert wird.

**[0093]** Der Benutzungsschritt umfasst vorzugsweise den Schritt des Veränderns des Verhaltens eines polymorphen Prozesses mindestens teilweise auf der Basis der deskriptiven Datenstruktur. Gemäß einer Ausführungsform umfasst der Benutzungsschritt den Schritt des Interpretierens mindestens eines Teils der deskriptiven Datenstruktur in der Laufzeit. Vorteilhafterweise umfasst der Benutzungsschritt den Schritt des dynamischen Anpassens mindestens eines Teils der Datenverarbeitung der Rechteverwaltungsdatenstruktur in der Laufzeit. Der Benutzungsschritt umfasst vorzugsweise das mindestens teilweise Benutzen der deskriptiven Datenstruktur als Anweisungen zum Steuern eines automatisierten digitalen Inhalts-Handlers. Gemäß einer bevorzugten Ausführungsform umfasst der Erstellungsschritt den Schritt des Erstellens mindestens einer Integritäts-einschränkung, und der Benutzungsschritt umfaßt den Schritt des Durchsetzens der Integritätseinschränkung.

**[0094]** Gemäß einer bevorzugten Ausführungsform des Verfahrens zum Erstellen und Benutzen eines sicheren Containers entspricht die deskriptive Datenstruktur einer atomischen Transaktion und umfasst ferner den Schritt des Ausführens der atomischen Transaktion mindestens einer der mehreren elektronischen Geräte mindestens teilweise gemäß der deskriptiven Datenstruktur. Vorzugsweise umfasst das verfahren ferner den Schritt des unabhängigen Benutzens und/oder Bereitstellens von Kontrollen in Bezug auf die deskriptive Datenstruktur. Vorteilhafter-

weise umfasst das Verfahren ferner den Schritt des Definierens mindestens einer Klasse deskriptiver Datenstrukturen auf der Basis mindestens eines Parameters.

### Patentansprüche

1. Verfahren zur Verwendung einer deskriptiven Datenstruktur in einem elektronischen Gerät, mit den folgenden Schritten:

Empfangen eines ersten sicheren Containers, wobei der erste sichere Container mindestens Inhalt **(102)** und assoziierte Kontrollen **(316A)** umfasst; Empfangen eines zweiten sicheren Containers, wobei der zweite sichere Container mindestens Folgendes umfasst: eine deskriptive Datenstruktur **(200)**, die Informationen enthält, die mindestens teilweise ein Format des Inhalts des ersten sicheren Containers beschreiben oder repräsentieren; und mindestens eine Regel **(316)**, die mindestens teilweise dafür ausgelegt ist, mindestens eine Benutzung der deskriptiven Datenstruktur **(200)** oder den Zugriff darauf zu kontrollieren;

Benutzen der Regel **(316)** des zweiten sicheren Containers, um Zugriff auf mindestens einen Teil der deskriptiven Datenstruktur **(200)** zu erhalten; und Benutzen des Teils der deskriptiven Datenstruktur im Verlauf mindestens eines Gebrauchs des Inhalts **(102)** des ersten sicheren Containers.

2. Verfahren nach Anspruch 1, bei dem: die Benutzung des Teils der deskriptiven Datenstruktur umfasst, Informationen aus der deskriptiven Datenstruktur bezüglich des Formats des Inhalts des ersten sicheren Containers zu benutzen.

3. Verfahren nach Anspruch 2, bei dem: die Benutzung des Teils der deskriptiven Datenstruktur ferner umfasst, die Formatinformationen zu benutzen, um einen spezifischen Teil des Inhalts des ersten sicheren Containers zu identifizieren.

4. Verfahren nach Anspruch 3, bei dem: der spezifische Teil des Inhalts des ersten sicheren Containers Informationen enthält, die mindestens einen zusätzlichen Teil des Inhalts des ersten sicheren Containers identifizieren oder beschreiben; und der Benutzung des Teils der deskriptiven Datenstruktur das Anzeigen der Identifikations- oder Beschreibungsinformationen folgt.

5. Verfahren nach Anspruch 4, bei dem: die Identifikations- oder Beschreibungsinformationen des Inhalts des ersten sicheren Containers einen Titel des zusätzlichen Teils des Inhalts des ersten sicheren Containers umfassen.

6. Verfahren nach Anspruch 4 oder 5, bei dem: die Identifikations- oder Beschreibungsinformationen des Inhalts des ersten sicheren Containers eine Zu-

sammenfassung des zusätzlichen Teils des Inhalts des ersten sicheren Containers umfassen.

7. Verfahren nach Anspruch 3, bei dem: das elektronische Gerät ferner ein erstes Computerprogramm enthält, das einen Interpreter für deskriptive Datenstrukturen umfasst; und der Schritt des Benutzens des Teils der deskriptiven Daten umfasst, dass das erste Computerprogramm diese Benutzung durchführt.

8. Verfahren nach Anspruch 7, bei dem: das erste Computerprogramm einen Browser umfasst und der Schritt des Benutzens des Teils der deskriptiven Daten ferner Folgendes umfasst: der Browser benutzt die deskriptive Datenstruktur, um einen Teil des Inhalts des ersten sicheren Containers zu identifizieren und zu finden; und der Browser bewirkt das Anzeigen des gefundenen Teils des Inhalts des ersten sicheren Containers.

9. Verfahren nach Anspruch 8, bei dem: der Schritt, in dem der Browser die deskriptive Datenstruktur benutzt, um einen Teil des Inhalts des ersten sicheren Containers zu identifizieren und zu finden, ferner umfasst, dass der Browser eine Elementkennung aus der deskriptiven Datenstruktur empfängt, wobei die Elementkennung den Teil des Inhalts des ersten sicheren Containers identifiziert; und der Schritt, in dem der Browser das Anzeigen des gefundenen Teils des Inhalts des ersten sicheren Containers bewirkt, ferner umfasst, dass der Browser unter Verwendung der Elementkennung auf den Teil des Inhalts des ersten sicheren Containers zugreift.

10. Verfahren nach Anspruch 9, ferner umfassend: nach der Identifikation, Benutzen mindestens einer Regel aus den Regeln des ersten sicheren Containers, um auf den identifizierten Teil des Inhalts des ersten sicheren Containers zuzugreifen.

11. Verfahren nach Anspruch 10, ferner umfassend: an dem Kommunikationsport, Empfangen eines dritten sicheren Containers, der Folgendes enthält: (1) mit der deskriptiven Datenstruktur in Beziehung stehende Metadaten; und (2) mindestens eine Regel, die dafür ausgelegt ist, mindestens teilweise eine Benutzung der Metadaten oder den Zugriff darauf zu regeln; und wobei der Schritt des Benutzens der deskriptiven Datenstruktur ferner Folgendes umfasst: in der deskriptiven Datenstruktur, Zugreifen auf eine Referenz auf die Metadaten; Benutzen mindestens einer Regel aus dem dritten Container, um auf mindestens einen Teil der Metadaten zuzugreifen; und Benutzen der Metadaten im Verlauf des Benutzens

der deskriptiven Datenstruktur in Verbindung mit dem Gebrauch der Inhalte des dritten sicheren Containers.

12. Verfahren nach Anspruch 11, bei dem: der Schritt des Benutzens der Metadaten umfasst, in den Metadaten enthaltene Informationen zu benutzen, um mindestens teilweise zu bestimmen, ob mindestens ein Teil des Inhalts des ersten sicheren Containers einem Benutzer angezeigt werden soll.

13. Verfahren nach Anspruch 12, wobei das Benutzen der Metadaten Folgendes umfasst: die Metadaten enthalten Informationen, die spezifizieren, dass spezifizierte Informationen angezeigt werden müssen, wenn bestimmte oder alle der Inhalte des ersten sicheren Containers angezeigt werden; und der Schritt des Benutzens der Metadaten umfasst, die spezifizierten Informationen anzuzeigen.

14. Verfahren nach Anspruch 13, bei dem: die spezifizierten Informationen Informationen umfassen, die mindestens einen Eigentümer oder Verfasser von mindestens einem Teil des Inhalts des ersten sicheren Containers identifizieren.

15. Verfahren nach Anspruch 1, bei dem: der erste sichere Container und der zweite sichere Container zu verschiedenen Zeiten in der ersten Datenverarbeitungsanordnung empfangen werden.

16. Verfahren nach Anspruch 1, bei dem: der erste sichere Container aus einem zweiten elektronischen Gerät empfangen wird; und der zweite sichere Container aus einem dritten elektronischen Gerät empfangen wird; wobei sich das zweite elektronische Gerät und das dritte elektronische Gerät an Orten befinden, die voneinander getrennt sind und die jeweils von dem Ort, an dem sich das erste elektronische Gerät befindet, getrennt sind.

17. Verfahren nach Anspruch 11, bei dem: der erste sichere Container aus einem zweiten elektronischen Gerät empfangen wird; und der zweite sichere Container und der dritte sichere Container aus einem dritten elektronischen Gerät empfangen werden; wobei sich das zweite elektronische Gerät und das dritte elektronische Gerät an Orten befinden, die voneinander getrennt sind und die jeweils von dem Ort, an dem sich das erste elektronische Gerät befindet, getrennt sind.

18. Verfahren nach einem der vorhergehenden Ansprüche, bei dem: die Kontrolle ferner umfasst, mindestens teilweise mindestens einen Aspekt eines Auditierungsprozesses zu kontrollieren.

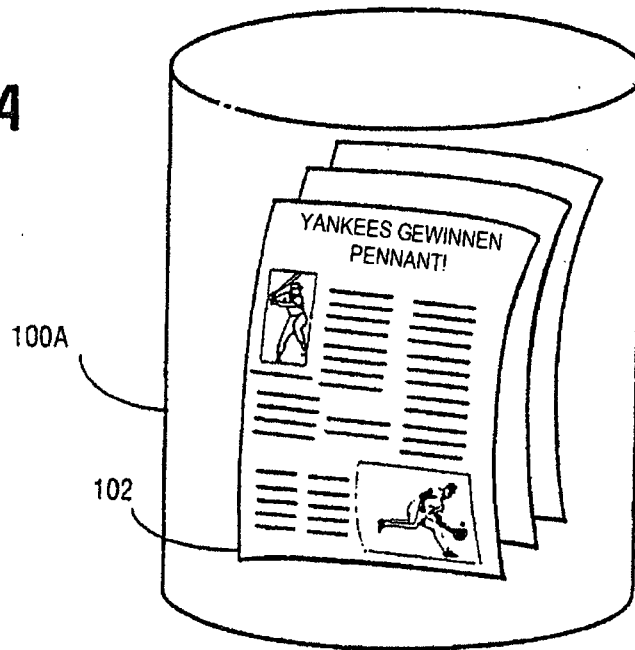
19. Verfahren nach einem der vorhergehenden Ansprüche, bei dem: die Kontrolle ferner umfasst, mindestens teilweise mindestens einen Aspekt eines Budgetierungsprozesses zu kontrollieren.

20. Verfahren nach Anspruch 1, bei dem das elektronische Gerät ein sicheres elektronisches Gerät ist.

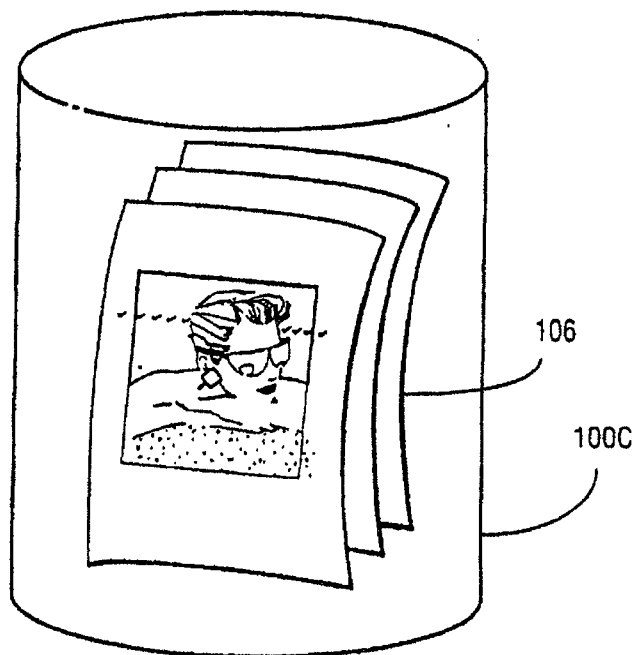
21. Elektronisches Gerät zur Verwendung mit einer deskriptiven Datenstruktur, wobei das elektronische Gerät Folgendes umfasst: Mittel zum Empfangen eines ersten sicheren Containers, wobei der erste sichere Container mindestens Inhalt **(102)** und assoziierte Kontrollen **(316A)** umfasst; Mittel zum Empfangen eines zweiten sicheren Containers, wobei der zweite sichere Container mindestens Folgendes umfasst: eine deskriptive Datenstruktur **(200)**, die Informationen enthält, die mindestens teilweise ein Format des Inhalts des ersten sicheren Containers beschreiben oder repräsentieren; und mindestens eine Regel **(316)**, die mindestens teilweise dafür ausgelegt ist, mindestens eine Benutzung der deskriptiven Datenstruktur **(200)** oder den Zugriff darauf zu kontrollieren; Mittel zum Benutzen der Regel **(316)** des zweiten sicheren Containers, um Zugriff auf mindestens einen Teil der deskriptiven Datenstruktur **(200)** zu erhalten; und Mittel zum Benutzen des Teils der deskriptiven Datenstruktur im Verlauf mindestens eines Gebrauchs des Inhalts **(102)** des ersten sicheren Containers.

Es folgen 14 Blatt Zeichnungen

**Fig. 1A**

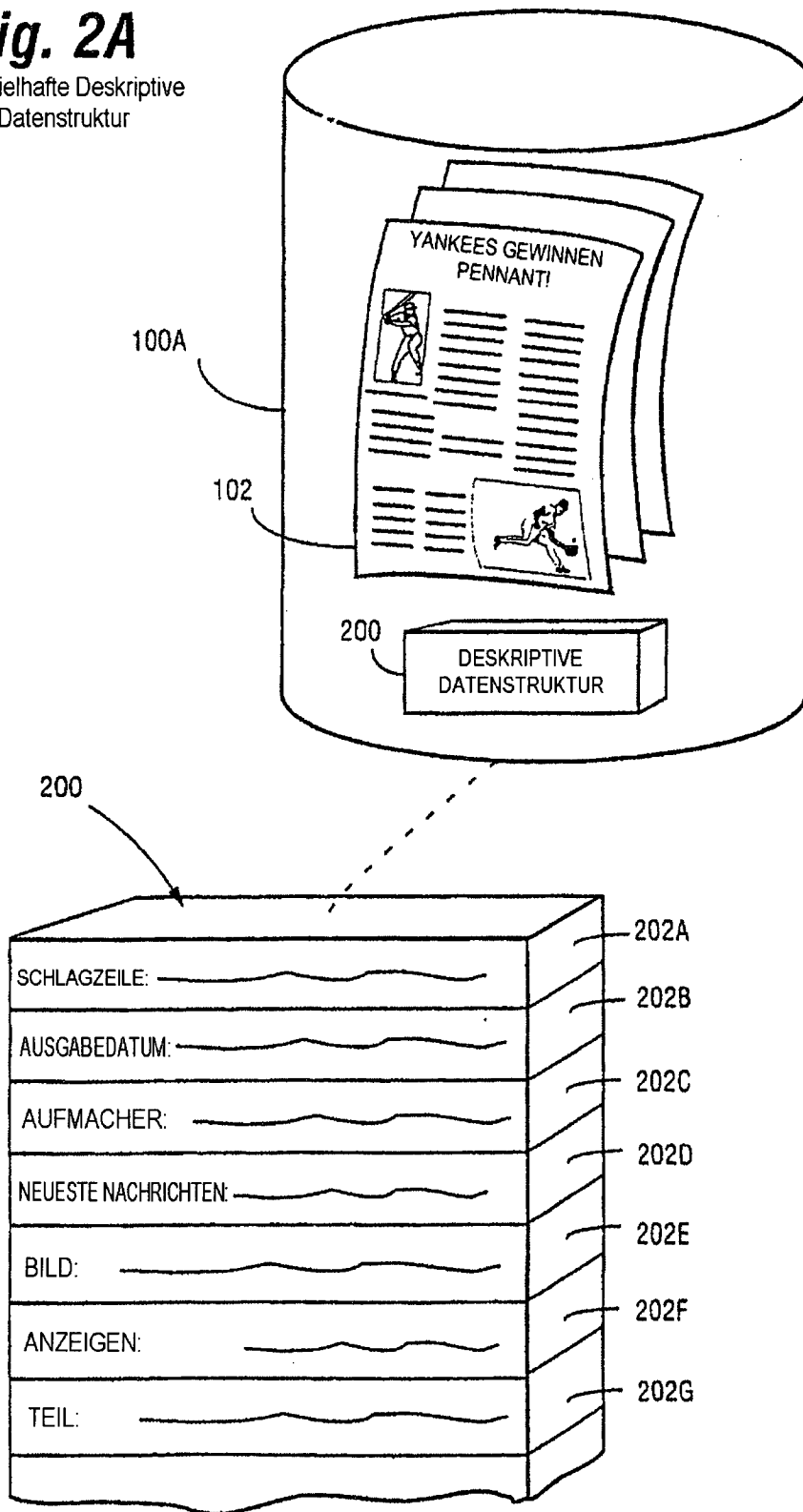


**Fig. 1B**

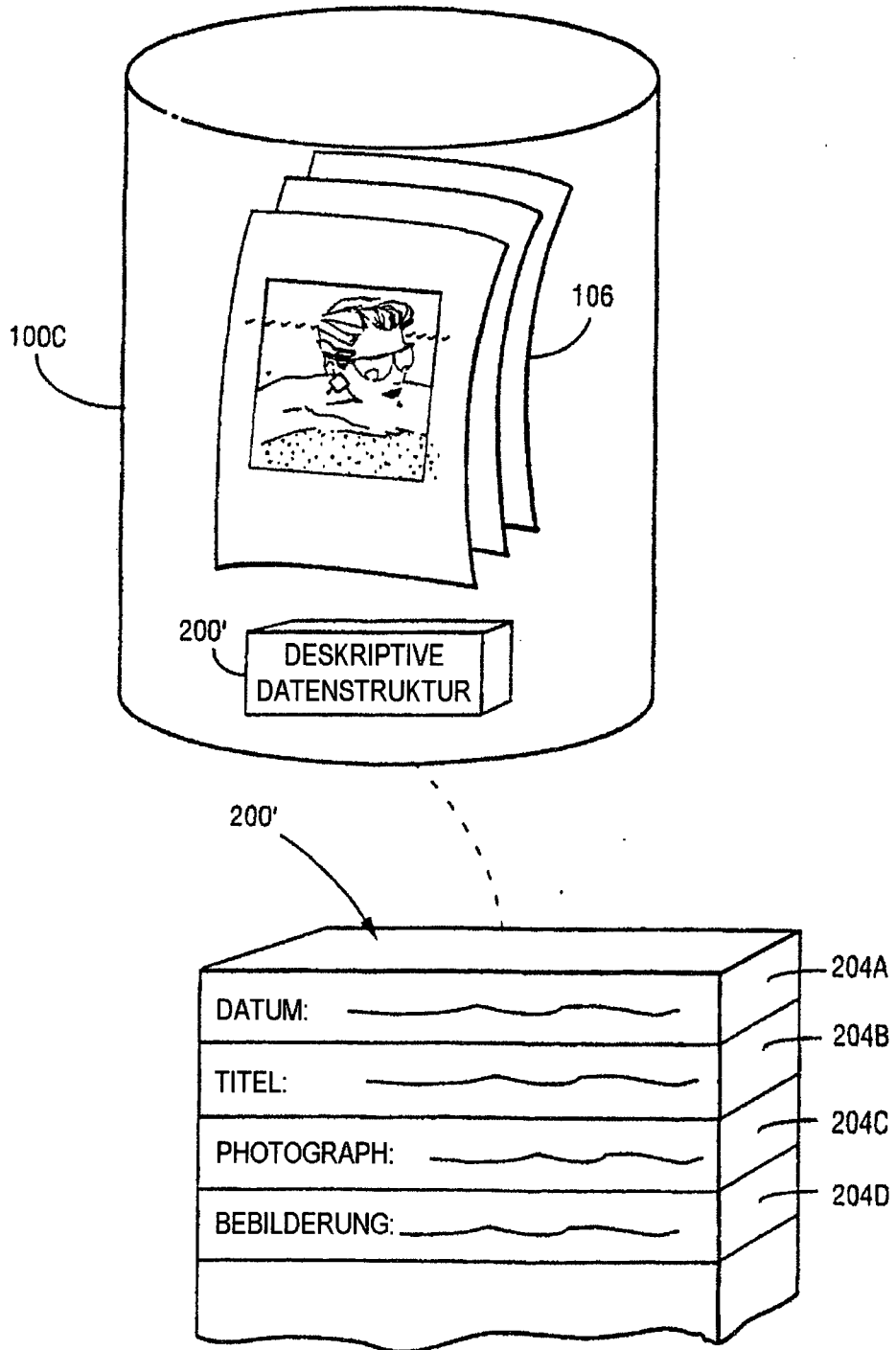


# Fig. 2A

Beispielhafte Deskriptive  
Datenstruktur

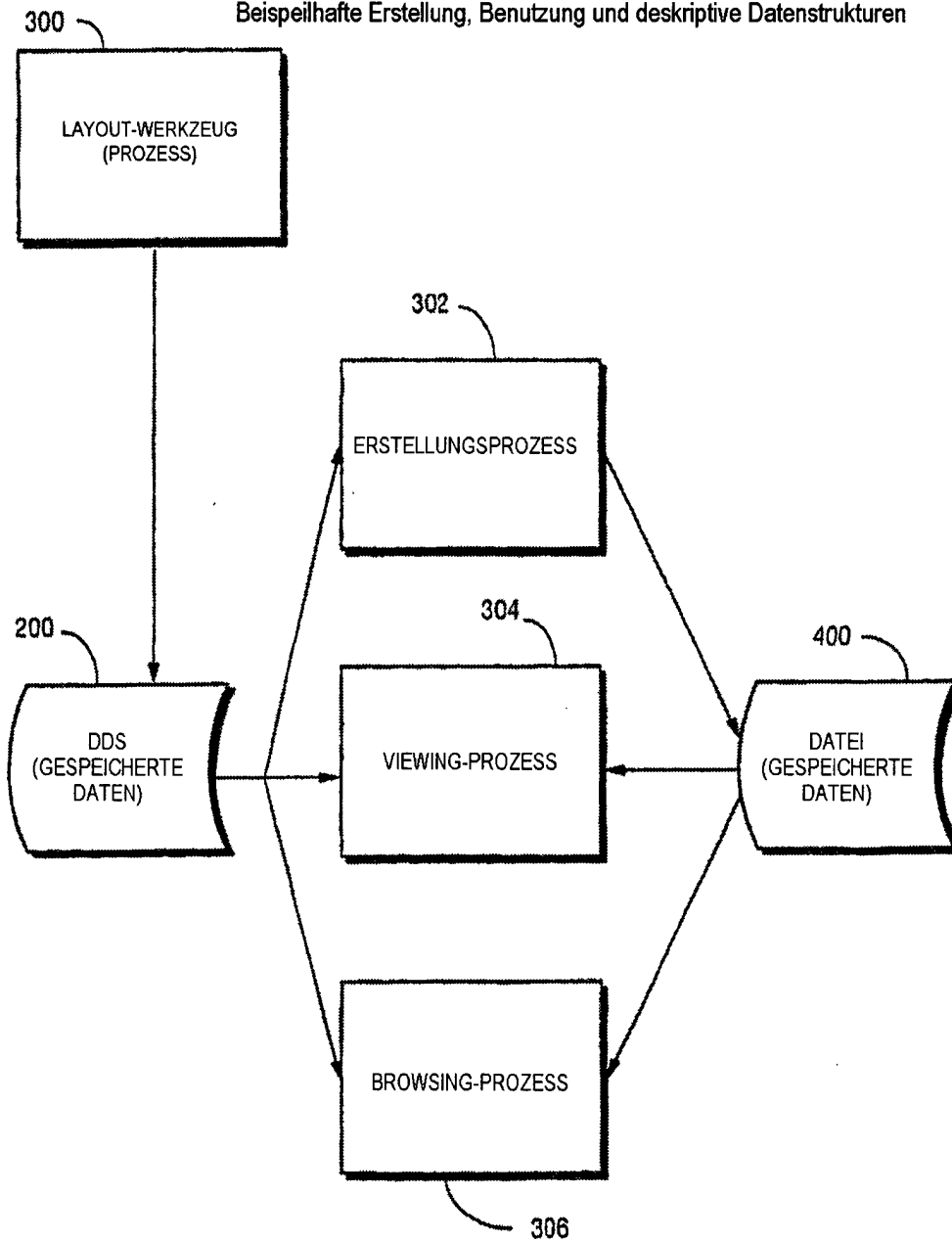


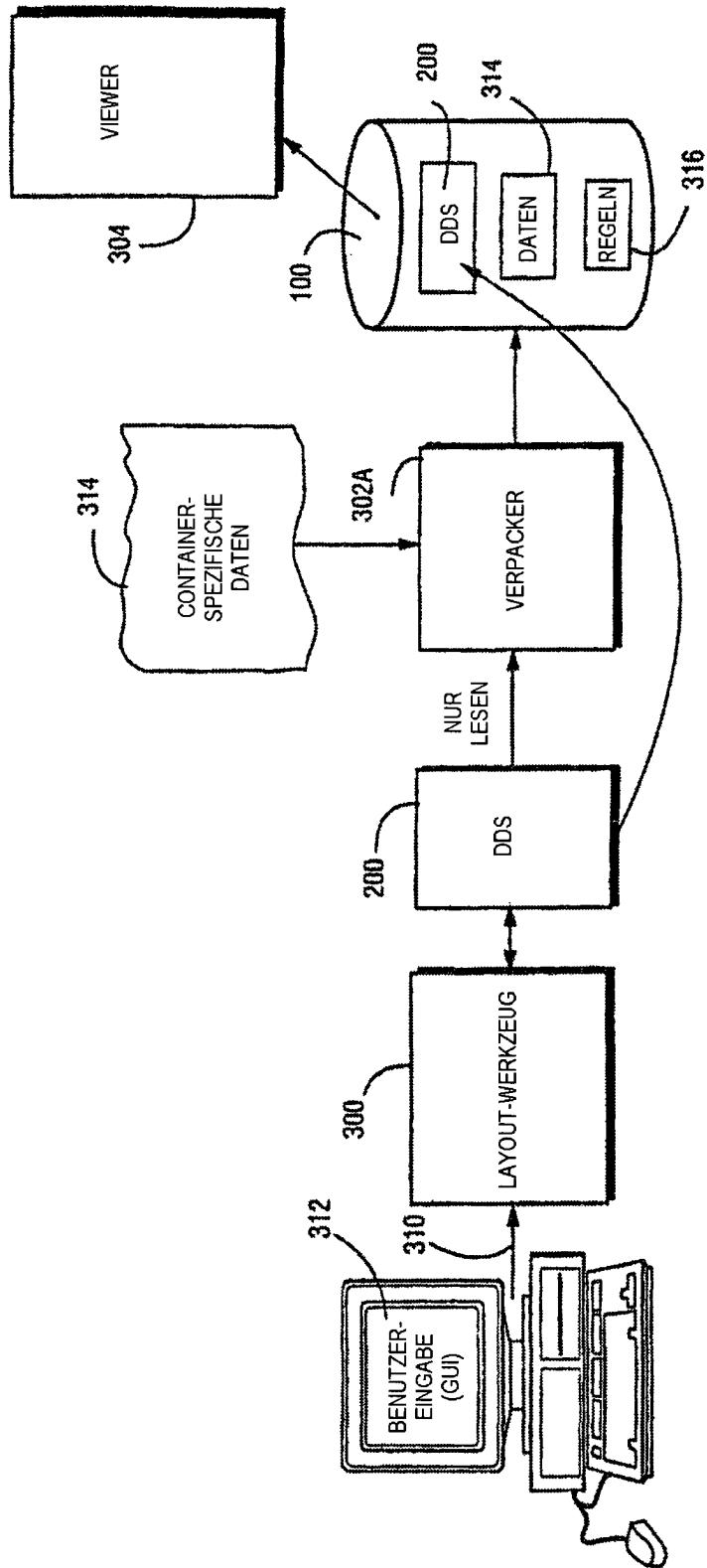
**Fig. 2B** Beispielhafte Deskriptive Datenstruktur



**Fig. 3**

Beispielhafte Erstellung, Benutzung und deskriptive Datenstrukturen

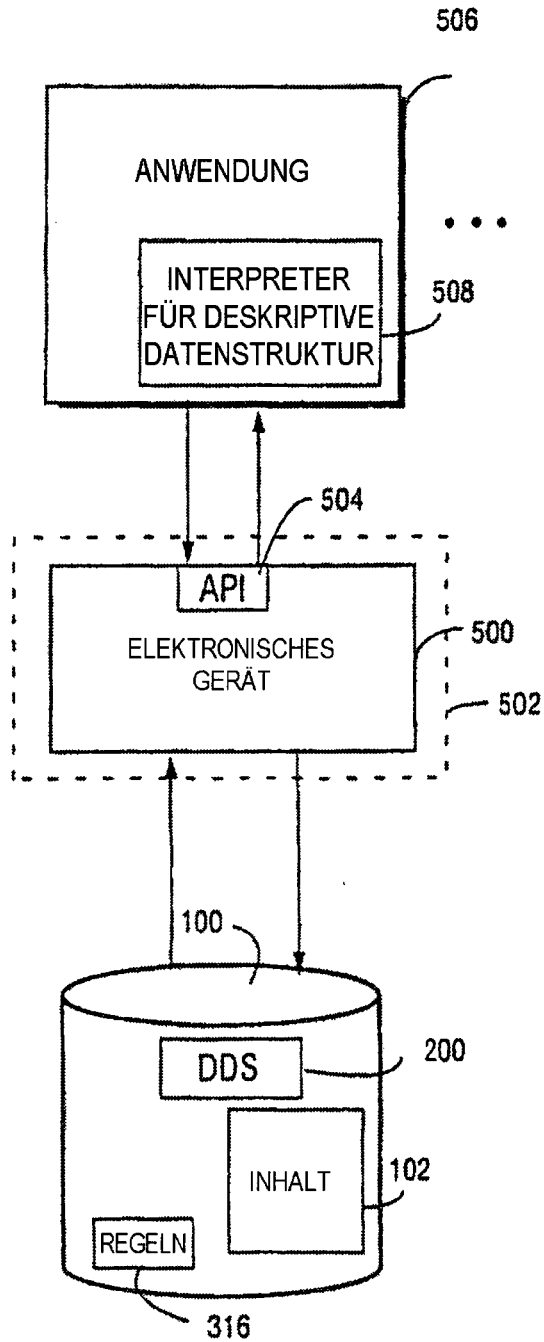




**Fig. 4** Beispielhafte Vorlagenerstellung und -benutzung

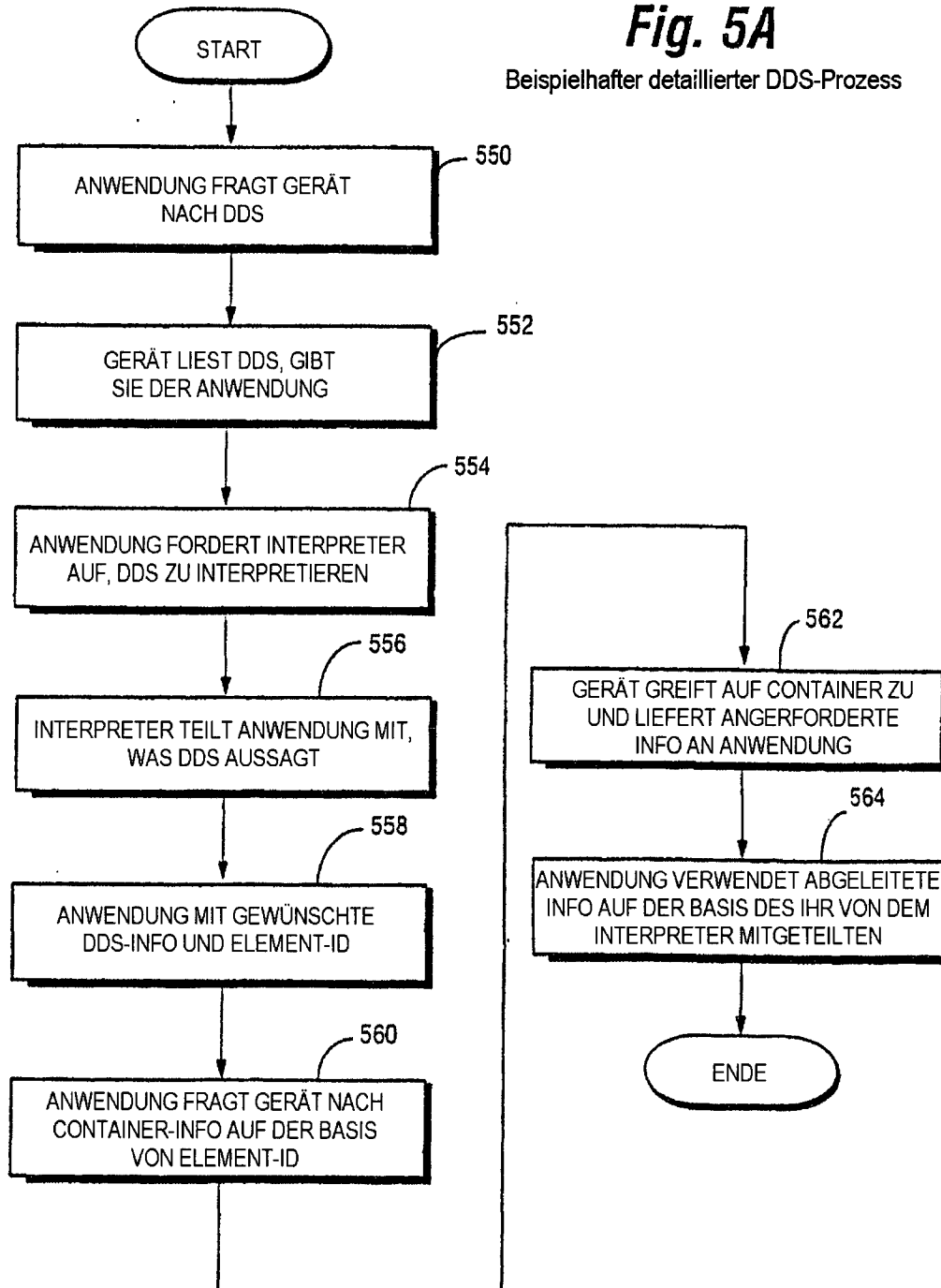


**Fig. 5** Beispielhafte gesicherte Systemarchitektur

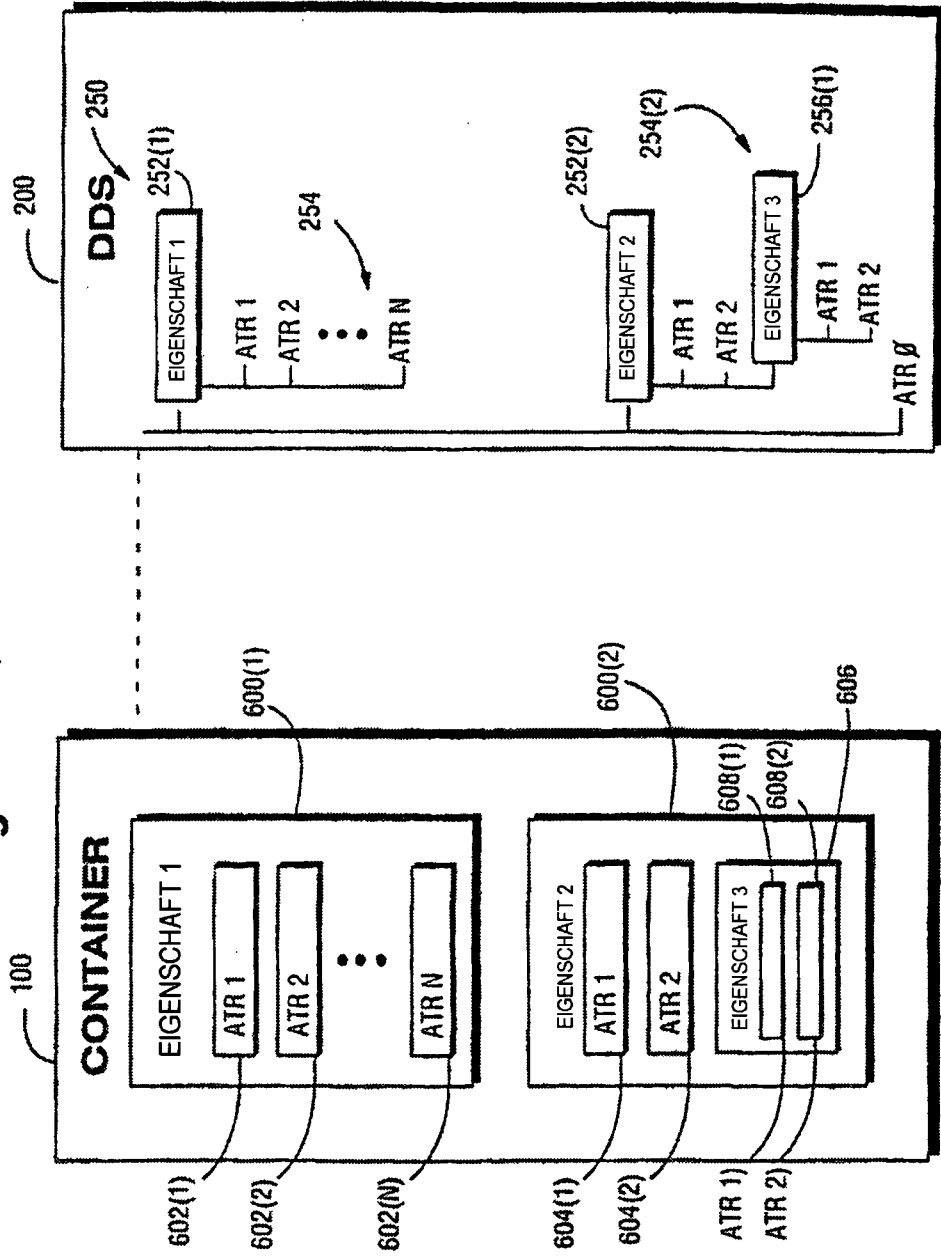


**Fig. 5A**

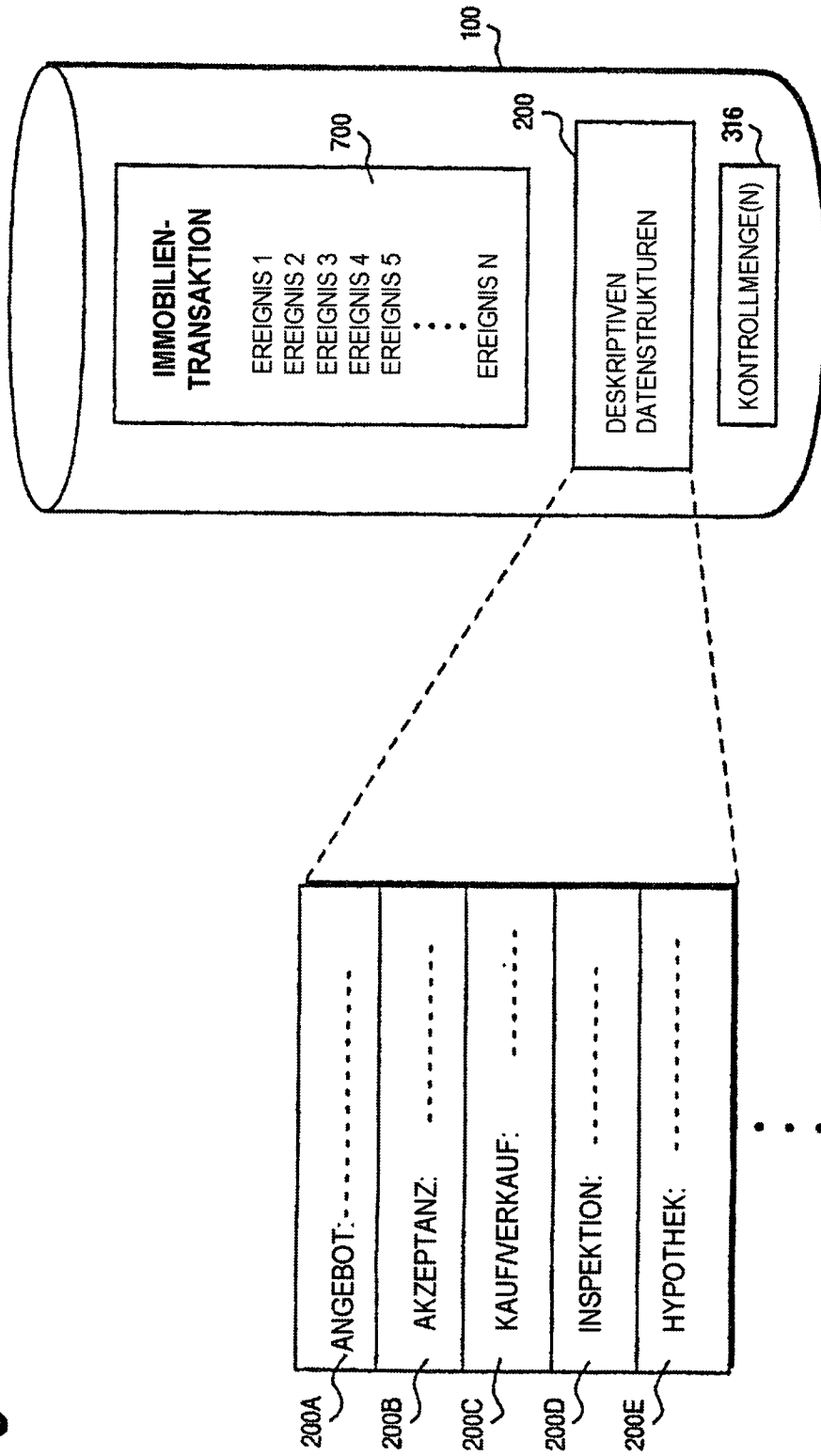
Beispielhafter detaillierter DDS-Prozess



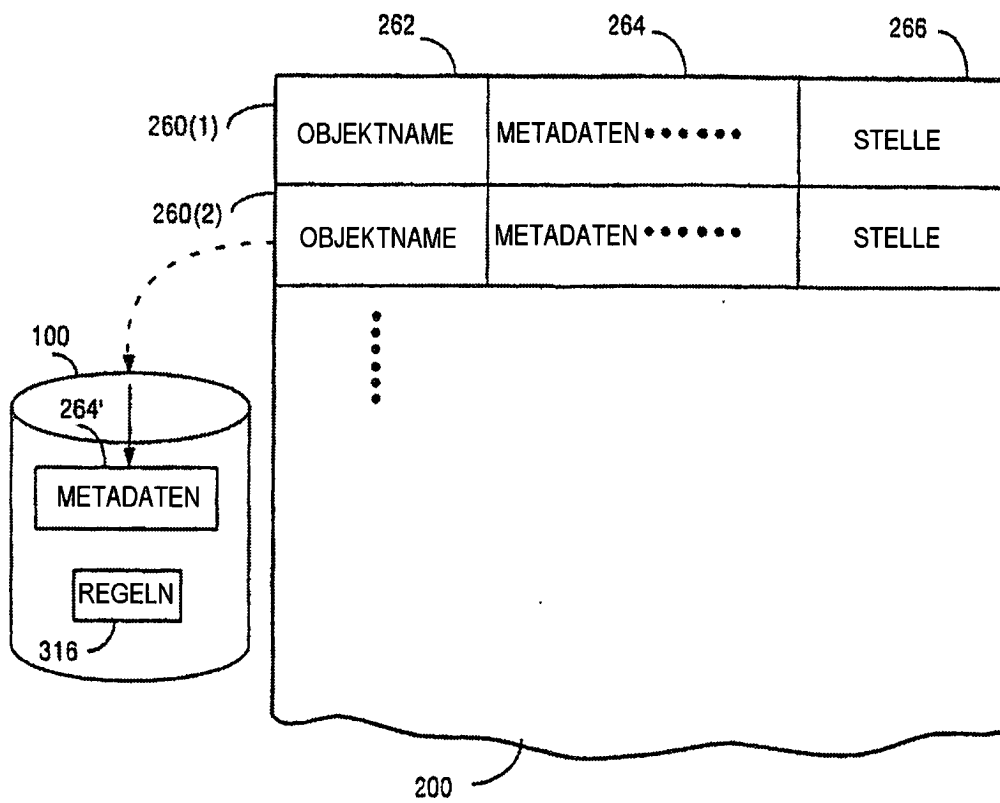
**Fig. 6** Beispielhafter hierarchische DDS-Struktur



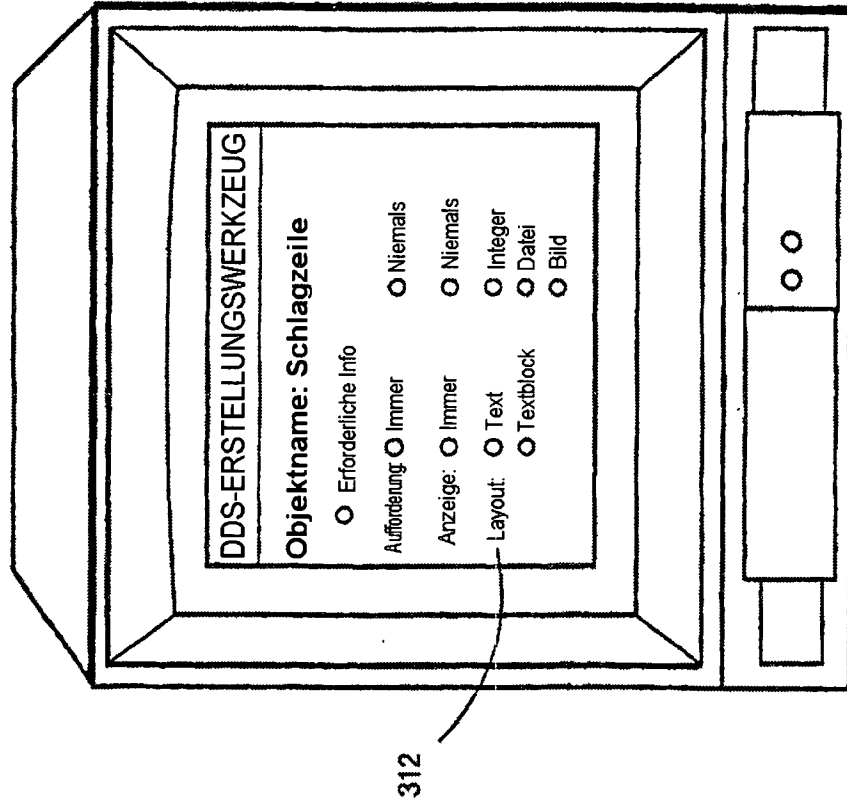
**Fig.6A** DDS unterstützt atomische Ereignisse



**Fig. 7** Beispielhaftes Format einer deskriptiven Datenstruktur



**Fig. 8** Beispielhafte graphische Oberfläche zur DDS-Erstellung



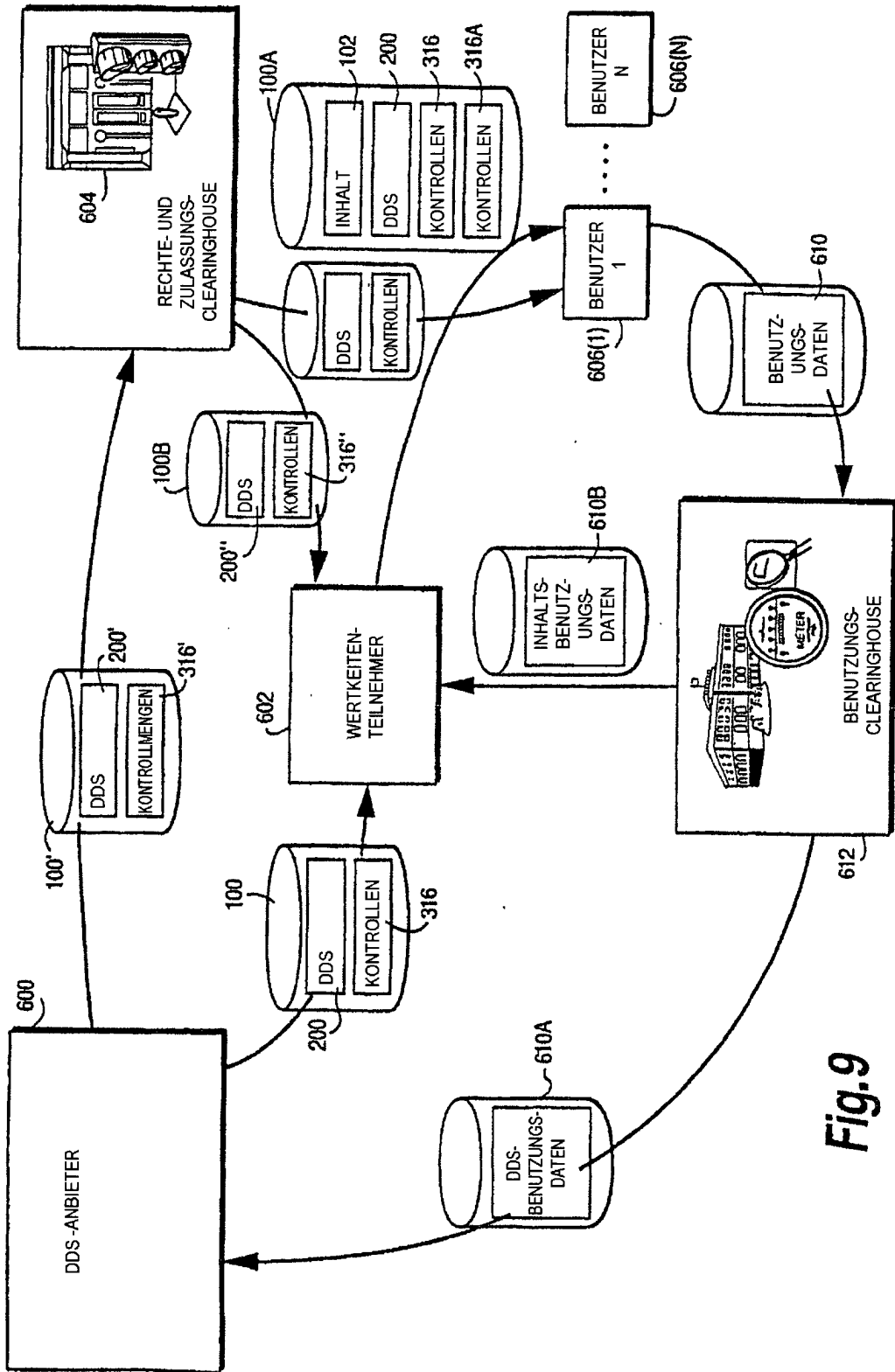
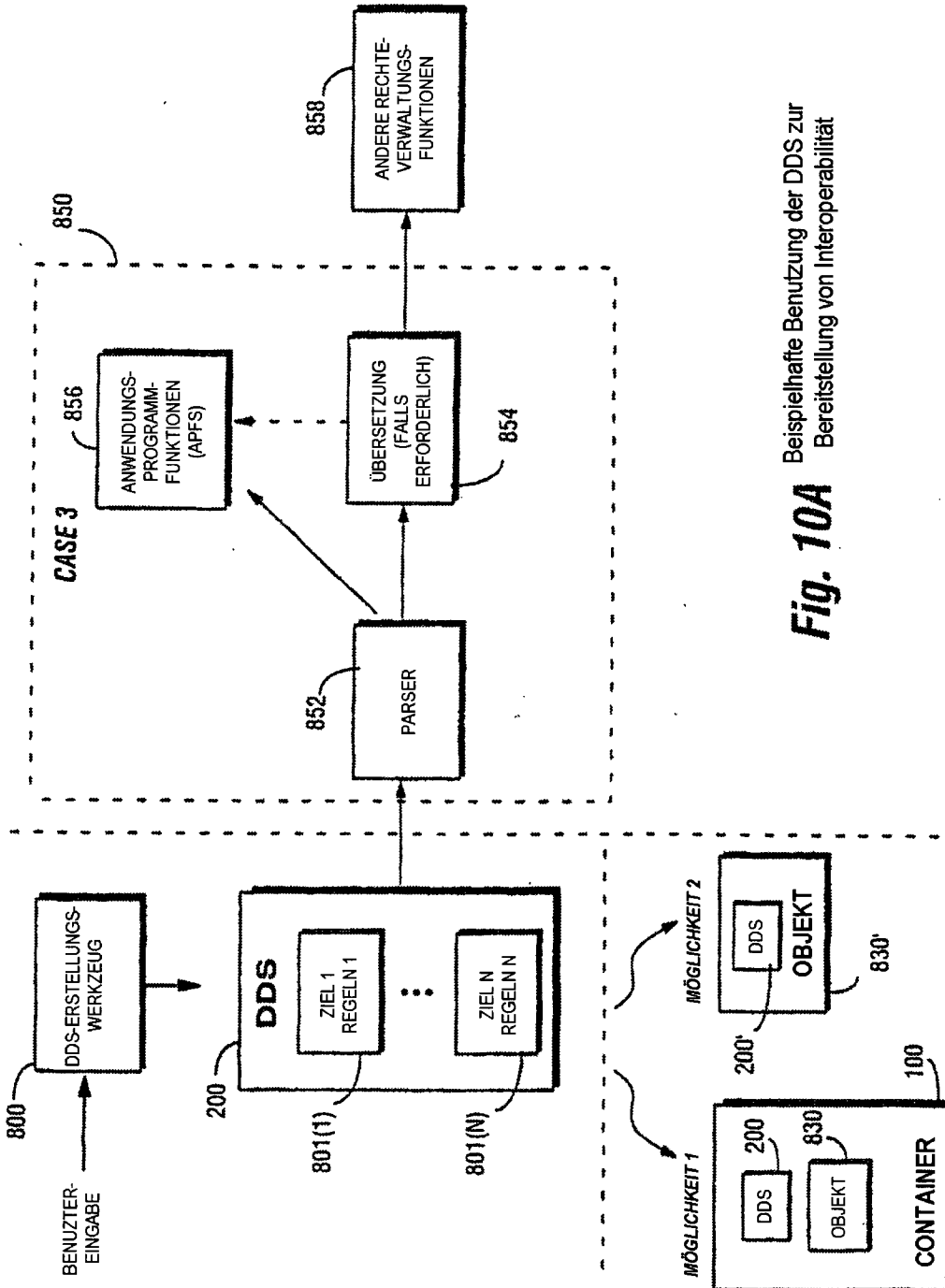
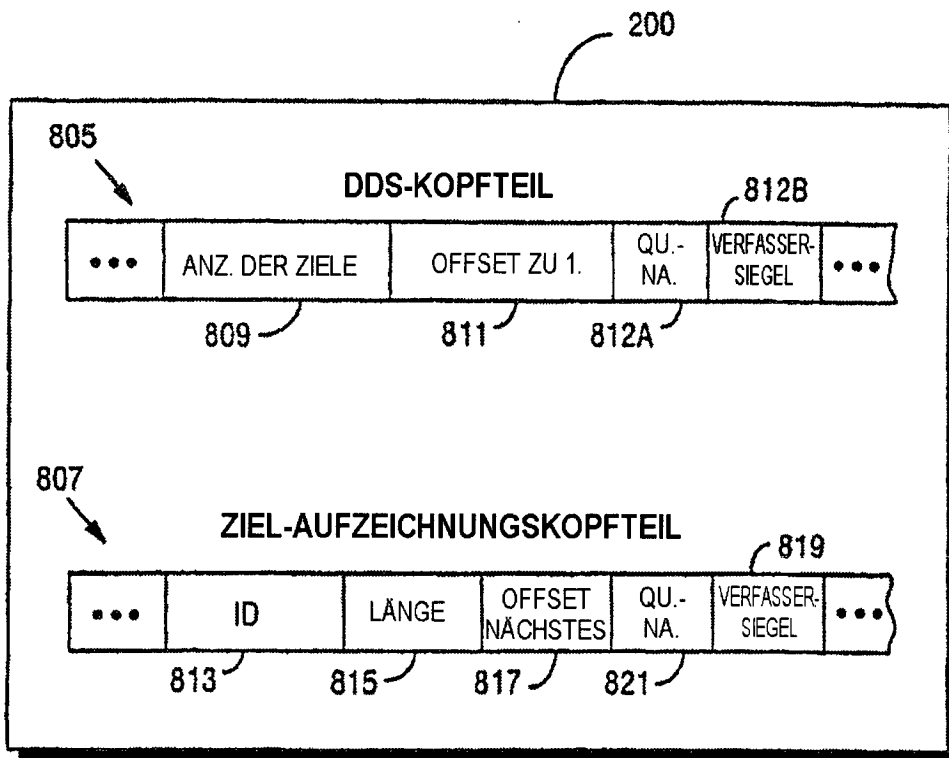


Fig.9



**Fig. 10A** Beispielhafte Benutzung der DDS zur Bereitstellung von Interoperabilität





**Fig. 10B** Beispielhafte DDS-Struktur