



República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(21) PI 0015347-8 A2



(22) Data do Depósito: 03/11/2000

(43) Data da Publicação Nacional: 29/09/2020

(54) **Título:** RECURSO DE ARMAZENAMENTO DE INTERNET COMPARTILHADA, SISTEMA DE INTERFACE COM O USUÁRIO, E MÉTODO

(51) **Int. Cl.:** G06F 17/30.

(71) **Depositante(es):** XDRIVE, LLC.

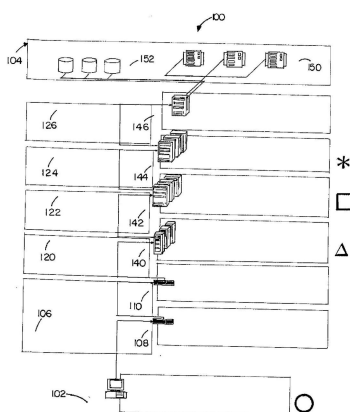
(72) **Inventor(es):** BRETT O' BRIEN; SEAN WHITELEY; LUCAS MCGREGOR; MARTIN HALD.

(86) **Pedido PCT:** PCT US2000030536 de 03/11/2000

(87) **Publicação PCT:** WO 2001/033381 de 10/05/2001

(85) **Data da Fase Nacional:** 06/05/2002

(57) **Resumo:** RECURSO DE ARMAZENAMENTO DE INTERNET COMPARTILHADA, SISTEMA DE INTERFACE COM O USUÁRIO, E MÉTODO Um recurso de armazenamento de Internet compartilhada provê armazenamento de arquivo baseado em Internet, recuperação, acesso, controle, e manipulação para o usuário. Adicionalmente, uma interface do usuário fácil de usar é provida tanto para um browser ou aplicação de parada sozinha. O método inteiro provê meios pelos quais usuários podem estabelecer, usar, e manter arquivos na internet em uma maneira remota a partir de seu computador local ainda em uma maneira que é similar à manipulação do arquivo usado em seus computadores locais. Uma capacidade alta e outros sistemas de armazenamento é ligado à internet via uma rede de internet opcional que também serve para gerar e dirigir meta-dados com relação aos arquivos armazenados. Um servidor web (140) usando um CGI, baseado em Java, ou outras interfaces transmite e recupera pacotes TCP/IP ou outras informações da internet através de um balanceador/parede de carga (110) pelo uso XML para envolver os pacotes de dados. Instruções de arquivos podem ser transmitidos pela Internet pelo sistema de Recurso Compartilhado. A conta do usuário pode ter uma senha protegida de maneira que somente o usuário possa acessar seus arquivos. No (...).



Relatório Descritivo da Patente de Invenção
para **“RECURSO DE ARMAZENAMENTO DE INTERNET
COMPARTILHADA, SISTEMA DE INTERFACE COM O
USUÁRIO, E MÉTODO”**

5 **CAMPO TÉCNICO**

A invenção se refere a recursos em redes de computador, particularmente Internet, e mais particularmente a armazenamento de arquivos e sistemas de recuperação que são disponíveis no mundo todo via Internet que adicionalmente
10 permite uma transferência direta dos arquivos de Internet para armazenamento de Internet, recuperação, e recurso compartilhado. A presente invenção atua na maneira de um “disco rígido de Internet” ou “drive hard de Internet” para prover armazenamento online e recursos de recuperação para usuários.

15 **TÉCNICA ANTERIOR**

A Internet é a rede de computador do mundo todo, tornada disponível em um vasto número de computadores e fonte de informação para instituições e individuais. Uma parte significativa da Internet é a web pelo mundo inteiro que permite
20 que as páginas da web sejam escritas em HTML e transmitidas sob demanda através da Internet. Desenvolvimentos recentes têm melhor estabelecido o uso de XML (Linguagem Markup Extensível) como um sub-conjunto de SGML (Linguagem Markup Generalizada Padrão, padrão ISO 8879: 1986). FTP
25 (Protocolo de Transferência de Arquivo) prove meios pelos quais arquivos podem ser transferidos sobre a Internet. Todos estes

protocolos são geralmente bem conhecidos na técnica, e recursos colaterais podem ser facilmente obtidos para descrever estes adicionais.

Patentes relevantes para a presente invenção inclui a patente U.S. Nº 5.771.354 depositada por Christopher Crawford em 23 de Junho de 1998 para um Sistema de Backup Online Internet, e patente U.S. Nº 5.901.228 depositada por Christopher Crawford em 4 de Maio de 1999 para um Serviço de Backup Online Comercial.

Adicionalmente, sistemas de programação portáteis tais como Java®, JavaBeans, e JavaScript têm sido extensivamente desenvolvidos com uma antecipação de portabilidade futura ao redor da vasta rede net que é a Internet. Sistemas relacionados a Java® permitidos para programas de orientação de objetos em que objetos ou “beans” permitem a passagem de módulos auto contidos com métodos de processamento associados que são usados para agir sob dos dados acompanhantes. Conseqüentemente, o “bean” pode deslocar através da rede net e, sob circunstâncias adequadas, ter certos processos ativados permitindo manipulação da informação contida no bean.

Avanços nos sistemas relacionados a Java® tem aumentado o JavaBean™ de Empresas (EJB). O JavaBean™ de Empresas permitido para grupamentos de servidores tais que o bean é dado independentemente a partir de servidores específicos nos sistemas, ainda podendo ser ativados ou “instantiated” tal que

a recuperação de erro é fácil, o sistema como um todo é mais robusto, e processamento do bean pode ser realizado asincronicamente de maneira que não aconteça em um tempo pré-determinado ou um seriamente após o outro.

5 JavaBeanTM de Empresas/EJBs permite a serialização de beans. Tal serialização permite que o bean seja representado como uma fonte de dados de comprimento pré-determinado. Em essência, este é justamente um arquivo de dados que é interpretado no contexto próprio, da mesma forma
10 que qualquer arquivo de informação eletrônica. Tal serialização do EJB permite que o mesmo seja replicado e armazenado no caso de falha catastrófica de um servidor preferido ou similar.

Se o servidor sob o qual os dados EJB instantiated, cai, ou falha, uma dupla previamente replicada pode
15 ser usada para continuar o processo e permitir a recuperação do erro. Mais informação acerca da tecnologia JavaBeanTM de Empresas podem ser encontradas em papel, “Tecnologia JavaBeanTM de Empresas: Modelos de Componente de Servidor para Plataforma JavaTM,” por Anne Thomas, revisado em
20 Dezembro de 1998, preparado por Sun Microsystems, Inc. e publicado/tornado disponível por Patrícia Seybold Group of Boston, Massachusetts.

Devido à natureza das novas tecnologias, termos tais como “bean” ou “instantiated” pode parecer não familiar
25 àqueles novos na arte pertinente. Razões para isto incluem a dificuldade de novas comunicações rapidamente e objetos

complexos bem como a natureza bem humorada daqueles que intensamente propõem o estabelecimento de novas tecnologias, particularmente sistemas de software. Conseqüentemente, para os sistemas relacionados a Java®, o tema café está freqüentemente presente indicando aqueles com conhecimento na técnica da matéria de objeto geral de interesse. Enquanto distinções podem ser sutis na técnica, elas podem ser muito importantes e servir as terminações daqueles atentos para estabelecer, compartilhar, e adiantar a tecnologia.

Geralmente, a página de origem ou outras páginas da web são requeridas pelo usuário através da designação de URL (Locador de Recurso Uniforme). Com a transmissão para o usuário via protocolo TCP/IP, a informação presente na URL (e geralmente um arquivo localizado em algum lugar em um computador) é transmitida para o usuário. O arquivo pode ter ligações, ou indicadores, para outros recursos incluindo imagens, gráficos, correntes de áudio e vídeo, ou outros recursos. Linguagem Mark-up é usada na Internet em uma tentativa de prover uma estrutura aberta-fechada pela qual a informação de qualquer tamanho possa ser armazenada eletronicamente (ou talvez igualmente de outra maneira) pode ser tornada disponível para o usuário final na demanda. Como tal, a Internet é vista como uma ferramenta poderosa fazendo quase qualquer fonte de informação disponível para qualquer computador ou para qualquer pessoa usando um computador.

Nos vários anos passados, o computador pessoal tem aumentado em potência e capacidade quando a demanda comercial acionou a pesquisa e desenvolvimento de produtor e vendedor. É agora não incomum ser capaz de facilmente
5 encontrar um sistema baseado em Pentium ® megahertz 500
manufaturado Interl tendo acima de 10 gigabytes de espaço de
disco rígido, bem como 32-256 megabytes de RAM. Como tal, a
energia pela qual os arquivos podem ser recebidos e atuados pelo
usuário local através de seu PC tem mantido o passo com os
10 avanços na tecnologia.

Entretanto, existem obstáculos que permanecem
correntemente para acesso universal a uma informação individual
própria armazenada em seu próprio computador. Primeiro de
tudo, computadores são muito pesados. Eles são grandes. Eles
15 geralmente pesam vários kilogramas e não são facilmente
transportáveis. Computadores laptop de peso baixo ou similares
não tem geralmente os mesmos recursos disponíveis aos usuários
de um PC regular. Adicionalmente, acesso a redes de área local
(LANs) é geralmente não disponível uma vez que o computador
20 deixa as premissas ocupadas pelo LAN. Adicionalmente, acesso a
Internet é freqüentemente restrito pelo uso de um modem.
Modems geralmente promovem velocidades de transmissão de
dados da ordem de 56 kilobits por segundo. Isto é
aproximadamente o mesmo que 7 kilobytes por segundo.
25 Entretanto, ferramentas e outras informações são requeridas para

propriamente transmitir informação sobre a Internet e aumentar o tamanho efetivo dos arquivos.

Igualmente com a disponibilidade aumentada do acesso de banda ampla para a Internet, torna-se uma característica importante do processamento da informação eletrônica e similares, de maneira a prover recursos residentes na Internet. Tais recursos podem incluir o compartilhamento de arquivos de uma maneira que sejam fáceis para usar e entender.

Devido a estas e outras restrições com relação ao transporte de dados, transmissão, e recepção, uma necessidade tem aparecido por meio da qual arquivos e outros dados podem estar disponíveis para todo o mundo através da Internet e não mantido em um computador local. A presente invenção trata desta demanda para prover meios pelos quais arquivos e outros dados podem ser estocados na Internet e tornados disponíveis no mundo todo através da Internet.

DESCRIÇÃO DA INVENÇÃO

A presente invenção prove um “Internet hard drive” ou “disco rígido de Internet” para e a partir do qual arquivos podem ser estocados e recuperados. Denominado comercialmente como “X:Drive”, a presente invenção permite aos usuários estocar arquivos de quaisquer tipos previsíveis no recurso disponível através da Internet. Uma vez disponível para a Internet, os arquivos estocados no X:Drive do usuário são disponíveis para a mesma extensão como na Internet, denominado mundo todo.

Nota deve ser feita que o termo X:Drive se refere tanto ao sistema por inteiro quanto ao espaço individual alocado para o usuário individual. Conseqüentemente, referência é alguma vez feita aqui ao sistema X:Drive ou ao X:Drive para
5 referir ao sistema como um todo. Por outro lado, o termo X:Drive indica o usuário individual do X:Drive, ou espaço alocado. Os usos diferentes são indicados pelo contexto.

De maneira a efetuar o Recurso de Armazenamento de Internet Compartilhada da presente invenção,
10 uma central ou facilidade de armazenamento distribuído é provida. Primeiro e principalmente é a facilidade de armazenamento de acesso de alta velocidade onde arquivos estão atualmente armazenados. Tais áreas de armazenamento individual podem ser alocadas em cotas limitadas individuais, ou ser
15 deixadas na extremidade aberta e limitadas somente pela capacidade dos dispositivos físicos responsáveis pelo armazenamento. Meta-dados, que é dado acerca dos arquivos armazenados no hard drive da rede ou outros dispositivos de armazenamento, é gerado e armazenado em uma base de dados
20 separada. A base de dados de meta-dados (a base de meta-dados) e a facilidade de armazenamento ligada à rede pode ser ligada por uma rede interna. É possível para a base de dados ser armazenada na mesma facilidade de armazenamento da rede ou dispositivo no qual os arquivos do usuário são também armazenados. Manejo do
25 sistema pode selecionar se ou não distribui ou consolida a base de dados com o armazenamento da rede.

Também ligado à rede interna está um servidor web que serve para gerar e transmitir a informação para a Internet, e finalmente para o usuário. Os arquivos de servidor web podem passar através de um balanceador de carga e/ou parede
5 antes de proceder para a Internet. O mesmo é similarmente verdade para informação vindo do servidor web a partir da Internet.

XML pode ser usado em combinação com JavaScript ou similares para prover dois meios pelos quais o
10 Recurso de Armazenamento de Internet Compartilhada da presente invenção pode ser encontrado. O primeiro é um objeto JavaScript que pode ser transmitido para um programa browser funcionando no computador do usuário. Tais browsers podem incluir uns que são bem conhecidos, incluindo um Comunicador
15 Netscape® e Explorador Internet Microsoft®. Alternativamente, uma aplicação de parada sozinha pode ser instalada e armazenada no computador do usuário. Esta aplicação de parada sozinha serve para intermediar os comandos do usuário com o servidor web e finalmente a base de meta-dados no dispositivo de
20 armazenamento da Internet.

Como um aumento adicional, a interface do usuário pode ser um programa de cliente que enreda sem emendas com apresentação do usuário padrão em WYSIWYG (o que você vê é o que você apanha) na interface do usuário gráfico (GUIs).
25 Como tal, um drive pode ser mostrado no computador do usuário e pode ser denominado “x:” (ou “y:” ou “z:”, etc. dependendo das

preferências do usuário). O usuário pode então ler do ou escrever para o drive de Recurso de Armazenamento da Internet Compartilhada da mesma maneira que poderia o hard drive local a:\ e c:\.

5 Quando o usuário paralisa seu computador, informação que está armazenada no Recurso de Armazenamento da Internet Compartilhada da presente invenção permanece na Internet. O usuário pode então acessar tal informação a partir de um outro computador, uma outra localização geográfica, ou
10 igualmente dar permissão para compartilhar arquivos no Recurso de Armazenamento da Internet Compartilhada com outros. Proteção de senhas ou outros protocolos de segurança podem ser usados para limitar ou discriminar o acesso aos arquivos do usuário.

15 O Recurso de Armazenamento da Internet Compartilhada da presente invenção permite transferir arquivos diretamente da Internet para a Internet para um usuário alocado em um espaço de arquivo X:Drive em um processo referido como “Baixar a Carga” ou “Salvar no meu X:Drive”.

BREVE DESCRIÇÃO DOS DESENHOS

20 Figura 1 é uma vista esquemática do sistema X:Drive da presente invenção. Os níveis diferentes das camadas são mostrados, longitudinalmente com os índices de marca de um circulo, triangulo, quadrado, e estrela/asterisco correspondendo ao
25 mesmo índice na Figura 3.

Figura 2 é uma vista esquemática dos objetos de biblioteca Java® operando na transação ou troca de dados ocorrendo na presente invenção.

Figura 3 é um diagrama de fluxo detalhado mostrando a operação da presente invenção. Índices incluindo um círculo, um triângulo, um quadrado, e uma estrela/asterisco correspondem aos níveis das camadas mostrados na Figura 1 e indicam o nível de operação das etapas mostradas no fluxograma da Figura 3.

Figura 4 é um fluxograma mostrando a operação do JavaBean® de Empresas XDArquivos (EJB) usados na presente invenção.

Figura 5 é uma vista superior da arquitetura Java® usada para efetuar transações da presente invenção.

Figura 6 é um diagrama esquemático alternativo da arquitetura Java® mostrada na Figura 5.

Figura 7 é um diagrama de fluxograma e esquemático mostrando o IO (entrada/saída) para as transações de base de dados da presente invenção.

Figura 8 é um diagrama esquemático de processo de recuperação de dados como efetuado pelo componente ArquivoIO do objeto ArquivoXD usado na presente invenção.

Figura 9 é uma representação esquemática da falha de recuperação de elementos.

Figura 10 é um diagrama de fluxograma e esquemático mostrando proteção de vírus efetuado na presente invenção.

Figura 11 é um diagrama de fluxograma e esquemático da transferência de recursos para a Internet (“Baixar a Carga”/“Salvar no meu X:Drive”) como demonstrado na presente invenção.

Figura 12 é um diagrama de fluxograma e esquemático do sistema do cliente usado na presente invenção.

Figura 13 é uma exposição desktop WindowsTM mostrando ambas as aplicações do cliente e do browser-web.

Figura 14 é uma exposição de um browser apontando para um usuário X:Drive.

BREVE DESCRIÇÃO DOS APÊNDICES

Apêndice 1 é uma listagem do código servidor/lado web usado para alcançar a presente invenção.

Apêndice 2 é uma listagem do código usado no lado cliente para alcançar a presente invenção em um meio Microsoft®WindowsTM.

Apêndice 3 é uma listagem do código JavaScript usado para alcançar a presente invenção em um meio Sun Microsystems® Java® (incluindo um em um browser).

MANEIRAS DE REALIZAR A INVENÇÃO

A descrição detalhada demonstrada abaixo em conexão com os desenhos apensos é pretendida como uma descrição das concretizações presentemente preferidas da

invenção é não pretendida para representar somente formas nas
quais a presente invenção pode se construída e/ou utilizada. A
descrição demonstra as funções e a seqüência das etapas para
construir e operar a invenção em conexão com as concretizações
5 ilustradas. Entretanto, é para ser entendido que as mesmas
funções e funções equivalentes e seqüências podem ser
acompanhadas por diferentes concretizações que são também
pretendidas para serem encampadas dentro do espírito e escopo da
invenção.

10 Apêndices 1, 2, e 3 provêm os códigos fonte
para, respectivamente, o Local Web/Código Servidor do sistema
de Recurso de Armazenamento da Internet Compartilhada
W:Drive da presente invenção; o Código de Cliente Windows; e
as Listagens JavaScript para a presente invenção. Estes apêndices
15 são incorporados aqui por esta referência aqui quando se
demonstrado em sua totalidade. É contemplado que estes
Apêndices provêm uma total, completa e habilitada descrição
para aqueles técnicos na arte ou pelo menos pelos quais a presente
invenção pode ser alcançada.

20 Adicionalmente, as referências numéricas
usadas em conjunção com as figuras são numeradas de tal forma
que o lugar 100 do número indica o número dos desenhos nas
figuras. Por exemplo, a série 600 da referência numérica refere-se
a Figura 6, enquanto a série 200 refere-se aos elementos
25 mostrados na Figura 2.

A presente invenção prove um método pelo qual o disco rígido da Internet ou hard drive pode ser alcançado em uma maneira similar ao disco rígido ou hard drive disponível localmente para o indivíduo no computador local.

5 Adicionalmente, quando o uso da Internet torna-se mais familiar e eventualmente a cada dia para pessoas, os recursos providos pela presente invenção podem permitir o uso atual do disco rígido da Internet ou X:Drive demonstrado aqui para atuar tal como um recurso com os arquivos sendo chamados para execução para
10 programas disponíveis e processados tanto localmente e/ou sobre a Internet. À luz dos acima mencionados, pode ser visto que a presente invenção pode agir como uma ponte ou pode pavimentar o caminho através de mais de uma comunidade inter-rede para uso e processamento de informação
15 eletrônica.

O disk drive virtual provido pela presente invenção pode ser seletivamente compartilhado com outros e mantidos inteiramente privados. Adicionalmente, e como demonstrado em mais detalhes abaixo, o uso de
20 base de meta-dados provê acesso mais rápido e a habilidade para distribuir a informação com relação a legião de contas X:Drive sobre uma área geográfica ampla, possibilitando a preservação redundante da informação do usuário pelos grupamentos servidores
25 implementando JavaBeans® de Empresas (EJBs), ou de outra forma.

O Recurso de Armazenamento da Internet Compartilhada, Sistema de Interface do Usuário, e Método demonstrado aqui é geralmente referido como “X:Drive”. Contexto revela se ou não o termo X:Drive está se referindo ao sistema como um todo ou a própria conta individual.

O sistema X:Drive da presente invenção usa aplicações de redes práticas e poder confiar em Java® de Empresas JavaBeans™ (EJBs) para possibilitar computação agrupada e distribuída e meios de administração de arquivos. Embora os desenhos orientados em rede e baseados em Java®, o sistema X:Drive da presente invenção também contempla o uso de padrões de programas abertos tais como XML e Web-DAV (Autorização e Versão Distribuída baseadas em Web). O uso de tais tecnologias é previsto como provendo amplo suporte pela comunidade usuária bem como velocidade e desenvolvimento, refinamento, e polimento.

Como demonstrado na Figura 1, o sistema X:Drive **100** possui uma infra-estrutura de aplicação baseada em rede, multi-camadas. A natureza multi-camadas do sistema permite operações separadas em uma maneira eficiente. O aspecto baseado em rede do sistema X:Drive permite recursos geograficamente dispersos bem como permitir um alto grau de comunicação entre diferentes aspectos ou facetas do sistema.

O sistema X:Drive pode ser considerado tecnologia como um meio que é independente das aplicações e usos para os quais é aplicado. O sistema X:Drive é correntemente

baseado em princípios orientados em objetos com cada camada de aplicação responsável para uma funcionalidade discreta ou aspecto de operação. Ambos os recursos de hardware ou software podem então prosperamente experimentar re-uso pesado com
5 tanto escalabilidade e flexibilidade inerentemente provida. Enquanto estes aspectos vantajosos do sistema X:Drive são alcançados, um sistema multi-camadas, X:Drive envolve um alto custo de complexidade e planejamento. Assim, aqueles que poderiam desejar erroneamente uma cópia do sistema X:Drive
10 poderiam fazer sem resultar em maior gasto de tempo e dinheiro necessário para alcançar o presente sistema X:Drive. Eles poderiam transportar nas costas daqueles que não somente desenvolvem o sistema, mas também aqueles que vão trabalhar direito e uma maneira comercialmente confiável.

15 O uso de camadas nos sistemas X:Drive da presente invenção é realizado em ambos os sistemas de rede e sistemas de aplicações envolvidas em alcançar o X:Drive.

Como mostrado na Figura 1, uma variedade de camadas, ou passos, estão presentes entre o cliente **102** e os
20 recursos de dados últimos **104**. Entre o cliente **102** e os recursos de dados **104**, estão um ou mais passos ou camadas, acompanhando o seguinte.

O cliente **102** pode ser acoplado a uma rede pública **106** (tal como a Internet) que pode incluir um re-diretor
25 DNS **108** bem como um balanceador de carga **110**. A rede pública **106** pode então levar a uma rede de servidor web **120**. O servidor

web pode então levar a uma rede de aplicação **122**, que por sua vez leva a rede EJB (JavaBeansTM Empresa) **124**. A rede EJB **124** pode levar a uma rede de transação **126**, que por sua vez leva a recursos de dados **104**.

5 O cliente **102** pode ser tanto uma aplicação baseada em web ou browser ou uma aplicação residente em um sistema WindowsTM X (o X indicando a versão do Windows aplicável, isto é, Windows ® 95, Windows® 98, Windows® 2000, etc). Requisições geralmente originadas a partir do cliente
10 como o sistema X:Drive **100** é um que opera no comando dos usuários direcionando o programa do cliente. Requisições do cliente podem ser feitas versus a função do Protocolo de Transferência de Hipertexto (HTTP) GET/POST. Na concretização preferida, a operação GET/POST pode ser
15 aumentada com extensões Web-DAV para o protocolo HTTP. Comandos são transmitidos pelo cliente **102** sendo enviados para o re-diretor DNS **108**, que então isola a requisição via um processo de servidor de procuração. Um processo de servidor de procuração evita a conexão direta entre o cliente **102** e as outras
20 fontes de recursos abaixo no sistema X:Drive **100**. Tal serviço de procuração evita interrupção inadvertida ou danosa de serviço permitindo somente certos comandos ou informações serem propagados através do sistema X:Drive **100**. Isso evita usuários danosos de interrupção do sistema tal como comandos nocivos
25 que são interrompidos pelo servidor de procuração e denegar adicional propagação.

Após o comando do cliente ter passado através do servidor de procuração/re-diretor DNS **108**, a requisição pelo cliente **102** então redirecionado para maior facilidade. Como o sistema X:Drive é escalável, facilidades podem ser distribuídas geograficamente, igualmente sobre a face do globo. Isto permite, no início, mais eficiência para tomar lugar no sistema X:Drive **100** da presente invenção de maneira que mais usuários possam ser mais rapidamente servidos e de maneira que as características vantajosas do sistema X:Drive possam ser realizadas por um número mais amplo de usuários o mais rapidamente possível.

Devido à construção e arquitetura do sistema X:Drive **100**, um número de máquinas/usuários funcionando em um número de processos diferentes pode ser distribuído sobre uma ampla área. Banda larga ou acesso de alta velocidade como provido pelo suporte principal da Internet ou similares podem permitir ao sistema X:Drive de ser efetivamente realizado sobre a face total do planeta. A escalabilidade e a flexibilidade da presente invenção aumenta sua utilidade. Tais vantagens são adicionalmente aumentadas pelo uso eficiente dos recursos de maneira que serviço maior e melhor possa ser provido.

Após receber o requisição a partir do cliente **102**, o re-diretor DNS **108** transmite as requisições para um balanceador de carga que pode prover um segundo processo de procuração sob o protocolo HTTP e transmitir a requisição para o servidor web menos carregado e mais disponível em uma interna, não routable, ou outra rede de servidor **120**.

A rede de servidor web **120** pode ser não routable e pode compreender um número de máquinas individuais ou servidores processando o HTTP ou outras requisições a partir de um ou mais balanceadores de carga **110**. Cada um dos
5 servidores web **140** na rede **120** pode manipular requisições HTTP para o teor estático, tal como HTML e arquivos gráficos. Os servidores web podem procurar todas as requisições para teor dinâmico para uma rede **122** de aplicação Java®.

Como usado no sistema X:Drive **100** da
10 presente invenção, as redes de aplicação Java® podem ser não routable. O uso de facilidades não routable no sistema X:Drive **100** da presente invenção indica suas operações em uma rede de área local (LAN). Entretanto, entre camadas, as redes individuais podem estar disponíveis de tal maneira que um servidor web **140**
15 em Illinois pode passar requisições de teor dinâmico para grupamentos de aplicação Java® **122** em Wisconsin.

Cada grupamento de aplicação Java® **122** pode ser composto de um número de servidores de aplicação Java® **142** com cada servidor **142** manipulando as funções mostradoras
20 necessárias para contas dos usuários, incluindo a geração de XML, HTML, e outros mostradores de instrução para cada browser ou clientes de aplicação **102**. Se um grupamento de aplicação Java® **122** recebe uma requisição de recurso a partir da camada do servidor web **120**, o grupamento de aplicação Java®
25 **122** pode passar a requisição de recurso na camada de rede EJB JavaBean™ de Empresas.

Como para o servidor web **120** e redes de aplicação Java® **122**, a rede EJB **124** pode também ser não routable e operar após a LAN. A rede EJB pode ser um grupamento EJB possuindo um número de servidores EJB **144**.

5 Cada grupamento EJB manipula a lógica dos negócios e métodos de acesso a recursos e protocolos requeridos para as requisições de recursos e administração. O grupamento EJB (EJBC) oculta memória de recursos comuns tais como a associação de conexões de dados e similares, bem como objetos de dados. Requisições de

10 acesso a recursos e transmissões são então passadas para as camadas de rede de transação **126**, que pode também ser não routable. A camada de rede de transação **126** possui um processador de transação **146** que controla, opera, e garante acesso e transações nos diferentes recursos. Estes diferentes

15 recursos são então os recursos de dados finais **104** que podem incluir arrumador de discos NFS (Servidor de Arquivo de Rede) **150** e base de dados **152**. O arrumador de discos NFS **150** pode suprir a capacidade de armazenamento atual para os arquivos de geralmente qualquer tamanho. A base de dados **152** compreende

20 registros de informação com relação a cada dos arquivos (metadados) armazenados pelo arrumador de discos NFS **150** sob o sistema X:Drive **100**.

Por bifurcação a informação do arquivo na base de dados **152** separada do arquivo atual no arrumador de discos

25 NFS **150**, a informação do arquivo e o desejo do usuário podem ser manipulados muito mais rapidamente como componentes

mostradores da presente invenção sendo importantes para prover a informação ao usuário com relação ao estado e disponibilidade do arquivo armazenado no sistema X:Drive **100**. Conseqüentemente, embora o usuário possa ter um cem arquivos separados no diretório X:Drive, ele ou ela podem estar somente interessado em um. Conseqüentemente de maneira a prover ao usuário a informação necessária para tomar a decisão quanto a qual arquivo receber, mover, renomear, apagar, ou armazenar, o uso de base de dados prove meios muito rápido e fáceis pelos quais tais requisições do usuário podem ser satisfeitas. É antecipado que o uso atual das facilidades de armazenamento de arquivo no arrumador de discos NFS **150** ou similares podem compreender somente uma parte das operações da presente invenção. Ter a habilidade para mostrar, selecionar, e determinar as operações de arquivo é uma das vantagens úteis providas pelo sistema X:Drive **100** da presente invenção.

Deve ser tomada nota do índice não numérico presente na Figura 1. Mais notavelmente, um circulo é associado com o cliente **102**, um triângulo com o grupamento de aplicação Java® **122**, um quadrado com a rede EJB **124**, e uma estrela/asterisco com a rede de transação. Estes índices não numéricos correspondem aqueles demonstrados na Figura 3. Quando diferentes ações são realizadas em camadas diferentes na presente invenção, os índices não numéricos promovem meios fáceis ou visuais pelos quais a operação das diferentes camadas pode ser indicada na Figura 3.

Figura 2 mostra um diagrama lógico na estrutura de seqüência para objetos em uma biblioteca Java® usado no sistema X:Drive **100** da presente invenção. Geralmente, em toda à parte da descrição do sistema X:Drive **100** da presente invenção o prefixo XD indica “X:Drive”. Por exemplo, na Figura 2 os indicadores de etapas/estado dos pontos ErroXD para Erro X:Drive, e pontos XDXML para Linguagem Markup Extensível X:Drive. Do mesmo modo, o uso do termo ArquivoXD indica Arquivo X:Drive como um objeto em uma biblioteca Java® e intermediando as operações do arquivo da presente invenção.

Na Figura 2, o sistema Java® **200** permite operações a serem realizadas na base meta-dados **202** e operando no sistema Sistema de Arquivo (OS) **204**. Adicionalmente, o objeto ArquivoXD **210** pode ativar ou inicializar o objeto de Busca da Base de Dados **216**. O objeto ArquivoXD **210** pode ser ativado, ou invocado, pelo objeto de Ação de Arquivo **220**. O objeto de Ação de Arquivo **220** pode também ativar a Busca de Base de Dados **216** e objetos da Grande Busca da Base de Dados **220**. Operações dos objetos da biblioteca Java® no sistema **200** como mostrado na Figura 2 podem ser parte do objeto da Segurança da Sessão **224**, que pode inicializar ou usar o objeto da Busca da Base de Dados **216** e/ou objeto da Transação da Base de Dados **214**. O objeto da Segurança da Sessão **224** pode retornar um objeto separado **226** para o objeto de Dados do Usuário **230**. O objeto da Base de Dados **236** pode receber ou transmitir a partir

de seus objetos de Transação **214**, Busca **216**, e/ou Grande Busca **222**.

A informação gerada pode então ser transmitida para a Base de Dados **202** para meta-informação e o Sistema de Arquivo OS **204** para os dados atuais. Se um erro é gerado durante a operação do sistema de objeto de biblioteca Java® **200**, um objeto ErroXD **240** pode servir para manipular o erro enquanto uma operação de sucesso pode ser retornada na forma de objeto XDXML **242**. No sistema de objeto de biblioteca Java ® **200** da Figura 2, a Base de Dados **202** pode conter inteligência ou programação para conexão à base de dados SQL e similares. Opções com relação às operações da base de dados **202** podem ser lidas a partir do arquivo de configuração. O objeto da Base de Dados **236** pode ser capaz de conectar múltiplas bases de dados para redundância no caso de repetição ou informação redundantemente arquivada, ou para funcionalidade de maneira a conectar aquela base de dados que responde mais rapidamente às requisições e comandos.

O objeto da Base de Dados **236** determina qual operação da base de dados para realizar e/ou aquela base de dados para enviar operações baseadas no tipo de requisição é recebido. Por exemplo, requisições de transações demandam uma base de dados separada a partir daqueles do desejo regular e requisições de Grande Busca **222**. De maneira a manter a operação mais eficiente, o objeto da Base de Dados **236** geralmente envia sessões dos usuários para a mesma base de dados se possível, de

maneira que a latência e a replicação da base de dados não é passada para o usuário.

O objeto da Transação da Base de Dados **214** é capaz de manipular relações SQL maiores tais como aquelas que poderiam causar uma carga na base de dados. O objeto da Transação da Base de Dados **214** pode criar classes de crianças que manipulam a lógica da transação de maneira a operar mais eficientemente.

O objeto de Busca de Base de Dados **216** é designado para manipular relações SQL menores e ter classes de crianças para tipos de buscas específicas, tais com aquelas ao longo dos campos antecipados e comuns ou tipos de informação.

O objeto da Grande Busca da Base de Dados **222** manipula relações SQL não transacional, mais largas tais como aquelas usadas nos sistemas de contagens, monitorias, ou outras. Classes de crianças do objeto da Grande Busca de Base de Dados **222** poderia manipular buscas específicas maiores tais como aquelas que poderiam ser implementadas mensalmente ou em outro período base.

O objeto do ArquivoIO **212** recebe e domina objeto de arquivo de dados Java®. O objeto de arquivo contém lógica para encaixar discos múltiplos ou recursos para redundância e/ou funcionalidade e contem as funcionalidades necessárias para manipular arquivos no Sistema de Arquivo OS **204**. O objeto de ArquivoIO **212** pode reagir com os eventos JMS

(Serviço de Mensagens Java) disparado por eventos nos discos do Sistema de Arquivo OS **204**.

Alternativamente, um ou mais objetos monitorados podem ser usados para coletar informação de estado
5 pertinente com relação ao Sistema de Arquivo OS **204**. Quando objetos monitorados são usados, objetos do ArquivoIO então examina os objetos de monitoramento comum para determinar o estado do sistema. No presente sistema, o objeto monitorado é denominado o bean do Estado do Ponto Base, ou bean MPS, **534**
10 (Figuras 5 e 9).

Adicionalmente, as transações no nível do disco são realizadas pelo objeto ArquivoIO **212**. Sob a administração do objeto ArquivoIO **212**, contas dos usuários são capazes de viajar com velocidade e atravessar vários discos. A viagem através de
15 tais vários discos possibilita melhor recuperação da falha podendo um erro ocorrer ou recursos do sistema tornar-se indisponíveis em uma maneira não prevista. O objeto do ArquivoXD **210** usa o ArquivoIO **212** para manipular as transações do sistema de arquivo. Pelo uso do objeto do arquivo de Transação de Base de
20 Dados, o objeto ArquivoXD **210** manipula transações de arquivos de base de dados. O objeto ArquivoXD **210** coordena transações para ambos os objetos do ArquivoIO **212** e objeto do arquivo de Transação de Base de Dados **214** para manter ambos sincronizados e para manipular falhas que podem ocorrer.

25 O objeto de Dados do Usuário **230** mantém dados do usuário em uma sessão do sistema X:Drive. Uma sessão

é basicamente uma viagem de tempo para a qual o usuário encaixa o sistema X:Drive. Métodos são incluídos no objeto de Dados do Usuário **230** para manipular o estado do usuário, de maneira que a atividade pode ser monitorada, bem como se o usuário está ligado ou não.

O objeto Segurança da Sessão **224** usa mecanismos de sessão lógica da web para criar o objeto de Dados do Usuário **230**. Isto acontece pelo retorno a um objeto separado **226**. O objeto da Segurança da Sessão **224** autentica a senha do usuário e expira a sessão velha com redirecionamento de tal sessão velha para páginas apropriadas.

O objeto da ação de Arquivo **220** pode ter classes de crianças e conter lógica para determinar tipos de requisições tais como requisições de usuários, requisições de administração, etc. Testes para requisições de ação de arquivo tais como cotas e permissões, etc., podem também ser manipuladas pelo objeto de Ação do Arquivo **220**. O objeto da ação do arquivo **220** acessa os métodos de arquivo no objeto de ArquivoXD **210**.

O objeto ErroXD **240** lê um arquivo de configuração de listas de erros que dá a cada erro um número I.D. Tais listas de erros preferivelmente giram na linguagem pela qual o sistema X:Drive **100** da presente invenção é programado. Tais listas podem ser capazes de girar no associado com o qual o sistema X:Drive **100** opera. Valores omitidos para a lista pode ser para erros X:Drive na linguagem em Inglês. O objeto de ErroXD **240** preferivelmente mantém os erros em uma pilha e retorna

qualquer de tais erros a partir da pilha. Adicionalmente, o objeto ErroXD **240** preferivelmente aceita um novo erro por código ou mensagem.

O objeto XDXML **242** aceita um objeto e distribui como saída em representação XML de uma transação ou estado requerido pelo usuário ou software do cliente.

Figura 3 mostra o fluxo de dados através do sistema X:Drive **100** da presente invenção, particularmente aqueles que refletidos pela configuração em camadas mostrada na Figura 1. A partir do ponto de partida **300**, uma requisição é enviada pelo comando HTTP POST/GET da etapa **302**. Protocolo Web-DAV pode também ser usado e é correntemente considerado preferível. A requisição enviada é implementada pelo cliente **102** e é avaliado pelo servidor web **120** como uma requisição para o teor estático na etapa **304**. Se a requisição é para um teor estático, o arquivo é servido pelo servidor web **120** na etapa **306**, e o arquivo é mostrado na etapa **308** pelo cliente **102**.

Se na etapa **304** a requisição para o teor estático é avaliada como negativa, uma requisição de procuração é emitida pela rede do servidor web **120** para o grupamento de aplicação Java® (JAC) **122** e submetido a uma entrada na etapa **314**. O grupamento de aplicação Java® (JAC) **122** então analisa gramaticalmente o cabeçalho da requisição da etapa **316**. A rede de JavaBeanTM de Empresas (EJB) **124** então autentica a requisição na etapa **318**. Se a autenticação não pode ser encontrada, controle de processo é então re-direcionado para a

página de re-senha através da rede JAC **122** na etapa **320**. Se autenticação ocorre na etapa **318**, a rede JAC **122** então analisa gramaticalmente os dados de forma multi-parte da etapa **324**.

A rede JAC **122** então determina o tipo de
5 requisição da etapa **326**. A requisição é então submetida à Ação de Arquivo EJB **220** da etapa **328**. A rede EJB **124** então avalia a requisição da etapa **330** de maneira a assegurar que todas as regras do negócio e outras limitações aplicáveis são encontradas, tais como as limitações de cota, permissões, e similares. Se a
10 avaliação tem sucesso na etapa **330**, a rede EJB **124** então submete a requisição ao ArquivoXD EJB **210** na etapa **332** e no processador de transação **146**. As ações apropriadas são então tomadas através da base de dados transacional **152** e os arrumadores de discos **150**. Se a avaliação das regras de negócios
15 **330** cai, um erro pode ser gerado e, como para outros erros no processo de fluxo de dados da Figura 3, um objeto de erro de sessão **334** pode ser gerado em uma pilha de erros de sessão **336**.

Na efetivação da transferência de dados para os resultados dos recursos do sistema **104**, avaliação é feita quando
20 para a operação na etapa **340**. Se a operação não é uma operação de leitura de dados tais como uma listagem de diretório ou leitura de arquivo, a pilha de erro é checada na etapa **342**. Se um erro ocorreu, o estado do erro é enviado para o cliente **102** na etapa **344**. O cliente **102** então aceita o código XML transmitido e
25 devolve o mostrador apropriado para o usuário na etapa **346**. Se a etapa de avaliação da pilha de erro **342** não revela qualquer erro,

uma mensagem de sucesso é gerada na etapa **350**, e o XML subsequente gerado é recebido pelo cliente **102** e mostrado para o usuário na etapa **346**.

Se na etapa de avaliação **340**, a operação não é
5 uma ação de leitura de dados, a pilha de erros é checada na etapa **352** da mesma forma como foi feita na etapa **342**. Se um erro ocorreu, o estado do erro é enviado para o cliente **102** na etapa **354**. A mensagem de estado de erro é então recebida como código XML pelo cliente **102** na etapa **346** e mostrado ao usuário. Se na
10 etapa de avaliação **352** a pilha de erro revela nenhum erro, a avaliação é então feita pelo grupamento EJB como para se ou não a operação é uma leitura de arquivo na etapa **360**. Se a operação é uma leitura de arquivo, a corrente de dados é convertida para uma corrente de rede e transmitida como um arquivo para o cliente **102**
15 pela rede de aplicação Java® **122** na etapa **362**. Os dados são então aceitos pelo cliente **102** e servem ao usuário na etapa **364**.

Se na etapa de avaliação **360** a operação não é uma leitura de arquivo (ver Figura 4), então por eliminação, a ação é uma requisição para meta-dados de arquivos tais como
20 uma indicação de listagem de diretório de atributos de arquivo ou similares. Na etapa **366**, os meta-dados recuperados a partir da base de dados **152** são então transformados em formato XML pelo grupamento EJB **124**. Os dados XML são então transmitidos para a rede JAC **122**, que encapsula o XML a partir da rede e
25 envia os mesmos para o cliente na etapa **368**. A rede JAC **122**

então envia o XML encapsulado para o cliente **102** para devolver e mostrar na etapa **346**.

Como indicado na descrição acima com relação à Figura 3, usuários utilizando o sistema cliente **102** para conectar
5 o Sistema X:Drive **100** via a Internet pública e então submeter requisição e receber respostas efetivas os indicar as requisições dos usuários. Requisições para manipulações de arquivos, tais como uploads, downloads, cópias, movimentos e deslocamentos updates através de cada camada funcional do sistema X:Drive
10 **100**.

O núcleo do grupamento EJB, e como indicado na Figura 2, o ArquivoXD EJB prove eficácia do núcleo no presente sistema X:Drive **100**. O ArquivoXD EJB **210** é um componente multi-camadas. O sistema X:Drive **100** armazena
15 meta-dados de arquivos (tais como estrutura de diretórios, nomes de arquivos, atributos de arquivos, etc.) na base de dados **152** para recuperação rápida, escolha, busca, ligação, e outras capacidades além de sistemas de arquivos padrões. Os dados do arquivo atual são armazenados pelo sistema X:Drive **100** na rede - ligando
20 unidades de armazenamento ou redes de áreas de armazenamento tais como aquelas mostradas na Figura 1, o arrumador de discos NFS **150**.

Para acessar arquivos que existem neste meio híbrido (bifurcado entre informação de arquivo e dados de
25 arquivos), X:Drive usa o objeto ArquivoXD **210** para manipular ambos os arquivos e dados de arquivos em transações de

perpetração em duas fases. Figura 4 mostra detalhes destas transações.

Na Figura 4, o sistema ArquivoXD EJB **400** permite entrar em qualquer um dos cinco triângulos tracejados. Se a ação é para ser copiada, entrada é feita pelo ponto de entrada da cópia **402**. Se a ação é uma leitura de arquivo, entrada é feita em um ponto de leitura de arquivo **404**. Se a ação é uma escrita de arquivo, entrada é feita pelo ponto de escrita de arquivo **406**. Se a ação é apagar um arquivo, entrada é feita pelo ponto de apagar arquivo **408**. Se a ação é mover um arquivo, entrada no ArquivoXD EJB **210** é pelo ponto de entrada de mover **410**.

Começando primeiramente pela ação de cópia de arquivo **402**, a avaliação da operação ocorre na etapa **420**, onde determinação é feita se ou não a ação é uma transação de leitura. Se a ação é uma transação de leitura, corrente de programa procede na ação de leitura e ponto de entrada **404**. A ação da base de dados correspondente **424** é então tomada. A ação é uma transação de leitura, o registro da base de dados correspondente é leitura e avaliação é feita como para se ou não a ação da base de dados, neste caso a ação de leitura, tenha sido satisfatória na etapa **428**. Se a ação de leitura não é satisfatória, as mudanças são então realizadas, se necessário, na etapa **432**. Um erro é então retornado na etapa **436** e o objeto de ArquivoXD aguarda adicionais instruções. Se a avaliação na etapa **428** com relação à ação da base de dados foi satisfatória pode então ser tomadas no arquivo atual no Sistema de Arquivo OS na etapa **440**. No presente caso, a

Ação de ArquivoOS **440** é uma ação de leitura, e o arquivo pode ser lido em um amortecedor temporário ou outro espaço de memória. A Ação de ArquivoOS é avaliada se satisfatória na etapa **444**. Se a etapa da Ação de ArquivoOS **440** não foi satisfatória, um erro fator é retornado na etapa **448**, e as mudanças, se houverem, são então realizadas na etapa **452**. Se a avaliação na etapa **444** foi satisfatória, avaliação é feita para se a ação é ou não uma leitura de cópia na etapa **456**. Se a ação foi uma leitura de cópia, retorno é feito para o ponto de entrada de cópia **402** na etapa **464** de maneira a realizar a porção branca da função cópia. Se a avaliação na etapa **456** indica que a ação não foi uma ação de leitura de cópia, avaliação é feita na etapa **468** para determinar se a ação foi uma ação de mover/copiar. Se a ação foi uma ação de mover/copiar, controle é então direcionado através do ponto de entrada móvel **410** via a etapa **472** de maneira a apagar o arquivo original quando a satisfação da transação mover/copiar na etapa de avaliação **444** indica o sucesso da etapa de escrever arquivo da etapa de Ação de ArquivoOS **440**. Controle do programa é então retornado a ponto da entrada de mover/ação **410** de maneira que o arquivo original possa ser deletado de seu local original via o ponto de entrada para deletar **408**.

Se a etapa de avaliação mover/copiar **468** indica que não foi somente uma leitura de cópia, ela foi também uma mover/copiar, então a ação é confiada ao sistema no nível de

recurso de sistema resultante **104** na etapa **480** e uma indicação de sucesso é então retornada na etapa **484**.

Após alcançar o ponto de entrada de mover em **410**, avaliação é feita na etapa **490** para determinar se a transação é ou não uma cópia de transação. Se é uma cópia de transação, o programa então entra e executa o ponto de entrada de cópia **402**. Se não, o ponto de entrada de apagar **408** é ativado para efetuar o restante da transação de mover.

Conseqüentemente, pode ser visto que uma variedade de ações pode ser realizada dependendo do estado do ArquivoXD EJB **210** nas etapas de ação da base de dados **424** e ação de ArquivoOS **440**.

Na realização de arquivos de leituras e escritas, ações de uma etapa simples são tomadas porque nenhuma destas ações de leitura ou escrita são leituras de cópia **456** ou mover/copiar **468** e de forma que elas caem no sistema de confiança **480** e retornam uma indicação satisfatória na etapa **484**. O mesmo é geralmente verdade para uma ação de apagar de uma etapa. Conseqüentemente, se um usuário deseja ler, escrever ou apagar um arquivo, entrada pode ser feita nos respectivos pontos de entrada em **404**, **406**, e **408**. Erros são retornados quando necessário.

Entretanto, as ações de copiar **402** e as ações de mover **410** requerem voltas múltiplas através do ArquivoXD EJB **210** de maneira a efetuar suas operações. Para a função copiar **402**, a leitura inicial pode ser feita satisfatoriamente com a etapa

de avaliação **456** então impelindo que a etapa de escrever ocorra por retorno para o ponto de entrada da cópia na etapa **464**. A etapa de transação de leitura **420** é então avaliada no negativo e a ação/ponto de entrada de escrever **406** é invocada com a ação da base de dados ocorrendo na etapa **424** para escrever a nova informação para a base de dados transacional **152** e, se satisfatória, a ação de escrever o ArquivoOS para os dados na etapa **440**. Se escrever o arquivo é satisfatório, a avaliação na etapa **456** como se a ação é ou não uma ação de leitura de cópia é respondida na negativa quando é a avaliação da transação como se ou não é uma transação de cópia executada sob a ação de mover na etapa **468**. Os recursos são então confiados, recursos temporários são relegados, e a indicação de sucesso é retornada na etapa **484**.

Conseqüentemente, para a transação copiar **402**, a volta é primeiro feita através da função ler **404** e então a função escrever **406**. Para a mover no ponto de entrada **410**, uma transação de cópia é primeiro executada com uma operação de duas voltas como previamente demonstrado. Após completar a ação copiar, a ação apagar **408** é implementada de maneira a apagar o arquivo original e dados de arquivo. Após a terceira volta através da etapa apagar **408**, a transação não é nem uma leitura sob o comando copiar na etapa **456** nem uma cópia sob o comando mover da etapa **468**. Conseqüentemente, a função mover é satisfatoriamente completada, os recursos do sistema são

confiados na etapa **480**, e um indicador de sucesso é retornado na etapa **484**.

Na Figura 5, uma sobre vista da arquitetura Java® do sistema X:Drive **100** da presente invenção é mostrada.

5 A arquitetura Java® **500** mostrada na Figura 5 pode geralmente chegar do cliente **102**. UM receptáculo de ação de arquivo **504** possui certos atributos e operações como feitas em outros beans da arquitetura **500**. Contidos dentro dos receptáculos de ação de arquivo **504** estão um número de beans existentes, completos e
10 incompletos, bem como outros receptáculos possuindo outros beans. O receptáculo de ação de arquivo **504** contém dois beans completos: um bean completo de dados do usuário **506** e um bean completo de requisições de processo **508**. O bean completo de dados do usuário **506** tem um bean existente de informações do
15 usuário **510** e um bean incompleto de segurança **512**.

O bean completo de requisição de processo **508** contém um receptáculo simples, o receptáculo ArquivoXD **520**. O receptáculo **520** contém três (3) beans e um receptáculo. Os três beans do receptáculo Arquivo XD **520** são: um bean completo da
20 base de dados IO **522**, um bean completo do arquivo IO **524**, e um bean completo de administração **526**. O receptáculo é um receptáculo de recuperação **530** que contém um bean completo de recuperação IO **532**, um bean completo do estado base **534**, um bean completo de administração de recuperação **536**, e um bean
25 completo do processo de recuperação **538**.

Como indicado pela natureza dos beans realizados pelos receptáculos, beans completos geralmente carregam informações acerca do estado do bean, processos, ou de outra forma informações úteis para as finalizações e operações

5 dop sistema X:Drive **100** da presente invenção. Beans incompletos geralmente não carregam informações, e beans existentes são geralmente para informação ou identificação somente. Como beans Java® são objetos pretendidos para realizar tanto dados e processos em associação um com o outro, é para as

10 operações do sistema X:Drive **100** da presente invenção seletivamente e apropriadamente ativar os beans e possibilitar que ações próprias aconteçam. Um receptáculo de ação de arquivo **504** é mostrado na representação alternativa na Figura 6. Na

Figura 6, um cliente **102** emite uma requisição de autenticação do

15 usuário **602** e uma requisição de operação **604**. A requisição de autenticação do usuário **602** é passada através do bean completo de dados do usuário **506** no receptáculo de ação de arquivo **504**. A requisição de operação **604** é passada no bean completo de requisição de processo **508**. O bean existente de informação do

20 usuário **510** então transmite informação para a base de dados de informação do usuário **610**, como faz o bean completo de segurança **512**. O bean completo de requisição de processo usa uma primeira propriedade do arquivo **612** que é carregada após desdobramento do receptáculo do ArquivoXD **520**. A propriedade

25 do arquivo é carregada no bean completo de administração **526** para uso com o sistema de arquivo OS **204**. Um servidor de

transação Java® **620** pode operar em conjunção com a base de dados **152** bem como o sistema de arquivo OS **204** de maneira a processar a requisição de operação **604**. A segunda propriedade do arquivo **630** pode também ser carregada pelo bean completo de administração de recuperação **536** após desdobramento do bean. O bean completo IO de recuperação **532** e o bean completo de administração de recuperação **536** transmitem informações para o disco de estocagem na fila de recuperação **640**. O bean do estado base **534** opera em conjunção com o estado base do sistema **650**.

O receptáculo de recuperação **530** é chamado quando uma vez que um recurso cai começa a recuperar. Descrição adicional do processo de recuperação é dado acima. Entretanto, Figuras 5 e 6 operam em tandem para mostrar linearmente (Figura 5) e organicamente (Figura 6) a estrutura do objeto do ArquivoXD **210**.

Figura 7 mostra o detalhe do componente da base de dados do ArquivoXD. Um processador de transação (tal como Tuxedo de BEA) trabalha em conjunção com o objeto de transação da base de dados **214** bem como o objeto de ArquivoIO **212** para prover um sistema forte e confiável. Tanto a transação da base de dados **214** e os objetos de ArquivoIO **212** incluem lógica e/ou programação para manipulações de situações onde as bases de dados ou acessos de arrumadores de discos não podem ser garantidas. O objeto de transação de base de dados **214** manipula dados inerentes presentes no sistema pelo uso de grupamentos replicados ou repetidos das bases de dados. O

processo de replicação cria latência ou parada, no sistema. De maneira a acomodar esta latência, o objeto de transação da base de dados **214** usa o objeto de sessão (a construção de dados representando uma sessão usuário no sistema X:Drive **100**) para
5 determinar se o requerimento do usuário pode ser transferido, ou replicado, a partir de um grupamento da base de dados para outro, no caso de falha do sistema futuro.

Um aspecto importante com relação à operação confiável do sistema X:Drive **100** é a necessidade de separar
10 bases de dados em grupos funcionais. Enquanto a base de dados examinada pode ser otimizada para exames rápidos e pequenos enquanto uma base de dados pode ser otimizada menos, maiores e mais tempo de consumo para updates, as camadas de base de dados **236** no sistema X:Drive **100** permite para associação dos
15 comandos SQL com grupamentos de base de dados diferentes na funcionalidade. Adicionalmente, a camada de base de dados do X:Drive **236** é configurado para consolidação e adição de base de dados no transporte.

Como mostrado na Figura 7, o comando SQL
20 **710** é emitido e passado para o avaliador de comando SQL **712**. O avaliador SQL determina o tipo SQL pode ser de maneira que o SQL pode ser enviado para o tipo de base de dados apropriados (que é, no sistema X:Drive **100**, a base de dados da transação **150**, a base de dados examinada **152**, ou ambos).

25 Após determina o tipo de base de dados da relação SQL **712**, a base de dados preferencial é avaliada na etapa

714 para determinar se o usuário pode ser enviado de volta para a mesma base de dados. Se o usuário é para não ser enviado de volta para a mesma base de dados, a base de dados correntemente carregada por menos carga é encontrada na etapa **716**, e exame é então feito na etapa **718** para assegurar que a base de dados selecionada, de menos carga, está ainda funcionando, e está disponível. Se ela é, uma especificação com relação à associação dos recursos da base de dados é criada **720** e transmitida para o objeto da base de dados **236**. Objeto da base de dados **236** então toma o comando SQL e passa o mesmo para a base de dados apropriada, tanto na base de dados da transação **150** ou base de dados examinada **152** via associação conexão associada **730**.

Se na etapa **718** a base de dados menos carregada não está disponível, uma base de dados alternativa pode ser usada e exame é feito na etapa **736** para determinar se ou não a base de dados alternativa está em movimento. Se a base de dados alternativa não está em movimento e a etapa de avaliação **736** cai, bases de dados adicionais podem ser examinadas ou, como indicado na Figura 7, um erro fatal pode ser gerado na etapa **738**. Se a base de dados alternativa está em movimento, uma especificação associada **720** é gerada e passada para o objeto da base de dados de maneira que o comando SQL possa ser implementado sob as bases de dados **152** transacionais via conexões associadas **730**.

Se na etapa **714** o usuário pode ser enviado de volta para as mesmas bases de dados, exame é feito na etapa **740**

para determinar se aquelas bases de dados ainda estão em movimento. Se ela está, a requisição é passada para a especificação associada **720** onde ela é subseqüentemente passada através do objeto da base de dados **236**, na associação de conexão **730**, e a base de dados apropriada, se a transação da base de dados **150** ou a base dados examinada **152**. Se a mesma base de dados não está em movimento e a avaliação na etapa **740** cai, uma base de dados alternativa pode ser usada, mas a requisição SQL é examinada na etapa **744** para determinar se o comando SQL é transferível, exame é feito na etapa **750** para verificar se a base de dados alternativa este em movimento e ativa. Pode a avaliação cair, base de dados subseqüentes podem também ser examinadas se o comando SQL é transferível. Entretanto, como mostrado na Figura 7, se a segunda base de dados é não avaliável, um erro fatal pode ser gerado em **746**. Por outro lado, a base de dados está em movimento, e a avaliação na etapa **750** é satisfatória e o comando é tornado disponível para o objeto da base de dados **236** via o padrão de especificação associado **720** e nas bases de dados através das associações de conexão **730**.

De maneira a assegurar a operação própria do objeto da base de dados do ArquivoXD **210**, um monitor do estado da base de dados **760** examina e persistentemente vai para as bases de dados **150**, **152**. O estado é então retornado para o objeto do estado da base de dados **762**. O objeto do estado da base de dados pode ser prover informações para o receptáculo de recuperação **530** do objeto do ArquivoXD **210**.

O mecanismo de recuperação para o sistema X:Drive 100 da presente invenção é mostrado na Figura 8. O objeto ArquivoIO 212 usa um objeto de recuperação tal como um receptáculo de recuperação 530 para manipular transações escritas 406 (quando opostas a transações para ler 404) quando o processador de transação 214 cai. O objeto de recuperação é transparente ao usuário, tornado mais fácil e mais conveniente para o usuário o uso do sistema X:Drive 100 enquanto diminui o interesse tal que o usuário poderia ter no caso de uma perda de energia ou outra falha em uma parte do sistema X:Drive 100.

O objeto de ArquivoIO 212 retorna um erro para o usuário, mas informa que ao usuário que sua requisição foi armazenada no sistema X:Drive 100 e que o sistema X:Drive 100 poderia tentar aplicar a mudança tão logo que possível. Se a unidade de armazenamento, representada como um ponto de base no grupamento EJB torna-se não disponíveis para transações escritas 406, o monitoramento do cliente avaliando a rede EJB 124 em que o estado do ponto de base está “baixo”. Uma vez que o ponto de base está disponível e checado para a integridade de dados, o estado é avaliado de “baixo” para “recuperado” e o objeto de recuperação 530 é chamado para aplicar todos as requisições enfileiradas para o receptáculo de ação do arquivo 504. Isto guarda o usuário de perder catastroficamente carregamento e outros arquivos escritos, mas pode causar algumas perdas em arquivos de ler.

No sistema de recuperação **800** da presente invenção, objetos de base de dados associados multi-conectados, o objeto de ArquivoIO **212** recuperado, e o processador de transação **146** trabalham juntos para criar uma camada de recurso
5 oferecendo maior disponibilidade, recuperação, e escalabilidade. Adicionalmente, esta camada de recurso (ArquivoXD EJB encapsulado **210**) tende a replicação dos dados, tanto geograficamente e localmente. Tal replicação preferivelmente possui os três tratos essenciais de serem off-site, aplicação-
10 acionamento e acessível. Com este nível de replicação controlada, grupamentos X:Drive secundários são habilitados em localizações geograficamente diversas de maneira a aumentar a confiança do sistema X:Drive **100**. Conseqüentemente, a perda de dados a partir de um centro de dados ou igualmente perda física de um
15 centro de dados inteiro não poderia causar perda de dados costumeiros ou acesso. Redirecionamento poderia ocorrer dinamicamente e esta informação poderia ser replicada em uma pluralidade de locais ao redor do sistema X:Drive **100**, o exame ou base de dados de meta-dados provendo pontos múltiplos para
20 os dados do usuário.

No sistema de recuperação **800** da Figura 8, o sistema de recuperação é inicialmente inicializado quando o Bean MPS **534** é colocado em um modo para detectar a recuperação do ponto base na etapa **804**. Na etapa **804**, um método de
25 recuperação é chamado e o ponto base externo é checado. Exame é feito na etapa **806** para avaliar se a recuperação já está

ocorrendo ou não. Se recuperação já está ocorrendo, uma exceção é dirigida na etapa **808** e terminação é feita neste ponto final. Se a recuperação ainda não está ocorrendo, uma lista de pontos base no modo recuperação é gerada na etapa **810**. Adicionalmente, na
5 etapa **812** uma lista de pontos base que estão baixados é também gerada. Exame é feito na etapa de avaliação **818** como para a presença de objetos de recuperação disponíveis na fila de recuperação. Se nenhum dos tais objetos estão disponíveis na fila, o disco ou outra base de dados é colocada no modo “up” na etapa
10 **820**. A fila para aquele disco é então fechada na etapa **822**, e o processo de recuperação está completo na etapa **824**. Se a etapa de avaliação **818** recuperação de objetos estão ainda na fila, avaliação é feita como se ou não o sistema tem ido passar a conta fechada na etapa **830**. Se assim, a fila para o disco em
15 recuperação está fechada na etapa **832** para tanto a avaliação da conta fechada **830** e a fila fechada na etapa **832**, controle é então direcionado para a etapa de avaliação se ou não a fila padrão **834** existe. Se a fila padrão não existe e a avaliação na etapa 834 cai, o objeto de recuperação é removido da fila na etapa **840**. O estado
20 da recuperação é subsequente colocado na requisição para a fila alerta na etapa **842** e retorno é então feito para a etapa de exame **818** para determinar se ou não objetos estão ainda disponíveis para recuperação na fila.

Se o arquivo padrão não existe quando avaliado
25 na etapa **834**, avaliação é feita para se ou não a requisição é mais corrente que o arquivo na etapa **850**. Se a requisição é mais antiga

que o arquivo corrente, o objeto de recuperação é removido a partir da fila da etapa **840**, e o estado para a requisição é colocado na requisição é ou fila alerta **842** e controla o retorno para a etapa de avaliação **818** para verificar se quaisquer objetos de recuperação estão disponíveis na fila de recuperação.

Se, na avaliação da requisição, é encontrado que a requisição é mais corrente que o arquivo, a requisição é submetida ao objeto ArquivoXD **210** na etapa **852**. A submissão da requisição para o objeto ArquivoXD **210** não é recuperável. Se a requisição submetida é satisfatória como indicada pela avaliação na etapa **854**, o objeto de recuperação é removido do fila na etapa **840**, seu estado é colocado na requisição para a fila alerta na etapa **842** e avaliação é feita na etapa **818** como para a presença de quaisquer objetos de recuperação adicional na fila de recuperação.

Entretanto, se na submissão da requisição para o objeto ArquivoXD **210** na etapa **852** a submissão cai, exame é feito na etapa **860** como para se ou não o ponto base foi baixado. Se na etapa **860** o ponto base está ainda acima, a requisição a partir deste ponto base é ignorada na etapa **862** e a fila para o disco está não fechado na etapa **864**. Controle do programa é então retornado para o exame de disponibilidade do objeto de recuperação na etapa de avaliação **818**.

Como mostrado na Figura 9, o bean do estado do ponto base **534** possui estados UP, BAIXAR e RECUPERAR. Este bean é aplicável à base de dados do arquivo, bem como discos do usuário **970**, **972** bem como discos de recuperação **974**,

976. Adicionalmente, o bean de estado completo de administração de recuperação **536** é direcionado através da base de dados de recuperação **980** de maneira a efetuar o processo de recuperação **800**.

5 De maneira a efetuar o escaneamento de vírus e características de reparo, o sistema X:Drive **100** preferivelmente usa o Java® JNI (Interface Nativa do Java) para acessar a Norton Anti-Vírus ou outra biblioteca dinamicamente ligada (NAV.DLL) para escanear arquivos contra vírus via um servidor Java®. O
10 servidor Java® funciona em uma versão Windows™ X servidor e pode usar JNI para fazer chamadas para a biblioteca NAV.DLL dinamicamente ligada. Com efeito, o Windows™ X máquina torna um servidor NAV.DLL especializado localizado na camada da rede EJB **124** do sistema X:Drive **100**, em uma sub-rede da
15 fonte de recurso. A integração lógica da biblioteca NAV.DLL dinamicamente ligada com todos arquivos X:Drive escritos é mostrada esquematicamente no diagrama de fluxo na Figura 10.

Como mostrado na Figura 10, o escaneamento de vírus do sub-sistema **1000** toma a transação/arquivo ID **1002** e
20 a transmissão ID **1004** a partir de um usuário **1006**. A transação/arquivo ID **1002** é passada para um processo de escrever arquivo **1008** executado por um SUN® ou outro servidor web **1010**. O arquivo é escrito por ambas as bases de dados genericamente indicadas na referência **1020** e por uma área de
25 armazenamento de arquivo temporária **1022**. O processo de escrever arquivo **1008** passa a transação de arquivo ID para um

processo Norton Anti-Vírus (NAV) **1024**. Dentro do processo NAV **1024** está o escaneador NAV **1026**. O escaneador NAV monitora a corrente de dados ou de outra forma determina e detecta a presença de qualquer vírus. Se após avaliação o processo

5 NAV **1024** detecta um vírus na etapa de avaliação **1028**, ação de destruição de dados é tomada com relação à base de dados **1020**. Se nenhum vírus é detectado, a sequência move-se para sua terminação final na etapa **1030** e ação de destruição de dados é tomada com relação ao arquivo temporário no meio **1032**.

10 Enquanto ambos os arquivos e transação ID **1002** são resgatados para o processo de escrever arquivos **1008**, a transação ID **1004** sozinha é transmitida para uma etapa de informação de localização de busca **1040** em um SUN® ou outro servidor web **1010**. A etapa de informação de localização de

15 busca **1040** transmite seus resultados para a etapa de avaliação **1042**, que determina se o arquivo está em área de armazenamento temporário **1022** ou não. Se o arquivo está em uma área temporária, o estado de carregamento do arquivo é mostrado na etapa **1044**. Se o arquivo não está no meio temporário **1022**,

20 informação de vírus é buscada na etapa **1050** no processo de estado do arquivo **1036**.

Uma vez que a informação do vírus tenha sido buscada, é avaliado se um vírus está presente ou não na etapa **1052**. Se não há vírus detectado, então a avaliação para vírus

25 termina e reveladas do mesmo modo feito na etapa **1054**.

Entretanto, se a etapa de avaliação **1052** indica a presença de um ou mais vírus, uma pluralidade de opções de vírus podem ser mostradas e apresentadas ao usuário na etapa **1060**. Dentre as opções de vírus disponíveis estão: a limpeza dos vírus na etapa **1062**, a movimentação dos vírus para uma localização diferente na etapa **1064**, e/ou apagar o vírus na etapa **1066**. Se a etapa **1064** é tomada com a movimentação do vírus – arquivo carregado desprezando sua natureza de infecção é efetuado, movimento do arquivo com sua destinação final é feita na etapa **1070**.

Como mostrado na Figura 10, um número de ações de sink dados são tomadas com relação à informação. Adicionalmente, como indicado pela Figura 10, o processo NAV **1024** é inteiramente separado e pode ser considerado como sendo servidor/daemon Java® deixando um Windowns® especializado ou outros servidores.

De maneira a fazer recursos disponíveis em uma base indo para o sub-sistema **1000** de escaneamento de vírus da presente invenção, um arquivo chron **1074** (um arquivo executando comandos em uma base periódica de acordo com o tempo) é usado para remover todos os arquivos a partir de um primeiro recurso de armazenamento temporário **1002**.

Figura 11 mostra o Skip do sistema Baixar a Carga/Salvar no meu X:Drive onde um arquivo na Internet pode ser transferido sobre um X:Drive individual a velocidades de dados geralmente mais rápidas que aqueles disponíveis para o

usuário final. Isto permite ao usuário exercitar domínio e controle sobre o arquivo sem ter carregar a carga do downloading para o seu computador local no momento presente. Uma vez que a transferência ocorreu ao redor da Internet a partir do hospedeiro para o sistema X:Drive **100**, então o usuário pode baixar o arquivo armazenado no seu diretório X:Drive para o seu computador local de acordo com sua conveniência.

Como o X:Drive existe na rede Internet, transferência de um arquivo a partir de um recurso de rede (tal como um servidor web ou FTP) para o X:Drive do usuário é feito muito mais rápido a partir do ponto de vista do usuário pela passagem da conexão local para o usuário e permitindo ao usuário submeter a requisição de transferência diretamente para a rede X:drive para execução. O sistema X:Drive **100** então baixa a carga dos dados requeridos a partir do servidor padrão para o X:Drive do usuário sobre uma velocidade de conexão presumivelmente mais alta da Internet pública.

Como mostrado na Figura 11, o Salvar no meu sistema X:Drive **1100** primeiro tem o usuário **1110** que submeter o URL na etapa **1112**. De maneira a acessar o sistema X:Drive **100** da presente invenção, o usuário submete o URL bem como seu nome de usuário e senha na etapa **1114**. Após submeter o URL e a informação de verificação apropriada, avaliação é feita da informação para a proposta de autenticação na etapa **1116**. Se a avaliação cai e a autenticação não é encontrada, uma forma de senha é mostrada em conjunção com o previamente indicado URL

na etapa **1118**. Se a requisição é autenticada, ela é submetida à fila STD/STMX (Skip o Baixar a Carga/Salvar no meu X:Drive) **1132** na etapa **1130**. Um processo de estado é então gerado na etapa **1134**.

5 O estado Salvar no meu X:Drive é então verificado nas bases pelo uso da fila na área de armazenamento temporário na etapa **1136**. Exame é feito para se a transferência está completa ou não na etapa **1140**. Se a transferência está completa na etapa **1140**, então a conclusão satisfatória é indicada
10 para o usuário na etapa **1142**. Entretanto, se a transferência não está completa, exame é feito para a presença de quaisquer erros de transferência na etapa **1146**. Se um erro ocorreu, uma mensagem de erro é mostrada ao usuário na etapa **1148**. Entretanto, se a transferência está incompleta, mas nenhum erro ocorreu, a mesma
15 é então é mostrada ao usuário na etapa **1150**, e uma curta pausa é tomada na etapa **1152** para invocar de novo o processo STD de checagem na etapa **1136**.

Uma vez que o fila STD **1132** recebe a requisição, um processo daemon processa a requisição a partir da
20 fila STD na etapa **1160**. Exame é feito para a lógica de negócios da requisição listada na etapa **1162**. Se a requisição cai na checagem da lógica de negócios **1162**, o estado é carregado na etapa **1164**. Controle pode transferir de volta para a fila STD **1132**.

25 Se a checagem da lógica de negócios sucede na etapa **1162**, o local URL é contactado pelo sistema X:Drive **100**

na etapa **1170** e o processo de baixar a carga é ativado. Os dados transmitidos pelo URL são então salvados no espaço X:Drive temporário na etapa **1172**, com os dados sendo transferidos então para o espaço de dados do usuário na etapa **1174**. O local URL **1180** pode existir em qualquer na Internet até que ele seja disponível no sistema X:Drive **100**. Em uma maneira similar, um espaço de armazenamento temporário **1182** pode também existir em qualquer lugar na Internet até que ele seja acessível e controlável pelo sistema X:Drive **100**.

Após transferir dados para o espaço de dados do usuário como mostrado na etapa **1174**, exame é feito para o sucesso da transferência na etapa **1188**. Tanto para sucesso ou falha da transferência de arquivo satisfatória na avaliação da etapa **1188**, o estado é carregado na etapa **1164** e é passado na fila STD **1132** até se sucesso ou erro é finalmente encontrado. O processo de estado criado na etapa **1130** monitora os estados avaliados gerados pela etapa **1164** e revela o estado para o usuário durante e após a carga baixada no arquivo a partir da Internet para o sistema X:Drive do usuário.

Figura 12 mostra um diagrama esquemático e um fluxograma para o sistema do cliente geralmente usado pelo Microsoft® Windows™ pra alcançar a presente invenção. O sistema X:Drive oferece a seus clientes dois serviços básicos: um serviço de acesso ao arquivo pelo qual arquivos podem ser carregados e baixados para e a partir do X:Drive, bem como um serviço de manipulação de arquivo a partir do qual meta-dados de

arquivo podem ser obtidos e manipulados. Ambos destes serviços
confiantes sob o contexto de seus usos. Por exemplo, o cliente
web da presente invenção usa características de baixar e carregar
nativas bem como diálogos no browsers web do usuário para
5 facilitar o serviço.

Com o uso de browsers web na maquina local,
clientes Windows® X usam pilhas Windows™ TCP/IP
inerentemente presente com o sistema de operação da versão
Windows® X. Todas as transferências de arquivos efetuadas pelo
10 sistema W:Drive podem ocorrer como HTTP POST/GET ou,
preferivelmente, transferência Web-DAV. Geralmente, duas
camadas básicas estão presentes nos servidores de manipulação
de arquivo do sistema X:Drive 100 da presente invenção. Uma
análise gramatical XML opera em conjunção com o revelador de
15 dados XML. Pela coordenação das duas camadas básicas de
serviços de manipulação de arquivo, o servidor é capaz de
responder com geralmente o mesmo código XML para todos os
clientes. O cliente é então responsável por converter o XML para
uma estrutura de dados relevantes e mostradas no XML em um
20 contexto apropriado. Na presente invenção, o cliente web
JavaScript recebe o código XML e analisa gramaticalmente em
uma estrutura de dados JavaScript. Uma camada mostrada em
associação com o cliente e/ou browser apresenta a estrutura de
dados como um documento HTML. O cliente Windows® analisa
25 gramaticalmente o mesmo código XML. Mas a camada mostrada
apresenta a estrutura de dados em um dispositivo listando o que é

entendido pela versão Windows® X sistema de operação. A importância desta arquitetura em camadas é que ela geralmente torna trivial a criação de novos clientes. Ao invés de simplesmente criar páginas de web dinâmicas (e assim limitar
5 serviço para browsers web sozinhos), o sistema X:Drive **100** pode possibilitar muitas plataformas, tais como sistemas de operação, sem alterar a estrutura do servidor. A maior parte das plataformas vem com algumas espécies de camadas de análise gramatical XML, e muitas plataformas vêm com camadas mostradoras feitas
10 prontas. Conseqüentemente, o tempo para mercado pode geralmente ser considerado baixo e estabelecimento eficiente e implementação do sistema X:Drive **100** da presente invenção pode ser encontrado rapidamente e regularmente. Adicionalmente, expansão em novas plataformas geralmente vêm
15 muito mais rápido e nenhuma alteração da estrutura do serviço geralmente necessita ocorrer como Java® e funcionalidades do programa relatado são altamente portáteis a partir de um sistema a outro.

No sistema do cliente **1200**, como mostrado na
20 Figura 12, o cliente **102** possui um serviço de acesso a arquivo **1202**, incluindo uma camada de processamento de requisição **1204** acoplada a uma camada I/O de rede **1206**. Comandos e dados são então transmitidos para o lado do servidor do sistema X:Drive **100** onde a requisição do lado do servidor processando a
25 camada **1210** transmite dados a um exame de avaliação se a requisição é uma para meta-dados ou não na etapa **1212**. Se a

avaliação cai e a requisição não é uma para meta-dados, a camada I/O de rede **1216** e a camada de acesso a recursos **1218** são envolvidos de maneira a prover acesso para e operação da base de dados da transação **152**.

5 Se a requisição para exame de meta-dados na etapa **1212** sucede, a requisição é passada para a camada de acesso de recursos **1218** e na camada de geração XML **1220**. A resposta para a requisição a partir da base de meta-dados **150** é transmitida para o sistema de servidor de manipulação de arquivo
10 **1230** do cliente **120**. O XML transmitido pela camada de geração de XML **1220** é recebido pelo servidor de manipulação de arquivo **1230** bem como seu manipulador XML **1232**. O XML é então passado na camada de análise gramatical XML da etapa **1234** para chegar a estrutura de dados **1236** que está, então,
15 pronta para mostrar e assim ser passada na camada mostradora de dados **1238** para mostrar ao usuário quem pode então re-inicializar o processo pela implementação do servidor de acesso de arquivo **1202**.

Figura 13 mostra o sistema X:Drive **100** como
20 implementado em um WindowsTM X máquina, neste caso, uma máquina Windows 98 (Um computador pessoal baseado em Intel funcionando no sistema de operação Microsoft Windows 98).

O segundo frontmost window **1310** da Figura 13 é encabeçado pela inscrição “Meu Computador” e mostra a
25 presença de um drive na letra lógica X: **1312** com o logotipo X:Drive e o modelo www.xdrive.com (X:) Isto é um exemplo da

interface do usuário provida pela aplicação do cliente. O sistema X:Drive é transparente ao usuário e funciona como qualquer outro drive presente no sistema.

Se o usuário for clicar no ou ativar o X:\drive no
5 window Meu Computador **1310**, o segundo window **1320** aparece (parcialmente obscurecendo o window Meu computador **1310**) e mostra a listagem sob o X:\Drive. O endereço do window **1320** mostra a localização do diretório como sendo em X:\ **1322**.

Também mostrado na Figura 13 é o ícone
10 desktop **1330**, o ícone menu iniciar **1336**, e o ícone bandeja do sistema **1340**. Estes ícones acompanham o programa do cliente **102** e prove maior funcionalidade para o usuário. Cada ícone serve para ativar o programa do cliente de acordo com as preferências agrupadas do usuário.

15 Figura 13 também mostra a aplicação baseada em web 1350 nos antecedentes, após o Meu Computador 1310 e window X:\ 1320. O window de aplicação baseada em web 1350 é mostrado na Figura 14. Nota deve ser tomada da correspondência exata entre as estruturas do diretório do window
20 de aplicação baseado em web 1350 e o window de aplicação baseado no cliente 1320. Esta correspondência prove o usuário com uma interface uniforme, familiar e dependente com o que o usuário pode confiar.

Como demonstrado acima, estes três Apêndices
25 acompanhantes são aqui incorporados em sua totalidade, como é a aplicação provisória previamente arquivada.

Enquanto a presente foi descrita com relação a concretizações particulares, é reconhecido que variações da presente invenção podem ser transmitidas sem sair do conceito inventivo.

5 **APLICABILIDADE INDUSTRIAL**

É um objeto da presente invenção prover um Recurso de Armazenamento de Internet Compartilhada na qual usuários podem armazenar e recuperar arquivos para torná-los disponíveis para eles próprios, ou possivelmente outros, por toda
10 a Internet.

É um objeto adicional da presente invenção prover todas as maneiras de acessar arquivos e controlar disponibilidades geralmente para arquivos locais para usuários por tais arquivos armazenados na Internet.

15 É um objeto adicional da presente invenção prover uma interface do usuário prontamente entendível e fácil para usar através da qual arquivos podem ser armazenados, recuperados, e manipulados na Internet.

É um objeto adicional da presente invenção
20 coletar meta-dados com relação a tais arquivos e armazenar tais meta-dados em uma base de dados.

É ainda um outro objeto da presente invenção prover uma pluralidade de meios pelos quais os arquivos armazenados na Internet podem ser manipulados e
25 controlados.

É ainda um outro objeto da presente prover acesso de aplicação de parada sozinha para arquivos armazenados na Internet.

É ainda um outro objeto da presente invenção
 5 prover meios pelos quais arquivos da Internet podem ser armazenados em um recurso da Internet por um sujeito de transferência direta Internet-a-Internet para o controle de um recurso do usuário limitado ou remoto.

Estes e outros objetos, vantagens, e utilidade
 10 industrial da presente invenção poderão ser aparentes a partir de uma revisão das especificações acompanhantes e desenhos.

Web Site/Server Code

15
 20
 25

```

###addspace.cgi.....Erro! Indicador não definido.
###cliente_informação.cgiErro! Indicador não definido.
###cookie.cgi.....Erro! Indicador não definido.
###descarregar cliente.cgiErro! Indicador não definido.
###mudança de email_.cgiErro! Indicador não definido.
###erro.cgi .....Erro! Indicador não definido.
###explorador.cgi .....Erro! Indicador não definido.
###explorador_usuário_dados.cgiErro! Indicador não del
###arquivo_carga.cgi.....Erro! Indicador não definido.
###salvar arquivo.cgi.....Erro! Indicador não definido.
#! arquivo carregar_stat.cgiErro! Indicador não definido.
###criar folder.cgi.....Erro! Indicador não definido.
###senha esquecida.cgi...Erro! Indicador não definido.
```

```

###nomedousuário esquecido.cgiErro! Indicador não def
###estrutura genérica.cgiErro! Indicador não definido.
###obter um arquivo dividido.cgiErro! Indicador não def
###obter um arquivo dividido descarregado.cgiErro! Indic
5 ###login.cgi .....Erro! Indicador não definido.
###logout.cgi .....Erro! Indicador não definido.
###barradenavegação.cgiErro! Indicador não definido.
###mudança de senha.cgiErro! Indicador não definido.
###promover.cgi .....Erro! Indicador não definido.
10 ###removerespaço.cgi ....Erro! Indicador não definido.
###retirada selecionada.cgiErro! Indicador não definido.
###re-nomeação selecionada.cgiErro! Indicador não defi
###ajustes de salvar.cgi ..Erro! Indicador não definido.
###dividir um arquivo.cgiErro! Indicador não definido.
15 ###contagem de sinal.cgiErro! Indicador não definido.
###forma de sinal.cgi.....Erro! Indicador não definido.
###sucesso de sinal.cgi...Erro! Indicador não definido.
###toc. de sinal.cgi .....Erro! Indicador não definido.
###pular a descarga.cgi ..Erro! Indicador não definido.
20 ###pular o estado de descaregamento.cgiErro! Indicador
###contar um amigo.cgi .Erro! Indicador não definido.
###web_não-autorizada.cgiErro! Indicador não definido.
#####espaçodeanúncio

#!/usr/bin/perl

25 #####
#####3

```

REIVINDICAÇÕES

1.- Sistema de armazenamento de rede de computador compartilhada, caracterizado pelo fato de compreender:

5 uma primeira base de dados contendo dados de arquivo;

 uma segunda base de dados contendo informação (meta-dados) sobre referidos dados de arquivo de referida primeira base de dados;

10 um servidor, referido servidor executando comandos de arquivo na referida primeira base de dados, referido servidor avaliando contemporaneamente referida segunda base de dados sob execução de referidos comandos de arquivo; e

 uma aplicação de cliente, referida aplicação de
15 cliente se comunicando com referido servidor, referida aplicação de cliente invocando os comandos de arquivo sob referido servidor, referido servidor executando referidos comandos de arquivo, e avaliando informação com relação ao referido primeiro arquivo e segunda base de dados de meta-dados reveladas por
20 referida aplicação de cliente;

 pelo que

 referida aplicação de cliente controla arquivos na referida primeira base de dados de arquivo, e informação com relação ao estado de referidos primeiros arquivos de base de
25 dados é mais prontamente disponível por referência à segunda base de dados.

2.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 1, caracterizado pelo fato de que referida primeira base de dados de arquivo é distribuída sobre pelo menos dois dispositivos de
5 armazenamento físico.

3.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 1, caracterizado pelo fato de que referida segunda meta-base de dados é distribuída sobre pelo menos dois dispositivos de
10 armazenamento físico.

4.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 1, caracterizado pelo fato de que referida aplicação de cliente se comunica com referido servidor via uma procuração.

15 5.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 1, caracterizado pelo fato de que referido servidor compreende uma rede sem roteiro.

6.- Sistema de armazenamento de rede de
20 computador compartilhada, de acordo com a reivindicação 1, caracterizado pelo fato de que referido servidor compreende um processador de transação.

7.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 6,
25 caracterizado pelo fato de que referido processador de transação

garante acesso a e transações nas referidas primeira e segunda base de dados.

8.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 1, caracterizado pelo fato de que referido servidor compreende um grupamento java bean de empresa (EJBC).

9.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 8, caracterizado pelo fato de que referido grupamento java bean de empresa (EJBC) manipula lógica de negócios e métodos de acesso de recurso, bem como ocultando memória para recursos comuns.

10.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 1, caracterizado pelo fato de que referido servidor compreende adicionalmente uma rede de aplicação.

11.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 10, caracterizado pelo fato de que referida rede de aplicação compreende adicionalmente um grupamento de aplicação de java.

12.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 10, caracterizado pelo fato de que referida rede de aplicação manipula funções mostradoras e requisições de recurso.

13.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 1,

caracterizado pelo fato de que referido primeiro servidor compreende adicionalmente um servidor de web.

14.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 13,
5 caracterizado pelo fato de que referido servidor de web manipula todas as requisições para conteúdo estático e requisições de procurações para conteúdo dinâmico.

15.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 1,
10 caracterizado pelo fato de que referido servidor compreende adicionalmente um balanceador de carga, referindo balanceador de carga agindo como procurador de requisições para um sub-servidor tendo o grau mais alto de disponibilidade ou funcionalidade.

15 16.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 1, caracterizado pelo fato de que referido servidor compreende adicionalmente um re-diretor de DNS, referindo re-diretor de DNS agindo como procurador de requisições para um recurso tendo um
20 grau mais alto de funcionalidade.

17.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 1, caracterizado pelo fato de que referido servidor compreende adicionalmente:

25 um processador de transação, referido processador de transação em uma rede sem rota, referido

processador de transação garantindo acesso a e transações nas referidas primeira e segunda base de dados;

um grupamento de java bean de empresa (EJBC) em uma rede sem rota, referido grupamento de java bean de empresa (EJBC) acoplado ao referido processador de transação, e manipulando lógica de negócios e métodos de acesso de recurso, bem como ocultando memória para recursos comuns;

uma rede de aplicação em uma rede sem rota, referida rede de aplicação acoplada ao referido grupamento de java bean de empresa, referida rede de aplicação incluindo um grupamento de aplicação de java, e manipulando função mostradora e requisições de recurso;

um servidor de web, referido servidor de web acoplado à referida rede de aplicação e manipulando todas as requisições para conteúdo estático e requisições de procurações para conteúdo dinâmico;

um balanceador de carga, referindo balanceador de carga acoplado ao referido servidor de web e agindo como procurador de requisições para um sub-servidor tendo o grau mais alto de disponibilidade ou funcionalidade; e

um re-diretor de DNS, referido re-diretor de DNS acoplado ao referido balanceador de carga e agindo como procurador de requisições para um recurso tendo um grau mais alto de funcionalidade.

18.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 1,

caracterizado pelo fato de que referida aplicação de cliente é baseada na web.

19.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 1, caracterizado pelo fato de que referida aplicação de cliente interage com um sistema de operação funcionando em um computador sob o qual referida aplicação de cliente está também funcionando, referida aplicação de cliente adotando e implementando um formato de mostrador visual similar ao referido sistema de operação.

20.- Sistema de armazenamento de rede de computador compartilhada, caracterizado pelo fato de compreender:

uma primeira base de dados contendo dados de arquivo, referida primeira base de dados de arquivo distribuída sobre pelo menos dois dispositivos de armazenamento físico;

uma segunda base de dados contendo informação (meta-dados) sobre referidos dados de arquivo de referida primeira base de dados, referida segunda base de dados distribuída sobre pelo menos dois dispositivos de armazenamento físico;

um servidor, referido servidor executando comandos de arquivo na referida primeira base de dados, referido servidor avaliando contemporaneamente referida segunda base de meta-dados sob execução de referidos comandos de arquivo; referido servidor incluindo:

um processador de transação, referido processador de transação em uma rede sem rota, referido processador de transação garantindo acesso a e transações nas referidas primeira e segunda bases de dados;

5 um grupamento java bean de empresa (EJBC) em uma rede sem rota, referido grupamento java bean de empresa (EJBC) acoplado ao processador de transação, e manipulando lógica de negócios e métodos de acesso de recurso, bem como ocultando memória para recursos comuns;

10 uma rede de aplicação em uma rede sem rota, referida rede de aplicação acoplada ao grupamento de aplicação de java, e manipulando funções mostradoras e requisições de recurso;

um servidor de web, referido servidor de web
15 acoplado à rede de aplicação, e manipulando todas as requisições para conteúdo estático e requisições de procurações para conteúdo dinâmico;

um balanceador de carga, referindo balanceador de carga acoplado ao referido servidor de web e agindo como
20 procurador de requisições para um sub-servidor tendo o grau mais alto de disponibilidade ou funcionalidade; e

um re-diretor de DNS, referido re-diretor de DNS acoplado ao referido balanceador de carga e agindo como procurador de requisições para um recurso tendo um grau mais
25 alto de funcionalidade; e

uma aplicação de cliente, referida aplicação de cliente se comunicando com referido servidor via uma procuração, referida aplicação de cliente invocando os comandos de arquivo sob referido servidor, referido servidor executando referidos comandos de arquivo, e avaliando informação com relação ao referido primeiro arquivo e segunda base de dados de meta-dados reveladas por referida aplicação de cliente; pelo que

referida aplicação de cliente controla arquivos na referida primeira base de dados de arquivo, e informação com relação ao estado de referidos primeiros arquivos de base de dados é mais prontamente disponível por referência à segunda base de dados.

21.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 20, caracterizado pelo fato de que referida aplicação de cliente é baseada na web.

22.- Sistema de armazenamento de rede de computador compartilhada, de acordo com a reivindicação 20, caracterizado pelo fato de que referida aplicação de cliente interage com um sistema de operação funcionando em um computador sob o qual referida aplicação de cliente está também funcionando, referida aplicação de cliente adotando e implementando um formato de mostrador visual similar ao referido sistema de operação.

23.- Método para provisão de espaço de arquivo privado e transferência de informação sobre uma rede de

computador pública, caracterizado pelo fato de compreender as etapas de:

provisão de um sistema de espaço de arquivo privado disponível publicamente acoplado à rede de computador pública;

provisão de um programa de cliente em comunicação com a rede de computador pública;

envio de uma requisição de referido programa de cliente para referido sistema de espaço de arquivo privado publicamente disponível (“sistema privado”);

avaliação de referida requisição;

autenticação de referida requisição;

satisfação de referida requisição; e

retorno de um indicador de sucesso para referido programa de cliente indicando o sucesso ou falha de referida requisição;

pelo que

referido programa de cliente pode criar e controlar arquivos mantidos por referido sistema privado.

24.- Método para provisão de espaço de arquivo privado e transferência de informação sobre uma rede de computador pública, de acordo com a reivindicação 23, caracterizado pelo fato de que a etapa de avaliar referida requisição compreende adicionalmente avaliação de referida requisição para conteúdo estático, e retorno de uma resposta apropriada se referida requisição é para conteúdo estático.

25.- Método para provisão de espaço de arquivo privado e transferência de informação sobre uma rede de computador pública, de acordo com a reivindicação 24, as etapas adicionalmente sendo caracterizadas pelo fato de compreender:

5 provisão de uma rede de aplicação dentro de referido sistema privado;

 atuação como procurador de referida requisição para referida rede de aplicação; e

10 analisar gramaticalmente um cabeçalho de referida requisição.

26.- Método para provisão de espaço de arquivo privado e transferência de informação sobre uma rede de computador pública, de acordo com a reivindicação 23, caracterizado pelo fato de que referida etapa de autenticação de referida requisição compreende adicionalmente:

15 autenticação de um usuário usando referido programa de cliente; e

 autenticação de referida requisição feita por referido programa de cliente para assegurar que se conforme com uma conta associada com referido usuário.

27.- Método para provisão de espaço de arquivo privado e transferência de informação sobre uma rede de computador pública, de acordo com a reivindicação 23, caracterizado pelo fato de compreender adicionalmente:

25 análise gramatical de dados de forma de parte múltipla associada com referida requisição;

determinação de referido tipo de requisição; e
submetimento de referida requisição.

28.- Método para provisão de espaço de arquivo
privado e transferência de informação sobre uma rede de
5 computador pública, as etapas sendo caracterizadas pelo fato de
compreender:

provisão de um sistema de espaço de arquivo
privado disponível publicamente acoplado à rede de computador
pública;

10 provisão de um programa de cliente em
comunicação com a rede de computador pública;

envio de uma requisição de referido programa
de cliente para referido sistema de espaço de arquivo privado
publicamente disponível (“sistema privado”);

15 avaliação de referida requisição para conteúdo
estático, e retorno de uma resposta apropriada se referida
requisição é para conteúdo estático;

provisão de uma rede de aplicação dentro de
referido sistema privado;

20 atuação como procurador de referida requisição
para referida rede de aplicação; e

análise gramatical de um cabeçalho de referida
requisição;

autenticação de referida requisição pela
25 autenticação de um usuário usando referido programa de cliente, e
autenticação de referida requisição feita por referido programa de

cliente para assegurar que se conforme com uma conta associada com referido usuário;

análise gramatical de dados de forma de parte múltipla associada com referida requisição;

5 determinação de referido tipo de requisição;

submetimento de referida requisição; e

satisfação de referida requisição; e

retorno de um indicador de sucesso para referido programa de cliente indicando o sucesso ou falha de

10 referida requisição;

pelo que

referido programa de cliente pode criar e controlar arquivos mantidos por referido sistema privado.

29.- Estrutura de dados para efetuar operações
15 de arquivo em um espaço de arquivo privado e sistema de transferência de informação sobre uma rede de computador pública, caracterizada pelo fato de compreender:

um objeto de dados de usuário;

um objeto de requisição de dados; e

20 um objeto de recuperação;

referido objeto de informação de usuário, referido objeto de requisição de processo, e referido objeto de recuperação, associados dentro de um objeto de ação de arquivo.

30.- Estrutura de dados para efetuar operações
25 de arquivo em um espaço de arquivo privado e sistema de transferência de informação sobre uma rede de computador

pública, de acordo com a reivindicação 29, caracterizada pelo fato de que o objeto de dados do usuário compreende adicionalmente:

um objeto de informação de usuário; e
um objeto de segurança.

5 31.- Estrutura de dados para efetuar operações de arquivo em um espaço de arquivo privado e sistema de transferência de informação sobre uma rede de computador pública, de acordo com a reivindicação 29, caracterizada pelo fato de que referido objeto de requisição de processo compreende
10 adicionalmente:

um objeto de operação de arquivo compreendendo referido objeto de recuperação e um objeto IO de base de dados, um objeto IO de arquivo, e um objeto de administração.

15 32.- Estrutura de dados para efetuar operações de arquivo em um espaço de arquivo privado e sistema de transferência de informação sobre uma rede de computador pública, de acordo com a reivindicação 29, caracterizada pelo fato de que referido objeto de recuperação compreende
20 adicionalmente:

um objeto IO de recuperação;
um objeto de estado de montagem;
um objeto de administração de recuperação; e
um objeto de processo de recuperação.

25 33.- Estrutura de dados para efetuar operações de arquivo em um espaço de arquivo privado e sistema de

transferência de informação sobre uma rede de computador pública, caracterizada pelo fato de compreender:

um objeto de dados de usuário, referido objeto de dados de usuário tendo um objeto de informação de usuário; e
5 um objeto de segurança;

um objeto de requisição de processo, referido objeto de requisição de processo incluindo um objeto de operação de arquivo, um objeto IO de base de dados, um objeto IO de arquivo, e um objeto de administração; e

10 um objeto de recuperação, referido objeto de recuperação incorporando no referido objeto de operação de arquivo, referido objeto de recuperação incluindo um objeto IO de recuperação, um objeto de estado de montagem, um objeto de administração de recuperação, e um objeto de processo de
15 recuperação;

referido objeto de informação de usuário, referido objeto de requisição de processo, e referido objeto de recuperação, associados dentro de um objeto de ação de arquivo; pelo que

20 operações de arquivo podem ser facilitadas pela estrutura de dados incluindo recuperação de falha de recurso.

34.- Recurso de armazenamento de arquivo compartilhado para uma rede de computador, caracterizado pelo de compreender:

25 um recurso de armazenamento de arquivo repartilhável;

um servidor, referido servidor acoplado ao referido recurso de armazenamento, referido servidor:

repartilhamento de espaço de arquivo individual para uma pluralidade de usuários no referido recurso de
5 armazenamento;

recebimento de arquivos para armazenamento no referido recurso de armazenamento;

transmissão de arquivos armazenados no referido recurso de armazenamento;

10 geração de códigos de protocolo de controle para transmissão de referidos arquivos;

recebimento de comandos de arquivo para controle de arquivos no referido recurso de armazenamento; e

transmissão de códigos mostradores indicando
15 estado de arquivo no referido recurso de armazenamento, referidos códigos mostradores representando referida fonte de armazenamento como um acionamento de rede;

uma primeira conexão de rede, referida primeira conexão de rede acoplando referido servidor à rede de
20 computador;

uma estação de operação, referida estação de operação:

recebendo arquivos para armazenamento no referido recurso de armazenamento;

25 transmissão de arquivos armazenados no referido recurso de armazenamento;

recebimento de comandos de arquivo para controle arquivos no referido recurso de armazenamento; e

transmissão de códigos mostradores indicando estado de arquivo no referido recurso de armazenamento, referidos códigos mostradores representando referido recurso de
5 armazenamento como um acionamento de rede; pelo que

um usuário pode armazenar, recuperar e controlar arquivos em uma área de armazenamento de arquivo única e segura no referido recurso de armazenamento
10 repartilhável disponível através de toda rede de computador, e destacada de referida estação de operação.

35.- Recurso de armazenamento de arquivo compartilhado para uma rede de computador, de acordo com a reivindicação 34, caracterizado pelo de que referidos códigos
15 mostradores compreendem adicionalmente:

um objeto interpretável de browser, tal como um objeto de JavaScript, referido objeto revelando estado de arquivo no referido recurso de armazenamento como uma página de web.

20 36.- Recurso de armazenamento de arquivo compartilhado para uma rede de computador, de acordo com a reivindicação 34, caracterizado pelo de compreender:

um programa standalone funcionando na referida estação de operação, referido programa standalone interpretando referidos códigos mostradores, e proporcionando
25 uma interface sem costura ao referido usuário, a interface sem

costura apresentando referido recurso de armazenamento como um recurso local ou de rede para referido usuário, e permitindo que referido usuário manipule arquivos no referido recurso de armazenamento da mesma maneira como recursos de armazenamento locais, tais como acionamento de disco floppy ou um acionamento de hard local.

37.- Recurso de armazenamento de arquivo compartilhado para uma rede de computador, de acordo com a reivindicação 34, caracterizado pelo de que referida rede de computador compreende adicionalmente a Internet.

38.- Método para transferência de dados de um primeiro recurso de rede para um segundo recurso de rede na direção de um usuário, as etapas sendo caracterizadas pelo fato de compreender:

15 submetimento de uma primeira locação de arquivo indicando dados a serem transferidos para o segundo recurso de rede;

 o segundo recurso de rede requisitando referidos dados na referida primeira locação de arquivo a partir do primeiro recurso de rede;

 o primeiro recurso de rede transmitindo referidos dados para o segundo recurso de rede; e

 o segundo recurso de rede notificando o usuário de transferência bem sucedida sob recepção bem sucedida de referidos dados; pelo que

o usuário pode usar os primeiro e segundo recursos de rede, para obter e controlar referidos dados.

39.- Método para transferência de dados, de acordo com a reivindicação 38, caracterizado pelo fato de que o
5 segundo recurso de rede compreende um sistema baseado em subscritor de espaço de armazenamento disponível de rede.

40.- Método para transferência de dados, de acordo com a reivindicação 38, caracterizado pelo fato de que os primeiro e segundo recursos de rede são acoplados à Internet.

10 41.- Método para transferência de dados, de acordo com a reivindicação 38, caracterizado pelo fato de que as etapas são caracterizadas pelo fato de adicionalmente compreenderem:

15 revelação ao usuário de um estado de transmissão de referidos dados de referido primeiro recurso de rede para referido segundo recurso de rede.

20 42.- Método para transferência de dados, de acordo com a reivindicação 38, caracterizado pelo fato de que as etapas são caracterizadas pelo fato de adicionalmente compreenderem:

verificação do usuário como um subscritor em ou membro do segundo recurso de rede.

25 43.- Método para transferência de dados de um primeiro recurso de rede para um segundo recurso de rede na direção de um usuário, as etapas sendo caracterizadas pelo fato de compreenderem:

submetimento de uma primeira locação de arquivo indicando dados a serem transferidos para o segundo recurso de rede, o segundo recurso de rede sendo um sistema baseado em subscritor de espaço de armazenamento de dados
5 disponível de rede;

verificação do usuário como um subscritor em ou membro do segundo recurso de rede;

o segundo recurso de rede requisitando referidos dados na referida primeira locação de arquivo a partir do primeiro
10 recurso de rede;

o primeiro recurso de rede transmitindo referidos dados para o segundo recurso de rede via Internet; e

revelação ao usuário de um estado de transmissão de referidos dados de referido primeiro recurso de
15 rede para referido segundo recurso de rede; e

o segundo recurso de rede notificando o usuário de transferência bem sucedida sob recepção bem sucedida de referidos dados; pelo que

o usuário pode usar os primeiro e segundo
20 recursos de rede, para obter e controlar referidos dados.

44.- Sistema servidor de cliente para um sistema de armazenamento e manipulação de dados baseados em rede, caracterizado pelo fato de compreender:

um sistema de cliente, referido sistema de
25 cliente tendo um serviço de acesso de arquivo e um serviço de manipulação de arquivo;

um servidor, referido servidor proporcionando recursos de armazenamento de dados baseado em rede, e respondendo à requisições transmitidas por referido sistema de cliente, referido servidor efetuando referidas requisições;

5 referido servidor determinando se uma requisição de cliente é uma para meta-dados com relação a dados armazenados sob referido servidor;

 referido servidor proporcionando referidos meta-dados se referida requisição de cliente é para meta-dados, e
10 transmissão de referidos meta-dados para referido serviço de manipulação de arquivo; e

 referido servidor realizando uma ação de arquivo se referida requisição de cliente não é para meta-dados, referido servidor avaliando referido meta-dados e transmitindo
15 referidos meta-dados para referido serviço de manipulação de arquivo; pelo que

 referido servidor opera, e referido sistema de cliente apresenta operações no referido servidor de uma maneira similar às operações locais para referido sistema de cliente.

20 45.- Sistema servidor de cliente para um sistema de armazenamento e manipulação de dados baseados em rede, de acordo com a reivindicação 44, caracterizado pelo fato de que referido serviço de acesso de arquivo compreende adicionalmente:

25 uma camada de processamento de requisições para processamento de requisições; e

uma primeira camada I/O de rede para transmissão de referidas requisições para referido servidor.

46.- Sistema servidor de cliente para um sistema de armazenamento e manipulação de dados baseados em rede, de acordo com a reivindicação 44, caracterizado pelo fato de que referido serviço de manipulação de arquivo compreende adicionalmente:

um analisador gramatical, referido analisador gramatical analisando gramaticalmente referidos meta-dados de referido servidor;

uma estrutura de dados, referida estrutura de dados recebendo e preservando dados analisados gramaticalmente de referido analisador gramatical; e

uma camada mostradora de dados, referida camada mostradora de dados operando e revelando referidos dados analisados gramaticalmente;

pelo que

os meta-dados podem ser revelados para informar sobre dados armazenados no referido servidor.

47.- Sistema servidor de cliente para um sistema de armazenamento e manipulação de dados baseados em rede, de acordo com a reivindicação 46, caracterizado pelo fato de que referido analisador gramatical é um analisador gramatical XML.

48.- Sistema servidor de cliente para um sistema de armazenamento e manipulação de dados baseados em rede, de

acordo com a reivindicação 44, caracterizado pelo fato de que referido servidor compreende adicionalmente:

uma segunda camada I/O de rede, referida
segunda camada I/O de rede engajada quando referidas
5 requisições não são para meta-dados, referida segunda camada
I/O de rede transmitindo requisições para ação de arquivo;
e

uma camada de acesso de recurso, referida
camada de acesso de recurso recebendo transmissões de referida
10 segunda camada I/O de rede e efetuando referidas requisições,
referida camada de acesso de recurso engajada quando referidas
requisições são para meta-dados, referida camada de acesso de
recurso obtendo e transmitindo referidos meta-dados; e

um compilador de meta-dados, referido
15 compilador de meta-dados recebendo referidos meta-dados de
referida camada de acesso de recurso, compilando referidos meta-
dados, e transmitindo referidos meta-dados compilados para
referido sistema de cliente.

49.- Sistema servidor de cliente para um sistema
20 de armazenamento e manipulação de dados baseados em rede, de
acordo com a reivindicação 48, caracterizado pelo fato de que
referido compilador de meta-dados é um gerador XML.

50.- Sistema servidor de cliente para um sistema
de armazenamento e manipulação de dados baseados em rede,
25 caracterizado pelo fato de compreender:

um sistema de cliente, referido sistema de cliente tendo um serviço de acesso de arquivo e um serviço de manipulação de arquivo.

um servidor, referido servidor proporcionando
5 recursos de armazenamento de dados baseado em rede, referido servidor criando e mantendo meta-dados com relação aos dados armazenados, referido servidor respondendo à requisições transmitidas por referido sistema de cliente, referido servidor efetuando referidas requisições;

10 referido servidor determinando se uma requisição de cliente é uma para meta-dados;

referido servidor proporcionando referidos meta-dados se referida requisição de cliente é para meta-dados, e transmissão de referidos meta-dados para referido serviço de
15 manipulação de arquivo;

referido servidor realizando uma ação de arquivo se referida requisição de cliente não é para meta-dados, referido servidor avaliando referido meta-dados e transmitindo referidos meta-dados para referido serviço de manipulação de
20 arquivo;

referido serviço de acesso de arquivo tendo uma camada de processamento de requisição para processamento de requisições, e uma primeira camada I/O de rede de transmissão de referidas requisições para referido servidor;

25 referido serviço de manipulação de arquivo compreende tendo um analisador gramatical XML, referido

analisador gramatical XML analisando gramaticalmente referidos meta-dados de referido servidor, referido de manipulação de arquivo tendo uma estrutura de dados, referida estrutura de dados recebendo e preservando dados analisados gramaticalmente de
5 referido analisador gramatical, e referido serviço de manipulação de arquivo tendo uma camada mostradora de dados, referida camada mostradora de dados operando e revelando referidos dados analisados gramaticalmente, de modo os meta-dados podem ser revelados para informar sobre os dados armazenados
10 no referido servidor; e

referido servidor tendo uma segunda camada I/O de rede, referida segunda camada I/O de rede engajada quando referidas requisições não são para meta-dados, referida segunda camada I/O de rede transmitindo requisições para ação
15 de arquivo, referido servidor tendo uma camada de acesso de recurso, referida camada de acesso de recurso recebendo transmissões de referida segunda camada I/O de rede, e efetuando referidas requisições, referida camada de acesso de recurso engajada quando referidas requisições são para meta-dados,
20 referida camada de acesso de recurso obtendo e transmitindo referidos meta-dados, e referido servidor tendo um compilador de meta-dados na forma de um gerador XML, referido compilador de meta-dados recebendo referidos meta-dados de referida camada de acesso de recurso, compilando referidos meta-dados, e
25 transmitindo referidos meta-dados compilados para referido sistema de cliente, pelo que

referido servidor opera como e referido sistema de cliente apresenta operações no referido servidor em uma maneira similar às operações locais para referido sistema de cliente.

1/13

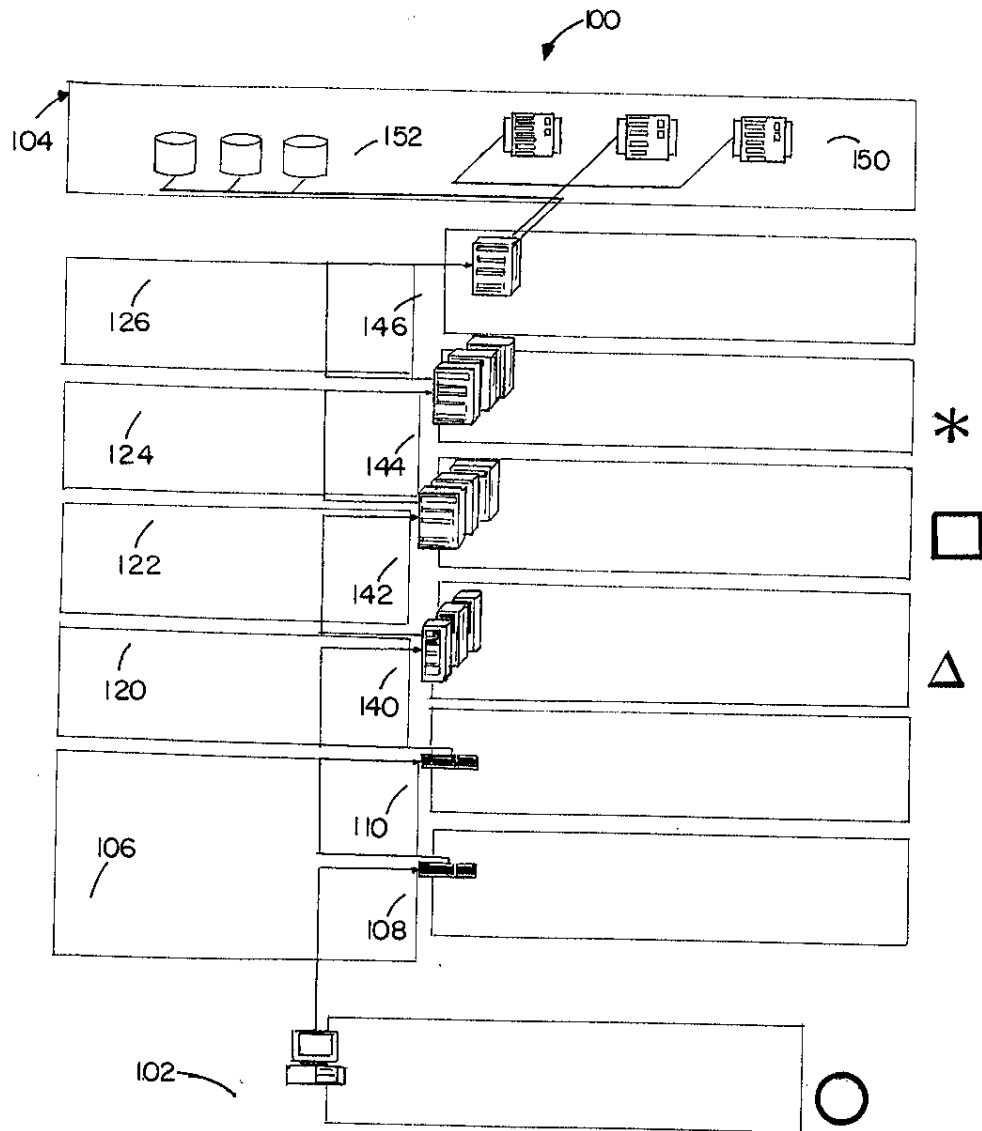


FIG. 1

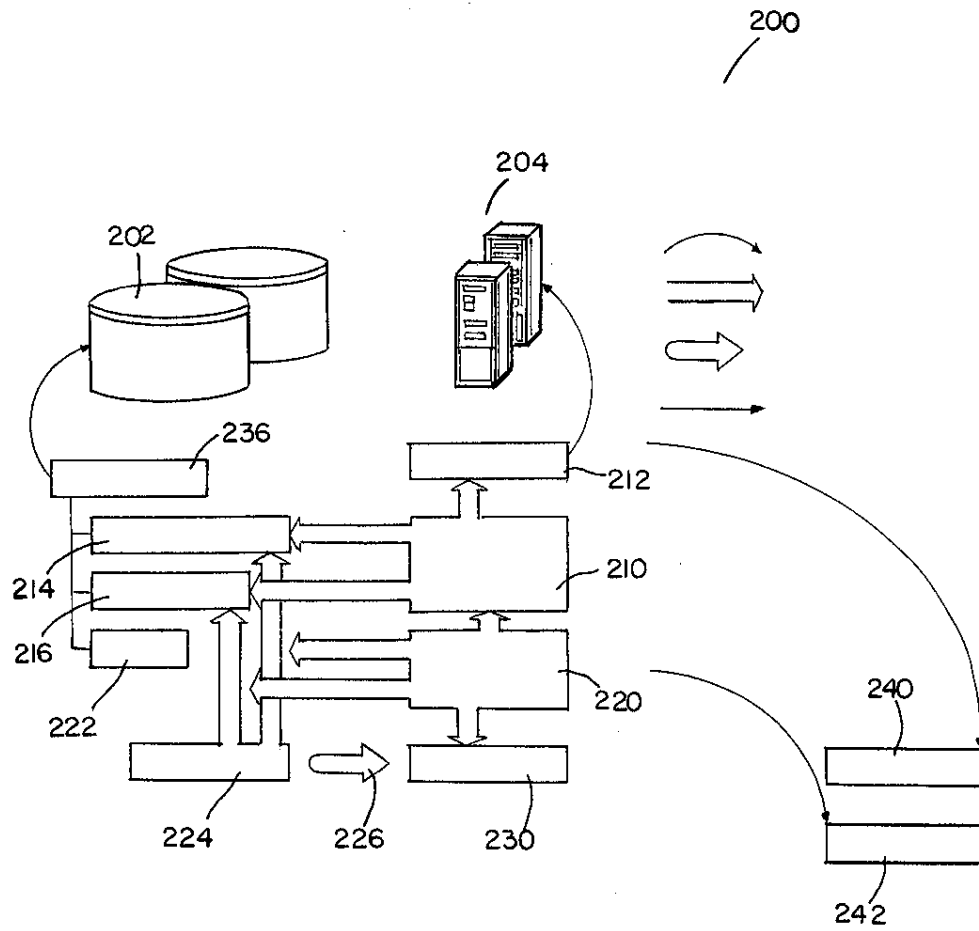
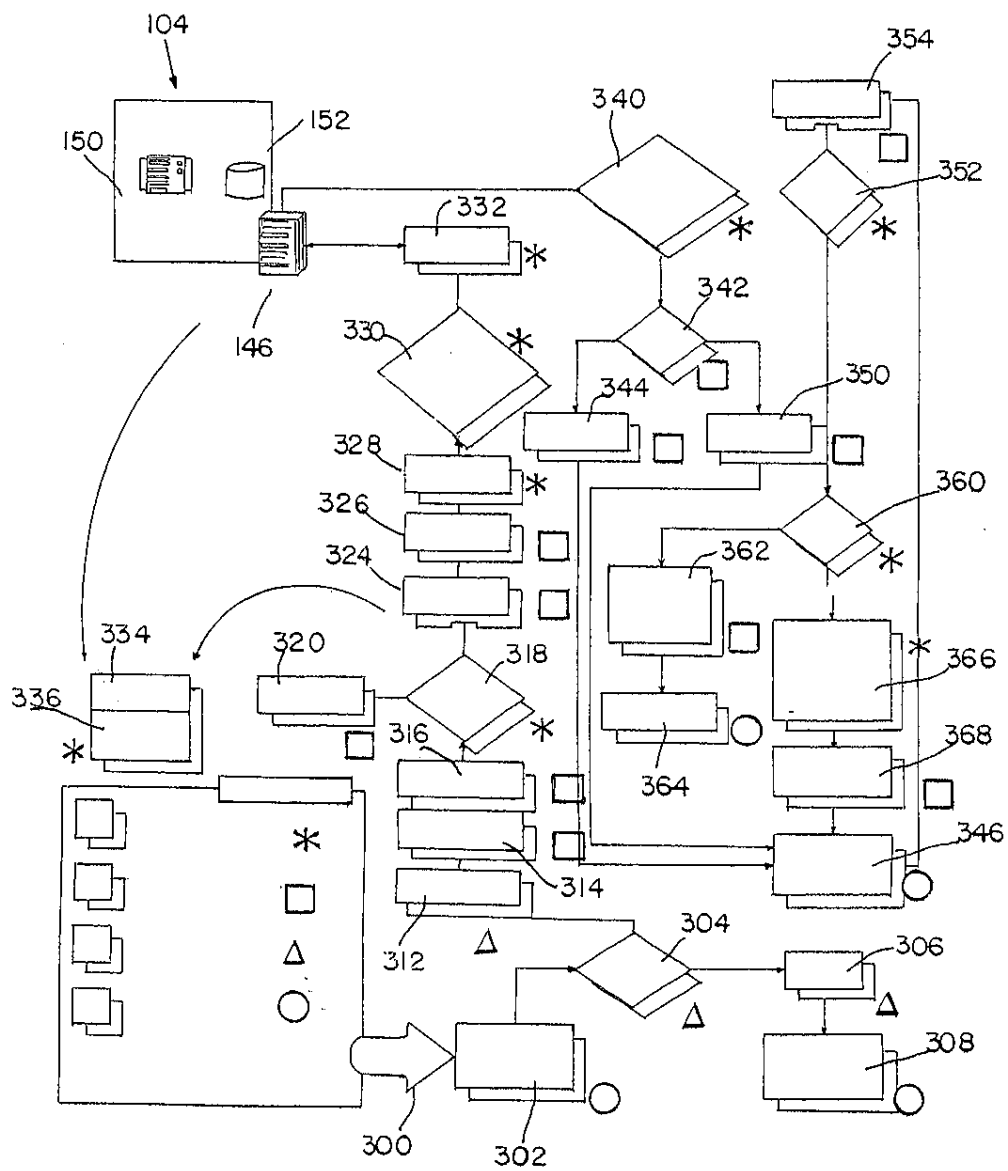
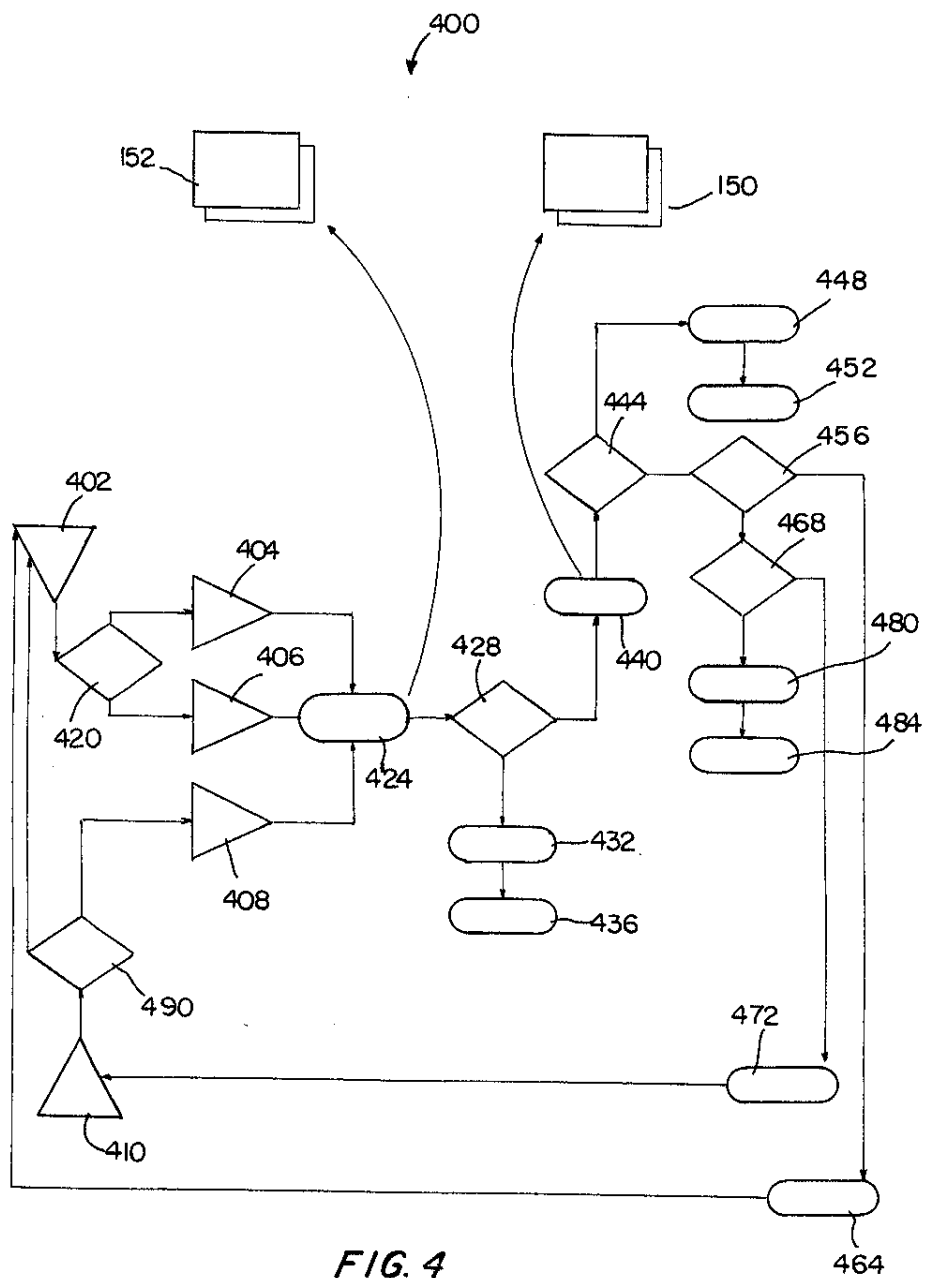


FIG. 2

FIG. 3

4/13



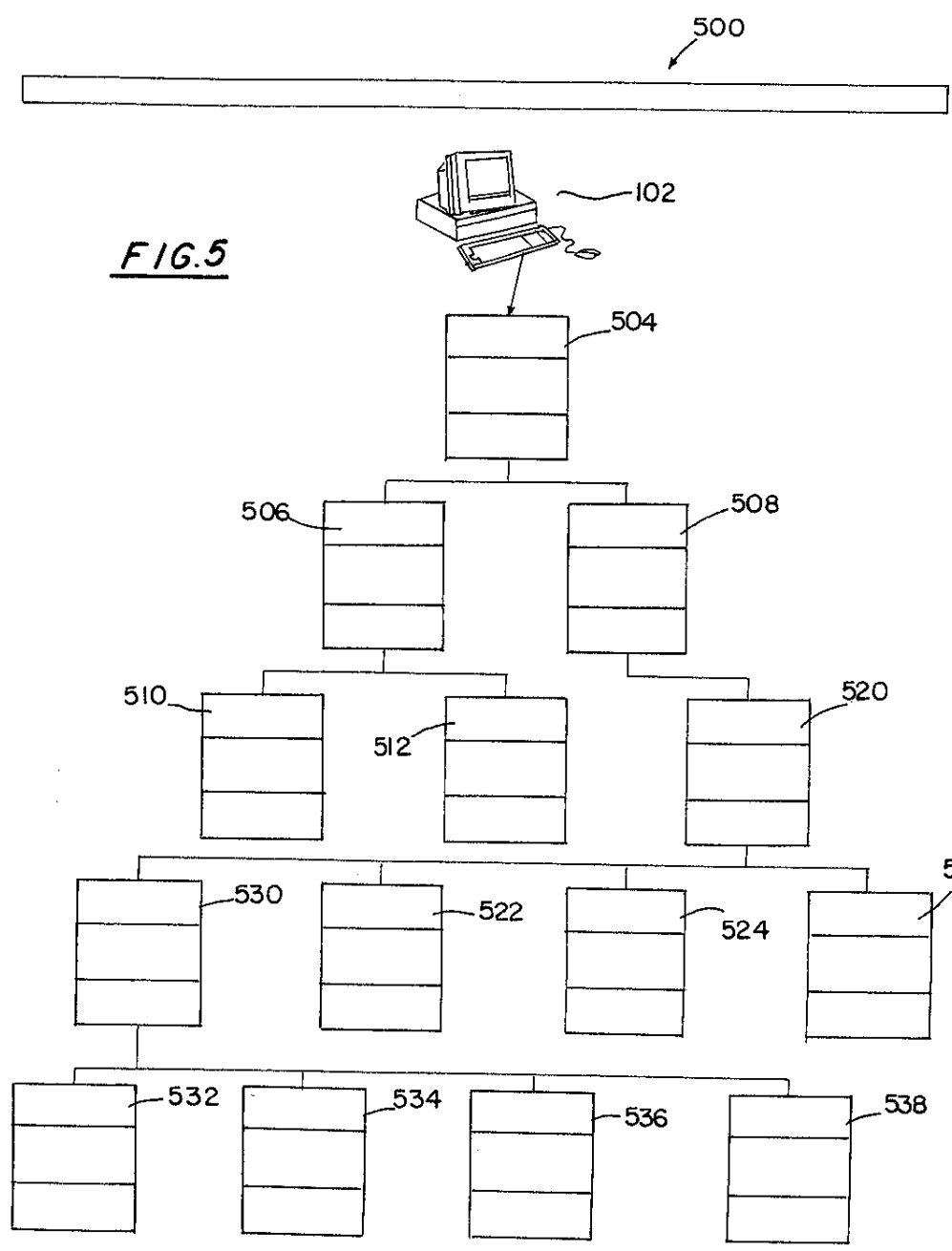
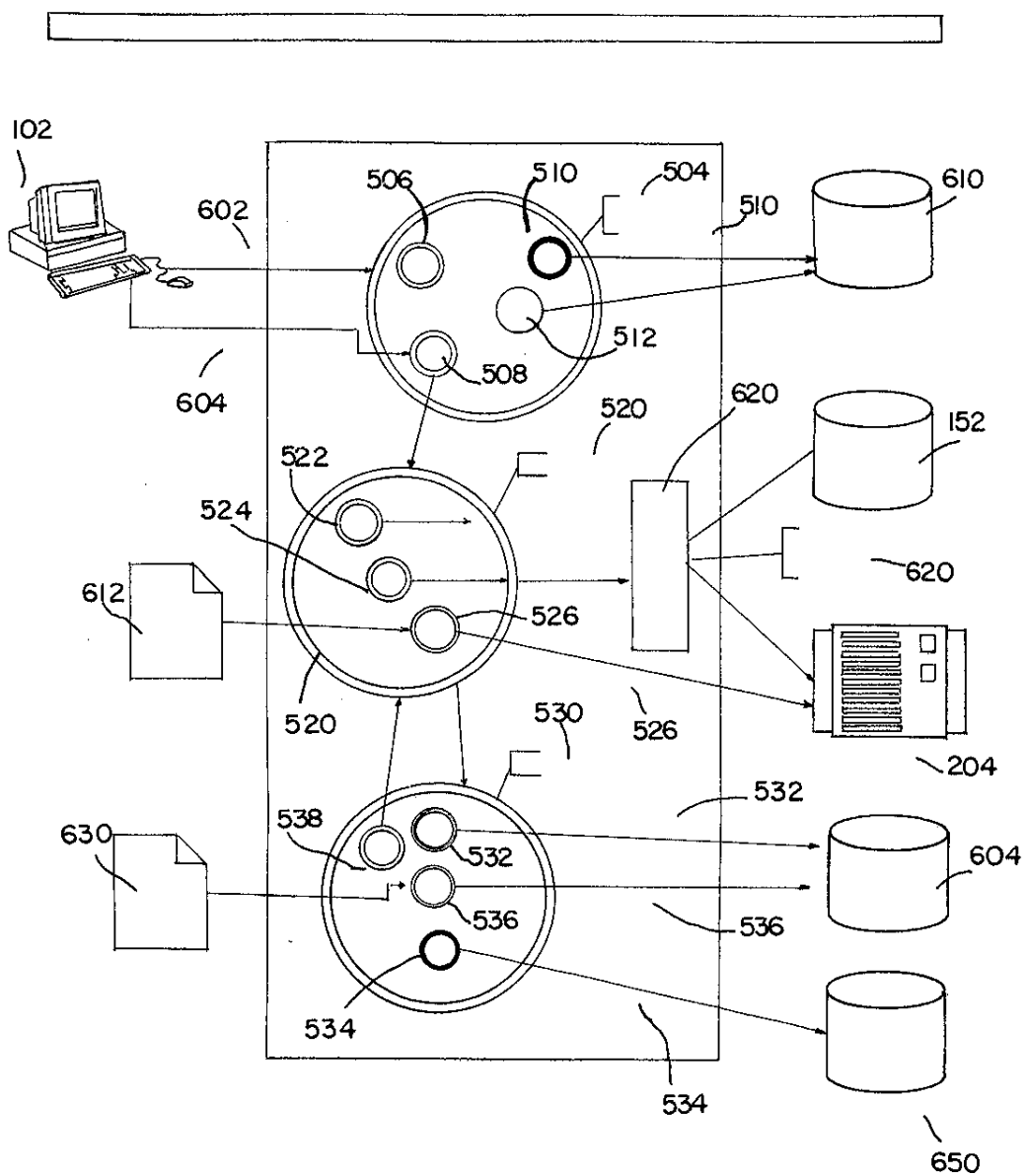
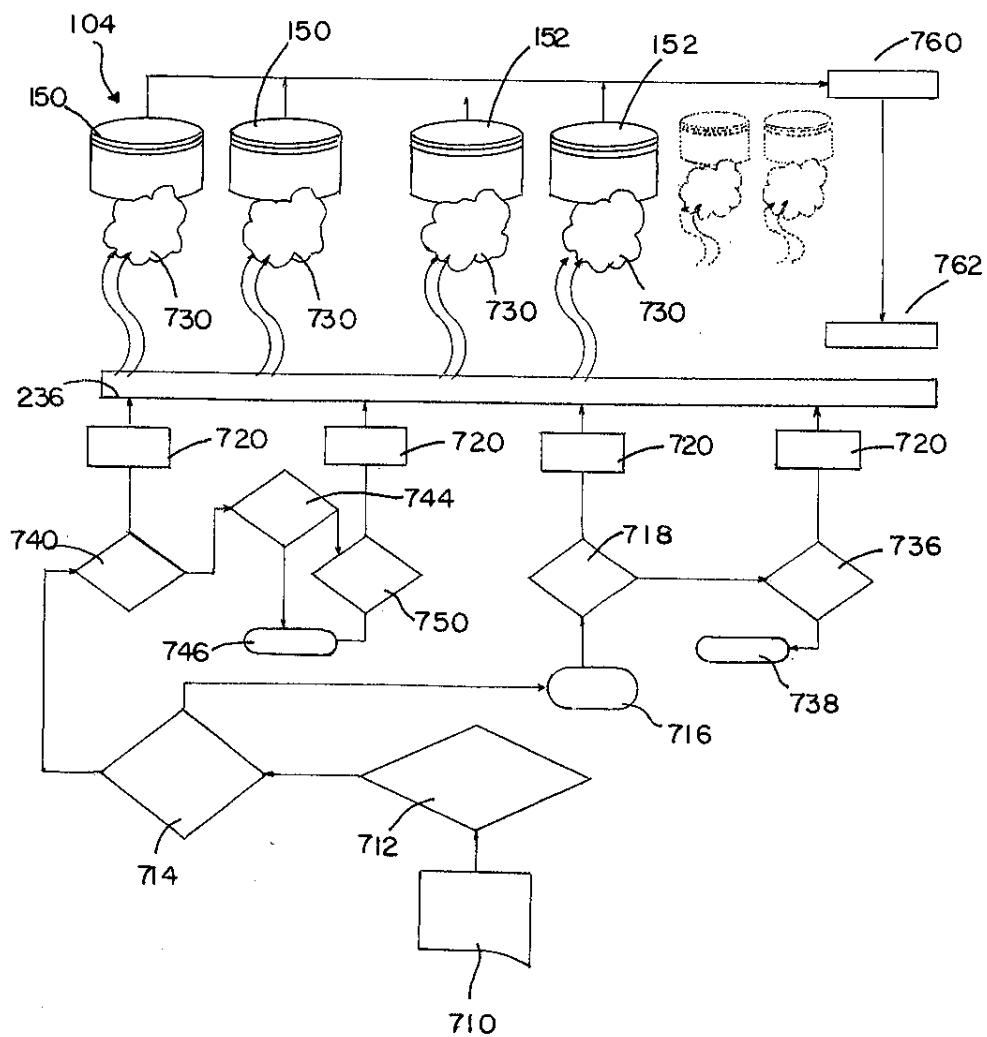


FIG. 6



7/13

FIG. 7

8/13

FIG. 8

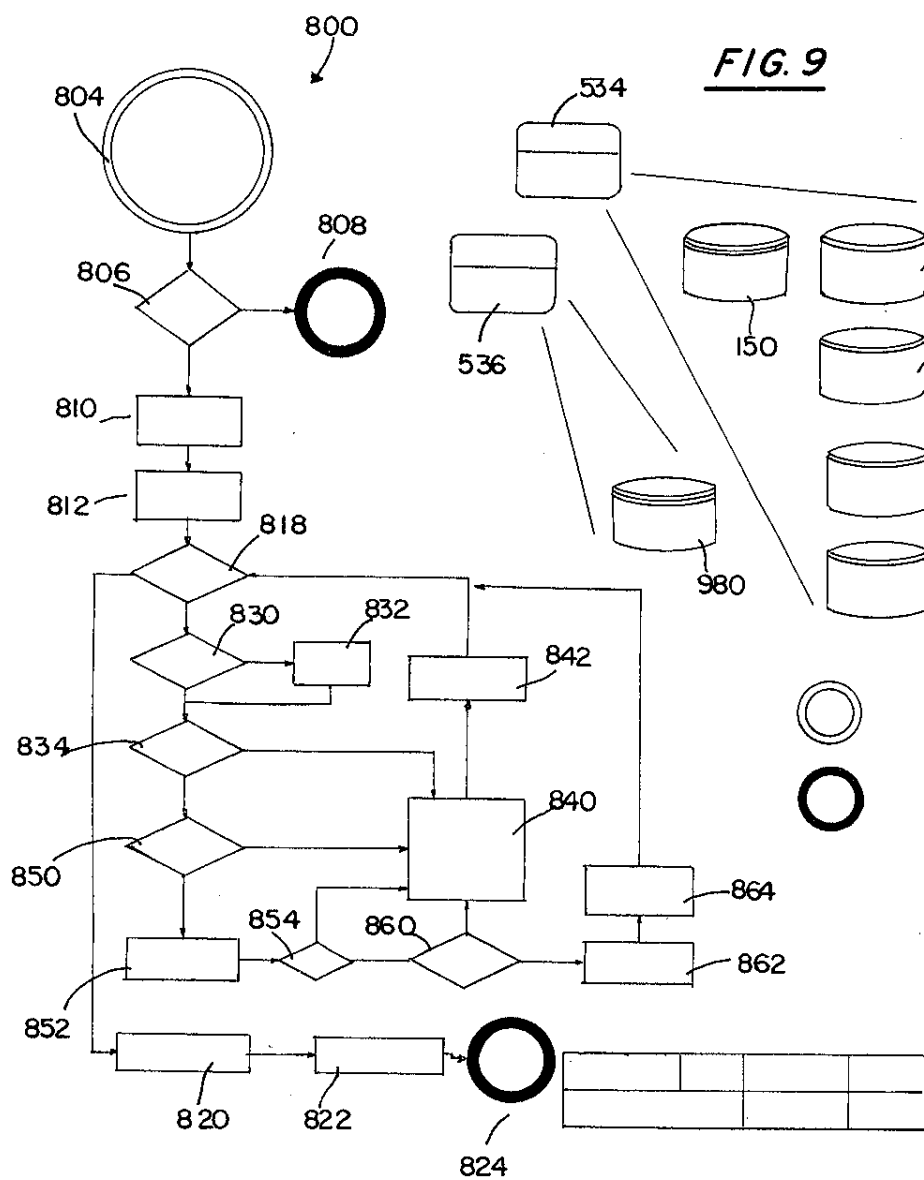
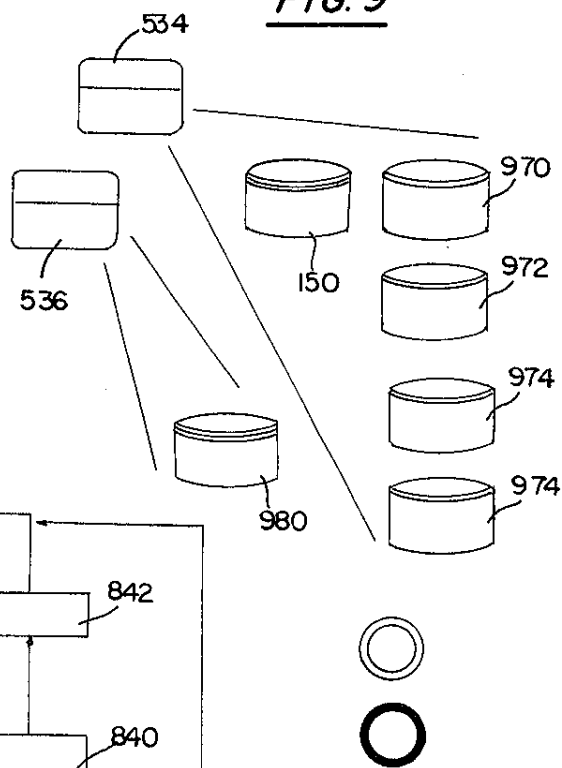
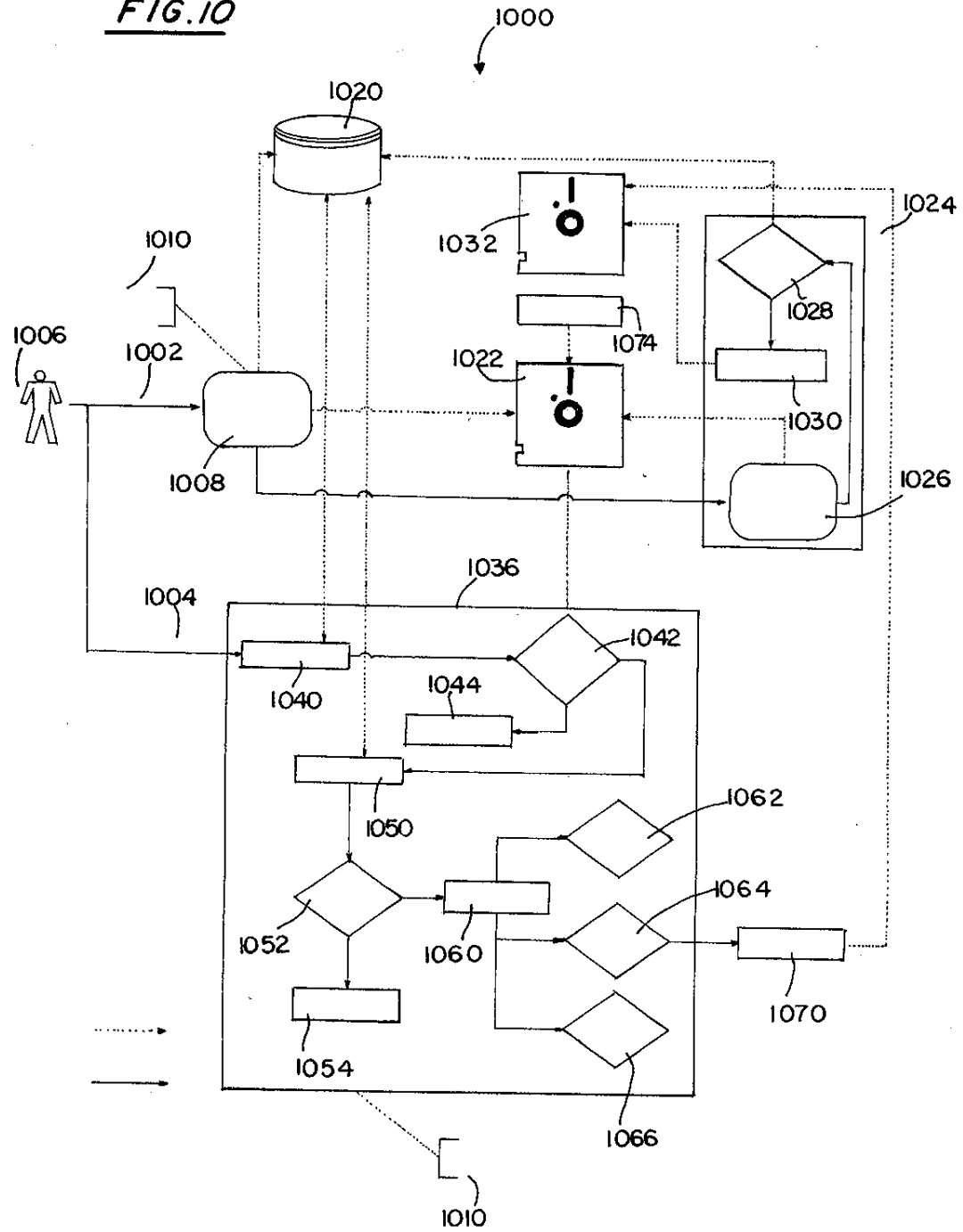


FIG. 9



9/13

FIG. 10

10/13

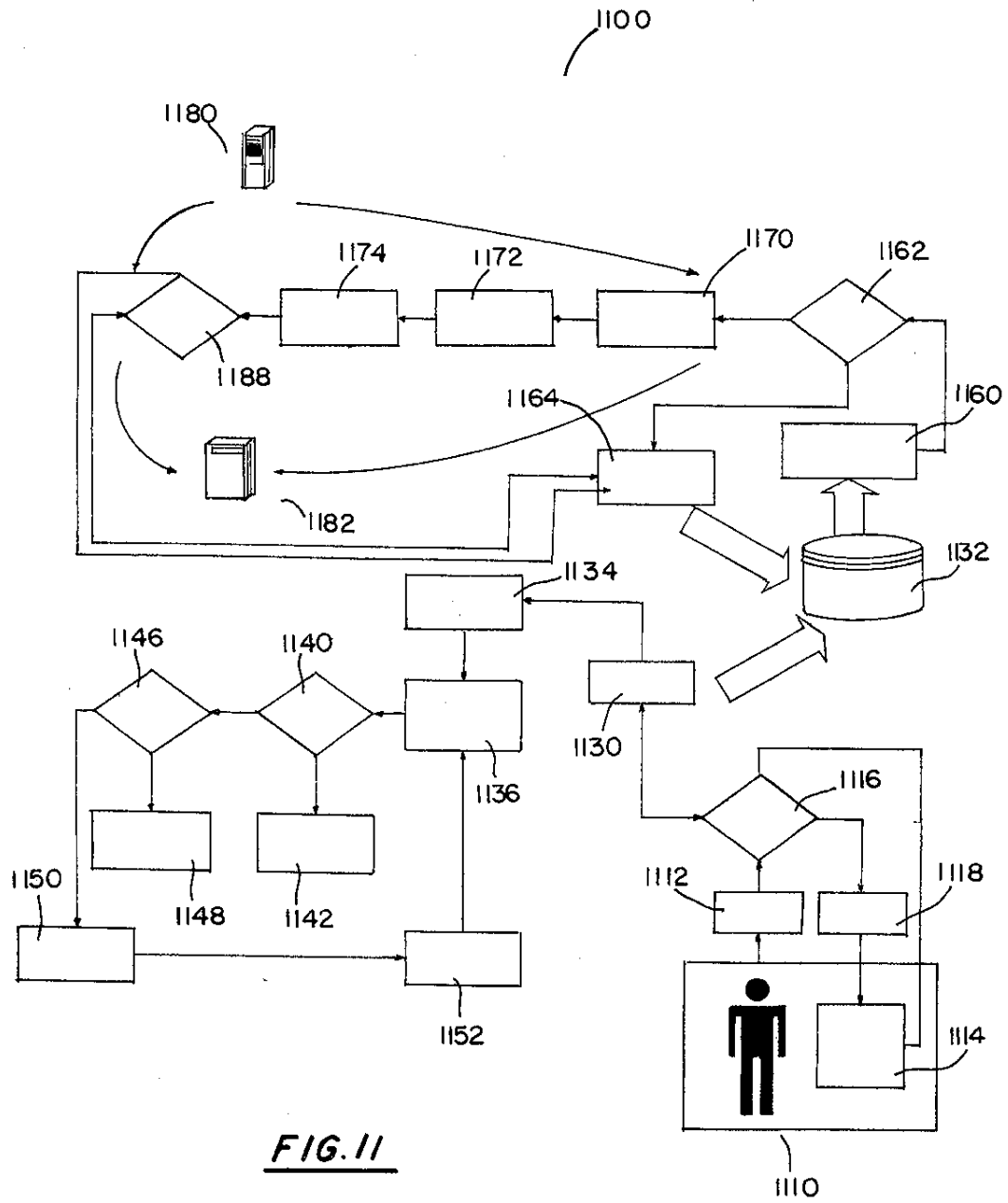


FIG. 12

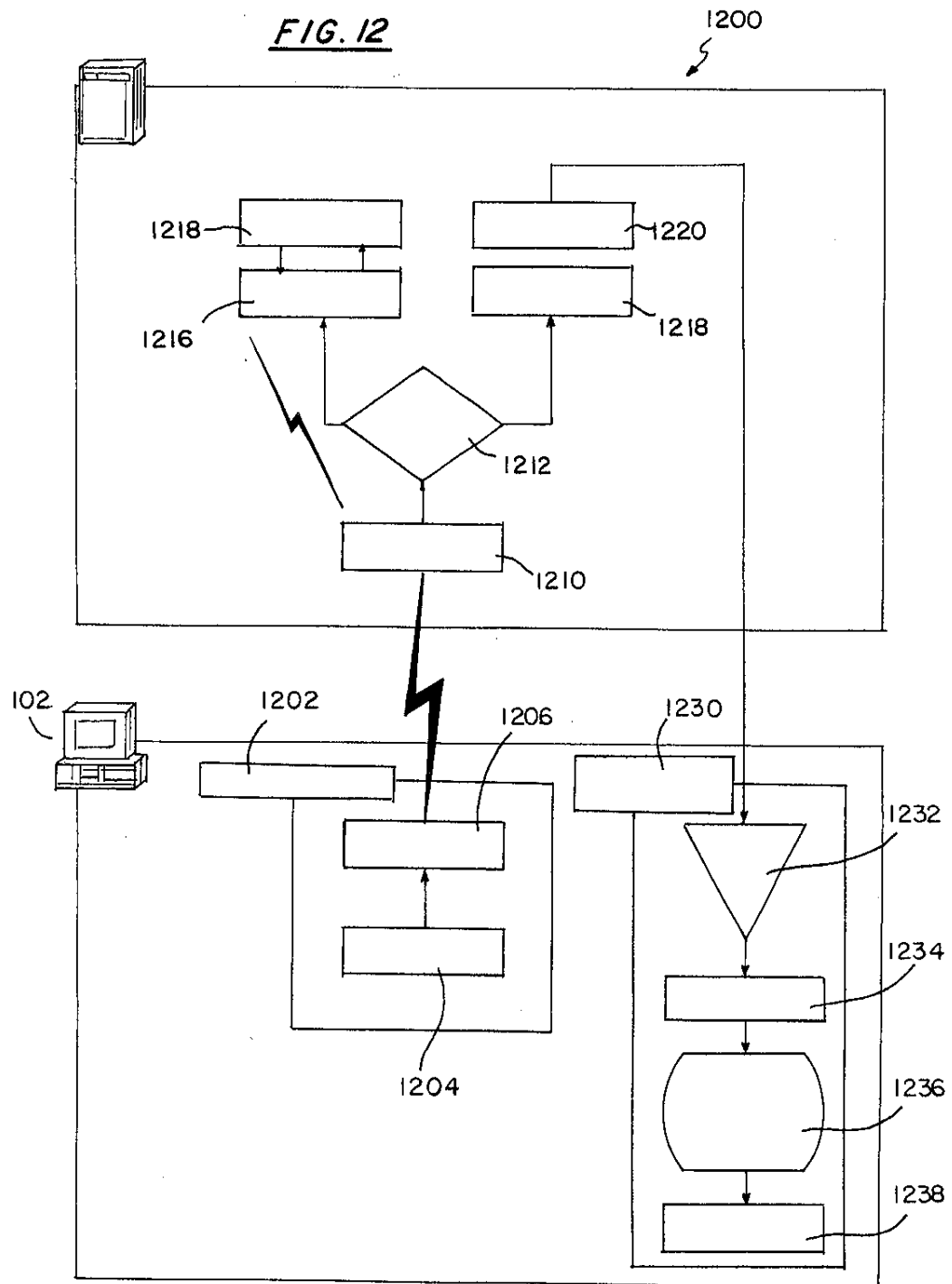
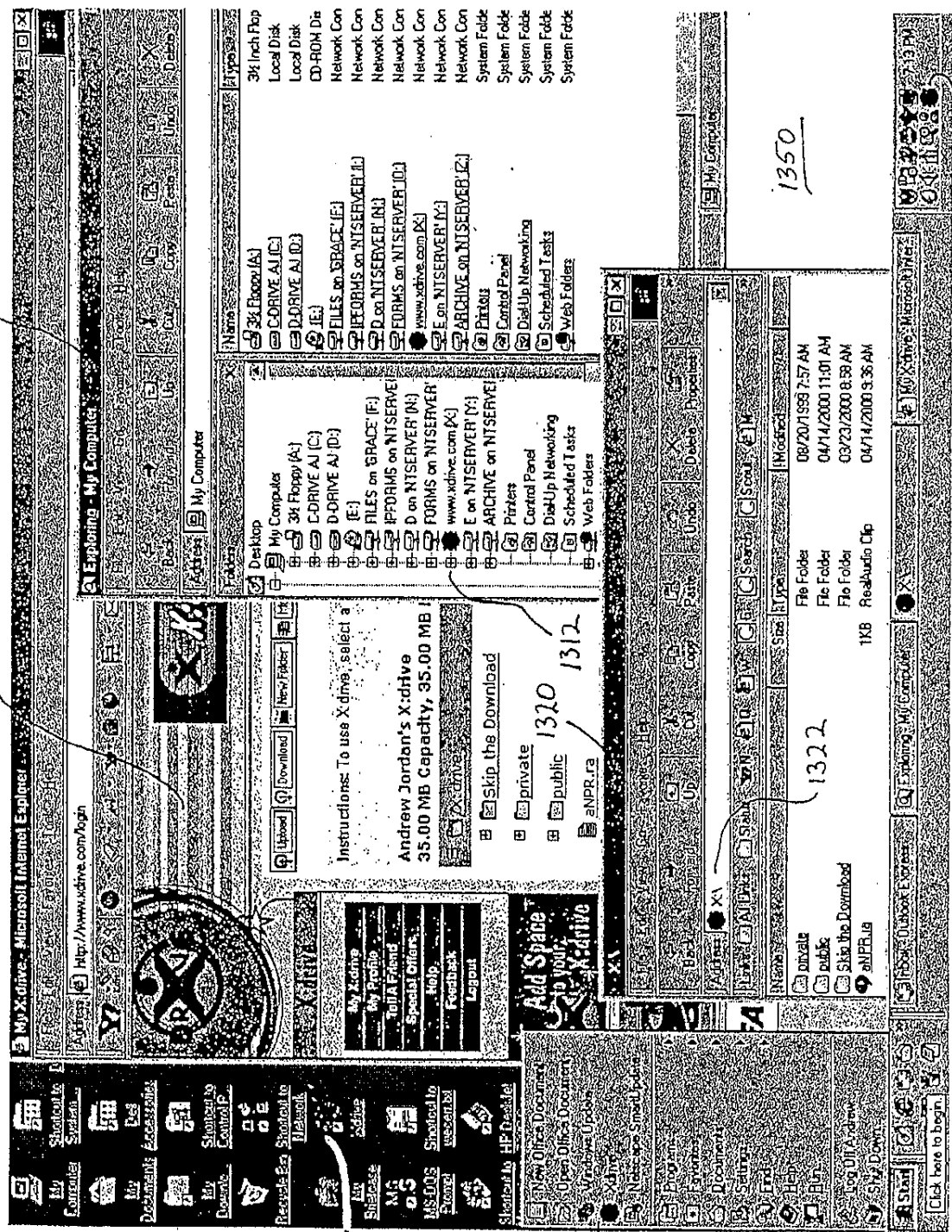


Fig. 13



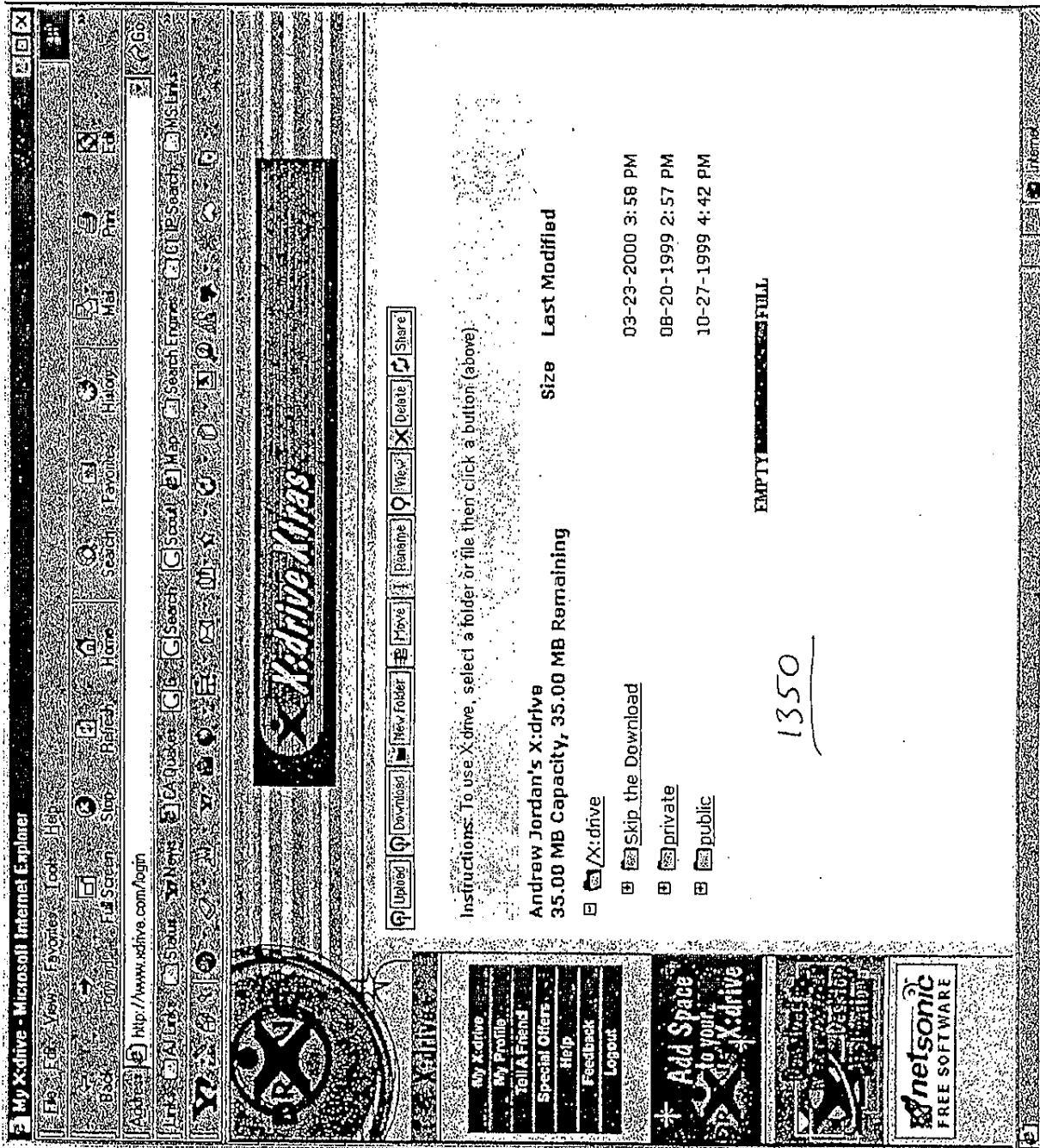


Fig. 14

RESUMO

Patente de Invenção para “**RECURSO DE
ARMAZENAMENTO DE INTERNET COMPARTILHADA,
SISTEMA DE INTERFACE COM O USUÁRIO, E
5 MÉTODO**”

Um recurso de armazenamento de Internet compartilhada provê armazenamento de arquivo baseado em Internet, recuperação, acesso, controle, e manipulação para o usuário. Adicionalmente, uma interface do usuário fácil de usar é
10 provida tanto para um browser ou aplicação de parada sozinha. O método inteiro provê meios pelos quais usuários podem estabelecer, usar, e manter arquivos na internet em uma maneira remota a partir de seu computador local ainda em uma maneira que é similar à manipulação do arquivo usado em seus
15 computadores locais. Uma capacidade alta e outros sistemas de armazenamento é ligado à internet via uma rede de internet opcional que também serve para gerar e dirigir meta-dados com relação aos arquivos armazenados. Um servidor web (140) usando um CGI, baseado em Java, ou outras interfaces transmite e
20 recupera pacotes TCP/IP ou outras informações da internet através de um balanceador/parede de carga (110) pelo uso XML para envolver os pacotes de dados. Instruções de arquivos podem ser transmitidos pela Internet pelo sistema de Recurso Compartilhado. A conta do usuário pode ter uma senha protegida
25 de maneira que somente o usuário possa acessar seus arquivos. No lado do usuário, uma aplicação do cliente de parada sozinha

- (142) ou objeto JavaScript interpretado através de um browser provendo dois meios pelos quais o XML ou outras correntes de dados de linguagem marcados podem ser recebidos e colocados para ser usado por outro usuário. Transferência de arquivo de Internet para Internet podem ser efetuado por baixa de carga diretamente para o espaço de conta do usuário.