

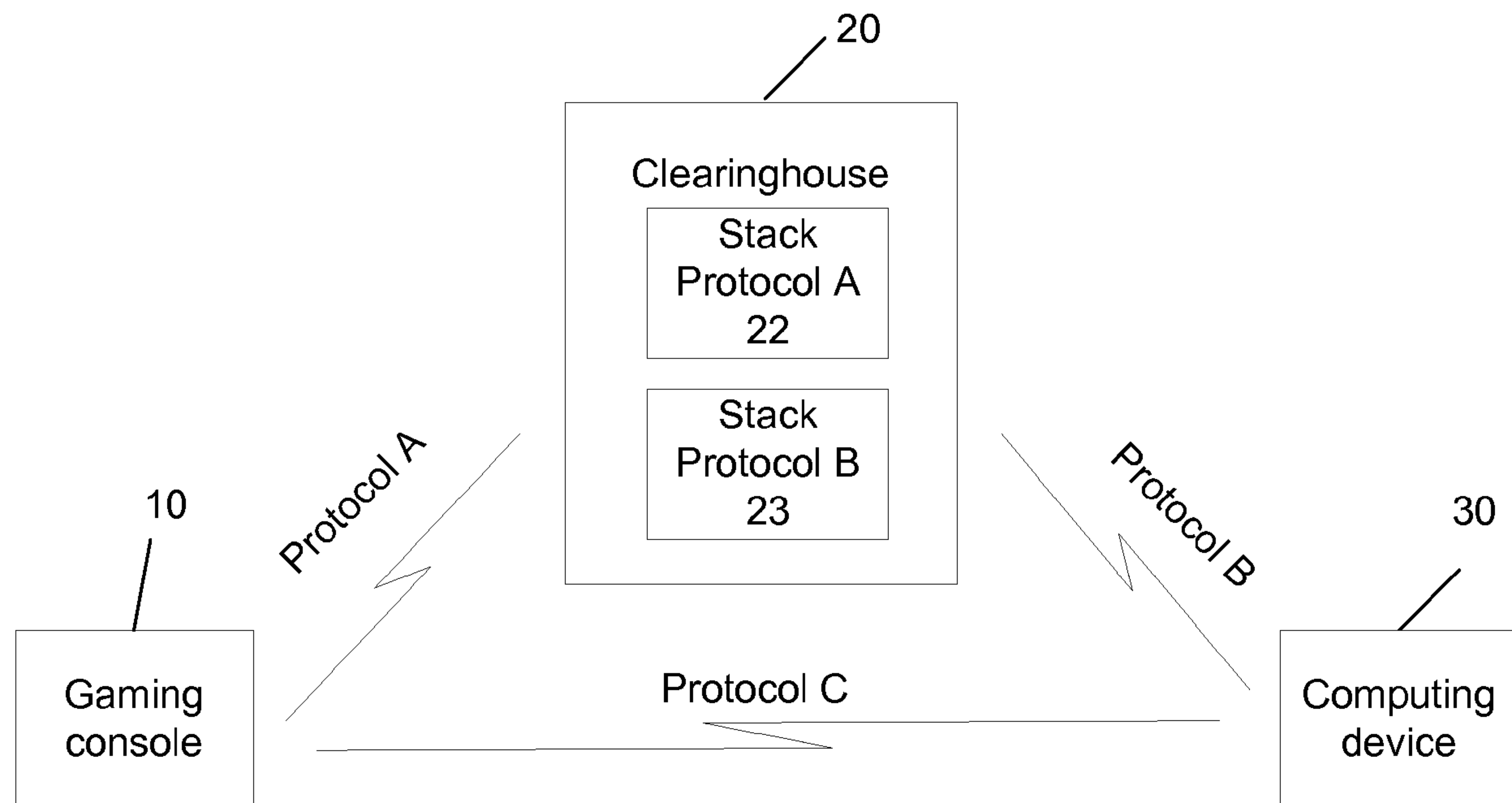


(86) Date de dépôt PCT/PCT Filing Date: 2008/02/29  
(87) Date publication PCT/PCT Publication Date: 2008/09/18  
(85) Entrée phase nationale/National Entry: 2009/09/11  
(86) N° demande PCT/PCT Application No.: US 2008/055573  
(87) N° publication PCT/PCT Publication No.: 2008/112448  
(30) Priorité/Priority: 2007/03/13 (US11/717,387)

(51) Cl.Int./Int.Cl. *G06F 21/20* (2006.01),  
*G06Q 50/00* (2006.01)  
(71) Demandeur/Applicant:  
MICROSOFT CORPORATION, US  
(72) Inventeurs/Inventors:  
CHOUDHRY, KAREEM AHMAD, US;  
SLEEMAN, MARTIN J., US;  
MOORE, BRIAN ANDREW, US;  
SANFILIPPO, ROBERT, US;  
O'NEILL, VANCE, US;  
CAIAFA, DANIEL, US  
(74) Agent: SMART & BIGGAR

(54) Titre : COMMUNICATION A JOUEURS MULTIPLES SECURISEE SUR PLATES-FORMES CROISEES, EN RESEAU, ET JEU

(54) Title: SECURED CROSS PLATFORM NETWORKED MULTIPLAYER GAME PLAY



**FIG. 1**

(57) Abrégé/Abstract:

A user of a software application (e.g., game or title) running at a gaming console may interact securely in real-time with a user of the same software application running at a general purpose computing device. Thus, a game player on the gaming console may play online over a secured communication channel with a game player on a PC.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau(43) International Publication Date  
18 September 2008 (18.09.2008)

PCT

(10) International Publication Number  
**WO 2008/112448 A1**

## (51) International Patent Classification:

**G06F 21/20** (2006.01) **G06Q 50/00** (2006.01)

## (21) International Application Number:

PCT/US2008/055573

## (22) International Filing Date:

29 February 2008 (29.02.2008)

## (25) Filing Language:

English

## (26) Publication Language:

English

## (30) Priority Data:

11/717,387 13 March 2007 (13.03.2007) US

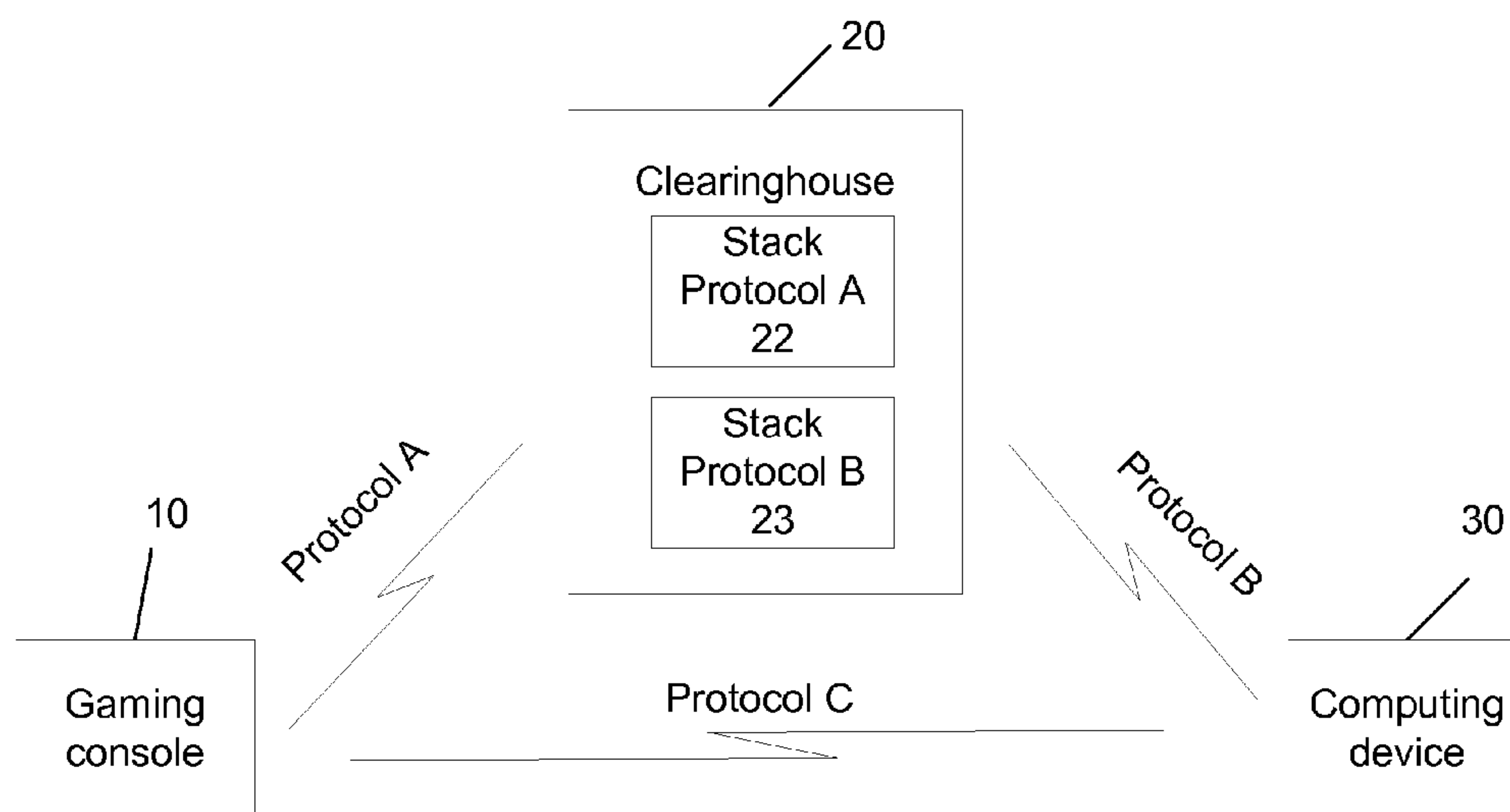
(71) Applicant (for all designated States except US): **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).(72) Inventors: **CHOUDHRY, Kareem Ahmad**; c/o Microsoft Corporation International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **SLEEMAN, Martin J.**; c/o Microsoft Corporation International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **MOORE, Brian Andrew**; c/o Microsoft Corporation International Patents, One MicrosoftWay, Redmond, Washington 98052-6399 (US). **SANFILIPPO, Robert**; c/o Microsoft Corporation International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **O'NEILL, Vance**; c/o Microsoft Corporation International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **CAIAFA, Daniel**; c/o Microsoft Corporation International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),

[Continued on next page]

(54) Title: SECURED CROSS PLATFORM NETWORKED MULTIPLAYER COMMUNICATION AND GAME PLAY

**FIG. 1**

(57) Abstract: A user of a software application (e.g., game or title) running at a gaming console may interact securely in real-time with a user of the same software application running at a general purpose computing device. Thus, a game player on the gaming console may play online over a secured communication channel with a game player on a PC.

WO 2008/112448 A1

# WO 2008/112448 A1



European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Published:**

— *with international search report*

**Declarations under Rule 4.17:**

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*



## SECURED CROSS PLATFORM NETWORKED MULTIPLAYER GAME PLAY

### BACKGROUND

[0001] Software applications, such as first-person shooter (FPS) games, are available for both gaming consoles (e.g., Microsoft Corporation's Xbox™) and personal computers (PCs). However, gaming consoles have not been able to communicate across a network connection with PCs. Therefore, for example, gaming consoles have not been able to play online video games with PCs.

[0002] Xbox Live™ is an online multiplayer gaming and content delivery system created and operated by Microsoft Corporation. Microsoft's Live Anywhere™ enables a variety of non-Xbox platforms such as a PC and mobile phones to connect to Xbox Live, though with lesser functionality. For example, cross-platform play is not available, such that PC players cannot compete against Xbox players.

### SUMMARY

[0003] A user of a software application (e.g., game or title) running at a gaming console may interact securely in real-time with a user of the same software application running at a general purpose computing device, such as a PC. Thus, a game player on the gaming console may play online with a game player on a PC. The gaming console may securely communicate with the general purpose computing device either via a clearinghouse or directly via a network or system link connection, for example.

[0004] The stack and ports are set to accommodate the cross-platform features. Additionally, a secure key exchange is provided, as well as big-endian and little-endian byte ordering.

[0005] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

**BRIEF DESCRIPTION OF THE DRAWINGS**

[0006] Figure 1 is a block diagram of an exemplary cross-platform networked system.

[0007] Figure 2 is a block diagram of another exemplary cross-platform networked system.

[0008] Figure 3 is a flow diagram of an example porting technique.

[0009] Figure 4 is a diagram of an example key exchange process for a clearinghouse embodiment.

[0010] Figure 5 is a diagram of an example key exchange process for a system link embodiment.

[0011] Figure 6 depicts an exemplary operating environment.

**DETAILED DESCRIPTION**

[0012] Figure 1 is a block diagram of an exemplary cross-platform system in which a gaming console 10 is networked to a general purpose computing device 30, via a clearinghouse 20. The gaming console 10 may be any computing device adapted for online gaming. For example, the gaming console 10 may be a personal computer (PC) loaded with one or more online game titles. The gaming console 10 may be a handheld device, and may be enabled for use with one or more game titles. A game title may be a discrete application that may be loaded and run on the gaming console 10. The game title may be stored on a medium readable by the gaming console 10, such as a CD, DVD, game cartridge, or the like.

[0013] Typically, a user may possess a number of game titles. The user may load and run an individual game title onto the gaming console 10 to play an online game with other users that have loaded and run the same game title on their respective gaming console or computing device.

[0014] The gaming console 10 is desirably a game-specific console, such as Microsoft Corporation's Xbox™ for example, and the general purpose computing device 30 is desirably a PC, handheld device, or mobile phone, for example. The clearinghouse 20 may be an online multiplayer gaming and content delivery service or system, for example, such as Xbox Live™. The gaming console 10 and the computing device 30 may each be in communication with the clearinghouse 20 via the internet or another network, for example.

[0015] The clearinghouse 20 desirably has its own network stack protocol 22 that allows for secure internet access. The network stack protocol 22, described further below,



is used by the gaming console 10, and may be ported to the computing device 30, such that the computing device 30 has code that has the same protocol as the clearinghouse 20. The protocol can talk cross-platform between the gaming console 20 and the computing device 30.

**[0016]** There is a key exchange that includes platform type (e.g., gaming console, PC, etc.) so that trust can be achieved. The platform type may be included in the key exchange message, and may be sent through the clearinghouse 20 which verifies that the device is a gaming console or a PC, for example. Key exchange is described further below.

**[0017]** Figure 2 is a block diagram of an exemplary system in which a gaming console 10 is networked to a general purpose computing device 30, not via a clearinghouse 20 as in Figure 1, but directly via a system link or local area network (LAN) connection, for example.

**[0018]** System link allows gaming consoles to connect directly to each other without having to go through the clearinghouse. The protocol for establishing the connection is different from establishing a connection to a console via the clearinghouse. Instead of going through the clearinghouse, the console sends a broadcast packet out and awaits a response. This means that system link will work on a subnet and will not work across any type of router unless it is configured to route broadcast traffic. Desirably, the broadcast packets are encrypted using a shared (per game title) key. The key structure may accommodate a flag that indicates cross-platform play.

**[0019]** Thus, a user of a software application (e.g., game or title) running at the gaming console may interact in real-time with a user of the same software application running at the computing device. As a result, a game player on a gaming console may play online with a game player on a PC.

**[0020]** Although one gaming console 10 and one computing device 30 are shown in Figures 1 and 2, it is contemplated that multiple consoles 10 and computing devices 30 may be networked together, via a clearinghouse, system link, LAN, or other means.

**[0021]** On the general purpose computing device, multiple processes may run simultaneously, such as when multiple dedicated server instances are running, or when a gaming dashboard is running as a separate application. Therefore, each process or application may need its own broadcast port and perhaps a different game data port.

**[0022]** An IP broadcast is transmitted to a specific port of the general purpose computing device, which can be title specific and possibly configured by the user. Thus, each title can default to a port of its choice or could offer the user a choice of port to use.

**[0023]** Figure 3 is a flow diagram of an example porting technique in a system link configuration. At step 200, a game title or other application is activated on a general purpose computing device. At step 210, the computing device determines which port to use. The port may be based on the game title or application that has been activated, or may be set by a user. At step 220, the computing device sets the port accordingly.

**[0024]** Ethernet level addressing is typically used in gaming console broadcasting. That is, packets are routed directly to the MAC address of the receiver below the IP level. Therefore, the address in the IP header is not used and does not matter, so it is desirably set to a particular address (e.g., 0.0.0.1) and sent from a particular port (e.g., port 3074). Thus, the port used in the computing device for cross-platform networked game play may be different from the port using on a gaming console (e.g., port 3074). It is contemplated that the port on the gaming console may vary depending on game title or user configuration, for example.

**[0025]** On a gaming console, cross-platform communication is desirably only enabled if a cross-platform privilege bit is set in the game's executable header. The bit that may be set indicates that the game will use IP addressing for system link instead of Ethernet addressing. This will allow system link between a gaming console and a general purpose computing device. If the bit is not set, the initial key exchange between the gaming console and the computing device will be rejected.

**[0026]** On the computing device, cross-platform communication is desirably always enabled.

**[0027]** To set the port to use for system link broadcasts, APIs are described that can be called from the application or game's code. The application or game can fix the port it uses in its code or expose a user interface for the user to view and modify the port used. Setting the system link port desirably does not affect which port is used for connectivity through a clearinghouse, which remains as 3074, for example, or the port negotiated with the device if it does not allow 3074.

**[0028]** An example API is:

XNetSetSystemLinkPort

```
INT WSAAPI XNetSetSystemLinkPort(
    WORD wSystemLinkPort
).
```



**[0029]** The parameter `wSystemLinkPort` refers to the port number to use for system link broadcasts and data. The API returns zero if successful, and an error code otherwise. For example, `XNetSetSystemLinkPort` fails unless the `XEX_PRIVILEGE_CROSSPLATFORM_SYSTEM_LINK` bit is set in the game's executable header. "Address in use", for example, may be returned if the port chosen is in use by the system.

**[0030]** In one example, port 3074 is the default port if `XNetSetSystemLinkPort` has not been called. Desirably, `XNetSetSystemLinkPort` is called before startup of the application or game, otherwise the application or game will fail.

**[0031]** If `XNetSetSystemLinkPort` succeeds, the set port is guaranteed to be available for binding when the application or game is started. That is, `XNetSetSystemLinkPort` locks the port for use so there is no possibility the port being retained by another process between calling `XNetSetSystemLinkPort` and the startup of the application or game.

**[0032]** The port number is not persisted to disk and is stored globally for use until the application or game ends. When an application or game quits, the system link port is reset (e.g., back to 3074).

**[0033]** Another API is described that gets the system link port:

`XNetGetSystemLinkPort`

```
INT WINAPI XNetGetSystemLinkPort(
    WORD* pwSystemLinkPort
);
```

**[0034]** The parameter `pwSystemLinkPort` is the port number currently set for system link broadcasts and data.

**[0035]** The API returns zero if successful, and an error code otherwise. `XNetGetSystemLinkPort` fails unless the `XEX_PRIVILEGE_CROSSPLATFORM_SYSTEM_LINK` bit is set in the game's executable header.

**[0036]** Secure key exchange is desirably performed. For embodiments which use a clearinghouse, an example connection protocol desirably requires that all connections



between the clients (gaming consoles and general purpose computing devices) require a key exchange. Packets are inspected to determine whether they were from a general purpose computing device. If so, that information is indicated to the clearinghouse or other destination so that it can handle the data differently, if desired.

**[0037]** Figure 4 is a diagram of an example key exchange process for a clearinghouse embodiment, and shows a client 400, such as a game client, in communication with an authentication service 410, a clearinghouse key server 420, and a clearinghouse gateway server 430.

**[0038]** The game client 400 sends login credentials 405 to the authentication service 410. The authentication service 410 responds to the client 400 with an authentication ticket 415. The game client 400 presents the authentication ticket 415 to the clearinghouse key server 420. The clearinghouse key server 420 responds with a clearinghouse ticket 425 and a pointer 427 to the clearinghouse gateway server 430. The game client 400 may then connect to the clearinghouse gateway server 430 to use the clearinghouse services.

**[0039]** The key exchange messages desirably include the port number that the machine running the game should use to connect to the clearinghouse.

**[0040]** In embodiments that do not use a clearinghouse (e.g., the system link embodiments), such as that described with respect to Figure 5, key exchange desirably uses a flag in the key that indicates a cross-platform connection.

**[0041]** Figure 5 is a diagram of an example key exchange process for a system link embodiment. Figure 5 shows a game client 500 that acts as a host and another game client 550. The clients may be in communication over a LAN subnet, for example. Additional game clients are contemplated on the network, although they are not shown.

**[0042]** The host game client 500 broadcasts its existence and availability 505. The game client 550 decrypts the broadcast and responds with a broadcast to join the game 510. The host game client 500 accepts the join request and broadcasts the game session key 515. The game clients 500, 550 desirably unicast with each other 520 to carry out the game session. In computer networks, unicast is the sending of information packets to a single destination. Desirably, all broadcasts are encrypted with the title specific key, and all unicasts are encrypted with the game session specific key.

**[0043]** Regarding byte ordering, in computing, endianness is the ordering used to represent some kind of data as a sequence of smaller units. Typical cases are the order in which integer values are stored as bytes in computer memory (relative to a given memory

addressing scheme) and the transmission order over a network. Regarding bytes, endianness is also referred to as byte order.

**[0044]** Most computer processors simply store integers as sequences of bytes, so that, conceptually, the encoded value can be obtained by simple concatenation. For an  $n$ -byte integer value this allows  $n!$  possible representations (one for each byte permutation). The two most common of them are increasing numeric significance with increasing memory addresses, known as little-endian, and its opposite, called big-endian.

**[0045]** All computer architectures are big-endian or little-endian. Big-endian architectures are found in Microsoft's Xbox and IBM's Power PC, for example. Intel x86 processors (and their clones) use the little-endian format.

**[0046]** Figure 6 and the following discussion are intended to provide a brief general description of a suitable computing environment in which the present invention and/or portions thereof may be implemented. Although not required, the invention is described in the general context of computer-executable instructions, such as program modules, being executed by a computer, such as a client workstation or a server. Generally, program modules include routines, programs, objects, components, data structures and the like that perform particular tasks or implement particular abstract data types. Moreover, it should be appreciated that the invention and/or portions thereof may be practiced with other computer system configurations, including hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

**[0047]** As shown in Figure 6, an exemplary general purpose computing system includes a conventional personal computer 120 or the like, including a processing unit 121, a system memory 122, and a system bus 123 that couples various system components including the system memory to the processing unit 121. The system bus 123 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read-only memory (ROM) 124 and random access memory (RAM) 125. A basic input/output system 126 (BIOS), containing the basic routines that help to transfer



information between elements within the personal computer 120, such as during start-up, is stored in ROM 124.

**[0048]** The personal computer 120 may further include a hard disk drive 127 for reading from and writing to a hard disk (not shown), a magnetic disk drive 128 for reading from or writing to a removable magnetic disk 129, and an optical disk drive 130 for reading from or writing to a removable optical disk 131 such as a CD-ROM or other optical media. The hard disk drive 127, magnetic disk drive 128, and optical disk drive 130 are connected to the system bus 123 by a hard disk drive interface 132, a magnetic disk drive interface 133, and an optical drive interface 134, respectively. The drives and their associated computer-readable media provide non-volatile storage of computer readable instructions, data structures, program modules and other data for the personal computer 120.

**[0049]** Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 129, and a removable optical disk 131, it should be appreciated that other types of computer readable media which can store data that is accessible by a computer may also be used in the exemplary operating environment. Such other types of media include a magnetic cassette, a flash memory card, a digital video disk, a Bernoulli cartridge, a random access memory (RAM), a read-only memory (ROM), and the like.

**[0050]** A number of program modules may be stored on the hard disk, magnetic disk 129, optical disk 131, ROM 124 or RAM 125, including an operating system 135, one or more application programs 136, other program modules 137 and program data 138. A user may enter commands and information into the personal computer 120 through input devices such as a keyboard 140 and pointing device 142. Other input devices (not shown) may include a microphone, joystick, game pad, satellite disk, scanner, or the like. These and other input devices are often connected to the processing unit 121 through a serial port interface 146 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port, or universal serial bus (USB). A monitor 147 or other type of display device is also connected to the system bus 123 via an interface, such as a video adapter 148. In addition to the monitor 147, a personal computer typically includes other peripheral output devices (not shown), such as speakers and printers. The exemplary system of Figure 6 also includes a host adapter 155, a Small Computer System Interface (SCSI) bus 156, and an external storage device 162 connected to the SCSI bus 156.

**[0051]** The personal computer 120 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 149. The

remote computer 149 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the personal computer 120, although only a memory storage device 150 has been illustrated in Figure 6. The logical connections depicted in Figure 6 include a local area network (LAN) 151 and a wide area network (WAN) 152. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the internet.

**[0052]** When used in a LAN networking environment, the personal computer 120 is connected to the LAN 151 through a network interface or adapter 153. When used in a WAN networking environment, the personal computer 120 typically includes a modem 154 or other means for establishing communications over the wide area network 152, such as the internet. The modem 154, which may be internal or external, is connected to the system bus 123 via the serial port interface 146. In a networked environment, program modules depicted relative to the personal computer 120, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

**[0053]** Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.



**What is Claimed:**

1. A cross-platform system, comprising:

a first computing device having a first platform running an application (10); and

a second computing device having a second platform running the application and in communication with the first computing device (30), the second platform being different from the first platform, wherein the first computing device and the second computing device are both operable to interact online in real-time with the application.

2. The system of claim 1, wherein the application is a game title (200).

3. The system of claim 1, wherein the first computing device is a gaming console (10) and the second computing device is a general purpose computing device (30).

4. The system of claim 3, wherein the gaming console has a network stack protocol which is ported to the general purpose computing device (22, 23).

5. The system of claim 1, wherein the first computing device and the second communication device are in communication with each other directly via a network (520).

6. The system of claim 1, wherein the first computing device and the second communication device are in communication with each other directly via a clearinghouse (20).

7. The system of claim 6, wherein the clearinghouse comprises a multiplayer gaming and content delivery service or system (420, 430).

8. The system of claim 6, wherein the clearinghouse is operable to verify the type of the first computing device and the type of the second computing device (420).

9. A method of cross-platform communication, comprising:

a first computing device having a first platform running an application (10) connecting to a clearinghouse (20); and

a second computing device having a second platform running the application (30) connecting to the clearinghouse (20), the second platform being different from the first platform, wherein the first computing device and the second computing device are both operable to interact online in real-time with the application.

10. The method of claim 9, further comprising authenticating the first computing device prior to connecting to the clearinghouse (420).

11. The method of claim 10, wherein authenticating the first computing device comprises:

the first computing device logging into an authentication server and a key server (420); and

providing a clearinghouse ticket (425) and pointer (427) to the first computing device.

12. The method of claim 11, wherein the connecting (400, 430) is performed responsive to the clearinghouse ticket (425) and the pointer (427).

13. The method of claim 9, wherein the application is a game title (200).

14. The method of claim 9, wherein the first computing device is a gaming console (10) and the second computing device is a general purpose computing device (30).

15. The method of claim 9, wherein the clearinghouse comprises a multiplayer gaming and content delivery service or system (420, 430).



16. A method of cross-platform communication, comprising:

broadcasting an availability request (505) from a first computing device (500) to a second computing device (550), the first computing device having a first platform and the second computing device having a second platform different from the first platform;

directly connecting the first computing device and the second computing device (515); and

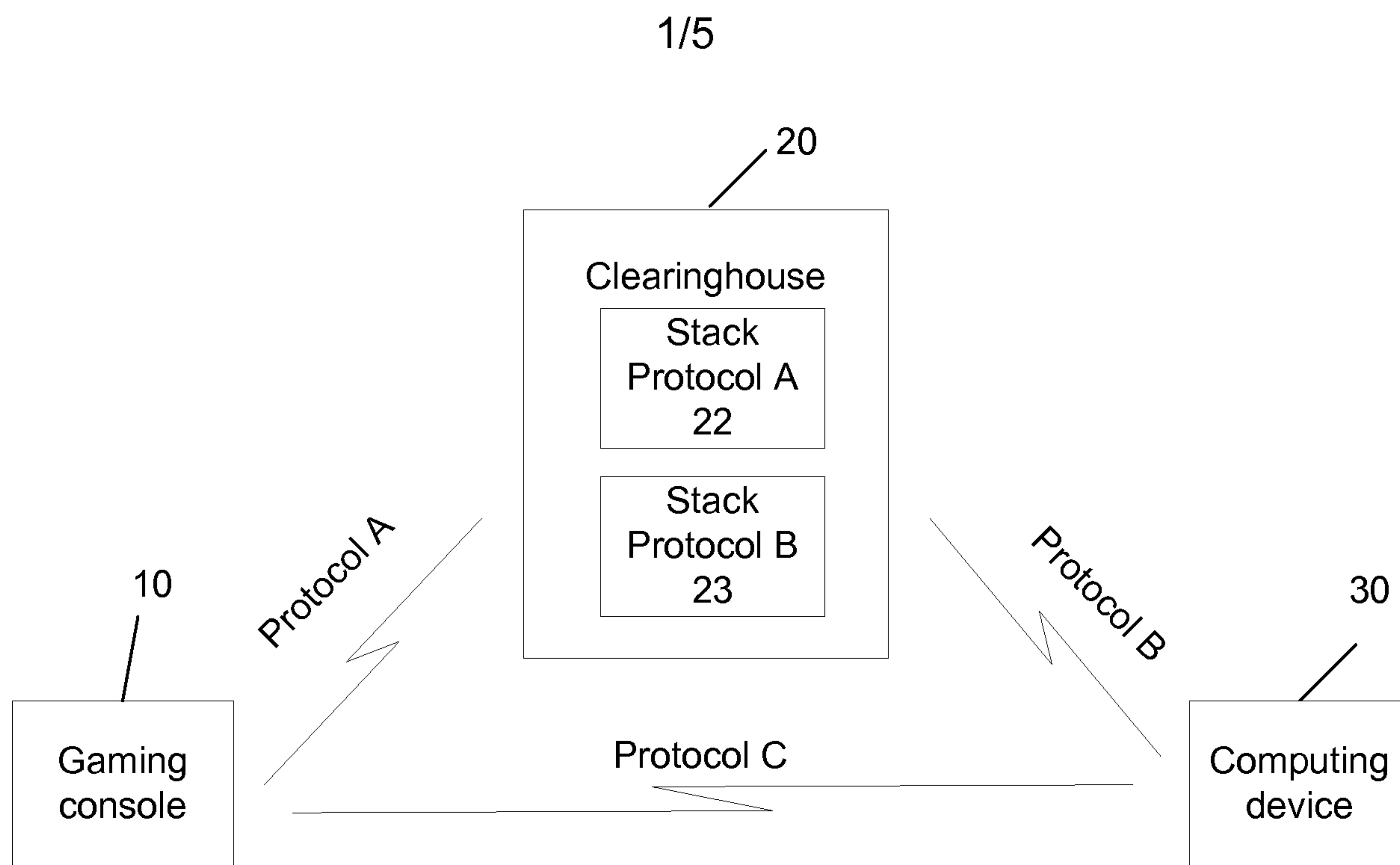
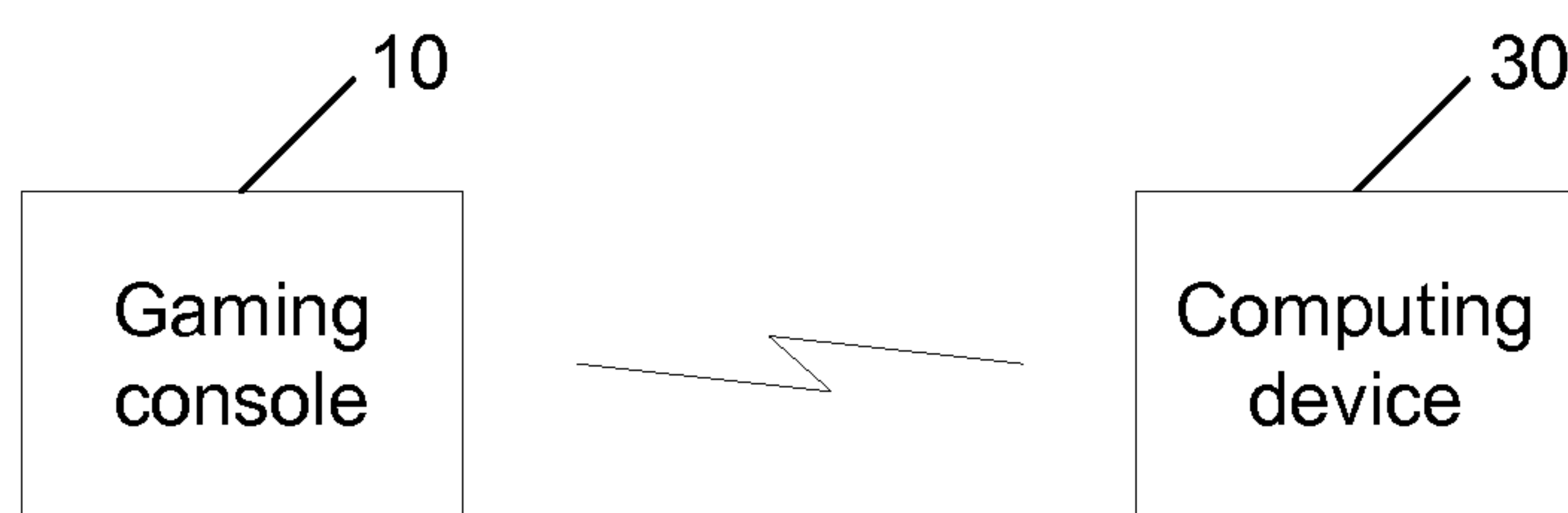
running an application on the first computing device and the second computing device such that the first computing device and the second computing device are both operable to interact online in real-time with the application (520).

17. The method of claim 16, wherein the first computing device transmits a session key to the second computing device prior to connecting the first computing device and the second computing device (515).

18. The method of claim 16, wherein the application is a game title (510).

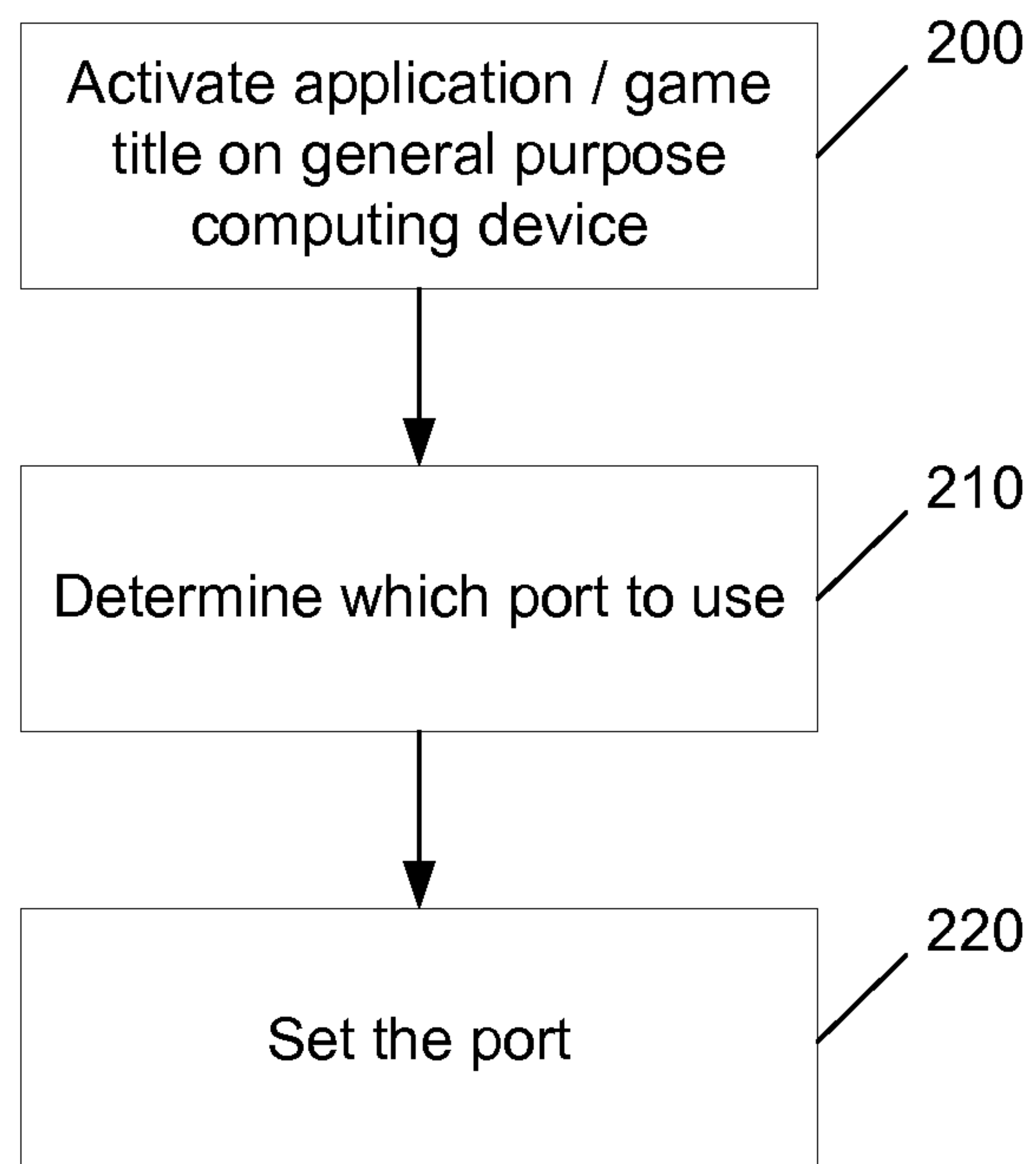
19. The method of claim 16, wherein the first computing device is a gaming console (10) and the second computing device is a general purpose computing device (30).

20. The method of claim 16, wherein the first computing device and the second computing device unicast with each other (520).

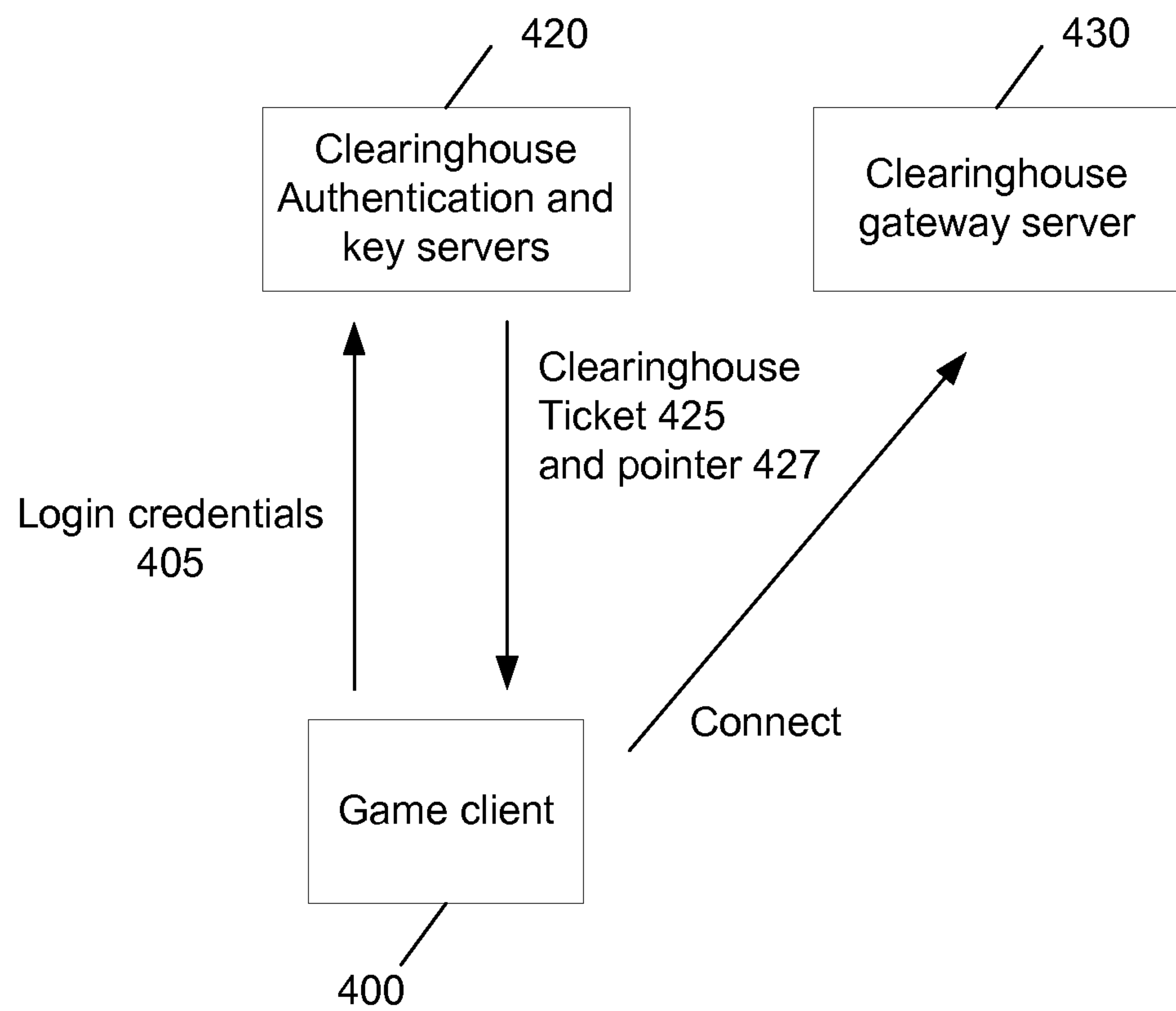
**FIG. 1****FIG. 2**



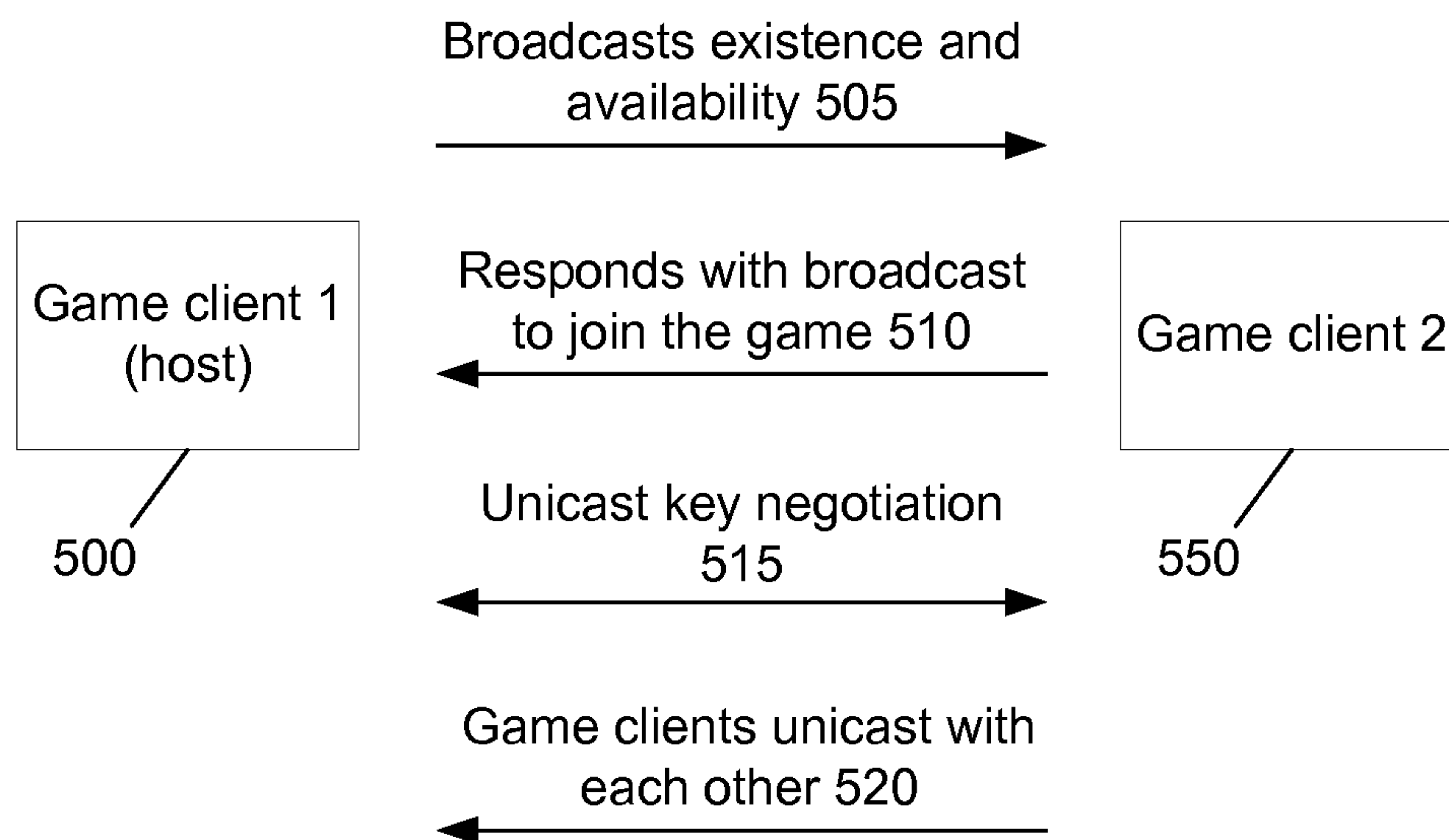
2/5

**FIG. 3**

3/5

**FIG. 4**

4/5

**FIG. 5**



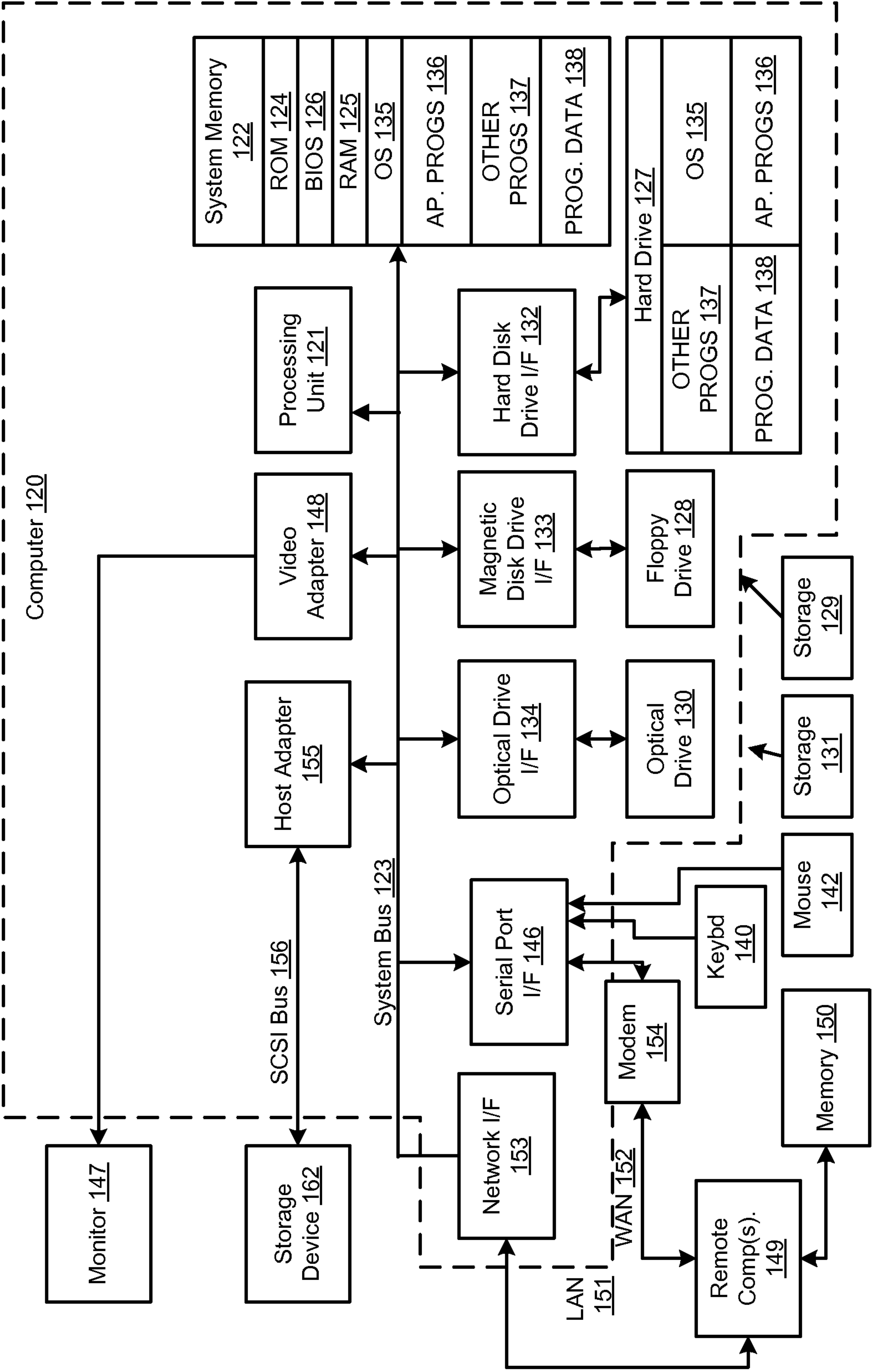
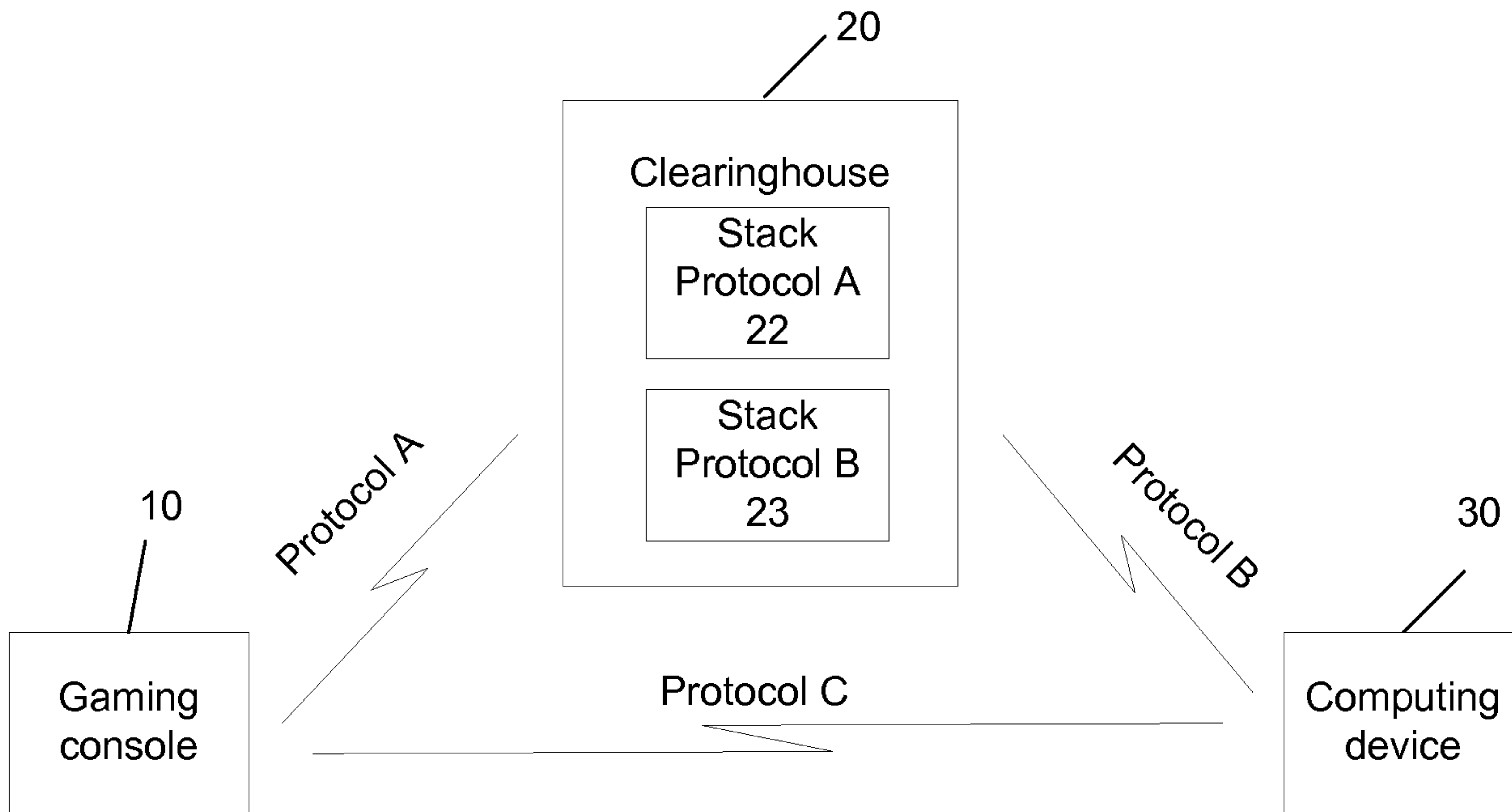


FIG. 6



**FIG. 1**