US 20020178215A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2002/0178215 A1**

Laksono et al. (43) **Pub. Date: Nov. 28, 2002**

(54) **METHOD AND APPARATUS FOR CENTRALIZING APPLICATION ACCESS WITHIN A MULTIMEDIA SYSTEM**

(76) Inventors: **Indra Laksono**, Richmond Hill (CA); **Shahid Saleem**, Toronto (CA)

Correspondence Address:
Timothy W. Markison
P.O. Box 342019
Austin, TX 78734 (US)

(21) Appl. No.: **09/946,920**

(22) Filed: **Sep. 5, 2001**

Related U.S. Application Data

(63) Continuation-in-part of application No. 09/864,524, filed on May 24, 2001.

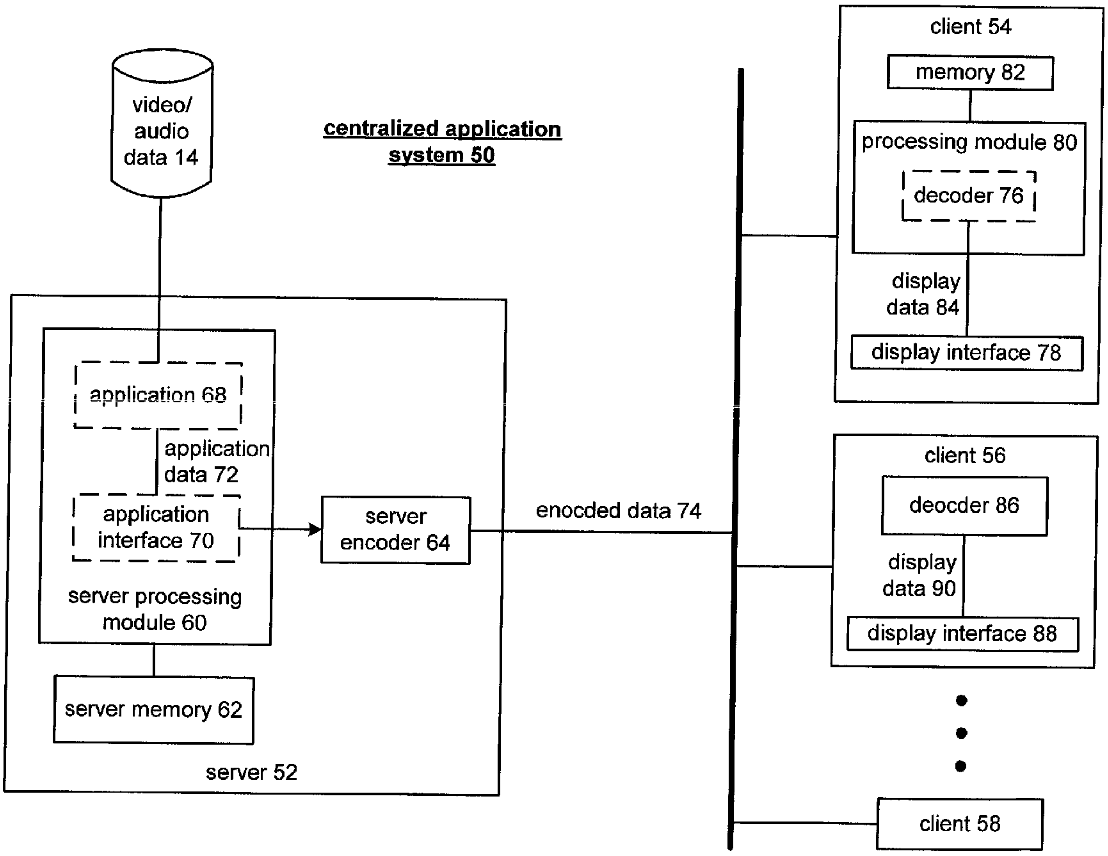
Publication Classification

(51) **Int. Cl.^{7}** **G06F 15/16**

(52) **U.S. Cl.** **709/203**; 709/231; 709/246

(57) **ABSTRACT**

A method and apparatus for providing accessing to applications in centralized application system includes processing that begins by having the server execute an application in response to a request and/or inputs by one of the clients to produce application data. The processing continues as the server provides an application interface for interfacing the application data for the requesting client to produce interface data. The processing continues when the server encodes the interface data to produce encoded data. The processing as the server provides the encoded data to the plurality of clients. The processing continues as the targeted client determines that it is being addressed by the encoded data. The processing continues as the targeted client decodes the encoded data to produce display data.



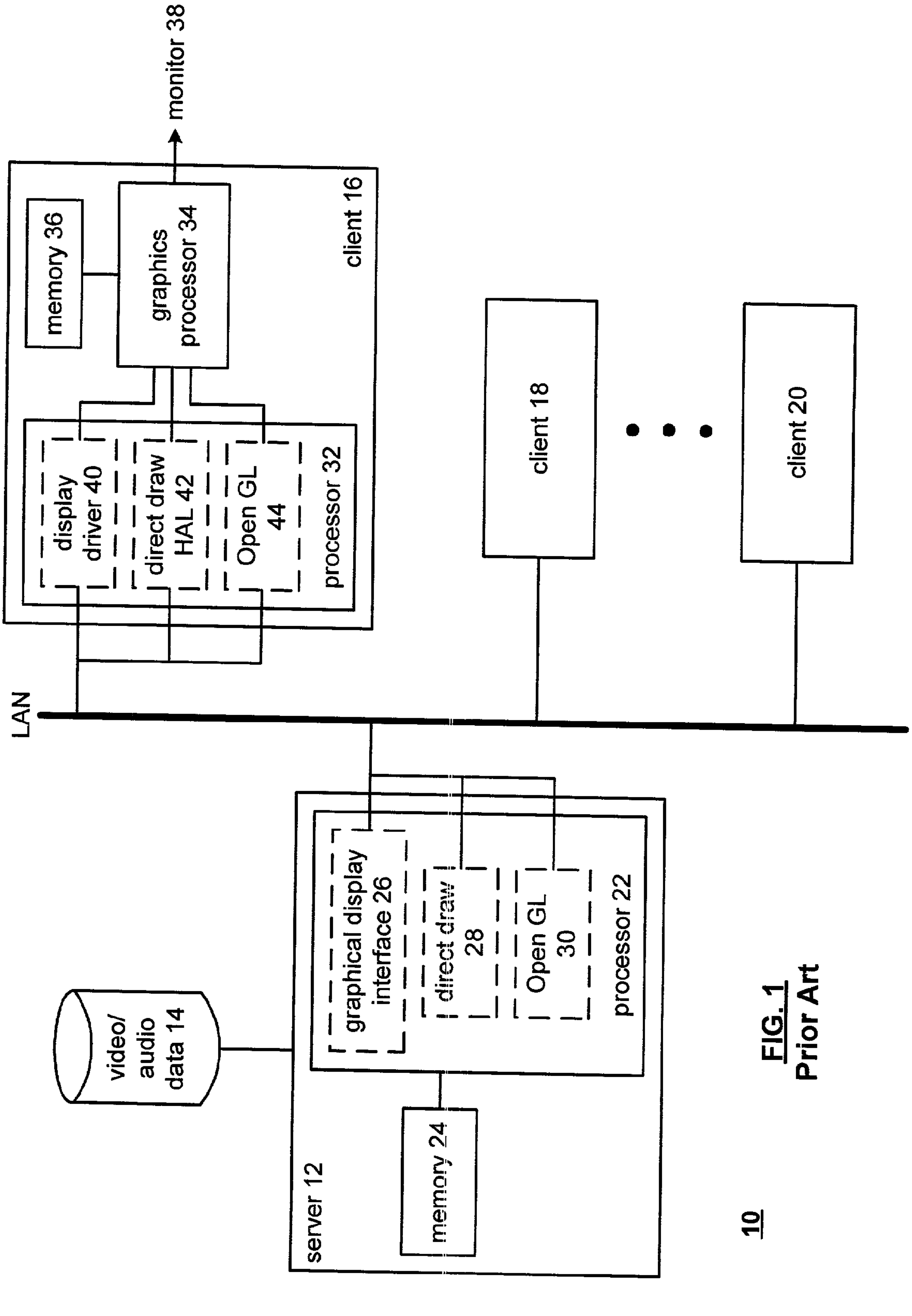
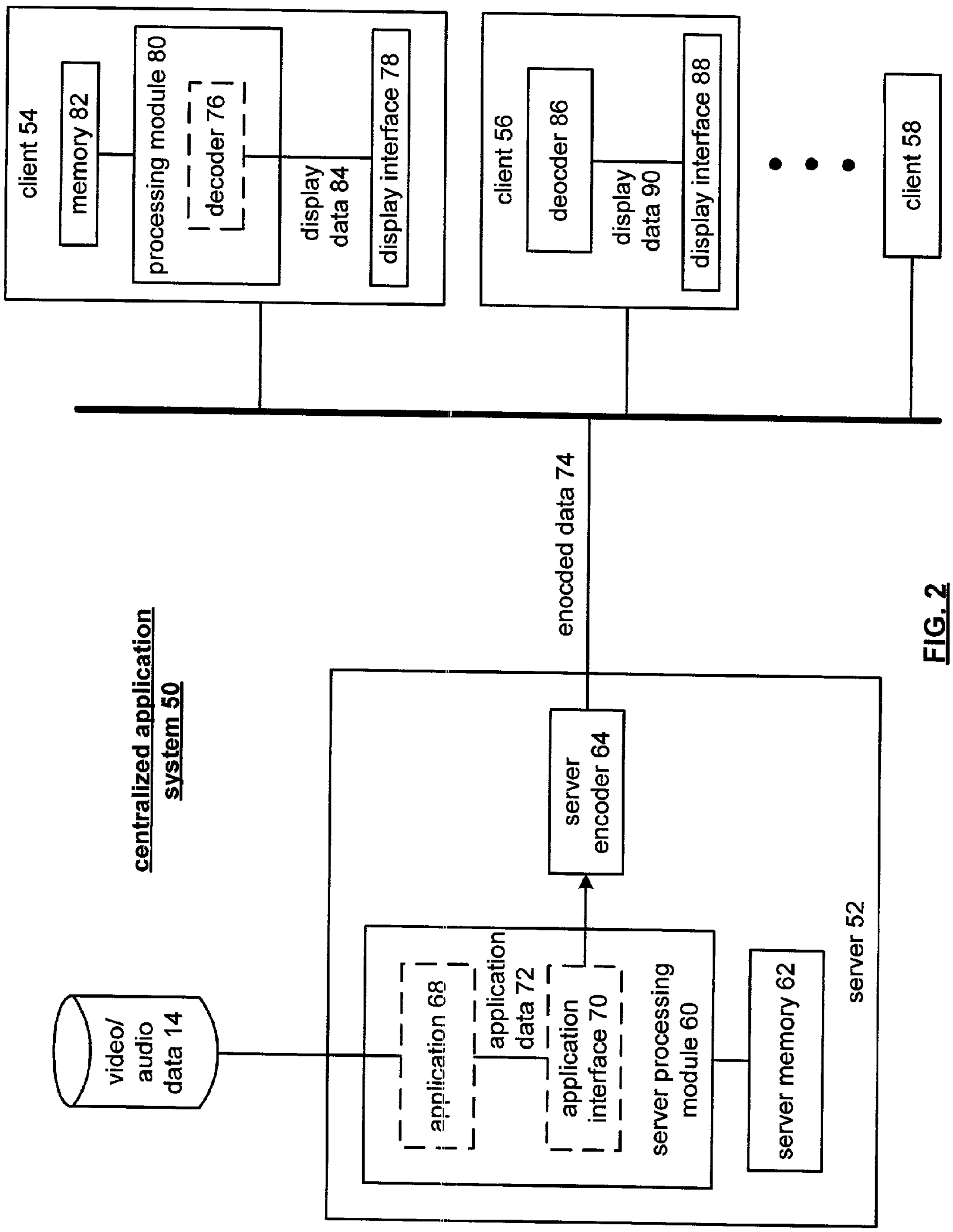


FIG. 1
Prior Art



centralized application system 50

FIG. 2

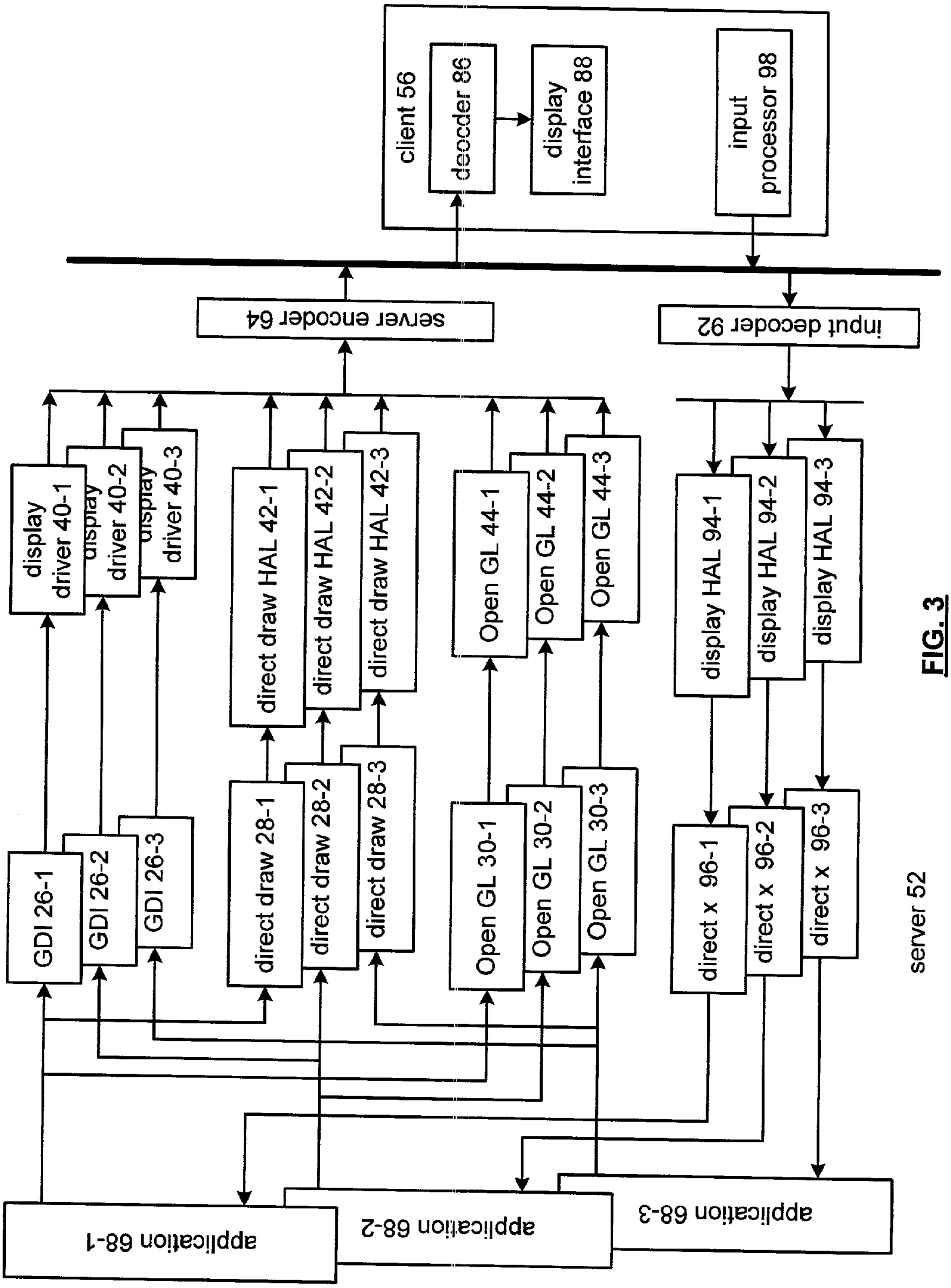

client 56
decoder 86
display interface 88
input processor 98
server encoder 64
input decoder 92
display driver 40-1
display driver 40-2
display driver 40-3
direct draw HAL 42-1
direct draw HAL 42-2
direct draw HAL 42-3
Open GL 44-1
Open GL 44-2
Open GL 44-3
display HAL 94-1
display HAL 94-2
display HAL 94-3
GDI 26-1
GDI 26-2
GDI 26-3
direct draw 28-1
direct draw 28-2
direct draw 28-3
Open GL 30-1
Open GL 30-2
Open GL 30-3
direct x 96-1
direct x 96-2
direct x 96-3
server 52
application 68-1
application 68-2
application 68-3

FIG. 3

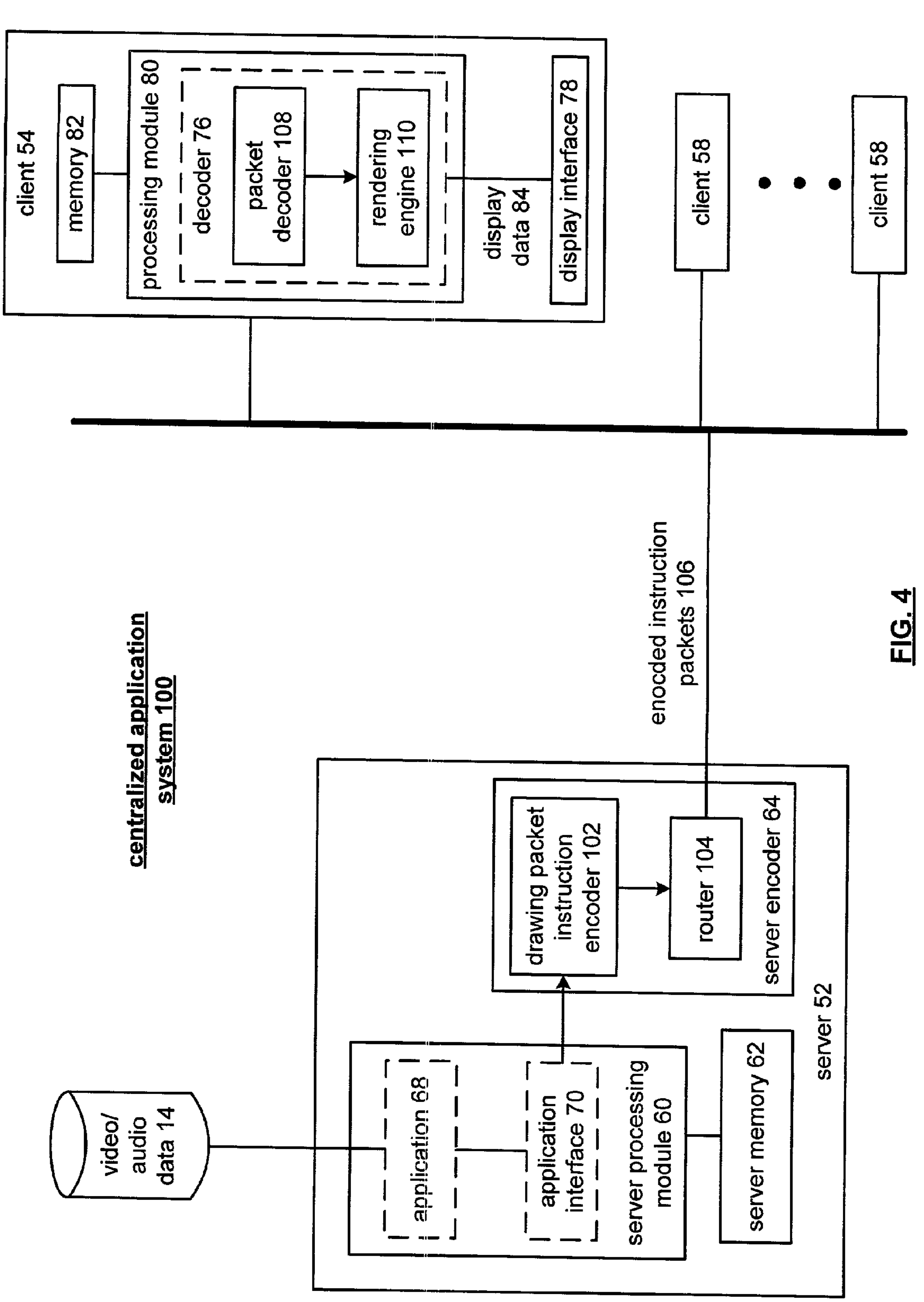

centralized application system 100
client 54
memory 82
processing module 80
decoder 76
packet decoder 108
rendering engine 110
display data 84
display interface 78
client 58
client 58
enocded instruction packets 106
video/ audio data 14
application 68
application interface 70
server processing module 60
server memory 62
drawing packet instruction encoder 102
router 104
server encoder 64
server 52

FIG. 4

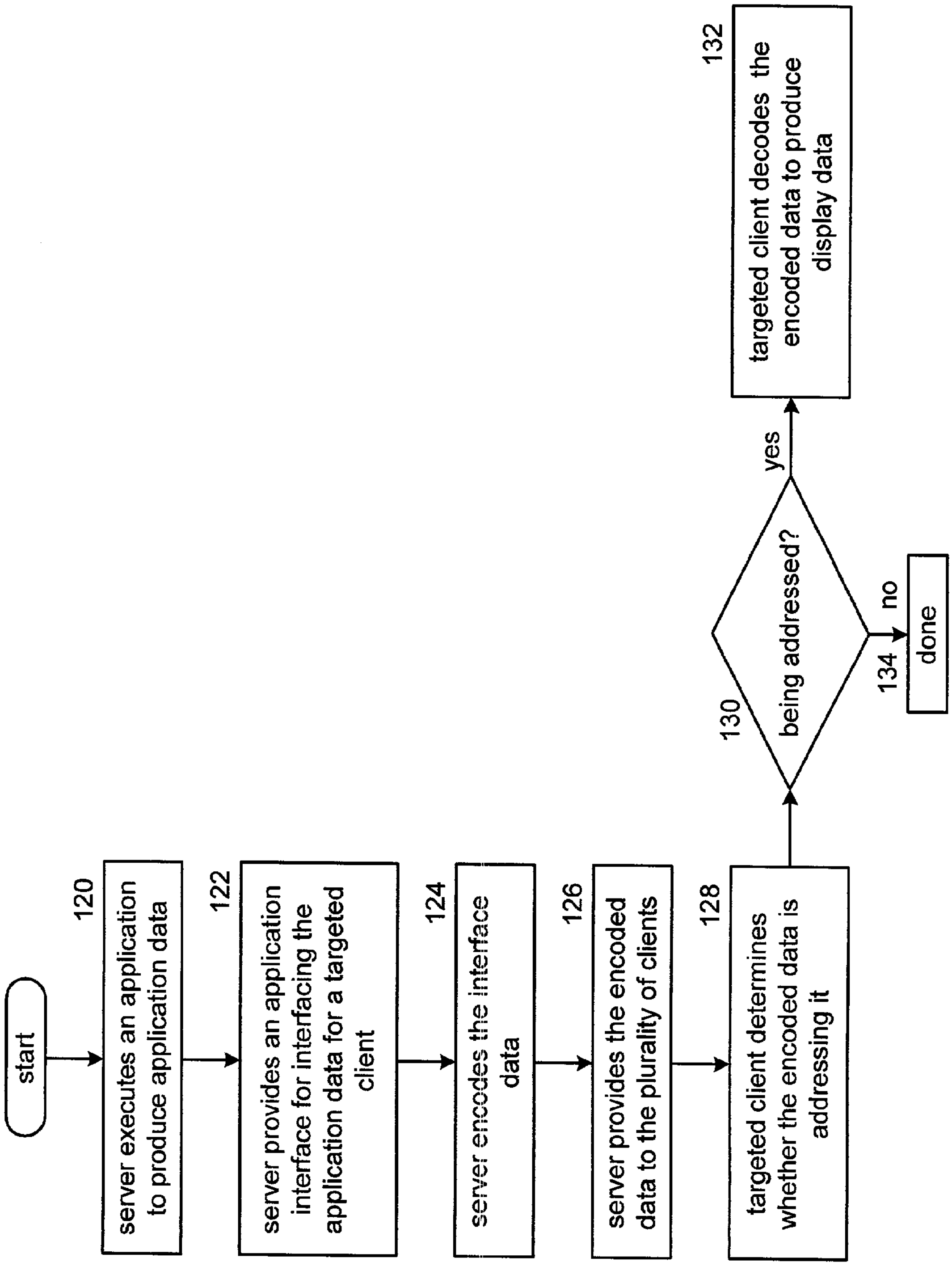

start
120
server executes an application to produce application data
122
server provides an application interface for interfacing the application data for a targeted client
124
server encodes the interface data
126
server provides the encoded data to the plurality of clients
128
targeted client determines whether the encoded data is addressing it
130
being addressed?
yes
132
targeted client decodes the encoded data to produce display data
no
134
done

FIG. 5

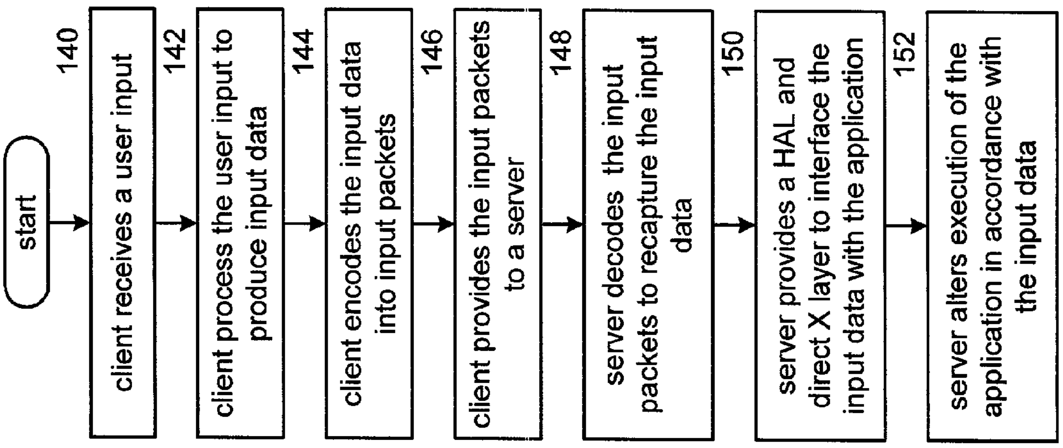

start
140
client receives a user input
142
client process the user input to produce input data
144
client encodes the input data into input packets
146
client provides the input packets to a server
148
server decodes the input packets to recapture the input data
150
server provides a HAL and direct X layer to interface the input data with the application
152
server alters execution of the application in accordance with the input data

FIG. 6

METHOD AND APPARATUS FOR CENTRALIZING APPLICATION ACCESS WITHIN A MULTIMEDIA SYSTEM

[**0001**] This patent application is a continuation-in-part of co-pending patent application entitled METHOD AND APPARATUS FOR A MULTIMEDIA SYSTEM, having a filing date of: May 24, 2001, and a serial number of: Ser. No. 09/864,524.

TECHNICAL FIELD OF THE INVENTION

[**0002**] This invention relates generally to local area networks and more particularly to a computer network that utilize a local area network.

BACKGROUND OF THE INVENTION

[**0003**] As is known, computer networks include a server operably coupled to a plurality of client computers. Such computer networks offer the advantage of sharing resources among computers, via the server, without each computer having to separately possess the particular resource. For example, the server may provide Internet access for the clients, store programs and/or data for downloading to the clients, include a database that stores media content and applications for client access, and/or provides virtual sessions for the clients.

[**0004**] In one known computer network, the server processes network traffic and provides routing functions for clients to access the Internet. As such, the server is functioning as a network server for the clients. Accordingly, each of the clients includes storage elements and processing resources to execute applications (e.g., word processing, drawing, spread sheets, web browser, etc) on data received via the server. Such a computer network may be enhanced by including file server functionality within the server to store programs and data required for a client to load and execute the application. However, the execution of the application still occurs on the client computer.

[**0005**] In another known computer network, the server creates virtual sessions with each client running terminal sessions, which are responsible only for rendering the results of the execution on the server. However, such a network is generally not suitable for relatively high bandwidth high-resolution live video rendering applications.

[**0006**] In yet another known computer network, the server contains a database that stores all relevant information about the media content and applications that might be requested by the client. Such a computer network is illustrated in **FIG. 1**. The computer network **10** includes a server **12**, a video/audio database **14**, a local area network (LAN) connection **15**, and a plurality of client computers **16-20**. The server includes a processor **22**, such as a central processing unit, and memory **24**, such as system memory. The memory **24** stores an operating system application that enables the processor **22** to perform a graphical display interface function (GDI) **26**, a direct draw function **28**, and/or an OpenGL function **30**. Each of these graphical interfacing functions is well known to provide an application interface (API) for the server when processing one or more applications stored in the video/audio database **14**.

[**0007**] The video/audio database **14** stores applications that are executed by the server on behalf of one or more of the client computers **16-20**. Such applications include, but are not limited to, video games, MP3 files, DVD files, video files (e.g., MPEG files, JPEG files, MJPEG files, etc.), word processing applications, spreadsheet applications, and drawing applications.

[**0008**] Each of the client computers **16-20** includes a processor **32**, a graphics processor **34**, and memory **36**. The memory **36** stores an operating system application that enables the processor **32** to function as a display driver **40**, a direct draw HAL (hardware application layer) **42**, and/or an OpenGL graphical interface **44**.

[**0009**] In operation, a client computer, for example client **16**, requests access to a particular application stored in the video/audio database **14**, for example a computer game. The server **12**, upon validating the request, opens the requested computer game application and begins execution thereof. As the server **12** is executing the computer game application, it is generating graphical data. The graphical data is processed by one of the graphical API's (i.e., the GDI **26**, the direct draw **28**, or the OpenGL **30**) into device independent packets, which includes the address for client **16**. The packets are provided on LAN **15** and interpreted by each of the clients **16-20**.

[**0010**] Client **16** retrieves the packets from the LAN **15** and processes them via its graphical API (i.e., the display driver **40**, the direct draw HAL **42**, or the OpenGL **44**) to retrieve the graphical data. The graphical data is then processed by the graphics processor **34** to produce display data that is provided to a monitor **38**. Note that the graphical API used by the client will correspond to the graphical API used by the server. For example, if the server used the GDI interface **26**, the client would use the display driver **40**. If the server **12** used the direct draw interface **28**, the client would use the direct draw HAL interface. And, if the server **12** used the OpenGL interface **30**, the client would use the OpenGL interface **44**. As is known, the graphical API's used by the server are device independent, while the graphical API's used by the clients are device dependent. As is also known, the server's execution of the computer game is affected by inputs provided by the client.

[**0011**] While it appears to the user of client computer **16** that all of the processing of the computer game is being done by the client computer **16**, in actuality there is a substantial amount of interaction between the server **12** and the client computer **16**. Accordingly, the server **12** requires a substantial amount of software and hardware to support the client in executing the computer game. Such software and hardware requirements increase proportionally with the amount of parallel processing the server supports for the clients. For example, if the server **12** is designed to support at least one simultaneous execution of an application for each of the clients **16-20**, it requires substantially more software and hardware than if the server **12** is designed to support only a few clients at any given time. As another example, if the application being processed for a client includes computations of physics, object occlusion and behavior, a substantial amount of hardware and memory is needed to store the corresponding data. Further, each execution of the application needs to be done in individual sessions by the server while the rendering is done on the client.

[**0012**] In addition, each client **16-20** includes a significant amount of software and hardware to function in the computer network **10**. As mentioned, each client requires a corresponding graphical API to that of the server, and will typically include multiple graphical API's, memory to store the software of the graphical API's, and a graphics processor. Such software and hardware adds cost to each of the client computers in the network. With a consumer demand for decreased cost of client computers, current implementation of computer networks is contra to market demands.

[**0013**] Therefore, a need exists for a method and apparatus that reduces software and/or hardware requirements within client devices within networks while maintaining the advantages of networking by having device and application independent rendering instructions on the client.

BRIEF DESCRIPTION OF THE DRAWINGS

[**0014**] **FIG. 1** illustrates a schematic block diagram of a prior art computer network;

[**0015**] **FIG. 2** illustrates a schematic block diagram of a centralized application system in accordance with the present invention;

[**0016**] **FIG. 3** illustrates a graphical representation of a server and a client in accordance with the present invention;

[**0017**] **FIG. 4** illustrates a schematic block diagram of an alternate centralized application system in accordance with the present invention;

[**0018**] **FIG. 5** illustrates a logic diagram of method for accessing an application in a centralized application system in accordance with the present invention; and

[**0019**] **FIG. 6** illustrates a logic diagram of an alternate method for accessing an application in a centralized application system in accordance with the present invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

[**0020**] Generally, the present invention provides a method and apparatus for providing access to applications in centralized application system that includes a server and a plurality of clients. The method and apparatus include processing that begins by having the server execute an application in response to a request and/or inputs by one of the clients to produce application data. The processing continues as the server provides an application interface for interfacing the application data for the requesting client to produce interface data. The providing of the application interface includes providing both a server application interface (e.g., a GDI, a direct draw, or the OpenGL) that functions as a device independent interface for the application data and a client application interface (e.g., a display driver, a direct draw HAL, or an OpenGL) that function as a translator for device independent commands into device dependent interface for the application data.

[**0021**] The processing continues when the server encodes the interface data to produce encoded data. The server may use any one of the Motion Picture Expert Group (MPEG) encoding protocols to perform the encoding. The processing as the server provides the encoded data to the plurality of clients. The processing continues as the targeted client determines that it is being addressed by the encoded data. The processing continues as the targeted client decodes the encoded data to produce display data. Accordingly, each of the clients includes a decoder module to decode the data, which is substantially less software and hardware than required by clients within the network of **FIG. 1**.

[**0022**] The present invention can be more fully described with reference to FIGS. **2-6**. **FIG. 2** illustrates a schematic block diagram of a centralized application system **50** that includes a server **52**, a video/audio database **14**, and a plurality of client devices **54-58**. The server **52** includes a processing module **60**, memory **62** and an encoder **64**. The encoder **64** may be an MPEG encoder, JPEG encoder, MJPEG encoder, MP3 encoder, and/or any other type of device that encoded video and/or audio data. The processing module **60** may be a single processing device or a plurality of processing devices. Such a processing device may be a microcontroller, microcomputer, microprocessor, digital signal processor, central processing unit, programmable gate array, logic circuitry, state machine, digital signal processor, and/or any device that manipulates signals (analog and/or digital) based on operational instructions. The memory **62**, which functions as a digital storage medium, may be a single memory device or a plurality of memory devices. Such a memory device may be volatile memory, nonvolatile memory, static memory, dynamic memory, read-only memory, random access memory, and/or any device that stores digital information. The memory **62** store operational instructions that, when executed by the corresponding processing module **60** causes the processing module to implement one or more of the processing steps illustrated in **FIGS. 5 and 6**, and as graphically illustrated in FIGS. **2-4**.

[**0023**] Each of the client devices **54-58**, which may be a computer, television, monitor, etc., includes a decoder **86** and a display interface **88**, or a processing module **80**, memory **82**, and a display interface **78**. The decoder **86** is the compliment of the encoder **64** employed by the server **52**. As such, the decoder **86** may be an MPEG decoder, JPEG decoder, MP3 decoder, MJPEG decoder, and/or any other device for decoding encoding video and/or audio data, such as a DVD chip set. The processing module **80** may be a single processing device or a plurality of processing devices. Such a processing device may be a microcontroller, microcomputer, microprocessor, digital signal processor, central processing unit, programmable gate array, logic circuitry, state machine, digital signal processor, and/or any device that manipulates signals (analog and/or digital) based on operational instructions. The memory **82**, which functions as a digital storage medium, may be a single memory device or a plurality of memory devices. Such a memory device may be volatile memory, nonvolatile memory, static memory, dynamic memory, read-only memory, random access memory, and/or any device that stores digital information. The memory **82** store operational instructions that, when executed by the corresponding processing module **80** causes the processing module to implement one or more of the processing steps illustrated in **FIGS. 5 and 6**, and as graphically illustrated in FIGS. **2-4**.

[**0024**] In operation, one of the clients requests access to an application stored in the video/audio database **14** by providing a request to the server **52**. Upon validating the request using conventional authentication processes of local area networks, opens the application **68** and begins executing the application **68** via the processing module **60**. As a result of executing the application **68**, the processing module **60** generates application data **72**. The application data **72** includes graphical data representing the current execution of the application **68**. Such an application **68** may be a game, a drawing application, a word processing application, a presentation application, a spreadsheet application, a web browser application, an email application, a time management application, a database application, drawing application, video application, audio application, and/or any other type of processing of video, text, and/or audio data.

[**0025**] The server **52** then processes the application data **72** via the application interface **70** routine being executed by the processing module **60** to produce interface data. In general, the application interface **70** provides a server application interface operable to provide a device independent interface for the application data and a client application interface to provide a device dependent interface for the application data. The server application interface may be a graphical display interface (GDI), while the client application interface may be a corresponding device driver. Alternatively, the server application interface may be an independent direct draw interface, while the client application interface may be at least one of direct draw hardware abstraction layer (HAL) and a direct draw three-dimensional HAL. As yet a further alternative, the server application interface and the client application interface may be an Open GL interface.

[**0026**] The server **52** then encodes the interface data, which is in a device dependent state, via the server encoder **64** to produce encoded data **74**. The server encoder **64** may be an MPEG encoder that provides the application data as MPEG packets. Alternatively, the server encoder may include a drawing packet instruction encoder and router to provide encoded instruction packets regarding the application data to the plurality of clients.

[**0027**] The server **52** provides the encoded data **74** as packets to the plurality of clients. As such, the server may include an Ethernet interface that is operably coupled to a network connection, which provides coupling to the plurality of clients. Alternatively, the server may provide the packets of data to the plurality of clients via a wireless network connection, such as a radio frequency transmission path and/or infrared transmission path.

[**0028**] Each of the clients receives the packets of application data and determines whether it is a target of the packets. To receive the packets, each client includes an Ethernet interface or a wireless interface. An Ethernet interface is operably coupled to a network connection, which provides coupling between the server and the plurality of clients. The wireless interface couples the client to the server via a wireless network connection, such as a radio frequency transmission path and/or infrared transmission path.

[**0029**] When the client is a target of the packets, the client decodes the packets to recapture display data **84** or **90**. The display data **84** or **90** is provided to a display interface **78** or **88**, which couples the client device to a display. As one of average skill in the art will appreciate, the display may be a monitor, television, computer, personal digital assistant, etc. As one of average skill will further appreciate, the client device may be incorporated in the display.

[**0030**] For example, if client **56** is the target of the packets, the client **56** decodes the packets via the decoder **86**. The type of the decoder **86** is dependent on the type of encoder used in the server. For instance, the decoder **86** may be an MPEG decoder and/or a DVD player chip set. Alternatively, the decoder **86** includes a packet decoder and a rendering engine. The packet decoder retrieves drawing instructions from the encoded instruction packets and the rendering engine generates display data from the retrieved drawing instructions.

[**0031**] If client **54** is the target of the packets, the client **54** decodes the packets via the processing module **80** functioning as a decoder **76**. The decoding function **76** may be programmed to perform as an MPEG decoder and/or a DVD player chip set. Alternatively, the decoder **76** may be programmed to retrieve drawing instructions from the encoded instruction packets and then generate display data from the retrieved drawing instructions.

[**0032**] As one of average skill in the art will appreciate, the centralized application system enables each of the client devices to be constructed of less expensive components, such as an MPEG decoder or DVD chip set, than the processing circuitry used to implement the clients of **FIG. 1**. Accordingly, the present invention enables inexpensive "thin" components to be used as the client device, thus reducing the overall cost of the network, while maintaining the quality and efficiency of application sharing.

[**0033**] **FIG. 3** illustrates a graphical representation of a server **52** and a client **56** of the system **50** of **FIG. 1**. The server **52** includes the processing module **60** and memory **62**, which function to provide a plurality of applications **68-1**, **-2**, and **-3**, a plurality of graphical display interfaces **26-1**, **-2**, and **-3**, a plurality of display drivers **40-1**, **-2**, and **-3**, a plurality of direct draw interfaces **28-1**, **-2**, and **-3**, a plurality of direct draw HAL interfaces **42-1**, **-2**, and **-3**, a plurality of OpenGL device independent interfaces **30-1**, **-2**, and **-3**, and a plurality of device dependent OpenGL interfaces **44-1**, **-2**, and **-3**. The plurality of display drivers **40-1**, **-2**, and **-3**, the plurality of direct draw HAL interfaces **42-1**, **-2**, and **-3**, and the plurality of device dependent OpenGL interfaces **44-1**, **-2**, and **-3** are coupled to the server encoder **64**.

[**0034**] The server **52** further includes an input decoder **92** operably coupled to receive inputs and/or requests from the plurality of clients via the network connection (wired or wireless). The server **52** still further includes a plurality of display HAL interfaces **94-1**, **-2**, and **-3**, and a plurality of direct x interfaces **96-1**, **-2**, and **-3**. The functioning of a display HAL interface and a direct x interface is known, thus no further discussion will be presented except to further illustrate the concepts of the present invention.

[**0035**] The client **56** includes a decoder **86**, a display interface **88** and an input processor **98**. The functionality of the decoder **86** and the display interface **88** is as previously described with reference to **FIG. 2**. The input processor **98** is operably coupled to receive inputs from an input device and provide packets of the input data to the server **52**. The input device may be a keyboard, joystick, mouse, voice recognition software, touch screen, eye tracking device, etc. The input processor **98** further includes a network interface such as an Ethernet card or a wireless network card.

[**0036**] In operation, the client **56** provides a request to access an application or provides inputs regarding the execution of the application to the server **52**. The server **52** receives the inputs via the input decoder **92**, which includes a network interface card corresponding to the network interface care included in the client **56**. Upon decoding the input, the server **52** processes the decoded input via one of the display HAL interfaces **94-1**, **-2**, and **-3** and the corresponding one of the direct x interfaces **96-1**, **-2**, and **-3**. The resulting recaptured input signal is provided to a corresponding application **68-1**, **-2**, and **-3**. As one of average skill in the art will appreciate, the server **52** includes a plurality of layers of device independent API's, applications, and device dependent API's. Each layer supports one of the clients associated with the server. Such layering may be implemented as a virtual layering such that one processing module performs all of the applications and API's and provides logical separation of the processing. Alternatively, separate processing modules may implement the layering, where each processing module processes a layer.

[**0037**] The server **52** adjusts the processing of the application **68-1**, **-2**, and **-3** based on the recaptured input signals. The resulting application data is then processed by one of the device independent API's (e.g., GDI, direct draw, or Open GL) and then the application data is processed by one of the device dependent API's (e.g., display driver, direct draw HAL, and Open GL). The interfaced processed data is provided to the server encoder **64**, which encodes the data into packets. The server **52** provides the packets to the targeted client via the wired or wireless network connection.

[**0038**] **FIG. 4** illustrates a schematic block diagram of an alternate centralized application system **100** that includes a server **52**, a video/audio database **14**, and a plurality of clients **54-58**. The server **52** includes the processing module **60**, the memory **62** and the server encoder **64**. The server encoder **64** includes a drawing packet instruction encoder **102** and a router **104**. The functionality of the processing module **60** and the memory **62** is as previously discussed with reference to **FIGS. 2 and 3**.

[**0039**] Each of the clients includes the processing module **80**, the memory **82**, and the display interface **78**. The memory **82** stores operational instructions that cause the processing module **80** to function as a decoder **76** that includes a packet decoder **108** and a rendering engine **110**.

[**0040**] In operation, the server **52** generates rendering instructions regarding the application data via the drawing packet instruction encoder **102**. The rendering instructions follow a convention video and/or audio rendering routing and include graphics data. In addition, the encoder **102** encodes the graphics data, which is provided to the clients via the router **104**. The clients **54-58** receive the encoded instruction packets **106** and determine whether it is a target of the packets **106**. When the client is a target, the client decodes the packets **106** to recapture the graphics data. The rendering engine **110** processes the graphics data to produce pixel data that is subsequently provided to a display.

[**0041**] **FIG. 5** illustrates a logic diagram of method for accessing an application in a centralized application system. The processing begins at step **120** where the server executes an application to produce application data. The application may be a game, a drawing application, a word processing application, a presentation application, a spreadsheet application, a web browser application, an email application, a time management application, and a database application. The execution of an application includes opening the application in response to a request and adjusting the execution of the application based on input signals received from the client.

[**0042**] The processing continues at step **122**, where the server provides an application interface for interfacing the application data for a targeted client to produce interface data. The providing of the application interface may be done by providing a device independent API and a device dependent API. The processing then continues at step **124**, where the server encodes the interface data to produce encoded data. The server may using an MPEG encoding scheme, drawing instruction packet encoding, or other means for encoding video and/or audio data to produce the encoded data. The processing continues at step **126** where the server provides the encoded data to the plurality of clients.

[**0043**] The processing then continues at step **128**, where each of the clients determines whether it is a target of the encoded data. The process then proceeds to step **130** where the processing branches depending on whether the client is being addressed by the packets. When the client is not addressed, the processing proceeds to step **134** where the processing is complete for this client. If, however, the client is addressed, the process proceeds to claim **132** where the client decodes the encoded data to recapture the application data. The client processes the application data to produce display data.

[**0044**] **FIG. 6** illustrates a logic diagram of an alternate method for accessing an application in a centralized application system. The processing begins at step **140** where a client receives a user input signal from a user of the client device. The user input signal may be to initiate execution of an application or providing instructions on how to manipulate the execution of the application. The processing then continues at step **142** where the client processes the user input signal to produce input data. The processing then continues at step **144** where the client encodes the input data to produce input packets. The processing then continues at step **146** where the client provides the input packets to the server.

[**0045**] The processing then proceeds to step **148** where the server decodes the input packets to recapture the input data. The processing then continues at step **150** where the server provides a HAL and Direct X application interface such that the input signals are interfaced with the application. The processing then proceeds to step **152** where the server alters execution of the application based on the input data.

[**0046**] The preceding discussion has presented a method and apparatus for reducing hardware and software requirements for client devices within local area networks. By including device independent and dependent API's in the server, the client devices may include less sophisticated circuitry to produce display data. As such, the cost of such client devices is reduced. As one of average skill in the art will appreciate, other embodiments may be derived from the teachings of the present invention without deviating from the scope of the claims.

What is claimed is:

1. A centralized application system comprises:

server; and

plurality of clients, wherein the server includes:

server processing module;

server encoder operably coupled to the server processing module; and server memory operably coupled to the server processing module, wherein the server memory stores operational instructions that cause the server processing module to:

execute an application to produce application data; and

provide an application interface for interfacing the application data to the server encoder;

wherein the server encoder encodes the application data to produce encoded data and wherein the server encoder provides the encoded data to at least one of the plurality of clients;

wherein each of the plurality of clients includes a decoder operably coupled to determine whether the encoded data is addressed for the each of the plurality of clients and, when the encoded data is addressed for the each of the plurality of clients, decode the encoded data to produce display data.

2. The centralized application system of claim 1, wherein the application interface further comprises:

server application interface operable to provide a device independent interface for the application data; and

client application interface to provide a device dependent interface for the application data.

3. The centralized application system of claim 2, wherein the server application interface further comprises a graphical display interface (GDI) and the client application interface further comprises a corresponding device driver.

4. The centralized application system of claim 2, wherein the server application interface further comprises an independent direct draw interface and the client application interface further comprises at least one of direct draw hardware abstraction layer (HAL) and a direct draw three-dimensional HAL.

5. The centralized application system of claim 2, wherein the server application interface and the client application interface further comprise an Open GL interface.

6. The centralized application system of claim 1, wherein the server processing module further comprises:

plurality of virtual processors, wherein each of the virtual processors executes an individual application to produce individual application data and provides a corresponding application interface that provides the individual application data to the server encoder, wherein the server encoder encodes the individual application data from each of the virtual processors.

7. The centralized application system of claim 1, wherein each of the plurality of clients further comprises:

input processor operably coupled to receive user inputs, to process the user inputs to produce input data, to encode the input data into input packets, and to provide the input packets to the server.

8. The centralized application system of claim 7, wherein the server further comprises:

input decoder operably coupled to recapture the input data from the input packets and to provide the input data to the server processing module, wherein the memory further comprises operational instructions that cause the server processing module to provide a display hardware abstraction layer (HAL) and direct X layer such that the executing of the application is altered in accordance with the input data.

9. The centralized application system of claim 7, wherein the each of the plurality of clients further comprises at least one of:

Ethernet interface operably coupled to a network connection to provide the input packets to the server; and

wireless transmitter operably coupled to provide the input packets via a radio frequency transmission to the server.

10. The centralized application system of claim 9, wherein the server encoder further comprises at least one of:

Ethernet interface operably coupled to the network connection to receive the input packets; and

wireless receiver operably coupled to receive the input packets via the radio frequency transmission.

11. The centralized application system of claim 1, wherein the application comprises at least one of: a game, a drawing application, a word processing application, a presentation application, a spreadsheet application, a web browser application, an email application, a time management application, and a database application.

12. The centralized application system of claim 1, wherein the server encoder further comprises an motion picture expert group (MPEG) encoder that provides the application data as MPEG packets, and wherein the decoder of each of the plurality of clients further comprises at least one of:

an MPEG decoder and DVD player chip set.

13. The centralized application system of claim 1, wherein the server encoder further comprises a drawing packet instruction encoder and router to provide encoded instruction packets regarding the application data to the plurality of clients, and wherein the decoder of each of the plurality of clients further comprises:

packet decoder operably coupled to retrieve drawing instructions from the encoded instruction packets; and

rendering engine operably coupled to produce display data based on the retrieved drawing instructions.

14. The centralized application system of claim 1, wherein the server encoder further comprises at least one of:

Ethernet interface operably coupled to a network connection to provide packets of the encoded data to the plurality of clients; and

wireless transmitter operably coupled to provide the encoded data via a radio frequency transmission to the plurality of clients.

15. The centralized application system of claim 14, wherein each of the plurality of clients further comprises at least one of:

Ethernet interface operably coupled to the network connection to receive packets of the encoded data from the server; and wireless receiver operably coupled to receive the encoded data via the radio frequency transmission from the server.

16. A server for use in a centralized application system, the server comprises:

server processing module;

server encoder operably coupled to the server processing module; and

server memory operably coupled to the server processing module, wherein the server memory stores operational instructions that cause the server processing module to:

execute an application to produce application data; and

provide an application interface for interfacing the application data to the server encoder;

wherein the server encoder encodes the application data to produce encoded data and wherein the server encoder provides the encoded data to at least one of a plurality of clients.

17. The server of claim 16, wherein the application interface further comprises:

server application interface operable to provide a device independent interface for the application data; and

client application interface to provide a device dependent interface for the application data.

18. The server of claim 17, wherein the server application interface further comprises a graphical display interface (GDI) and the client application interface further comprises a corresponding device driver.

19. The server of claim 17, wherein the server application interface further comprises an independent direct draw interface and the client application interface further comprises at least one of direct draw hardware abstraction layer (HAL) and a direct draw three-dimensional HAL.

20. The server of claim 17, wherein the server application interface and the client application interface further comprise an Open GL interface.

21. The server of claim 16, wherein the server processing module further comprises:

plurality of virtual processors, wherein each of the virtual processors executes an individual application to produce individual application data and provides a corresponding application interface that provides the individual application data to the server encoder, wherein the server encoder encodes the individual application data from each of the virtual processors.

22. The server of claim 16 further comprises:

input decoder operably coupled to recapture input data from input packets received from at least one of the plurality of clients and to provide the input data to the server processing module, wherein the memory further comprises operational instructions that cause the server processing module to provide a display hardware abstraction layer (HAL) and direct X layer such that the executing of the application is altered in accordance with the input data.

23. The server of claim 22, wherein the server encoder further comprises at least one of:

Ethernet interface operably coupled to the network connection to receive the input packets; and

wireless receiver operably coupled to receive the input packets via the radio frequency transmission.

24. The server of claim 16, wherein the server encoder further comprises an motion picture expert group (MPEG) encoder that provides the application data as MPEG packets.

25. The server of claim 16, wherein the server encoder further comprises a drawing packet instruction encoder and router to provide encoded instruction packets regarding the application data to the plurality of clients.

26. The server of claim 1, wherein the server encoder further comprises at least one of:

Ethernet interface operably coupled to a network connection to provide packets of the encoded data to the plurality of clients; and

wireless transmitter operably coupled to provide the encoded data via a radio frequency transmission to the plurality of clients.

27. A client for use in a centralized application system, the client comprises:

a decoder operably coupled to determine whether encoded data addresses the client and, when the encoded data addresses the client, decode the encoded data to produce display data;

interface operable to provide to the display data to a monitor.

28. The client claim 27 further comprises:

input processor operably coupled to receive user inputs, to process the user inputs to produce input data, to encode the input data into input packets, and to provide the input packets to the server.

29. The client of claim 27 further comprises at least one of:

Ethernet interface operably coupled to a network connection to provide the input packets to a server; and

wireless transmitter operably coupled to provide the input packets via a radio frequency transmission to the server.

30. The client of claim 27, wherein the decoder further comprises at least one of: a motion picture expert group (MPEG) decoder and DVD player chip such that the client is capable of receiving application data as MPEG packets.

31. The client of claim 27, wherein the decoder further comprises:

packet decoder operably coupled to retrieve drawing instructions from encoded instruction packets; and

rendering engine operably coupled to produce the display data based on the retrieved drawing instructions.

32. The client of claim 27 further comprises at least one of:

Ethernet interface operably coupled to a network connection to receive packets of the encoded data from a server; and

wireless receiver operably coupled to receive the encoded data via a radio frequency transmission from the server.

33. A method for accessing an application in a centralized application system, the method comprises:

executing an application to produce application data;

providing an application interface for interfacing the application data for a targeted client of a plurality of clients to produce interface data;

encoding the interface data to produce encoded data;

providing the encoded data to the plurality of clients;

determining, by the targeted client, whether the encoded data is addressing the targeted client; and

when the encoded data is addressing the targeted client, decoding, by the targeted client, the encoded data to produce display data.

34. The method of claim 33, wherein the providing an application interface further comprises:

providing a server application interface that functions as a device independent interface for the application data; and

providing a client application interface that function as a device dependent interface for the application data.

35. The method of claim 33 further comprises:

executing a plurality of applications to produce a plurality of application data;

providing a plurality of application interfaces for interfacing the plurality of application data for the plurality of clients to produce a plurality of interface data; and

encoding the plurality of interface data to produce the encoded data.

36. The method of claim 33 further comprises:

receiving, by the client, user inputs;

processing the user inputs to produce input data;

encoding the input data into input packets; and

providing the input packets to a server.

37. The method of claim 36 further comprises:

decoding, by the sever, the input packets to recapture the input data;

providing, by the server, a display hardware abstraction layer (HAL) and direct X layer to interface the input data with the application; and

altering execution of the application in accordance with the input data.

38. The method of claim 33, wherein the application comprises at least one of: a game, a drawing application, a word processing application, a presentation application, a spreadsheet application, a web browser application, an email application, a time management application, and a database application.

39. A method for supporting access of an application in a centralized application system, the method comprises:

executing an application to produce application data;

providing an application interface for interfacing the application data for a targeted client of a plurality of clients to produce interface data;

encoding the interface data to produce encoded data; and

providing the encoded data to the plurality of clients.

40. The method of claim 39, wherein the providing an application interface further comprises:

providing a server application interface that functions as a device independent interface for the application data; and

providing a client application interface that function as a device dependent interface for the application data.

41. The method of claim 39 further comprises:

executing a plurality of applications to produce a plurality of application data;

providing a plurality of application interfaces for interfacing the plurality of application data for the plurality of clients to produce a plurality of interface data; and

encoding the plurality of interface data to produce the encoded data.

42. The method of claim 39 further comprises:

receiving input packets for the target client, wherein the target client processes user inputs to produce input data and encodes the input data to produce the input packets;

decoding the input packets to recapture the input data; providing a display hardware abstraction layer (HAL) and direct X layer to interface the input data with the application; and

altering execution of the application in accordance with the input data.

43. A method for a client of a plurality of clients to access an application in a centralized application system, the method comprises:

determining whether encoded data is addressing the client, wherein a server executes an application to produce application data, provides an application interface for interfacing the application data to produce interface data, and encodes the interface data to produce the encoded data; and

when the encoded data is addressing the targeted client, decoding the encoded data to produce display data.

44. The method of claim 43 further comprises:

receiving user inputs;

processing the user inputs to produce input data;

encoding the input data into input packets; and

providing the input packets to the server.

45. A digital storage medium for storing operational instructions that cause a processing module to access an application in a centralized application system, the digital storage medium comprises:

first storage section for storing operational instructions that cause the processing module to execute an application to produce application data;

second storage section for storing operational instructions that cause the processing module to provide an application interface for interfacing the application data for a targeted client of a plurality of clients to produce interface data;

third storage section for storing operational instructions that cause the processing module to encode the interface data to produce encoded data; and

fourth storage section for storing operational instructions that cause the processing module to provide the encoded data to the plurality of clients.

46. The digital storage medium of claim 45, wherein the second section for storing operational instructions that cause the processing module to provide an application interface further comprises operational instructions that cause the processing module to:

provide a server application interface that functions as a device independent interface for the application data; and

provide a client application interface that function as a device dependent interface for the application data.

47. The digital storage medium of claim 45 further comprises:

fifth storage section for storing operational instructions that cause the processing module to execute a plurality of applications to produce a plurality of application data;

sixth storage section for storing operational instructions that cause the processing module to provide a plurality of application interfaces for interfacing the plurality of application data for the plurality of clients to produce a plurality of interface data; and

seventh storage section for storing operational instructions that cause the processing module to encode the plurality of interface data to produce the encoded data.

48. The digital storage medium of claim 45 further comprises:

fifth storage section for storing operational instructions that cause the processing module to receive input packets for the target client, wherein the target client processes user inputs to produce input data and encodes the input data to produce the input packets;

sixth storage section for storing operational instructions that cause the processing module to decode the input packets to recapture the input data;

seventh storage section for storing operational instructions that cause the processing module to provide a display hardware abstraction layer (HAL) and direct X layer to interface the input data with the application; and

eighth storage section for storing operational instructions that cause the processing module to alter execution of the application in accordance with the input data.

49. A digital storage medium for storing operational instructions that cause a client to access an application in a centralized application system, the digital storage medium comprises:

first storage section for storing operational instructions that cause the processing module to determine whether encoded data is addressing the client, wherein a server executes an application to produce application data, provides an application interface for interfacing the application data to produce interface data, and encodes the interface data to produce the encoded data; and

second storage section for storing operational instructions that cause the processing module to, when the encoded data is addressing the targeted client, decode the encoded data to produce display data.

50. The digital storage medium of claim 49 further comprises:

third storage section for storing operational instructions that cause the processing module to receive user inputs;

fourth storage section for storing operational instructions that cause the processing module to process the user inputs to produce input data;

fifth storage section for storing operational instructions that cause the processing module to encode the input data into input packets; and

sixth storage section for storing operational instructions that cause the processing module to provide the input packets to the server.

* * * * *