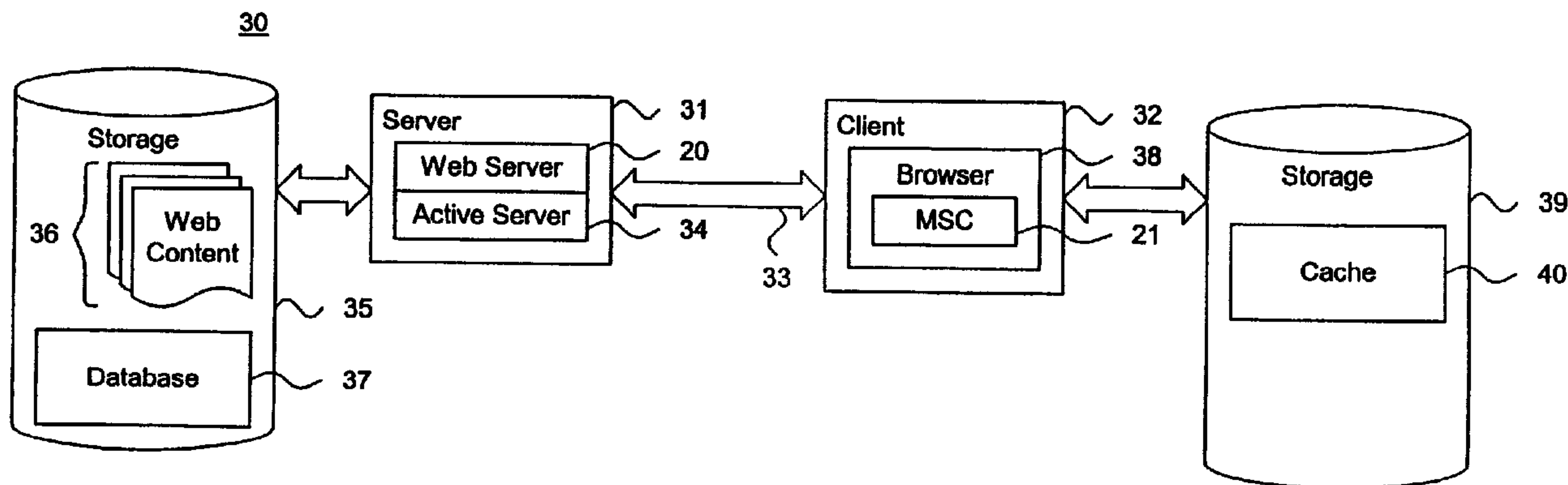




(22) Date de dépôt/Filing Date: 2001/03/09
 (41) Mise à la disp. pub./Open to Public Insp.: 2001/09/09
 (30) Priorités/Priorities: 2000/03/09 (60/188,076) US;
 2000/04/27 (09/560,684) US

(51) Cl.Int.⁷/Int.Cl.⁷ H04N 7/173, H04N 5/765, H04L 12/16
 (71) Demandeur/Applicant:
 ETHEREAL MINDS, INC., US
 (72) Inventeurs/Inventors:
 EILERS, TERRY L., US;
 NELSON, THOMAS, US
 (74) Agent: OSLER, HOSKIN & HARCOURT LLP

(54) Titre : SYSTEME ET METHODE DE PRECHARGEMENT DES DONNEES EN IIMAGERIE DANS UN ENVIRONNEMENT DE PRESENTATION MULTIMEDIA INTERACTIF
 (54) Title: SYSTEM AND METHOD FOR PRE-LOADING STILL IMAGERY DATA IN AN INTERACTIVE MULTIMEDIA PRESENTATION ENVIRONMENT



(57) **Abrégé/Abstract:**

A system and a method for efficiently pre-loading still imagery data in an interactive presentation environment. A database is organized into a plurality of records. Each record refers to one of a video file, a still image file, and a user record. Each video file is logically associated with at least one image file and at least one user record. A selection of one such video file is processed. The selected video file record is retrieved from the database. Each user record associated with the selected video file is also retrieved. The user record is updated to reflect the selection of the selected video file. The images files logically associated with the selected video file are pre-loaded. Each still image file associated with the selected video file is iteratively retrieved and downloaded into a cache. A signal is generated upon completion of the iterative retrieval of substantially all of the still image files associated with the selected video file. The selected video file is played back within the interactive presentation environment responsive to the completion signal. The cached still image files are displayed in synchronization with the playback.

**SYSTEM AND METHOD FOR PRE-LOADING STILL IMAGERY DATA
IN AN INTERACTIVE MULTIMEDIA PRESENTATION
ENVIRONMENT**

Abstract

5 A system and a method for efficiently pre-loading still imagery data in an
interactive presentation environment. A database is organized into a plurality of
records. Each record refers to one of a video file, a still image file, and a user
record. Each video file is logically associated with at least one image file and at
least one user record. A selection of one such video file is processed. The
10 selected video file record is retrieved from the database. Each user record
associated with the selected video file is also retrieved. The user record is
updated to reflect the selection of the selected video file. The images files
logically associated with the selected video file are pre-loaded. Each still image
file associated with the selected video file is iteratively retrieved and downloaded
15 into a cache. A signal is generated upon completion of the iterative retrieval of
substantially all of the still image files associated with the selected video file. The
selected video file is played back within the interactive presentation environment
responsive to the completion signal. The cached still image files are displayed in
synchronization with the playback.

5

Patent Application
Docket No. 006.0148.01

10

**SYSTEM AND METHOD FOR PRE-LOADING STILL IMAGERY DATA
IN AN INTERACTIVE MULTIMEDIA PRESENTATION
ENVIRONMENT**

15

Field of the Invention

The present invention relates in general to interactive multimedia presentation environments and, in particular, to a system and method for pre-loading still imagery data in an interactive multimedia presentation environment.

Background of the Invention

20

In general, multimedia refers to those forms of communication utilizing several different types of media, such as audio and still imagery, organized in an integrated manner. To be most effective, the individual media elements should be complementary to and integrated with each other to help communicate a cohesive message, rather than presenting a disconnected collection of individual ideas.

25

In the computer arts, multimedia refers to an arrangement of several forms of electronic media representing conventional media, such as written text, analog video and audio, and moving and still imagery, or their digital facsimiles. As

used in the present patent application, the term "multimedia" will henceforth refer only to those computer-implemented arrangements of electronic media.

In particular, multimedia usage has become quite popular as content integrated into Web pages. A "Web" page, shorthand for Worldwide Web page, is an interpretable script, including multimedia content, written in a tag-delimited language, such as the Hypertext Markup Language (HTML) or the Extensible Markup Language (XML). Individual Web pages are downloaded from Web servers over some form of interconnection and interpreted by Web browsers executing on individual clients. The speed of the interconnection, generally measured in bandwidth, is critical to Web page loading and viewing times.

Web pages can be used both in stand-alone and in distributed computing environments, including both local area networks or "intranetworks" and wide area networks or "internetworks," such as the Internet, as well as in other settings, such as television, movie theaters, or in any similar format which includes some type of display, preferably with interactive capabilities. Web pages, HTML, XML, and related concepts are described generally in R. Orfali, "Client/Server Survival Guide," Chs. 26-29, John Wiley & Sons, Inc. (3d ed. 1999), the disclosure of which is incorporated herein by reference.

Web pages can contain video content, such as provided through streaming video. Video content, including audio and video tracks, is divided into a set of small data packets which are sent as a data stream to Web browsers for viewing. The video content is available for viewing as the packets are received, thereby avoiding the potentially significant delay which would otherwise be incurred for a full download. The alternative to streaming video is live video. As used herein, "video" refers to moving imagery which preferably includes an audio soundtrack.

Video content resolution is largely dependent on the available bandwidth of the underlying network. Generally, higher network bandwidth results in higher resolution. Lower resolution video content represents a compromise between network bandwidth and the need for clear video images. Furthermore, for a large user base, the availability of higher resolution content can be constrained by the

lowest network bandwidth that is commonly available. Presenting multimedia content to users who lack the necessary network bandwidth for higher resolution video content can be achieved by providing a presentation environment which augments lower resolution video content with still images. To be effective,
5 though, the video and still imagery content must be synchronized and coordinated whereby portions of the video are associated with particular still images in some logical time sequence.

Two prior art approaches to providing integrated streaming video and still imagery content are the RealPresenter Plus G2 product, licensed by
10 RealNetworks, Seattle, Washington, and T.A.G. Composer 2.0 for RealSystem G2 product, licensed by Extend Media, Toronto, ON, Canada. The RealPresenter product enables the addition of audio and video content to a slideshow presentation generated by PowerPoint, an automated presentation creation product, licensed by Microsoft Corporation, Redmond, Washington. The T.A.G.
15 Composer products enables the integration and synchronization of a wide range of medias in a video-driven manner. The RealPresenter product builds on the existing PowerPoint presentation environment by augmenting slides with audio and video content which is played back during the slideshow. For both products, connection speed can be selected, but video, audio, and still imagery content are
20 all simultaneously downloaded. On low speed connections, the content can be rendered virtually unviewable due to the amount of data exceeding the capacity of the available bandwidth, potentially causing jittery video and audio, incomplete images, and similar concerns. The content could also take an unacceptably long time to download.

25 Therefore, there is a need for an efficient approach to downloading multimedia content specifically combining video and still imagery content, particularly over low bandwidth connections. Preferably, the video and still images would appear in separate yet coordinated viewing panels with smooth video playback. Moreover, the video and still imagery content should be

downloadable and widely usable with minimal impact on network bandwidth requirements independent of still image resolution.

Summary of the Invention

The present invention provides an image pre-loading system and method
5 that maximizes available network bandwidth. Upon the selection of a video,
descriptive information regarding the selected video is retrieved from a database
maintained on a server. User and video records are updated to reflect usage. The
image pre-loader is logically defined by downloadable script. Each image file
10 corresponding to the selected video is downloaded and stored into a local cache as
a staged image file. Upon completion of the image file download, a signal is sent
to the server to request video playback. The image files are then retrieved from
the cache by index and displayed concurrently to the video playback.

An embodiment of the present invention is a system and a method for
efficiently pre-loading still imagery data in an interactive presentation
15 environment. A database is organized into a plurality of records. Each record
refers to one of a video file, a still image file, and a user record. Each video file is
logically associated with at least one image file and at least one user record. A
selection of one such video file is processed. The selected video file record is
retrieved from the database. Each user record associated with the selected video
20 file is also retrieved. The user record is updated to reflect the selection of the
selected video file. The images files logically associated with the selected video
file are pre-loaded. Each still image file associated with the selected video file is
iteratively retrieved and downloaded into a cache. A signal is generated upon
completion of the iterative retrieval of substantially all of the still image files
25 associated with the selected video file. The selected video file is played back
within the interactive presentation environment responsive to the completion
signal. The cached still image files are displayed in synchronization with the
playback.

Still other embodiments of the present invention will become readily
30 apparent to those skilled in the art from the following detailed description,

wherein is described embodiments of the invention by way of illustrating the best mode contemplated for carrying out the invention. As will be realized, the invention is capable of other and different embodiments and its several details are capable of modifications in various obvious respects, all without departing from the spirit and the scope of the present invention. Accordingly, the drawings and detailed description are to be regarded as illustrative in nature and not as restrictive.

Brief Description of the Drawings

FIGURE 1 is a functional block diagram showing a distributed computing environment with multiple instances of a multi-screen clinic;

FIGURE 2 is a block diagram showing a system for pre-loading still imagery data in an interactive multimedia presentation environment in accordance with the present invention;

FIGURE 3 is a block diagram showing the functional software modules of a server for use in the system of FIGURE 2;

FIGURE 4 is a block diagram showing the functional software modules of a client for use in the system of FIGURE 2;

FIGURE 5 is a screen shot showing, by way of example, an integrated presentation space with navigational controls exported by the client of FIGURE 4;

FIGURE 6 is a screen shot showing, by way of example, an integrated presentation space performing an image content pre-load exported by the client of FIGURE 4;

FIGURE 7 is a screen shot showing, by way of example, an integrated presentation space performing a video content playback exported by the client of FIGURE 4;

FIGURE 8 is a flow diagram showing a method for pre-loading still imagery data in an interactive multimedia presentation environment in accordance with the present invention;

FIGURE 9 is a flow diagram showing the routine for updating a selected video record for use in the method of FIGURE 8;

FIGURE 10 is a flow diagram showing the routine for pre-loading images for use in the method of FIGURE 8; and

FIGURE 11 is a flow diagram showing the routine for playing a video for use in the method of FIGURE 8.

5

Detailed Description

FIGURE 1 is a functional block diagram showing a distributed computing environment 10 with multiple instances of a multi-screen clinic (MSC) 21. The multi-screen clinic 21 is an integrated presentation environment exported within a browser application executing on a client, as further described below beginning
10 with reference to FIGURE 4. In the described embodiment, each system operating within the distributed computing environment 10 implements a Transmission Control Protocol/Internet Protocol (TCP/IP) network stack, although other preferably packet-based network protocols could equally apply.

Within the distributed computing environment 10, a remote server 11
15 provides conventional network services, such as file access, remote system access, and content provision, to various clients. Typically, these types of network services are available through standardized TCP/IP protocols, such as the File Transport Protocol (FTP), Telnet, and Hypertext Transport Protocol (HTTP). In particular, the remote server 11 includes a Worldwide Web ("Web") server 20
20 with which to serve Web content in the form of Web pages.

Individual clients are communicatively interfaced to the remote server 11. By way of illustration, several types of clients are interfaced to the remote server 11 via a plurality of interfacing means and having various network bandwidth capabilities. For instance, a dedicated client 12 is interfaced via a dedicated,
25 direct connection to the remote server 11, such as via a serial interface. Typically, serial interfaces, depending upon type, offer network bandwidth in the range of 75 baud to 115.2 Kbaud. As well, the dedicated client 12 could be logically interfaced as part of an integrated system 9 including both the remote server 11 and dedicated client 12 operating on the same physical machine. Also, a dial-up
30 client 13 is interfaced via a modem 14. Typically, modems, depending upon type,

offer network bandwidth in the range of 300 baud to 56.6 Kbaud. Similarly, a remote client 15 is interfaced via an internetwork 16, such as the Internet. Typically, internetwork connections, depending upon type, offer network bandwidth up to 1.544 Mbps on a T1 carrier or 45 Mbps on a T3 carrier. In addition, a network client 17 is configured to operate within an intranetwork 18 locally serviced by a local server 19 and is interfaced to the remote server 11 via the internetwork 16. Typically, intranetwork connections, depending upon type, internally offer network bandwidth up to 100 Mbps. Other network topologies and configurations of computational resources, including various combinations of intranetworks and internetworks, are possible.

The individual computer systems included in the distributed computing environment 10 are general purpose, programmed digital computing devices consisting of a central processing unit (CPU), random access memory (RAM), non-volatile secondary storage, such as a hard drive or CD ROM drive, network interfaces, and peripheral devices, including user interfacing means, such as a keyboard and display. Program code, including software programs, and data are loaded into the RAM for execution and processing by the CPU and results are generated for display, output, transmittal, or storage.

In a further embodiment of the described invention, the multi-screen clinic 21 operates as an integrated system 9 interfaced to a digital, preferably interactive, display system, such as used in information kiosks, movie theaters, and similar technologies in which digital multimedia is utilized.

FIGURE 2 is a block diagram showing a system 30 for pre-loading still imagery data in an interactive multimedia presentation environment in accordance with the present invention. A server 31 is interfaced to a client 32 via some form of interconnection 33, such as via a direct connection, dial-up connection, internetwork, intranetwork, or similar means, including a logical interconnection internal to the same physical machine. Thus, the server 31 and client 32 could be interfaced over a distributed computing environment or configured as a stand-alone system. Both the server 31 and client 32 include a persistent storage device

35, 39, respectively, such as a hard drive or other form of randomly accessible read-write device.

The server 31 includes a Web server 20 for serving Web content 36 to the client 32. The Web content 36 consists primarily of Web pages written as scripts
5 in an interpretable, tag-delimited language. Two commonly used languages for creating Web pages 56 are the Hypertext Markup Language (HTML), which is used in the described embodiment, and the Extensible Markup Language (XML), although other interpretable tag-delimited languages are equally applicable.

The Web content 36 is interpreted by a browser application 38 executing
10 on the client 32. The browser application 38 logically defines a multi-screen clinic (MSC) 21 for providing an integrated presentation environment, as further described below beginning with reference to FIGURE 4. Both the browser application 38 and the multi-screen clinic 21 stage individual items of Web content 36, such as still image and graphics files, in a cache 40 on the storage
15 device 39. The use of the cache 40 for temporarily storing the individual Web content items avoids having to repeatedly incur the same time hit needed by repeatedly downloading these items from the server 31.

The server 31 also logically incorporates an active server 34. Like a browser application 38, the active server 34 interprets executable scripts, known
20 as Active Server Pages, embedded within the Web content 36. In the described embodiment, the Active Server Page technology, licensed by Microsoft Corporation, Redmond, Washington, is used. Upon execution, the Active Server Pages are converted into pure Web content 36, typically written in HTML or XML. In the described embodiment, the Active Server Pages are written as either
25 JavaScripts or VBScripts, both of which are described in A.K. Weissinger, "ASP in a Nutshell, A Desktop Quick Reference," chs. 1-3, O'Reilly & Assocs. (1999), the disclosure of which is incorporated herein by reference. Finally, the server 31 maintains a database 37 in the storage device 35, as further described below with reference to FIGURE 3, in which user records and select Web content are stored.

FIGURE 3 is a block diagram showing the functional software modules 50 of a server 31 for use in the system 30 of FIGURE 2. Each module is a computer program written as source code in a conventional programming language, such as the C or C++ programming languages, and is presented for execution by the CPU 5 of the server 31 as object or byte code, as is known in the art. The various implementations of the source code and object and byte codes can be held on a computer-readable storage medium or embodied on a transmission medium in a carrier wave.

The server 31 consists of six functional components: Web server 20, 10 database server 51, video server 52, image server 53, advertisement rotator 54, and shopping cart 55. Basically, the Web server 20 serves Web pages 63, principally consisting of interpretable, tag-delimited scripts, in response to requests 62 received from clients 32. In the described embodiment, the Web pages 63 are written in HTML, although any similar Web page description 15 language, such as XML, could be used. In providing Web service, the Web server 20 accesses a collection of stored Web pages 56 and Active Server Page (ASP) scripts 57.

The database server 51 interfaces to the database 28 which includes libraries of video files 58, image files 59, and advertisement and information files 20 60. The video files 58, image files 59, and advertisement and information files 60 are retrieved from the database 28 responsive to requests respectively received from the video server 52, image server 53, and advertisement rotator 54. In the described embodiment, the database server 51 connects to the database 28 using an Open Database Connectivity (ODBC) driver which enables access to database 25 management systems using the Structured Query Language (SQL) for access.

The video server 52 plays the video files 58 as either streaming or live video to the requesting clients 32. Similarly, the image server 53 serves the image files 59 which contain still imagery corresponding the requested video file 58 to the requesting clients 32. As described below, the image files 59 are stored by an 30 image pre-loader into a cache local to the requesting client 32. The video player

52 begins playing the requested video file 58 upon the receipt of a signal from the image pre-loader, thereby enabling the use of a low network bandwidth connection since only video is served.

The advertisement rotator 54 serves advertisement and information files 5 60 to the requesting clients 32. Generally, the advertisements and information are selected to relate to the playing video and slideshow. To help maintain user interest level, the advertisements and information are automatically changed by the advertisement rotator 54 at a pre-determined frequency. In the described embodiment, the Ad Rotator component of Microsoft Corporation's Active Server 10 Page technology is used, although any similar application providing automated advertisement and information content rotation could also be used. The Ad Rotator component is described in A.K. Weissinger, "ASP in a Nutshell, A Desktop Quick Reference," ch. 12, *infra.*, the disclosure of which is incorporated herein by reference.

15 The shopping cart 55 provides the means with which to facilitate commercial transactions through requesting clients 32. Through hyperlinks embedded in the served Web pages 63, a user can indicate a desire to purchase an item described in either the video, still imagery, or advertisements and information. A running tally of items to be purchased is included in the multi- 20 screen clinic 30. Any of the well known shopping cart technologies for transacting electronic commerce as are known in the art could be employed.

FIGURE 4 is a block diagram showing the functional software modules 70 of a client 32 for use in the system 30 of FIGURE 2. As in the server 31, each module is a computer program written as source code in a conventional 25 programming language, such as the HTML, XML, or JavaScript programming languages, and is presented for execution by the CPU of the client 32 as object or byte code, as is known in the art. The various implementations of the source code and object and byte codes can be held on a computer-readable storage medium or embodied on a transmission medium in a carrier wave.

The client 32 consists of four functional components: browser 29, multi-screen clinic 30, video player 71, and image pre-loader 72. The browser 29 is an ordinary browser application for viewing Web pages 63, as is known in the art. Web content 27, including Web pages 63, downloaded from the Web server 20 are staged as temporary files 74 maintained in a local cache 40. In the described embodiment, the Internet Explorer Web browser, licensed by Microsoft Corporation, Redmond, Washington, is used, although any equivalent Web browser, such as the Navigator Web browser, licensed by Netscape Corporation, Mountain View, California, and the like could be used.

The multi-screen clinic (MSC) 30 is a modular framework defining an integrated presentation space within the browser 29. In the described embodiment, the multi-screen clinic 30 is logically defined via a series of Web pages 63 received from the Web server 20 in response to a video selection. The multi-screen clinic 30 provides an extensible architecture which allows integrated functionality between presentation viewing panels for interactively viewing multimedia segments, such as video, still imagery, advertisements, and information. A multi-screen clinic 30 suitable for use in the present invention is described in the commonly-assigned, related U.S. patent application, Serial No. _____, entitled "System And Method For Providing An Interactive Multimedia Presentation Environment With Low-Bandwidth Capable Sessions," filed April 27, 2000, pending, the disclosure of which is incorporated herein by reference. These multimedia segments can also be integrated with electronic commerce via a shopping cart 55 (shown in FIGURE 3).

The video player 71 plays the requested video in a video presentation panel (not shown) logically defined within the multi-screen clinic 30. Video playback does not start until substantially all of the still image files 59 for the requested video file 58 have been downloaded into the local cache 40 by the image pre-loader 72. In the described embodiment, the video presentation panel is generated by the Windows Media Player product, licensed by Microsoft Corporation, Redmond, Washington. Alternatively, the Real Media Player,

licensed by Real Networks, Inc., Seattle, Washington, or any similar video playback application, could also be utilized.

The image pre-loader 72 downloads substantially all of the image files 59 constituting a slideshow. Each requested video includes a slideshow of still
5 imagery content for augmenting lower resolution video content. The slideshow is downloaded prior to video playback with the individual still image files 59 stored as temporary image files 73 staged in the local cache 40. The image pre-loader 72 enables the use of a lower network bandwidth connection between the client 32 and the server 31. Rather than attempting to simultaneously receive both video
10 and still imagery, the image pre-loader serializes the data into separate downloads, thereby enabling the video player 71 to utilize maximum available network bandwidth. Upon completion of the download, the image pre-loader 72 signals the video server 52 to begin sending the video to the video player 71.

The image pre-loader 72 includes four components: information display
15 75, status display 76, loader 77, and indexer 78. In the described embodiment, the image pre-loader 72 is logically defined as an interpreted script written as a JavaScript and downloaded from the server 31. The JavaScript is then executed by the browser application 29. JavaScripts allow limited programming instructions to be incorporated into the Web pages 56, thereby enabling logically
20 definable client-side functionality. Other programming languages are equally applicable, including Visual Basic.

The information display 76 provides a visual description of the video and still imagery content, including, non-exclusively, the name of the video program, a brief description, the host server, the length of the video program, the number of
25 still images, the location of the video and still images in terms of host servers. The status display 76 indicates the status of the image pre-load. The estimated time to download the slideshow varies dependent on the number and resolution of the still image content and upon the network bandwidth and effective data transfer rate.

The loader 77 actually downloads the image files 59 from the server 31. Each image file 59 is sequentially retrieved by index from the database 28 via the database server 51. As each image file 59 is received by the client 32, the image file 59 is stored in the cache 40 as a staged image file 73. In the described
 5 embodiment, the cache 40 is logically defined in a temporary subdirectory within the native file system. By default, the same temporary subdirectory used to store staged Web content is also used for staged image files 73.

The indexer 78 loads the staged image files 73 from the cache 40 into a still image presentation panel during video playback. Each image is identified by
 10 a unique index. Each video file 58 includes a two- to three-part track consisting of video (optional), audio, and events. The events are tags which include instructions to load individual image files 59 identified by an index according to playing time. For example, a sample event might be as follows:

15 00:03:30 URL Video_Images.asp?go=13&&main

where the image with an index of "13" and referenced by the Uniform Resource Locator (URL) *Video_Images.asp*, an Active Server Page, will be loaded by the
 20 indexer 78 at three minutes, thirty seconds. The indexer 78 continues to repeatedly load image files 73 from the cache 40 as indicated by the event track.

In a further embodiment, the indexer 78 can be combined with a set of user controls via which select image files 73 are loaded and displayed for viewing on-demand. The indexer 78 stops the video playback by signaling the video
 25 player 71 while the user navigates through the staged image files 73. Video playback then resumes upon user command.

The client 31 operates in accordance with a sequence of process steps, as further described below beginning with reference to FIGURE 8.

FIGURE 5 is a screen shot showing, by way of example, an integrated presentation space 90 with navigational controls exported by the client 32 of
 30 FIGURE 4. The user is presented with a set of Web sites and indicates a selection

91 for a video program 92. In the described embodiment, the video programs 92 are organized according to available network bandwidth which includes, for example, a slideshow 93 for playback at 28.8 Kbaud, a video 94 for playback at 56.6 Kbaud, and a movie 95 for playback at 300 Kbaud. Other combinations of
5 video programs 92 are feasible.

FIGURE 6 is a screen shot showing, by way of example, an integrated presentation space 100 performing an image content pre-load exported by the client 32 of FIGURE 4. The integrated presentation space 100 embodies the multi-screen clinic 30 and is functionally organized into four main viewing areas
10 presented within a stylized view window: video presentation panel 101, still image presentation panel 102, advertisement presentation panel 103, and shopping cart status panel 104. The viewing areas are functionally integrated to cooperatively provide an interactive presentation environment. For instance, the still images presented in the still image presentation panel 102 are keyed to appear
15 at pre-defined times during playback of videos in the video presentation panel 101. Similarly, the advertisement presentation panel 103 is also preferably linked to the shopping cart status panel 104.

The video presentation panel 101 represents the user interface of the video player 71 (shown in FIGURE 4) and incorporates a plurality of controls 105 for
20 controlling the playback of video files 58 downloaded from the video server 52 (shown in FIGURE 3). In addition, the video presentation panel 101 also contains status information generated by the multi-screen clinic 30, including the program name 106, host 107, and approximate playing time (length) 108. Other status information could be displayed, as is known in the art.

25 The still image presentation panel 102 displays still images files 59 downloaded from the image server 53 and temporarily staged in the cache 40. A status message 109 is displayed to the user while the image files 73 are being downloaded by the image pre-loader 72 (shown in FIGURE 4) into the cache 40. A start message 110 is displayed to the user after the image pre-loader 72 signals
30 the video server 52 to start sending the video.

The advertisement presentation panel 103 displays advertisements or other information stored in the advertisement files 60 downloaded from the server 31. The advertisements are displayed either asynchronously or, preferably, keyed to the video playback or still image display and are rotated on a periodic basis by the advertisement rotator 54.

Finally, the shopping cart status panel 104 displays a running tally 111 of items "placed" into the user's shopping cart for purchase.

FIGURE 7 is a screen shot showing, by way of example, an integrated presentation space 120 performing a video content playback exported by the client 32 of FIGURE 4. Still images 121, keyed to the video content playback, are displayed in the still image presentation panel. An item displayed in the advertisement presentation panel can be selected for purchase and placed into the user's shopping car by selecting the "Add Product to Shopping Cart" area 122 of the shopping cart status panel 94. The running tally 123 would be updated accordingly.

FIGURE 8 is a flow diagram showing a method 140 for pre-loading still imagery data in an interactive multimedia presentation environment in accordance with the present invention. Preliminarily, the connection speed is obtained (block 131) by the image pre-loader 72 (shown in FIGURE 4) for determining the available network bandwidth. The connection speed can be determined directly based on a user prompt, indirectly based on the type of video program 92 selected, or via a cookie which can be used by the server 31 to retrieve a stored connection speed from the user records 61 stored in the database 28, as further described below. Next, a database connection is established (block 132). Information about the selected video maintained in the database 28 is retrieved and the program name 106, host 107, and approximate playing time (length) 108 (shown in FIGURE 6) are updated (block 133). The video record for the selected video is then updated (block 134) as further described below with reference to FIGURE 9.

The selection of the video triggers the downloading of the pre-loader script to the client 32 (block 135). Upon receipt, the browser application 29

executes the pre-loader script to logically define the image pre-loader 72 (shown in FIGURE 4), which then begins pre-loading the image files 73 staged in the cache 40 (block 136), as further described below with reference to FIGURE 10. Finally, the video is played back upon the completion of the pre-load (block 137),
5 as further described below with reference to FIGURE 11. The method then terminates.

FIGURE 9 is a flow diagram showing the routine 140 for updating a selected video record for use in the method 140 of FIGURE 8. The purpose of this routine is to update records on individual users and video programs. If the
10 user is a first time user of the Web site (block 151), a cookie is generated and sent to the client 32 from which the user has selected the video program (block 152). A new user record 61 (shown in FIGURE 3) is created in the database 28 for that user (block 153). Otherwise, the cookie for the current user is obtained and the corresponding user record 61 is retrieved from the database 28 (block 154). Next,
15 the index counter tracking the video programs selected by the current user is updated (block 155). Similarly, the index counter tracking all users who have selected the current video is updated (block 156). The routine then returns.

FIGURE 10 is a flow diagram showing the routine 170 for pre-loading images for use in the method 140 of FIGURE 8. The purpose of this routine is to
20 iteratively retrieve and store image files 59 from the server 31 into the cache 40. The image files 59 are retrieved and stored using an iterative processing loop (blocks 171-175). During each iteration, the image file 59 is requested from the server 31 by index (block 172). The image file 59 is retrieved from the database 28 and sent to the requesting client 32 (block 173). Finally, the retrieved image
25 file 59 is stored into the cache 40 (block 174). After the image files 59 for the selected video have been retrieved and stored, a signal indicating the completion of the download is sent requesting the server 31 to start playing the video (block 176). The routine then returns.

FIGURE 11 is a flow diagram showing the routine 190 for playing a video
30 for use in the method 140 of FIGURE 8. The purpose of this routine is to play the

video while concurrently displaying the image files 73 staged in the cache 40. Thus, the video play 71 (shown in FIGURE 4) begins playing the video (block 191). Streaming video is used in the described embodiment, although other forms of video playback, including live video, could also be used. While the video is playing (blocks 192-197), two concurrent execution threads (blocks 193 and 194-196) are performed as follows. In the first execution thread (block 193), streaming video is received by the video player 71 and displayed in the video presentation panel 101 (shown in FIGURE 6). In the second execution thread (blocks 194-196), as each image tag in the event track of the streaming video is encountered (block 194), the matching image referenced by the image tag, that is, having a matching index, is retrieved from the cache 40 (block 195) and displayed in the still image presentation panel 102 (block 196). Both execution threads terminate upon completion of the video playback and slideshow. The routine then returns.

Thus, the present invention provides an image pre-loader that minimizes network bandwidth requirements by serializing the downloading of video and still imagery content. Consequently, higher resolution still imagery can be associated with a lower resolution video presentation. The video and still imagery content can be coordinated using indices to synchronize content relating to a common message.

While the invention has been particularly shown and described as referenced to the embodiments thereof, those skilled in the art will understand that the foregoing and other changes in form and detail may be made therein without departing from the spirit and scope of the invention.

What is claimed is:

1 1. A system for efficiently pre-loading still imagery data in an
2 interactive presentation environment, comprising:
3 a database organized into a plurality of records, each record referring to
4 one of a video file, a still image file, and a user record, each video file being
5 logically associated with at least one image file and at least one user record;
6 a server processing a selection of one such video file by retrieving the
7 selected video file record from the database, retrieving each user record associated
8 with the selected video file, and updating the user record to reflect the selection of
9 the selected video file;
10 an image pre-loader pre-loading the images files logically associated with
11 the selected video file by iteratively retrieving each still image file associated with
12 the selected video file and downloading the retrieved still image file into a cache
13 and generating a signal upon completion of the iterative retrieval of substantially
14 all of the still image files associated with the selected video file; and
15 a video player playing back the selected video file within the interactive
16 presentation environment responsive to the completion signal and a viewer
17 displaying the cached still image files in synchronization with the playback.

1 2. A system according to Claim 1, further comprising:
2 a modular framework defining an integrated presentation space within a
3 browser application executing on a client, comprising:
4 a view window formed as an interactive user interface with one or
5 more hyperlinked controls for selecting video content;
6 a still image presentation panel generated within the view window
7 and displaying the cached still images; and
8 a video presentation panel generated within the view window and
9 displaying selected video file.

1 3. A system according to Claim 1, further comprising:

2 each image file having an associated unique index;
3 each video file incorporating a list of indices; and
4 the viewer retrieving the still image files from the cache matching each
5 index in the list of indices during the playback.

1 4. A system according to Claim 3, further comprising:
2 each video file recording the list of indices in an event track.

1 5. A system according to Claim 1, further comprising:
2 the server logically defining the image pre-loader to operatively
3 downloading the retrieved still image files responsive to the selection of the
4 selected video file.

1 6. A system according to Claim 5, wherein the image pre-loader is
2 logically defined as an interpretable script written in a programming language
3 comprising JavaScript.

1 7. A system according to Claim 1, further comprising:
2 the database interfaced using a Structured Query Language as a data
3 access standard.

1 8. A system according to Claim 1, further comprising:
2 an informational display generated within the interactive presentation
3 environment describing the selected video file.

1 9. A system according to Claim 1, further comprising:
2 each user record in the database uniquely associated with a user that has
3 selected each such video file by exchanging cookies.

1 10. A system according to Claim 1, further comprising:
2 the server obtaining network bandwidth on a connection over which the
3 selected video file and each still image file referenced by the selected video file
4 will be downloaded and retrieving one such selected video file from the database

5 substantially optimized for download over the connection with the obtained
6 network bandwidth.

1 11. A system according to Claim 1, wherein the interactive
2 presentation environment comprises at least one environment selected from the
3 group comprising a stand-alone computing environment, a distributed computing
4 environment, an information kiosk, a movie theater, and a digital display system.

1 12. A method for efficiently pre-loading still imagery data in an
2 interactive presentation environment, comprising:
3 maintaining a database organized into a plurality of records, each record
4 referring to one of a video file, a still image file, and a user record, each video file
5 being logically associated with at least one image file and at least one user record;
6 processing a selection of one such video file, comprising:
7 retrieving the selected video file record from the database; and
8 retrieving each user record associated with the selected video file
9 and updating the user record to reflect the selection of the selected video file;
10 pre-loading the images files logically associated with the selected video
11 file, comprising:
12 iteratively retrieving each still image file associated with the
13 selected video file and downloading the retrieved still image file into a cache; and
14 generating a signal upon completion of the iterative retrieval of
15 substantially all of the still image files associated with the selected video file; and
16 playing back the selected video file within the interactive presentation
17 environment responsive to the completion signal and displaying the cached still
18 image files in synchronization with the playback.

1 13. A method according to Claim 12, further comprising:
2 exporting a modular framework defining an integrated presentation space
3 within a browser application executing on a client, comprising:

4 forming a view window as an interactive user interface with one or
5 more hyperlinked controls for selecting video content;
6 generating a still image presentation panel within the view window
7 and displaying the cached still images; and
8 generating a video presentation panel within the view window and
9 displaying selected video file.

1 14. A method according to Claim 12, further comprising:
2 associating a unique index with each image file;
3 incorporating a list of indices into each video file; and
4 retrieving the still image files from the cache matching each index in the
5 list of indices during the playback.

1 15. A method according to Claim 14, further comprising:
2 recording the list of indices into an event track included in each video file.

1 16. A method according to Claim 12, further comprising:
2 logically defining an image pre-loader operatively downloading the
3 retrieved still image files responsive to the selection of the selected video file.

1 17. A method according to Claim 16, wherein the image pre-loader is
2 logically defined as an interpretable script written in a programming language
3 comprising JavaScript.

1 18. A method according to Claim 12, further comprising:
2 interfacing to the database using a Structured Query Language as a data
3 access standard.

1 19. A method according to Claim 12, further comprising:
2 generating an informational display describing the selected video file.

1 20. A method according to Claim 12, further comprising:

2 exchanging cookies to uniquely associate each user record in the database
3 with a user that has selected each such video file.

1 21. A method according to Claim 12, further comprising:
2 obtaining network bandwidth on a connection over which the selected
3 video file and each still image file referenced by the selected video file will be
4 downloaded; and

5 retrieving one such selected video file from the database substantially
6 optimized for download over the connection with the obtained network
7 bandwidth.

1 22. A method according to Claim 12, wherein the interactive
2 presentation environment comprises at least one environment selected from the
3 group comprising a stand-alone computing environment, a distributed computing
4 environment, an information kiosk, a movie theater, and a digital display system.

1 23. A computer-readable storage medium holding code for efficiently
2 pre-loading still imagery data in an interactive presentation environment,
3 comprising:

4 maintaining a database organized into a plurality of records, each record
5 referring to one of a video file, a still image file, and a user record, each video file
6 being logically associated with at least one image file and at least one user record;

7 processing a selection of one such video file, comprising:

8 retrieving the selected video file record from the database; and

9 retrieving each user record associated with the selected video file
10 and updating the user record to reflect the selection of the selected video file;

11 pre-loading the images files logically associated with the selected video
12 file, comprising:

13 iteratively retrieving each still image file associated with the
14 selected video file and downloading the retrieved still image file into a cache; and

15 generating a signal upon completion of the iterative retrieval of
16 substantially all of the still image files associated with the selected video file; and
17 playing back the selected video file within the interactive presentation
18 environment responsive to the completion signal and displaying the cached still
19 image files in synchronization with the playback.

1 24. A storage medium according to Claim 23, further comprising:
2 exporting a modular framework defining an integrated presentation space
3 within a browser application executing on a client, comprising:
4 forming a view window as an interactive user interface with one or
5 more hyperlinked controls for selecting video content;
6 generating a still image presentation panel within the view window
7 and displaying the cached still images; and
8 generating a video presentation panel within the view window and
9 displaying selected video file.

1 25. A storage medium according to Claim 23, further comprising:
2 associating a unique index with each image file;
3 incorporating a list of indices into each video file; and
4 retrieving the still image files from the cache matching each index in the
5 list of indices during the playback.

1 26. A storage medium according to Claim 23, further comprising:
2 recording the list of indices into an event track included in each video file.

1 27. A storage medium according to Claim 23, further comprising:
2 logically defining an image pre-loader operatively downloading the
3 retrieved still image files responsive to the selection of the selected video file.

1 28. A storage medium according to Claim 23, further comprising:
2 interfacing to the database using a Structured Query Language as a data
3 access standard.

1 29. A storage medium according to Claim 23, further comprising:
2 generating an informational display describing the selected video file.

1 30. A storage medium according to Claim 23, further comprising:
2 exchanging cookies to uniquely associate each user record in the database
3 with a user that has selected each such video file.

1 31. A storage medium according to Claim 23, further comprising:
2 obtaining network bandwidth on a connection over which the selected
3 video file and each still image file referenced by the selected video file will be
4 downloaded; and
5 retrieving one such selected video file from the database substantially
6 optimized for download over the connection with the obtained network
7 bandwidth.

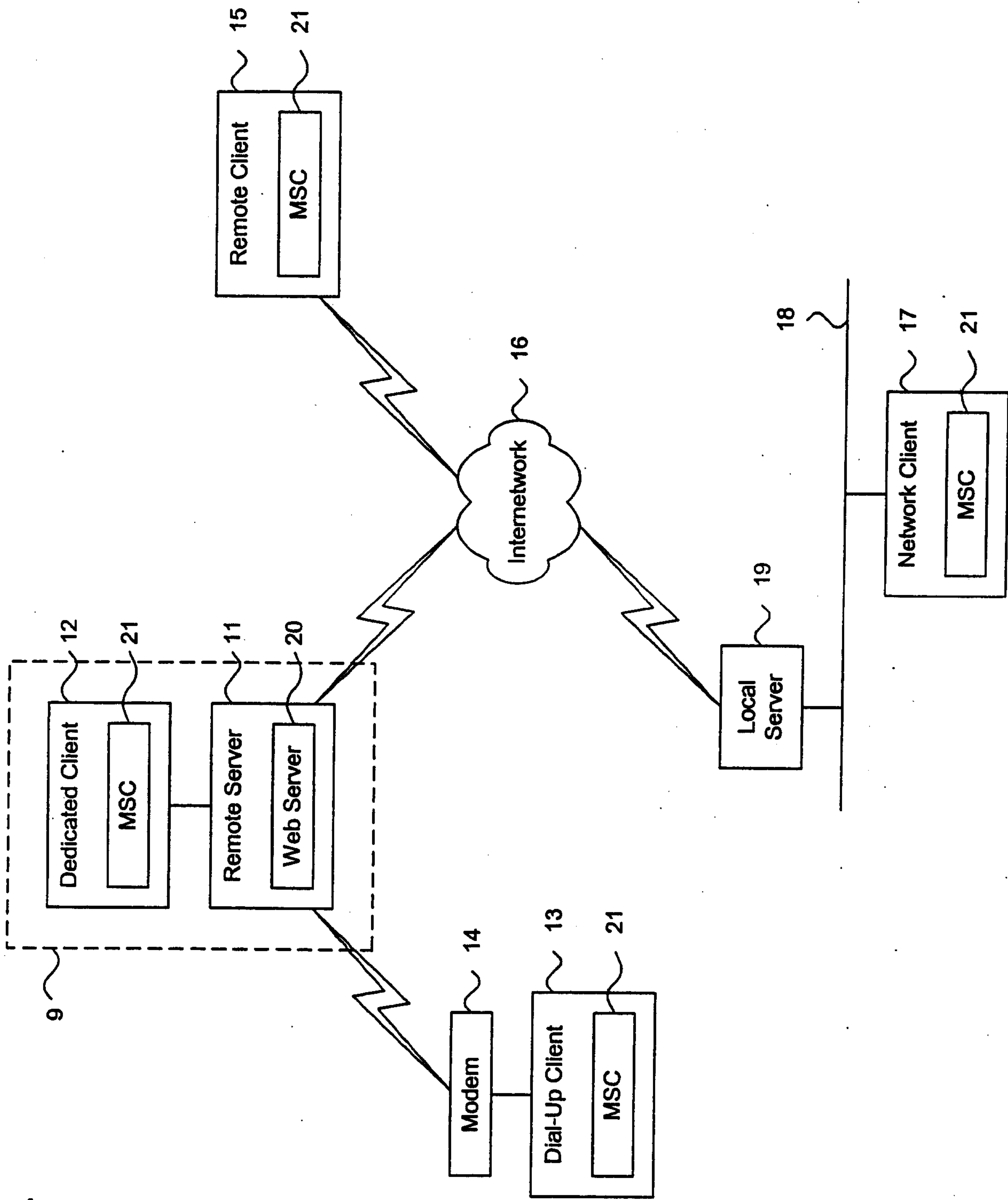
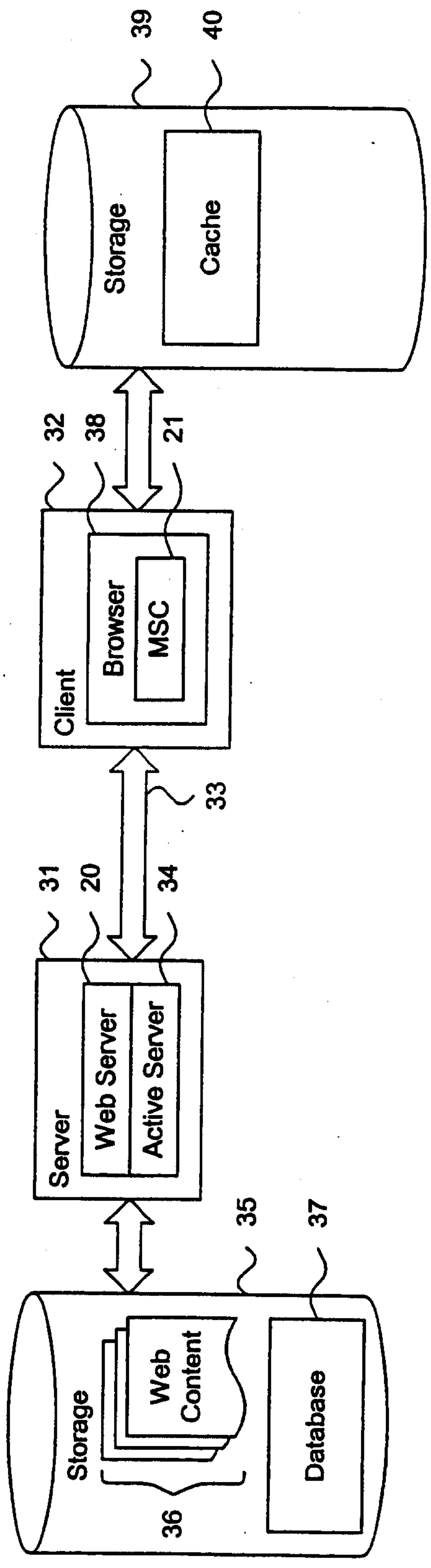


Figure 1.

Figure 2.
30



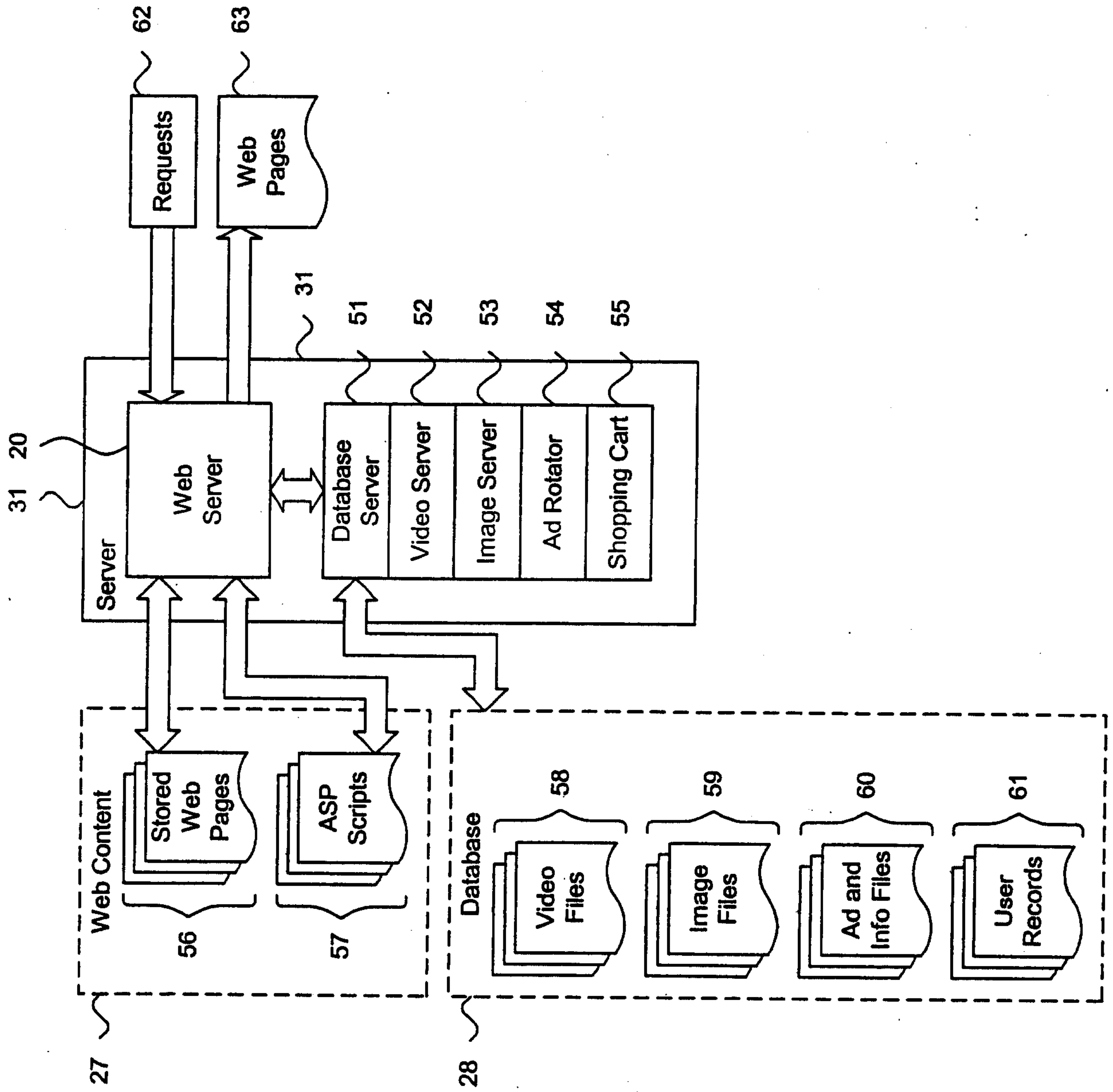


Figure 3.

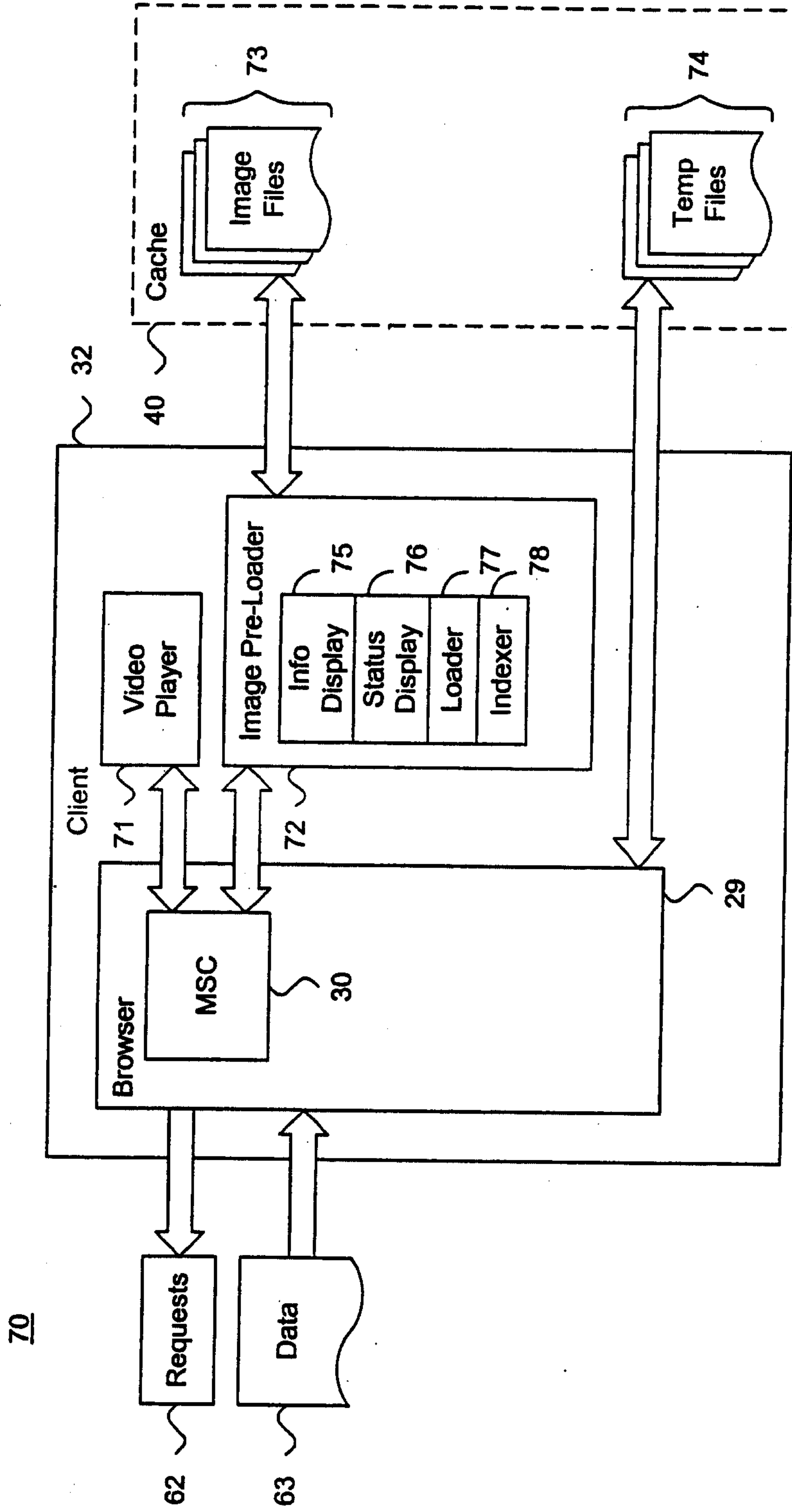
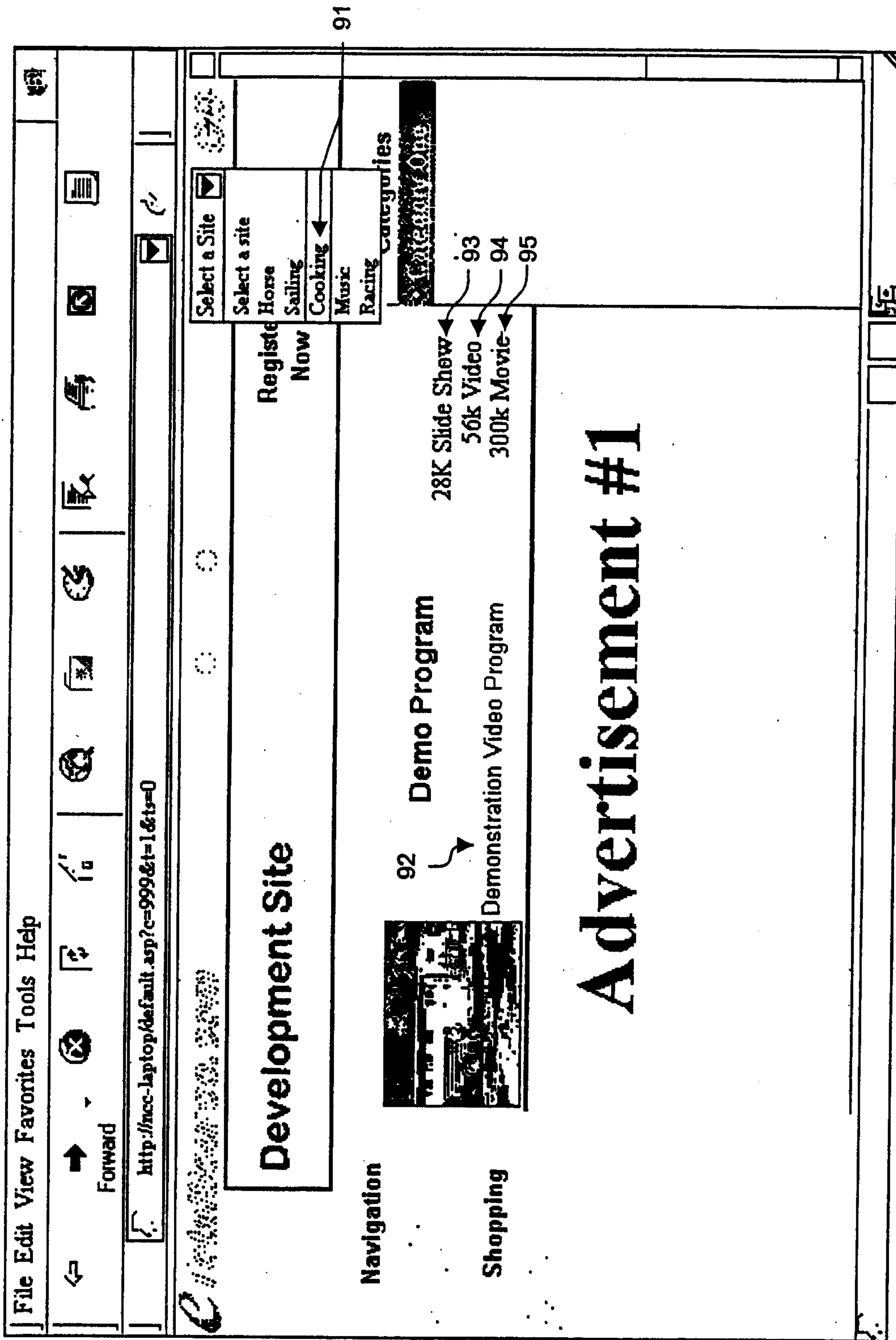


Figure 4.

Figure 5.

90



91

Navigation

Shopping

92

Demo Program

Demonstration Video Program

28K Slide Show

56k Video

300k Movie

93

94

95

Advertisement #1

Figure 6.

100

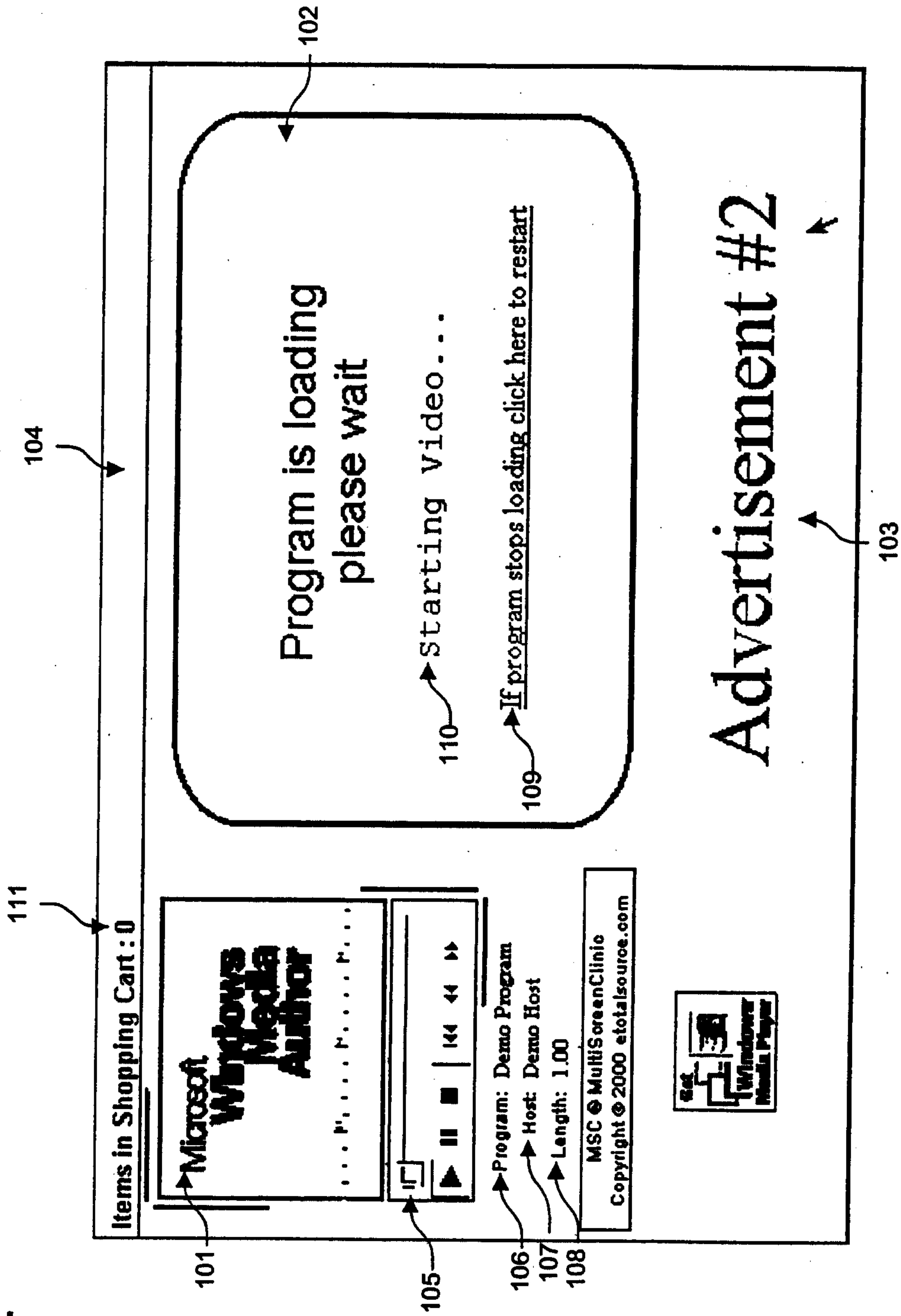


Figure 7.

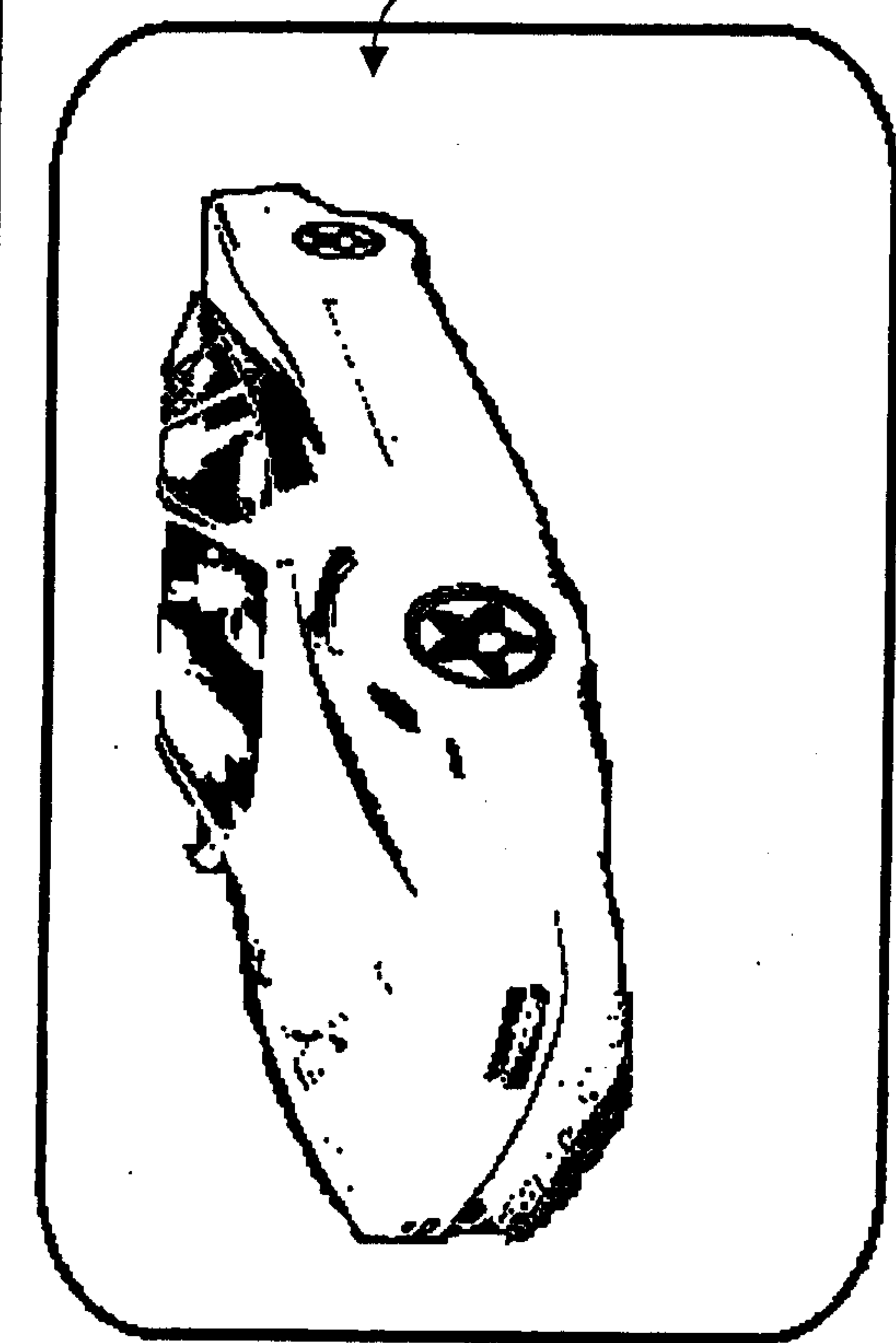
120

123

122

Items in Shopping Cart : 0

Add Product to Shopping Cart



Advertisement #2

121

Figure 8.

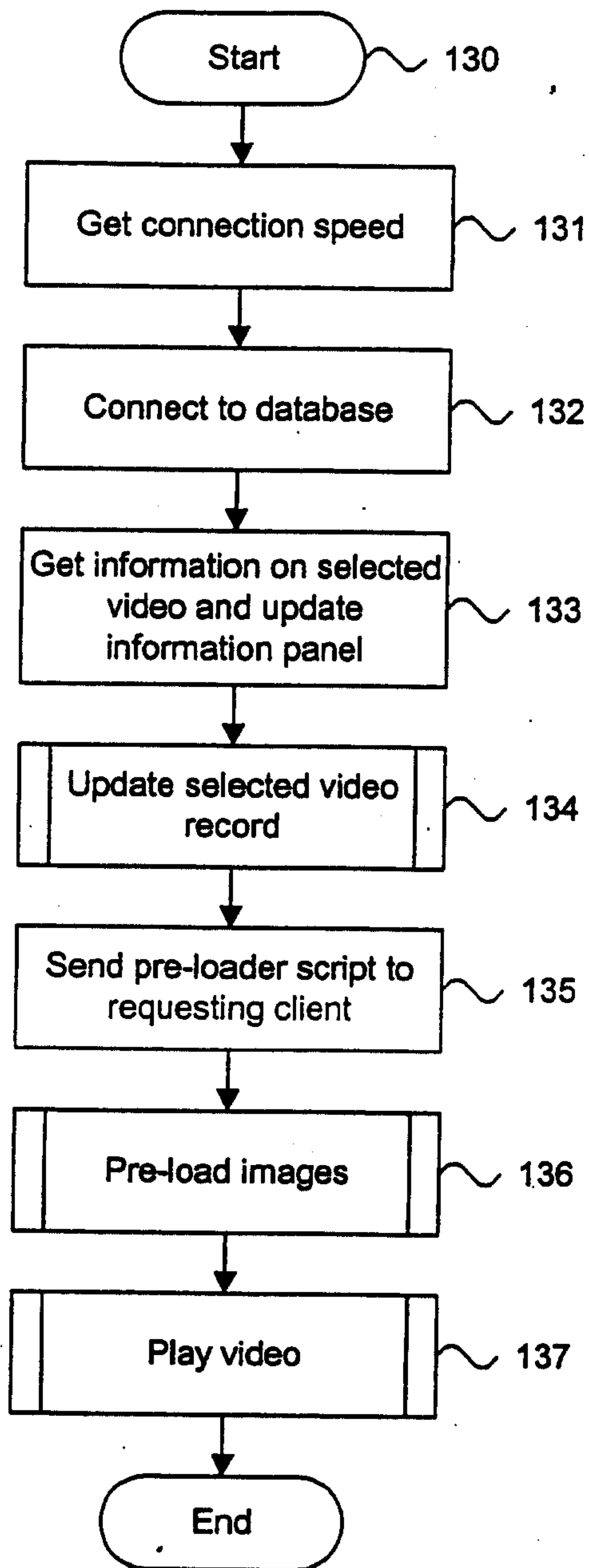


Figure 9.

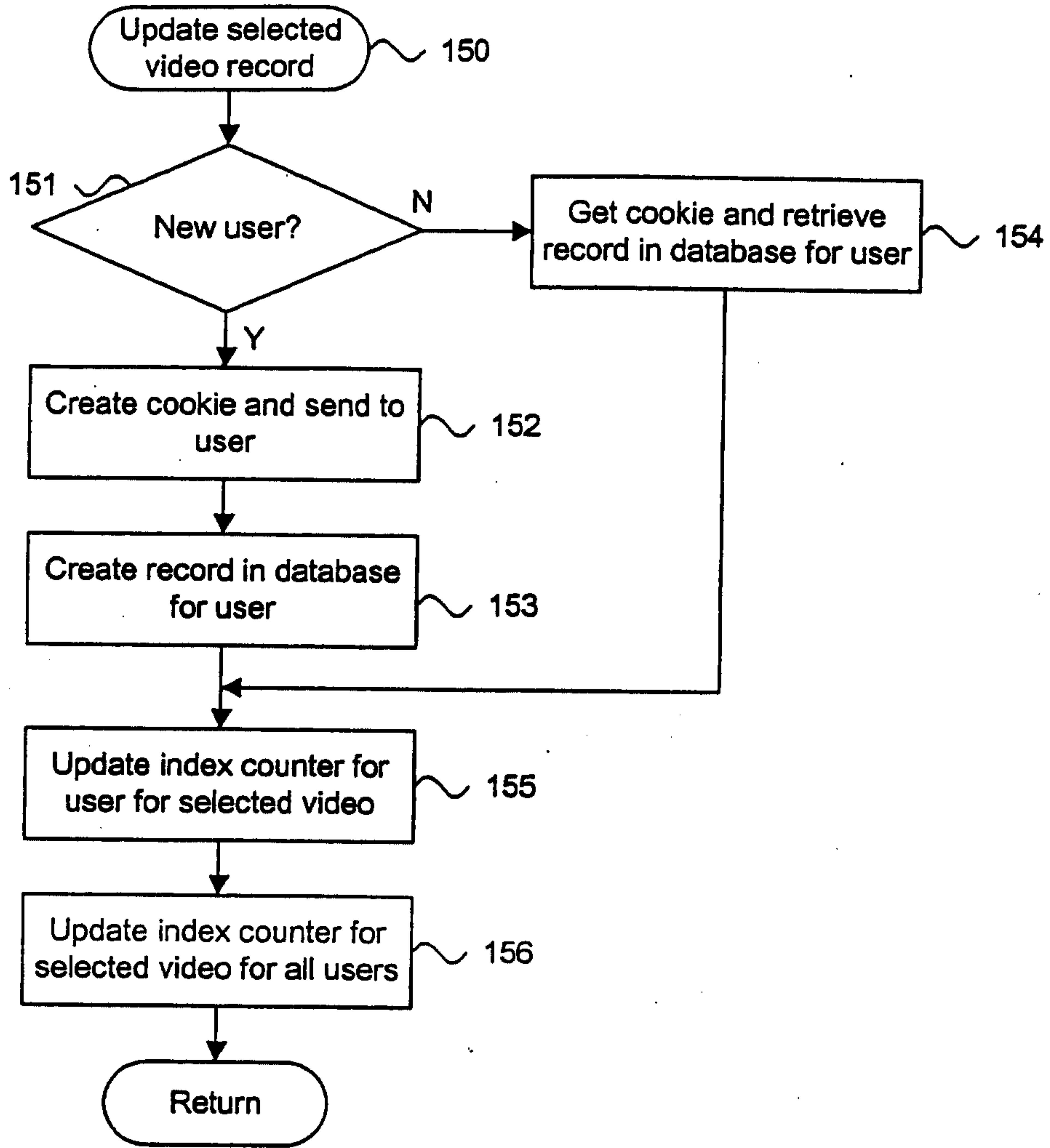


Figure 10.

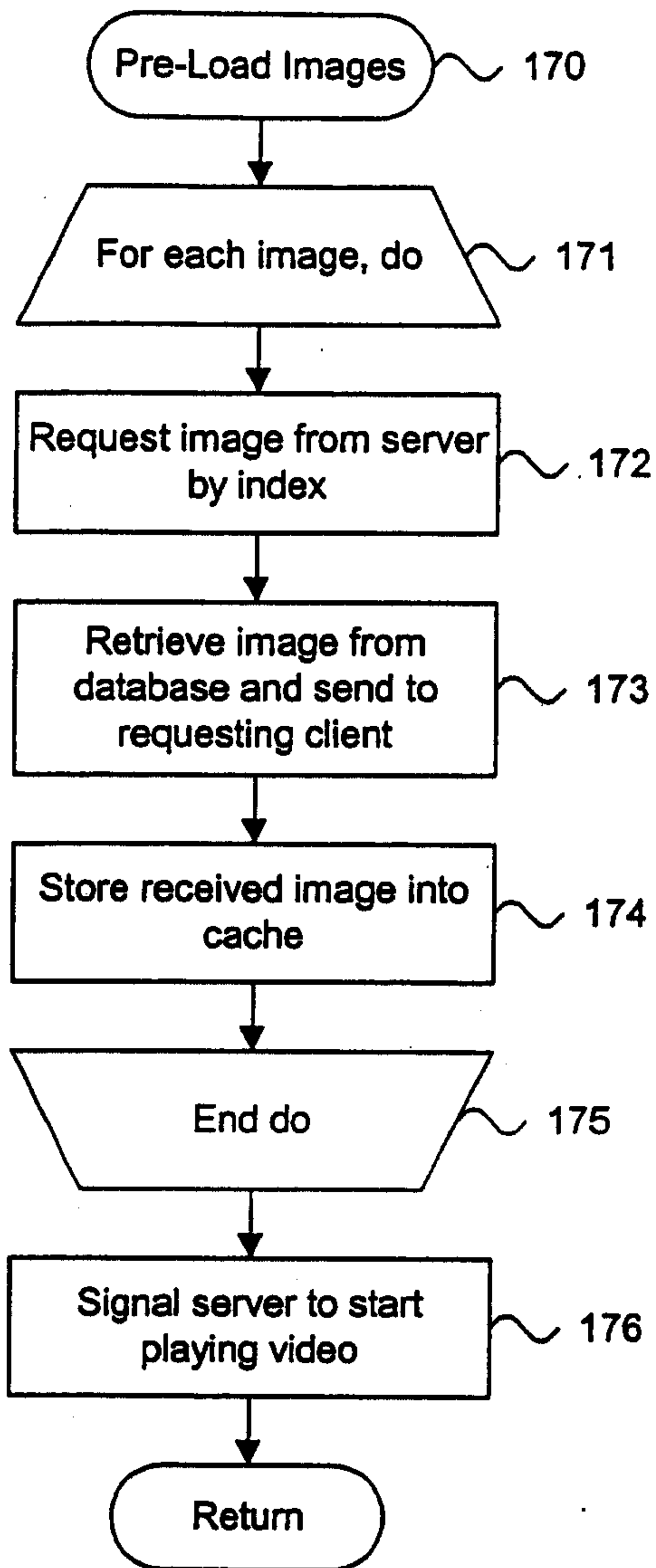


Figure 11.

